Dissertation

Zur Erlangung des akademischen Grades

Doctor rerum naturalium

(Dr. rer. nat.)

Computational analysis and prediction of RNA-protein interactions

Albert-Ludwigs-Universität Freiburg

Technische Fakultät

Institut für Informatik

Michael Uhl

Dipl. Biologe

M.Sc. Bioinformatik

Dekan

Prof. Dr. Roland Zengerle

Gutachter

Prof. Dr. Rolf Backofen

Prof. Dr. Wolfgang Hess

Beisitz und Vorsitz

Prof. Dr. Frank Hutter

Prof. Dr. Gerald Urban

Datum der Promotion

06.12.2022

Acknowledgements

I would like to take this time and thank the people who were involved with and supported me in my quest for PhD.

First of all, I would like to thank my supervisor, Prof. Dr. Rolf Backofen, for sparking my interest in this fascinating world we call RNA bioinformatics, and subsequently welcoming me as a PhD student in his group. I'm deeply grateful for his continuous support, help, and guidance throughout my PhD. Thank you Rolf.

I want to thank my second advisor Prof. Dr. Wolfgang Hess for accepting this task, and his time and effort to review my PhD thesis. I'm also extremely grateful to Prof. Dr. Frank Hutter and Prof. Dr. Gerald Urban, for taking their time to be the chairperson and the observer for my thesis.

I would like to thank all the people I was fortunate to collaborate with during my scientific studies: thank you for doing science together.

Of course there are lots of thanks due to the group I was lucky to spent my PhD time in: to all the recent and former members I had the chance to work, talk and hang out with, thank you. Special thanks for their special support go to: Sita, Martin, Pavan, Egg, Björn, Torsten, Dinh, Fabrizio, and of course Monika. All the others too numerous to mention here, I haven't forgotten you, thank you again for being there.

To my family: you have been a die-hard fan of mine ever since. I could always rely on you, thank you for everything.

Last but not least: to Teresa, who made it possible, with love. Thank you.

We have met the enemy and he is us.
— Pogo

Contents

Abstract

This dissertation is about the computational analysis and prediction of RNA-protein interactions. Ribonucleic acids (RNAs) and proteins both are essential for the control of gene expression in our cells. Gene expression is the process by which a functional gene product, namely a protein or an RNA, is produced from a gene, starting from the gene region on the DNA with the transcription of an RNA. Once regarded primarily as a messenger to transmit the protein information, recent years have seen RNA moving further into the biomedical spotlight, thanks to its increasingly uncovered roles in regulating gene expression. In addition, RNA has showcased its therapeutic potential, as famously demonstrated by the groundbreaking success of RNA vaccines in the COVID-19 pandemic. However, RNAs rarely function on their own: In humans, more than 1,500 different RNA-binding proteins (RBPs) are involved in controlling the various stages of an RNA's life cycle, creating a highly complex regulatory interplay between RNAs and proteins. It is therefore of fundamental importance to study these RNA-protein interactions, in order to deepen our understanding of gene expression.

Over the last decade, CLIP-seq has become the dominant experimental method to identify the set of cellular RNA binding sites for an RBP of interest. However, analysing the resulting CLIP-seq data can be challenging, as there are many analysis steps and CLIP-seq protocol variants available, each requiring specific adaptations to the analysis workflow. Consequently, there is a need for analysis guidelines, providing easy access to tools, as well as the constant improvement of tools and workflows to increase the accuracy of the analysis results.

The first set of works included in this thesis (publications P1, P4, and P5) deals with these topics, by providing a review article on CLIP-seq data analysis, as well as two articles on how to further improve CLIP-seq data analysis. Publication P1 supplies readers with an overview of tools and protocols, as well as guidelines to conduct a successful analysis, drawing largely from our own experience with analysing CLIP-seq data. Publication P4 demonstrates the issues current binding site identification tools have with CLIP-seq data from RBPs that bind to processed RNAs, and that the integration of RNA processing information improves the resulting binding site quality. On top of this, publication P5 presents Peakhood, the first tool that utilizes RNA processing information in order to increase the quality of RBP binding sites identified from CLIP-seq data.

A natural drawback of experimental methods is that a target RNA needs to be sufficiently expressed in the observed cells for an RNA-protein interaction to be detected. Hence, since gene expression is a dynamic process that differs between cell types, time points, and conditions, a CLIP-seq experiment cannot recover the complete set of cellular RBP binding sites. This creates a demand for computational methods which can learn the binding properties of an RBP from existing CLIP-seq data, in order to predict RBP binding sites on any given target RNA. Besides interacting with proteins, RNAs can also interact with other RNAs, further increasing the amount of possible regulatory interactions between RNAs and pro-

teins. In this regard, long non-coding RNAs (lncRNAs), a large class of non-protein-coding RNAs whose functions are still vastly unexplored, have become especially important, as it has been shown that they can engage in RNA-RNA interactions, whose regulatory mechanisms also include RNA-protein interactions. As such mechanistic studies are typically slow and expensive, computational tools that combine RNA-protein and RNA-RNA interaction predictions to infer potential mechanisms could be of great help, e.g., by screening a set of target RNAs and proteins and suggesting plausible mechanisms for experimental validation.

The second set of works included in this thesis (publications P2 and P3) thus deals with the computational prediction of RNA-protein interactions, RNA-RNA interactions and the functional mechanisms that can be inferred from these interactions. Publication P2 introduces MechRNA, the first tool to infer functional mechanisms of lncRNAs based on their predicted interactions with RBPs and other RNAs, as well as gene expression data. We demonstrated MechRNA's capability to identify formerly described lncRNA mechanisms and experimentally validated one prediction, underlining its value for functional lncRNA studies. Finally, publication P3 presents RNAProt, a flexible and performant RBP binding site prediction tool based on recurrent neural networks. Compared to other popular deep learning methods, RNAProt achieves state-of-the-art predictive performance, as well as superior runtime efficiency. In addition, it is more feature-rich than any other available method, including the support of user-defined predictive features. We further showed that its visualizations agree with known RBP binding preferences, and demonstrated that its additional predictive features can increase the specificity of predictions.

Zusammenfassung

Diese Dissertation beschäftigt sich mit der computergestützten Analyse und Vorhersage von RNA-Protein-Interaktionen. Ribonukleinsäuren (RNAs) und Proteine sind essentielle Bestandteile der Genexpressionskontrolle in den Zellen unseres Körpers. Genexpression bezeichnet den Prozess der Herstellung eines funktionellen Genprodukts, welches ein Protein oder eine RNA sein kann, angefangen mit der Transkription einer RNA von der betreffenden Genregion auf der DNA. In den letzten Jahren hat sich unser ursprüngliches Bild der RNA als Überträger der Proteininformation erheblich erweitert: Diverse Forschungsarbeiten haben zahlreiche neue RNA-Funktionen bei der Regulierung der Genexpression offengelegt, wodurch sich der wissenschaftliche Fokus in der biomedizinischen Forschung weiter in Richtung RNA verschoben hat. Darüber hinaus hat der bahnbrechende Erfolg der RNA-Impfstoffe in der COVID-19 Pandemie auf beeindruckende Weise das therapeutische Potential von RNA aufgezeigt. RNAs führen ihre Funktionen jedoch in den seltensten Fällen alleine aus: Mehr als 1500 RNA-Bindeproteine (RBPs) sind im Menschen an der Kontrolle der verschiedenen Phasen des RNA-Lebenszyklus beteiligt, was zu einem hochkomplexen regulatorischen Zusammenspiel zwischen RNA und Proteinen führt. Es ist daher von grundlegender Bedeutung, diese RNA-Protein-Interaktionen zu untersuchen, um ein tieferes Verständnis der Genexpression zu erlangen.

Im Laufe des letzten Jahrzehnts hat sich CLIP-seq als experimentelle Methode zur Identifizierung der zellulären RNA-Bindestellen eines bestimmten RBPs durchgesetzt. Die Analyse der resultierenden CLIP-seq-Daten ist jedoch alles andere als trivial, da sie ein fundiertes Wissen über die zahlreichen Analyseschritte und die unterschiedlichen CLIP-seq-Protokolle voraussetzt. Es ist daher notwendig, dem Anwender Anleitungen und Programme für die einzelnen Analyseschritte und Protokollvarianten zur Verfügung zu stellen. Ebenso wichtig ist die kontinuerliche Verbesserung der Programme und Workflows, um die Qualität der Analyseergebnisse weiter zu erhöhen.

Die ersten drei in dieser Dissertation enthaltenen Publikationen (Publikationen P1, P4 und P5) behandeln diese Themen: Publikation P1 ist ein Übersichtsartikel zur Analyse von CLIP-seq-Daten, der die wichtigsten Analyseschritte, Protokolle und Programme beschreibt, mit dem Ziel, dem Leser eine erfolgreiche Datenanalyse zu ermöglichen. Die enthaltenen Anleitungen basieren dabei weitgehend auf unseren eigenen Erfahrungen mit der Analye von CLIP-seq-Daten. Publikation P4 stellt die Probleme aktueller Programme zur Identifizierung von Bindestellen dar, wenn die CLIP-seq-Daten von RBPs stammen die an prozessierte RNAs binden. Weiterhin zeigen wir, dass die Integration von Informationen zur RNA-Prozessierung die Qualität der resultierenden Bindestellen verbessert. Darauf aufbauend präsentieren wir in Publikation P5 Peakhood, das erste Programm welches Informationen zur RNA-Prozessierung benutzt um die Qualität der aus CLIP-seq-Daten ermittelten RBP-Bindestellen zu erhöhen.

Ein offensichtlicher Nachteil experimenteller Methoden ist, dass diese auf eine ausreichend

hohe Expression der RNA angewiesen sind, um die sich darauf befindlichen RBP-Bindestellen detektieren zu können. Da die Genexpression dynamisch ist und deshalb unterschiedlich ausfällt zwischen verschiedenen Zelltypen, Zeitpunkten und Konditionen, kann ein CLIP-seq-Experiment folglich niemals den kompletten Satz an zellulären RBP-Bindestellen ermitteln. Dies führt zu einem Bedarf an computergestützten Methoden, welche die Bindeeigenschaften eines RBP aus existierenden CLIP-seq-Daten lernen können, um damit neue RBP-Bindestellen auf beliebigen RNAs vorherzusagen. Neben der Interaktion mit Proteinen können RNAs auch mit anderen RNAs interagieren, wodurch sich die Anzahl der möglichen regulatorischen Interaktionen zwischen RNAs und Proteinen nochmals deutlich erhöht. In diesem Zusammenhang sind vor allem lange nicht-kodierende RNAs (lncRNAs) zu nennen, eine große noch weitgehend unerforschte Klasse nicht-proteinkodierender RNAs, da gezeigt werden konnte, dass diese RNA-RNA-Interaktionen ausbilden können, deren regulatorische Mechanismen auch RNA-Protein-Interaktionen mit einbeziehen. Diese mechanistischen Studien sind allerdings mit einem erheblichen Zeit- und Kostenaufwand verbunden. Dementsprechend entsteht ein Bedarf an computergestützten Methoden zur Vorhersage potentieller Mechanismen anhand von vorausberechneten RNA-Protein- und RNA-RNA-Interaktionen. Diese dienen dann beispielsweise zur Vorauswahl plausibler Mechanismen, welche anschließend experimentell validiert werden können.

Die restlichen zwei in dieser Dissertation enthaltenen Publikationen (Publikationen P2 und P3) befassen sich deshalb mit der computergestützten Vorhersage von RNA-Protein-Interaktionen, RNA-RNA-Interaktionen, sowie den funktionellen Mechanismen, die sich aus diesen Interaktionen ableiten lassen. In Publikation P2 stellen wir MechRNA vor, das erste Programm zur Vorhersage funktioneller Mechanismen von lncRNAs, abgeleitet aus vorausberechneten Interaktionen der lncRNA mit RBPs und anderen RNAs sowie aus Genexpressionsdaten. Wir zeigen dass MechRNA in der Lage ist, bekannte lncRNA-Mechanismen zu identifizieren. Ebenso konnten wir eine Vorhersage erfolgreich experimentell validieren, was nochmals den Wert des Programms für funktionelle lncRNA-Studien unterstreicht. Schließlich präsentieren wir in Publikation P3 RNAProt, ein flexibles und leistungsfähiges Programm zur Vorhersage von RBP-Bindestellen, basierend auf rekurrenten neuronalen Netzen. Im Vergleich zu anderen populären Deep-Learning-Methoden bietet RNAProt sowohl eine überragende Vorhersageleistung als auch eine überlegene Laufzeiteffizienz. Darüber hinaus bietet das Programm mehr Funktionen als jede andere verfügbare Methode, einschließlich der Unterstützung benutzerdefinierter Vorhersage-Features. Zudem haben wir gezeigt, dass die in RNAProt enthaltenen Visualisierungen mit bekannten RBP-Bindepräferenzen übereinstimmen, und dass die zusätzlichen Vorhersage-Features von RNAProt die Spezifität der Vorhersagen weiter erhöhen können.

List of publications

This thesis covers the following five publications (P1-P5):

- [P1] **Michael Uhl***, Torsten Houwaart*, Gianluca Corrado, Patrick R. Wright, and Rolf Backofen*. **Computational analysis of CLIP-seq data**. *Methods*, 2017.
- [P2] Alexander R. Gawronski, **Michael Uhl**, Yajia Zhang, Yen-Yi Lin, Yashar S. Niknafs, Varune R. Ramnarine, Rohit Malik, Felix Feng, Arul M. Chinnaiyan, Colin C. Collins, S. Cenk Sahinalp, and Rolf Backofen. **MechRNA: prediction of lncRNA mechanisms from RNA-RNA and RNA-protein interactions**. *Bioinformatics*, 2018.
- [P3] **Michael Uhl**, Van Dinh Tran, Florian Heyl, and Rolf Backofen. **RNAProt: an efficient and feature-rich RNA binding protein binding site predictor**. *GigaScience*, 2021.
- [P4] **Michael Uhl**, Van Dinh Tran, and Rolf Backofen. **Improving CLIP-seq data analysis by incorporating transcript information**. *BMC Genomics*, 2020.
- [P5] **Michael Uhl**, Dominik Rabsch, Florian Eggenhofer, and Rolf Backofen. **Peakhood: individual site context extraction for CLIP-seq peak regions**. *Bioinformatics*, 2021.

The five publications are ordered by publication date, except for P4, which was published before P3. The two publications were shifted since P5 directly builds on the findings established in P4.

Further publications to which I contributed to:

- Florian Heyl, Daniel Maticzka, **Michael Uhl**, and Rolf Backofen. **Galaxy CLIP-Explorer: a web server for CLIP-Seq data analysis**. *GigaScience*, 2020.
- Martin Raden, Fabio Gutmann, **Michael Uhl**, and Rolf Backofen. **CopomuS - Ranking Compensatory Mutations to Guide RNA-RNA Interaction Verification Experiments**. *International Journal of Molecular Sciences*, 2020.
- Yajia Zhang, Sethuramasundaram Pitchiaya, Marcin Cieslik, Yashar S. Niknafs, Jean C.-Y. Tien, Yasuyuki Hosono, Matthew K. Iyer, Sahr Yazdani, Shruthi Subramaniam, Sudhanshu K. Shukla, Xia Jiang, Lisha Wang, Tzu-Ying Liu, **Michael Uhl**, Alexander R. Gawronski, Yuanyuan Qiao, Lanbo Xiao, Saravana M. Dhanasekaran, Kristin M. Juckette, Lakshmi P. Kunju, Xuhong Cao, Utsav Patel, Mona Batish, Girish C.

* joint first authors

Shukla, Michelle T. Paulsen, Mats Ljungman, Hui Jiang, Rohit Mehra, Rolf Backofen, Cenk S. Sahinalp, Susan M. Freier, Andrew T. Watt, Shuling Guo, John T. Wei, Felix Y. Feng, Rohit Malik, and Arul M. Chinnaiyan. **Analysis of the androgen receptor-regulated lncRNA landscape identifies a role for ARLNC1 in prostate cancer progression**. *Nature Genetics*, 2018.

- Michael Briese, Lena Saal-Bauernschubert, Changhe Ji, Mehri Moradi, Hanaa Ghanawi, **Michael Uhl**, Silke Appenzeller, Rolf Backofen, and Michael Sendtner. **hnRNP R and its main interactor, the noncoding RNA 7SK, coregulate the axonal transcriptome of motoneurons**. *PNAS*, 2018.

- Michael Daume, **Michael Uhl**, Rolf Backofen, and Lennart Randau. **RIP-Seq Suggests Translational Regulation by L7Ae in *Archaea***. *mBio*, 2017.

- Giuseppe Nicastro, Adela M. Candel, **Michael Uhl**, Alain Oregioni, David Hollingworth, Rolf Backofen, Stephen R.Martin, and Andres Ramos. **Mechanism of β-actin mRNA Recognition by ZBP1**. Cell Reports, 2017.

- Yashar S. Niknafs, Sumin Han, Teng Ma, Corey Speers, Chao Zhang, Kari Wilder-Romans, Matthew K. Iyer, Sethuramasundaram Pitchiaya, Rohit Malik, Yasuyuki Hosono, John R. Prensner, Anton Poliakov, Udit Singhal, Lanbo Xiao, Steven Kregel, Ronald F. Siebenaler, Shuang G. Zhao, **Michael Uhl**, Alexander Gawronski, Daniel F. Hayes, Lori J. Pierce, Xuhong Cao, Colin Collins, Rolf Backofen, Cenk S. Sahinalp, James M. Rae, Arul M. Chinnaiyan, and Felix Y. Feng. **The lncRNA landscape of breast cancer reveals a role for DSCAM-AS1 in breast cancer progression**. *Nature Communications*, 2016.

- Yasuhiro Murakawa, Michael Hinz, Janina Mothes, Anja Schuetz, **Michael Uhl**, Emanuel Wyler, Tomoharu Yasuda, Guido Mastrobuoni, Caroline C. Friedel, Lars Dölken, Stefan Kempa, Marc Schmidt-Supprian, Nils Blüthgen, Rolf Backofen, Udo Heinemann, Jana Wolf, Claus Scheidereit, and Markus Landthaler. **RC3H1 post-transcriptionally regulates A20 mRNA and modulates the activity of the IKK/NF-κB pathway**. *Nature Communications*, 2015.

- **Michael Uhl**, Kevin Mellert, Britta Striegl, Martin Deibler, Markus Lamla, Joachim P. Spatz, Ralf Kemkemer, and Dieter Kaufmann. **Cyclic stretch increases splicing noise rate in cultured human fibroblasts**. *BMC Research Notes*, 2011.

- Kevin Mellert, **Michael Uhl**, Josef Högel, Markus Lamla, Ralf Kemkemer, and Dieter Kaufmann. **Aberrant Single Exon Skipping is not Altered by Age in Exons of *NF1*, *RABAC1*, *AATF* or *PCGF2* in Human Blood Cells and Fibroblasts**. *Genes*, 2011.

Chapter 1

Introduction

1.1 Motivation

Throughout evolution, life on earth has taken on a stunning variety of forms. Still, their common ancestry has preserved the fundamental aspects of life in every single living organism: All organisms are made up of cells. In addition, cells share many fundamental biochemical processes, as well as the same biomolecules responsible for the flow of genetic information inside the cell: the genetic information is stored in DNA, which gets transcribed into RNA, and RNA is translated into protein. The units of genetic information are known as genes, while the sum of the genetic information of a particular organism is also referred to as genome. Proteins thus execute the gene function, while RNA can either be protein-coding or non-coding, meaning that an RNA molecule either acts as a messenger by coding for a protein, or executes the gene function on its own. As a consequence, genes can be further divided into protein-coding and non-(protein-)coding genes.

There are approximately 20,000 protein-coding genes in the human genome, with less than 2% of the human genome being transcribed into protein-coding RNA [1]. The roughly three-billion-letter long DNA sequence of the human genome was famously first determined by an enormous publicly-funded international research effort known as the Human Genome Project (HGP) (1990 - 2003). Before the finishing stage of the HGP in the early 2000s, little attention was paid to the non-coding genome fraction, and scientists expected the number of protein-coding genes to be 80,000 or more [2]. This changed with the release of the initial human genome draft sequence in 2001, suddenly narrowing down the estimate to 30,000-40,000 [3]. The steep drop was especially puzzling to scientists because: (i) the numbers of protein-coding genes became more and more similar to the ones of much less complex organisms, such as the well-studied roundworm *Caenorhabditis elegans* (genome sequencing finished in 1998); and (ii) prior to the HGP, the functional elements in the genome were thought to be almost exclusively (apart from some well-studied non-coding RNA classes) protein-coding genes, while the non-coding fraction was considered to be mostly non-functional, appropriately termed "junk DNA". The finding propelled a shift of focus towards studying the functions of the non-coding regions of DNA, consequently dropping the "junk DNA" designation in favor of the more promising "dark matter of the genome" [4].

In 2003, the Encyclopedia of DNA Elements (ENCODE) consortium was founded, with the goal to identify all functional elements (i.e., DNA regions with some biochemical function) in the human genome [5]. Their first set of comprehensive results was published in 2012

over various articles and journals, concluding that functional elements could make up an astounding 80% of the human genome. Besides protein-coding genes, these include elements that regulate the transcription of genes, either on the DNA level or by transcribing regulatory RNAs [6]. The activity of a functional element is further controlled by various reversible chemical modifications (e.g., DNA or histone methylations), which together form the subject of epigenetics. Interestingly, epigenetic modifications can also be inherited and are sensitive to environmental stimuli, although mechanistic evidence in humans is still sparse [7]. The set of functional elements and their regulatory interplay forms our current understanding of the genome, which can be described as a dynamic and reactive system of interacting molecules (DNA, RNA, and proteins) to orchestrate gene expression.

Based on the observed pervasive transcription of regulatory RNAs, many new non-coding genes were annotated, with recent estimates even surpassing the number of protein-coding genes [8]. This has led to an increased and ongoing effort in elucidating their functions, further contributing to the popularity of RNA studies. Once seen primarily as a messenger, RNA is now recognized as a central player involved in all major biochemical processes. This is due to RNA's versatility regarding interaction partners (proteins, RNA, and DNA), as well as its diversity (both in numbers and functions): the total number of transcribed RNAs (i.e., the transcriptome) in humans is much higher than the number of genes, as different RNAs can be produced from the same gene (see section 2.1.4 *Splicing*). Moreover, numerous non-coding RNA classes exist besides protein-coding RNAs, often with distinct cellular functions (see section 2.1.4 *Classes and functions*).

Regulatory potential is especially rich between RNAs and the more than 1,500 RNA-binding proteins (RBPs) encoded in the human genome, which can recognize and bind specific sequence or structural elements of RNA molecules [9]. Consequently, RBPs of various functions interact with target RNAs at different positions (i.e., binding sites), time points, and cellular locations. This leads to a complex regulatory interplay, with changing RNA-protein complex compositions throughout the RNA life cycle. RBPs are thus involved in all stages of the cycle, essentially controlling the processing, localization, stability, translation, and decay of an RNA molecule. In addition, RNAs can sequester proteins to other RNAs through RNA-RNA interactions, further expanding the set of possible regulatory mechanisms [10]. Given their fundamental roles in post-transcriptional gene regulation, it is not surprising that RBPs have also been implicated in various diseases, such as genetic disorders, neurodegeneration, and cancer [11, 12, 13]. All this underlines the importance of studying RNA-protein interactions and their functional characterization.

Early genome sequencing in the 1990s was a tremendously labor-intensive and time-consuming effort: back then state-of-the art capillary sequencers could only determine the sequences of less than 100 small DNA fragments in one run [14]. As a result, the HGP relied on huge factory-like sequencing centers, equipped with hundreds of sequencers to speed up the sequencing and meet its ambitious time schedule [15]. This changed in the mid 2000s with the introduction of so called next-generation sequencing (NGS) technologies. While similar in concept to previous approaches, these methods achieve a much higher throughput

by simultaneously sequencing millions of DNA fragments in a single machine run [16]. The drastically reduced runtimes and costs made genome-scale sequencing accessible to more and more scientists, with machines and methods becoming affordable also for smaller research institutions. Adaptations for the high-throughput sequencing of RNA (i.e., RNA-seq) quickly followed, allowing the sequencing of whole transcriptomes to measure and compare gene expression between different experimental conditions, or to identify new RNAs [17]. RNA-seq has subsequently become part of many specialized transcriptome-wide methods, which cleverly combine RNA-seq with for example protein detection techniques to identify RBP binding sites. One particular successful and widely-used method in this regard is CLIP-seq (crosslinking and immunoprecipitation followed by high-throughput sequencing), which allows the identification of RBP binding sites on a transcriptome-wide level [18].

The widespread application of NGS methods over the last decade has led to a tsunami-like flood of NGS data. A single high-throughput experiment typically produces several gigabytes (GBs) of raw data, and as of 2021 dedicated online databases such as the Sequence Read Archive (SRA) contain more than 20 petabytes (i.e., 20 million GBs) of NGS data [19]. This naturally has created a high demand for efficient computational methods to process and study these datasets. Moreover, the introduction of a novel experimental method often results in the development of computational methods specialized on the respective experimental data. Bioinformatics, the study of biological data by computational methods, has thus become a central part of today's biomedical research and its increasingly data-driven approach to knowledge gain. This includes the statistical analysis of experimental datasets to learn the properties of biological systems and to generate new hypotheses. Furthermore, predictive models can be learned from the data to complement or replace experimental results. Both classical machine learning and deep learning methods have successfully been applied to these and many other bioinformatics tasks [20]. Given the ever increasing amount of experimental data and non-stop advancements in deep learning, there is no doubt that we will see many more exciting applications in the near future.

1.2 Objectives

As motivated in the previous section, the study of RNA-protein interactions is critical to our understanding of cellular mechanisms. CLIP-seq is by far the most common experimental method used to identify the RNA binding sites of a specific RBP on a transcriptome-wide scale. CLIP-seq data thus allows the study of RBP binding properties, as well as global RBP binding patterns to learn more about their cellular functions. However, conducting a successful CLIP-seq data analysis can be challenging, as there are many analysis steps and protocol variants available, each requiring specific adaptations to the different steps. Moreover, CLIP-seq data analysis could be further improved, e.g., by integrating information on transcript structure and splicing events into the binding site definition process. The first set of thesis objectives deals with these issues, by providing guidelines and tools for the

analysis of CLIP-seq data, as well as ways and methods to improve the definition of RBP binding sites from CLIP-seq data (see publications P1, P4, and P5).

A natural limitation of the CLIP-seq protocol is its dependency on target RNA expression, meaning that it can only recover RBP binding sites on expressed RNAs. This causes a need for computational methods which can learn the principal binding properties of an RBP from CLIP-seq data, and use these to predict binding sites on any given RNA. Transcriptome-wide RBP binding site predictions can further be combined with other types of predictions, such as RNA-RNA interaction predictions, to learn more about regulatory mechanisms. As described in the motivation, non-coding RNAs (especially long non-coding RNAs, see section 2.1.4 *Long non-coding RNA*) make up a huge fraction of the transcribed genome, urging the need to speed up their functional characterization by computational methods. Lately, various long non-coding RNAs (lncRNAs) have been reported to interact with other RNAs and RBPs to exert their functions. It therefore makes sense to develop a method which can predict these interactions and infer their functional mechanisms. Besides applying existing RBP binding site prediction tools, there is also a need for the development of new tools, as current methods often have issues regarding runtimes, usability, or feature support. The second set of thesis objectives therefore includes the development of novel methods for the functional characterization of lncRNAs, as well as the prediction of RBP binding sites (see publications P2 and P3).

Together, the described objectives are addressed by the following works presented in this thesis:

- Publication P1 is a review article on CLIP-seq data analysis, with the goal to assist readers in performing a successful CLIP-seq data analysis. It describes the available CLIP-seq protocol variants and their specialities regarding data processing, as well as the major analysis steps, including data preprocessing, identification of RBP binding sites from CLIP-seq data (also termed peak calling), and the analysis of RBP binding properties.

- Publication P2 presents MechRNA, the first tool capable of inferring functional mechanisms for lncRNAs based on their interactions with RBPs and other RNAs. MechRNA utilizes RNA-protein and RNA-RNA interaction predictions together with expression data to deduce possible mechanisms, and reports the most likely mechanism for each lncRNA-target RNA pair, ranked by their joint p-values. The results demonstrate that MechRNA is capable of detecting known lncRNA mechanisms, making it a valuable tool for the study of non-coding RNA functions.

- Publication P3 introduces RNAProt, a flexible and performant RBP binding site predictor based on Recurrent Neural Networks (RNNs). RNAProt offers state-of-the-art predictive performance, as well as superior runtimes compared to other recent methods. In addition, it supports more predictive features and input data types than any other available method, including user-defined predictive features.

- Publication P4 investigates the consequences of ignoring transcript information in CLIP-seq data analysis, as well as the benefits of adding them. The results demonstrate that ignoring transcript information compromises peak calling, and that a substantial amount of publicly available CLIP-seq data is susceptible to this problem. Moreover, including transcript information influences the performances of RBP binding site prediction tools, and known motifs of spliced-RNA-binding RBPs are enriched in sites where the genomic (i.e., unspliced RNA) context is exchanged with transcript (i.e., spliced RNA) context.

- Publication P5 presents Peakhood, the first tool that takes RBP binding sites identified by CLIP-seq and determines the most likely context individually for each site. Motivated by the findings described in P4, Peakhood was developed to include transcript information in the site context selection process. P5 shows that Peakhood's context extraction agrees with known RBP roles, demonstrating its capability to improve the quality of RBP binding data. Peakhood can be applied as a post-processing tool inside a CLIP-seq analysis pipeline, or to reanalyze any of the millions of publicly available CLIP-seq peak regions determined by various peak callers.

1.3 Thesis structure

Chapter 2 provides the biological, experimental, and computational background necessary to understand the topics presented in this thesis. This includes experimental methods such as RNA-seq, CLIP-seq, and CLIP-seq protocol variants, as well computational methods for the prediction of RNA-protein interactions, and an introduction into deep neural networks and RNNs. Chapter 3 summarizes the contents of the five publications (P1-P5). Chapter 4 concludes the findings presented in this thesis, and also gives an outlook on possible future research directions. Detailed descriptions on the contributions of all co-authors can be found in Chapter 5, together with the five publications in their published form. The thesis is completed by the bibliography, an index of keywords, and the appendix, which includes a list of abbreviations and the supplementary material for all five publications.

Chapter 2

Background

2.1 Biology

This section provides an introduction into the biological background necessary to understand the presented scientific work. The focus in the content is on human biology, as the included work was conducted mainly on human data. Given the topics of this thesis, special attention is paid to the description of RNA and its defining features, its lifecycle, long non-coding RNAs, as well as the regulatory interplay between RNA and RNA-binding proteins.

2.1.1 A very brief history of life

Life on earth has come a long way since its emergence around 3.42 billion years ago, possibly earlier [21]. The current scientific notion is that its origin can be traced back to molecules with the ability to self-replicate, in a (periodically) aquatic environment [22]. This way they could serve as templates for future versions of themselves, which changed over time, clearing the way to new molecules and functions. At some point these reactions started to become enclosed by membranes. Molecules that catalyze the replication of other molecules could emerge. Auto-catalytic loops evolved, in which one molecule catalyzes the synthesis of another, and vice versa. Sooner or later, autonomous systems of reactions able to maintain and reproduce themselves could follow; a key characteristic of life. The enclosure led to the development of new functions, by containing and concentrating chemical processes and molecules, eventually allowing interactions with and movement within the environment. Through time and space, these tiny membrane-enclosed entities went their separate ways and evolved into an enormous variety of life forms. Today, we observe that all living things, also termed organisms, are made up of these entities, which we call cells. Despite the staggering diversity of organisms on this planet, they are all related by some last common ancestor. Among many other evidence, this is backed by the fact that their cells still share the same fundamental biomolecules, as well as many fundamental biochemical processes.

2.1.2 Molecules of life

All cells are made up of water, inorganic ions, and organic (i.e, carbon-containing) molecules. Among these organic molecules, we can spot four types of large and interestingly repetitive molecules (also termed macromolecules), which can be found in all present life forms: DNA

(deoxyribonucleic acids), RNA (ribonucleic acids), proteins, lipids, and (complex) carbohydrates. The first three are usually of particular interest to bioinformatics, since they store, convey, and execute the genetic information.

2.1.3 DNA

The structure of DNA

All organisms store their genetic information, i.e., the information necessary to maintain and reproduce themselves, in DNA molecules. DNA is a chain-like polymer made up of four distinct monomers, or nucleotides (nt) (Figure 2.1 a). Each nucleotide is made up of a common sugar-phosphate part, as well as one of the four distinct nucleobases: adenine (A), cytosine (C), guanine (G), and thymine (T) (Figure 2.1 b). In RNA molecules, the deoxyribose sugar and T are replaced by ribose and the nucleobase uracil (U). The structure of DNA is that of a double(-stranded) helix, with two polynucleotide chains wrapped around each other in an anti-parallel orientation. [23, 24]. The double strand is formed by interactions between complementary nucleotides via hydrogen bonds, with two possible base pairings (A with T, C with G, see Figure 2.1 c). Since the two strands are complementary to each other, and because of the strict pairing rules, the sequence of the second strand can be easily deduced by reversing the first strand and taking the complementary nucleotides (i.e., taking the reverse complement). A single DNA molecule is also termed chromosome, while the sum of all unique DNA molecules present in an organism is called genome.

Genomes of different forms and sizes occur throughout the three domains of life (Archaea, Bacteria, Eukaryota). While Bacteria often have a single, circular genome, the genomes of Eukaryotes (e.g., including animals, plants, and fungi) feature multiple chromosomes. Eukaryotic cells further contain several membrane-enclosed compartments, also termed organelles. Most of the eukaryotic genome is stored in the nucleus. Other DNA-containing organelles are the mitochondria (mitochondrial DNA, mtDNA) and the chloroplasts (present in plants and algae). Nuclear chromosomes are tightly packed, which is necessary due to their increased number and sizes. Packaging is achieved through DNA-protein complexes, also termed chromatin. Packaging is hierarchical, with the basic unit, the nucleosome, created by a DNA segment of $\sim$ 147 base pairs (bp) wrapped around a set of 8 histone proteins [25]. Depending on the density of packaging, chromatin can be further divided into euchromatin (light) and heterochromatin (tight). Chromosomes can have remarkable lengths, for example the human chromosome 1 with a length of close to 250 million bp. For humans, the size of the (haploid) reference genome is about 3.1 billion bp.

DNA strands (just like RNA strands) are directed, i.e., their two ends differ from each other. This is because the nucleotides are connected via a sugar-phosphate backbone, which contains a 5' (five prime) and a 3' (three prime) end, derived from the numbering of the carbon atoms in the (deoxy)ribose sugar (see Figure 2.1 a + c). The direction is important, since strand synthesis (during DNA replication or RNA transcription) only works in 5' to

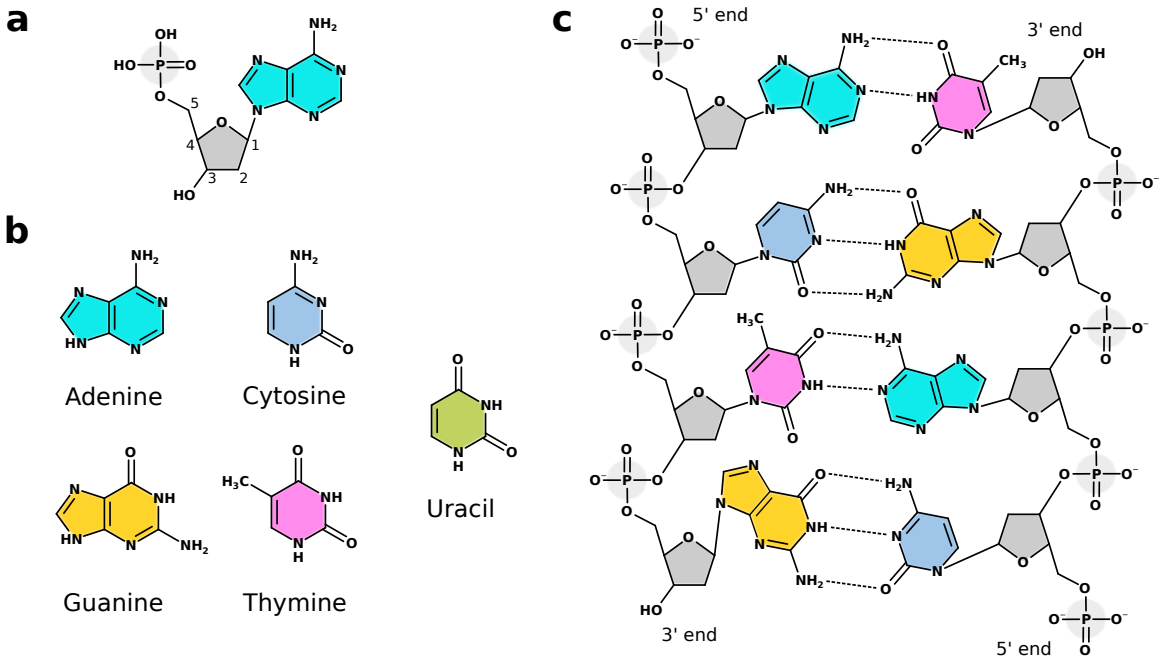


Figure 2.1: The structure of DNA. **(a)** Structural formula of a nucleotide (Deoxyadenosine monophosphate), with sugar carbon numbering. A nucleotide consists of a (nucleo)base (here adenine), and a sugar-phosphate backbone (deoxyribose sugar and phosphate group). In RNA, a second hydroxyl group (-OH) is attached to the sugar at C2. **(b)** Structural formulas of the four DNA bases (adenine, cytosine, guanine, thymine), and the RNA base uracil, which substitutes thymine in RNA molecules (together with ribose instead of deoxyribose in the sugar-phosphate backbone). **(c)** Structure of a DNA double strand with a length of 4 base pairs. Sugar-phosphate backbone in grey, bases in their respective colors. Solid lines denote covalent bonds, dashed lines denote hydrogen bonds. A-T pairs feature two hydrogen bonds, G-C pairs three hydrogen bonds. Structures drawn with [26] and adapted.

3' direction. Moreover, DNA or RNA sequences by convention are written and stored in 5' to 3' direction (left to right), with index 1 to the length of the sequence. The written strand is also called forward strand (denoted with a "+" symbol), while the reverse strand (3' to 5' direction, denoted with "-" symbol) as described is the (reverse) complement of the forward strand. Due to the 5' to 3' convention, upstream is frequently used as a synonym for "towards the 5' end", while downstream equals to "towards the 3' end".

Genes are the units of genetic information

The genetic information is stored in DNA regions called genes. Genes are defined as regions from which a functional RNA (also different versions of it) can be produced[1]. The process is also termed transcription, since the initial (unprocessed) RNA sequence is an exact copy of the transcribed DNA sequence (with T replaced by uracil (U)). Genes can be located on the plus and minus strand, and they can partially or fully overlap, which also allows

[1] for ongoing discussions about the difficulties of defining a gene see [27]

for a more compact storage of the genetic information. RNA transcribed from a gene can either be protein-coding or non-coding. For protein-coding RNAs (also termed messenger RNA or mRNA), the gene function is executed by the protein (into which the RNA gets translated). For non-coding RNAs, the RNA itself exerts the gene function. Here we also use the term gene expression, i.e., a gene is expressed at a certain point if there is RNA or protein produced from the gene at that time point. Besides genes, the genome also contains functional elements outside of gene regions which are involved in the regulation of gene expression, either on the DNA level, or by transcribing regulatory RNAs [6] (also see sections 1.1 and 2.1.4 *Transcription*).

The estimated number of protein-coding genes in the human genome interestingly has decreased over past two decades. In 2001, when the first draft sequence of the human genome was published, it was estimated to be about 30,000 to 40,000 [3]. The current estimate, which is considered to be fairly robust, settled down at 19,000 to 20,000, resulting in only 1.9% of the human genome ending up in processed protein-coding transcripts [1]. On the other hand, the widespread use of high-throughput sequencing since the late 2000s has shown that most of the human genome is transcribed [28]. Subsequently, a lot of new potentially functional long non-coding RNA (lncRNA) genes were annotated, often surpassing (depending on the source) the mentioned number of protein-coding genes [8, 29]. Long non-coding RNAs, defined as non-protein coding RNAs with lengths > 200 nt, have consequently become of high interest to research. But because of their poor functional characterization, it is still unclear how many of these are actually functional. Some long non-coding RNAs have shown to possess protein-coding potential, namely to encode for small proteins [30]. It has also been argued that even though most are likely non-functional, their abundant existence might provide a strategy for swift function generation and evolutionary adaptation [31]. Interestingly, the amount of non-protein-coding DNA strongly correlates with organism complexity, while the genome size or number of protein-coding genes does not [32]. A similar correlation has also been brought up for lncRNA genes (although so far only tested for a small number of organisms), further hinting at their functional importance [33].

Genetic information flow

Figure 2.2 illustrates the genetic information flow inside cells. Its principal flow $DNA \rightarrow RNA \rightarrow Protein$ was famously first described by Francis Crick in 1958, who named it the central dogma of molecular biology [34, 35]. Even though his use of the term "dogma" has been criticized, it proved to be remarkably accurate and valid (with two adaptations) to this day for all observed organisms and viruses. The two adaptations are the less common information flows from RNA to DNA (reverse transcription) and from RNA to RNA (RNA replication)[1].

Reverse transcription requires the presence of an enzyme, a protein that catalyzes chemical reactions, termed reverse transcriptase. This enzyme catalyzes the synthesis of DNA

[1] Crick regarded these as possible flows in [34], but there was no evidence for them yet in 1958

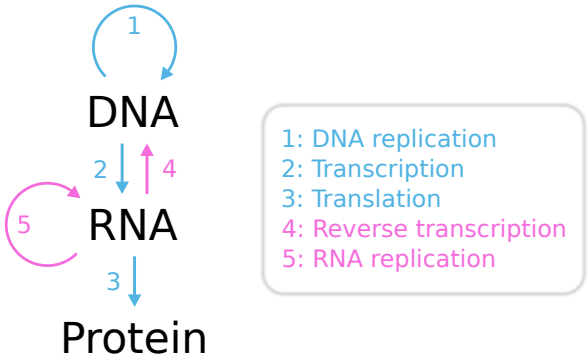


Figure 2.2: Genetic information flow. The principle genetic information flow from DNA over RNA to protein (DNA replication, transcription, translation) is colored in blue. Subsequently discovered and less frequent flows (reverse transcription, RNA replication) are depicted in pink.

from an RNA template [36]. Reverse transcription is applied by retroviruses (e.g., HIV), to insert their RNA-based genome into the DNA-based host genome, and has also been brought up for the SARS-CoV-2 virus [37, 38]. It is also found in Eukaryotes, enabling retrotransposition and telomere synthesis [39, 40]. Synthesizing RNA from another RNA (RNA replication) is achieved by enzymes termed RNA-dependent RNA polymerases, which are typically found in RNA viruses (e.g., SARS-CoV-2), to replicate their genome. RNA replication is observed in Eukaryotes as well, for example in RNA interference [41]. Apart from the flows depicted in Figure 2.2, many important modification steps are carried out between protein and DNA, RNA, or protein. These include: RNA processing by proteins (e.g., splicing, processing of RNA ends, RNA modification, RNA editing), DNA processing by proteins (e.g., DNA methylation), and protein processing by proteins (post-translational modification).

2.1.4 RNA

Classes and functions

RNA in the cell can be divided into protein-coding RNA and non-coding RNA (ncRNA). Protein-coding RNA is also termed messenger RNA (mRNA), as it stores and delivers the message necessary to produce a protein. The sequence portion of the mRNA that contains the protein information (i.e., that gets translated into protein) is also termed the coding sequence (CDS) or open-reading frame (ORF), while the 5' and 3' regions surrounding the CDS are termed 5' and 3' untranslated regions (5'UTR, 3'UTR). The protein information is stored in the CDS as consecutive, non-overlapping triplets of nucleotides, also termed codons. Each triplet encodes for a specific amino acid, the building blocks of proteins, including the first codon (start codon), which also marks the translation start. The last codon (stop codon) marks the end of translation, but does not encode for an amino acid. UTR regions usually have regulatory functions, often by providing binding sites for RNA-binding proteins, which can influence the stability, localization, or the translation of the RNA. For mRNAs, the gene

function is carried out by the produced protein. This is in contrast to ncRNAs, where the function is provided by the RNA itself.

A great variety of classes exists among ncRNAs, often with primary, class-defining functions [6]. For example, transfer RNA (tRNA) and ribosomal RNA (rRNA) have distinct roles in translation, while microRNAs (miRNAs) repress the translation of target RNAs. In contrast, long non-coding RNAs (lncRNAs), a more recently discovered class of ncRNAs, are defined solely based on their length (non-coding RNAs of > 200 nt length). Not surprisingly, lncRNAs show a diverse range of functions, although most of the potential lncRNAs in the human genome still await functional characterization (see section 2.1.4 *Long non-coding RNA* below for more details). RNA is not present in a straight linear form inside the cell, but instead folds into complex secondary and tertiary structures. Consequently, RNA is often regulated and functions based on its structural conformation, on top of its sequence composition (see section 2.1.4 *RNA structure* below for more details). In addition, RNAs can also be present in circular form, which are termed circular RNAs (circRNAs). Similarly to lncRNAs, circRNAs have gained considerable interest in recent years, since their widespread appearance has first been shown in 2013 [42].

Transcription

RNA is the primary gene product, transcribed from gene regions. A gene can have one or more transcript variants (also termed isoforms) encoded in its region. Gene expression, the production of functional RNA and protein products, determines cellular identity and behavior. Depending on its gene expression profile, a cell might for example maintain its cellular state, communicate with other cells, metabolize fat, progress in the cell cycle, divide, differentiate, migrate, and so on. Consequently, gene transcription is a highly regulated and complex process. Transcription is carried out by DNA-dependent RNA polymerases, as well as a multitude of regulating cis and trans-acting factors. In gene regulation, a cis-acting factor is a factor which regulates a target gene based and depending on its location. The target gene is often in near proximity, but can also be farther away on the chromosome. Such factors include DNA regions (e.g., promoters and enhancers) to which regulatory proteins can bind, but also cis-acting long-non coding RNAs, which are transcribed from these regions [43]. In contrast, a trans-acting factor is an RNA or protein transcribed from a genomic region whose relative location to the target gene has no influence on the regulatory function. Trans-acting factors that bind to cis-acting DNA regions like promoters or enhancers are also called transcription factors. Eukaryotes possess three different DNA-dependent RNA polymerases (Pol I, Pol II, Pol III) for transcribing their nuclear genes, each transcribing different classes of RNA [44]. Pol II transcribes all protein-coding but also certain non-coding RNAs, including microRNAs and most long non-coding RNAs [45, 29]. Most ribosomal RNAs are transcribed by Pol I, while Pol III transcribes tRNA, as well as the remaining ribosomal RNAs and other short non-coding RNAs [46].

The transcription process can be divided into three phases: initiation, elongation, and

termination. The following briefly describes them for protein-coding genes (Pol II transcription). Transcription typically starts at promoter regions located at the gene 5' end, which include the transcription start site (TSS) (i.e., the site where RNA synthesis begins). A pre-initiation complex (PIC) is formed by binding of general transcription factors (GTFs), to recruit Pol II to the core promoter [47]. PIC then activates Pol II, opens the double-stranded DNA (i.e., creating a transcription bubble), and positions Pol II on the TSS. After a few synthesized nucleotides, Pol II is set free from the GTFs and the core promoter and leaves the TSS (promoter escape). After 30-50 synthesized nucleotides, Pol II comes to a hold (promoter-proximal pausing), awaiting further instructions. Pause-release is signaled by positive transcription elongation factor P-TEFb through phosphorylation of Pol II and other bound proteins, unleashing Pol II into productive elongation. During elongation, Pol II and associated proteins interact with various other RNA processing pathways. For example, 5' end capping is induced early on during elongation, and splicing factors to assemble the splicing machinery are recruited [48]. The last phase, transcription termination, results in the release of the Pol II complex and the synthesized transcript. It is the least understood phase of the three, since there are many different termination pathways, which depend on the given context and transcribed gene type [49]. The best studied pathway occurs in protein-coding genes, which usually contain specific nucleotide elements at their 3'ends, including the polyadenylation signal (PAS) (AAUAAA motif). The cleavage and polyadenylation (CPA) complex recognizes these elements, cleaves the RNA and adds a poly(A) tail to its 3' end. However, only a fraction of Pol II reaches the 3' end, since promoter-proximal and premature termination are frequently occurring events.

Whether transcription continues after initiation is controlled to a large extent by enhancer elements. The same holds for the elongation and transcription in general [50]. Surprisingly, RNA is also transcribed from enhancers, and these non-coding enhancer RNAs (eRNAs) have shown to regulate transcription as well, mostly in cis, but possibly also in trans [51]. Apart from enhancers, other important cis-acting factors are silencers and insulators. While proteins binding to silencer regions repress gene transcription, insulators can restrain enhancer or silencer activities [52]. Such seemingly long-range interactions work because of the three-dimensional structure of the chromosome, which can bring two distant genomic regions close enough for protein-protein interactions to occur. Finally, chromatin accessibility plays a huge role in gene transcription [25]. Transcription can be initiated by transcription factors in accessible genomic regions (euchromatin), while tightly packed genomic regions (heterochromatin) are usually transcriptionally silenced. Histone modifications (mainly methylations and acetylations) largely contribute to chromatin accessibility [53]. Moreover, DNA methylation, which can also be inherited (thus termed an epigenetic modification), strongly influences accessibility and therefore also gene transcription [54].

Splicing

Most nuclear mRNAs are made up of two different kinds of regions, termed exons and introns. While exons are the regions that end up in the mature RNA which serves as a template for translation, introns get removed prior to translation in a process called splicing. This process of intron removal usually occurs co-transcriptonally, i.e., during transcription, and is often coupled with other precursor mRNA (pre-mRNA) processing steps [55]. The splicing machinery, also termed spliceosome, is a large and dynamic RNA-protein complex, where small nuclear RNAs (snRNAs) are associated with proteins to form small nuclear ribonucleoproteins (snRNPs), and together with various other non-snRNP proteins recognize and remove intron regions. Two different spliceosomes exist in the nucleus, the major (U2-dependent) spliceosome and the minor (U12-dependent) spliceosome. U2 and U12 refer to the use of functionally analog U2 and U12 snRNAs in the two complexes. Over 99% of human introns are spliced by the major spliceosome (U2-type introns), while minor (U12-type) introns only occur in 700-800 genes, typically present only once per gene among several major introns [56]. In addition to spliceosomal introns, there are also introns capable of self-splicing, i.e., the intron acts as a ribozyme to catalyze its own removal. Three such groups (I, II, III) have been described in the three domains of life and in viruses, of which group II introns are regarded as the ancestors of spliceosomal introns [57]. While the human genome does not contain self-splicing introns anymore, it is assumed that they had a profound impact on eukaryotic evolution, not only as progenitors of spliceosomal introns, but, e.g., also on the development of the nuclear envelope [58]. Looking at a recent reference annotation of the human genome, human transcripts feature a median number of 8 introns, with a minimum of 1 and a maximum of 362 introns, a mean length of 6,938 nt (median 1,747 nt), and a minimum and maximum length of 26 and 1,160,411 nt, respectively [1]. In contrast, exons are usually much shorter, with a mean and maximum length of 160 and 21,693 nt, respectively. Besides mRNAs, lncRNAs also contain introns, although these tend to be spliced less efficiently [59].

In order to recognize and remove an intron, spliceosomes depend on three characteristic sequence elements present in both U2-type and U12-type introns (see Figure 2.3 a): the 5' splice site (5'SS), the branch point sequence (BPS), and the 3' splice site (3'SS). U2-type introns additionally feature a polypyrimidine tract (PPT), i.e., a sequence stretch rich in pyrimidines (cytosines and uracils), immediately upstream the 3'SS. Splicing out an intron involves two transesterification reactions, also termed branching and exon ligation (Figure 2.3 b) [61]. To catalyze these reactions, the spliceosome needs to adopt various functional states or complexes, each with different snRNP as well as additional protein compositions [55, 62]. The principle splicing process is best studied in budding yeast but well conserved in humans. For U2-type introns it works as follows: U1 snRNP recognizes the 5'SS through base pairing of U1 snRNA with the 5'SS, while the 3' splicing elements are bound by U2 snRNP and associated proteins. These include branchpoint binding proteins, as well as U2 auxiliary factors (U2AFs), recognizing the PPT and the 3'SS. Additional proteins can bind to nearby exonic regions and interact with the snRNPs, further aiding in the exon-intron border

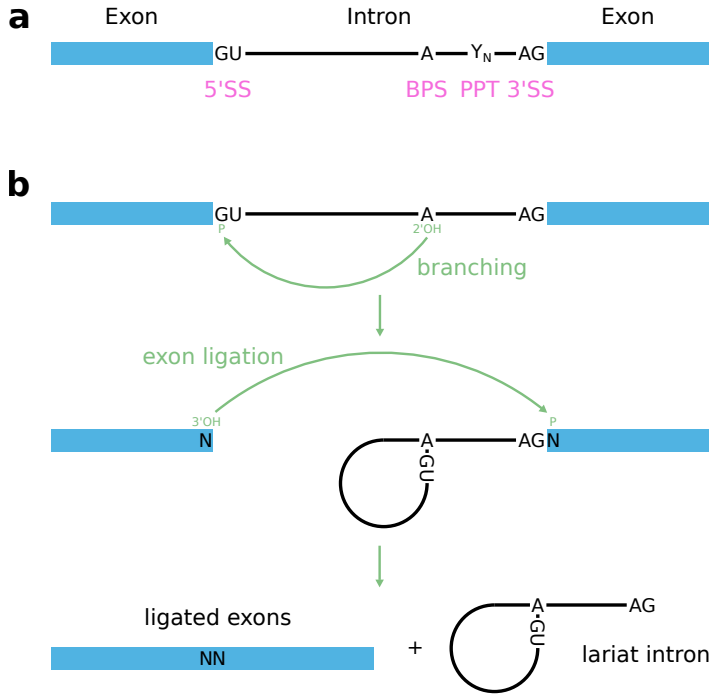


Figure 2.3: Intronic sequence elements recognized by the splicing machinery and the principal splicing process. **(a)** Intronic sequence elements necessary for intron recognition and removal: 5' splice site (5'SS), branch point sequence (BPS) including an adenosine, polypyrimidine tract (PPT) and 3' splice site (3'SS). A PPT between BPS and 3'SS is found in U2-type introns, but not in U12-type introns. 5'SS and 3'SS nucleotides GU and AG are canonical, i.e., they occur in most (U2-type) introns. **(b)** The principal splicing process, consisting of two transesterification reactions (branching and exon ligation) to remove the intron and ligate the exons. For U2-type and U12-type introns, these steps are catalyzed by the major and minor spliceosome, with varying snRNP and additional protein compositions in each step. For group II introns, these reactions are catalyzed by the intron itself [60].

recognition. Interestingly, the spliceosome also interacts with the transcription machinery, and the 5'SS remains attached via U1 snRNA to Pol II during intron synthesis, which might ease the splicing process and increase its precision, especially for longer introns [63]. For the branching reaction, the 2'OH group of the branch point adenosine performs a so called nucleophilic attack (nucleophilic substitution) on the phosphodiester group of the first 5'SS nucleotide (G in Figure 2.3 b), leading to a cleaved 5' exon, a lariat intron with its characteristic 2'-5' phosphodiester bond, and the yet connected 3'exon. For the exon ligation reaction, the now uncovered 3'OH group of the last 5' exon nucleotide performs a second nucleophilic attack, this time on the phosphodiester group of the first 3' exon nucleotide, consequently ligating the two exons and releasing the lariat intron.

Compared to yeast, human splice sites are rather short and loosely conserved, with a few canonical nucleotides found in most splice sites, like the 5'SS GU, the 3'SS AG, as well as the branch point adenosine [55, 61]. In addition, distances between the elements (e.g., between BPS and 3'SS) are more variable, and intron regions as noted often have

considerable (> 1,000 nt) lengths. This development has two important consequences: 1) to secure splicing fidelity in the light of increased intron lengths and splice site variability, the spliceosome has to rely on additional RNA-binding proteins for guidance; and 2) more additional factors (proteins and their RNA binding sites) increase the flexibility, but also the complexity of splicing and its regulation. Not surprisingly, alternative splicing (AS), the generation of different transcripts (also termed isoforms or splice variants) from the same gene, is ubiquitous in multicellular eukaryotes [64]. Various types of alternative splicing exist, such as exon skipping, alternative 5'SS and 3'SS usage, mutually exclusive exons, or intron retention. In humans, there are ∼ 150 additional proteins associated with splicing, and 95% of multi-exon genes undergo AS [65, 66]. AS events are often tissue-specific, with fundamental roles in organ development, and many human diseases are caused by splicing errors [67, 68, 69]. The extent of AS is correlated with organism complexity, as measured by the number of distinct cell types [70]. A similar correlation has been shown for the amount of non-protein-coding DNA in general (including introns and intergenic regions), and has also been indicated for the number of lncRNA genes [32, 33]. AS (together with these factors and other RNA processing steps described below) therefore can help to explain the increased complexity we observe in our dear selves *Homo sapiens*, even though humans only have four times as many protein-coding genes as the budding yeast *Saccharomyces cerevisiae*, and about the same as the roundworm *Caenorhabditis elegans*.

Additional RNA processing

Apart from splicing, RNA goes through various additional co- and post-transcriptional processing steps. For example, mRNA processing usually involves the modification of RNA ends, such as 5' end capping, as well as 3' end cleavage and polyadenylation [71]. Other important processing steps include internal modifications of nucleobases (reversible or irreversible) within a specific RNA sequence and structure context. The most common reversible (also termed dynamic) RNA modification in mRNA is the m6A methylation, where a methyl group is added to the amino group at position 6 of the adenine base [72]. Other frequent internal modifications include m1A and m5C, and specific reader, writer, and eraser proteins have already been identified for many of these modifications. As for irreversible modifications, the most abundant form of RNA editing (i.e., changing the identity of specific nucleotide(s) in an RNA sequence) in Metazoans is adenosine to inosine (A-to-I) editing, catalyzed by adenosine deaminases acting on RNA (ADAR) [73]. All together, over 150 different RNA modifications have been reported so far [74]. These can influence the structure of the RNA, as well as its interactions with proteins or other RNA molecules. This in turn affects the further processing, localization, translation, or decay of the RNA. RNA modifications thus serve as important control elements in post-transcriptional gene regulation.

RNA localization

In order to execute its function, an RNA molecule needs to be at the right location in the cell at the right time. Consequently, RNA localization plays a vital role in post-transcriptional gene regulation, although there is still a considerable lack of understanding its precise mechanisms [75]. The prime example is the translation of mRNA, which in order to be translated into protein must travel to the cytoplasm, but often also further to more remote subcellular spaces for local(ized) translation [76]. Other examples include the transport of RNAs to specific membrane-less compartments, such as stress granules or processing bodies (P-bodies), for their transient silencing or degradation. Localization is not just important for the function of mRNA, but also for small and long non-coding RNA [6, 77]. Subcellular localization is typically determined by specific cis elements present in the RNA, also termed zip codes. These can be single or recurring sequence or structure elements (in mRNAs often located in the 3'UTR region), which are recognized by trans-acting RNA-binding proteins. The resulting ribonucleoprotein (RNP) complexes can then interact with other proteins along the way to lead the RNA to its destination. Localization is often active, through interactions with cytoskeletal proteins, but can also be passive through diffusion of the RNA. Even though there are presumably thousands of RNAs that travel to subcellular locations, localization elements so far have only been characterized for a few dozen. A classic RNA localization example is the localized translation of β-actin mRNA in developing neurons. The transport is mediated by a bipartite zip code element in the β-actin 3'UTR, which is recognized by the RNA-binding protein IGF2BP1 [78].

RNA quality control and decay

The detection and disposal of aberrant transcripts, i.e., transcripts with incomplete or erroneous information content, is handled by a number of RNA quality control pathways in mammals. Such RNA surveillance pathways are necessary in order to prevent the accumulation of RNAs or proteins with no or differing function, which otherwise can cause cell death or disease. Three major pathways with well-studied roles in mRNA quality control are no-go decay (NGD), nonsense-mediated decay (NMD), and non-stop decay (NSD) [79]. All three detect abberant mRNAs by interactions with ribosomes (the molecular machines that translate RNA into protein): NGD and NSD both detect abberant mRNAs based on the occurence of slowly moving or stalled ribosomes, due to a persistent RNA structure or a specific tRNA shortage (NGD), or due to a missing stop codon (NSD). NMD on the other hand detects mRNAs containing premature stop codons or extended 3'UTRs. NMD has recently also been implicated in lncRNA quality control [80]. In addition, RNA surveillance pathways prior to translation have been described, which can detect abberant mRNAs as well as ncRNAs [81]. In general, these pathways compete with regular RNA processing pathways, such as 5' and 3' end processing, transcription elongation, splicing, or nuclear export, and transcripts that fail to form or feature the respective RNP complexes at certain time points or locations are usually targeted for degradation.

RNA decay (or degradation) is the ultimate step in the RNA lifecycle, and therefore also part of many regulatory pathways, including the described RNA surveillance pathways. As such it is highly regulated, and there are a number of RNA-degrading complexes coupled with various pathways [82]. RNA-degrading enyzmes (also termed ribonucleases or RNases) can be classified into three types based on their mode of action: endouncleases cleave the RNA backbone internally, while exonucleases digest RNA by removing single nucleotides either at the 3' end (3' to 5') or at the 5' end (5' to 3'). Exonucleases typically require single-stranded RNA as substrates. This means that RNA ends can be protected from degradation, either by RNA structure (which includes double-stranded RNA), RBP binding, or chemical modification. RNA decay thus usually starts with the removal of protective ends, e.g., 5' cap removal and 3' deadenylation for most mRNAs and many lncRNAs. Other ways of granting RNase access include RNA helicase activity for unwinding structured RNA (e.g., MTR4 against aberrant nuclear RNAs), or the enzymatic addition of short A or U tails [81]. 5' to 3' degradation is usually catalyzed by cytoplasmic Xrn1/2 exonucleases, whereas 3' to 5' decay is catalyzed by cytoplasmic Dis3L2 exonuclease or the (nuclear or cytoplasmic) RNA exosome complex. Furthermore, RNA stability can be regulated by chemical modifications (as described in the above section 2.1.4 *Additional RNA processing*), miRNA binding mediated by RBPs (AGO1-4 in mammals [83]), as well as RBP binding in general to regulatory sequences or structures. For example, histone mRNAs lack a poly(A) tail, but instead form a conserved stem loop structure at their 3' end bound by the RBP SLBP, which is also involved in their degradation [84]. Another prominent example are Pumilio RBPs (PUM1 and PUM2 in mammals), which usually inhibit translation and promote decay of their target RNAs [85].

RNA structure

Because of their innate base pairing abilities, RNA sequences tend to fold into complex three-dimensional (3D) structures. RNA structure can be described with regard to its primary, secondary, tertiary, and quaternary structure. The primary structure is the sequence of nucleotides, connected by the ribose-phosphate backbone. The secondary structure is described by the base pairing of complementary sequence stretches, interspersed by various loop regions. The tertiary structure is the actual 3D structure determining the function of the RNA, such as the catalysis of chemical reactions, the binding of ligands, or interactions with proteins and other nucleic acids. In case of interactions with other molecules, the quaternary structure describes the 3D structure of the formed complexes. In the following, we will refer to RNA structure meaning the folded sequence (i.e., secondary or higher structure).

RNA structure is formed spontaneously in the cell, since stacks of base pairs (also termed duplexes, helices, or stems) increase the thermodynamic stability. This means that the Gibbs free energy G of the formed structure is lower than the energy of the single-stranded RNA, or, in other words, ΔG is negative. In addition to the two canonical Watson-Crick base pairs (A-U, C-G), many non-canonical base pairs occur in RNA structures, albeit with lower

frequencies [86]. Of these the G-U wobble base pair is the most common base pair in well-studied RNA structures such as rRNA and tRNA, with crucial roles in RNA function [87]. RNA folding is hierarchical, meaning that secondary structure forms faster and is more stable than tertiary interactions [88]. Consequently, secondary structure folding is less dependent on tertiary contacts, making it possible to predict RNA secondary structure without taking into account tertiary information. RNA structure is dynamic, meaning that an RNA can adopt different structural conformations with varying probabilities and formation times, often guided by interactions with other molecules [89].

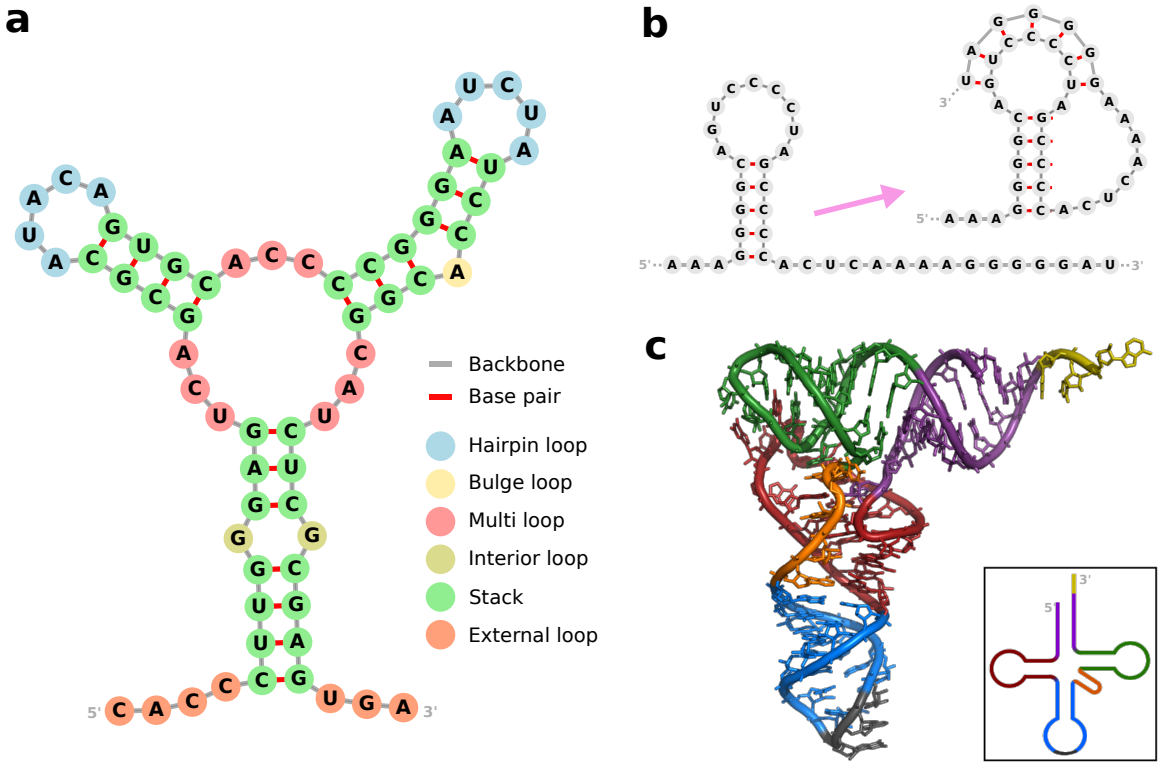


Figure 2.4: RNA secondary and tertiary structure examples. **(a)** Structural elements used to described RNA secondary structure: stacks of base pairs create five distinct loop regions: external loops, interior loops, multi(branched) loops, bulge loops, and hairpin loops. **(b)** Secondary structure of a frameshifting element inside mouse mammary tumor virus mRNA [90]. The structure contains a pseudoknot, formed by base pairing between two loop regions (hairpin loop and external). Structures drawn with `forna` [91] and adapted. **(c)** Secondary and tertiary structure of a tRNA (phenylalanine tRNA from yeast), with its characteristic cloverleaf (2D) and L-shape (3D). The single structural elements are colored: 3' CCA tail in yellow, acceptor arm in violet, D-arm in red, T-arm in green, variable loop in orange, anticodon arm in blue, and anticodon in grey. Figure adapted from [92] (author: Yikrazuul, license: CC BY-SA 3.0 [93]).

Figure 2.4 a illustrates the elements used to describe RNA secondary structure. Energetically favorable stacks of base pairs create five distinct loop regions: external loops, interior loops, multi(branched) loops, bulge loops, and hairpin loops. Base pairing can also occur between nucleotides from different loop contexts, an interaction termed pseudoknot [94]. Figure

2.4 b shows the secondary structure of a pseudoknot found in the mouse mammary tumor virus, which causes a frameshift during the translation of the RNA. Since tertiary structure elements (also termed tertiary motifs) are described as interactions between distinct secondary structure elements, pseudoknots are usually also regarded as tertiary motifs [95]. Besides pseudoknots, many other types of tertiary motifs have been observed in RNA, such as coaxial stacking of helices, triplexes, tetraloops, kink turns, or quadruplexes [95, 96]. For example, the two highly abundant human lncRNAs MALAT1 and NEAT1 have been shown to form triplex structures at their 3'ends, protecting them from rapid nuclear deadenylation-dependent decay [97]. Figure 2.4 c depicts the typical cloverleaf secondary structure and resulting L-shaped tertiary structure of a tRNA (phenylalanine tRNA from yeast). The L-shape is formed by tertiary motifs, including a pseudoknot between the D- and T-arm loop region, as well as coaxial stacking between stems of the D- and anticodon arm, and the T- and acceptor arm.

Long non-coding RNA

As described (Section 2.1.3 *Genes are the units of genetic information*), long non-coding RNAs (lncRNAs) are ncRNAs of > 200 nt length which are ubiquitous and widely expressed in the human genome. While the functions of most annotated lncRNAs are still unknown, great effort has been put into their functional characterization over the last years [98]. Just like its vague length-only definition suggests, lncRNAs make up a functionally diverse class. Any two lncRNAs can exert their functions through different subcellular localization and interaction partners [29]. These interactions include conventional RNA-RNA and RNA-protein interactions, but also RNA-DNA interactions, e.g., through the formation of triple helices (also termed triplexes) with double-stranded DNA, or R-loop formation with single-stranded DNA [99]. In addition, some lncRNAs bind large numbers of the same RBP or miRNA, effectively acting as sponges to regulate the function of their interaction partners [100]. LncRNAs can be further categorized based on their genomic location relative to protein-coding genes: long intergenic non-coding RNAs (lincRNAs), sense lncRNAs (intragenic or intronic lncRNAs), antisense lncRNAs, bidirectional lncRNAs, and enhancer RNAs (eRNAs). LincRNAs do not overlap with protein-coding genes, while sense lncRNAs overlap on the sense strand, and antisense lncRNAs on the opposite strand. Bidirectional lncRNAs are transribed from the same promoter as the protein-coding gene, but in opposite direction. Enhancer RNAs are transcribed from enhancer regions in both directions. The location of the lncRNA can give clues about or determine its function, e.g., by acting in cis to regulate chromatin structure or transcription of nearby genes [43]. But lncRNAs also frequently travel from the nucleus to the cytoplasm, where they are involved in various processes, such as translational regulation or signalling pathways [101]. Interestingly, lncRNAs often show tissue- or condition-specific expression, which can hint at their biological functions or can be used for disease diagnosis. For example, certain lncRNAs are upregulated in specific cancer types, which makes them promising biomarkers as well as potential drug targets [102]. Moreover, lncRNAs and RBPs

can have connected roles in cancer [103].

2.1.5 Protein

The genetic code

The elucidation of the genetic code was a major scientific breakthrough in the early 1960s, and arguably one of the great discoveries of the 20th century [104]. The riddle to solve: how do the 4 building blocks (nucleotides) of RNA translate to the 20 building blocks (amino acids) of proteins? It was found that RNA encodes the information necessary to produce a protein in triplets (groups of three nucleotides). Each triplet (or codon) codes for a specific amino acid. Some triplets code for the same amino acid, since there are 64 possible triplets, but only 20 amino acids (i.e., the code is degenerate). Moreover, the triplets do not overlap, there are no non-coding nucleotides in the coding sequence (introns get removed prior to translation), and the coding sequence is read consecutively from 5' to 3' starting from a specific mRNA position. Due to the triplet grouping, there are three possible reading frames, and skipping one or two nucleotides (e.g., by deletions, insertions, or roadblocks during translation) leads to a frameshift. Frameshifting events can also be programmed, and are often used by RNA viruses such as SARS-CoV-2 to produce different proteins from the same genomic region [105]. Besides coding for amino acids, special triplets mark the start and end of translation. The start codon (AUG) is recognized by the ribosome, but also encodes for the amino acid methyonine. On the other hand, the three stop codons that provide the translation termination signal for the ribosome (UAA, UAG, UGA) do not encode for amino acids.

The genetic code has shown to be almost universal throughout all present organisms, although exceptions have been identified over the years in various species: besides the co-translational incorporation of additional amino acids (e.g., selenocysteine in eukaryotes, also referred to as the 21st amino acid), the meaning of individual codons can differ (e.g., UGA codes for tryptophan in vertebrate mitochondria) [106, 107]. Despite its remarkable conservation, these changes reflect the evolvability of the genetic code, including the translation machinery, as well as tRNAs and their amino acid loading enzymes (aminoacyl-tRNA synthetases). Moreover, the context of a triplet can change its encoding or affect translation efficiency [108].

Translation

The translation of mRNA into protein is achieved by an enormous ribonucleoprotein complex called the ribosome, with the help of many additional factors, including initiation factors, release factors, tRNAs, and aminoacyl-tRNA synthetases. The eukaryotic ribosome (also termed 80S ribosome, where S is the Svedberg unit as a measure of particle size) contains four different rRNAs (together $>$ 5,500 nucleotides) and around 80 proteins, and is made up of a large (60S) and a small (40S) subunit [109]. Assigning the correct amino acid to a codon

on the mRNA is accomplished by tRNA, which recognizes the codon by forming a duplex with its anticodon (situated in the loop region of the tRNA anticodon arm, see Figure 2.4 c). The tRNA is loaded (or charged) with the corresponding amino acid, i.e., the amino acid is covalently linked to the tRNA 3' end by aminoacyl-tRNA synthetases. There are 20 different synthetases in humans, one for each amino acid. Since there are more anticodons and tRNAs, tRNAs with anticodons corresponding to the same amino acid are recognized and charged by the same synthetase. As with *amino acid* → *codon*, there is again no one-to-one mapping for *anticodon* → *codon*. Instead, for some tRNAs, optimal (i.e., Watson-Crick) base pairing is necessary only for the first two codon positions, whereas the third position also tolerates wobble base pairing. While non-optimal codons were shown to decrease translation rates, synonymous codons in general can have various effects on translation and mRNA metabolism [110].

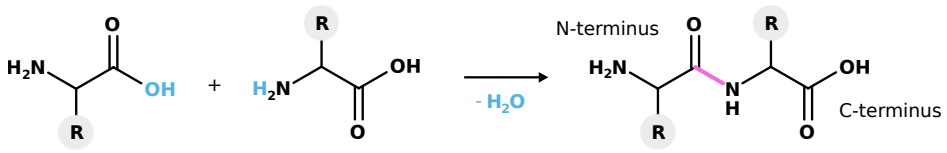


Figure 2.5: Peptide bond formation between two amino acids. The condensation reaction produces a dipeptide and a water molecule. Peptide bond highlighted in pink. R: amino acid side chain (also termed residue) different for each amino acid. The carboxyl group tail marks the C-terminus of a protein, the amino group tail the N-terminus. Structures drawn with [26] and adapted.

Just like transcription, translation can be divided into three phases: initiation, elongation, and termination. In the canonical model of translation by the eukaryotic ribosome, mRNA translation starts with the recognition of its 5' cap by a cap-binding protein complex [108]. This recruits the small (40S) ribosomal subunit, additional initiation factors, as well as the methionine-carrying initiator tRNA, positioned in the peptidyl transfer (P) site of the 40S subunit. The resulting pre-initiation complex (PIC) proceeds by scanning the mRNA in 3' direction, until it detects an AUG start codon through basepairing with the initiator tRNA anticodon. This causes scanning arrest and the release of certain initiation factors, which allows binding of the large (60S) ribosomal subunit and subsequent ribosome assembly. In the elongation phase, the ribosome moves along the mRNA from 5' to 3', codon by codon, each time pausing for the duplex formation with a matching tRNA anticodon. Once a stable codon-anticodon duplex is formed, the addition of the amino acid from the tRNA to the growing amino acid chain is catalyzed in the peptidyl transferase center of the 60S ribosomal subunit. For this the amino acid chain attached to the P-site tRNA forms a peptide bond (see Figure 2.5) with the A-site tRNA amino acid, resulting in an uncharged P-site tRNA and an A-site tRNA connected to the peptide chain. Both tRNAs are then translocated in the 5' direction, the uncharged tRNA to the exit (E) site, and the A-site tRNA to the P-site, clearing the way for the next codon-anticodon interaction in the A-site. The elongation cycle continues until a stop codon is detected in the A-site by a protein release factor. This triggers the termination of translation, which includes release of the amino acid chain and

the dissociation of the ribosomal subunits from the mRNA through recycling factors.

Translation is of course a highly regulated process, with many factors that influence the different phases of translation, such as post-translational modifications of involved proteins, mRNA sequence (e.g., codon usage), mRNA structure, mRNA modifications, or trans-acting RNAs and RBPs [110, 111, 112, 113]. There are many examples of RBPs involved in translational regulation, such as RBPs that bind to AU-rich elements (AREs) located in 3'UTRs to repress or enhance translation (e.g., HuR, TIA1), or RBPs binding to internal ribosome entry sites (IRESs) in 5'UTRs to facilitate cap-independent translation initiation (e.g., PTBP1) [114]. The most important and by far best-studied class of trans-acting RNAs are microRNAs, which through their association with the RNA-induced silencing (RISC) complex bind to microRNA-responsive elements (MREs) located mainly in the 3'UTR [115]. In addition, translation regulation has been implicated for lncRNAs [111, 116]. Moreover, regulatory crosstalk between RBPs and microRNAs has been described in many works [117, 118, 119].

Protein structure

Most proteins need to fold into distinct three-dimensional structures in order to execute their biological functions, while an estimated 15 to 30% of proteins in mammalian proteomes either partially or fully lack such ordered (i.e., stable) structures [120]. The modular building blocks and functional units of structured proteins are called domains, which can be defined as protein regions that fold and evolve independently of each other, with specific structures and functions [121]. Through this modular concept, the function of a protein largely depends on the unique combination of its domain functions, and new protein functions and proteins typically emerge by adaptations and combinations of existing domains. The majority of eukaryotic proteins contain both multiple domains and unstructured regions, which are also known as intrinsically disordered regions (IDRs) [122, 123]. Nevertheless, IDRs can still be functional, as they can contain small peptide motifs (typically < 10 amino acids), which serve as binding sites to domains of other proteins or post-translational modification sites [124]. Protein structure can be described in terms of its primary structure (i.e., the amino acid sequence), secondary structure (i.e., local structure elements such as α-helices, β-sheets and turns), tertiary structure (i.e., the three-dimensional atomic arrangement of the protein), and its quaternary structure (i.e., the combined tertiary structures of protein oligomers) [125]. Proteins often engage in protein complexes or form oligomers with same or different proteins to fulfill their biological functions. To distinguish the two terms, protein complexes are typically regarded as dynamic assemblies, whereas in oligomers (e.g., homodimers, heterodimers, etc.) the protein subunits are permanently associated and degraded together [126].

Protein folding is a cotranslational process that starts as soon as the polypeptide chain emerges from the ribosomal exit tunnel of the 60S subunit, or even earlier inside the tunnel for some small structure elements and proteins [120]. Protein folding is essentially assisted

by proteins termed molecular chaperones, which start interacting with the protein as it exits the tunnel to guide its folding process and guarantee its efficient and correct folding. Folding does not just take place in the cytoplasm, but also in the endoplasmatic reticulum (ER). Proteins destined to be folded in the ER, including secretory or membrane proteins, typically carry a signal peptide at their N-terminus (the first part that emerges from the tunnel), which gets recognized by a signal recognition particle, leading to the translocation of the ribosome to the ER membrane and the insertion of the polypeptide chain into the ER lumen. This for example also includes immunoglobulins (i.e., antibodies), which specialized antibody-producing cells can fold and assemble at a staggering pace, secreting 1000s of antibodies per second and cell [126]. As protein structure is dynamic and misfolding frequent, organisms from all domains of life have evolved a complex network of chaperones, cofactors, and protein degradation pathways termed the proteostasis network (PN), to keep up cellular homeostasis and prevent toxic aggregations of misfolded proteins. In mammals, the PN consists of ≈ 1,400 components, and the mutation or age-related decline of PN functionality is connected to many chronic and neurodegenerative diseases, such as Alzheimer or Parkinson. Besides folding, proteins often also undergo further processing steps, known as post-translational modifications (PTMs). These include the proteolytic cleavage of protein parts, as well as the addition of functional groups (e.g., phosphoryl, glycosyl, or methyl). PTMs in turn effect various properties of the protein, such as the folding, enzymatic functions, interactions with other molecules, or its subcellular localization [127]. This of course also includes RBPs, for which various regulatory PTMs have been described [128].

RNA-binding proteins

RNA-binding proteins (RBPs) are defined as proteins with the ability to interact with RNA in a functional manner, either independently (i.e., without the aid of other RBPs) or dependently (i.e., with the aid of other RBPs as part of an RNA-binding protein complex) [129]. For example, in Drosophila, Nanos RBP binding to its target mRNA is dependent on Pumilio RBP binding [130]. RNAs are usually bound by various RBPs to form ribonucleoprotein (RNP) complexes, with changing compositions throughout time and cellular location. RBPs are essentially involved in every stage of the RNA life cycle, including transcription, splicing, additional processing, localization, stability, translation, and decay (see Figure 2.6 for a visual overview). Examples of RBP functions in the different stages can be found in the respective subsections above. RBPs typically bind to specific RNA sequence or structural motifs, and the binding properties of an RBP are largely defined by the set of RNA-binding domains (RBDs) it carries. Humans contain about 1,500 RBPs with around 600 structurally distinct (i.e., known or canonical) RBDs, although only 20 RBDs are found in 10 or more genes [129]. Among these prevalent RBDs are domains such as the RNA recognition motif (RRM), the K homology (KH) domain, the DEAD motif, the double-stranded RNA-binding motif (DSRM), or the zinc-finger domain. These RBDs usually occur in repeats or combinations with other RBDs, which typically increases the sequence specificity and affinity of

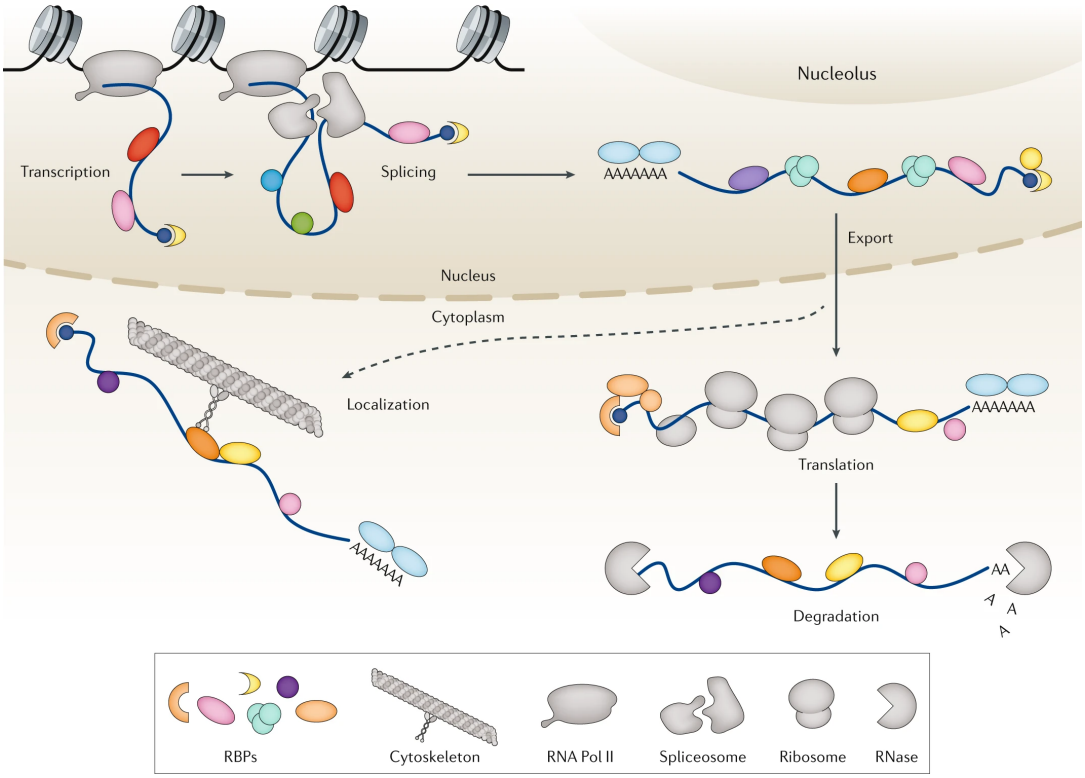


Figure 2.6: RNA-binding proteins are essentially involved in all stages of the cellular RNA life cycle. These include transcription, splicing, additional processing such as 5' capping and 3' polyadenylation, RNA export into the cytoplasm, localization, translation, or regulating the stability and the degradation of the RNA. Moreover, they are involved in controlling RNA quality, RNA-protein granule formation, or RNA modifications. Reprinted by permission from Springer Customer Service Centre GmbH: Springer Nature, Nature Reviews Genetics, Gebauer et al. [11], copyright 2020.

the RBP [131]. Apart from the mentioned class of domains whose sole function is to bind to RNA, there are also several classes of enzymatic RNA-interaction domains, including nucleotidyltransferases, ribonucleases, RNA-modifying enzymes, helicases, and GTPases [132]. More recently, high-throughput proteomic methods have identified 100s to 1000s of potentially new RBPs lacking canonical RBDs, also termed uncoventional RBPs [11]. Subsequent experimental approaches to pinpoint the RNA-binding protein regions in these RBPs showed that for many of these RBPs IDRs seem to be involved in the interactions, e.g., through short, unstructured RNA-binding motifs, or PTM sites [9]. However, the amount of RNA-specific binding in these datasets is still hard to tell, as: (i) the binding modes are not well understood yet; and (ii) a possibly large amount of these proteins could just be opportunistic (i.e., weak and non-specific) binders, which may not exhibit any RNA-related functions [133]. Interestingly, some RBPs have been shown to interact with chromatin as well, pointing at additional roles of RBPs as transcriptional regulators [134].

The majority of RBDs recognize short RNA sites of 3 to 6 nucleotides, usually with a high degree of tolerable sequence variation [131]. To further increase their sequence specificity, RBPs often contain several RBDs, whose unique combinations shape the binding behavior of

the RBP. For example, IGF2BP1 contains six RBDs (two RRM and four KH domains), and recognizes a bipartite sequence motif, with a variable-length spacer sequence between the two sequence elements [135]. There are, however, also single domains that exhibit high sequence specificity, such as the RBP PUM2, which features the Pumilio homology domain, allowing it to bind a well-defined 8-nucleotide sequence motif [85]. RBP binding site specificity can also be increased by interactions with other RBPs, which includes structure remodelling to modify binding behavior [136], by the cellular context (i.e., co-localization of interaction partners), or by RNA modifications [137]. Apart from recognizing certain sequence elements, RBPs can also have an affinity for structured RNA, or a combination of sequence and structural elements. For example, Roquin and SLBP bind to conserved stem loop structures with specific loop nucleotide compositions [138, 139]. Binding to stem loops is likely not uncommon among RBPs, as a large-scale in vitro study on RBP binding preferences has found that around 30% of the observed 86 RBPs showed affinity for stem loop motifs [140]. Moreover, DSRM-containing RBPs such as Staufen proteins can recognize double-stranded RNA (dsRNA) through the shape of their backbone, but also to some extent the underlying sequence [141]. The target dsRNA site can be formed both from intramolecular and intermolecular interactions, e.g., as part of the target RNA structure, or between an mRNA and a lncRNA [142]. Furthermore, various studies have pointed out the impact of site context on RBP binding, including the surrounding sequence and structure composition, and its influence on site structure and accessibility [143, 144]. Based on their fundamental cellular functions, it is not surprising that RBPs are also involved in many diseases, such as cancer or in general genetic diseases which are caused by mutations in functional regions of the respective RBPs [145, 11].

2.2 Experimental methods

The following section briefly introduces the experimental methods that produced the high-throughput sequencing data used to generate most of the results in this thesis. This includes a primer on high-throughput sequencing and the detection of RNA-protein interactions, as well as a more detailed description of the CLIP-seq protocol and its most commonly applied variants.

2.2.1 High-throughput sequencing

Experimental methods to determine (or read out) the nucleotide sequence of DNA or RNA molecules are generally referred to as sequencing methods. The emergence of high-throughput sequencing (also termed second-generation or next-generation sequencing (NGS)) methods in the mid 2000s has revolutionized biomedical research, as it led to a steep decline in sequencing times and costs, thanks to the methods' massively parallel sequencing design. For example, the costs of sequencing the human genome have dropped from around 100,000,000$ in 2001 to as little as 1,000$ in 2021 [146]. Besides the sequencing of DNA, NGS was quickly

adapted to RNA sequencing (RNA-seq) [17]. The key concept behind NGS is the massively parallel sequencing of millions of short DNA fragments in a single machine run, which is why NGS methods are also referred to as short-read sequencing methods. The determined short read sequences subsequently can be used for example to assemble a genome or transcriptome, or to map them to an existing reference genome or transcriptome. RNA-seq thus for instance allows to measure and compare gene expression between different experimental conditions, observe expressed transcripts and splice isoforms, or to detect novel transcripts. Upon their successful introduction, short-read RNA-seq methods have been adapted and become parts of many more specialized transcriptome-wide experimental procedures. These include methods to detect DNA-protein interactions (e.g., ChIP-seq), RNA-protein interactions (e.g., RIP-seq, CLIP-seq), RNA-RNA interactions (e.g., PARIS, SPLASH, LIGR-seq), ribosome footprints (Ribo-seq), or the cellular entity of RNA structures (e.g., SHAPE-seq) [147]. Illumina short-read sequencing platforms have been by far the most widely applied ones to generate NGS data, as of 2019 accounting for $> 95\%$ of the NGS data deposited in the Sequence Read Archive (SRA) [147]. In the following we will therefore focus on the Illumina sequencing technology, also known as "sequencing by synthesis".

Sequencing library preparation starts with DNA or RNA extraction from the observed biological sample. For RNA, this often involves two options: enrichment of poly(A)-tailed RNAs or total RNA extraction with rRNA depletion. The first approach is more exclusive with a focus on mRNAs (but also other Pol II transcripts such as most lncRNAs), since only poly(A)-tailed RNAs are selected, while the second one also allows the recovery of RNAs without poly(A) tails, or in general fragmented RNAs. As rRNA constitutes by far the largest amount of RNA in cells, rRNA depletion (typically through binding to DNA oligos and enzymatic digestion) is necessary in order to focus sequencing on the RNA species of interest [148]. This is usually followed by RNA fragmentation, which is necessary mainly due to the read length restrictions of short-read sequencing platforms (commonly 100 to 300 bp), thus allowing a relatively even coverage of the transcriptome. The read length limitation mainly stems from dephasing issues, where nucleotide read-out from a cluster (see next paragraph for details) becomes increasingly out-of-phase (i.e., desynchronized extension of individual cluster strands), and thus more noisy after each sequencing cycle. The next step is to synthesize complementary DNA (cDNA) strands from the fragmented RNAs, also termed reverse transcription (RT), which allows the use of conventional DNA sequencing machines. In addition, DNA is chemically more stable than RNA, mostly because of RNA's 2'OH group reactiveness and the notorious pervasiveness of RNases. cDNA synthesis is frequently coupled with sequencing adapter ligation. For this, platform-specific adapters necessary to sequence the fragments are added to both ends of the RNA fragments before cDNA synthesis, avoiding the use of oligo(dT) or random primers [148]. A primer is a short DNA sequence complementary to the sequence to be replicated, which DNA polymerases need to start strand synthesis, i.e., by adding nucleotides to the 3' end of the primer. In order to retain the strand information, i.e., from which strand the sequenced fragment originated from, there are also several methods available, such as applying specific 5' and 3' adapters,

or the use of dUTP in second strand synthesis, to degrade the second strand and suppress its amplification [149]. If the amount of cDNA is less than the amount of input DNA required by the sequencing machine, it is further necessary to add a PCR amplification step, i.e., the cyclic duplication of cDNAs by the famous polymerase chain reaction (PCR) method (awarded Nobel Prize in chemistry 1993) [150]. Unfortunately, differing cDNA lengths and compositions can lead to uneven amplification (i.e., PCR biases), resulting among other things in the accumulation of PCR duplicates. To detect these, short typically random sequences termed unique molecular identifiers (UMIs) can be introduced into the sequencing adapters or RT primers. As PCR duplicates originate from the same PCR template, they will contain the same UMI, which allows to collapse them into one read later on in the computational analysis. This is especially important for applications where read counts or in general read coverage is used to infer results, such as in RNA-seq or CLIP-seq. In general, the choice of preparation steps determines the bias types introduced into the sequencing library [148]. It therefore makes sense to keep some common biases in mind, which otherwise can become an issue in subsequent data analysis steps.

The Illumina sequencing technology was originally commercialized by Solexa in 2006, before being acquired by Illumina the following year. It is based on a sequencing-by-synthesis chemistry using four reversible terminator nucleotides labelled with different fluorescent dyes [151]. The two main steps carried out in the sequencing machine are cluster generation and sequencing by synthesis. First, the purified DNA is washed over the flow cell, which is coated by a lawn of two different oligos, complementary to the two adapter sequences. The DNA fragments spread out on the flow cell and their ends basepair with the complementary oligos. Next, a copy of the hybridized fragments is created with the attached oligo as primer, and the fragments are washed away, leaving their reverse complementary (RC) strands covalently attached to the surface. This is followed by bridge amplification, which involves the strands being hybridized with the second surface oligo and copied with the second oligo as a primer. Through repeated bridge (or clonal) PCR amplification, clusters of identical strands are generated (cluster generation), typically containing around 1,000 copies per cluster, and 10s to 100s of millions of clusters over the entire flow cell. At the end, the RC strands are cleaved and washed off, leaving the remaining oligos and the forward strands which are now ready to be sequenced. Sequencing by synthesis starts with sequencing reagents (including DNA primers, polymerases, and the labelled nucleotides) being washed over the cell, and attached to the chain. Due to the reversible terminator, only one nucleotide per cycle is added, and a camera system detects the identity of the nucleotide by the fluorescent light the attached fluorescent dye is emitting upon excitation (also termed base calling). To reliably call the added base, one light signal is recorded for an entire cluster, in order to get sufficient light intensity. After detection, fluorescent dye and terminator are cleaved from the nucleotide, and a new cycle begins. This is repeated until the desired read length is obtained. Ideally, strand extension of a cluster is synchronized, resulting in a unison fluorescent signal. However, due to dephasing issues from uneven extension, base calling becomes increasingly unreliable from cycle to cycle, until the noise exceeds the signal

and prevents further sequencing. It is also possible to sequence multiple samples per run (also termed multiplexing), by using sample-specific barcode sequences (also termed indices) as parts of the sequencing adapters. Moreover, the fragments can be sequenced from one end (single-end, as described) or from both ends (paired-end). For this, the process is repeated by copying the present forward strands, cleaving them off and sequencing the RC strands with the same number of cycles. Even though more expensive, paired-end sequencing allows for a more accurate read mapping, including the detection of genetic or splice variants, and thus is usually preferred over single-end sequencing if possible.

Despite their huge success, short-read RNA-seq methods also have a number of drawbacks, which besides platform-specific biases [152] are mainly related to their limited read lengths: multi-mapped reads can occur more often, which due to their ambiguity usually cannot be considered in data analysis. Furthermore, it is often difficult or impossible to detect new or differentiate between certain splice isoforms, as reads cannot always be unambiguously assigned to one isoform. Long-read or direct RNA-seq methods, which are currently transforming the sequencing field, avoid these problems through their extended read lengths, opening up exciting new possibilities in biomedical research [147] (also see the thesis outlook 4.2). For example, they enable direct end-to-end sequencing of native RNA molecules, including the detection of modified bases [153]. Still, short-read RNA-seq methods so far provide a much higher throughput, lower error rates, and are well established, and thus will likely not fade away any time soon. Moreover, individual strengths of short- and long-read methods can be combined to create interesting hybrid sequencing approaches, e.g., to improve genome assemblies [154].

2.2.2 Detecting RNA-protein interactions

Methods for the detection of RNA-protein interactions can be grouped into RNA-centric and protein-centric methods [155]. While the first group typically starts with an RNA molecule to detect its protein interaction partners, the second group takes an RBP to identify its RNA interaction partners. The two approaches can thus be utilized as complementary methods, in order to get a more complete picture of cellular RNA-protein interactions and the compositions of RNP complexes.

RNA-centric methods can further be subdivided into in vitro and in vivo methods, and the second group again into cross-linking and non-cross-linking-based approaches [155]. In vitro methods typically use immobilized in-vitro-transcribed RNA, which gets incubated with a cell lysate, to extract and identify proteins bound to the RNA. Their in vitro design makes it easy to conduct mutagenesis studies, i.e., by mutating parts of the RNA or protein to better understand their binding contributions. An obvious drawback of in vitro methods is their inability to capture certain in vivo properties that might fundamentally influence binding, such as cellular RNA localization, structure, modifications, or RNA-RNA interactions. In vivo methods that use cross-linking apply UV radiation or formaldehyde to cells in order to covalently cross-link proteins with RNA, which helps to preserve specific and get

rid of unspecific interactions during RNA extraction. Both cross-linking approaches have their pros and cons: UV cross-linking is more specific, cross-linking only close interactions between RNA and proteins, while formaldehyde also can cross-link and thus recover typically less tight protein-protein interactions. This means that UV cross-linking could miss less stable interactions, e.g., for structure-binding RBPs [156], whereas formaldehyde cross-linking might also recover proteins that do not directly interact with RNA. A number of these methods have been proposed over the years, using various experimental strategies to extract the target RNA after cross-linking and cell lysis. For example, RNA affinity purification (RAP) uses biotin-labelled nucleotide probes which hybridize to the RNA of interest, this way enabling its purification via streptavidin beads and subsequent identification (typically via mass spectrometry) of cross-linked proteins [157, 158]. In contrast, in vivo methods based on proximity proteomics circumvent cross-linking by recruiting a specific labelling enzyme to the target RNA, which subsequently labels the bound proteins, enabling the identification. For example, RNA-protein interaction detection (RaPID) utilizes a modified biotin ligase together with a stem loop RNA element fused to the RNA of interest. The modified enzyme binds to the stem loop with high affinity, and biotinylates all proteins in its neighborhood (within $\approx$ 10 nm) [159]. This way, biases from cross-linking and RNA purification are omitted, which also allows the detection of more transient interactions, as well as using less cells. On the downside, the construct of stem loop and target RNA has to be brought into the cells and expressed artificially, causing various other issues (e.g., non-physiological concentrations). In addition, the range of biotinylation limits the approach to shorter RNAs, or in general local RNA environments.

Protein-centric methods typically involve purifying the protein of interest, together with its bound RNAs. The by far most commonly used set of methods uses UV cross-linking of RNA-protein complexes in combination with immunoprecipitation (CLIP) of the protein of interest, together with high-throughput sequencing of the cross-linked RNAs (CLIP-seq, described in the following sections). As such, these methods rely on a sufficient UV cross-linking efficiency of the observed RNA-protein interactions, as well as a good antibody specificity. For RBPs with lower cross-linking efficiencies (e.g., double-strand-binding RBPs), it might thus make sense to omit UV cross-linking, instead relying on formaldehyde [160], or to skip cross-linking altogether. The second option is more common, and is also known as RNA immunoprecipitation sequencing (RIP-seq) [161]. This has the advantage of capturing less proximate (or more transient) interactions as well, although signal-to-noise ratio and resolution are typically lower than in CLIP-seq [155]. Other options to increase cross-linking efficiency are the use of modified nucleotides (e.g., 4-thiouridine (4SU) in PAR-CLIP [162]), or different cross-linking reagents (e.g., methylene blue for dsRNA-binding RBPs [163]). In addition, there are also some more recent methods that instead of protein purification rely on the chemical modification of interacting RNAs. For example, targets of RNA-binding proteins identified by editing (TRIBE) fuses the catalytic domain of an A-to-I RNA editing (ADAR) enzyme to the RBP of interest, which catalyzes deamination of adenosines at the RBP binding site, this way allowing the identification of RBP binding sites through char-

acteristic A to G mutations in the RNA sequencing data [164]. This way, the number of preparation steps is reduced, as well as the amount of required cells. On the other hand, overexpression of the fused RBP can be problematic, as well as the fused domain itself, potentially influencing the functions and binding of the RBP. Still, further improvements might make these methods an interesting alternative to CLIP-seq in the near future, especially for RBPs which are hard to CLIP.

2.2.3 The CLIP-seq procedure

CLIP-seq is currently the by far most widely applied experimental procedure to identify RBP binding sites on a transcriptome-wide scale. The identification of RNAs bound by a specific RBP through cross-linking and immunoprecipitation (CLIP) was initially proposed in 2003 [165]. In 2008, the first high-throughput sequencing extension HITS-CLIP (high-throughput sequencing of RNA isolated by CLIP) was presented, allowing the transcriptome-wide identification of RBP binding sites [166]. More CLIP-seq variants followed over the years, each providing their own take on improving the original protocol [18]. The currently most popular ones are PAR-CLIP (photoactivatable-ribonucleoside-enhanced CLIP) [162], iCLIP (individual-nucleotide CLIP) [167], and eCLIP (enhanced CLIP) [168]. In addition, several CLIP-seq variants specialized on detecting the binding sites of dsRNA-binding RBPs have been described [169, 170, 171]. CLIP-seq typically identifies the transcriptome-wide binding sites for a single specific RBP, but it can also be applied to obtain global RBP occupancy profiles, i.e., which transcript regions are bound by RBPs [172].

Figure 2.7 depicts the principle CLIP-seq workflow (for variant-specific steps see the following section). The process starts with UV radiation of the observed cell or tissue culture, to covalently cross-link cellular RBPs to their bound RNAs. UV radiation is typically done at a wavelength of 254 nm (apart from PAR-CLIP), which has long been known to covalently cross-link nucleic acids and proteins [174]. Nearly all amino acid residues (except 4) have been shown to cross-link to RNA at 254 nm, whereas the preferably cross-linked nucleotide is U [175]. Cells are subsequently lysed, followed by RNA fragmentation through partial RNase digestion, and immunopurification of the RBP of interest together with its crosslinked RNA fragments (i.e., corresponding to its binding sites). After 3' adapter ligation, stringent purification of RBP-RNA complexes continues with a denaturing (sodium dodecyl sulphate (SDS)) polyacrylamide gel electrophoresis (PAGE) step, which separates the complexes in the sample by their molecular weight. Here the desired RNA-protein complex is extracted (together with nitrocellulose membrane transfer to remove unbound RNA, not shown) based on its known weight (i.e., molecluar weight of the RBP + estimated weight of the attached RNA fragments). Visualization of RBP-RNA complexes at this point also serves as a control step, i.e., to assess sample and purification quality, and is typically achieved by radioactive labelling (except from eCLIP). This is followed by proteinase K treatmeant, to cleave off the RBP from the RNA fragments, and RNA isolation. The RNA fragments are subsequently reverse transcribed into cDNA, amplified by PCR, and submitted to high-throughput se-

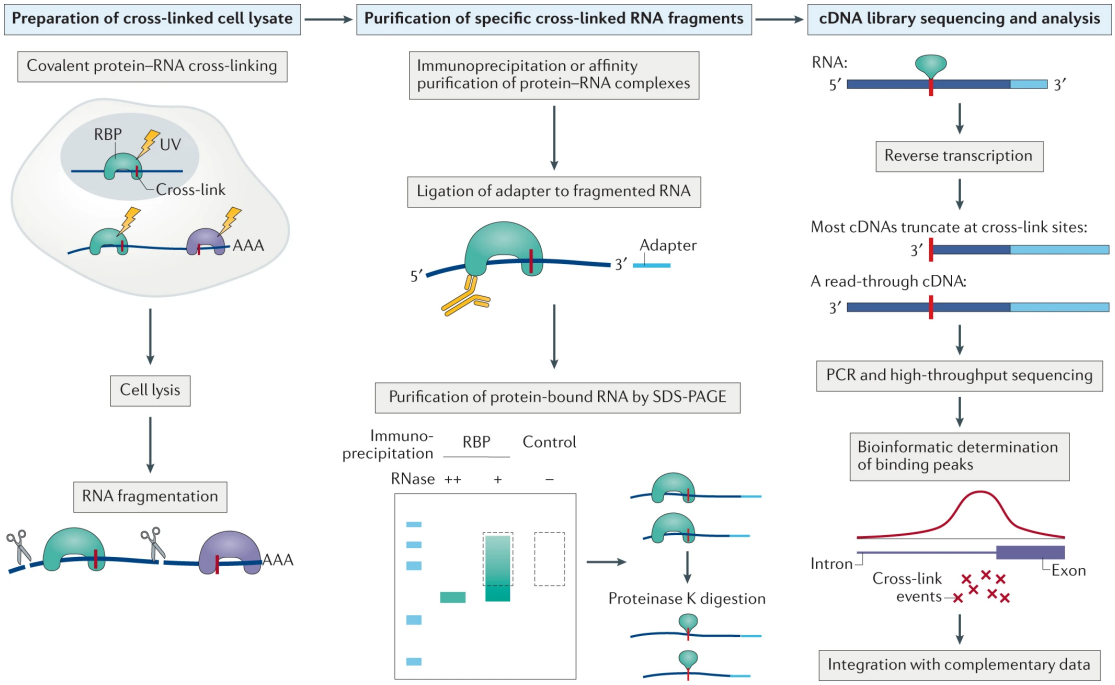


Figure 2.7: Principle CLIP-seq workflow. RBPs are crosslinked to their interacting RNAs by UV radiation. Cells are lysed, RNAs fragmented, and the RBP of interest immunopurified together with its cross-linked RNA fragments. After 3' adapter ligation, stringent purification of RBP-RNA complexes continues with SDS-PAGE, followed by proteinase K digestion of the RBP, and RNA isolation. RNAs are reverse-transcribed to cDNAs, often resulting in cDNAs truncated at cross-link sites. cDNAs are consequently amplified by PCR and sequenced. Data analysis follows, including the determination of binding regions (i.e., peak calling), as well as further data analysis and integration with different datasets. Reprinted by permission from Springer Customer Service Centre GmbH: Springer Nature, Nature Reviews Methods Primers, Hafner et al. [173], copyright 2021.

quencing. Reverse transcription results both in cDNAs that extend over (i.e., read through), as well as cDNAs that truncate at the the cross-link site, presumably due to physical hinderance by the remaining amino acid residue. Protocol variants like PAR-CLIP do not amplify truncated cDNAs, while variants like iCLIP or eCLIP can recover both truncated and read-through cDNAs. Read-through cDNAs also frequently contain characteristic mutations at their cross-link positions, which is exploited (e.g., by PAR-CLIP) to achieve single-nucleotide resolution in subsequent computational binding site detection. After sequencing, computational analysis of sequenced reads includes pre-processing (e.g., quality control), mapping of the reads to a reference genome or transcriptome, as well as binding site detection (i.e., peak calling) from the mapped read profiles. Afterwards, the resulting peak regions can be further analysed and complemented with other types of data, to learn more about the binding properties and functions of the RBP.

2.2.4 CLIP-seq variants

PAR-CLIP

Compared to the original HITS-CLIP, PAR-CLIP serves photoactivatable ribonucleoside 4-thiouridine (4SU) to the cell culture dish, which the cells readily incorporate into RNA instead of uridine [162]. The substitution considerably increases UV cross-linking efficiency at 4SU positions, which is done at 365 nm. Moreover, the modification results in a high number of T to C mutations during cDNA synthesis at cross-link sites. This makes it possible to pinpoint the cross-link positions through mutational analysis, effectively achieving single-nucleotide resolution in binding site identification. On the other hand, 4SU addition can also be seen as a drawback, since it is not clear how much the uridine analog influences celluar well-being, and in particular RNA interactions with other molecules. For example, 4SU has an increased affinity towards G:U base pairing [176], which might modify cellular RNA structures and thus also RBP binding. Other problems include the use of inducible tagged proteins instead of endogenously expressed proteins, or the application of RNase T1 for RNA fragmentation, which results in the depletion of G-containing binding sites, although these issues have been addressed in subsequent publications [177]. Several PAR-CLIP adaptations have been proposed over the years, from iPAR-CLIP (in vivo PAR-CLIP in the model organism *C. elegans*) to fPAR-CLIP (fluorescence-based PAR-CLIP, discarding radioactive labelling) [178, 179]. Overall, PAR-CLIP has been applied in many works, including a large-scale study with 64 distinct RBPs [180].

iCLIP

HITS-CLIP and PAR-CLIP by design are unable to detect truncated cDNAs, which originate from reverse transcriptase stalling at cross-link sites due to left-over peptide residues. iCLIP [167] addresses this drawback, which as shown can result in the loss of up to 80% of cDNA fragments [181]. The issue is solved by using a two-part cleavable adapter combined with an additional circularization step. Moreover, the introduction of UMIs into the adapters allows for an easy removal of PCR duplicates. Single-nucleotide resolution is achieved by the observation that most cDNAs truncate at the cross-link position, this way pinpointing the position to the read 5' end. However, like HITS-CLIP and PAR-CLIP, the protocol remains time-intensive ($\approx$ 5 days) and there are various error sources due to the many preparation steps [182]. Moreover, in order to generate a sufficiently complex library, the amount of cells needed is relatively high ($\approx 10^6$ to 10^8 cells), which can be problematic especially for lowly expressed RBPs or RBPs where CLIP in general is more difficult. Recently, two improved versions (named iCLIP2 and iiCLIP) have been proposed, promising faster, more efficient workflows and increased library complexities [183, 184].

eCLIP

Inspired by the shortcomings of previous methods, eCLIP takes on improving preparation times and efficiency, mainly by omitting or modifying certain protocol steps [168]. In particular, the inefficient circularization step from iCLIP gets replaced by two separate adapter ligation steps, while radioactive labelling of RBP-RNA complexes is completely omitted. This considerably increases the amount of recovered RNA, which also results in a much lower amount of input material (typically around 20,000 cells) [182]. Just like its cousin iCLIP, eCLIP achieves single-nucleotide resolution through the recovery of truncated cDNAs. In addition, eCLIP introduces a new control library concept, termed size-matched input control, which promises to improve background normalization and thus the specificity of recovered binding sites. On the downside, omission of radioactive labelling in eCLIP means that IP quality can no longer be monitored, which can influence the quality of results [18]. Being the CLIP-seq method of choice for the ENCODE consortium, eCLIP is currently the most widely applied method by the numbers of CLIPped RBPs, which currently amounts to 150 RBPs (altogether 223 eCLIP datasets) [185].

2.3 Computational methods

This section briefly describes the computational methods and concepts necessary to understand the presented work. At first, an overview on sequencing data analysis is given, which basically applies to both RNA-seq and CLIP-seq data analysis. Next, an outline of common methods for the computational prediction of RBP binding sites is supplied. Finally, an introduction into deep learning is given with a focus on binary classification through supervised learning and recurrent neural networks (as applied in publication P3).

2.3.1 Sequencing data analysis

Computational analysis of short-read sequencing data such as RNA-seq or CLIP-seq data basically involves three major steps, namely pre-processing of raw sequence data, mapping of processed reads, and the actual data analysis to draw conclusions from the experimental data. In the case of CLIP-seq, another major step before the actual analysis is peak calling, i.e., the identification of RBP binding sites from the mapped read data. The following briefly describes these steps. A more detailed review on commonly used RNA-seq and CLIP-seq data analysis tools and workflows can be found, e.g., in the following papers [186, 187, 188].

Pre-processing of raw sequencing data involves various quality control steps, such as the removal of adapter, UMI, and low-quality sequence parts, as well as PCR duplicates. Moreover, if the sequencing data is made up of several samples, it has to be de-multiplexed, i.e., through index-specific assignment of each read to its respective sample. Numerous quality control tools are available, which can detect and remove known or repetitive sequences in the read library. These can stem from adapters, barcodes, or even contamination due to

sloppy lab work. In addition, they can trim off bases based on specified quality cutoffs, as the raw sequencing data coming from the sequencing machine includes quality scores (i.e., base call accuracies) for each base. In addition, PCR duplicates are removed, e.g., by UMI-tools [189], to obtain more accurate read counts (more on UMIs in Section 2.2.1).

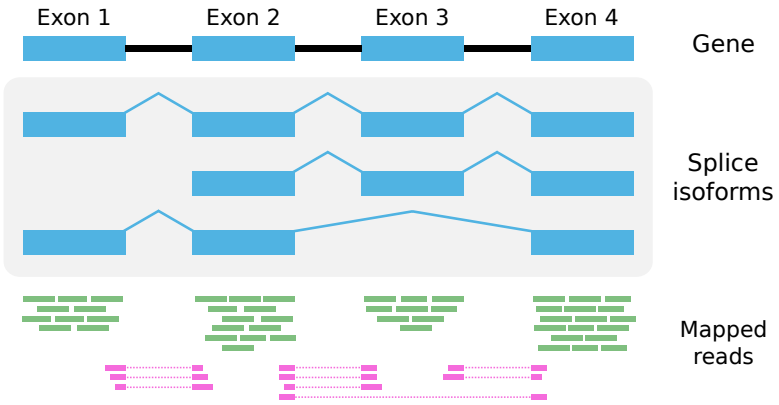


Figure 2.8: Mapped RNA-seq reads to a gene with multiple exons and splice isoforms. Continuously mapped reads (green) cover the exons, while for intron-spanning reads (pink) the mapping is split, resulting in a two-part mapping to separate exons.

After the raw sequencing data has been checked and filtered in pre-processing, the reads are ready to be mapped to the respective reference genome or transcriptome. This is necessary as the raw reads do not contain any information on where they originate from in the genome or transcriptome, except for their sequence. The read sequence is thus aligned against the set of reference sequences (chromosomes or transcripts), with the best match being the most likely region of origin. As sequencing errors can occur during library preparation as well as sequencing, a certain amount of mismatched bases (i.e., mutations) are allowed too, as well base deletions or insertions. If there is a single best match, the read becomes a uniquely-mapped read. If there are no matches of sufficient quality, the read is assigned as unmapped. If there are several equally good matches, the read becomes a multi-mapped read, which frequently happens if the read originates from repeat-rich regions. In addition, reads originating from multi-exon transcripts which are mapped to the reference genome can also become split reads (also termed intron-spanning or chimeric reads) (illustrated in Figure 2.8). As many genes contain multiple splice isoforms, mapping of short-read sequencing data can be ambiguous. For example, a read might map to an exon which is shared by several splice isoforms of the same gene. In this case, intron-spanning reads can help to resolve which isoform is the dominant isoform, or in general which isoforms are present in the observed sample. Nevertheless, certain unambiguities cannot be resolved by short-read sequencing, which is one of the main reasons why, despite of their own drawbacks, direct or long-read read sequencing have become increasingly popular in recent years (see Section 2.2.1 for more details). To be able to map split reads, a splice-aware mapping software such as STAR [190] has to be used, which typically also allows the discovery of novel splice junctions. Mapping accuracy and rates can be further enhanced by using paired-end sequencing instead of single-end sequencing. Here, twice the read information is obtained by

sequencing the cDNA fragments from both ends, which if the fragment size is known and sufficiently large, allows for a much more precise mapping, especially for multi-exon genes. CLIP-seq data is usually treated no different from RNA-seq data during the mapping stage. After mapping, read coverage refers to the number of mapped reads overlapping with a given reference position, whereas read depth refers to the total number of sequenced reads in the sample before mapping. The read coverage distribution over the reference is also termed read profile.

Following pre-processing and mapping, the actual data analysis deals with obtaining interesting new insights from the mapped read information. For RNA-seq data, this often includes differential gene expression analysis, where read counts of genes are first normalized and then compared between different conditions [186]. However, there are countless other applications for RNA-seq, such as variant detection, transcript discovery, alternative splicing analysis, or de novo transcriptome assembly. For CLIP-seq data, data analysis is usually preceded by peak calling, i.e., the identification of binding sites from the read profiles [187]. The identified binding sites can subsequently be analysed in a number of ways. For example, RBP functions can be deduced from the set of target RNAs (i.e., their gene functions) and binding site locations (e.g., in 3'UTRs or introns). Moreover, the binding sites can be utilized to extract binding motifs or to learn binding properties of the RBP, which can then be used to predict new binding sites on unseen RNA sequences.

2.3.2 Predicting RNA-protein interactions

Many approaches for the prediction of RBP binding sites have been proposed over the years. In the simplest case, sequence motif discovery tools such as the ones included in the MEME suite [191] can be used to identify sequence motifs in the binding sites, which can then be utilized to search for potential binding sites in RNA sequences. From the early 2010s on, more sophisticated approaches appeared, including classical machine learning methods, which incorporated both RNA sequence and structure features. For example, RNAcontext uses a probabilistic motif model encompassing both sequence and structure information [192], while GraphProt applies a combination of graph kernel and support vector machine, which showed superior performance over motif-based approaches [193]. It is not surprising that the appearance and success of machine learning methods such as GraphProt coincided with the increasing availability of large-scale RBP binding site collections in the early 2010s, generated by CLIP-seq or by in vitro methods such as RNAcompete [194]. Machine learning models typically depend on large amounts of training data in order to obtain their superior prediction performances, which these high-throughput experimental methods enabled for the first time. In 2015, the first deep-learning based approach named DeepBind [195] was proposed, using a convolutional neural network (CNN) architecture to train predictive models based on RNA sequence information. From 2015 on, numerous deep-learning based approaches have been presented, typically following the CNN architecture, or expanding it by an additional recurrent neural network (RNN) part [196]. Moreover, certain methods

have incorporated additional predictive features such as secondary structure, evolutionary conservation scores, or region type information such as transcript region annotations. Most approaches treat RBP binding site prediction as a binary classification problem, i.e., they train an RBP-specific model to predict whether a given RNA sequence is bound by the RBP or not, although multi-class classification has also been proposed [197].

As deep learning models are hard to interpret, prediction methods usually also supply visualizations to better understand what the model has learned, and thus also learning more about the RBP binding properties. Various visualization techniques are available, which can be grouped into local and global model interpretability methods [198]. The first group visualizes local (i.e., site-level) position-wise binding preferences, whereas the second group attempts to visualize global model preferences. In addition, visualization techniques can be applicable independent of the network type, such as *in silico* mutagenesis and saliency maps, or specifically for certain network architectures, such as first layer filter visualizations for CNNs. More information on upcoming architectures and trends regarding the computational prediction of RBP binding sites can be found in the thesis outlook (Section 4.2).

2.3.3 Deep learning concepts

Introduction

Since the early days of programmable computers, scientists have been interested in designing intelligent computers, i.e., machines with the ability of human-like cognition and reasoning, a discipline widely known as artificial intelligence (AI) [199]. Inside the realm of AI, the field of machine learning (ML) studies how computers can learn from experience (i.e., data), a prerequisite for intelligent behavior. Instead of applying hard-coded knowledge, ML algorithms obtain their knowledge to achieve a specific task by taking the data at hand and learn relevant patterns from it. ML can be categorized into three major types: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning uses labelled data to learn a model which can predict the target class (classification) or a real-numbered value (regression) for a given input. For example, e-mails could be classified as spam or non-spam based on specific word occurrences, or housing prices could be predicted based on various properties of the house and surroundings. In contrast, unsupervised learning utilizes unlabelled data to learn specific properties of the dataset, e.g., whether the data can be clustered to identify similar subsets. For example, RBPs could be clustered to find RBPs with similar binding characteristics, based on the k-mer frequencies of their binding sites. The third type reinforcement learning encompasses models that learn tasks from trial and error (i.e., without external guidance), typically applied in real-time applications, e.g., to balance a pole, fly a helicopter, or to play video games.

ML methods can be further divided into methods that depend on manually designed features (i.e., that rely on feature engineering by the user), and methods that learn informative features (also termed representations) on their own, directly from the raw input data.

In ML, a feature is defined as a certain characteristic representation of the data at hand, e.g., the frequency of a specific k-mer in an RNA sequence. The ML branch that deals with the second type of methods is therefore also known as representation learning. Representation learning is necessary because for many real-life problems, decisive features are either unknown or highly complex and thus impossible to grasp by humans, making manual feature engineering unfeasible. In addition, hand-designed features are often specific to solving a certain task, urging the need for methods that integrate feature identification into their learning algorithms. Deep learning (DL) again is a part of representation learning, and refers to methods with particularly deep architectures, specifically deep neural networks (DNNs). Due to the strikingly increased popularity of DL methods in recent years, ML methods which do not apply DL are nowadays also referred to as classical ML methods.

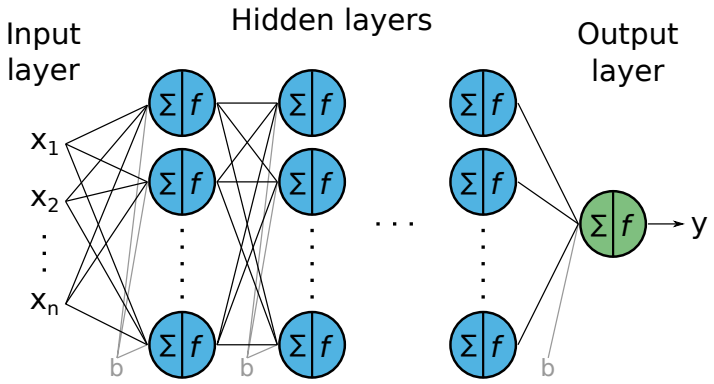


Figure 2.9: Scheme of a common feedforward artificial neural network, also known as multilayer perceptron (MLP). An MLP consists of an input layer with several inputs x, one or several hidden layers of hidden neurons (or units), and an output layer, here with one output unit producing one output value (y), thus suitable for binary classification or regression. Information flows from the inputs to the outputs, with weighted connections between the inputs, hidden and output neurons. The input to a neuron is the sum of the weighted inputs and (usually) a weighted bias term b. The output of the neuron is the sum fed into an activation function f, which is typically non-linear for hidden neurons (e.g., sigmoid or ReLU), while output neurons apply both linear and non-linear functions (depending on the prediction task).

DL methods are based on artificial neural networks (ANNs), named after and inspired by neural networks in the animal brain, which enable intelligent behavior through vast numbers of small interconnected units (i.e., neurons). The prime example for an ANN is the multilayer perceptron (MLP) (see Figure 2.9). An ANN architecture is termed deep if it exceeds a certain number of hidden layers (usually > 2), although there is no strict definition. MLPs are fully connected, feedforward ANNs, meaning that each neuron in the current layer is connected to each neuron in the next layer, and that there are no cycles in the network. The layers that make up a feedforward ANN are also termed linear layers. A neural network (NN) is basically a mathematical function, composed of many simple functions inside its neurons. It can be trained to map its input values to one or more desired output values by tuning the network parameters (i.e., its weights). Training is usually done in a supervised

learning setting, to enable classification or regression of given inputs (see section 2.3.3 *Model training* below for more details). DNNs tackle the challenge of learning representations from raw data by first learning simple features and then in deeper layers combine these into more and more complex or abstract features. For example, a DNN to recognize faces in an image would typically start recognizing simple features in the first layer such as edges, followed by contours and larger facial parts in subsequent layers (see Figure 1.2. [199]). Certain network architectures by design are particularly well suited for specific tasks, such as convolutional neural networks (CNNs) for image recognition, or recurrent neural networks (RNNs) for sequential data learning taks [200].

Input encoding

Categorical features such as the characters of an RNA sequence first have to be converted into a numerical encoding, as deep learning methods require numeric input variables. The most common way to do this is to apply a one-hot encoding (see Figure 2.10) to each character in the sequence. For this, a vector of size $1 \times N$ is created for each character, with N being the size of the alphabet (e.g., $N = 4$ for RNA). The vector consists of 0s in all positions, except for 1 at the position designating the respective character. Additional predictive features can be added to the vector, such as base pair probabilities, or region type annotations (again one-hot encoded), or conservation scores (as done in publication P4). The input to the network thus consists of a matrix of $L \times N$, with L being the length of the sequence.

Figure 2.10: One-hot input sequence encoding. Each character of the RNA sequence is encoded as a one-hot vector of length equals the alphabet size ($N = 4$), with 0s in all positions except for 1 at the position designating the respective character. Additional numeric features can be added to the vector, such as base pair probabilities or conservation scores.

Model training

Training a DNN model essentially involves two steps: (i) fitting the network weights to the training data at hand, which is done by an optimization algorithm; and (ii) making sure that the model is able to generalize well on unseen data, which is done by constantly monitoring the model performance on a separate validation set. The following explanations focus on binary classification, although the basic principles of model training apply to both classification and regression tasks. In a typical DNN binary classification task, the dataset gets

split into a training, validation, and test set. For the optimization, gradient-based methods such as stochastic gradient descent (SGD) and its derivatives (including adaptive gradient methods such as Adam [201]) are typically used. SGD is a first-order iterative optimization method which iteratively adapts the network weights such that the error on the training set gets minimized. More specifically, in each SGD step, a single instance of the training set is put through the network, and the predicted and true label are compared. The error is calculated using an appropriate differentiable error (or loss) function (for binary classification typically the cross-entropy loss function). The gradient of the loss function is then computed by backpropagating the loss through the network, using the backpropagation algorithm [202]. The gradient is the slope (i.e., first-order iterative) of the loss function at the position defined by the currently set weights. In other words, it is a vector containing the partial derivatives of each network weight with respect to the loss. This makes it possible to decrease the loss, by adjusting the weights into the opposite direction of the gradient. In SGD, the weight update is done for every input example. Alternatively, gradient descent can be performed with batches (i.e., the entire training set), or mini-batches (e.g., 50 input examples), to calculate the average loss and do a weight update for each (mini-)batch. Mini-batches are usually the preferred way for training DNNs, as they provide the most practical solution in terms of computational costs and convergence speed. Typically, training continues over several rounds of training set run-throughs, also termed epochs. In addition, the generalization performance of the adapted network on the validation set is monitored after each epoch. Once the validation error does not decrease anymore for a number of epochs, training is stopped and the model (i.e., a network with a specific weight configuration) with the lowest validation error is stored.

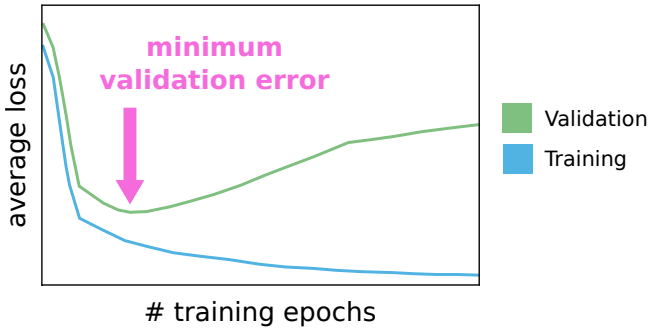


Figure 2.11: Learning curve to observe model fitting and generalization performance. Validation and training set errors are plotted over the number of training epochs. To avoid overfitting, early stopping is applied, i.e., training is stopped once the validation error stops decreasing for a certain number of epochs, and the model with the minimum validation error is selected.

Monitoring the validation error is an essential part of learning, mostly to avoid overfitting. As DNNs are highly complex function approximators, they can easily overfit on the training data, up to a point where the model basically memorizes the training set. This includes learning the noise inherent to the training dataset, which counteracts the model's ability to generalize well on unseen data. In order to obtain a model that generalizes well,

training therefore needs to be stopped once the validation error stops decreasing. Measures to counteract overfitting are also known as regularization techniques. There are various such techniques, which can be applied at different stages of the learning. As regions of large curvature in the network function are connected to large weights, regularization methods typically aim at preventing or penalizing large weights. For example, as described, training is stopped once the validation error does not decrease anymore over a certain number of epochs, which is also termed early stopping (see Figure 2.11). Other techniques such as weight decay add a regularization term to the error function, which increases with larger weights, thus leading to increasingly smaller weight updates. Temporarily decreasing the complexity of the network during training is also popular, as done by the dropout method, which randomly selects and ignores certain neurons (i.e, their contributions to following neurons) during training (i.e., in the forward pass). All these regularization techniques are parameterized, i.e., they can be fine-tuned to improve model performances just like the network parameters. Network parameters which do not get tuned during training but have to be set before training are termed hyperparameters. These include network architecture settings such as the number of layers or the number of neurons per layer, as well as optimizer settings such as learning rate, dropout rate, batch size, or the number of epochs to wait before early stopping kicks in (also termed patience). All these hyperparameters can have large effects on the model's predictive performance, and thus need to be tuned as well, in a procedure called hyperparameter optimization (HPO). Various tools are available for automatic HPO, such as BOHB [203].

In the end, the generalization performance of the selected model is calculated on the remaining test dataset. This is done as the validation set can become biased towards the training set, because model training continues as long as the validation error decreases. In order to get an accurate estimate of the generalization performance over the entire set, the whole procedure is repeated several times through cross-validation. In cross-validation, the dataset is typically split into k equally-sized subsets (i.e., k-fold cross validation, typically with $k = 10$), resulting in k different train-validation-test splits (i.e., for 10 folds: 80%, 10%, 10%). Importantly, each split is used once as a test set, which averages out the bias from using a single small test dataset. In each fold, the model is trained on the training set, monitored on the validation set, and finally evaluated on the test set. This way, k test set performance measures are generated, and the average is taken and returned as the model generalization performance. The procedure can also be coupled with HPO, where in each fold a model is trained for each hyperparameter combination. In this case, the validation set is used for monitoring, as well as for the selecting the best hyperparameter combination.

Recurrent neural networks

Recurrent neural networks (RNNs) describe a specific type of NN architecture, which can take sequences of variable length as inputs and process the input sequence one step at a time (i.e., in a recurrent fashion). In contrast to feedforward NN hidden layers, an RNN layer is made up of only one neuron, also termed cell, which takes the input sequence parts one at a

time, with the cell output being part of the cell input in the next time step (see Figure 2.12). This cycle is repeated until the last sequence part was processed, resulting in a final output which can then be forwarded to a linear output layer to produce the network output, just as in feedforward NNs. Intermediate RNN layer outputs (i.e., at each time step) can also be output, depending on the application. Typically the outputs of the RNN cell correspond to its hidden state vector h, which gets recurrently fed into the cell throughout the cycles. In addition, the basic RNN cell also has two weight matrices and a bias vector. Formally, at time step t, the cell computes the hidden state vector h_t by:

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b) \tag{2.1}$$

where f is a non-linear activation function (in basic RNN cell implementations usually the hyperbolic tangent (tanh) function), W_{xh} the weight matrix connecting the input feature vector to the hidden state vector, x_t the input feature vector at time step t, W_{hh} the weight matrix connecting the previous with the current hidden state, h_{t-1} the hidden state vector at time step $t-1$, and b the bias vector connecting the bias weight to the hidden state vector. An RNN layer processing an input sequence of length 10 can thus be seen as a feedforward NN encompassing 10 single-neuron hidden layers, with the distinction that the weights are shared between the layers and new input is supplied in each layer. The hidden state vector length is also referred to as the dimension of the RNN layer, while a 2-layer RNN means that the final hidden state serves as input to the next RNN layer. Furthermore, an RNN layer can be extended to a bidirectional RNN (BRNN), meaning that the sequence is processed both in forward and backward direction [204]. A BRNN thus contains separate weights for each direction, and outputs two final hidden state vectors, which then get concatenated and forwarded to the output layer. This way, both past and future time steps (or for RNA sequences both up- and downstream context) are considered by the model, which depending on the task or dataset can improve predictive performance.

As long sequences require the RNN cell to unfold many times, backpropagation for standard RNNs can run into the problem of vanishing or exploding gradients. Both hinder or prevent the optimizer to converge during model training, either through oscillating weights that diverge from a good solution (exploding gradients), or too small or falsely-directed weight updates (vanishing gradients). Likewise, the problem makes it difficult for standard RNNs to capture long-term dependencies between more distant time steps [205]. In order to counteract the vanishing gradient problem and to better capture long-term dependencies, different gated RNNs have been proposed, most prominently the long-term short-term (LSTM) unit and the gated recurrent unit (GRU) [206, 207]. These methods replace the simple non-linear activation function in the original RNN cell by a small network of different gate structures, which effectively control what content to keep, to add, and to remove from the cell memory. This way, important content will not be overwritten, enabling long-term dependencies to be kept in memory. Moreover, the additive nature of the introduced gates counteracts the vanishing gradient problem [208].

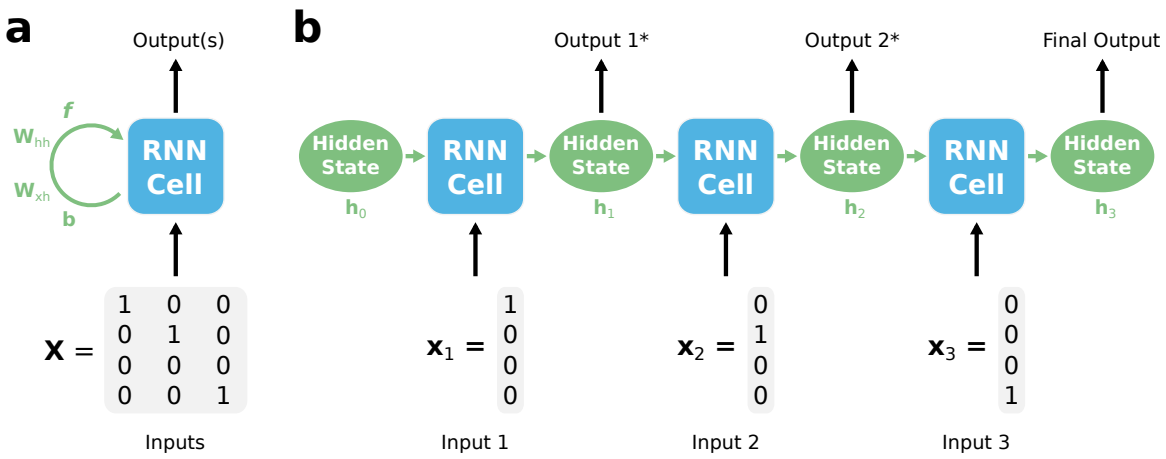


Figure 2.12: Standard RNN architecture showing the forward propagation of information in the RNN layer. **(a)** An RNN layer is made up of a special neuron, the RNN cell, which cyclically processes the input sequence, one step at a time. Example input sequence (one-hot encoded RNA sequence) of length 3, with 4 predictive features. **(b)** unfolded RNN cell, showing the processing of the 3 input sequence parts. At each step, the input feature vector is combined with the former hidden state through weighted connections and put through an activation function. All steps share the same weights. h_0: initial hidden state. *optional outputs.

Predictive performance measures

Various performance measures are applied in ML, depending on the type of task and whether the dataset is balanced or not (i.e., equal numbers of positive and negative examples). In a binary classification setting, all common measures can be deduced from the confusion matrix (see Figure 2.13). Of the measures used or mentioned in this thesis, these include the true positive rate (i.e., sensitivity or recall), the false positive rate, the true negative rate (i.e., specificity), the precision, the accuracy, the F-score, as well as the area under the curve (AUC).

		predicted	
		positive	negative
actual	positive	TP	FN
	negative	FP	TN

Figure 2.13: Confusion matrix for binary classification tasks, with the two classes $\{positive, negative\}$. The confusion matrix contrasts the predicted class with the actual class for each dataset example and displays their numbers, resulting in the four possible outcome categories: true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN).

Based on the four categories (TP, FP, FN, TN) from Figure 2.13, the true positive rate (TPR), also known as sensitivity or recall, is the proportion of correctly classified positive examples:

$$TPR = \frac{TP}{TP + FN} \tag{2.2}$$

Likewise, the false positive rate (FPR) is the proportion of misclassified negative examples:

$$FPR = \frac{FP}{FP + TN} \tag{2.3}$$

In contrast, the true negative rate (TNR), also termed specificity, is the proportion of correctly classified negatives:

$$TNR = \frac{TN}{TN + FP} \tag{2.4}$$

The precision is defined as the proportion of predicted positive examples that were correct:

$$Precision = \frac{TP}{TP + FP} \tag{2.5}$$

The accuracy is defined as the proportion of correctly classified examples (i.e., negatives and positives):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.6}$$

For binary classification, the accuracy of a random classifier is expected to be 0.5, while for multi-class classification it is $1/N$ for N classes. Moreover, for imbalanced classes the accuracy measure might not be the best choice. This is because even a dummy classifier which, e.g., always outputs *negative* can have high accuracies if the dataset contains many negative examples.

The F-score (also known as F1 score) incorporates both precision and recall:

$$F - score = 2 * \frac{Precision \cdot Recall}{Precision + Recall} \tag{2.7}$$

This means that the F-score assesses both how many of the positive predictions were correct, but also how many of the positive examples are correctly classified. Perfect precision and recall results in a maximum F-score of 1.0, while either a precision or recall of 0 results in a minimum F-score of 0. The F-score thus focuses on false negatives and false positives, while the accuracy focuses on true positives and true negatives. Moreover, the F-score can also be used in case of imbalanced sets.

Probably the most common performance measure for binary classification with balanced datasets is the area under the curve (AUC), more precisely the receiver operator characteristic (ROC) curve (see Figure 2.14). The curve is constructed by taking the prediction scores of a binary classification model on the test dataset, and for each test score s calculating the TPR and the FPR with scores $> s$ being positive predictions, and scores $\leq s$ being negative predictions. Consequently, an AUC of 1.0 corresponds to a TPR of 1.0 and a FPR of 0, while an AUC of 0.5 corresponds to a random classifier. The AUC is thus more informative than a single threshold performance measure such as the accuracy, whose threshold can be chosen arbitrarily, and thus can make comparisons between different models or methods difficult. However, for imbalanced datasets, the precision-recall curve is more informative [209].

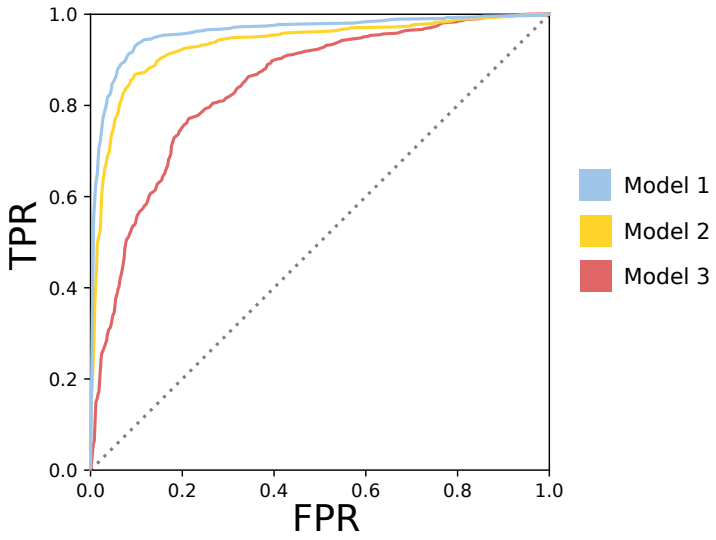


Figure 2.14: ROC curves for three different models, trained on the same dataset with different hyperparameter settings. FPRs and TPRs are recorded for different model score thresholds to construct the ROC curve. The dotted line corresponds to an AUC of 0.5, which would be expected from a random classifier.

Chapter 3

Publication summaries

This chapter summarizes the contents of the five publications (P1-P5) included in this thesis. Each summary contains an overview of the work, as well as a concise description of the methodology (for the method papers P2, P3, P5), and the results and discussion. For full details please refer to the attached published articles, their supplements, and the tool online manuals. The focus of the descriptions will be on my own contributions. However, given that the presented works are collaborative efforts, including some co-author contributions for clarity reasons as well as overlapping contributions (i.e., parts done together with other authors) is inevitable. I therefore opted for using the common first person plural form (i.e., "we" instead of "I") for summarizing the contents of P1-P5 in Chapters 3 and 4, even if a depicted part was done exclusively by myself. Detailed author contributions can be found in Chapter 5, preceding the published articles.

3.1 Computational analysis of CLIP-seq data

This section sums up the contents of the following publication:

- [P1] **Michael Uhl***, Torsten Houwaart*, Gianluca Corrado, Patrick R. Wright, and Rolf Backofen*. **Computational analysis of CLIP-seq data**. *Methods*, 2017.

In addition, the section also references some work which I contributed to papers where I was not a main contributor. While these contributions do not qualify as principal papers that make up the cumulative thesis, they provide application examples for the topics discussed in P1, and thus help to better understand their relevance.

3.1.1 Overview

Over the last decade, CLIP-seq has become the go-to method for determining RBP binding sites on a transcriptome-wide scale. Several protocol variants have been published, each requiring specific adaptations to the computational analysis of the generated datasets. Conducting a CLIP-seq data analysis thus demands detailed knowledge about protocol specifics, as well as the various steps of the data analysis. The lack of comprehensive and up-to-date reviews on the topic prompted us to write this review article (P1), in order to assist readers in performing a successful CLIP-seq data analysis. For the article we mainly drew from our own experience with analysing CLIP-seq data, which already resulted in several publications prior to P1.

The first part of P1 describes the different CLIP-seq protocol variants and their characteristics. In the second part, the various CLIP-seq data analysis steps are discussed. We further divided the data analysis part into three sections, corresponding to the three principal analysis steps: preprocessing of the raw sequencing data and mapping to the genome, binding site identification from the mapped read data (also termed peak calling), and analysis of binding site properties (which also includes the learning of predictive models). Since peak calling is arguably the most critical part, we enhanced this section and added a peak caller comparison.

3.1.2 Results and discussion

Peak caller comparison

As mentioned, various CLIP-seq protocol variants exist, of which PAR-CLIP, iCLIP, and eCLIP are currently the most popular ones. We described their individual characteristics in the first part of P1. For example, PAR-CLIP induces characteristic T-to-C mutations in its read library, which can be utilized to detect crosslink positions, i.e., RNA positions where the protein and RNA got covalently linked via UV radiation as part of the protocol. The

* joint first authors

distribution of crosslink counts across the mapped read profiles thus allows the discovery of RBP binding sites with single-nucleotide resolution. On the other hand, iCLIP and eCLIP pinpoint the majority of crosslink positions to the 5' end of the sequenced reads (second read of the pair in eCLIP, first read of the pair in iCLIP). Consequently, the protocol at hand affects the processing and types of tools applied in the following data analysis, in particular the peak calling. For example, PARalyzer is a peak caller specialized on PAR-CLIP data, by taking into account T-to-C mutations to determine binding site locations from the read data. In contrast, peak callers such as Piranha, CLIPper, or the block-based method described in P1 can be applied to data generated by various CLIP-seq protocols. These can be further divided into methods that only take into account read starts (Piranha), and methods that use the full-length read information (CLIPper, block-based method) to define peak regions. Another popular method called PureCLIP (released shortly after P1) also focuses on read starts [210].

For the peak caller comparison in P1, we chose three popular peak callers which can be applied to various CLIP-seq data: Piranha, CLIPper, and the block-based method (later also implemented in the PEAKachu peak caller). In general, comparing peak callers is difficult, as these tools often include various parameters, which can drastically influence their behavior. Most importantly however, there exists no transcriptome-wide set of experimentally confirmed binding sites for a particular RBP. Without such evidence, ideally recorded under same conditions (cell type and transcriptional state), we cannot be certain whether a CLIP-seq peak region is a true or false positive. Binding motifs have been reported and confirmed for a number of RBPs. However, these cannot safely rule out false positive sites detected in a CLIP-seq experiment, as RBPs often do not have clearly defined motifs, but can bind to sites with varying structure or sequence composition [140]. An alternative is to use an RBP which is known to target a specific set of RNAs. One particularly special RBP in that regard is SLBP (Stem-Loop Binding Protein), which shows high specificity for a stem-loop containing region at the 3' end of histone mRNAs. We thus chose SLBP for the comparison, for which an eCLIP dataset is available.

To make the comparison more even, the identified peak regions of the three peak callers were normalized and filtered by the same procedure , instead of using tool specific quality scores for filtering. Figure 3.1 shows the peaks called by the three peak callers on the histone gene HIST2H2AC (H2AC20), containing a stem-loop motif region (SLM) at its 3' end. As expected, the peaks called by CLIPper and the block-based method (track 6 and 7) follow the full read profile (track 4), while Piranha by design focuses on the read starts (i.e., crosslink positions) (track 5). In this case, Piranha misses the actual SLBP binding site, due to the accumulation of crosslink positions upstream of the SLM. A known issue of CLIP-seq is that double-stranded regions are less efficiently crosslinked with UV [156], which might explain the scattering of crosslink positions around the SLM. Large-scale *in vitro* binding motif studies suggest that most RBPs recognize short unstructured motifs, so this problem might be less relevant for the majority of RBPs [194, 131, 140]. This is also indicated by the results of a subsequent study from our group, where we can see a clear enrichment of known motifs for

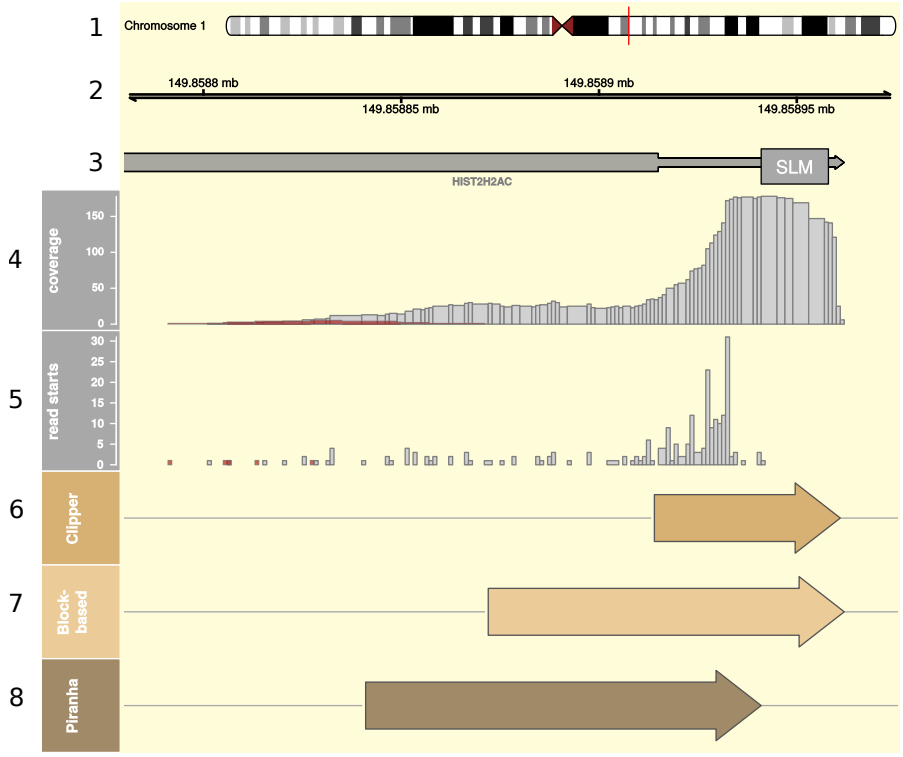


Figure 3.1: Peaks called on human SLBP eCLIP data by three peak callers (CLIPper, block-based method, Piranha), covering the histone HIST2H2AC (H2AC20) gene region on chromosome 1. A stem-loop motif region (SLM) recognized by SLBP is located at its 3' end. 1: chromosomal region, 2: chromosome coordinates, 3: gene annotations, 4: read profile (coverage), 5: crosslink positions profile, 6: CLIPper peak regions, 7: Block-based method peak regions, 8: Piranha peak regions. Coverage of control library in red (tracks 4 and 5). Figure adapted from publication P1 [187].

several RBPs in the called peaks of all tested peak callers [188]. In our comparison, all three peak callers call similar amounts of peaks, with the block-based method calling the broadest peaks and Piranha the smallest (see P1, Table 1). Moreover, they share a substantial amount of called genomic positions (block-based method 42.4%, CLIPper 64.8%, Piranha 88,7%). The higher number of unique positions of the block-based method can be explained by its higher peak number and the increased length of its peaks. Note that we did not use replicate information in this comparison, as CLIPper and Piranha do not support it. The block-based method however does, which in theory should yield more robust binding sites (Pros and Cons of the three tools are listed in P1, Table 2).

In conclusion, both crosslink-position-based (e.g., Piranha, PureCLIP) and read-profile-based peak callers (CLIPper, block-based method) have their advantages and disadvantages. We have seen that structure binding RBPs can be problematic for the first group. An easy solution here is to extend the called sites by a certain amount, although this also increases the amount of non-specific background. The second group of peak callers have their own issues, which are further discussed in publication P4. The support of replicate information is an important feature, which has long become a standard in other high-throughput protocols

(e.g., RNA-seq, ChIP-seq). Unfortunately, many published CLIP-seq experiments consist of few or even no replicates. For example, eCLIP datasets available from ENCODE ([211]) offer two replicates and one control, but ideally one would aim for at least three replicates, both for experiment and control, in order to more accurately estimate the biological variation in the samples [186]. For a comprehensive analysis, we suggest to run more than one peak caller, especially if little is known about the binding properties of the studied RBP. Moreover, if replicate information is available, one should definitely include peak callers which can take advantage of it.

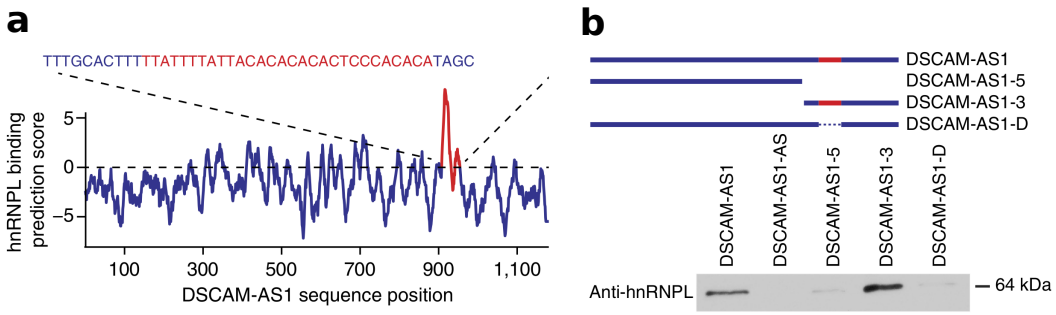


Figure 3.2: hnRNPL binding site prediction on *DSCAM-AS1* lncRNA and experimental verification of the binding site. **(a)** Position-wise binding scores on *DSCAM-AS1* predicted by the GraphProt model trained on hnRNPL iCLIP data. The highest-scoring site (around top-scoring RNA position 923, marked red) was subjected to mutational analysis. **(b)** Mutational analysis results, verifying the hnRNPL binding site on *DSCAM-AS1*. Western blot of hnRNPL and the different mutant forms demonstrated hnRNPL binding to forms including the predicted binding site, but not to forms missing the site. Figures taken from [212] (license: CC BY 4.0 [213]).

Application examples

As detailed in the postprocessing section of publication P1, binding sites identified by CLIP-seq can be further utilized to learn predictive binding models for the CLIPped RBP. The main reason to do this is the expression dependency of a CLIP-seq experiment, meaning that CLIP-seq can only recover binding sites on transcripts which are sufficiently expressed in the studied cell type or condition. Since gene expression is dynamic, we have to rely on binding site prediction tools to get binding profiles for transcripts with no or too low expression, or across the entire transcriptome for systematic studies. One of the collaborations I was involved with included the prediction of hnRNPL binding sites on a newly identified breast cancer-associated lncRNA (*DSCAM-AS1*) [212]. For this I utilized a GraphProt model, which I trained on hnRNPL iCLIP data from the literature. The predicted binding site (Figure 3.2 a) was subsequently verified (Figure 3.2 b), establishing the interaction between hnRNPL and *DSCAM-AS1* as an important factor of *DSCAM-AS1* oncogenicity. To obtain peak regions from the data, I did not apply a peak caller, but instead defined genomic sites with high crosslink counts as peak regions, since the raw data was not available. Notably, the *DSCAM-AS1* gene region did not contain any crosslink positions. In general, it is important

to correct for unspecific binding or expression differences in CLIP-seq by using a control library, e.g., by calculating a fold change (i.e., read coverage of experiment versus control), to filter the peak regions. This is because a strong signal in the experiment could be due to specific RBP binding, but also due to unspecific binding, especially in combination with a high expression of the underlying transcript. Since such unspecific binding signals are ideally detected in the control library as well, using a control can help to remove low confidence sites. On the other hand, machine learning methods such as GraphProt are typically robust to a certain amount of noise in the data, as long as the principal binding preferences are captured in the sites. Whether peak calling or a more simple approach is sufficient therefore depends on the quality of the data, as well as the goal of the study.

In a second collaboration, I applied the block-based method described in P1 to RIP-seq data for the archaeal L7Ae protein, in order to identify novel L7Ae-RNA interactions [214]. L7Ae recognizes and stabilizes specific RNA structures termed kink-turn (k-turn) motifs, which are found in various RNAs, most notably rRNA. L7Ae-RNA interactions are essentially involved in the regulation of translation, and have also been applied in synthetic biology as translational on-off switches. RIP-seq is similar to CLIP-seq, but omits certain steps such as UV crosslinking and RNAse digestion, which leads to a lower specificity and resolution. However, UV crosslinking applied in CLIP-seq requires close association of RNA nucleobases and aromatic amino acid residues, which, depending on the binding modes of the CLIPped RBP, might lead to a decreased sensitivity. RIP-seq can therefore help to uncover RNA-protein interactions with lower specificity or in general more diverse interactions, such as protein interactions with dsRNA or the RNA backbone, or indirect interactions where the immunopurified protein is connected to the RNA via another protein [215]. Due to the decreased resolution, we focused on transcript-level interactions, and chose the block-based method, since the experiment offered replicates (i.e., two replicates and one control). As a result, we identified several novel L7Ae interactions, including ncRNAs as well as mRNAs. Interestingly, L7Ae also binds to its own mRNA to regulate its translation, which was experimentally confirmed in the paper.

Installing and running data analysis pipelines can be challenging, especially for scientists working in the wet lab or with less experience in bioinformatics. Moreover, reproducing earlier results often fails due to a lack of documentation and missing datasets. To help making CLIP-seq data analysis accessible and results reproducible, I integrated several useful CLIP-seq related tools into the Galaxy framework [216]. These include GraphProt [193], PureCLIP [210], and the functional annotation tool RCAS [217], which are also part of the Galaxy CLIP-Explorer pipeline published by our group [188]. Galaxy is a free web-based data analysis platform, driven by an immensely prolific open-source community and tens of thousands of active users from all around the world. It offers easy access to a multitude of different data analysis pipelines, from various scientific disciplines. Results are fully reproducible, since all analysis steps and intermediate results are stored in the project history, and histories and datasets can be shared among users.

3.2 MechRNA: prediction of lncRNA mechanisms from RNA-RNA and RNA-protein interactions

This section summarizes the contents of the following publication:

- [P2] Alexander R. Gawronski, **Michael Uhl**, Yajia Zhang, Yen-Yi Lin, Yashar S. Niknafs, Varune R. Ramnarine, Rohit Malik, Felix Feng, Arul M. Chinnaiyan, Colin C. Collins, S. Cenk Sahinalp, and Rolf Backofen. **MechRNA: prediction of lncRNA mechanisms from RNA-RNA and RNA-protein interactions**. *Bioinformatics*, 2018.

3.2.1 Overview

As detailed in chapter 2, lncRNAs make up a number rich and functionally diverse class of ncRNAs. Estimates of up to 100,000 lncRNAs in the human genome combined with a generally poor functional characterization has spurred great interest among the scientific community in determining their modes of action. In addition, lncRNAs often show tissue- or condition-specific expression, and are upregulated in various cancer types, further contributing to the importance of studying their mechanisms. The function of a lncRNA is essentially determined by its interactions with other biomolecules (RNA, DNA, or protein). Due to the availability of prediction tools for RNA-RNA and RNA-protein interactions, we decided to combine the two approaches with expression data in order to predict lncRNA interactions and infer their mechanisms on a transcriptome-wide scale. Publication P2 introduces this approach we named MechRNA, the first tool to offer mechanistic function predictions for lncRNAs.

P2 starts with describing the method, which essentially consists of an RNA-RNA interaction prediction part using IntaRNA2 [218], an RNA-protein interaction prediction part using GraphProt, a gene expression correlation part including cancer RNA-seq data, and a combination part where the predicted interactions and correlations are merged and potential mechanisms are evaluated using a combined p-value. In the end, MechRNA reports the most likely mechanism (i.e., the mechanism with the lowest combined p-value) for each lncRNA-target RNA pair. This is followed by the results and discussion section, where MechRNA predictions are compared with (partially) known lncRNA mechanisms of 3 prostate cancer-associated lncRNAs. In addition, we ran MechRNA on 5 more lncRNAs to find out more about their potential mechanisms. The results demonstrate that MechRNA is capable of detecting known lncRNA mechanisms, underlining its value for lncRNA research.

3.2.2 Methods

MechRNA workflow

Figure 3.3 outlines the MechRNA workflow. Given a user-specified lncRNA, MechRNA first predicts RNA-RNA interactions (either transcriptome-wide or on a subset), and retrieves precomputed RNA-protein interactions for a set of (up to) 22 RBPs involved in post-transcriptional gene regulation. Next, interaction p-values are merged with precomputed partial correlation p-values of all interaction partners (calculated from a large cancer gene expression set) to obtain a joint p-value, using Stouffer's Z-score method. The partial correlation between two genes (in our case the interacting RNA or RBP genes) is the correlation of their gene expression values, with the effects of all other genes removed. Depending on the combination of interactions (lncRNA, target RNA, RBPs), various candidate mechanisms are generated (see section below for a description of supported mechanisms). For example, if the specified lncRNA and its target RNA have a negative partial correlation, one possible mechanism would be *"direct downregulation"*. In the end, the mechanism with the lowest joint p-value is reported for each lncRNA-target RNA pair.

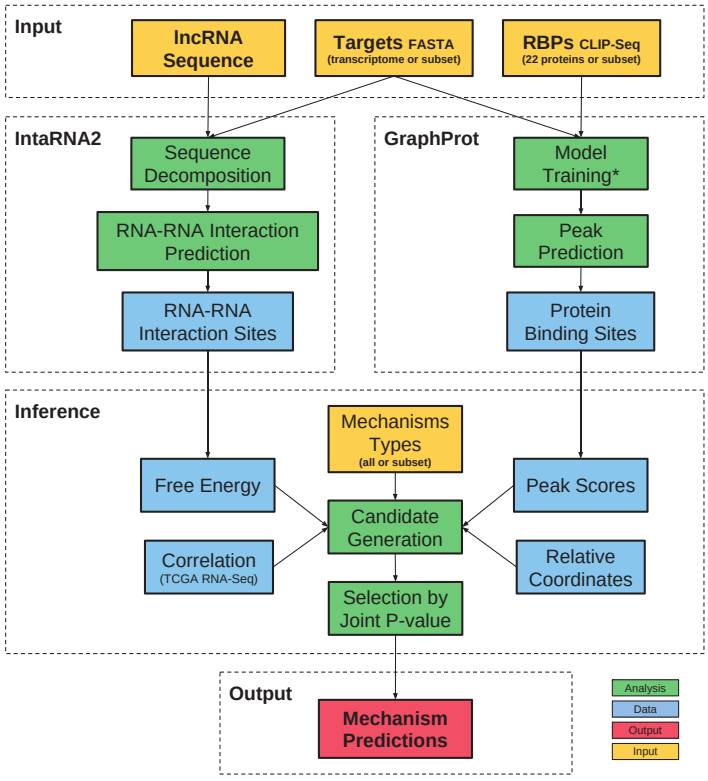


Figure 3.3: Overview of the MechRNA workflow. For a given lncRNA, MechRNA predicts RNA-RNA interactions using IntaRNA2, and retrieves precomputed RNA-protein interactions for a set of 22 RBPs involved in post-transcriptional gene regulation. Resulting p-values are combined with p-values of all interaction partners obtained from gene expression correlation data to a joint p-value. The Mechanism with the lowest joint p-value is reported for each lncRNA-target RNA pair. *precomputed models and predictions to speed up runtime. Figure taken from publication P2 [219].

MechRNA offers two modes of operation: 1) hypothesis-driven and 2) screening mode. In screening mode, the user only needs to specify a lncRNA, for which MechRNA then infers potential mechanisms, using the correlation data, as well as (by default) all available mechanisms, all 22 RBP prediction sets, and the entire reference transcriptome. This is the mode of choice for lncRNAs without any known mechanisms. If more is known about the lncRNA function (e.g., mechanisms or specific targets), the user can also run MechRNA in hypothesis-driven mode, which does not rely on correlation data. Instead, p-values and mechanisms are deduced solely from RNA-RNA and RNA-protein interactions, which are based on the user-specified RNA targets, RBPs, and mechanisms.

Supported mechanisms

Depending on the existence, location, and number of predicted RNA-protein interaction sites relative to the predicted RNA-RNA interaction site, various lncRNA mechanisms can be inferred. Figure 3.4 a visualizes possible mechanisms, as well as the mechanisms which are supported by MechRNA (marked in green). Due to MechRNA's reliance on RNA-RNA interactions for inference, mechanisms without RNA-RNA interactions are not supported (*"direct RBP regulation"* and *"decoy"*). Moreover, *"dsRNA binding"* is not inferred since GraphProt does not predict binding sites occurring between two different RNAs, and CLIP-seq in general does not inform about the RNA structure at the binding site (i.e., which regions are interacting). As for the supported mechanisms, the simplest mechanism *"direct RNA regulation"* involves only the RNA-RNA interaction. The remaining mechanisms all include RBP interactions on the lncRNA, the target RNA (mRNA), or both. In addition, the RBP binding site can overlap the RNA-RNA interaction, and the mechanisms can be further divided into downregulatory or upregulatory (destabilizing or stabilizing), based on the partial correlations between lncRNA or RBP(s) and the target RNA. A list of known lncRNA mechanisms corresponding to the mechanisms in 3.4 a can be found in P2 (Table 2).

Based on the configuration of RBP binding sites at the RNA-RNA interaction site, MechRNA infers different candidate mechanisms and assigns p-values to each mechanism. Figure 3.4 b shows an inference example, with three RBPs (A,B,C). RBP B, RBP C, and the lncRNA are negatively correlated with the target RNA. RBP A is positively correlated with the target RNA, and its predicted binding site overlaps with the RNA-RNA interaction site. This configuration leads to the six shown candidate mechanisms for the given lncRNA-target RNA interaction (described as tuples with the format: *(target_peak, lncRNA_peak, mechanism_type)*. Finally, MechRNA reports the most likely mechanism for each lncRNA-target RNA pair, i.e., the mechanism with the lowest joint p-value. In addition, the user can specify which mechanisms to consider, as well as which RNA targets and RBPs (by default all).

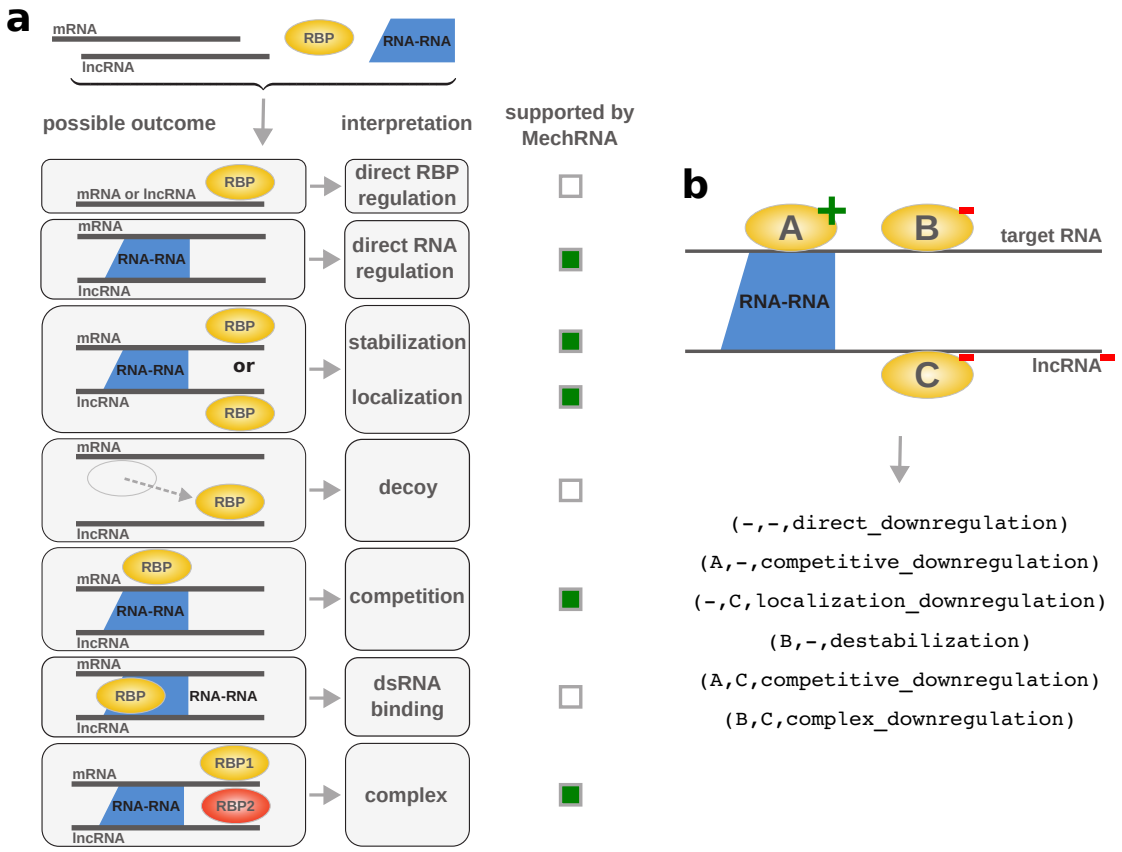


Figure 3.4: Possible lncRNA mechanisms defined by different combinations of RNA-protein interactions with the RNA-RNA interaction. **(a)** Possible mechanisms and mechanisms supported by MechRNA (marked in green). Supported mechanisms can be inferred by MechRNA. Depending on the partial correlations of the interaction partners, certain mechanisms can be further categorized into downregulatory or upregulatory. Figure adapted from publication P2 [219]. **(b)** Inference example with three different RBPs (A,B,C). RBP A binding site overlaps with the lncRNA-target RNA interaction region. RBP A is positively correlated with the target, RBP B, C, and the lncRNA are negatively correlated with the target. Resulting candidate mechanisms extracted by MechRNA are shown below, encoded as tuples with the format: *(target_peak, lncRNA_peak, mechanism_type)*.

Integrating RBP binding site information

As mentioned in section 3.1.2 *Application examples* (also see publication P1, section 3.3), CLIP-seq alone cannot provide a complete transcriptome-wide RBP binding landscape, since binding site detection by CLIP-seq requires sufficient expression of the target RNA. In order to obtain the complete transcriptome-wide binding profiles needed for MechRNA, we therefore relied on the RBP binding site prediction tool GraphProt. More specifically, GraphProt was utilized to train models on a collection of 17 CLIP-seq datasets, encompassing 22 RBPs with known roles in post-transcriptional gene regulation. The discrepancy in numbers is due to two CLIP-seq datasets which contain the merged CLIP-seq regions of 3 and 4 RBPs, respectively (AGO1-4, IGF2BP1-3) [162]. These have been merged by the authors because the RBPs are highly related and bind similar sets of targets. Datasets with called peaks

were obtained from the original GraphProt publication, as well as from ENCODE [211]. We further undertook an extensive literature search to evaluate RBP functions. Based on our findings, RBPs were further annotated as destabilizing or stabilizing the target RNA, which is included as additional information for mechanism inference in hypothesis-driven mode. For example, HuR is known to stabilize target RNAs [220], while PUM2 is known to inhibit their translation and downregulate them [85]. For most RBPs however roles are less clear or more diverse, so we always decided based on the amount of available studies and the described main functions of the RBPs. Hyperparameter optimization was run for all models, using 500 positive and 500 negative sites separately from the training set (see P2 Table S1 for full hyperparameter, dataset, and model details). Negative sites were randomly sampled from gene regions containing CLIP-seq peak regions (i.e., positive sites), such that they did not overlap with any positive sites. GraphProt structure models were chosen over sequence models in seven cases, where the structure models showed superior performance.

To identify binding site locations, GraphProt was run in profile prediction mode. This returns position-wise scores for each given input sequence (in our case the human reference transcriptome). Scores were further averaged to smooth out the profiles and incorporate more context information into each position-wise score, by using a moving window approach. For this the new position-wise score is calculated by averaging over all scores up to 5 positions left and right to the score (i.e., windows of length 11). Binding sites were subsequently defined from the averaged position-wise score profiles as continuous regions with GraphProt model scores > 0. The highest position-wise score inside the GraphProt peak region was further taken as the peak score.

In order to make model scores between models comparable and to integrate them into the MechRNA workflow, we further calculated p-values for each peak score. For this, each model was run on 5,000 randomly selected transcripts, and the resulting position-wise scores were used to generate an empirical cumulative density function (ECDF). The ECDF was then used to calculate the p-value p_x for the peak score x with the formula $p_x = 1 - ecdf(x)$. This non-parametric approach was chosen since the GraphProt scores for most models did not show a clear unimodal distribution, preventing the use of conventional fitting procedures, such as to fit a gamma distribution on the data as done for the IntaRNA2 RNA-RNA interaction energies. In addition to the p-value assignment, we also prefiltered the predicted binding sites to include only high-confidence site predictions in the inference process. The site score thresholds we obtained by constructing a second ECDF from the position-wise scores of the positive training sites (top position-wise score for each site), and by selecting the score at 50% of the distribution. This gave us a model-specific threshold score for filtering (or p-value when inserted into the first ECDF, see Table S1), which allows us to keep sites comparable score-wise to the sites in the positive set.

3.2.3 Results and discussion

To check whether MechRNA is capable of detecting known lncRNA mechanisms, we selected 3 prostate cancer-associated lncRNAs (*7SL*, *PCAT1*, *ARlnc1*) whose mechanisms have been studied in various publications. For these we used MechRNA's hypothesis-driven mode, since there is *a priori* information available on their mechanisms. A more systematic study of MechRNA's prediction quality was not possible due to only a handful of studied mechanisms available so far. In addition, we predicted potential mechanisms for 5 mostly prostate cancer-associated lncRNAs without known mechanisms, by utilizing MechRNA's screening mode. In the following we summarize and discuss the results for *7SL* and *PCAT1*.

7SL is an ubiquitously expressed lncRNA which constitutes the RNA component of the signal recognition particle RNP complex. In addition, it was shown to be highly expressed in cancer tissues, where it promotes cancer cell growth [10]. Based on their experimental results, the authors proposed a mechanism of competitive downregulation, where *7SL* binding to the *TP53* mRNA (encoding the tumor suppressor protein p53) competes with HuR binding to *TP53*, causing inhibition of p53 translation. We therefore ran MechRNA with *7SL* in hypothesis-driven mode, using all 16 TP53 isoforms, all downregulatory mechanisms, and all RBP models. In agreement with the postulated mechanism, MechRNA predicted *"competitive downregulation"* with HuR as the most likely mechanism for all 16 TP53 isoforms. Moreover, our predicted RNA-protein and RNA-RNA interaction sites overlap with the interaction sites postulated in the paper (see P2 Table S2 for predicted interaction coordinates and [10] Figure S2B for a graphical view). It should be noted that the authors used a simple BLAST search to find *TP53* 3'UTR regions complementary to *7SL*, as well as HuR PAR-CLIP data to identify HuR binding sites in the *TP53* 3'UTR. Nevertheless, MechRNA successfully identifies the proposed mechanism, and on top suggests the most likely interaction site, whereas [10] simply reported all HuR PAR-CLIP sites and sites complementary to *7SL* in the *TP53* 3'UTR.

Similarly to *7SL*, *PCAT1* is a prostate cancer-associated lncRNA which has been shown to post-transcriptionally repress the tumor suppressor BRCA2 via an interaction with the *BRCA2* 3'UTR [221]. The authors further found that the first 250 nt of *PCAT1* are essential for this mechanism. We therefore reused the settings from *7SL* (hypothesis-driven mode, all RBPs, all downregulatory mechanisms), this time with *PCAT1* as lncRNA and the *BRCA2* 3'UTR region used in the study as target RNA. Again, MechRNA predicted *"competitive downregulation"* with HuR as the most likely mechanism. In addition, the predicted RNA-RNA interaction agreed with [221], as it is located in the first 250 nt of *PCAT1*. To test our proposed mechanism, we applied immunoprecipitation assays to check HuR binding to *BRCA2*, as well as *PCAT1*'s ability to compromise this interaction. Figure 3.5 a + b show the binding assay results. The binding assays confirm the interaction between HuR and *BRCA2* (Figure 3.5 a). Furthermore, full-length *PCAT1* inhibits HuR binding to *BRCA2*, whereas *PCAT1* with its first 250 nt deleted does not (Figure 3.5 b). MechRNA's proposed mechanism of competitive binding involving *PCAT1* and HuR thus offers a plausible explanation for the

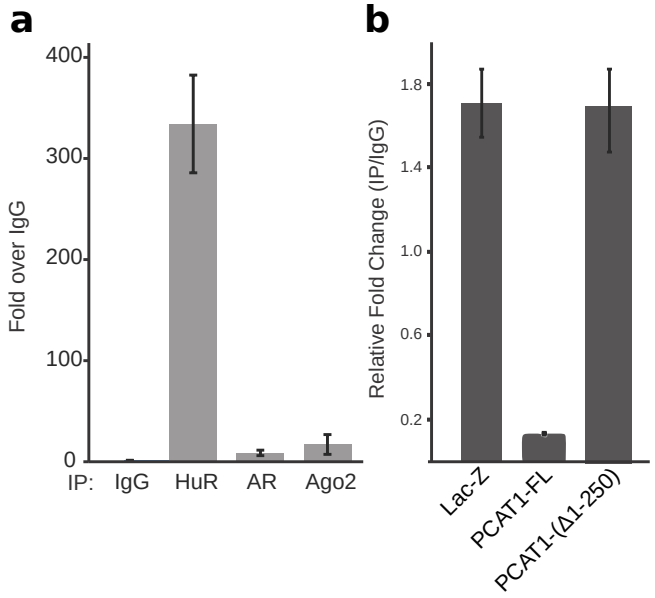


Figure 3.5: HuR binding assay results to check HuR binding to *BRCA2* mRNA, as well as *PCAT1*'s ability to disrupt the interaction. **(a)** Amount of *BRCA2* RNA pulled down by immunoprecipitation of HuR in LNCaP cells, compared to other proteins (IgG, AR, AGO2), normalized by IgG control to get fold enrichment. *BRCA2* RNA amount was measured by qPCR. **(b)** Amount of *BRCA2* RNA pulled down by immunoprecipitation of HuR in LNCaP cells, when stably expressing *Lac-Z*, *PCAT1-FL*, or *PCAT1-(Δ 1-250)*, again normalized by IgG control to get fold enrichment. *BRCA2* RNA amount was measured by qPCR. *PCAT1-FL*: full-length *PCAT1*. *PCAT1-(Δ 1-250)*: *PCAT1* with first 250 nt deleted. Figure taken from publication P2 [219].

repression of BRCA2 by *PCAT1* observed in [221].

Summing up, MechRNA is the first tool capable of inferring lncRNA mechanisms from interactions with other RNAs and proteins. Even though there exist only a few studies on such mechanisms so far, we have shown that MechRNA is able to detect known mechanisms. In addition, MechRNA can offer new clues about existing mechanisms, and suggest binding sites and hypotheses for experimental testing. In screening mode, MechRNA can further provide lists of potential targets and mechanisms for initial examination. A number of improvements could further enhance MechRNA's usability and performance: For example, MechRNA is currently restricted to reference transcriptome interactions, as it relies on precomputed RBP interaction predictions as well as correlation data. Adapting MechRNA's workflow so that users can easily compute and integrate these data by themselves would greatly increase its flexibility. As for the RNA-RNA interaction prediction part, including experimental structure probing data and constraint folding to inform structure prediction about RBP binding sites could further improve its accuracy. For the RNA-protein interaction prediction part, using more advanced models (as presented in publication P3) would likely boost performance too. In addition, microRNA binding site predictions could be integrated similarly to RBP binding site predictions.

3.3 RNAProt: an efficient and feature-rich RNA binding protein binding site predictor

This section summarizes the following publication:

- [P3] **Michael Uhl**, Van Dinh Tran, Florian Heyl, and Rolf Backofen. **RNAProt: an efficient and feature-rich RNA binding protein binding site predictor**. *GigaScience*, 2021.

3.3.1 Overview

As described in section 3.1.2 *Application examples* (also see publication P1, section 3.3), RBP binding profiles obtained by CLIP-seq offer incomplete information, since detection of RBP binding sites by CLIP-seq depends on the expression of the target RNA. Furthermore, there can be mapping or antibody specificity issues in CLIP-seq, necessitating the development of computational methods to learn RBP binding preferences from CLIP-seq data, in order to predict transcriptome-wide binding profiles. Over the years, various prediction methods have been proposed, from simple motif-based approaches to more sophisticated classical machine learning methods, including GraphProt (see P1 and P2 for application examples). Lately, deep learning methods have become the state-of-the-art in terms of predictive performance, typically implementing convolutional neural networks (CNNs) in combination with recurrent neural networks (RNNs). While these methods certainly provide improved performance over older but still popular tools like GraphProt, we encountered several issues: many are not well documented or maintained, complicating their installation and usage, or not even available. Moreover, we experienced runtime efficiency issues, as well as limited sets of options for processing datasets and supported predictive features. To counteract these problems, we implemented RNAProt, an RNA-protein interaction prediction tool based on RNNs. RNAProt offers state-of-the-art predictive performance as well as superior runtime efficiency. It is easy to install and use, facilitated by its comprehensive documentation on GitHub and the availability of a Conda package. In addition, it supports more input types and features than any other tool available so far, including user-defined features. Taken together, RNAProt provides a flexible and performant solution for large-scale RBP binding site predictions and related studies.

Publication P3 starts with describing the RNAProt method, including the utilized RNN model architecture, the general RNAProt workflow with its various program modes, and a summary of supported features. The methods are completed by descriptions on how RNAProt visualizes what was learned by the model, and how the benchmark datasets for the tool comparison were constructed. This is followed by the results and discussion section, in which RNAProt is compared to GraphProt as well as two deep learning methods, including benchmarks on predictive performance as well as runtime. Next, RNAProt visualizations are compared to known RBP binding preferences, and RNAProt's additional features are

benchmarked against baseline models using sequence information only. Finally, we present a use case on how additional features (i.e., secondary structure information) can improve the specificity of predictions for the structure-binding RBP Roquin.

3.3.2 Methods

Model architecture

RNAProt utilizes a recurrent neural network (RNN)-based model to learn RBP binding preferences from CLIP-seq or related data. Various hyperparameters can be adjusted or optimized for the dataset at hand using state-of-the-art hyperparameter optimization by Bayesian Optimization and Hyperband (BOHB) [203]. By default, the type of RNN is a Gated Recurrent Unit (GRU) [207], and the network consists of one GRU layer, followed by one fully connected layer (see P3 for full settings). All results reported in P3 were generated using the default settings. RNN-based models are known to work well with linear sequence information, since they can learn dependencies between parts of variable distance in a given sequence. As such they have been successfully applied to a range of tasks, from natural language processing over time-series data studies to the analysis of biological sequences like DNA or RNA. In our case, the input to the network is a one-hot encoded RNA sequence, optionally with additional feature channels (for details see *Supported features* section below). The output of the network is a value which can be interpreted as a score, with values > 0 corresponding to the RNA sequence being classified as positive, and values ≤ 0 to a negative classification.

RNAProt workflow

Figure 3.6 shows the RNAProt workflow, which is organized into five different program modes: training set generation (`rnaprot gt`), prediction set generation (`rnaprot gp`), model training (`rnaprot train`), model evaluation (`rnaprot eval`), and model prediction (`rnaprot predict`). Since training RNAProt models is fast, we decided to separate dataset generation from training and prediction. This way, the time-consuming extraction of additional features (e.g., conservation scores, secondary structure, region annotations) has to be done only once. In model training, the user can then select which features to take into account, allowing for a quick assessment of which features work best for the given dataset.

For dataset generation, RNAProt accepts binding sites in three different formats (or input types): sequences, genomic regions, or transcript regions (a motivation for using transcript regions is given by publications P4 and P5). While genomic regions include the genomic coordinates of binding regions, transcript regions contain their coordinates on transcripts. Various options are available for input site filtering and negative set generation (in case of training set generation). For genomic or transcript regions, RNAProt by default generates a negative training set, by randomly selecting sites which do not overlap with any input sites, from genes or transcripts covered by input sites. Negatives can be further filtered based on

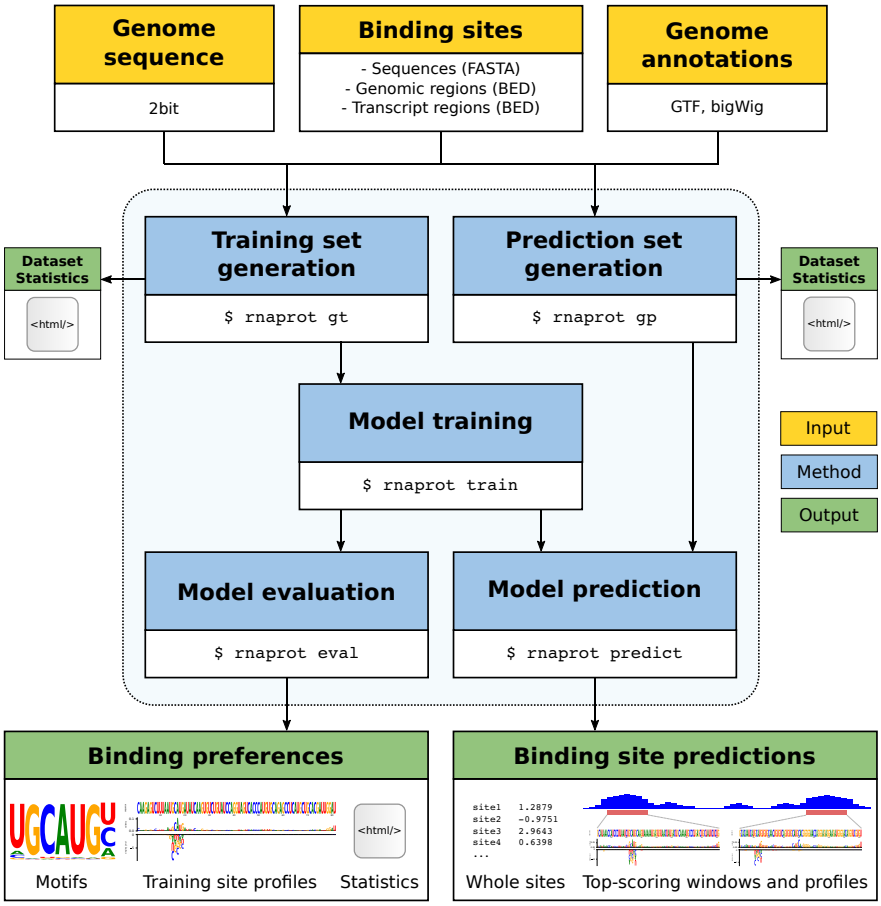


Figure 3.6: Overview of the RNAProt workflow. Yellow boxes mark mandatory inputs, blue boxes the five program modes of RNAProt, and green boxes the workflow outputs. Arrows show the dependencies between inputs, modes, and outputs. Figure taken from publication P3 [222].

their sequence complexity, in order to remove low-complexity sites (e.g., di-nucleotide repeat regions). By default, RNAProt centers and extends all training sites, i.e., to obtain sites of same length (unless not possible, e.g., at transcript ends).

To train a model, the user provides the generated training set to `rnaprot train` and specifies which features the model should include for training. Due to its use of RNNs, RNAProt natively supports training sets with variable site lengths, while popular CNN-based methods need to apply workarounds (i.e., padding techniques). In training, the user can specify various hyperparameters, and learning curves (displaying training loss versus validation loss) can be plotted. In addition, 10-fold cross-validation can be run to assess model generalization performance, as well as hyperparameter optimization using BOHB.

Once a model is trained, it can be further evaluated with respect to its learned binding preferences (evaluation mode), or utilized to predict binding sites on a generated prediction set (prediction mode). In evaluation mode, binding preferences can be visualized through sequence and additional feature logos, as well as whole-site profiles (see *Visualization of RBP binding preferences* section below). Model prediction can be run in two ways: (i) whole site prediction and (ii) top-scoring windows prediction. In the first case, each input site is

scored as a whole by RNAProt, meaning that each input site gets one score assigned. In the second case, RNAProt applies a sliding window approach, running over each input sequence to report high-scoring windows and peak regions. The amount of reported regions can be controlled by setting three different threshold levels (termed relaxed, standard, strict). The actual threshold scores behind the levels are extracted from the whole-site model scores of the positive set during model training, and thus are individually set for each model. In addition, profiles can be plotted for each reported region to visualize local binding preferences.

Supported features

In addition to the RNA sequence information, RNAProt supports several additional predictive features, which can be extracted or calculated in dataset generation and selected in model training. Table 3.1 lists the available additional features for each input type (sequences, genomic regions, transcript regions). Since conservation scores and region annotations (exon-intron, transcript, and repeat regions) require genomic or transcript coordinates of the input sites, they are not supported when only sequences are supplied. On the other hand, secondary structures can be calculated for sequences and regions (after sequence extraction). In case of regions, RNAProt automatically extends the provided genomic or transcript regions, in order to get the most accurate structure predictions (an important feature which to our knowledge is not used by any other related tool). In addition, RNAProt also supports user-defined features (from version 0.4 on also for sequences). Here, numerical or categorical features can be defined for each sequence or region position.

Table 3.1: Additional features available for each of RNAProt's three supported input types (sequences, genomic regions, transcript regions). *also available for sequences since version 0.4.

	Input		
Additional feature	Sequences	Genomic regions	Transcript regions
structure	YES	YES	YES
conservation scores	NO	YES	YES
exon-intron regions	NO	YES	NO
transcript regions	NO	YES	YES
repeat regions	NO	YES	YES
user-defined	NO*	YES	YES

Visualization of RBP binding preferences

In order to visualize RBP binding preferences learned by the model, RNAProt applies two different techniques: (i) saliency maps and (ii) *in silico* mutagenesis (see Figures 3.9 and 3.10 for visualization examples). Both approaches evaluate which positions in the input sequence contribute most to the model prediction. Saliency maps show the importance the trained model attributes to each sequence position, by visualizing the gradient with respect to the input for each sequence position [223]. In contrast, *in silico* mutagenesis works by mutating

each sequence position three times (inserting the three non-wild-type nucleotides), and each time plotting the model score differences compared to the wild type sequence. This way, the visualization shows us the effect of each possible mutation on the whole-site score, at each sequence position. *In silico* mutagenesis thus provides us with additional information (positive or negative effect), while the saliency value is always positive. On the other hand, *in silico* mutagenesis is computationally more expensive, since for a sequence of length n we need to generate $3 * n$ sequences to calculate scores for. Moreover, additional features are not mutated, limiting the observed effects on the sequence feature. For the logo generation, we therefore relied on the top saliency positions over a specified number of top-scoring sites.

3.3.3 Results and discussion

Predictive performance comparison

To evaluate RNAProt's predictive performance, we compared it with GraphProt and two deep learning approaches (DeepCLIP [224] and DeepRAM [225]). Both approaches showed superior performance compared to various other RBP binding site prediction methods in their original publications. The cross-validation comparison was conducted between GraphProt, DeepCLIP, and RNAProt, while DeepRAM and RNAProt were compared in a hold-hout validation setting. This was done because DeepRAM did not offer a cross-validation option, and because its hyperparameter optimization could not be disabled, which would further have resulted in unfeasible cross-validation runtimes. Both comparisons utilized the same benchmark data, consisting of two sets of CLIP-seq data: (i) a set of 23 CLIP-seq datasets from various protocols and 20 different RBPs; and (ii) a set of 30 eCLIP datasets from 30 different RBPs.

For the cross-validation comparison, all three methods were run using their default hyperparameters. Figure 3.7 a and 3.7 b show the 10-fold cross-validation results for GraphProt, DeepCLIP, and RNAProt. RNAProt produces the highest total average AUC on both benchmark sets (87.26% and 89.30%), followed by DeepCLIP and GraphProt. To signify the differences, we calculated two-sided Wilcoxon tests on the AUC distributions for each method combination and each of the 53 datasets. Figure 3.7 c and 3.7 d contrast the single dataset AUCs of GraphProt with RNAProt and DeepCLIP with RNAProt, using the method colors (GraphProt: red, DeepCLIP: yellow, RNAProt: blue) to highlight significantly better AUCs (gray: no significant difference). We can see that RNAProt performs significantly better in 49 (versus GraphProt) and 42 (versus DeepCLIP) out of 53 cases, while GraphProt and DeepCLIP both score higher in only two cases. Out of the box, DeepCLIP's CNN+RNN architecture thus performs worse than RNAProt's RNN-only architecture. In addition, RNAProt is superior with regard to its runtime (see *Runtime comparison* below).

For the comparison with DeepRAM, we chose DeepRAM's best performing network architecture, featuring a Word2Vec embedding, a CNN layer, and a bidirectonal Long Short-Term Memory (LSTM) layer. We used a hold-out setting (i.e., one train-test split) as motivated,

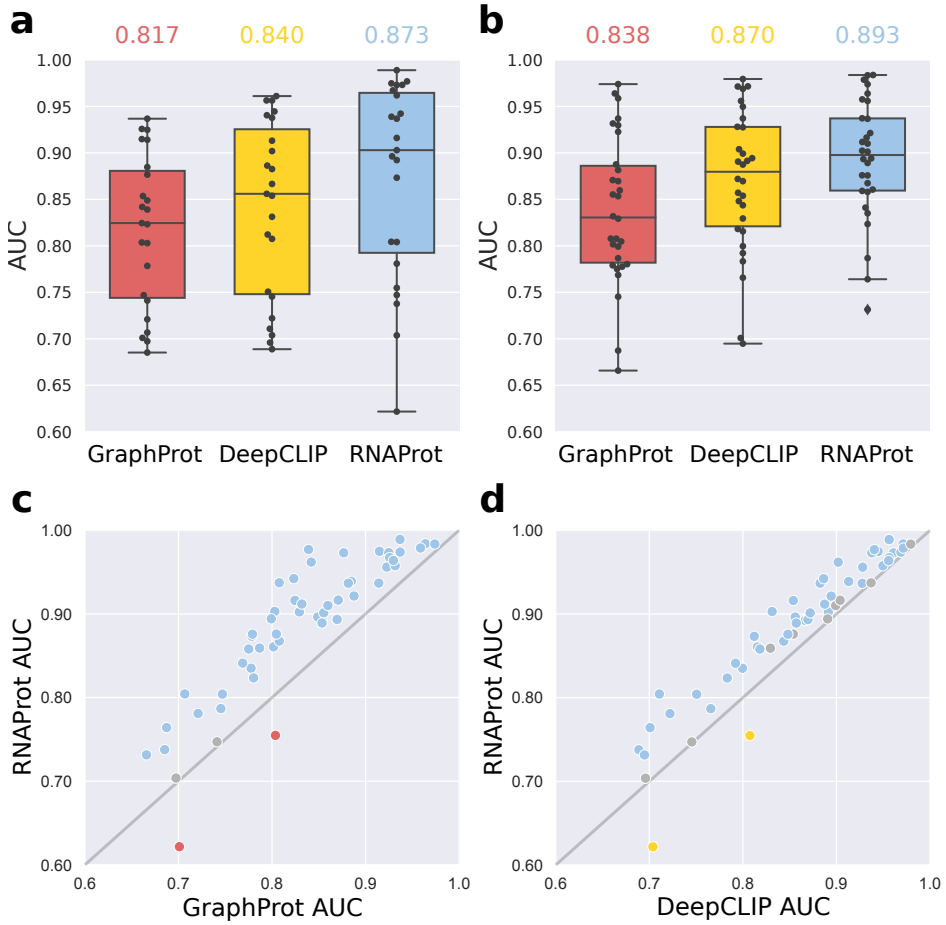


Figure 3.7: 10-fold cross-validation results for GraphProt, DeepCLIP, and RNAProt. **(a)** Results for the first benchmark set (23 CLIP-seq datasets). **(b)** Results for the second benchmark set (30 eCLIP datasets). **(c)** Comparing all 53 single dataset AUCs between GraphProt and RNAProt. **(d)** Comparing all 53 single dataset AUCs between DeepCLIP and RNAProt. Blue dots indicate a significantly better AUC for RNAProt (49, 42), gray dots no significant difference (2, 9), red and yellow dots a significantly better AUC for GraphProt or DeepCLIP (2, 2). Two-sided Wilcoxon test was used to calculate p-values (significance threshold = 0.05). Figure taken from publication P3 [222].

and reduced the number of random search iterations for its hyperparameter optimization from 40 to 20, to make the comparison more fair. DeepRAM runtimes remained an issue though, with model training on a typical dataset still taking $\sim$ 5-6 h versus 1-2 min for the RNAProt model. As shown in publication P3 Figure 3, both methods perform very similar to each other (average AUCs: DeepRAM 87.42% and 89.28%, RNAProt 87.50% and 89.34%), even though DeepRAM could have a slight advantage due to its active hyperparameter optimization. We can thus conclude that a more complex architecture like the one applied by DeepRAM is not necessary to achieve top predictive performances on the datasets at hand. As further discussed in P3, this could change for larger datasets. On the other hand, the used benchmark datasets feature typical set sizes, similar to or even larger than the typical set sizes obtained from ENCODE. Since different methods can learn different features of a dataset, it would be interesting to closer examine ensemble predictions in future studies.

Runtime comparison

Model training is known to be the computationally most expensive part in deep learning. We therefore compared training runtimes for the three methods GraphProt, DeepCLIP, and RNAProt. As shown in Figure 3.8, RNAProt's training time is comparable to GraphProt's (72 sec versus 40.3 sec), and 31 times faster than DeepCLIP's (72 sec versus 37 min). Even in CPU mode (i.e., disabling GPU support), RNAProt achieves a speedup of 4.7x. This clearly demonstrates RNAProt's ability for on-the-fly model training, which also enables quick testing of hypotheses regarding dataset, parameter, or predictive feature choices. In addition, it shows the benefits of using a GPU (even an average consumer-grade GPU as applied in this study).

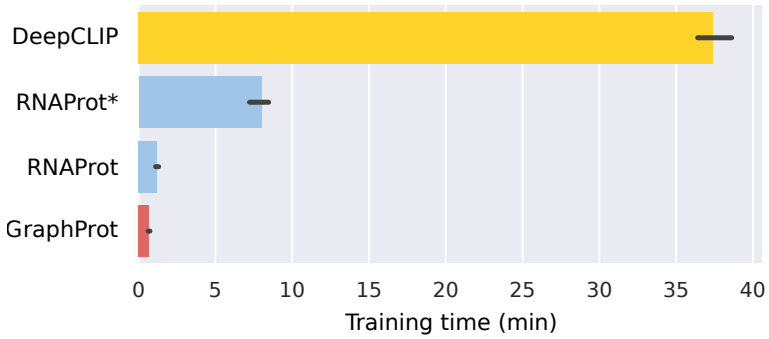


Figure 3.8: Model training runtime comparison. Training times in minutes (averaged over three runs) for training a single model with 10,000 instances for GraphProt, RNAProt, and DeepCLIP. *RNAProt using CPU for calculations (no GPU). Figure taken from publication P3 [222].

RNAProt captures known RBP binding preferences

Deep neural networks by design are complex and hard to interpret, necessitating the use of visualization methods that help to explain what was learned by the model. To check whether RNAProt models capture known RBP binding preferences, we compared preferences of 6 RBPs to the outputs of RNAProt's two visualization methods (i.e., saliency maps, *in silico* mutagenesis). As we can see in Figure 3.9, RNAProt sequence logos and profiles clearly capture known RBP binding preferences. While the logos provide a compact view on local preferences, the site profiles give further clues about the scatteredness and the importance of individual motifs. Moreover, the mutation tracks inform about which mutations have the strongest influence on the model score. In the future, the integration of global model interpretability methods should give further insight into what the model has learned from the data [198].

Additional features to improve predictions

As described, RNAProt offers various additional predictive features which can be included in model training. We thus first checked their effects on predictive performance for all 53 CLIP-seq datasets. P3 Figure 6 shows that additional features can strongly boost the overall model

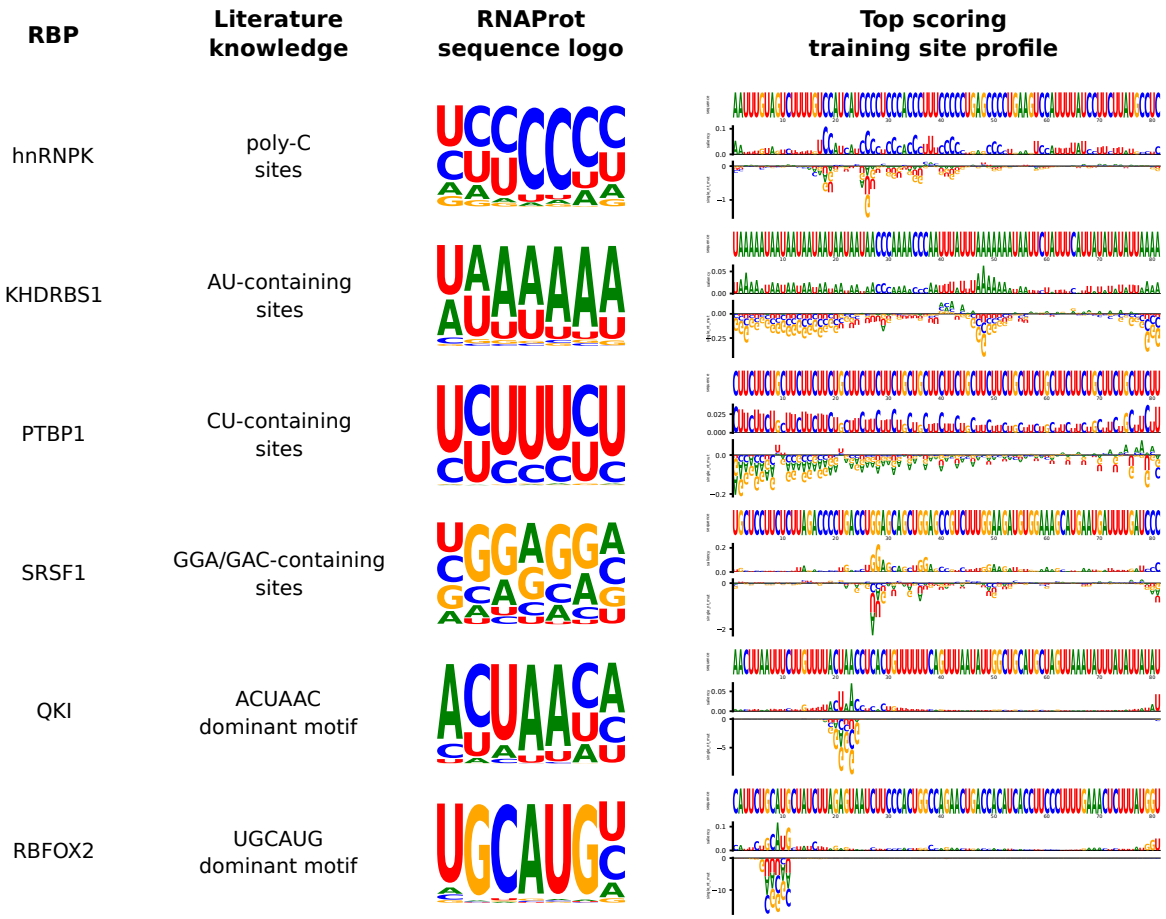


Figure 3.9: Comparison of RNAProt sequence logos and profiles with known RBP binding preferences. Literature knowledge was obtained from ATtRACT [226]. Sequence logos were generated from top site saliency positions (top 200 scoring training sites), with character heights corresponding to their respective saliency values at each position. Training site profiles on the right offer several tracks: site nucleotide sequence (top), position-wise saliency (middle), and single mutation effects (bottom). Figure taken from publication P3 [222].

performance, i.e., when using conservation scores and exon-intron annotations. Secondary structure information did not boost the overall performance, but this also depends on the dataset and probably can be tuned by adapting structure calculation parameters. Regarding the region type and conservation features, the resulting performances of course strongly depend on the selected negative regions. A large imbalance of intronic and exonic regions in the positive and negative regions will naturally boost predictive performances. The focus of the prediction thus becomes more important: a natural preference of an RBP for exon or intron regions could be exploited when predicting on whole gene regions, while training and predicting on transcript regions (i.e., containing only exons) would make exon-intron annotations meaningless. In this regard, RNAProt offers several options to fine-tune negative site selection, e.g., by defining genomic regions which should be excluded from negative site selection.

To check whether additional features can be useful in a more defined use case, we also

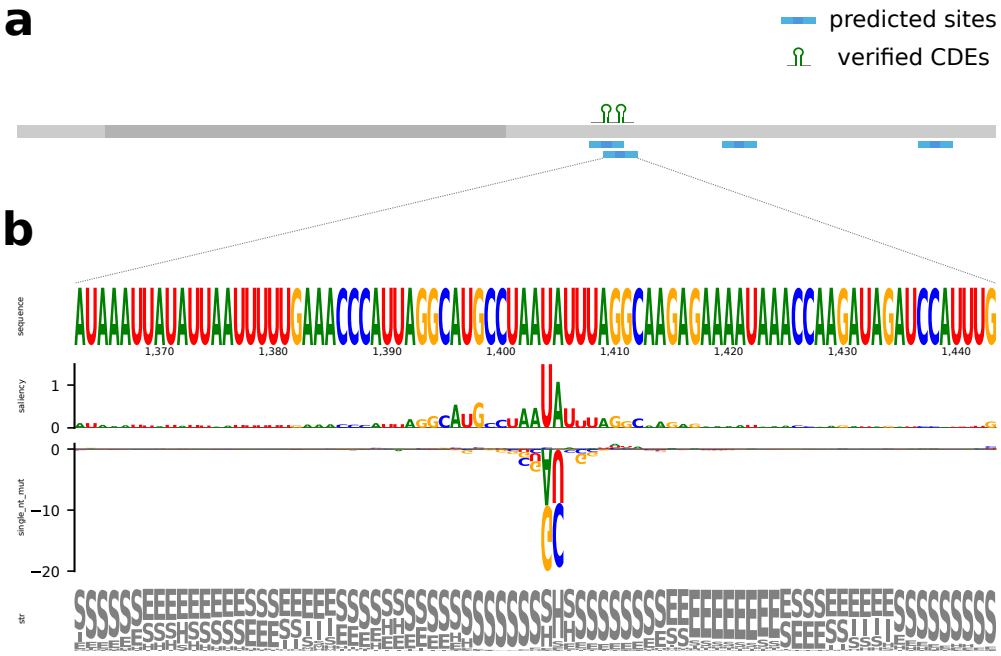


Figure 3.10: Roquin structure model predictions on the UCP3 gene transcript ENST00000314032.9. **(a)** ENST00000314032.9 transcript (length 2,277 nt, 5'UTR (light gray), CDS (dark gray), 3'UTR (light gray)) together with verified and predicted Roquin binding sites (CDEs). **(b)** RNAProt site profile for the second verified CDE. Shown profile tracks from top to bottom: sequence, saliency map, *in silico* mutation track, structural elements track. Figure taken from publication P3 [222].

trained models on a set of predicted structurally conserved binding sites of the RBP Roquin (also termed constitutive decay elements (CDEs)) [138]. A CDE consists of a stem loop, with variable stem sequences and a more conserved trinucleotide loop sequence. 10-fold cross-validation results produced an average AUC for the sequence model of 79.22%, while the structure model performs almost 20% better (99.02%). As a reference, the GraphProt structure model achieved an average AUC of 78.49%. This shows that additional structure information can be detected and utilized by RNAProt, in order to obtain a much improved predictive performance. The authors of [138] also verified 2 CDEs in the 3'UTR of the UCP3 gene (ENSEMBL transcript ID ENST00000314032.9), which we further used to assess the specificity of the model. We thus retrained the structure model leaving out the 2 CDEs, and ran a sliding window prediction on the whole transcript. Figure 3.10 a shows the predicted site and verified CDE locations. As we can see, both verified sites are predicted by the model, plus two additional sites in the 3'UTR. To visualize the learned model preferences, Figure 3.10 b shows the RNAProt profile of the higher-scoring second site. As presumed, the saliency track highlights the loop nucleotides, and to a lesser extent the stem parts, while the mutations track focuses on the loop. Moreover, the stem loop can be recognized in the structural elements track. As a comparison, the sequence model predicted 18 sites on the transcript, including only one verified CDE, making the structure model predictions much more specific (F-score sequence model = 0.10; F-score structure model = 0.67).

3.4 Improving CLIP-seq data analysis by incorporating transcript information

This section sums up the contents of the following publication:

- [P4] **Michael Uhl**, Van Dinh Tran, and Rolf Backofen. **Improving CLIP-seq data analysis by incorporating transcript information**. *BMC Genomics*, 2020.

3.4.1 Overview

CLIP-seq enables the transcriptome-wide identification of RBP binding sites by producing a library of reads bound by the target RBP. The precise binding locations are subsequently identified from the mapped read profiles by tools termed peak callers. All currently available peak callers identify binding sites by analysing the genomic read profiles, effectively ignoring the underlying transcript information (i.e., information on splicing events and transcript structure). Intuitively, this is far from optimal, especially for RBPs that predominantly bind to spliced RNA. We therefore decided to closer examine this issue and assess its significance, as there were no studies available on the topic.

Publication P4 begins with an example illustrating the problems current peak callers have with CLIP-seq data from predominantly exon-binding RBPs. Next, the extent of exon and exon border binding in a large collection of 223 eCLIP datasets is quantified, which turns out to be substantial. Based on this finding, we further verified that different sequence contexts (i.e., transcript versus genomic context) influence predictive performances of binding site prediction tools. Moreover, we found out that RBP binding motifs of exon-binding RBPs are more frequent in transcript context surrounding the sites compared to genomic context. Finally, we discuss possible strategies to improve CLIP-seq data analysis workflows by integrating transcript information.

3.4.2 Results and discussion

Ignoring transcript information compromises peak calling

To showcase the problems current peak callers have with predominantly exon-binding RBPs, we chose one out of the many RBP eCLIP datasets (RBP: YBX3, cell line: K562) with high amounts of exonic binding regions from the ENCODE eCLIP dataset collection (see P4 Table S1 for full dataset statistics). Figure 3.11 shows two genomic regions with mapped YBX3 eCLIP data, together with the peak regions called by CLIPper, CLIPper IDR (reporting only high-confidence CLIPper peaks reproducible between replicates), PEAKachu, and PureCLIP. As we can see, the read profiles clearly follow the exon region annotations, with high amounts of reads mapping to exons and many intron-spanning reads (petrol blue lines). This shows that the RBP actually binds to the spliced RNA (which we refer to as transcript context), and not to the unspliced RNA (which we refer to as genomic context). As expected, all

reported peak regions are located in exons. However, extending these peak regions with genomic context as routinely done in CLIP-seq data analysis (e.g., for motif search, secondary structure prediction, or predictive model learning) is obviously wrong. Instead, the peak regions should be extended with their transcript context.

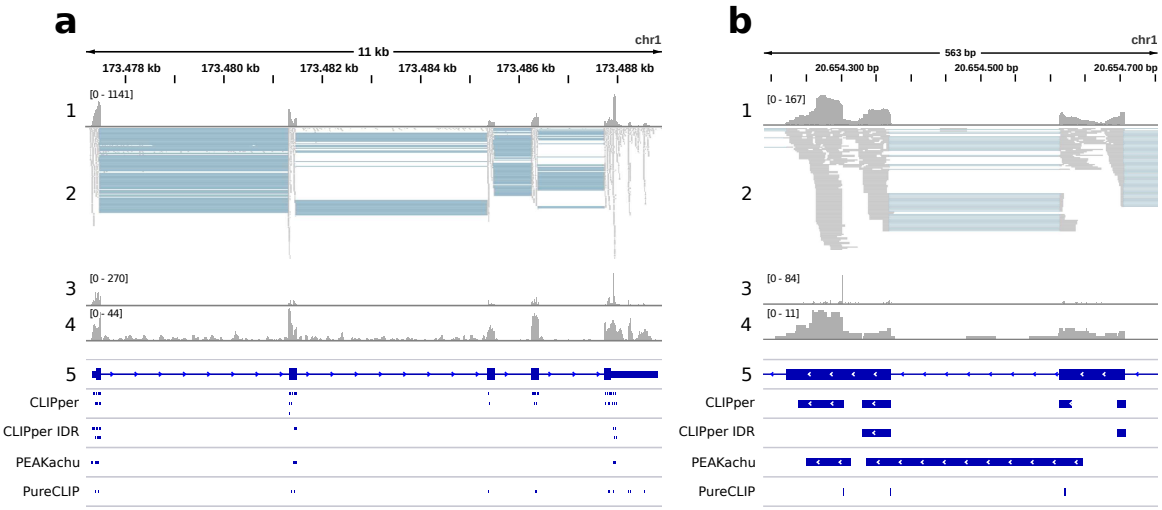


Figure 3.11: Two genomic regions with mapped YBX3 eCLIP reads. 1: read profile (coverage range in brackets), 2: read alignments, 3: crosslink positions profile, 4: input control profile, 5: gene annotations (thick blue regions are exons, thin blue regions introns), and peak regions called by CLIPper, CLIPper IDR, PEAKachu, and PureCLIP. **(a)** *PRDX6* whole gene region (length 11 kb, maximum read coverage 1,141). **(b)** *DDOST* gene exons 6 and 7 region (length 562 bp, maximum read coverage 167). Figure taken from publication P4 [227].

Looking closer at two neighboring exon regions, Figure 3.11 b brings up a second problem: peak regions in a transcript context called at two adjacent exon borders arguably should be merged, instead of being interpreted as two separate binding events. However, none of the three peak callers are aware of this, as they work directly and only on the genomic read profiles. And even if there is only one peak called, the first problem (i.e., context choice) still remains. As described in section 3.1.2 *Peak caller comparison*, peak callers can be categorized into methods that take into account the full read profile (e.g., CLIPper, PEAKachu) and methods that focus on single positions or read starts (e.g., PureCLIP) to identify binding locations. Split exon border sites should thus be more prevalent in the first group, since these methods usually treat the mapped parts of intron-spanning reads as separate reads. In contrast, methods like PureCLIP count each read only once (i.e., at the read start position), which should reduce the calling of split peaks. In addition, there is also a PEAKachu-specific problem when dealing with exon-binding RBP read profiles: since PEAKachu was originally designed for bacterial CLIP-seq data, the existence of introns was never considered. More precisely, PEAKachu replaces each read with a Gaussian, and takes the genomic center of the read as the mean of the Gaussian. This explains why PEAKachu in our example (3.11 b) falsely called a peak over the entire intron.

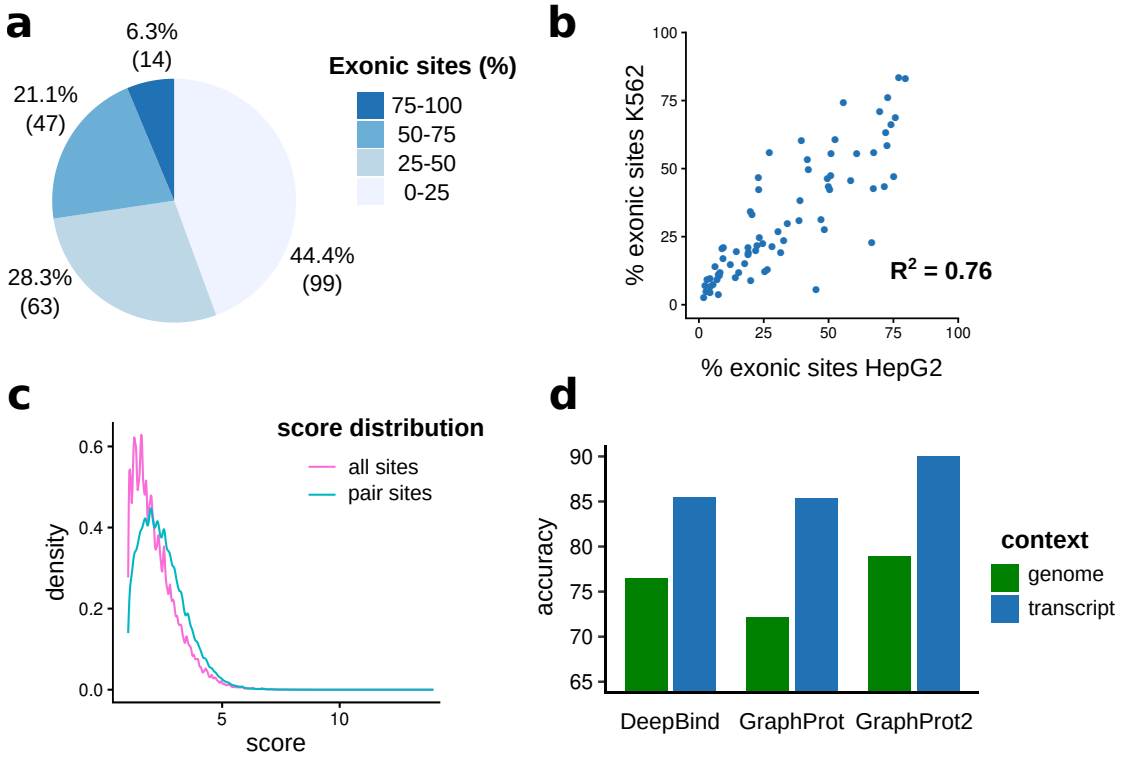


Figure 3.12: Exon binding statistics of eCLIP datasets and prediction results for different sequence contexts. **(a)** Distribution of exonic site ratios for 223 eCLIP datasets over four percentage ranges. For each range, the percentage (number) of sets within this range is given. **(b)** Correlation plot of exonic site ratios for RBPs present in two cell lines (HepG2, K562). **(c)** Site score distributions for all exonic sites and exonic sites that form pairs at adjacent exon borders. CLIPper log_2 fold change values were taken as site scores. Only pair sites with an exon border distance of < 10 nt were considered. **(d)** Average classification accuracies over 6 eCLIP datasets for 3 RBP binding site prediction methods, comparing genome and transcript context. Figure taken from publication P4 [227].

Exon binding is substantial in public CLIP-seq data

We next determined the extent of exon and near exon border binding in a collection of 223 eCLIP datasets, covering 150 different RBPs in two cell lines (HepG2, K562). As shown in Figure 3.12 a, exonic site percentages of $\geq 50\%$ are observed for 61 datasets (27.4%), out of which 14 datasets reach percentages of $\geq 75\%$ (see P4 Table S1 for full dataset statistics). Looking closer at the 61 datasets, 63.3% of exonic sites are located within ≤ 50 nt to exon borders, and 20.7% form pair sites (i.e., sites at adjacent exon borders). The datasets thus contain a substantial amount of sites susceptible to split peak calling and false context choice. We also compared exonic site percentages across the cell lines (Figure 3.12 b), showing a general agreement ($R^2 = 0.76$). On the other hand, the deviations also indicate that strict RBP classifications do not work, and that site context should ideally be determined individually for each site based on the mapped read data. One might ask whether pair sites on average feature lower scores (i.e., log_2 fold changes), but this is not

the case, as shown in Figure 3.12 c (mean score for pair sites 2.47 versus 2.17 for all exonic sites).

Sequence context influences model performances

To check whether context choice also influences the performance of RBP binding site prediction tools, we randomly chose 6 eCLIP datasets from 6 different RBPs with exonic site percentages $\geq 80\%$, keeping only sites ≤ 10 nt from exon borders. Sites were centered and extended either with transcript or genomic context (uniform length of 161 nt), to obtain two training sets for each RBP (i.e., transcript versus genomic context set). Figure 3.12 d shows the average accuracies obtained for three binding site prediction tools, both for the transcript and genomic context sets. As we can see, the transcript context models of all methods perform considerably higher, meaning that context choice influences predictive performances. To exclude the possibility that the models primarily learned RBP-unspecific context information, we also looked at GraphProt's sequence logos (P4 Figure S1). These provide a simplified view on what sequence information the models regard as most important, and we observed a general agreement between logos and known RBP binding preferences in both contexts. This shows that the models do not primarily detect generic context information, but rather a mixture of site-specific and context information. Depending on the prediction task, including the authentic context could thus be favorable and make models more specific.

Known motifs are enriched in transcript context

Given the influence of context choice on predictive performance, we also looked at motif enrichment for known RBP motifs in sites with added transcript context versus the same sites with genomic context. CLIPper IDR peak regions for 9 RBPs with known motifs (in total 28 motifs) and increased percentages of exonic sites (40.23 to 84.06%) were filtered to keep only sites near exon borders, which were again extended with transcript and genomic context. Table S2 summarizes the motif search results, showing that out of the 23 motifs with > 10 hits, 20 are 10 - 57% more frequent in transcript context sets, while the remaining 3 motifs showed minor frequency changes ($< 3\%$). Moreover, the five motifs with < 10 hits were all enriched from 35 to 709%. While a certain transcript context motif enrichment is expected for exon-binding RBPs, more well-defined motifs showed clear enrichment in the transcript context set as well (PUM2 107 vs. 89 hits, IGF2BP3 7 hits vs. 1). This indicates that more authentic sites were recovered by adding the transcript context. As an example, Figure 3.13 shows two genomic regions with mapped IGF2BP3 and PUM2 eCLIP data, where the known binding motifs (IGF2BP3: GGC-$N_{15\text{-}25}$-CA-$N_{7\text{-}20}$-CA-$N_{15\text{-}25}$-GGC-$N_{2\text{-}8}$-$[CA]_4$, PUM2: UGUANAUA, N: any nucleotide) are split by the exon border. In addition, both CLIPper and PureCLIP falsely call split peak regions.

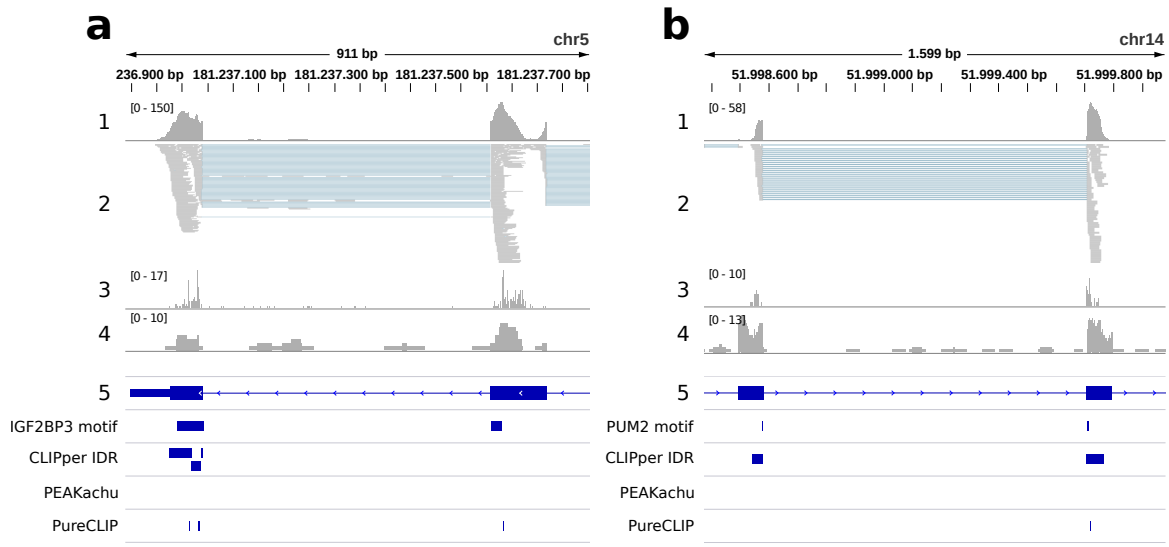


Figure 3.13: Two genomic regions with mapped IGF2BP3 and PUM2 eCLIP reads and split binding motifs. 1: read profile (coverage range in brackets), 2: read alignments, 3: crosslink positions profile, 4: input control profile, 5: gene annotations (thick blue regions are exons, thin blue regions introns), split IGF2BP3 / PUM2 motif locations, and peaks called by CLIPper, CLIPper IDR, PEAKachu, and PureCLIP. **(a)** *RACK1* gene exons 7 and 8 region (length 911 bp, maximum read coverage 150) with split IGF2BP3 motif. **(b)** *RTRAF* gene exons 4 and 5 region (length 1,599 bp, maximum read coverage 58) with split PUM2 motif. Figure taken from publication P4 [227].

Strategies to improve CLIP-seq data analysis workflows

In this study we used a relatively simple approach to select representative transcripts for each gene to extract the transcript context from. This of course can be further improved, ideally leading to approaches which select the most likely context based on the available read data, individually for each peak region. Context selection on a site level is important, as RBPs often have diverse functions, e.g., in the nucleus or the cytoplasm, which likely also result in binding to different contexts. Here we used reference annotations to define the transcript context. However, it is known that these do not cover the full transcript diversity present in a specific cell type or condition. It would thus make sense to substitute or refine the reference annotations with *de novo* transcript assemblies, ideally constructed from RNA-seq data of the same cell type or condition. As for the influence of context choice on recovering complete binding sites, we expect this to be especially important for RBPs with multiple RNA-binding domains, as exemplified by IGF2BP3 and its extended binding motif. While combinatorial binding studies are still scarce, most RBPs in fact are comprised of multiple RNA-binding domains [228]. CLIP-seq studies combined with a proper context selection could thus be of great help to determine the binding modes of these RBPs.

3.5 Peakhood: individual site context extraction for CLIP-seq peak regions

This section summarizes the contents of the following publication:

- [P5] **Michael Uhl**, Dominik Rabsch, Florian Eggenhofer, and Rolf Backofen. **Peakhood: individual site context extraction for CLIP-seq peak regions**. *Bioinformatics*, 2021.

3.5.1 Overview

In publication P4 we demonstrated the importance of including transcript information, i.e., information on transcript structure and splicing events, to improve CLIP-seq data analysis. P4 also introduced the terms genomic and transcript context, to denote RBP binding to an unspliced or spliced RNA context. As described, all currently available peak callers only take into account the genomic context for determining peak regions, effectively ignoring the underlying transcript information. While this can be acceptable for RBPs binding to unspliced RNA, P4 showcased that peak calling is compromised for spliced-RNA-binding RBPs. Moreover, selecting the more likely transcript context showed an enrichment of known binding motifs, underlining the importance of a proper context selection. We therefore decided to implement a tool which can assign the most likely context to a given set of peak regions, based on the original CLIP-seq read data and individually for each region. This tool is called Peakhood and is presented in publication P5.

Publication P5 briefly describes the concept behind Peakhood, with a more detailed method description found in the supplement, and a full documentation on GitHub. Peakhood offers several program modes, including site context extraction for a single set of peak regions, as well as multiple sets. Moreover, the extracted transcript context sites of multiple sets can be merged into comprehensive transcript context site collections. We further show that Peakhood's site context extraction results agree with known RBP roles. As a supplement, P5 provides several precomputed transcript context site collections for 49 RBPs with known roles in post-transcriptional gene regulation, from two different cell lines (HepG2, K562).

3.5.2 Methods

Approach

Peakhood's idea is to take peak regions determined by current peak callers, together with the original CLIP-seq read data and genomic annotations, in order to determine the most likely context individually for each input peak region. Peakhood therefore can be applied as a post-processing step after peak calling inside a CLIP-seq data analysis pipeline, or to reanalyze any of the millions of publicly available CLIP-seq peak regions identified by various peak callers. Context selection needs to be performed at the site level because: (i) the RBP

might have different roles in the cell, and thus bind to different contexts; (ii) in cases of inconclusive read information the tool should rather vote for a genomic context. The second point is important to assure that there is a certain amount of evidence for the selected transcript context, independent of whether the RBP has a tendency to bind to spliced RNA or not. Peakhood therefore uses rather conservative default settings for transcript context assignment.

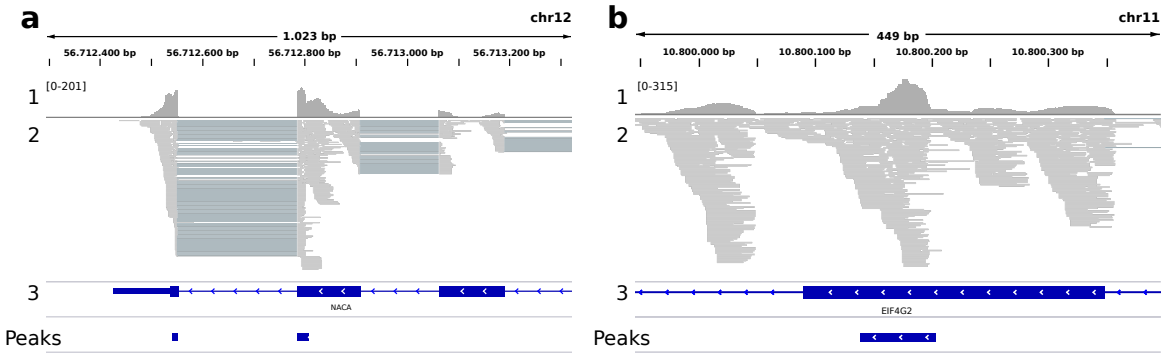


Figure 3.14: Two genomic regions with mapped PUM2 and U2AF2 eCLIP reads. 1: read profile (coverage range in brackets), 2: read alignments, 3: gene annotations (thick blue regions are exons, thin blue regions introns), and peaks called by CLIPper IDR. **(a)** Example region for the predominantly spliced-RNA-binding RBP PUM2, where an exon border site is falsely split in two peaks. **(b)** Example region for the splicing factor U2AF2, with higher read counts over exon borders and introns. Figures taken from publication P5 [229].

As for the concept behind Peakhood's site-level context assignment, consider the exonic peak regions of two RBPs: the spliced-RNA-binding translational repressor PUM2 (Figure 3.14 a), and the unspliced-RNA-binding splicing factor U2AF2 (Figure 3.14 b). The read profile in Figure 3.14 a shows a typical transcript context region, featuring considerable coverage drops from exons to introns, and high amounts of intron-spanning reads. In contrast, the read profile in Figure 3.14 b suggests a genomic context because of the higher intron coverage, as well as the presence of reads that cover the exon-intron border. When selecting the context of an exonic peak region, Peakhood therefore essentially looks at coverage differences between potential transcript exons and introns, as well as coverage drops at the exon borders. In addition, it merges exon border sites connected by intron-spanning reads, as these very likely constitute single peak regions (for example in Figure 3.14 a). The following sections provide more details on Peakhood's workflow, its site context extraction, choosing the most likely transcript, and the generation of transcript context site collections (i.e., the merging of transcript context datasets).

Peakhood workflow

Figure 3.15 shows the Peakhood workflow, comprising the two main program modes `peakhood extract` and `peakhood merge` and their connection. In site context extraction (`peakhood extract`), Peakhood uses the input peak regions, CLIP-seq read data, and exon annotations

to extract the most likely context for each input site (see *Site context extraction* section for more details). Given an exonic peak region assigned to transcript context, `peakhood extract` further determines the most likely transcript (see *Choosing the most likely transcript* section for more details). Next, `peakhood merge` can be used to merge different transcript context sets to obtain comprehensive transcript context site collections (see *Merging transcript context sets* section for more details).

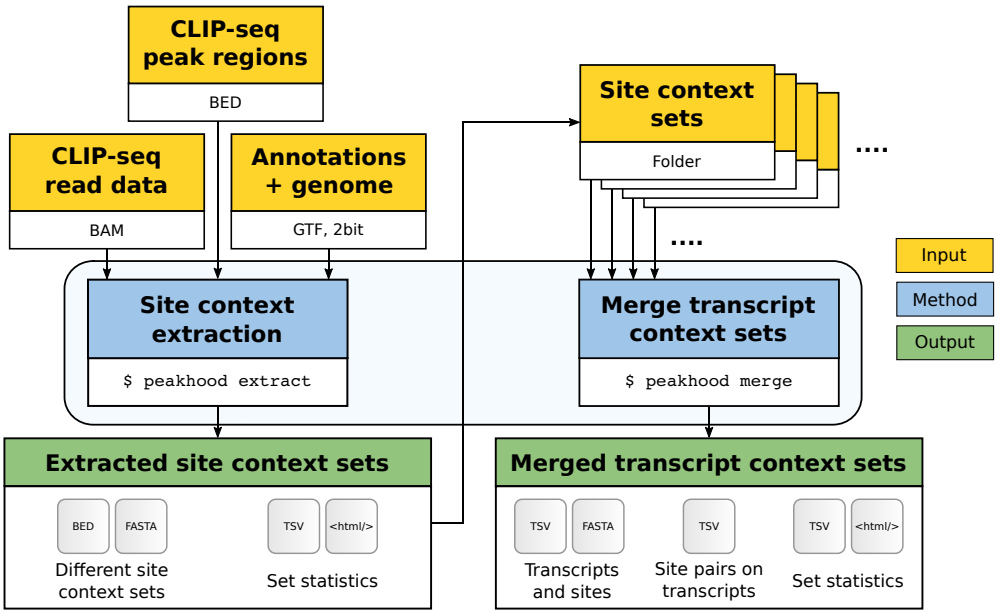


Figure 3.15: Overview of the Peakhood workflow for the two main program modes `peakhood extract` and `peakhood merge`. Yellow boxes mark necessary inputs, blue boxes the two program modes, and green boxes the outputs. Arrows show the dependencies between inputs, modes, and outputs. Figure taken from publication P5 [229].

Site context extraction

For the individual site context extraction, Peakhood's input consists of the genomic CLIP-seq peak regions, the mapped CLIP-seq reads, a genomic annotations file containing transcript and exon annotations, and a genome sequence file. Peakhood first intersects the peak regions with transcript and exon regions extracted from the annotations file, to obtain exonic, intronic, and intergenic sites. In the next step, the most likely context (i.e., genomic or transcript) is determined for each exonic site. For this Peakhood utilizes the exon-intron and exon-intron border region coverage ratios in the site neighborhood, as well as over the whole transcript. Here we make use of the observation that an exonic site inside a transcript context (for example Figure 3.14 a) usually features considerably more reads mapping to the transcript exons, as well as a sharp drop in coverage at the exon borders. Ideally we can observe this both locally (i.e., around the overlapping exon) and globally (i.e., on the whole transcript). However, due to how the CLIP-seq protocol works, read coverage is frequently limited to the peak region and its neighborhood, so Peakhood weighs the local context infor-

mation higher than the global one. In addition, intron-spanning reads are weighted higher than continuously mapped reads, as they provide strong support for a transcript context. Sites with sufficiently high local and global ratios (see Peakhood's online manual for full details on filter steps and default thresholds) get assigned to transcript context. Exonic sites with lower ratios (for example Figure 3.14 b) get assigned to genomic context.

Choosing the most likely transcript

An assigned transcript context site can have several possible site-transcript combinations, because there usually is > 1 transcript isoform for a gene, and several exons and transcripts can overlap the site and pass the filters. Peakhood therefore also determines the most likely combination, based on a number of informative filters: co-occurrence of other sites on the same transcript, read coverage, intron-spanning read numbers, and transcript support level. A combination score is assigned to each site-transcript combination, informing about the support level of a combination, and to make them comparable. Settings for filter order, choice of filters, and filter behavior (serial filtering or majority vote) can be specified. In addition, sites at exon borders connected by intron-spanning reads (for example in Figure 3.14 a) are merged into single sites. Peakhood further supports custom annotation files, and we recommend to use these if there is RNA-seq data available for the cell type or condition at hand. Peakhood also accepts RNA-seq data input to extract additional intron-spanning read information for transcript selection.

Merging transcript context sets

Transcript context sets extracted from single CLIP-seq datasets can be further merged by Peakhood into transcript context site collections. The resulting output table files include information on transcripts and their overlapping sites, both for all possible and the most likely site-transcript combinations. Moreover, site pairs on transcripts together with their transcript and original genomic distances are reported. This for example allows us to quickly filter for and spot interesting site pairs (for the same or two different RBPs), where the transcript site distance is lower than the original genomic distance, or in general lower than some desired value.

3.5.3 Results and discussion

Agreement with known RBP roles

To check Peakhood's agreement with known RBP roles, we selected three typical spliced-RNA-binding RBPs (IGF2BP1, PUM1, PUM2), as well as the splicing factor U2AF2, and ran site context extraction on their eCLIP datasets with default parameters. Among other statistics, Peakhood reports three informative percentages when performing the site context extraction: the percentage of exonic sites (divided by all sites), the percentage of extracted transcript context sites (divided by all exonic sites), and the percentage of exon border sites

(divided by all transcript context sites). Figure 3.16 shows these three percentages obtained for each of the four datasets. We can see that for the spliced-RNA-binding RBPs, most sites overlap with exons (>= 95%), and out of these >= 95% are assigned to transcript context. This is in contrast to U2AF2, where we get around 20% of exonic sites, and out of these only 5.9% are assigned to transcript context. We can thus see a general agreement between Peakhood's site context selection and known RBP roles. Interestingly, exon border site percentages can be quite substantial, reaching almost 25% for PUM1. Even higher percentages (up to 39%) were observed in the supplementary transcript context site collections. These numbers again underline the importance of a proper site context selection, including the merging of split peaks connected by intron-spanning reads.

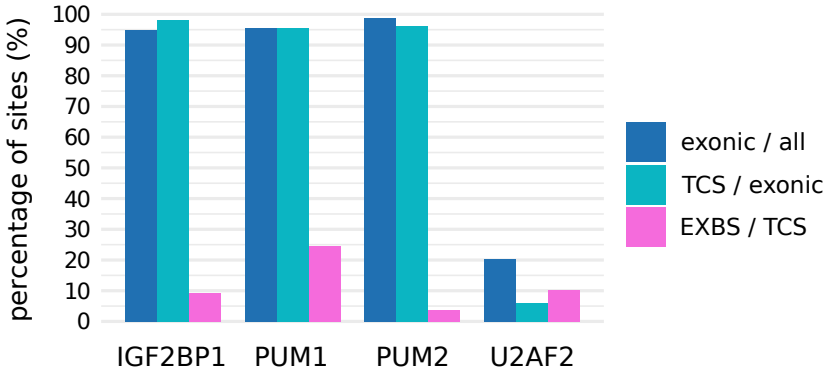


Figure 3.16: Peakhood site context extraction results for four eCLIP datasets and four RBPs (number of all sites in brackets): IGF2BP1 (4,776), PUM1 (2,146), PUM2 (4,578), and U2AF2 (3,250). The plot shows percentages of exonic sites (exonic sites divided by all sites), transcript context sites (TCS) (TCS divided by all exonic sites), and exon border sites connected by intron-spanning reads (EXBS) (EXBS divided by TCS). Figure taken from publication P5 [229].

Conclusion

Publication P5 presented Peakhood, the first tool capable of selecting the most likely context for a set of CLIP-seq peak regions, based on the available CLIP-seq read data and individually for each input peak region. As shown, Peakhood's context extraction agrees with known RBP roles. In addition, Peakhood offers various command-line parameters for the filtering and selection of sites, allowing for adaptations to different input data. As the current default parameters were chosen manually, it would make sense to further optimize them using some objective function. For example, an objective could be formulated based on the maximization and minimization of site percentages over several datasets (i.e., maximized for spliced-RNA-binding RBPs, minimized for unspliced-RNA-binding RBPs). The optimization could then also be repeated for different CLIP-seq data (i.e., from different protocols) and peak callers, in order to obtain a number of default settings, each optimized for a specific combination of input data.

Chapter 4

Conclusion and outlook

4.1 Conclusion

This thesis presented three novel computational methods (publications P2, P3, and P5), as well as a research and a review article (publications P1 and P4) on CLIP-seq data analysis, all aimed at improving the computational analysis and prediction of RNA-protein interactions.

Publication P1 reviewed the principal CLIP-seq data analysis steps, to provide readers with a practical and informative guide on how to analyse CLIP-seq data. As the identification of the precise RBP binding locations (i.e., peak calling) is arguably the most critical analysis step, we also conducted a peak caller comparison, showcasing the pros and cons of current peak callers. The summary of P1 (Section 3.1) further contains application examples related to the analysis of RNA-protein interaction data, drawn from work I contributed to other publications. These included the discovery of an RNA-protein interaction involved in prostate cancer progression, the identification of an RBP-RNA interactome, as well as the integration of CLIP-seq-related tools into the Galaxy platform. Integrating tools into Galaxy facilitates their usage and helps to make data analysis easily accessible, especially for users with less experience in bioinformatics. Moreover, it ensures long-term tool availability and support, as well as full reproducibility of results.

Publication P2 introduced MechRNA, the first tool capable of inferring the functional mechanisms of lncRNAs based on their predicted interactions with other RNAs and proteins. Despite the scarcity of studies on functional lncRNA mechanisms involving RNA-RNA interactions, we showed that MechRNA is able to detect known mechanisms, demonstrating its usefulness by providing plausible functional hypotheses. Computational methods like MechRNA should therefore become valuable assistants for the study of lncRNA mechanisms, especially since the huge majority of annotated lncRNAs in the human genome are still lacking functional characterization.

Publication P3 presented RNAProt, a novel RNA-protein binding site prediction tool based on recurrent neural networks. Compared to other recent deep learning-based methods, RNAProt offers both state-of-the-art predictive and superior runtime performance. In addition, it supports more predictive features and input data types than any other available method, including user-defined predictive features. The fast runtimes allow for on-the-fly model training, enabling the quick testing of different features, parameter settings, or input data types before moving on to predictions. We showed that RNAProt's visualizations agree with known RBP binding preferences. Moreover, its additional features can boost predictive

performance and enhance the specificity of predictions compared to models using only sequence information. RNAProt comes with a comprehensive documentation on GitHub and is available as a package for the Conda package manager, ensuring easy installation and usage. All this makes RNAProt a valuable tool for the large-scale prediction of RBP binding sites and related studies.

Publications P4 and P5 both dealt with improving CLIP-seq data analysis through the incorporation of transcript information. As RBPs can either bind to a genomic (i.e., unspliced RNA) or transcript (i.e., spliced RNA) context, CLIP-seq data also includes information on the underlying transcript structure and splicing events. However, none of the currently available peak callers takes into account transcript information when determining RBP binding site locations, instead relying solely on the genomic context. Publication P4 investigated the consequences of ignoring this information and its effects on the quality of peak calling. We showed that peak calling of current peak callers is compromised for RBPs binding predominantly to exons (i.e., to a transcript context), and that the amount of publicly available CLIP-seq peak regions susceptible to the problem is substantial. Furthermore, changing the genomic context of peak regions to transcript context boosted performances of RBP binding site prediction tools, and led to an enrichment of RBP binding motifs associated with exon-binding RBPs. The findings of P4 enabled us to develop a method to specifically target the described problems. Introduced in P5, Peakhood is the first tool capable of selecting the most likely context for a set of CLIP-seq peak regions, individually for each region. We showed that its context extraction is in agreement with known RBP roles. In addition, Peakhood determines the most likely transcript for each transcript context site, and can merge transcript context sets into comprehensive transcript site collections. These for example allow the user to quickly identify interesting site pairs of same or different RBPs, where the distance on the transcript is lower than the original genomic distance, or within some desired interval. Just like RNAProt, Peakhood comes with a Conda package and a comprehensive documentation for easy installation and usage. Peakhood is thus ideally suited for the application in CLIP-seq data analysis pipelines, as a post-processing step after peak calling, or to reanalyze any of the millions of publicly available CLIP-seq peak regions determined by various peak callers.

4.2 Outlook

Throughout its life cycle, an RNA can travel to different cellular locations and interact with various proteins and RNAs to form RNP complexes of changing composition. Consequently, to solve the gene expression puzzle, scientists will have to elucidate the composition and functions of these RNP complexes, how they form, their dynamics, and their localization. A continuously growing amount of experimental and computational methods enables and supports these studies. This section briefly describes some trends regarding experimental and computational methods to investigate RNA-protein interactions and RNP complexes,

as well as a more broader outlook on upcoming research directions at the end.

Experimental outlook

As each experimental approach provides a certain type of information, but also has its own biases and limitations, combining data obtained from different methods allows us to get a less biased and more complete view on RNA-protein interactions. There are many factors that determine or influence RNA-protein interactions in the cell, such as competing or co-operatively binding RBPs, RNA structure and RNA-RNA interactions, RNA modifications, the subcellular location of RNAs and proteins, as well as the precise identity of the target RNA (i.e., which splice isoform is targeted). Many of these can already be detected by current experimental approaches, and future approaches will undoubtedly improve upon them to further increase their accuracy. For example, various protocols are available for the transcriptome-wide identification of RNA structures and RNA-RNA interactions, which also include CLIP variants to detect the binding sites of double-stranded RNA binding proteins [230]. Moreover, RNA-centric approaches can identify the set of proteins bound by a specific RNA [155]. In addition, in vitro CLIP approaches can be used to investigate RBP binding isolated from in vivo factors and their contributions to the binding [231, 232]. Studying RNP complexes while considering the subcellular locations in which they occur will be especially important, as the availability of RNA and proteins inside a subcellular compartment determines the types of possible interactions. While there are already some methods available to identify RNA-protein interactions and RNA structuromes specific to certain subcellular compartments [233, 234, 235], more primary research will be needed to better understand RNA localization and RNP granule formation [75, 236]. In this regard, a number of recently developed microscopy techniques to visualize the insides of cells with unprecedented detail, including proteins and other biomolecules, will likely be of great help as well [237]. Another important task is to study the dynamics of RNA-protein interactions, to learn more about RBP binding kinetics and affinities towards different target sequences. In this regard, a number of in vitro studies have been carried out to measure RBP or single protein domain affinities to rather short sequences, but so far in vivo studies are still scarce [132, 238]. As for determining which splice isoforms are present in a given sample, long-read or direct sequencing methods allow us to differentiate between isoforms which cannot be distinguished by standard short-read sequencing methods [147].

Computational outlook

Advancements in experimental methods and the combined analysis of the datasets they produce will further improve our understanding of the molecular processes that underlie RNA-protein interactions. Moreover, the increasing quality and quantity of datasets will allow for more accurate computational models, which in turn enable the generation of new hypotheses to test and further deepen our knowledge. In addition, more and more powerful machine learning and in particular deep learning methods get published and refined at a

staggering pace. Typically, such methods are successfully applied in bioinformatics within a few years after their original publication, but this could be further accelerated by interdisciplinary cooperations and a focus on biological tasks early on in their development [239]. As deep learning methods are typically hard to interpret, they might not be the best choice if explainability of the model predictions is the prime goal. On the other hand, the development of methods to explain model predictions by visualizing model preferences is a highly active research area on its own, which will likely further increase the usability and acceptance of deep learning methods in the near future, especially for tasks that depend on explainable predictions [240]. The same holds for the required amount of training data and computational resources, which is typically high for deep learning methods, especially for the more recently developed top-performing methods. Nevertheless, there are also plenty of promising techniques to counteract these issues, such as data augmentation, transfer learning, generative models (including autoencoders), or meta learning [241, 242, 243, 244]. For example, highly popular transformer methods such as BERT use transfer learning, meaning that they rely on pre-trained models which can be fine-tuned with relatively small amounts of data and training time to a new task [245]. Transformers present the current state-of-the-art in natural language processing, and have recently also shown great potential on biological sequence data [239, 246, 247]. Furthermore, they have been part of two recent deep learning methods, obtaining unprecedented accuracies in predicting the 3D structures of proteins from their primary sequences, which has spurred hopes for similarly capable methods to predict 3D RNA structures in the near future [248]. Merging these approaches and combining them with additional information on RNA modifications, localization, or transcript and protein identity could eventually allow us to realistically model the 3D complex structures of RNA-protein interactions. Another drawback of deep learning methods is that they usually demand a profound knowledge of the subject matter in order to successfully implement them. One solution to this issue can be automated machine learning (AutoML) approaches, which ease the implementation of machine learning applications by automatising, e.g., data preparation, neural architecture search, and hyperparameter optimization, this way making them available also to less-experienced users [249].

From a broader perspective

From a broader perspective, there are a number of key challenges that have been or need to be taken up, which will eventually allow us to get a more profound understanding of gene expression. The following discusses three of them: ncRNAs' roles in genome organization, single-cell studies, and extensions of the reference genome.

A number of recent studies indicate that ncRNAs strongly contribute to 3D genome organization, and therefore to gene expression as well [250]. Although their mechanisms are not well understood yet, RNA-DNA triple helix (or triplex) interactions could play an important part in them [251]. Moreover, there is evidence that RNAs bound to DNA can act as scaffolds for the recruitment of protein complexes to specific genomic loci [251, 252],

motivating the development of new computational methods that combine triplex and RNA-protein interaction predictions.

The second challenge lies in better understanding the cellular diversity of gene expression, both in between cells of same and different cell types and tissues, but also over time and developmental stages, which can be captured by single-cell experimental procedures. Naturally, the limited amount of biological material obtained from single cells amplifies certain issues, such as an increased technical variability between experiments or the problem of missing data [253, 254], calling for specialized experimental and computational methods. Over the last years, various methods have been proposed to study single-cell genomics, epigenomics, transcriptomics, and more recently also proteomics [255, 256]. As even optimized CLIP-seq protocols typically require $>$ 20,000 cells [182], they cannot be easily adapted to single cells. However, a recent approach which relies on fusing a C-to-U editing enzyme to an RBP of interest promises the transcriptome-wide identification of RBP binding sites at single-nucleotide and single-cell resolution [257], yielding hope for more methods alike to come out in the near future. Moreover, the method can be coupled with long-read sequencing for the detection of isoform-specific RBP binding sites, allowing for an even more accurate site identification.

The third challenge is the extension of the reference genome. This is necessary since our current reference only includes one copy of the genome (i.e., the haploid genome), whereas most of the cells in our body contain two sets of chromosomes (i.e., a diploid genome), one from each parent (also termed maternal and a paternal haplotypes). Higher ploidy levels can be found in other organisms, e.g., in many plants such as crops [258]. Extended references will allow us to to study and understand genetic variation between haplotypes, as well as on a broader scale between individuals in a population. Over the last years, various solutions have been proposed to generate such references (also termed consensus references or pan-genomes), including the implementation of computational methods to efficiently represent and process these new references [259, 260]. Long-read sequencing methods again play an important part in this development, since, due to their extended read lengths, they can identify genetic variants which commonly used short-read sequencing methods fail to or have problems to detect, such as structural variants and repeat regions [261, 262]. Nevertheless, there are also approaches which utilize short-read sequencing methods, since these are widely available and currently still offer higher efficiency with regards to costs and throughput [263].

Chapter 5

Publications

This chapter contains the five publications P1-P5 in their published form. Each publication is preceded by a declaration of co-authorship, which describes the individual contributions of all co-authors. The declarations were confirmed by all co-authors via e-mail and signature, which are also provided below each declaration.

[P1] Computational analysis of CLIP-seq data

Publication:

- [P1] **Michael Uhl***, Torsten Houwaart*, Gianluca Corrado, Patrick R. Wright, and Rolf Backofen*. **Computational analysis of CLIP-seq data**. *Methods*, 2017.

Contributions of individual authors:

"I am the main contributor to this work, together with Torsten Houwaart and Rolf Backofen. I was largely involved in discussing, selecting, and writing the content of this review article. I contributed major parts to the paper, including the introduction, the overview of CLIP-seq variants, as well as parts of the analysis of CLIP-seq data and the conclusion. Torsten Houwaart contributed most of the analysis of CLIP-seq data, including the peak caller comparison. Rolf Backofen contributed the postprocessing section. Gianluca Corrado provided the section on PARalyzer, and Patrick R. Wright the section on block-based peak calling. All authors contributed to the revision and approved the final manuscript."

Michael Uhl

The following co-authors confirm the above-stated contributions:

* joint first authors

Methods 118–119 (2017) 60–72

Contents lists available at ScienceDirect

Methods

journal homepage: www.elsevier.com/locate/ymeth

Computational analysis of CLIP-seq data

Michael Uhl [a,1], Torsten Houwaart [a,1], Gianluca Corrado [c], Patrick R. Wright [a], Rolf Backofen [a,b,*,1]

[a] Bioinformatics Group, Department of Computer Science, University of Freiburg, Germany
[b] Centre for Biological Signalling Studies (BIOSS), University of Freiburg, Freiburg, Germany
[c] Department of Information Engineering and Computer Science, University of Trento, Italy

ARTICLE INFO

Article history:
Received 4 October 2016
Received in revised form 17 February 2017
Accepted 20 February 2017
Available online 22 February 2017

Keywords:
CLIP-seq data analysis
Peak calling
RBP binding models
RBP binding site prediction

ABSTRACT

CLIP-seq experiments are currently the most important means for determining the binding sites of RNA binding proteins on a genome-wide level. The computational analysis can be divided into three steps. In the first pre-processing stage, raw reads have to be trimmed and mapped to the genome. This step has to be specifically adapted for each CLIP-seq protocol. The next step is peak calling, which is required to remove unspecific signals and to determine bona fide protein binding sites on target RNAs. Here, both protocol-specific approaches as well as generic peak callers are available. Despite some peak callers being more widely used, each peak caller has its specific assets and drawbacks, and it might be advantageous to compare the results of several methods.

Although peak calling is often the final step in many CLIP-seq publications, an important follow-up task is the determination of binding models from CLIP-seq data. This is central because CLIP-seq experiments are highly dependent on the transcriptional state of the cell in which the experiment was performed. Thus, relying solely on binding sites determined by CLIP-seq from different cells or conditions can lead to a high false negative rate. This shortcoming can, however, be circumvented by applying models that predict additional putative binding sites.

Contents

* Corresponding author.
[1] These authors contributed equally to this work.

http://dx.doi.org/10.1016/j.ymeth.2017.02.006
1046-2023/

1. Introduction

The rise of next-generation sequencing (NGS) techniques over the past decade has led to an enormous boost in RNA research thanks to numerous discoveries concerning the fundamental role of RNA in gene regulation [1]. To exert these functions, RNAs in eukaryotic cells can form ribonucleoprotein complexes by interacting with a multitude of RNA-binding proteins (RBPs), allowing for the evolution of complex regulatory networks. Recent studies revealed more than 1500 RBPs in human cells, which emphasizes their fundamental importance for virtually all aspects of post-transcriptional gene regulation (PTGR), including RNA maturation, alteration, transport, stability, and translation [2–5]. Beside their physiological roles, various diseases have been linked to dysregulated or deficient RNA-binding proteins [6,7]. Hence, a comprehensive understanding of RNA-based networks is only possible when also considering the contributions of these RBPs. The scientific community is therefore increasingly turning to the characterization of RBP-based regulation. RBPs regulate their target gene(s) by directly binding to the transcribed RNA. Typically, specific sequence motifs are required for binding site recognition, although the relative contributions of RNA sequence, structure and backbone to the binding can differ greatly among RBPs [8,9]. The majority of RBPs appears to prefer single-stranded regions [4]. There are, however, also many RBPs that prefer structured RNA such as Staufen 1 [10], Roquin [11], or MLE [12]. It is currently unclear to what extent the observed general tendency towards single-stranded RNA regions is caused by biases in the experimental protocols. As RNA molecules generally form extensive secondary structures, it is not surprising that the binding specificity of RBPs also strongly depends on the structural context of their binding sites. Indeed, the importance of binding site accessibility has been shown for many RBPs [13].

The recent development of high-throughput protocols for determining RBP binding sites on a genome-wide scale has greatly influenced the field and opened up new avenues for the investigation of regulatory relationships. Particularly, CLIP-seq (cross-linking and immunoprecipitation followed by next generation sequencing) [14] has become the standard experimental procedure for studying transcriptome-wide RBP binding. Briefly, RBPs are crosslinked to their RNA binding sites, followed by extraction and sequencing of the crosslinked RNA fragments. After mapping of the sequenced fragments, binding regions are identified based on the read profiles and various additional information (e.g. from control experiments or replicates). The process of determining significantly enriched binding regions is also known as peak calling. Subsequently, binding motifs or predictive models can be derived from the identified sites. These can then be employed to identify potential binding sites in yet unreported target sequences.

In this paper, we describe selected tools and pipelines required for a comprehensive bioinformatics analysis of CLIP-seq datasets. We do not intend to give a complete overview of available methods, since there is already plenty of literature available on CLIP-seq data analysis [15–17]. Rather, we will concentrate on tools which have proven valuable to us in the past. For these, we will describe important aspects of a comprehensive analysis. A special focus will lie on the process of peak calling, which is the process of recovering bona fide protein binding sites by signal detection and removal of false positives originating from unspecific interactions. To our knowledge, this component of the data analysis is still lacking a more comprehensive discussion in literature, even though it is arguably the most critical part of the whole analysis. We will start with a description of the different CLIP-seq variants available, addressing specific features and (dis) advantages. In the following section on CLIP-seq data analysis, we will describe the different steps of pre-processing, mapping and peak calling in greater detail. The last section considers the task of determining binding models for computational binding site prediction. Such models are needed to reduce the false negative rates of CLIP-seq experiments which originate from their dependency on the expression of the detected RNA binding sites. Without these models, information from published data cannot be transferred to different cells or conditions.

2. Overview of CLIP-seq variants

In recent years CLIP-seq has become the standard experimental procedure to identify binding sites of RBPs on a transcriptome-wide level. Several variants have been proposed since the introduction of CLIP [18,19] in 2003 and its first high-throughput sequencing extension HITS-CLIP (high-throughput sequencing of RNA isolated by CLIP) [14] in 2008, each addressing various shortcomings of the previous versions. The most widely used modifications over the last years are PAR-CLIP (photoactivatable-ribonucleoside-enhanced CLIP) [20] and iCLIP (individual-nucleotide CLIP) [21], while recently the eCLIP (enhanced CLIP) protocol [22] was introduced and promoted by the ENCODE consortium. Another protocol termed irCLIP (infrared-CLIP) [23], which has been compared to eCLIP [24,25], has also been published in 2016. Besides, several specialised modifications for double-strand binding RBPs exist [26,10,27]. These protocols add an additional ligation step to the standard protocol in which the two double-strand RNA segments bound by the RBP are connected, leading to chimeric reads that allow for the simultaneous identification of both RNA strand regions. So far, CLIP-seq has been applied in numerous studies on single RBPs. Furthermore, the method has been employed by a study on global mRNA binding preferences [2].

2.1. Principle CLIP-seq workflow

The principle workflow of a CLIP protocol starts with UV radiation of the cell or tissue culture, which induces covalent crosslinks between RBPs and their bound RNAs. This is followed by immunoprecipitation of the RBP-RNA complexes and partial RNase digestion to narrow down the binding sites to appropriate sequencing and mapping lengths. Further steps aim at stringent purification, including radioactive labeling, recovery by SDS–PAGE, transfer to nitrocellulose membrane to abolish loose RNA fragments, excision and proteinase K treatment to remove the RBP and recover the trimmed RNA fragments. Finally, the fragments are reverse-transcribed and their cDNAs are subjected to deep sequencing. The resulting sequencing data is then analysed to obtain RBP binding sites which can be identified based on the mapped read profiles.

2.2. PAR-CLIP

PAR-CLIP [20] marked the first successful adaptation of the original protocol, introducing a number of modifications over HITS-CLIP. To increase crosslinking efficiency, cells are additionally supplemented with 4-thiouridine (4SU), and UV radiation is applied at

365 nm instead of 256 nm. Interestingly, these modifications also lead to a high number of thymidine to cytidine transitions in the cDNA at the crosslink sites, which can be exploited in a subsequent mutational analysis for pinpointing the crosslink position, thus basically enabling PAR-CLIP to achieve single-nucleotide resolution. On the other hand, 4SU usage restricts the method to cell cultures and preferential crosslinking to 4SU naturally biases site recovery towards U-containing sites. Also, 4SU exhibits an increased affinity towards G:U base pairing [28], which might influence cellular RNA structure and thus also RBP binding. In addition, RNase T1 digestion leads to a depletion of G-containing sites, due to the enzyme's preferential cleaving after G nucleotides [29]. Another problem is the usage of inducible tagged proteins in the original publication, which can result in the recovery of non-physiological binding events due to overexpression. The last two problems can and have been addressed in subsequent PAR-CLIP versions [29,30], where the latter one also describes an *in vivo* approach for *C. elegans*.

2.3. iCLIP

iCLIP [21] has been particularly designed to address a specific problem inherent to HITS-CLIP and PAR-CLIP: during cDNA synthesis, the reverse transcriptase frequently stalls at crosslink sites still containing residual peptides, leading to an estimated loss of over 80 % of cDNA fragments [31]. To solve this issue, the authors developed a two-part cleavable adapter together with an additional circularization and linearization step, allowing for the recovery of both complete and truncated cDNAs. Additionally, random barcodes are used, enabling easy identification and removal of PCR duplicates after mapping. These measures lead to increased efficiency, while single-nucleotide resolution is achieved due to the truncated cDNAs which pinpoint the crosslink position to the reads' 5' ends. Still, as with PAR-CLIP and the original HITS-CLIP, the protocol remains time-intensive (up to 5 days) and error-prone due to its many different steps [24]. Also, a fairly huge amount of starting material (typically 10^6–10^8 cells) is required in order to generate a library of sufficient complexity. This often makes successful library preparation difficult. This is especially true for lowly expressed RBPs, RBPs with widespread binding or RBPs with low crosslinking efficiencies and/ or antibody affinities.

2.4. eCLIP

Both eCLIP [22] and irCLIP [23] have been developed to deal with the shortcomings of previous CLIP-seq variants. Particularly, high demands in cell numbers, many different preparation steps including radioactive reagents and long preparation times frequently result in poor library generation efficiency. In the eCLIP protocol, the inefficient circularization step from iCLIP is exchanged by two separate adapter ligation steps, which results in much higher RNA fragment recovery. This ultimately leads to a significantly improved library complexity. Furthermore less cells are needed. Both aforementioned improvements enable the application of this method on formerly difficult RBPs. Single-nucleotide resolution is achieved the same way as in iCLIP, meaning that the reads' 5' end should mark the crosslink position for the huge majority of reads. Moreover, the autoradiographic visualization step is omitted and different samples can be pooled early in the protocol. This allows for much faster preparation times, but leaving out the autoradiographic step is also a clear drawback since the quality of the IP can no longer be monitored. Another new feature is the inclusion of a size-matched input control (SMInput), which enables efficient background normalization and thus leads to a higher specificity in subsequent binding site identification. For SMInput, 2 % of the pre-immunoprecipitation sample is taken and sequenced together with the immuno-purified sample. It was shown that normalization by SMInput significantly improves authentic binding site recovery, whereas an IgG control, which is frequently employed as a CLIP-seq control, was found unsuitable for this task. The authors also provide a peak calling pipeline called CLIPper [32], which will be discussed in a later section. The described improvements have made eCLIP the method of choice for the ENCODE consortium. So far, the consortium has published eCLIP data for more than 70 diverse RBPs, which underlines its usability and will likely help eCLIP to become more popular in the near future.

2.5. irCLIP

Compared to eCLIP, irCLIP [23] uses a complementary approach to deal with the described shortcomings of previous CLIP protocols: the circularization step from iCLIP is kept but optimized and applied in a single-tube reaction together with reverse transcription to reduce preparation time. In addition, both circularization and reverse transcription are performed at 60°C using thermostable enzymes to resolve potential RNA secondary structures. It will be interesting to see whether this step also helps to improve binding site recovery in the case of structure-binding RBPs, which might yield low library complexities for other CLIP protocols. irCLIP achieves single-nucleotide resolution analogous to iCLIP and eCLIP. Like eCLIP, irCLIP too skips radioactivity steps, but instead introduces an infrared fluorescent dye to visually check IP quality. It can thus prevent certain IP-related quality issues which can become a problem in the eCLIP protocol, since eCLIP ommits the autoradiographic visualization without substitution. Infrared dye labeling also improves other steps of the protocol, which as with eCLIP results in lesser starting material (typically only 20,000 cells) and overall increased efficiency. On the other hand, working with infrared dyes also requires specialized equipment, such as a gel documentation system with near-infrared capabilities, which might not be highly available or affordable [24]. It remains to be seen which of the two protocols will be applied more frequently by the field. In any case, future comparisons in recovered binding profiles should help to reveal protocol-specific advantages and biases.

3. Analysis of CLIP-seq data

The analysis of CLIP-seq data usually involves three major steps which will be addressed here. As in many other protocols, the reads first have to be mapped to a reference genome. If the CLIP experiment was performed for a specific RBP, the generated reads should agglomerate in regions to which the RBP binds. To identify these regions, a second step is performed under application of a peak caller. Peak callers are used on the coverage profiles to determine regions that are bound by the RBP with high affinity. Once the peaks are identified, they can be quantified and their statistical significance should be evaluated by comparing them to a control experiment. In a third step, the resulting data can be utilized to find binding motifs and to train binding models, which enable the prediction of novel RBP binding sites on transcripts not present in the CLIP-seq data. The last step is especially important when investigating RBP binding sites in cells or conditions for which no CLIP-seq data is publicly available.

3.1. Preprocessing of raw data and mapping

Most CLIP-seq studies are performed on organisms with well annotated genomes like human, mouse or *C. elegans* [33]. Reads

from CLIP-seq experiments performed on these organisms can be mapped to the according reference genome or transcriptome. A major problem regarding the quantification of read data is the reliance of sequencing-based techniques on PCR amplification of the sequence libraries prior to sequencing. Although necessary in order to generate a sufficient amount of sequencing material, the occurrence of some sequences can be artificially boosted in the process because of biases in the PCR protocol, where so-called PCR duplicates are introduced. With the introduction of random barcodes or unique molecular identifiers (UMI) in iCLIP this problem is mitigated, as reads which contain the same random barcodes and map to the same coordinates can be collapsed to unify all PCR duplicates into just one representative. The methodology is not completely flawless though, as it has been shown that during library preparation mutations can be introduced in the random barcodes which can have a big effect on the crosslinking-event counts [12]. Before mapping the reads, these UMIs have to be removed. Tools such as flexbar [34] can be used to accomplish this. If no UMIs were used then tools such as FastUniq can be employed to collapse potential PCR duplicates [35]. Adapters that are used in the amplification steps of the sequences also have to be trimmed from the sequences. Several programs can be used for this, e.g. cutadapt [36], Trim Galore[2], which is based on cutadapt and fastqc, or trimmomatic [37], which is specifically made for Illumina sequencing data.

A few things have to be considered in order to correctly map the trimmed reads to a reference genome. In most cases, this step consumes the most computational power. The sequences stem from RNA molecules which can be subject to splicing in eukaryotes. The choice of the mapping software depends on prior knowledge about the targets of the RBP and is not independent from the following peak calling step, since the peak caller has to deal with gaps which occur in spliced reads. The reads can either be mapped to the genome or the transcriptome. The advantage of mapping the reads to the transcriptome is that a higher sensitivity can be achieved, but it also comes at the cost of limiting the analysis to known transcripts. Since RBP binding sites can be located in introns (especially in the case of splicing regulators), mapping only to exonic parts would lead to the exclusion of these sites. Mapping to exons also leads to a depletion of sites spanning exon borders, since the read parts are often too short to be mapped to their corresponding exons with sufficient quality. All these issues have to be considered in order to choose a meaningful mapping strategy. A layered procedure of first mapping strictly to the transcriptome and afterwards mapping the remaining reads to the genome is often used and might work best in such cases. A wide range of mapping algorithms originally developed for RNA-seq are available. To list a few good choices for this task, TopHat [38], GSnap [39] and segemehl [40] fulfill the aforementioned requirements and are widely used, but also STAR [41] should be mentioned, which is the mapper of choice in the eCLIP pipeline used by the ENCODE consortium. Of course this list is not comprehensive and many other good choices exist. Benchmarking and isolating the best program for this task go beyond the scope of this review and can be found elsewhere [42–44].

3.2. Methods for peak calling

The next task after mapping reads to a reference genome or transcriptome is to extract authentic binding sites from the mapped read profiles. Many reads stem from unspecific binding and thus have to be discarded, which is done in the process of *peak calling*. This task can typically be divided into two parts: one first extracts potentially interesting peaks based on peak shape or height and then filters the resulting peaks such that only sites enriched over a certain threshold or background are kept. The first part usually results in a huge number of initial sites, including many false positive predictions. The second part therefore should incorporate additional experimental information like read profiles from replicates, controls, or RNA-seq samples in order to increase the signal-to-noise ratio. Information on underlying transcript abundances is particularly important to peak calling on CLIP-seq data, since transcript amounts differ between transcripts from different loci, and thus directly influence the peak heights found in the read profiles. Therefore one cannot be sure if e.g. a high peak corresponds to a strong binding site or if this is just the result of the underlying transcript being highly expressed in the observed cell type or condition. A correction for transcript abundance is therefore of fundamental importance in CLIP-seq peak calling. Interestingly, correction for transcript abundance has been shown to significantly improve peak calling results even in the case of external RNA-seq data [45]. Ideally however, one should choose a CLIP-specific control for background correction, which also incorporates protocol-intrinsic biases. The authors of eCLIP [22] e.g. showed that using a pre-immunoprecipitation control (as described in the eCLIP section) led to a significant enrichment of true binding sites, whereas an IgG control, which is frequently used in CLIP, was not suitable for background correction. In general, controls that produce low complexity libraries and thus poor coverage of the underlying transcriptome should be avoided. Besides using controls, results from different replicates can be intersected to further increase specificity. In order to assign significance values to peaks, it is also important to find a suitable probability distribution for modeling the underlying read counts. In the following, some prominent CLIP-seq peak callers which have been used by our group will be discussed in more detail.

3.2.1. Piranha

Piranha [45] is a CLIP-seq peak caller which can be applied to all available CLIP-seq as well as RIP-seq datasets in order to identify significant peaks. It was the first generic CLIP-seq peak caller developed, i.e. it does not depend on certain CLIP variant properties in order to call peaks, as opposed to PARalyzer [46], which relies on PAR-CLIP data, or CIMS [47] and CITS [48], which were developed for HITS-CLIP. Based on the mapped reads as input, Piranha first divides the genome into non-overlapping bins of a user-defined size and counts the number of read starts falling into each bin. Piranha assumes that the read starts define the site where the crosslink events take place. Bins with zero counts are discarded, and the counts of the remaining bins are then used to fit a probability distribution. Covariates, e.g. in the form of reads from RNA-seq or a CLIP-seq control experiment, can be supplied to correct for different transcript abundances or protocol biases. In the case of covariates, Piranha uses a zero-truncated negative binomial regression for fitting the read counts together with the supplied covariate data. If no covariates are given, the user has the choice between four different distributions. However, the zero-truncated negative binomial distribution is set as default and recommended, as it was shown to have the best fit on a collection of over 100 CLIP-seq datasets. Since Piranha assumes that most read-covered sites represent background binding, the fitted distributions essentially model background probabilities. Therefore, the p-value of a given bin corresponds to the probability of the site being background. By default, Piranha reports p-values corrected for multiple testing using the Benjamini-Hochberg method [49] with a default threshold of 0.05. As for the bin size, the authors suggest the size to be adapted to the depth of coverage and the CLIP-seq variant used. This is of course not intuitive, especially for novice users. According to the authors, a good starting point for RIP-seq is 100, while e.g.

[2] Felix Krueger. Trim galore. https://github.com/FelixKrueger/TrimGalore, 2016.

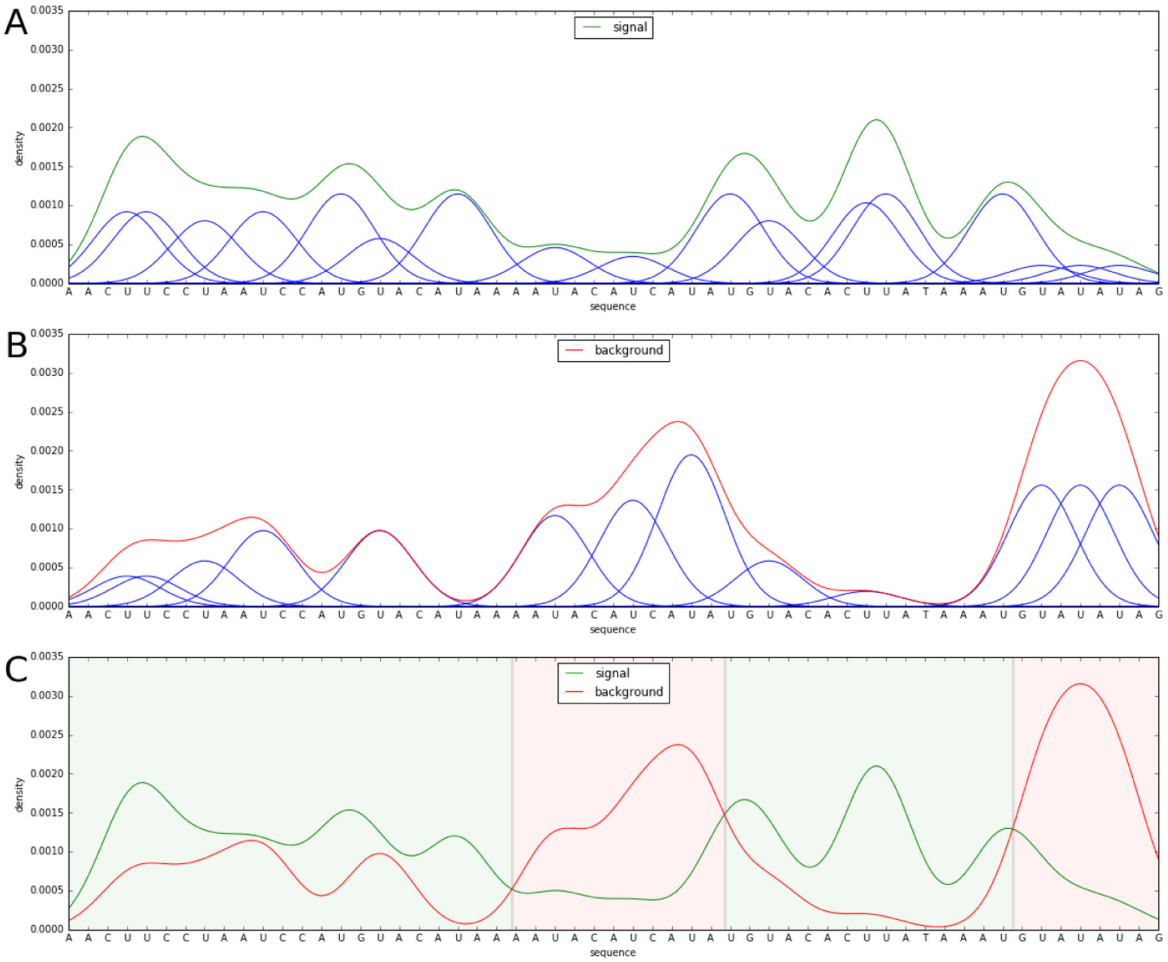


Fig. 1. Crosslink site indentification with PARalyzer on a synthetic example. Class-specific densities for both the signal and the background are estimated using a Gaussian kernel density estimator. (A) Density estimation of the signal (green function). For each T nucleotide a Gaussian with fixed variance is peaked representing the number of T to C conversions occurring in the position, normalized by the total number of T to C conversion in the associated read group. The normalized sum of all their Gaussian functions is the signal. (B) Density estimation of the background (red function). The estimation is based on the number of T nucleotides that have not turned into Cs. (C) After estimating the class-specific densities, the interaction sites are defined by the nucleotides where the density estimate of the signal (T to C conversions, green line) is greater than the one for the background (non T to C conversions, red line) (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.).

for iCLIP, one could start with low sizes (e.g. 5 nt) and then depending on the amount of noise in the dataset gradually increase the window size. Either way, having to deal with manually adjustable bin sizes is a clear drawback of Piranha. In addition, it lacks support for the integration of replicate information, although one could still do a manual intersection by calling peaks on all replicates separately and merging the results afterwards.

3.2.2. PARalyzer

PARalyzer [46] is a computational tool[3] for the discovery of crosslinking sites from PAR-CLIP sequencing data. In the PAR-CLIP protocol the protein crosslinking is boosted by additionally culturing the cells with a photoreactive ribonucleoside analogue, usually 4SU. The crosslink product of 4SU is known to have a preferential base pairing to guanine (G) instead of adenine (A), resulting in thymine (T) to cytosine (C) conversions in PCR-amplified cDNA.

[3] PARalyzer is available at https://ohlerlab.mdc-berlin.de/software/PARalyzer_85/.

The rationale of PARalyzer is to examine the pattern of T to C conversions in order to spot, with high confidence, RNA–protein interaction sites. A kernel-density-based classifier is used to characterize crosslinked regions, identified by T to C conversions (the signal), against not crosslinked ones, characterized by the absence of T to C conversions (the background).

Class-specific densities (one for the signal and one for the background) are assessed by employing a Gaussian kernel density estimator that, for each T nucleotide, considers the number of T to C conversions and the number of non T to C conversions in the aligned reads. For each T nucleotide in the RNA sequence, the number of T to C conversions occurring in that position is represented using a Gaussian distribuion with fixed variance. The distribution is peaked on the T nucleotide and the variance distributes the signal over the neighbouring nucleotides. The function in green, shown in Fig. 1A, is the sum of all the individual Gaussian distributions that indicate T to C conversions and represents the signal. The background (red function in Fig. 1B) is estimated by summing all the Gaussian contributions of T nucleotides that have not turned

into C nucleotides instead of the T to C conversions. After estimating the class-specific densities, the interaction sites are defined by the nucleotides for which the density estimate of the signal (T to C conversions) is greater than the one for the background (non T to C conversions) (Fig. 1C).

3.2.3. CLIPper

To distinguish peak regions from non-peak regions, the CLIPper software [50] utilizes different statistical measures. CLIPper is intended for calling CLIP-seq peaks on known genes only and therefore requires annotation. It provides annotations for a few genome assemblies, i.e. hg19, mm9, mm10, and ce10. For other species the user has to provide the annotation. The program defines sections on the genome where reads agglomerate and identifies peaks on the read profiles. A threshold is defined based on the amount of reads in each section, the amount of reads in the vicinity of the section, and the amount of reads in the gene. The threshold specifies the minimum amount of reads necessary within this region to be deemed statistically significant. This procedure makes sure the false positive rate of peaks is controlled. By default, CLIPper then fits a spline function to the read profile and defines regions which are above the threshold and those that are in between local minima of the fitted spline as peaks. For these peaks, a p-value is calculated with the amount of reads in the peak region X being modeled as $X \sim \text{Poisson}\left(1 + \text{reads\_in\_gene} \cdot \frac{\text{peak\_length}}{\text{gene\_length}}\right)$. This procedure assigns p-values to all peaks which in turn can be corrected for multiple testing with respect to all tested peaks using the Benjamini-Hochberg procedure [49]. The local maxima in the fitted splines are explicitly highlighted (in the resulting BED file) because these positions are the best candidates for where the analysed RBP binds to. In the eCLIP pipeline that is used by the ENCODE consortium [51] the peaks are annotated qualitatively after their identification. Each CLIP experiment dataset can be compared to one control dataset. For each peak a log2-fold-change is calculated based on the mapped reads within the peak region for the experiment in comparison with the control. Furthermore, a p-value is determined for each peak using a χ^2 test or Fisher's exact test using the mapped and total reads of the experiment and control. Given that eCLIP pinpoints the crosslink positions to the read starts, it is surprising that CLIPper does not take advantage of this information, instead considering the full-length reads for peak calling.

3.2.4. Block-based peak calling

A recent CLIP-seq study on the transcriptome-wide binding sites of the bacterial RBPs Hfq and CsrA in the human pathogen *Salmonella enterica* [52] introduced an experimental procedure including paired-end signal and background libraries in triplicates. The library preparation for the background data solely differs in the fact that no UV induced cross linking is performed. The identification of significant peaks in the signal data was performed based on a three step procedure. In the first step, the blockbuster algorithm [53] subdivides the pooled signal sequencing data into clusters (C) of blocks (B). A block is fundamentally a pile of similar reads which is characterized by its beginning position b(B), its ending position e(B), its size S(B) and its length l(B) (see Fig. 2). Since the block boundaries set in this initial structure do not provide appropriate peak boundaries, overlapping blocks are joined into peaks in the second step of the procedure. The biggest block (B_m) is selected from C if $S(B_m) \geqslant S(C) * 0.01$. Then, all B overlapping with B_m are selected and removed from C. The peak boundaries are extended using all blocks that overlap with at least half of B_m and also fulfill $S(B) \geqslant S(B_m) * 0.1$. The leftmost and rightmost coordinates of the remaining B are set as final boundaries and the procedure restarts by selecting the next B_m. In the last step, the DESeq2 algorithm [54] assesses the statistical significance of each peak based on the individual amount of reads counted for each of the peaks in the signal and background libraries. The final output is a p-value sorted list.

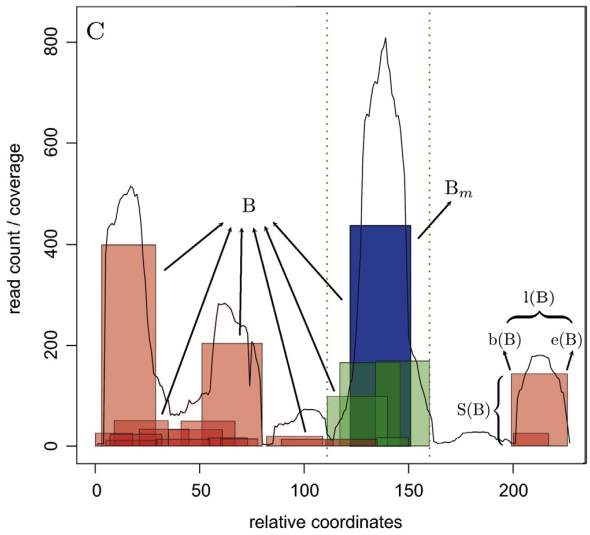


Fig. 2. The figure shows a specific cluster (C) of blocks (B) and their attributes e(B), b(B), l(B), S(B). The first B_m selected for this C is shown in blue, while overlapping blocks that reach at least into the middle of B_m are green. The dotted green lines show the borders of the first peak. All B not used for the definition of the boundaries of the first peak are red. The density of sequencing reads at nucleotide resolution is shown as black line (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.).

3.2.5. Summary and comparison

A meaningful and fair comparison of the different peak callers is problematic. On the one hand each tool incorporates several parameters which change the behaviour of each peak caller significantly. On the other hand no datasets of absolute truth exist on which the different tools can be benchmarked. Tools that work only with very specific protocols because they rely on signatures in the data that are introduced in these protocols can not be fairly compared. In the following discussion, PARalyzer was not considered because its method of finding peaks is specific to PAR-CLIP data and not applicable to other CLIP-seq methods. For the other tools (CLIPper, Piranha and block-based peak calling) specific filtering steps were undertaken as explained in the following. To give a quantitative measure for the comparison of the different peak callers we propose a genomic position based metric. One position corresponds to one nucleotide in the reference genome of the investigated organism. Each position that is assigned to a peak by at least one peak caller is evaluated on whether it is also within a peak region defined by the other tools.

The problem of peak calling can be considered as two distinct steps. The first step consists of defining regions of interest solely based on the fact that one or more signal libraries show an agglomeration of reads in these areas. The second step is a statistical evaluation of these regions of interest where both the signal and the background libraries are taken into account. The two peak callers CLIPper and Piranha can perform a statistical analysis on just signal libraries and report a p-value for the peaks they find. The block-based method can only perform the first step of finding read-enriched areas and relies on other programs, i.e. DESeq2, to do the statistical analysis. If no replicate or control samples are available, CLIPper's or Piranha's built-in functionality to estimate the background distribution is obviously the only possibility to assign significances to peaks in the read profiles. In theory this is also pos-

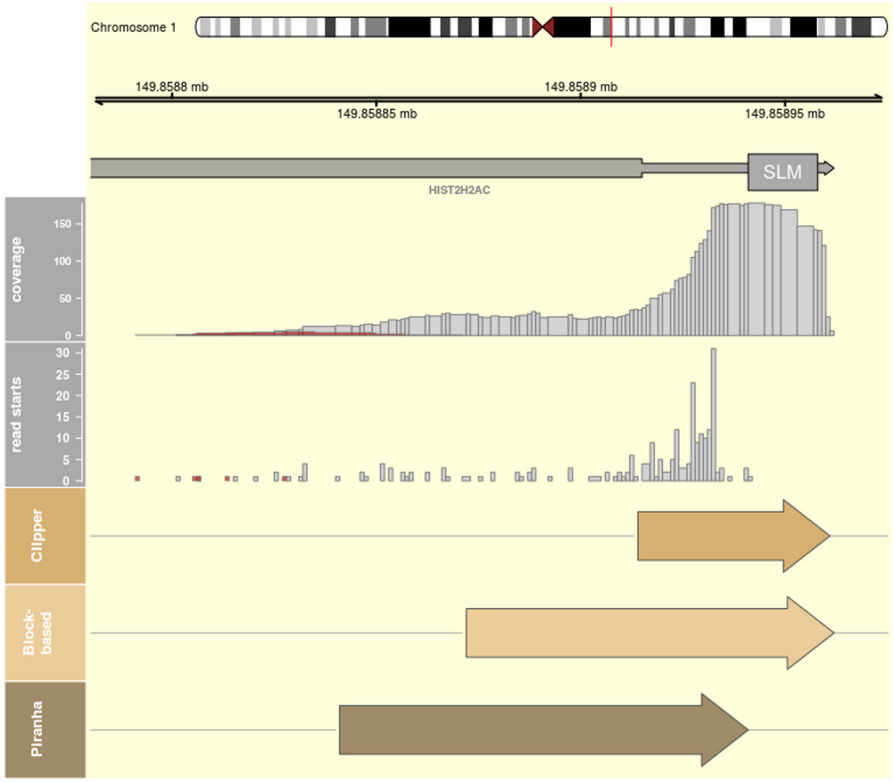


Fig. 3. Comparison of called peaks on data stemming from an eCLIP experiment of the RBP SLBP. Tracks from top to bottom. (1) and (2) location on chromosome 1. (3) gene HIST2H2AC (thick) with 3' UTR (thin) and stem-loop motif region (SLM), (4) read coverage (SMI coverage in red), (5) read start coverage (SMI coverage in red), (6) peaks called by CLIPper, (7) peaks called by extended blockbuster, (8) peaks called by Piranha. The figure was generated in the R environment [56] with gviz [57].

sible for the block-based method, but has not yet been implemented. Piranha offers the possibility to add covariate datasets to improve estimation of the background. CLIPper itself does not offer this capability, yet the significance of the regions can be reevaluated with the scripts used in the eCLIP pipeline of the ENCODE consortium [51]. Both Piranha and CLIPper cannot handle replicate samples which are essential for the estimation of the technical or biological variance in the experiments, which is why splitting up the two tasks as mentioned above is advised when replicates are available. In the following paragraphs the three tools Piranha, CLIPper, and blockbuster based peak calling are compared with each other on one example dataset to illustrate similarities and some distinguishing features between the tools.

For this example the human RBP Histone Stem-Loop-Binding Protein (SLBP) was chosen. An eCLIP experiment [22] was recently published for this RBP and is available on the ENCODE consortium website.[4] For the analysis we utilized the files that provide the already mapped reads to reference genome hg19. The second-in-pair reads which should contain the cross link position at their 5' ends have an average length of approximately 38 nucleotides. SLBP targets histone protein mRNAs and has a well known stem-loop binding motif [55]. One target of SLBP are transcripts of HIST2H2AC with the aforementioned stem-loop motif in its 3' UTR. In Fig. 3 eCLIP read profiles for this one target site are depicted. The 3' end of gene HIST2H2AC lies in the region 149.858800 mb – 149.858910 mb (see Fig. 3 tracks 1–3). The read coverage and the read start coverage (tracks 4,5) are the determining signals for the three different peak callers CLIPper (track 6), extended blockbuster (track 7) and Piranha (track 8). Comparing the coverage tracks with its size matched input counterparts for this region (track 4,5 red), it can be safely stated that this region is targeted by SLBP. This one example already clearly illustrates some of the three tools' major differences. Where CLIPper and the block-based approach follow the overall read coverage, Piranha is more aligned with the read start distribution. It should be noted that in this example the stem-loop motif starts right after the peak that was called by Piranha (a more detailed discussion of this issue can be found in the Conclusion section).

As stated above, a fair comparison is difficult to achieve when the tools are very flexible with different parameter settings. For the following more general analysis the tools were called with standard parameters where possible. The other parameter settings are best guesses as a thorough evaluation of these settings is beyond the scope of this review. To achieve an even higher parity in the evaluation of the peak callers, the normalization and the statistical analysis were done with the same pipeline. For CLIPper the ENCODE consortium offers peak files in a BED-like format where the signal library is normalized with a size matched input library (SMI). The peak boundaries in these BED files were taken as input for the second step of the peak analysis: counting the reads of the signal and the input library that fall into each peak region and evaluating the fold change in the region with DESeq2. Piranha was called with the signal library only to define the peak boundaries and the normalization with the SMI was done thereafter with the same pipeline as for CLIPper and the block-based approach. Piranha can be used with covariates that should normalize the results, but the results of this analysis did not allow for the filtering steps that were applied afterwards as the output of Piranha in this mode

[4] Datasets available at https://www.encodeproject.org/experiments/ENCSR483NOP/.

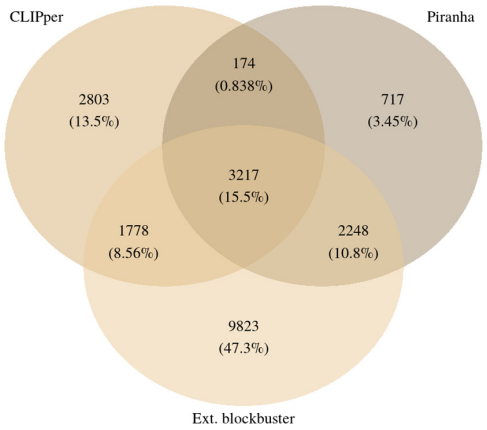


Fig. 4. Venn diagram of genomic positions contained in peak regions defined by the three peak callers.

Table 1
General statistics of called peaks for the three methods (in brackets: adjacent peaks not merged).

Tools	Number of peaks	Average peak length	Positions in peaks
Piranha	116 (312)	53.79 (20.00)	6240
CLIPper	135 (180)	58.05 (43.54)	7837
Block-based	146 (180)	115.89 (97.97)	17635

was not verbose enough. Furthermore it has to be mentioned that Piranha was called with a bin size of 20 (-z 20) and the merging of bins was disabled (-u 0). Piranha calculates p-values for merged bins counter-intuitively such that results with merged bins were inconsistent because the implicit output filtering of peaks relies on these p-values. In the block-based approach blockbuster was called with a minimum block height of 10 (-minBlockHeight 10), the blocks were extended as described and the resulting regions were again evaluated with DESeq2. Afterwards the identified peak regions were filtered such that only those peaks were kept that had a normalized fold change of at least 2 when comparing signal library to SMI.

The genomic position based overlap between the different peak callers is depicted in Fig. 4 and the overall distribution of the peaks determined by the different tools is shown in Table 1. The block-based approach is the most inclusive as it generates the biggest total number of positions in peaks. The number of peaks is quite similar for all three peak callers with the block-based approach generating the longest peaks. Piranha generates many small peaks that are adjacent to each other as expected due to disabled bin merging. Only 11.3 % of peak positions generated by Piranha are exclusive to that tool, for CLIPper and extended blockbuster this percentage is much higher (35.2 % and 57.6 % respectively). This shows that there are significant differences between the tools and it might be worth applying the different tools to the same dataset to find significant regions and subsequently motifs. In any case, an in-depth knowledge of the tools is advisable and the most appropriate tools should be chosen based on the given wet-lab protocol. Table 2 gives an overview of the three tools, addressing strengths, weaknesses and some general observations we gathered during this analysis.

3.3. Postprocessing

The purpose of peak calling is to reduce the false positive rate and provide a set of high affinity binding sites. Albeit peak calling corrects for differences in expression levels to some extent, the results of a CLIP-seq experiment will still be highly dependent on the expression state of the cells in which the experiment was performed. This implies that the problem of false negatives remains since binding sites in lowly expressed genes or genes that are not expressed at all cannot be detected in a CLIP-seq experiment. Consider Fig. 5, where we display the read starts of a CLIP-seq experiment on an artificial genomic locus. Due to unspecific binding, reads can be detected outside of true binding sites. Most of these reads are discarded by peak calling. However, the false negatives, i.e. the binding sites which are not covered by reads from the experiment, cannot be found by the analysis described so far. This is a problem when using published CLIP-seq data to analyse cell lines or tissues different from the cell lines that were used to produce the CLIP-seq data. Even for same cell lines there can be considerable variances in expression profiles and thus also differences in recovered binding sites. To give one example, Maticzka et al. [58] re-analysed data from an AGO knockdown by Schmitter et al. [59] using more recent CLIP-seq data [29]. Schmitter et al. showed that genes up-regulated in an AGO knockdown are enriched with putative miRNA-binding sites, consistent with a direct regulation by miRNAs in the wild type. However, one may be inclined to perform an analysis using published CLIP-seq data from the same cell line (which exists), instead of *in silico* seed-based miRNA-binding site prediction, as it was done by Schmitter et al. in the original publication. Surprisingly, Maticzka et al. showed that CLIP binding sites are not enriched in the up-regulated genes, probably due to the low expression of the miRNA-regulated genes in the wild type.

Another example is the work in [60], which shows that publicly available data can be more or less useless (or even harmful by leading to wrong biological conclusions) when only the peak-called sites are used. The group was studying the tumor suppressor *ANXA7*, which is alternatively spliced in glioblastoma compared

Table 2
Observed assets and drawbacks of the described CLIP-seq peak callers.

Tools	Pros	Cons	Observations
Piranha			
	• models background • fast	• p-value for merged bin counterintuitive • fixed bin width • no replicates	• calls peaks on read starts • takes read ends instead of starts for minus strand
CLIPper			
	• models background • dynamic peak width	• slow • needs specific annotation • no replicates	• calls peaks only on known transcripts • broad peaks
Block-based			
	• dynamic peak width • fast • supports replicates	• does not model background	• relies on blockbuster and DESeq2 • broad peaks • peaks can overlap

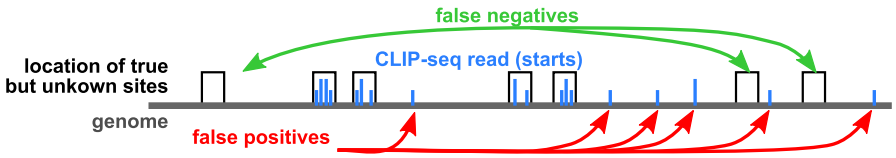


Fig. 5. False positives and negatives for a CLIP-seq experiment with respect to true but unknown binding sites.

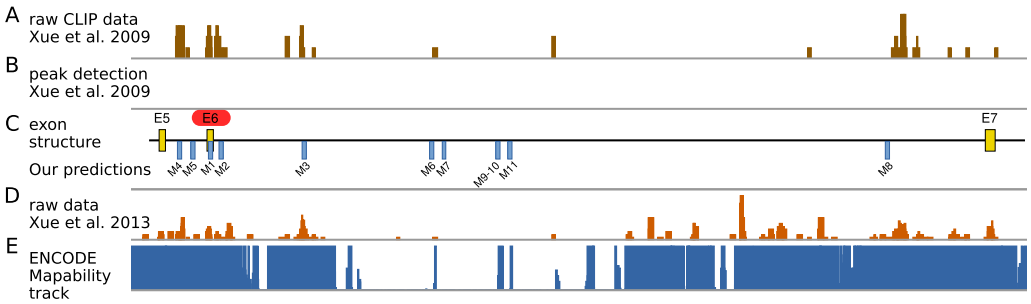


Fig. 6. Exon structure of *ANXA7* with the alternative exon E6, which is differentially spliced in glioblastoma. Since exon E6 is repressed by PTBP1, one would expect binding sites of PTBP1 to the left and right of exon E6 [61]. Albeit the raw data from the publicly available CLIP-seq experiment [61] shows some reads in that region (A), no binding sites from the CLIP-seq experiment (as determined by peak calling) can be found (B). We predicted ten binding sites with GraphProt (C). Nine out of the ten predicted sites could be validated by mutation experiments [60]. Track (D) shows the raw read data of a newer CLIP-seq experiment [62]. Some reads accumulate around our predicted binding sites. However, as shown in the mappability track (E), the predicted sites M6,M7, M9-10 and M11 cannot be identified by the CLIP-seq experiment since the reads cannot be uniquely mapped in that region.

to normal tissue. They did several experiments to show that an RNA-binding protein, namely the splice factor PTBP1, is involved. Firstly, they showed that *ANXA7* is alternatively spliced. Secondly, they searched for differentially expressed splice factors (again between glioblastoma and normal tissue) and showed that PTBP1 is the only such factor. Thirdly, they did an RNA immunoprecipitation with PTBP1, finding that PTBP1 coprecipitates *ANXA7* RNA. The final step would have been to determine binding sites from a publicly available CLIP-seq dataset, which exists [61]. In this publication a set of binding sites was determined by peak calling. However, as shown in Fig. 6B, there are no called binding sites in the vicinity of the alternatively spliced gene, which would lead one to wrongly conclude that there are no binding sites in this transcript region.

Thus, to overcome these kinds of problems and to make publicly available CLIP-seq data usable for a wider community, one has to predict these missing binding sites. Of course, these predictions have to be accompanied by additional experimental approaches to verify them. The general approach for predicting binding sites is to learn a model from the sites detected by a CLIP-seq experiment, and to use this model to determine missing binding sites. In the following we will focus on two approaches for binding site identification most commonly used in CLIP-seq data analysis.

3.3.1. Affinity-based approaches

The first types are affinity-based approaches, which try to learn a model that estimates the affinity of the RNA-binding protein P for a specific sequence s. In more detail, consider the binding reaction of a protein P to an RNA sequence s at equilibrium. Then the affinity can be determined by

$$K_a(s) = \frac{[P-s]}{[P][s]} = \frac{k_{on}}{k_{off}} = e^{-\Delta G/RT} \tag{1}$$

where k_{on} (resp. k_{off}) is the rate of association (resp. dissociation), and ΔG is the free energy of binding. $[P-s]$, $[P]$ and $[s]$ are the concentrations of the protein-sequence complex, the protein, and the sequence, respectively. Now given a set of sequences $\{s_1, \ldots, s_n\}$ that are bound by P, let $\{[P-s_1], \ldots, [P-s_n]\}$ be the associated counts indicating how often the sequence s_i occurs as a binding site of protein P. The purpose of motif finding tools is to determine parameters Θ for their models such that the associated score $S_\Theta(s)$ for a sequence s is a good estimate for the affinity, i.e. that $S_\Theta(s) \approx K_a(s)$. If we had enough data and knew the concentration $[s]$ of unbound s for each sequence, then the following score

$$S_\Theta(s) = \frac{[P-s_i]}{[s_i]\sum_{j=1}^{n}[P-s_j]}$$

provides such an estimate for the (relative) affinity. However, there are two caveats. First, $[s_i]$ is usually unknown, and is thus often estimated from the background distribution of sequences. Secondly, datasets are usually too small to provide a reliable estimate for $S_\Theta(s)$ for all sequences s. Hence, these scores are often approximated by assuming fixed-length motifs and independent contributions of each position. This basically assumes that the free energy contribution for each base is additive. Since the affinity is related to the free energy as described by equation (1), additivity in the free energy contribution translates to multiplicity in the score. Examples of these types of models are position weight matrices (PWM) [63], as used by the popular MEME tool [64], or position-specific affinity matrices (PSAM) [65].

Given fixed-length motifs, the question is how to score binding sites that are longer than the motif size. Early approaches used the sum of the different subsequences, however, this does not take the concentration of the protein and the effect of binding on the concentration into account. A better approach is to model the occupancy of the sequence by the protein. For a sequence s, the occupancy $N(s)$ is the probability that s is bound by P:

$$N(s) = \frac{[P-s]}{[P-s]+[s]} = \frac{[P-s]\frac{[P]}{[P-s]}}{[P-s]\frac{[P]}{[P-s]}+[s]\frac{[P]}{[P-s]}} = \frac{[P]}{[P]+K_d(s)},$$

where $K_d(s) = K_a(s)^{-1}$ is the dissociation constant. Assuming that the protein concentration $[P]$ is small compared to $K_d(s)$ to ensure an efficient regulatory scheme [66], one yields

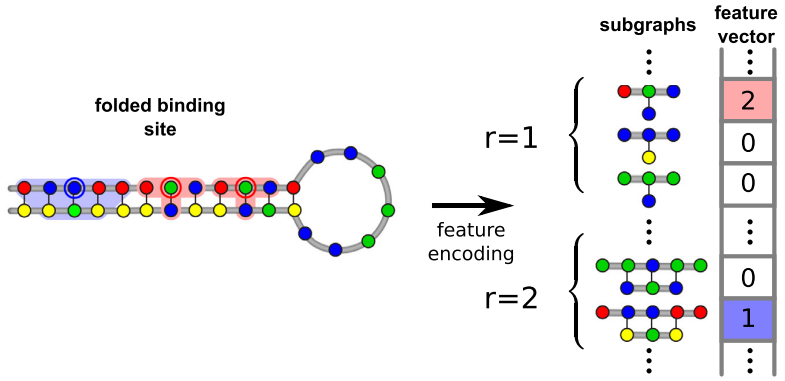


Fig. 7. A folded binding site in graph representation (left) and its associated feature vector (right). The red shaded areas on the left indicate two subgraphs of radius (r) 1 with the centre indicated by a red circle (two occurences). The blue shaded area is an example of a subgraph with r = 2 (one occurence). Again, the blue circle indicates the central node of the subgraph (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.).

$$N(s) \approx \frac{[P]}{K_d(s)} = [P]K_a(s).$$

Thus, for the small k-mers recognized by the models explained above, the occupancy can be estimated from the score $S_\Theta(s) \approx K_a(s)$. For larger sequences, one can determine the occupancy as the probability that at least one k-mer of the sequence binds. This can be done using a "noisy OR" function [67] by calculating this probability as 1 minus the probability that none of the k-mers bind. RNAcontext [68] is a recent approach for learning sequence and structure preferences for RNA-binding proteins which directly estimates the occupancy of the k-mers using a logistic regression formulation of the occupancy term.

However, it is already known that pure sequence-based models are not good for modeling binding sites on RNAs due to the disregard of secondary structure. Examples of early models that take secondary structure into account are BioBayesNet [69] for DNA and MEMERIS [70] for RNA. More recently, RNAcontext presented a more integrated approach, where the occupancy for a k-mer s is determined by taking a sequence contribution $N^{seq}(s)$, which is interpreted as the occupancy of the sequence s in its optimal context, and multiplying it with a contribution of the structural context $N^{struct}(s, p)$. Here, p is the matrix that assigns to each position a distribution of possible structural states such as being in a hairpin, internal or multi loop, or being in a stem. p is calculated from s using SFOLD [71].

3.3.2. Classification- and regression-based approaches

Another type of approach is not based on a physical model but considers the problem of determining binding sites as a classification or regression problem. As a classification task, in contrast to the previous approaches, one needs a positive set (i.e. the regions determined by the peak caller) and a negative set, which are sites that are *not* bound by protein. Since the latter is usually not available, the set of negative instances has to be generated, e.g. by shuffling the true regions on the genome. The idea is to determine features that differentiate the binding sites from the non-binding sites. Oversimplifying, when using k-mers, one would try to determine k-mers that are highly enriched in the positive data and depleted in the negative data. However, a simple k-mer approach would not work due to the complexity of the task. Instead, advanced machine learning approaches have to be used.

One example for such an approach is GraphProt [58], which uses sequence- and structure-based features for that purpose. The binding site together with a collection of different near optimal foldings is encoded as a graph. Then, GraphProt considers small subgraphs that are determined by two different parameters, namely radius r and distance d, as features. Starting from each node of the graph, the radius defines how many edges can be visited to determine the subgraph. The distance parameter d includes as features all possible pairs of subgraphs determined by the radius r that have an edge distance of exactly d. Thus, these features can be considered as the upgrade from sequence k-mers with gaps to graphs. The number of occurrences of these subgraphs are stored into a huge but sparse feature vector (see Fig. 7). These feature vectors for each binding and non-binding site are then used by a support vector machine as input to discriminate positive from negative sites. If quantitative binding data is available, support vector regression instead of support vector classification can be used.

Another example for a regression-based approach is iONMF [72], which uses orthogonal matrix factorization to determine a model for the strength of binding sites. iONMF basically uses as features the probability of each position around the binding site to be double-stranded, the number of occurrences for all possible 4-mers in a region around the binding sites, the region type, the GO annotation of the RNA, and the CLIP-seq counts for a collection of proteins different from the one investigated as possible features. The idea for training a model is to determine a coefficient matrix for a linear regression task. I.e. multiplying these coefficients with the values for the features listed above should approximate the CLIP-seq counts of the actual experiment as well as possible, using the determined values for all features. Once this is achieved, new binding sites can be scored by determining the feature values and multiplying them with the coefficients. However, due to the large number of features, one would immediately run into overfitting problems. Omitting a lot of details, iONMF introduces a new approach for orthogonal matrix factorization. The idea is to yield a low-rank approximation of the feature matrices by determining modular projection of the original data matrices, yielding an effective regularization by avoiding multicollinearity between feature vectors.

4. Conclusion

CLIP-seq is currently one of the most important means to determine binding sites of RNA-binding proteins on a genome-wide level. Since peak height alone is not a good measure of significance, we advise preparing signal and background CLIP-seq libraries in replicates. This enables highly specific removal of background noise from the signal data under application of statistical modeling.

The computational analysis of CLIP-seq data requires three steps, which have to be adapted to the specificities of the CLIP protocols to different extents. The first and most protocol-specific step is the preprocessing of the raw data. Sequenced reads have to be trimmed and mapped to the genome or the transcriptome. What exactly has to be trimmed depends on the adapter sequences, as well as barcode sequences for PCR duplicate removal and demultiplexing. For the mapping part, several widely-applied tools exist which can also handle splice-sensitive mapping.

The second, and one of the most important steps, is peak calling, which determines high confidence binding sites by removing signals corresponding to unspecific binding. Here, both protocol-specific and generic peak callers exist. However, as shown in the comparison of different peak callers, results can vary drastically. This can even hold within individual tools when they are run using slightly different parameter settings. Thus, depending on the data, it might be worth to apply and compare the results of different peak calling techniques.

In the example case (see Fig. 3) Piranha did not include the actual binding motif, which forms a stem-loop structure recognized by SLBP. Instead, bins get called in the upstream vicinity where most read starts occur. This is expected and not a flaw of the algorithm, since Piranha only takes read starts into account. It is known that double-stranded regions are less efficiently crosslinked in CLIP, which would explain the upstream accumulation of crosslink events. On the other hand, transcriptome-wide RBP binding preferences, whether sequence- or structure-dependent or both, are usually not known in advance, and thus one has to rely on the called sites to extract these preferences. Clearly, one may extend the called sites to include more nucleotides, but this can increase the amount of noise and other potentially (non-) RBP specific motifs returned by the analysis. All tested tools can be a reasonable choice, depending on the CLIP-seq protocol, but one should keep in mind their assets and drawbacks (Table 2). For a comprehensive analysis, we recommend trying more than one peak caller, especially if control experiments and replicates are present, which should become standard in future CLIP-seq experiments. A compound strategy, where the steps of site definition and their statistical evaluation are split between programs, could further improve results. Newly developed peak callers should combine the aforementioned strengths of the described programs. In addition, a more thorough study with true positive sets for RBPs targeting structure and sequence features could help to answer the question of which peak caller is suitable in which scenario.

In the last step, which is more or less protocol-independent, motifs are determined and binding models are inferred from the regions identified by the peak caller. The importance of this step is currently largely underestimated. However, without training binding models, published CLIP-seq data can hardly be utilized as they are. In the worst case, the direct use of regions identified in CLIP-seq data on different cells/ conditions can lead to wrong conclusions concerning the underlying regulatory mechanisms. The reason is simply that a CLIP-seq experiment is expression-dependent, and binding sites in lowly or not expressed genes are not discovered. If an RNA is expressed in the currently investigated cell type but not in the cell type used for the original CLIP-seq experiment, then binding models can be applied to determine potential missing binding sites. Utilizing these prediction approaches in combination with validation experiments can therefore largely extend the explanatory power of CLIP-seq datasets.

5. Funding

This work was funded by the Baden-Württemberg-Stiftung (BWST_NCRNA_008), the German Research Foundation (DFG Grant BA2168/11-1 SPP 1738) and the BMBF Verbundprojekt Deutsches Netzwerk für Bioinformatik-Infrastruktur (de.NBI).

Acknowledgments

We thank Daniel Maticzka and Philip Uren for their valuable comments on the manuscript.

References

[1] V. Kevin Morris, John S. Mattick, The rise of regulatory RNA, Nat. Rev. Genet. 15 (6) (2014) 423–437.
[2] Alexander G. Baltz, Mathias Munschauer, Björn Schwanhäusser, Alexandra Vasile, Yasuhiro Murakawa, Markus Schueler, Noah Youngs, Duncan Penfold-Brown, Kevin Drew, Miha Milek, Emanuel Wyler, Richard Bonneau, Matthias Selbach, Christoph Dieterich, Markus Landthaler, The mRNA-bound proteome and its global occupancy profile on protein-coding transcripts, Mol. Cell 46 (5) (2012) 674–690.
[3] Alfredo Castello, Bernd Fischer, Katrin Eichelbaum, Rastislav Horos, Benedikt M. Beckmann, Claudia Strein, Norman E. Davey, David T. Humphreys, Thomas Preiss, Lars M. Steinmetz, Jeroen Krijgsveld, Matthias W. Hentze, Insights into RNA biology from an atlas of mammalian mRNA-binding proteins, Cell 149 (6) (2012) 1393–1406.
[4] Debashish Ray, Hilal Kazan, Kate B. Cook, Matthew T. Weirauch, Hamed S. Najafabadi, Xiao Li, Serge Gueroussov, Mihai Albu, Hong Zheng, Ally Yang, Hong Na, Manuel Irimia, Leah H. Matzat, Ryan K. Dale, Sarah A. Smith, Christopher A. Yarosh, Seth M. Kelly, Behnam Nabet, Desirea Mecenas, Weimin Li, Rakesh S. Laishram, Mei Qiao, Howard D. Lipshitz, Fabio Piano, Anita H. Corbett, Russ P. Carstens, Brendan J. Frey, Richard A. Anderson, Kristen W. Lynch, Luiz O.F. Penalva, Elissa P. Lei, Andrew G. Fraser, Benjamin J. Blencowe, Quaid D. Morris, Timothy R. Hughes, A compendium of RNA-binding motifs for decoding gene regulation, Nature 499 (7457) (2013) 172–177.
[5] Stefanie Gerstberger, Markus Hafner, Thomas Tuschl, A census of human RNA-binding proteins, Nat. Rev. Genet. 15 (12) (December 2014) 829–845.
[6] Stefanie Gerstberger, Markus Hafner, Manuel Ascano, Thomas Tuschl, Evolutionary conservation and expression of human RNA-binding proteins and their role in human genetic disease, Adv. Exp. Med. Biol. 825 (2014) 1–55.
[7] Silvia Carolina Lenzken, Tilmann Achsel, Maria Teresa Carrı, Silvia M.L. Barabino, Neuronal RNA-binding proteins in health and disease, Wiley Interdiscip. Rev. RNA 5 (4) (2014) 565–576.
[8] Aditi Gupta, Gupta Aditi, Gribskov Michael, The role of RNA sequence and structure in RNA–Protein interactions, J. Mol. Biol. 409 (4) (2011) 574–587.
[9] Eckhard Jankowsky, Michael E. Harris, Specificity and nonspecificity in RNA-protein interactions, Nat. Rev. Mol. Cell Biol. 16 (9) (2015) 533–544.
[10] Yoichiro Sugimoto, Alessandra Vigilante, Elodie Darbo, Alexandra Zirra, Cristina Militti, Andrea D'Ambrogio, Nicholas M. Luscombe, Jernej Ule, hiCLIP reveals the in vivo atlas of mRNA secondary structures recognized by staufen 1, Nature 519 (7544) (2015) 491–494.
[11] Yasuhiro Murakawa, Murakawa Yasuhiro, Hinz Michael, Mothes Janina, Schuetz Anja, Uhl Michael, Wyler Emanuel, Yasuda Tomoharu, Mastrobuoni Guido, Caroline C. Friedel, Dölken Lars, Kempa Stefan, Schmidt-Supprian Marc, Blüthgen Nils, Backofen Rolf, Heinemann Udo, Wolf Jana, Scheidereit Claus, Landthaler Markus, RC3H1 post-transcriptionally regulates A20 mRNA and modulates the activity of the IKK/NF-κB pathway, Nat. Commun. 6 (2015) 7367.
[12] Ibrahim Avsar Ilik, Jeffrey J. Quinn, Plamen Georgiev, Filipe Tavares-Cadete, Daniel Maticzka, Sarah Toscano, Yue Wan, Robert C. Spitale, Nicholas Luscombe, Rolf Backofen, Howard Y. Chang, Asifa Akhtar, Tandem stem-loops in rox RNAs act together to mediate X chromosome dosage compensation in drosophila, Mol. Cell 51 (2) (2013) 156–173.
[13] Xiao Li, Hilal Kazan, Howard D. Lipshitz, Quaid D. Morris, Finding the target sites of RNA-binding proteins, Wiley Interdiscip. Rev. RNA 5 (1) (January 2014) 111–130.
[14] Donny D. Licatalosi, Aldo Mele, John J. Fak, Jernej Ule, Melis Kayikci, Sung Wook Chi, Tyson A. Clark, Anthony C. Schweitzer, John E. Blume, Xuning Wang, Jennifer C. Darnell, Robert B. Darnell, HITS-CLIP yields genome-wide insights into brain alternative RNA processing, Nature 456 (7221) (2008) 464–469.
[15] Valentine Murigneux, Jérôme Saulière, Hugues Roest Crollius, Hervé Le Hir, Transcriptome-wide identification of RNA binding sites by CLIP-seq, Methods 63 (1) (2013) 32–40.
[16] Tao Wang, Guanghua Xiao, Yongjun Chu, Michael Q. Zhang, David R. Corey, Yang Xie, Design and bioinformatics analysis of genome-wide CLIP experiments, Nucleic Acids Res. 43 (11) (2015) 5263–5274.
[17] Eric L. Van Nostrand, Stephanie C. Huelga, Gene W. Yeo, Experimental and computational considerations in the study of RNA-Binding Protein-RNA interactions, Adv. Exp. Med. Biol. 907 (2016) 1–28.
[18] Jernej Ule, Kirk B. Jensen, Matteo Ruggiu, Aldo Mele, Aljaž Ule, Robert B. Darnell, Clip identifies nova-regulated rna networks in the brain, Science 302 (5648) (2003) 1212–1215.
[19] Jernej Ule, Kirk Jensen, Aldo Mele, Robert B. Darnell, Clip: a method for identifying protein–rna interaction sites in living cells, Methods 37 (4) (2005) 376–386.

[20] Markus Hafner, Markus Landthaler, Lukas Burger, Mohsen Khorshid, Jean Hausser, Philipp Berninger, Andrea Rothballer, Manuel Ascano Jr., Anna-Carina Jungkamp, Mathias Munschauer, Alexander Ulrich, Greg S. Wardle, Scott Dewell, Mihaela Zavolan, Thomas Tuschl, Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP, Cell 141 (1) (2010) 129–141.

[21] Julian König, Kathi Zarnack, Gregor Rot, Tomaz Curk, Melis Kayikci, Blaz Zupan, Daniel J. Turner, Nicholas M. Luscombe, Jernej Ule, iCLIP reveals the function of hnRNP particles in splicing at individual nucleotide resolution, Nat. Struct. Mol. Biol. 17 (7) (2010) 909–915.

[22] Eric L. Van Nostrand, Gabriel A. Pratt, Alexander A. Shishkin, Chelsea Gelboin-Burkhart, Mark Y. Fang, Balaji Sundararaman, Steven M. Blue, Thai B. Nguyen, Christine Surka, Keri Elkins, Rebecca Stanton, Frank Rigo, Mitchell Guttman, Gene W. Yeo, Robust transcriptome-wide discovery of RNA-binding protein binding sites with enhanced CLIP (eCLIP), Nat. Methods 13 (6) (2016) 508–514.

[23] Brian J. Zarnegar, Ryan A. Flynn, Ying Shen, Brian T. Do, Howard Y. Chang, Paul A. Khavari, irCLIP platform for efficient characterization of protein-RNA interactions, Nat. Methods 13 (6) (2016) 489–492.

[24] Georges Martin, Mihaela Zavolan, Redesigning CLIP for efficiency, accuracy and speed, Nat. Methods 13 (6) (2016) 482–483.

[25] Nazmul Haque, J. Robert Hogg, Easier, better, faster, stronger: Improved methods for RNA-Protein interaction studies, Mol. Cell 62 (5) (2016) 650–651.

[26] Aleksandra Helwak, Grzegorz Kudla, Tatiana Dudnakova, David Tollervey, Mapping the human miRNA interactome by CLASH reveals frequent noncanonical binding, Cell 153 (3) (2013) 654–665.

[27] Michael J. Moore, Troels K.H. Scheel, Joseph M. Luna, Christopher Y. Park, John J. Fak, Nishiuchi Eiko, Charles M. Rice, Robert B. Darnell, miRNA–target chimeras reveal miRNA 3'-end pairing as a major determinant of argonaute target specificity, Nat. Commun. 6 (2015) 8864.

[28] Stephen M. Testa, Matthew D. Disney, Douglas H. Turner, Kierzek Ryszard, Thermodynamics of RNA-RNA duplexes with 2- or 4-thiouridines: Implications for antisense design and targeting a group I intron, Biochemistry 38 (50) (1999) 16655–16662.

[29] Shivendra Kishore, Lukasz Jaskiewicz, Lukas Burger, Jean Hausser, Mohsen Khorshid, Mihaela Zavolan, A quantitative analysis of CLIP methods for identifying binding sites of RNA-binding proteins, Nat. Methods 8 (7) (July 2011) 559–564.

[30] Anna-Carina Jungkamp, Marlon Stoeckius, Desirea Mecenas, Dominic Grün, Guido Mastrobuoni, Stefan Kempa, Nikolaus Rajewsky, In vivo and transcriptome-wide identification of RNA binding protein target sites, Mol. Cell 44 (5) (2011) 828–840.

[31] Yoichiro Sugimoto, Julian König, Shobbir Hussain, Blaž Zupan, Tomaž Curk, Michaela Frye, Jernej Ule, Analysis of CLIP and iCLIP methods for nucleotide-resolution studies of protein-RNA interactions, Genome Biol. 13 (8) (2012) R67.

[32] Michael T. Lovci, Dana Ghanem, Henry Marr, Justin Arnold, Sherry Gee, Marilyn Parra, Tiffany Y. Liang, Thomas J. Stark, Lauren T. Gehman, Shawn Hoon, Katlin B. Massirer, Gabriel A. Pratt, Douglas L. Black, Joe W. Gray, John G. Conboy, Gene W. Yeo, Rbfox proteins regulate alternative mRNA splicing through evolutionarily conserved RNA bridges, Nat. Struct. Mol. Biol. 20 (12) (2013) 1434–1442.

[33] Yu-Cheng T. Yang, Chao Di, Boqin Hu, Meifeng Zhou, Yifang Liu, Nanxi Song, Yang Li, Jumpei Umetsu, Zhi Lu, CLIPdb: a CLIP-seq database for protein-RNA interactions, BMC Genomics 16 (1) (2015) 51.

[34] Matthias Dodt, Johannes Roehr, Rina Ahmed, Christoph Dieterich, FLEXBAR-flexible barcode and adapter processing for next-generation sequencing platforms, Biology 1 (3) (Dec 2012) 895–905.

[35] Haibin Xu, Xiang Luo, Jun Qian, Xiaohui Pang, Jingyuan Song, Guangrui Qian, Jinhui Chen, Shilin Chen, Fastuniq: a fast de novo duplicates removal tool for paired short reads, PLOS ONE 7 (12) (2012) 1–6.

[36] Marcel Martin, Cutadapt removes adapter sequences from high-throughput sequencing reads, EMBnet J. 17 (1) (2011).

[37] Anthony M. Bolger, Marc Lohse, Bjoern Usadel. Trimmomatic: a flexible trimmer for illumina sequence data. Bioinformatics, page btu170, 2014.

[38] Cole Trapnell, Lior Pachter, Steven L Salzberg, Tophat: discovering splice junctions with rna-seq, Bioinformatics 25 (9) (2009) 1105–1111.

[39] Thomas D. Wu, Serban Nacu, Fast and snp-tolerant detection of complex variants and splicing in short reads, Bioinformatics 26 (7) (2010) 873–881.

[40] Steve Hoffmann, Christian Otto, Stefan Kurtz, Cynthia M. Sharma, Philipp Khaitovich, Jörg Vogel, Peter F. Stadler, Jörg Hackermüller, Fast mapping of short sequences with mismatches, insertions and deletions using index structures, PLoS Comput. Biol. 5 (9) (2009) e1000502.

[41] A. Dobin, C.A. Davis, F. Schlesinger, J Drenkow, C. Zaleski, S. Jha, P. Batut, M. Chaisson, T.R Gingeras, STAR: ultrafast universal RNA-seq aligner, Bioinformatics 29 (1) (2013) 15–21.

[42] Ayat Hatem, Doruk Bozdağ, Amanda E Toland, Ümit V Çatalyürek, Benchmarking short sequence mapping tools, BMC Bioinf. 14 (1) (2013) 184.

[43] Pär G. Engström, Tamara Steijger, Botond Sipos, Gregory R. Grant, André Kahles, Tyler Alioto, Jonas Behr, Paul Bertone, Regina Bohnert, Davide Campagna, Carrie A. Davis, Alexander Dobin, Pär G. Engström, Thomas R. Gingeras, Nick Goldman, Gregory R. Grant, Roderic Guigó, Jennifer Harrow, Tim J. Hubbard, Géraldine Jean, André Kahles, Peter Kosarev, Sheng Li, Jinze Liu, Christopher E. Mason, Vladimir Molodtsov, Zemin Ning, Hannes Ponstingl, Jan F. Prins, Gunnar Rätsch, Paolo Ribeca, Igor Seledtsov, Botond Sipos, Victor Solovyev, Tamara Steijger, Giorgio Valle, Nicola Vitulo, Kai Wang, Thomas D. Wu, Georg Zeller, Gunnar Rätsch, Nick Goldman, Tim J. Hubbard, Jennifer Harrow, Roderic Guigó, Paul Bertone, Systematic evaluation of spliced alignment programs for RNA-seq data, Nat. Methods 10 (12) (2013) 1185–1191.

[44] Giacomo Baruzzo, Katharina E. Hayer, Eun Ji Kim, Barbara Di Camillo, Garret A. FitzGerald, Gregory R. Grant, Simulation-based comprehensive benchmarking of rna-seq aligners, Nat. Methods (2016).

[45] Philip J Uren, Emad Bahrami-Samani, Suzanne C. Burns, Mei Qiao, Fedor V. Karginov, Emily Hodges, Gregory J. Hannon, Jeremy R. Sanford, Luiz O.F. Penalva, Andrew D. Smith, Site identification in high-throughput RNA-protein interaction data, Bioinformatics 28 (23) (2012) 3013–3020.

[46] David L Corcoran, Stoyan Georgiev, Neelanjan Mukherjee, Eva Gottwein, Rebecca L Skalsky, Jack D Keene, Uwe Ohler, PARalyzer: definition of RNA binding sites from PAR-CLIP short-read sequence data, Genome Biol. 12 (8) (2011) R79.

[47] Michael J. Moore, Chaolin Zhang, Emily Conn Gantman, Aldo Mele, Jennifer C. Darnell, Robert B. Darnell, Mapping argonaute and conventional RNA-binding protein interactions with RNA at single-nucleotide resolution using HITS-CLIP and CIMS analysis, Nat. Protoc. 9 (2) (February 2014) 263–293.

[48] Sebastien M. Weyn-Vanhentenryck, Aldo Mele, Qinghong Yan, Shuying Sun, Natalie Farny, Zuo Zhang, Chenghai Xue, Margaret Herre, Pamela A. Silver, Michael Q. Zhang, Adrian R. Krainer, Robert B. Darnell, Chaolin Zhang, HITS-CLIP and integrative modeling define the rbfox splicing-regulatory network linked to brain development and autism, Cell Rep. 6 (6) (2014) 1139–1152.

[49] Yoav Benjamini, Yosef Hochberg, Controlling the false discovery rate: a practical and powerful approach to multiple testing, J. R. Statist. Soc. Ser. B (Methodological) (1995) 289–300.

[50] Gabriel Pratt, Michael Lovci, Jill Moore, ppliu. clipper: release to trigger doi, 2014.

[51] Olga Botvinnik, Gabriel Pratt, Michael Lovci, ppliu, Leen, Boyko Kakaradov. gscripts: release 0.1, 2014.

[52] Erik Holmqvist, Patrick R. Wright, Lei Li, Thorsten Bischler, Lars Barquist, Richard Reinhardt, Rolf Backofen, Jörg Vogel, Global RNA recognition patterns of post-transcriptional regulators hfq and CsrA revealed by UV crosslinking in vivo, EMBO J. 35 (9) (2016) 991–1011.

[53] David Langenberger, Clara Bermudez-Santana, Jana Hertel, Steve Hoffmann, Philipp Khaitovich, Peter F Stadler, Evidence for human microRNA-offset RNAs in small RNA sequencing data, Bioinformatics 25 (18) (2009) 2298–2301.

[54] I. Michael Love, Wolfgang Huber, Simon Anders, Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2, Genome Biol. 15 (12) (2014) 550.

[55] Dazhi Tan, William F. Marzluff, Zbigniew Dominski, Liang Tong, Structure of histone mrna stem-loop, human stem-loop binding protein, and 3'hexo ternary complex, Science 339 (6117) (2013) 318–321.

[56] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2015.

[57] Florian Hahne, Robert, Ivanek, Visualizing Genomic Data Using Gviz and Bioconductor, Springer, New York, New York, NY, 2016. pages 335–351.

[58] Daniel Maticzka, Sita J Lange, Fabrizio Costa, Rolf Backofen, GraphProt: modeling binding preferences of RNA-binding proteins, Genome Biol. 15 (1) (2014) R17–R22.

[59] Daniela Schmitter, Jody Filkowski, Alain Sewer, Ramesh S. Pillai, Edward J. Oakeley, Mihaela Zavolan, Petr Svoboda, Witold Filipowicz, Effects of dicer and argonaute down-regulation on mRNA levels in human HEK293 cells, Nucleic Acids Res. 34 (17) (2006) 4801–4815.

[60] Roberto Ferrarese, Griffith R. 4th Harsh, Ajay K. Yadav, Eva Bug, Daniel Maticzka, Wilfried Reichardt, Stephen M. Dombrowski, Tyler E. Miller, Anie P. Masilamani, Fangping Dai, Hyunsoo Kim, Michael Hadler, Denise M. Scholtens, Irene L.Y. Yu, Jurgen Beck, Vinodh Srinivasasainagendra, Fabrizio Costa, Nicoleta Baxan, Dietmar Pfeifer, Dominik V. Elverfeldt, Rolf Backofen, Astrid Weyerbrock, Christine W. Duarte, Xiaolin He, Marco Prinz, James P. Chandler, Hannes Vogel, Arnab Chakravarti, Jeremy N. Rich, Maria S. Carro, Markus Bredel, Lineage-specific splicing of a brain-enriched alternative exon promotes glioblastoma progression, J. Clin. Invest. 124 (7) (2014) 2861–2876.

[61] Yu. Yuanchao Xue, Tongbin Wu Zhou, Tuo Zhu, Xiong Ji, Young-Soo Kwon, Chao Zhang, Gene Yeo, Douglas L. Black, Hui Sun, Fu. Xiang-Dong, Yi Zhang, Genome-wide analysis of PTB-RNA interactions reveals a strategy used by the general splicing repressor to modulate exon inclusion or skipping, Mol Cell 36 (6) (2009) 996–1006.

[62] Yuanchao Xue, Kunfu Ouyang, Yu. Jie Huang, Hong Ouyang Zhou, Hairi Li, Gang Wang, Wu. Qijia, Chaoliang Wei, Yanzhen Bi, Li Jiang, Zhiqiang Cai, Hui Sun, Kang Zhang, Yi Zhang, Ju Chen, Xiang-Dong Fu, Direct conversion of fibroblasts to neurons by reprogramming PTB-regulated microRNA circuits, Cell 152 (1–2) (2013) 82–96.

[63] Gary D. Stormo, Dna binding sites: representation and discovery, Bioinformatics 16 (1) (2000) 16–23.

[64] Timothy L. Bailey, Nadya Williams, Chris Misleh, Wilfred W. Li, Meme: discovering and analyzing dna and protein sequence motifs, Nucl. Acids Res. 34 (suppl 2) (2006) W369–W373.

[65] Barrett C. Foat, S. Sean Houshmandi, Wendy M. Olivas, Harmen J. Bussemaker, Profiling condition-specific, genome-wide regulation of mRNA stability in yeast, Proc. Natl. Acad. Sci. U.S.A. 102 (49) (2005) 17675–17680.

[66] Barrett C. Foat, Alexandre V. Morozov, Harmen J. Bussemaker, Statistical mechanical modeling of genome-wide transcription factor occupancy data by MatrixREDUCE, Bioinformatics 22 (14) (2006) e141–9.

[67] Joshua A Granek, Neil D Clarke, Explicit equilibrium modeling of transcription-factor binding and gene regulation, Genome Biol. 6 (10) (2005) R87.

[68] Hilal Kazan, Debashish Ray, Esther T. Chan, Timothy R. Hughes, Quaid Morris, RNAcontext: a new method for learning the sequence and structure binding preferences of RNA-binding proteins, PLoS Comput. Biol. 6 (2010) e1000832.
[69] Swetlana Nikolajewa, Rainer Pudimat, Michael Hiller, Matthias Platzer, Rolf Backofen, BioBayesNet: a web server for feature extraction and bayesian network modeling of biological sequence data, Nucleic Acids Res. 35 (Web Server issue) (2007) W688–93.
[70] Michael Hiller, Rainer Pudimat, Anke Busch, Rolf Backofen, Using RNA secondary structures to guide sequence motif finding towards single-stranded regions, Nucl. Acids Res. 34 (17) (2006) e117.
[71] Ye Ding, Charles E. Lawrence, A statistical sampling algorithm for RNA secondary structure prediction, Nucl. Acids Res. 31 (24) (2003) 7280–7301.
[72] Martin Stražar, Marinka Žitnik, Blaž Zupan, Jernej Ule, Tomaž Curk, Orthogonal matrix factorization enables integrative analysis of multiple RNA binding proteins, Bioinformatics 32 (10) (2016) 1527–1535.

[P2] MechRNA: prediction of lncRNA mechanisms from RNA-RNA and RNA-protein interactions

Publication:

- [P2] Alexander R. Gawronski, **Michael Uhl**, Yajia Zhang, Yen-Yi Lin, Yashar S. Niknafs, Varune R. Ramnarine, Rohit Malik, Felix Feng, Arul M. Chinnaiyan, Colin C. Collins, S. Cenk Sahinalp, and Rolf Backofen. **MechRNA: prediction of lncRNA mechanisms from RNA-RNA and RNA-protein interactions**. *Bioinformatics*, 2018.

Contributions of individual authors:

"I made an important contribution to this work, by providing the transcriptome-wide RBP binding site prediction data for 22 RNA-binding proteins, including developing approaches for identifying peaks from position-wise prediction scores and calculating p-values. I further contributed to the paper writing, together with Alexander R. Gawronski and Rolf Backofen, which also included extensive literature search on RBP and lncRNA functions from my side. Moreover, I was involved in developing the study from start to end. Alexander R Gawronski is the main contributor. He implemented the tool, created the tool results, and wrote the major part of the paper. S. Cenk Sahinalp and Rolf Backofen conceived the study. The remaining authors conducted the wet lab experiments and provided the experimental results. All authors contributed to and approved the final manuscript."

Michael Uhl

The following co-authors confirm the above-stated contributions:

Bioinformatics, 34(18), 2018, 3101–3110
doi: 10.1093/bioinformatics/bty208
Advance Access Publication Date: 3 April 2018
Original Paper

OXFORD

Sequence analysis

MechRNA: prediction of lncRNA mechanisms from RNA–RNA and RNA–protein interactions

Alexander R. Gawronski[1,*], Michael Uhl[2], Yajia Zhang[3,4], Yen-Yi Lin[1,5], Yashar S. Niknafs[6], Varune R. Ramnarine[5], Rohit Malik[6,†], Felix Feng[6,7,‡], Arul M. Chinnaiyan[3,4,6,8], Colin C. Collins[5], S. Cenk Sahinalp[5,9,*] and Rolf Backofen[2,*]

[1]Computing Science, Simon Fraser University, Burnaby BC V5A 1S6, Canada, [2]Centre for Biological Signalling Studies, University of Freiburg, Freiburg im Breisgau 79104, Germany, [3]Department of Pathology, University of Michigan, Ann Arbor, MI 48109, USA, [4]Department of Computational Medicine and Bioinformatics, Ann Arbor, MI 48109, USA, [5]Vancouver Prostate Centre, Vancouver, BC V6H 3Z6, Canada, [6]Michigan Center for Translational Pathology, [7]Department of Radiation Oncology and [8]Howard Hughes Medical Institute, University of Michigan, Ann Arbor, MI 48109, USA and [9]Department of Computer Science, Indiana University, Bloomington 47405, USA

*To whom correspondence should be addressed.
[†]Present address: Bristol-Myers Squibb Co, Princeton, NJ 08543, USA
[‡]Present address: Departments of Radiation Oncology, Urology, and Medicine, UCSF, San Francisco, CA 94115, USA
Associate Editor: Inanc Birol

Received on January 24, 2018; revised on March 14, 2018; editorial decision on March 23, 2018; accepted on March 27, 2018

Abstract

Motivation: Long non-coding RNAs (lncRNAs) are defined as transcripts longer than 200 nt that do not get translated into proteins. Often these transcripts are processed (spliced, capped and polyadenylated) and some are known to have important biological functions. However, most lncRNAs have unknown or poorly understood functions. Nevertheless, because of their potential role in cancer, lncRNAs are receiving a lot of attention, and the need for computational tools to predict their possible mechanisms of action is more than ever. Fundamentally, most of the known lncRNA mechanisms involve RNA–RNA and/or RNA–protein interactions. Through accurate predictions of each kind of interaction and integration of these predictions, it is possible to elucidate potential mechanisms for a given lncRNA.

Results: Here, we introduce MechRNA, a pipeline for corroborating RNA–RNA interaction prediction and protein binding prediction for identifying possible lncRNA mechanisms involving specific targets or on a transcriptome-wide scale. The first stage uses a version of IntaRNA2 with added functionality for efficient prediction of RNA–RNA interactions with very long input sequences, allowing for large-scale analysis of lncRNA interactions with little or no loss of optimality. The second stage integrates protein binding information pre-computed by GraphProt, for both the lncRNA and the target. The final stage involves inferring the most likely mechanism for each lncRNA/target pair. This is achieved by generating candidate mechanisms from the predicted interactions, the relative locations of these interactions and correlation data, followed by selection of the most likely mechanistic explanation using a combined *P*-value. We applied MechRNA on a number of recently identified cancer-related lncRNAs (PCAT1, PCAT29 and ARLnc1) and also on two well-studied lncRNAs (PCA3 and 7SL). This led to the identification of hundreds of high confidence potential targets for each lncRNA and corresponding mechanisms. These predictions include the known competitive mechanism of 7SL with HuR for binding on the tumor suppressor TP53, as well as

mechanisms expanding what is known about PCAT1 and ARLn1 and their targets BRCA2 and AR, respectively. For PCAT1-BRCA2, the mechanism involves competitive binding with HuR, which we confirmed using HuR immunoprecipitation assays.
Availability and implementation: MechRNA is available for download at https://bitbucket.org/compbio/mechrna.
Contact: agawrons@sfu.ca or cenksahi@indiana.edu or backofen@informatik.uni-freiburg.de
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

With the advance of large-scale transcriptome analysis, it has become evident that the majority of the human genome is transcribed into RNA (Djebali *et al.*, 2012). Out of all currently annotated genes, only a minority is known to code for proteins, while most are believed to be non-coding RNAs (ncRNAs). Beside several small ncRNAs, including small nucleolar RNAs (snoRNAs) and microRNAs (miRNAs), manifold analyses showed that especially long non-coding RNAs (lncRNAs), a designation given to any ncRNA longer than 200 nt, play an important role in cell regulation (Marchese *et al.*, 2017). The major classes of lncRNAs include natural antisense transcripts (NATs), promoter-associated ncRNAs (pncRNAs), pseudogenes and long intergenic non-coding RNAs (lincRNAs). They have a variety of known functions influencing transcription, splicing, mRNA stability and translation (Kung *et al.*, 2013).

For some lncRNAs, the specific mechanism of action is known, however often only isolated examples exist. For many others, the precise mechanism still needs to be determined. At the most fundamental level, every lncRNA mechanism involves RNA–RNA interaction and/or RNA–protein interaction (and via proteins, DNA interactions). So in order to model lncRNA mechanisms computationally, algorithms for predicting these kinds of interactions are essential. There are a number of tools to predict RNA–RNA interactions. These follow four general approaches, in order of complexity: hybridization-only [RNAHybrid (Rehmsmeier *et al.*, 2004), RNADuplex (Lorenz *et al.*, 2011)], sequence concatenation [PairFold (Andronescu *et al.*, 2005), RNAcofold (Bernhart *et al.*, 2006)], accessibility-based [RNAup (Muckstein *et al.*, 2006), IntaRNA2 (Mann *et al.*, 2017)] and full joint structure prediction—leading to the first joint free energy model for interacting RNA strands (Alkan *et al.*, 2006) and follow-up work [piRNA (Chitsaz *et al.*, 2009), inRNAs (Salari *et al.*, 2010), RIP (Huang *et al.*, 2009)]. Hybridization-only methods, where only intermolecular base-paring is considered, and sequence concatenation methods, where standard algorithms for secondary structure prediction are applied to the concatenation of the input RNA, are very fast but produce unrealistic interactions. Accessibility-based tools compute the partition function of each input sequence and determine the energy required for any given region to be unpaired. These energies are then used as penalties when predicting hybridizations. At the expense of a little higher complexity, the modeled interactions are much more realistic. Accessibility-based tools are efficient enough to have been successfully applied to prokaryotic sRNA and eukaryotic miRNA target prediction on a transcriptome-wide scale. However, due to the complexity of these algorithms, the problem of predicting lncRNA interactions on a transcriptome-wide scale quickly becomes intractable for any method more complex than hybridization-only predictions.

It is possible to use RNA–RNA interaction prediction software for transcriptome-wide, lncRNA–RNA interaction prediction, through the use of existing tools such as IntaRNA [on a supercomputer (Terai *et al.*, 2016)] or by new pipelines such as RISearch2 (Alkan *et al.*, 2017). All these approaches need to apply the following steps (not necessarily in order): (i) determine accessible regions on every target sequence [e.g. using Raccess (Kiryu *et al.*, 2011) and remove repeat regions]; (ii) determine 'seeds' with perfect complementary and extend each seed with flanking sequences of fixed length; and (iii) predict (and refine) the interaction between the lncRNA and each of these sequences [e.g. using IntaRNA or RactIP (Kato *et al.*, 2010)]. Unfortunately, the targets of ncRNAs identified through the above approach are typically not very specific. For short ncRNAs such as sRNAs and miRNAs, it is possible to improve specificity via sequence conservation (Wright *et al.*, 2013, 2014) across species. However this does not extend to lncRNAs, which are typically poorly conserved (Iyer *et al.*, 2015). As we will discuss below, one way to improve specificity may be to incorporate RNA–protein interactions with RNA–RNA interactions with RNA–protein interactions.

RNA–protein interactions can be determined experimentally using CLIP-Seq, which is currently the standard protocol for the transcriptome-wide identification of RNA-binding protein (RBP) binding sites. Several protocol variants exist, most notably photoactivatable-ribonucleoside-enhanced CLIP (PAR-CLIP) (Hafner *et al.*, 2010) and individual-nucleotide CLIP (iCLIP) (Konig *et al.*, 2010). Lately, enhanced CLIP (eCLIP) (Van Nostrand *et al.*, 2016) and infrared-CLIP (irCLIP) (Zarnegar *et al.*, 2016) have been introduced to further improve protocol efficiency with varying approaches, as discussed by Uhl *et al.* (2017).

A drawback of CLIP-Seq protocols to identify RBP binding sites is that they naturally rely on the expression of the target transcripts, which is often cell- or tissue-specific, especially in the case of lncRNAs (Brunner *et al.*, 2012; Liu *et al.*, 2016). Computational prediction of missing binding sites is therefore in high demand. While initial prediction methods such as MEME (Bailey and Elkan, 1994) have relied solely on sequence information, more recent tools such as MEMERIS (Hiller *et al.*, 2006), RNAcontext (Kazan *et al.*, 2010) and GraphProt (Maticzka *et al.*, 2014) also incorporate structural information to further improve their predictions.

To our knowledge, no tool exists that integrates both RNA–RNA and RNA–protein interactions. This is crucial for lncRNA interaction prediction since their long length increases the probability of protein binding. The type of RBP, whether it binds to the lncRNA or the target and the location of the RBP relative to the RNA–RNA interaction site can allow inference of the potential lncRNA mechanism.

To solve this problem, we propose MechRNA, a pipeline for combining interaction predictions and biological data to discover potential mechanisms. Specifically, this pipeline aims to discover potential mechanisms of an input lncRNA by (i) predicting lncRNA–target interactions using IntaRNA2 with a new feature improving transcriptome-wide performance, (ii) identifying RBP binding sites predicted by GraphProt on both the targets and the

lncRNA, (iii) finding correlation between the lncRNA and targets using the cancer genome atlas (TCGA) expression or user-provided data, (iv) combining this evidence to generate candidate mechanisms and finally (v) computing joint *P*-values to select the candidate mechanisms that best explain the observed data.

2 Materials and methods

MechRNA has four inputs (Ensembl IDs of lncRNA sequence, target sequences and RBPs and a list of mechansims) and two modes (screening and hypothesis-driven modes). In screening mode, the user only specifies the lncRNA, and the entire transcriptome with all available RBP models is used to predict all possible mechanisms. Since nothing is known about the relationships between the lncRNA, targets and RBPs, correlation data are used to reduce the number of candidates. Hypothesis-driven mode allows the user to specify any *a priori* information they may have on the lncRNA. For example, a common case would be that the lncRNA was experimentally shown to downregulate a set of targets. In this case, the user would specify a list of all downregulatory mechanisms from those that are available and the list of suspected targets. From these inputs, MechRNA predicts lncRNA–target interactions, RBP binding sites and determines the most likely mechanism given these interactions. Here we will describe each stage in detail. An overview of the pipeline is shown in Figure 1.

2.1 Sequence decomposition by accessibility

Since IntaRNA2 uses accessibility to predict RNA–RNA interactions, areas of low accessibility can be removed from the search space. An added benefit of this approach is that long transcripts can be naturally split into smaller sequences that can be analyzed independently. Since IntaRNA2 complexity increases quadratically with sequence length, sequence splitting makes cases tractable that are intractable otherwise, i.e. even transcripts with length >20 kb can be considered. To accomplish a proper splitting, we developed a new algorithm that incrementally detects the least accessible (most structured) positions in the sequence to be used as split positions. The minimal number of splits are selected that are necessary to make every subsequence shorter than a user-specified length and for each of these subsequences to contain no position less accessible than its split positions. A default maximum length threshold of 1500 nt was selected to ensure that the memory usage does not exceed the typical amount of RAM on a PC or the per-core resource availability of a computing cluster. It should be noted that the majority of transcripts are less than the default threshold and therefore the heuristic will usually not be used, i.e. it is mainly applicable to extreme cases.

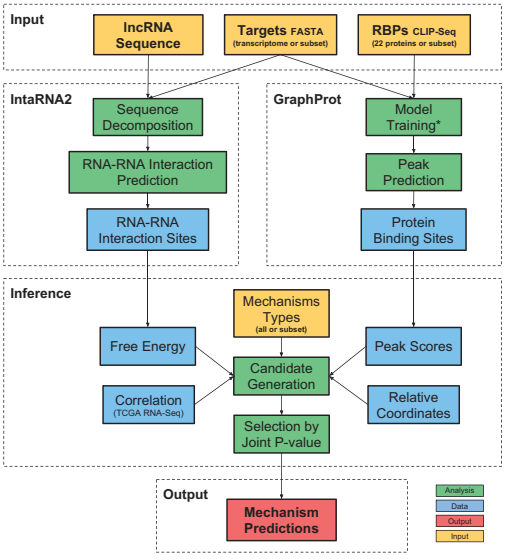


Fig. 1. Overview of the MechRNA pipeline. IntaRNA2 computes the optimal RNA–RNA interaction sites between the lncRNA and the accessible regions of targets/transcriptome. GraphProt predicts protein binding sites for all specificied RBPs on all targets and the lncRNA. Information derived from these predictions, as well as correlation data, is used to generate candidate mechanisms. Finally, the candidate with the lowest joint *P*-value is selected for each lncRNA-target pair, and a output list of mechanisms is produced. (*)Since at the time of publication only 22 RBP CLIP-Seq datasets were available for non-splicing related, post-transcriptional regulation proteins

The algorithm finds the minimal set of most structured points at which to split a long input sequence according to a given length restriction as follows: given a sequence S, the algorithm begins with position $x = 0$ and $y = |S| - l$ where l is a fixed window length (IntaRNA seed length by default). First, the algorithm computes $\max_i(ED(i, i + l))$, where $x \leq i \leq y$ and ED is the accessibility energy for that range. Accessibility energy is the energy required for a region of RNA to be single stranded, inversely proportional to the probability of the bases being paired in that region and computed via the partition function. With the detected position i, a new interval $(i + l, |S| - 1)$ is created and put on the stack. Furthermore, for the current interval, y is updated to $i - 1$. This process is repeated until $y - x + 1$ is less than the length threshold, at which time it is added to the final list of intervals. The algorithm then moves to the next interval from the stack, i.e. the interval created in the last iteration. The iteration continues until the last interval is reached (the first interval created with endpoint $|S| - l$). Highly structured regions will produce many maximum ED windows in close proximity, so a minimum interval length is enforced (again, IntaRNA seed length by default) and regions shorter than this minimum are discarded. The final output is a set of intervals, which are then used as input for IntaRNA. More specifically, IntaRNA will sequentially go through each interval and find the optimal hybridization of the lncRNA with the subsequence contained within the interval. An example execution is shown in Figure 2.

In a test with 100 random sequences of length over 1500 nt, the algorithm reduced the runtime by 13% and peak memory usage by 65%. Since peak memory has a constant upper bound when using this approach, the peak usage reduction is even more dramatic for extreme cases. It must be noted that the full accessibility matrix for the entire target/lncRNA structure is used for computing hybridization energies and is reused for each interval within a target. This allows us to limit the search space of possible hybrids as described without any loss in optimality. In other words, any interaction calculated in an accessible region using subsequences of the input RNAs will be identical to those computed using the full input sequences. In the test above, out of the top 10% of predictions using the vanilla algorithm, 95% of them were identical to those found when using the decomposition. This number increases to 97% when we allow for small differences in predicted sites. The only case where an 'optimal' interaction may be missed is if a highly energetic hybrid exists between highly structured regions of both RNAs where the difference in energy is still greater than the difference in energy for interactions in more accessible regions. It is unclear whether this

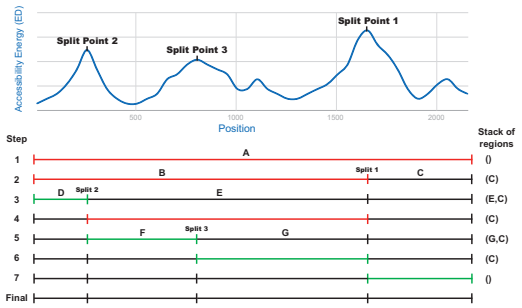


Fig. 2. Example execution of the splitting algorithm with a max sequence length of 1000 nt, where the red interval is the one being processed. (i) The first iteration starts with the entire sequence which is longer than the threshold. (ii) The first split occurs at the position with max ED at ~1700 nt. (iii) The interval is still too long, so a second split is made at the next position of max ED at ~250 nt. (iv) The interval is now below the threshold so the iteration continues to the next interval. (v) This interval is over the threshold and is split at ~800 nt. (vi) and (vii) The next two intervals are below the threshold. (final) The end result is four intervals, all below the length threshold and more accessible than their split positions

type of interaction actually occurs in nature as such interactions exhibit slow kinetics.

2.2 RNA–RNA interaction predictions

The next stage is the prediction of RNA–RNA interactions using IntaRNA2 (Mann *et al.*, 2017) with the modifications outlined above. Details on the IntaRNA2 algorithm can be found in the Supplementary Section 1. IntaRNA2 is executed with the parameters –tAccL 150 –tAccW 200 –qAccL 150 –qAccW 200 -n 5 –tRegionLenMax 1500. The AccL and AccW options used by RNAplfold within IntaRNA2 are recommended by Lange *et al.* (2012). The *n* option specifies the number of predictions (optimal + suboptimals). The tRegionLenMax option specifies the maximum length of an accessible sequence. This value was selected based on the available computational resources and the average RNA length in the reference transcriptomes. This reduces the usage of the heuristic to minimize the effect on the sensitivity of the algorithm.

MechRNA can run IntaRNA2 on a standard machine or distribute the computation across multiple jobs on a computing cluster. Interactions are predicted between the lncRNA and one of the two reference transcriptomes (Ensembl GRCh37.75 and GRCh38.86). The transcriptomes include all mRNA and ncRNA transcripts, excluding sequences <40 nt. This threshold was selected in order to include primary miRNA transcripts while removing dubious, unclassified transcripts. A subset of these transcriptomes is used if the user specifies a list of targets. Once all predictions are completed, the top most energetic interactions (default 3%) are selected for further analysis. *P*-values are computed for each of these interactions using a distribution estimated from the free energies of all interactions (details in Supplementary Section 2.1).

2.3 RNA–protein interaction predictions

For determining RBP binding sites on transcripts, we rely on publicly available CLIP-Seq data. However, since CLIP-Seq depends on transcript expression, binding sites on transcripts specific to certain cell types or conditions cannot be recovered. As we want to study interactions across a reference transcriptome including lncRNAs specifically expressed in certain cancers, we would consequently miss many sites by relying only on direct binding evidence from CLIP-Seq. Therefore, to comprehensively capture protein binding information into our interaction models, we used GraphProt to create transcriptome-wide binding site predictions for 22 RBPs which are known to participate in post-transcriptonal gene regulation and influence transcript stability. As an example, using this approach, we successfully predicted the interaction between hnRNP-L and the lncRNA DSCAM-AS1, for which there were no reads present in the hnRNP-L CLIP-Seq data (Niknafs *et al.*, 2016). Based on the binding sites inferred from CLIP-Seq data for a given RBP, GraphProt learns its binding preferences and integrates these into a predictive model, incorporating either sequence (referred to as sequence model) or sequence and structure information combined (referred to as structure model). A detailed description of the algorithm can be found by Maticzka *et al.* (2014).

For the 22 RBPs, we trained 20 sequence and 8 structure models based on various CLIP-Seq data sources (Table 1). Models for each RBP were selected based on their performance in 10-fold-cross validation, preferring models with higher area under the receiver operating characteristic and mean average precision values. The trained models were then used to predict nucleotide-wise binding score profiles (GraphProt setting: -action predict_profile) on two different reference transcriptomes (described in the previous section). Nucleotide-wise profile scores were further averaged with a sliding window approach, taking all scores up to 5 nt up- and downstream of the score position to calculate the new average score. Peaks were extracted from the average score profiles, where a peak is defined as the maximum score in a contiguous region of positive scores. In order to estimate score significances and to make scores comparable between models, *P*-values for each peak score were calculated (details in Supplementary Section 2.2).

2.4 Correlation data from TCGA protstate tumor samples

If screening mode is selected, correlation data are also incorporated for all RNA–RNA and RNA–protein pairs predicted in the previous stages. To obtain correlation data, we used the GeneNet R package (Schafer and Strimmer, 2005). This approach first computes partial correlations for every pair of genes. The partial correlation is the correlation when the effects of all other variables (genes) are negated. These partial correlations are then used to create a graph where each edge is assigned a *P*-value. We used default parameters and a FDR cutoff of 0.2 to obtain the final correlation network. We deliberately allow a false discovery rate of 20% since the main information will be provided by the RNA–RNA and RNA–protein interactions.

The gene expression data used for correlation computation were derived from TCGA (Weinstein *et al.*, 2013) patient samples. Specifically, this includes 551 RNA-Seq samples, 499 tumor and 52 normal. Only the tumor samples were used in the analysis. The raw read counts were normalized using DeSeq2 (Love *et al.*, 2014). All genes with an average read count <1 were removed, resulting in 32 709 genes (coding/non-coding).

2.5 Combining evidence

At this stage, we incorporate the RNA–RNA and RNA–protein predictions in order to infer a potential mechanism for the lncRNA. For each target transcript, all combinations of RNA–RNA and RNA–protein interactions are classified into candidate mechanisms as shown in Figure 3. The number of combinations is reduced by considering the *a priori* information provided by the user and known functions of the RBPs [for example, HuR is primarily known to

Table 1. List of RBPs used in the analysis including the source CLIP-Seq data and model type

Gene ID	Gene symbol	Protein	Model type	Protocol	Reference
ENSG00000092199	HNRNPC	hnRNP C	Sequence	eCLIP	(Van Nostrand *et al.*, 2016)
ENSG00000165119	HNRNPK	hnRNP K	Sequence	eCLIP	(Van Nostrand *et al.*, 2016)
ENSG00000066044	ELAVL1	HuR	Sequence	PAR-CLIP	(Mukherjee *et al.*, 2011)
ENSG00000102081	FMR1	FMR-1	Structure	eCLIP	(Van Nostrand *et al.*, 2016)
ENSG00000121774	KHDRBS1	Sam68	Structure	eCLIP	(Van Nostrand *et al.*, 2016)
ENSG00000172660	TAF15	TAF15	Sequence	PAR-CLIP	(Hoell *et al.*, 2011)
ENSG00000092847	AGO1	argonaute	Structure	PAR-CLIP	(Hafner *et al.*, 2010)
ENSG00000123908	AGO2	argonaute-2	Structure	PAR-CLIP	(Hafner *et al.*, 2010)
ENSG00000126070	AGO3	argonaute-3	Structure	PAR-CLIP	(Hafner *et al.*, 2010)
ENSG00000134698	AGO4	argonaute-4	Structure	PAR-CLIP	(Hafner *et al.*, 2010)
ENSG00000182944	EWSR1	EWS	Structure	eCLIP	(Van Nostrand *et al.*, 2016)
ENSG00000089280	FUS	FUS	Sequence	PAR-CLIP	(Hoell *et al.*, 2011)
ENSG00000159217	IGF2BP1	IGF2BP1	Structure	PAR-CLIP	(Hafner *et al.*, 2010)
ENSG00000073792	IGF2BP2	IGF2BP2	Structure	PAR-CLIP	(Hafner *et al.*, 2010)
ENSG00000136231	IGF2BP3	IGF2BP3	Structure	PAR-CLIP	(Hafner *et al.*, 2010)
ENSG00000155363	MOV10	MOV-10	Sequence	PAR-CLIP	(Sievers *et al.*, 2012)
ENSG00000055917	PUM2	Pumilio-2	Sequence	eCLIP	(Van Nostrand *et al.*, 2016)
ENSG00000112531	QKI	Hqk	Structure	eCLIP	(Van Nostrand *et al.*, 2016)
ENSG00000120948	TARDBP	TDP-43	Sequence	eCLIP	(Van Nostrand *et al.*, 2016)
ENSG00000116001	TIA1	TIA-1	Sequence	eCLIP	(Van Nostrand *et al.*, 2016)
ENSG00000090905	TNRC6A	TNRC6A	Structure	eCLIP	(Van Nostrand *et al.*, 2016)
ENSG00000197157	SND1	SND1	Structure	eCLIP	(Van Nostrand *et al.*, 2016)

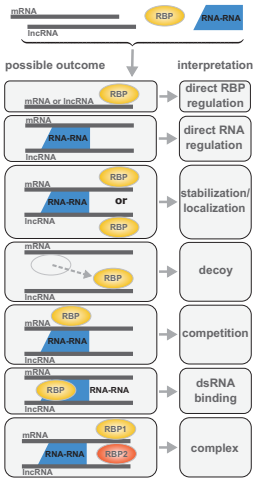


Fig. 3. Illustration of the possible mechanisms that can be inferred from RNA–RNA and RNA–protein interactions

stabilize its bound RNA (Srikantan and Gorospe, 2012)]. In screening mode, the correlations are also used at this stage to determine whether a candidate mechanism is valid. For example, let a target RNA has a peak for RBP *A* and *B*, a lncRNA has a peak for *C* and the RNA–RNA interaction between the two overlaps at the *A* peak. *A* is positively correlated with the target, *B*, *C*, and the lncRNA are negatively correlated with the target. Then the following tuples would be generated, where ([*target_peak*], [*lncRNA_peak*], [*mechanism_type*]) and a dash indicates absence of binding:

- $(-, -, direct\_downregulation)$
- $(A, -, competitive\_downregulation)$
- $(-, C, localization\_downregulation)$
- $(B, -, destabilization)$
- $(A, C, competitive\_downregulation)$
- $(B, C, complex\_downregulation)$

An explanation of each mechanism type with known examples is shown in Table 2. Decoy and direct RBP mechanisms are not included in the predictions since they do not include RNA–RNA interactions, making target prediction too non-specific (fully dependant on correlations). Double-stranded RNA binding mechanisms are not predicted either since the CLIP-Seq protocol does not capture such interactions.

The free energies of the RNA–RNA interactions and the peak scores of the RNA–protein interactions both have associated *P*-values. As mentioned before, each lncRNA-target and protein-target pair of correlations also has a *P*-value. These *P*-values can be used to quantitatively assess whether one mechanism is more likely than another. This requires the combining of up to six *P*-values (depending on the number of interactions involved) into a single *P*-value for each candidate. The intuitive way to accomplish this is to multiply the *P*-values together, however this is not correct since the product of *P*-values is not uniform under the null model. To solve this problem, we use the Stouffer's Z-score method (Stouffer, 1949), which involves computing the sum of the inverse of a normal distribution of each *P*-value, followed by normalization. This approach also allows for weighting *P*-values, but we set all weights to be equal. The final output of the pipeline is the list of potential mechanisms sorted and filtered by the joint *P*-values.

3 Results and discussion

We selected eight lncRNAs to analyze using MechRNA, as summarized in Table 3. 7SL (Abdelmohsen *et al.*, 2014), PCAT1 (Prensner *et al.*, 2011) and ARlnc1 (Accepted in principle, Zhang *et al.* Nature Genetics 2018) recently investigated lncRNAs with known roles in prostate cancer and mechanistic hypotheses are used to test the hypothesis-driven mode. The remaining five lncRNA are used to test the screening mode. PCA3 (Bussemakers *et al.*, 1999) and PCAT29 (Malik *et al.*, 2014) are well-studied prostate cancer related

Table 2. Descriptions of known lncRNA mechanisms

Mechanism	Description	Example
Direct RBP	RBP interaction directly impacts the target or lncRNA	hnRNPL binding to DSCAM-AS1 (Niknafs *et al.*, 2016)
Direct RNA	RNA–RNA interaction directly impacts the target with no RBP involvement	TINCR stabilization of various mRNAs (Kretz *et al.*, 2013)
(De-)stabilization	RNA-RNA interaction increases/decreases the affinity of RBP binding nearby	iNOS stabilization by AS via HuR (Matsui *et al.*, 2007)
Localization	RBP bound to the lncRNA is brought into the vicinity of the target through RNA-RNA interaction	MALAT1 localization of splicing factors (Bernard *et al.*, 2010)
Decoy	RBP is sequestered from the target by the lncRNA	Gas5-AS binding transcription factors (Kino *et al.*, 2010)
Competitive	RBP and lncRNA compete for the same binding location on the target	7SL disrupts HuR stabilization of TP53 (Abdelmohsen *et al.*, 2014)
dsRNA binding	A dsRNA binding protein interacts with stems created from lncRNA interaction	STAU1-mediated decay (Kim *et al.*, 2007)
Complex	The lncRNA facilitates the formation of a complex between multiple proteins	HOTAIR and the polycomb complex (Zhang *et al.*, 2014)

Note: Mechanisms in italics are not included in the predictions.

Table 3. Selected LncRNAs for MechRNA analysis

LncRNA	Length	Target	Protein binding	Mechanism	Cancer type
7SL	299	TP53	HuR	Competitive	Prostate
PCAT1	1992	BRCA2	HuR	Competitive?	Prostate
ARlnc1	2786	AR	Unknown	Unknown	Prostate
PCA3	3922	Unknown	Unknown	Unknown	Prostate
PCAT29	694	Unknown	Unknown	Unknown	Prostate
LINC00514	3385	CLDN9	Unknown	Unknown	NEPC
SSTR5-AS1	2864	SSTR5	Unknown	Unknown	NEPC
TINCR	3733	STAU1	Many	Stabilization	Various

Note: The lncRNAs vary in terms of what is known about their mechanisms, allowing MechRNA to be tested with various amounts of *a priori* data. PCAT1 has a question mark indicating that competitive binding is the hypothesis not been validated yet.

lncRNAs without a known mechanism. SSTR5-AS1 is one of the highest expressed lncRNAs in neuroendocrine prostate cancer (NEPC) and LINC00514 is one of the highest persistently expressed lncRNAs identified in the neuroendocrine transdifferentiation process, which is shown to cause NEPC (Ramnarine *et al.*, 2018). Finally we selected TINCR (Kretz *et al.*, 2013) as a well-known regulator of cell differentiation mediated by interaction with target mRNAs.

3.1 Hypothesis-driven mode results on prostate cancer lncRNAs

We first tested our hypothesis-driven mode with three prostate cancer lncRNAs. The first lncRNA is 7SL, which we use as a validation case since it has a good deal of evidence supporting the proposed mechanism. The next two lncRNAs, PCAT1 and ARlnc1, are less understood and so we aim to build a more complete picture of their potential mechanisms.

3.1.1 7SL downregulation of TP53 through competitive binding with HuR (ELAVL1)

Abdelmohsen et al. (2014) provided the first experimental evidence supporting a competitive lncRNA mechanism. 7SL is a housekeeping ncRNA that is part of the signal recognition particle ribonucleoprotein complex, but also leads to increased cell proliferation when over-expressed in cancer cells. It was demonstrated that 7SL binds to the transcript of the tumor suppressor TP53 near HuR binding sites, preventing HuR from binding and subsequently reducing the stability of TP53. The experimentally validated RNA–RNA interaction was between nucleotide positions 10–56, 256–298 of 7SL and positions 2167–2300 of TP53 (ENST00000269305). Using PAR-CLIP data, they determined that HuR binds at positions 2125–2160, 2452–2472 and 2531–2556.

For this case, we ran MechRNA with 16 protein-coding TP53 transcripts as targets, all downregulatory mechanisms and all RBP models. For all 16 transcripts, 'competitive downregulation' with HuR was predicted to be the most likely mechanism ($P < 10^{-15}$ for the combined P-value as described in Section 2.5). The predicted binding locations of 7SL and HuR for each transcript are shown in Supplementary Table S2. The IntaRNA2 interaction prediction was in agreement with the crude BLAST search done in the experimental study. The 10–56 (actually 10–96 is more energetically favorable) interaction was also predicted but not included in the final results since it is not close enough to the HuR binding site to have an effect. In terms of RBP binding, GraphProt only predicts the 2125–2160 as significant when compared to all HuR binding across the transcriptome. This demonstrates the superiority of using GraphProt over raw PAR-CLIP data. We also show here that this mechanism appears to be ubiquitous across splice variants of TP53.

Another RBP, EWS, was included in this prediction. GraphProt detected a binding site for EWS on 7SL at 140–161, in between the two RNA–RNA interaction sites. EWS is best known for its role in Ewing sarcoma through its translocation with other genes. However, wild-type EWS also acts as a translation repressor by causing mRNA to be retained in the nucleus (Huang *et al.*, 2014). It may be that EWS is aiding in the displacement of HuR and furthering the downregulation of TP53.

3.1.2 PCAT1 downregulation of BRCA2 through competitive binding with HuR (ELAVL1)

PCAT1 was identified by Prensner et al. (2011) as the most differentially expressed lncRNA in prostate cancer. Shortly afterward it was discovered that this lncRNA regulates the important tumor suppressor BRCA2 (Prensner *et al.*, 2014). Specifically, it was shown that PCAT1 reduces BRCA2 mRNA stability and that the first 250 nt of PCAT1 were essential for this process. Furthermore, they demonstrated that this regulation was occurring via the BRCA2 3′ UTR. Since mRNA stability was decreased, our hypothesis is that a similar

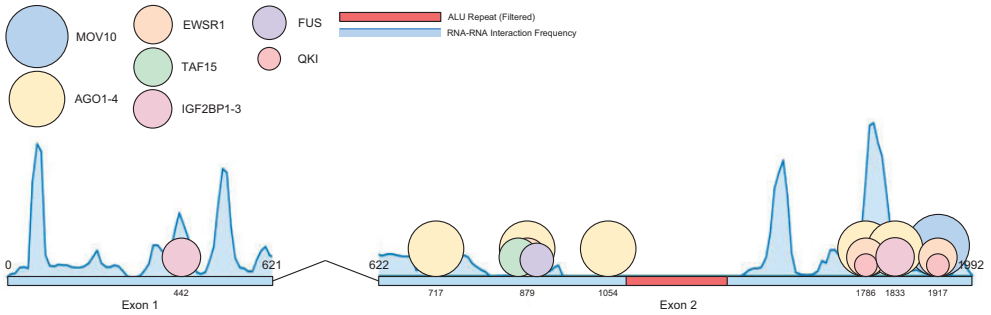


Fig. 4. Distribution of RNA–RNA and RNA–protein interactions for PCAT1. The circles represent proteins and have a diameter relative to their molecular masses. The curve above the exons is a histogram of the frequency of RNA–RNA binding of each position with other transcripts

mechanism to 7SL exists for PCAT1 and BRCA2, so we used the same parameters of all downregulatory mechanisms and all RBP models. For this analysis, we used the BRCA2 3′UTR from the RefSeq transcript as it was used by Prensner et al. (2011).

Figure 4 summarizes the interaction predictions by showing the frequency of interaction for each position of PCAT1 and the significant RBP binding peaks. Our findings appear to support that the first 250 nt play an important role due to high frequency of interaction with targets an no significant binding with RBPs. The predicted mechanism was 'competitive downregulation' (combined *P*-value $P < 10^{-4}$) involving HuR on the 3′UTR. The RNA–RNA interaction is between 11204–11237 on BRCA2 and 65–90 on PCAT1 (−12.493 kcal/mol), with a HuR peak at 11216–11236 on BRCA2. There are also two other HuR binding sites predicted by GraphProt downstream and upstream of the interaction site with similar binding affinity.

To validate this mechanism experimentally, we first confirmed that HuR indeed binds to BRCA2 3′UTR. As shown in Figure 5A, immunoprecipitation of HuR in LNCaP cells pulled down more BRCA2 mRNA than the IgG control. Next, we conducted a competitive binding assay in RWPE cells. This assay immunoprecipitated HuR using an anti-HuR antibody and the bound RNA (BRCA2) was detected by qPCR. In the presence of unmodified PCAT1, the amount of bound BRCA2 RNA was reduced. When using a modified PCAT1 construct with the first 250 nt deleted, there was no affect on the amount of bound BRCA2. This suggests that an interaction involving the 5′ end of PCAT1 is competitively reducing the amount of HuR bound to BRCA2 (Fig. 5B).

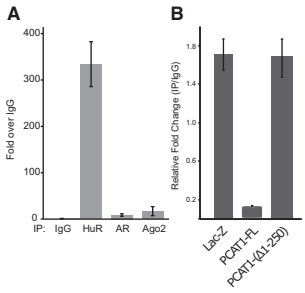


Fig. 5. (**A**) IgG/HuR/AR/Ago2 proteins were immunoprecipitated by antibodies in LNCaP cells and the bound BRCA2 RNA was detected by qPCR. The result confirms binding of BRCA2 mRNA to HuR protein in cells. (**B**) RWPE cells stably expressing lac-Z, PCAT1-FL or PCAT1-delta-1-250 were harvested. HuR was immunoprecipitated using anti-HuR antibody and bound RNA (BRCA2) was detected by qPCR. As shown in A, HuR can bind to BRCA2. In presence of FL-PCAT1 this binding is inhibited. In presence of PCAT1-delta-1-250 there was no effect on HuR and BRCA2 binding. The result confirms the role of PCAT1 in mediating BRCA2-HuR binding

3.1.3 ARlnc1 upregulatory feedback loop with androgen receptor

ARlnc1 has recently been identified as an upregulator of androgen receptor (AR) in prostate cancer (Zhang *et al.* 2018). In turn, AR upregulates ARlnc1, leading to a positive feedback loop contributing to cancer progression. The mechanism was identified with the aid of the first stage of MechRNA, which predicted an RNA–RNA interaction between ARlnc1 and the 3′UTR of AR. However, how exactly ARlnc1 upregulates AR remains unclear. Similarly to 7SL, we ran MechRNA with all RBPs on all AR protein coding transcripts but with all upregulation mechanisms.

The most common and important AR transcript, ENST00000374690, as well as two other splice variants (ENST00000612452 and ENST00000396044) had predicted mechanisms involving the experimentally validated interaction at 815–851 on ARlnc1 (−35.8 kcal/mol). In all three cases, a 'stabilization' mechanism was predicted (respectively, $P < 10^{-6}$, $P < 10^{-5}$, $P < 10^{-4}$ for the combined *P*-values) involving the protein Sam68, which has a strong binding site upstream of the ARlnc1 interaction on the AR 3′UTR. In agreement, Sam68 3′UTR interaction has been shown to enhance target translation (Paronetto *et al.*, 2009). Sam68 is known to increase AR-V7 (ENST00000504326) expression (Stockley *et al.*, 2015), but the authors observed that upregulation of AR-V7 (and full-length) was still present when using a mutated exonic splicing enhancer site. They suggested a synergistic stabilization mechanism via the 3′UTR. Although the 3′UTR of AR-V7 and full-length AR is not shared, a similar binding pattern is observed for Sam68 and ARLnc1 in the AR-V7 3′UTR. Our findings appear to support the additional stabilization mechanism they observed and that all major AR isoforms are regulated in the same manner.

3.2 Screening mode results on prostate cancer lncRNAs

We ran MechRNA on all eight lncRNAs (three from the hypothesis-driven analysis and five additional cancer-related lncRNA as described in Table 3) using the entire transcriptome for potential targets for a broad, unbiased screen. This yielded several hundred to several thousand potential targets for each lncRNA. The number of predictions increased with the length of the lncRNA, since longer lncRNAs are more likely to have RNA–RNA and RNA–protein interactions and consequently more viable combinations of interactions, indicating potential mechanisms. Since our focus here is on cancer, we extracted predicted mechanisms involving known cancer

Table 4. Select lncRNA mechanisms predictions for known cancer genes, selected based on rank (joint p-value) and agreement with known roles of the cancer genes and RBPs

lncRNA	Target		RNA–RNA interaction				RBP–target interaction			RBP–lncRNA interaction			Mechanism	
Gene symbol	Gene symbol	Iso.	FE	Context	Cor.	Cor. FDR	RBP	Cor.	Cor. FDR	RBP	Cor.	Cor. FDR	Type	*P*-value
LINC00514	**AKT1**	3	−65.97	5′UTR	+	1.3×10^{13}	None	NA	NA	None	NA	NA	Direct	2.6×10^{20}
PCAT1	*LEFTY2*	2	−31.63	3′UTR	−	0.001	IGF2BP2	+	1.3×10^{13}	None	NA	NA	Competitive	1.2×10^{19}
PCAT29	BMPR1A	1	−28.32	3′UTR	+	0.182	IGF2BP3	+	2.3×10^{10}	None	NA	NA	Stabilization	6.6×10^{16}
PCA3	*ABI1*	5	−44.55	5′UTR	−	0.023	TAF15	+	0.111	None	NA	NA	De-stabilization	6.8×10^{15}
PCAT1	**HOXC13**	1	−26.57	5′UTR	+	1.3×10^{13}	None	NA	NA	None	NA	NA	Direct	4.8×10^{14}
LINC00514	**FLI1**	1	−60.4096	5′UTR	+	0.083	EWSR1	−	0.006	None	NA	NA	Competitive	5.5×10^{14}
SSTR5-AS1	TP53	7	−33.18	3′UTR	+	0.006	HNRNPC	+	0.081	None	NA	NA	Stabilization	1.6×10^{13}
SSTR5-AS1	**RAC1**	1	−32.97	3′UTR	+	0.159	KHDRBS1	+	4.1×10^{05}	None	NA	NA	Stabilization	2.2×10^{13}
SSTR5-AS1	**HLF**	5	−27.25	3′UTR	+	2.4×10^{07}	None	NA	NA	None	NA	NA	Direct	5.6×10^{13}
PCA3	**HOXC13**	1	−48.02	5′UTR	+	0.013	None	NA	NA	None	NA	NA	Direct	1.7×10^{12}
ARlnc1	**CAMK1D**	1	−27.7706	5′UTR	+	1.4×10^{11}	None	NA	NA	None	NA	NA	Direct	2.0×10^{12}
TINCR	DAXX	7	−106.10	CDS	None	NA	None	NA	NA	IGF2BP2	+	0.046	Localization	2.7×10^{12}
PCAT1	**CCND1**	1	−30.86	3′UTR	+	0.007	ELAVL1	+	0.001	none	NA	NA	Stabilization	1.1×10^{11}
ARlnc1	**BRD4**	2	−31.95	3′UTR	+	0.039	ELAVL1	+	0.052	none	NA	NA	Stabilization	3.0×10^{11}
LINC00514	**CHD4**	4	−36.99	CDS	+	0.029	TAF15	+	0.002	none	NA	NA	Stabilization	7.2×10^{11}
PCAT29	**ALK**	2	−32.85	CDS	+	4.6×10^{07}	None	NA	NA	none	NA	NA	Direct	8.6×10^{11}
TINCR	**NAB2**	3	−66.93	5′UTR	None	NA	None	NA	NA	IGF2BP2	+	0.045	Localization	5.4×10^{09}

Note: Genes in boldface indicate oncogenes, italics indicate tumor suppressors and normal text are uncategorized. The first section indicates the target and how many isoforms (Iso) it interacts with. The next three sections describe the interactions involved. For RNA-RNA, the free energy in kcal/mol (FE) and genomic context are included. For RBP-RNA, the protein name is provided. In all three cases the correlation (+ positive, − negative) and the correlation FDR are shown if applicable. The final section displays the mechanism categorization and the combined *P*-value.

genes from the TSGene (Zhao *et al.*, 2015) and ONGene (Liu *et al.*, 2017) database. These mechanisms are shown in Table 4.

As shown in the table, these prostate cancer lncRNAs generally act as positive regulators of oncogenes with the exception of the PCA3–ABI1 and PCAT1–LEFTY2 interaction. Also the most favorable RNA–RNA interactions commonly occur in the 5′ and 3′ UTRs, as would be expected for post-transcriptional regulation. It is unclear whether the coding sequence (CDS) interactions have any functionality. TINCR-DAXX falls within a small simple repeat region, which may indicate non-specific binding. Another observation is that PCAT1 and PCA3 share the target gene HOXC13 and even bind to the same location on the HOXC13 transcript. HOXC13 is commonly dysregulated in prostate cancer (Komisarof *et al.*, 2017). It may be that the same phenotype is induced by both lncRNAs.

Our most significant result is an interaction involving AKT1, an important and well-studied prostate cancer gene (Cariaga-Martinez *et al.*, 2013). LINC00514 binds very strongly to the 5′UTR and has a strong positive correlation, implying a direct upregulatory effect. No significant protein binding was detected in the region for the included proteins. This would suggest the lncRNA alone is able to regulate AKT1. We observed several other cases like this, labeled as ‘direct’ in the table. It may be the case that some other RBP, which was not included in our analysis due to missing CLIP data, also interacts with AKT1 in this region. As the number of RBPs with available CLIP data is ever increasing, it is likely that a future run of MechRNA with more RBPs might provide additional evidence.

Another significant result was the predicted competitive downregulation of LEFTY2 by PCAT1. This is the most significant result for PCAT1 involving a known cancer gene. It has a close similarity to the PCAT1-BRCA2 mechanism, as it involves the same part of PCAT1 (61–97) binding to a 3′UTR overlapping a protein binding site (in this case IGF2BP2). LEFTY2 is an important tumor suppressor in endometrial cancer (Alowayed *et al.*, 2016). We do not have data for PCAT1 expression in endometrial cancer, but there is high expression in ovarian and breast cancer (Iyer *et al.*, 2015).

The PCA3-ABI1 mechanism is an interesting example demonstrating the importance of sequence accessibility for interaction prediction. ABI1 is known to negatively regulate cell growth and transformation and is down-regulated in a variety of cancers (Chen *et al.*, 2010; Cui *et al.*, 2010; Zhang *et al.*, 2015). The gene has 11 annotated protein-coding isoforms in Ensembl, 9 of which have an identical 5′ UTR sequence. However, three of the splice variants exclude exon three, leading to a much more energetic binding to PCA3 (−13 kcal/mol difference). This is because the exclusion affects the accessibility of the 5′UTR by reducing the probability that this region is bound by intramolecular interactions. If PCA3 is indeed down-regulating ABI1, as the correlation indicates, there may be selection for these isoforms in cancer cells to increase the effect of PCA3. Naive approaches to RNA–RNA interaction prediction computing only the hybridization would not capture the difference in interaction energy between the different splice variants. This is because the sequence of the best hybridization site is always the same, the only feature considered when computing the optimal interaction. However, the accessibility can differ between different isoforms, which may affect the location of the true optimal interaction site, as we see in the case of PCA3-ABI1.

4 Conclusion

Recent discoveries of lncRNA mechanisms indicate that there exists a complex interplay between RBPs, lncRNAs and their target RNAs. Until now, RNA–RNA and RNA–protein interaction predictions were carried out independently, failing to capture this complexity. Here we present MechRNA, the first tool to integrate both kinds of interactions in order to more accurately predict lncRNA mechanisms. We accomplish this by combining the output of IntaRNA2 and GraphProt into a novel inference tool, which determines the most likely combination of interactions. These sets of interactions are then classified into mechanisms using correlation data from publicly available patient gene expression samples or

user-defined *a priori* data. We demonstrated the functionality of MechRNA by analyzing eight prostate cancer lncRNAs with varying amounts of information available with respect to their mechanisms. The results confirm one known mechanism, provide new insights into poorly understood mechanisms and offer new hypotheses for the remaining lncRNAs without known mechanisms. Despite the challenges involved in this kind of analysis (discussed in Supplementary Section 3), our results show that MechRNA is a useful tool for identifying potential roles of lncRNAs in cancer and for furthering our understanding on lncRNA mechanisms in general.

Funding

This work was supported in part by the Indiana University Grand Challenges Program, The Precision Health Initiative and the NSERC Discovery Frontiers Program, The Cancer Genome Collaboratory to SCS. Furthermore, this work has been suppported by the Baden-Württemberg-Stiftung [BWST NCRNA 008], the German Research Foundation [DFG grant BA2168/11-1 SPP 1738] and the BMBF Verbundprojekt Deutsches Netzwerk für Bioinformatik-Infrastruktur(de.NBI).

Conflict of Interest: none declared.

References

Abdelmohsen,K. *et al.* (2014) 7SL RNA represses p53 translation by competing with HuR. *Nucleic Acids Res.*, **42**, 10099–10111.

Alkan,C. *et al.* (2006) RNA-RNA interaction prediction and antisense RNA target search. *J. Comput. Biol.*, **13**, 267–282.

Alkan,F. *et al.* (2017) RIsearch2: suffix array-based large-scale prediction of RNA-RNA interactions and siRNA off-targets. *Nucleic Acids Res.*, **45**, e60.

Alowayed,N. *et al.* (2016) LEFTY2 controls migration of human endometrial cancer cells via focal adhesion kinase activity (FAK) and miRNA-200a. *Cell. Physiol. Biochem.*, **39**, 815–826.

Andronescu,M. *et al.* (2005) Secondary structure prediction of interacting RNA molecules. *J. Mol. Biol.*, **345**, 987–1001.

Bailey,T.L. and Elkan,C. (1994) Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **2**, 28–36.

Bernard,D. *et al.* (2010) A long nuclear-retained non-coding RNA regulates synaptogenesis by modulating gene expression. *EMBO J.*, **29**, 3082–3093.

Bernhart,S.H. *et al.* (2006) Partition function and base pairing probabilities of RNA heterodimers. *Algorithms Mol. Biol.*, **1**, 3.

Brunner,A.L. *et al.* (2012) Transcriptional profiling of long non-coding rnas and novel transcribed regions across a diverse panel of archived human cancers. *Genome Biol.*, **13**, R75.

Bussemakers,M.J. *et al.* (1999) DD3: a new prostate-specific gene, highly overexpressed in prostate cancer. *Cancer Res.*, **59**, 5975–5979.

Cariaga-Martinez,A.E. *et al.* (2013) Distinct and specific roles of AKT1 and AKT2 in androgen-sensitive and androgen-independent prostate cancer cells. *Cell. Signal*, **25**, 1586–1597.

Chen,H. *et al.* (2010) Integrity of SOS1/EPS8/ABI1 tri-complex determines ovarian cancer metastasis. *Cancer Res.*, **70**, 9979–9990.

Chitsaz,H. *et al.* (2009) A partition function algorithm for interacting nucleic acid strands. *Bioinformatics*, **25**, i365–i373.

Cui,M. *et al.* (2010) Downregulation of ABI1 expression affects the progression and prognosis of human gastric carcinoma. *Med. Oncol.*, **27**, 632–639.

Djebali,S. *et al.* (2012) Landscape of transcription in human cells. *Nature*, **489**, 101–108.

Hafner,M. *et al.* (2010) Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP. *Cell*, **141**, 129–141.

Hiller,M. *et al.* (2006) Using RNA secondary structures to guide sequence motif finding towards single-stranded regions. *Nucleic Acids Res.*, **34**, e117.

Hoell,J.I. *et al.* (2011) RNA targets of wild-type and mutant FET family proteins. *Nat. Struct. Mol. Biol.*, **18**, 1428–1431.

Huang,F.W. *et al.* (2009) Partition function and base pairing probabilities for RNA-RNA interaction prediction. *Bioinformatics*, **25**, 2646–2654.

Huang,L. *et al.* (2014) EWS represses cofilin 1 expression by inducing nuclear retention of cofilin 1 mRNA. *Oncogene*, **33**, 2995–3003.

Iyer,M.K. *et al.* (2015) The landscape of long noncoding RNAs in the human transcriptome. *Nat. Genet.*, **47**, 199–208.

Kato,Y. *et al.* (2010) RactIP: fast and accurate prediction of RNA-RNA interaction using integer programming. *Bioinformatics*, **26**, i460–i466.

Kazan,H. *et al.* (2010) RNAcontext: a new method for learning the sequence and structure binding preferences of RNA-binding proteins. *PLoS Comput. Biol.*, **6**, e1000832.

Kim,Y.K. *et al.* (2007) Staufen1 regulates diverse classes of mammalian transcripts. *EMBO J.*, **26**, 2670–2681.

Kino,T. *et al.* (2010) Noncoding RNA gas5 is a growth arrest- and starvation-associated repressor of the glucocorticoid receptor. *Sci. Signal*, **3**, ra8.

Kiryu,H. *et al.* (2011) A detailed investigation of accessibilities around target sites of siRNAs and miRNAs. *Bioinformatics*, **27**, 1788–1797.

Komisarof,J. *et al.* (2017) A four gene signature predictive of recurrent prostate cancer. *Oncotarget*, **8**, 3430–3440.

Konig,J. *et al.* (2010) iCLIP reveals the function of hnRNP particles in splicing at individual nucleotide resolution. *Nat. Struct. Mol. Biol.*, **17**, 909–915.

Kretz,M. *et al.* (2013) Control of somatic tissue differentiation by the long non-coding RNA TINCR. *Nature*, **493**, 231–235.

Kung,J.T. *et al.* (2013) Long noncoding RNAs: past, present, and future. *Genetics*, **193**, 651–669.

Lange,S.J. *et al.* (2012) Global or local? Predicting secondary structure and accessibility in mRNAs. *Nucleic Acids Res.*, **40**, 5215–5226.

Liu,S.J. *et al.* (2016) Single-cell analysis of long non-coding rnas in the developing human neocortex. *Genome Biol.*, **17**, 67.

Liu,Y. *et al.* (2017) Ongene: a literature-based database for human oncogenes. *J. Genet. Genomics*, **16**, 1–620.

Lorenz,R. *et al.* (2011) ViennaRNA Package 2.0. *Algorithms Mol. Biol.*, **6**, 26.

Love,M.I. *et al.* (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.*, **15**, 550.

Malik,R. *et al.* (2014) The lncRNA PCAT29 inhibits oncogenic phenotypes in prostate cancer. *Mol. Cancer Res.*, **12**, 1081–1087.

Mann,M. *et al.* (2017) IntaRNA 2.0: enhanced and customizable prediction of RNA-RNA interactions. *Nucleic Acids Res.*, **45**, W435–W439.

Marchese,F.P. *et al.* (2017) The multidimensional mechanisms of long noncoding RNA function. *Genome Biol.*, **18**, 206.

Maticzka,D. *et al.* (2014) GraphProt: modeling binding preferences of RNA-binding proteins. *Genome Biol.*, **15**, R17.

Matsui,K. *et al.* (2007) Natural antisense transcript stabilizes inducible nitric oxide synthase messenger RNA in rat hepatocytes. *Hepatology*, **47**, 686–697.

Muckstein,U. *et al.* (2006) Thermodynamics of RNA-RNA binding. *Bioinformatics*, **22**, 1177–1182.

Mukherjee,N. *et al.* (2011) Integrative regulatory mapping indicates that the RNA-binding protein HuR couples pre-mRNA processing and mRNA stability. *Mol. Cell*, **43**, 327–339.

Niknafs,Y.S. *et al.* (2016) The lncRNA landscape of breast cancer reveals a role for DSCAm-AS1 in breast cancer progression. *Nat. Commun.*, **7**, 12791.

Paronetto,M.P. *et al.* (2009) Sam68 regulates translation of target mrnas in male germ cells, necessary for mouse spermatogenesis. *J. Cell Biol.*, **185**, 235–249.

Prensner,J.R. *et al.* (2011) Transcriptome sequencing across a prostate cancer cohort identifies PCAT-1, an unannotated lincRNA implicated in disease progression. *Nat. Biotechnol.*, **29**, 742–749.

Prensner,J.R. *et al.* (2014) PCAT-1, a long noncoding RNA, regulates BRCA2 and controls homologous recombination in cancer. *Cancer Res.*, **74**, 1651–1660.

Ramnarine,V.R. *et al.* (2018) The long noncoding RNA landscape of neuroendocrine prostate cancer and its clinical implications. *GigaScience*, 10.1093/gigascience/giy050.

Rehmsmeier,M. *et al.* (2004) Fast and effective prediction of microRNA/target duplexes. *RNA*, **10**, 1507–1517.

Salari,R. *et al.* (2010) Fast prediction of RNA-RNA interaction. *Algorithms Mol. Biol.*, **5**, 5.

Schafer,J. and Strimmer,K. (2005) An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, **21**, 754–764.

Sievers,C. *et al.* (2012) Mixture models and wavelet transforms reveal high confidence rna-protein interaction sites in MOV10 PAR-CLIP data. *Nucleic Acids Res.*, **40**, e160–e160.

Srikantan,S. and Gorospe,M. (2012) HuR function in disease. *Front. Biosci. (Landmark Ed)*, **17**, 189–205.

Stockley,J. *et al.* (2015) The RNA-binding protein Sam68 regulates expression and transcription function of the androgen receptor splice variant AR-V7. *Sci Rep.*, **5**, 13426.

Stouffer,S. (1949) The American soldier. Vol. 1: adjustment during army life. In: *Studies in Social Psychology in World War II*. Princeton University Press Princeton, pp. xiii, 599, 675.

Terai,G. *et al.* (2016) Comprehensive prediction of lncRNA-RNA interactions in human transcriptome. *BMC Genomics*, **17**(**Suppl 1**), 12.

Uhl,M. *et al.* (2017) Computational analysis of clip-seq data. *Methods*, **118–119**, 60–72.

Van Nostrand,E.L. *et al.* (2016) Robust transcriptome-wide discovery of RNA binding protein binding sites with enhanced clip (eclip). *Nat. Methods*, **13**, 508.

Weinstein,J.N. *et al.* (2013) The Cancer Genome Atlas Pan-Cancer analysis project. *Nat. Genet.*, **45**, 1113–1120.

Wright,P.R. *et al.* (2013) Comparative genomics boosts target prediction for bacterial small RNAs. *Proc. Natl. Acad. Sci. U.S.A*, **110**, E3487–E3496.

Wright,P.R. *et al.* (2014) CopraRNA and IntaRNA: predicting small RNA targets, networks and interaction domains. *Nucleic Acids Res.*, **42**, W119–W123.

Zarnegar,B.J. *et al.* (2016) irclip platform for efficient characterization of protein–RNA interactions. *Nat. Methods*, **13**, 489.

Zhang,J. *et al.* (2014) Long non-coding RNA HOTAIR in carcinogenesis and metastasis. *Acta Biochim. Biophys. Sin. (Shanghai)*, **46**, 1–5.

Zhang,J. *et al.* (2015) Upregulation of Abelson interactor protein 1 predicts tumor progression and poor outcome in epithelial ovarian cancer. *Hum. Pathol.*, **46**, 1331–1340.

Zhao,M. *et al.* (2015) Tsgene 2.0: an updated literature-based knowledgebase for tumor suppressor genes. *Nucleic Acids Res.*, **44**, D1023–D1031.

Zhang,Y. *et al.* (2018) Analysis of the androgen receptor-regulated lncRNA landscape identifies a role for ARLNC1 in prostate cancer progression. *Nature Genet.*, NG-A45277R1.

[P3] RNAProt: an efficient and feature-rich RNA binding protein binding site predictor

Publication:

- [P3] **Michael Uhl**, Van Dinh Tran, Florian Heyl, and Rolf Backofen. **RNAProt: an efficient and feature-rich RNA binding protein binding site predictor**. *GigaScience*, 2021.

Contributions of individual authors:

"I am the main contributor to this work. I conceived the study together with Van Dinh Tran and Rolf Backofen. I implemented the tool, performed the data analysis, wrote the online manual and the manuscript. Van Dinh Tran contributed to and supported me in implementing the deep neural network part of the tool. Florian Heyl contributed Wilcoxon test p-values and DeepCLIP cross validation results. All authors reviewed and approved the final manuscript."

Michael Uhl

The following co-authors confirm the above-stated contributions:

GigaScience, 10, 2021, 1–13

https://doi.org/10.1093/gigascience/giab054
Technical Note

TECHNICAL NOTE

RNAProt: an efficient and feature-rich RNA binding protein binding site predictor

Michael Uhl [1], Van Dinh Tran [1,*], Florian Heyl [1] and Rolf Backofen [1,2,*]

[1]Bioinformatics Group, Department of Computer Science, University of Freiburg, Georges-Koehler-Allee 106, 79110 Freiburg, Germany and [2]Signalling Research Centres BIOSS and CIBSS, University of Freiburg, Schaenzlestr. 18, 79104 Freiburg, Germany

*Correspondence address. Van Dinh Tran and Rolf Backofen, Bioinformatics Group, Department of Computer Science, University of Freiburg, Georges-Koehler-Allee 106, 79110 Freiburg, Germany. E-mail: backofen@informatik.uni-freiburg.de http://orcid.org/0000-0002-7357-4959, Phone: +49 (0) 761 / 203 - 7461.

Abstract

Background: Cross-linking and immunoprecipitation followed by next-generation sequencing (CLIP-seq) is the state-of-the-art technique used to experimentally determine transcriptome-wide binding sites of RNA-binding proteins (RBPs). However, it relies on gene expression, which can be highly variable between conditions and thus cannot provide a complete picture of the RBP binding landscape. This creates a demand for computational methods to predict missing binding sites. Although there exist various methods using traditional machine learning and lately also deep learning, we encountered several problems: many of these are not well documented or maintained, making them difficult to install and use, or are not even available. In addition, there can be efficiency issues, as well as little flexibility regarding options or supported features. **Results:** Here, we present RNAProt, an efficient and feature-rich computational RBP binding site prediction framework based on recurrent neural networks. We compare RNAProt with 1 traditional machine learning approach and 2 deep-learning methods, demonstrating its state-of-the-art predictive performance and better run time efficiency. We further show that its implemented visualizations capture known binding preferences and thus can help to understand what is learned. Since RNAProt supports various additional features (including user-defined features, which no other tool offers), we also present their influence on benchmark set performance. Finally, we show the benefits of incorporating additional features, specifically structure information, when learning the binding sites of an hairpin loop binding RBP. **Conclusions:** RNAProt provides a complete framework for RBP binding site predictions, from data set generation over model training to the evaluation of binding preferences and prediction. It offers state-of-the-art predictive performance, as well as superior run time efficiency, while at the same time supporting more features and input types than any other tool available so far. RNAProt is easy to install and use, comes with comprehensive documentation, and is accompanied by informative statistics and visualizations. All this makes RNAProt a valuable tool to apply in future RBP binding site research.

Keywords: CLIP-seq; eCLIP; RBP binding site prediction; deep learning; recurrent neural networks; visualization

Introduction

RNA-binding proteins (RBPs) regulate many vital steps in the RNA life cycle, such as splicing, transport, stability, and translation [1]. Recent studies suggest there are more than 2,000 human RBPs, including hundreds of unconventional RBPs, such as those lacking known RNA-binding domains [2–4]. Numerous RBPs have been implicated in diseases like cancer, neurodegeneration, and genetic disorders [5–7], lending urgency characteriz-

Received: 2 April 2021; **Revised:** 18 May 2021; **Accepted:** 27 July 2021

ing their functions and shedding light on their complex cellular interplay.

An important step to understanding RBP functions is to identify the precise RBP binding locations on regulated RNAs. In this regard, CLIP-seq (cross-linking and immunoprecipitation followed by next-generation sequencing) [8], together with its popular modifications photoactivatable-ribonucleoside-enhanced cross-linking and immunoprecipitation (PAR-CLIP) [9], individual-nucleotide resolution UV cross-linking and immunoprecipitation (iCLIP) [10], and enhanced CLIP (eCLIP) [11], has become the state-of-the-art technique used to experimentally determine transcriptome-wide binding sites of RBPs. A CLIP-seq experiment for a specific RBP results in a library of reads bound and protected by the RBP, making it possible to deduce its binding sites by mapping the reads back to the respective reference genome or transcriptome. In practice, a computational analysis of CLIP-seq data has to be adapted for each CLIP-seq protocol [12]. Within the analysis, arguably the most critical part is the process of peak calling: that is, inferring RBP binding sites from the mapped read profiles. Among the many existing peak callers, some popular tools are Piranha [13], CLIPper [14], and PureCLIP [15].

While peak calling is essential to separate authentic binding sites from unspecific interactions and thus reduce the false positive rate, it cannot solve the problem of expression dependency. In order to detect RBP binding sites by CLIP-seq, the target RNA has to be expressed at a certain level in the experiment. Since gene expression naturally varies between conditions, CLIP-seq data cannot be used directly to make condition-independent binding assumptions on a transcriptome-wide scale. Doing so would only increase the false negative rate: for example, marking all regions not covered by CLIP-seq reads as non-binding, while in fact one cannot tell due to the lack of expression information. Moreover, expression variation is especially high for long non-coding RNAs, an abundant class of non-coding RNAs gaining more and more attention due to their diverse cellular roles [16]. It is therefore of great importance to infer RBP binding characteristics from CLIP-seq data in order to predict missing binding sites. To give an example, Ferrarese et al. [17] investigated the role of the splicing factor Polypyrimidine tract-binding protein 1 (PTBP1) in differential splicing of the tumor suppressor gene Annexin A1 (ANXA7) in glioblastoma. Despite strong biological evidence for PTBP1 directly binding ANXA7, no binding site was found in a publicly available CLIP-seq data set for PTBP1. Instead, only a computational analysis was capable to detect and correctly localize the presence of potential binding sites, which were then experimentally validated.

Over the years, many approaches for RBP binding site prediction have been presented, from simple sequence motif searches to more sophisticated methods incorporating classical machine learning and, lately, also deep learning. Some popular earlier methods include RNAcontext [18] and GraphProt [19], which can both incorporate RNA sequence and structure information into their predictive models. While RNAcontext utilizes a sequence and structure motif model, GraphProt uses a graph kernel coupled with Support Vector Machine, showing improved performance over motif-based techniques. From 2015 on, various deep learning–based methods have been proposed, starting with DeepBind [20], which uses sequence information to train a convolutional neural network (CNN). Subsequent methods largely built upon this methodology, often using CNNs in combination with recurrent neural networks (RNNs) [21]. Some of them also incorporate additional features, usually specializing in a specific feature, such as structure, evolutionary conservation, or region type information, to demonstrate its benefits. While these methods can certainly provide state-of-the-art predictive performance, we encountered several issues: many lack proper documentation, are not maintained, or are not even available, even though they are presented as prediction tools in the original papers. Moreover, efficiency in terms of run time can be a problem, as well as restricted options regarding data processing and, in general, only a few supported features.

Here, we present RNAProt, a computational RBP binding site prediction framework based on RNNs that takes care of the described issues: RNAProt provides both state-of-the-art performance and efficient run times. It comes with comprehensive documentation and is easy to install via Conda. The availability of a Conda package, which no other related deep-learning tool offers to our knowledge, also allows for easy integration into larger workflows, such as Snakemake pipelines [22] or inside the Galaxy framework [23]. RNAProt offers various position-wise features on top of the sequence information, such as secondary structure, conservation scores, or region annotations, which can also be user supplied. Through its use of an RNN-based architecture, RNAProt natively supports input sequences of variable lengths. In contrast, CNNs are constrained to fixed-sized inputs that, for example, exclude the direct usage of variable-sized inputs, usually defined by peak callers. Moreover, RNAProt is currently the most flexible method with regard to the support of input data types: apart from sequences and genomic regions, it can also handle transcript regions, providing automatic feature annotations for all 3 types. Comprehensive statistics and visualizations are provided as well in the form of HTML reports, site profiles, and logos. In addition, the short run times allow for on-the-fly model training to quickly test hypotheses regarding data set, parameter, or feature choices.

Methods

The RNAProt framework

RNAProt utilizes RBP binding sites identified by CLIP-seq and related protocols to train an RNN-based model, which is then used to predict new binding sites on given input RNA sequences. Fig. 1 illustrates the RNAProt framework and its general workflow. RNAProt accepts RBP binding sites in FASTA or Browser Extensible Data (BED) formats. The latter also requires a genomic sequence file (.2bit format) and a genomic annotations file (Gene Transfer Format (GTF)).Compared to FASTA, genomes in binary 2bit format occupy less disk space, allow for faster sequence extraction, and also store repeat region information, which can be used as a feature. Binding sites can be supplied either as sequences, genomic regions, or aranscript regions (GTF file with corresponding transcript annotation required). Additional inputs are available depending on the binding site input type, as well as the selected features (see the “Supported features” section).

RNAProt can be run in 5 different program modes: generation of training and prediction sets, model training and evaluation, and model prediction (see the “Program modes” section). Depending on the executed mode, various output files are generated. For the data set generation modes, HTML reports can be output, which contain detailed statistics and visualizations regarding the positive, negative, or test data set. This way, for example, one can easily compare the positive input set with the generated negative set and spot possible similarities and differences. Reports include statistics on: site lengths, sequence complexity, di-nucleotide distributions, k-mer statistics, target re-

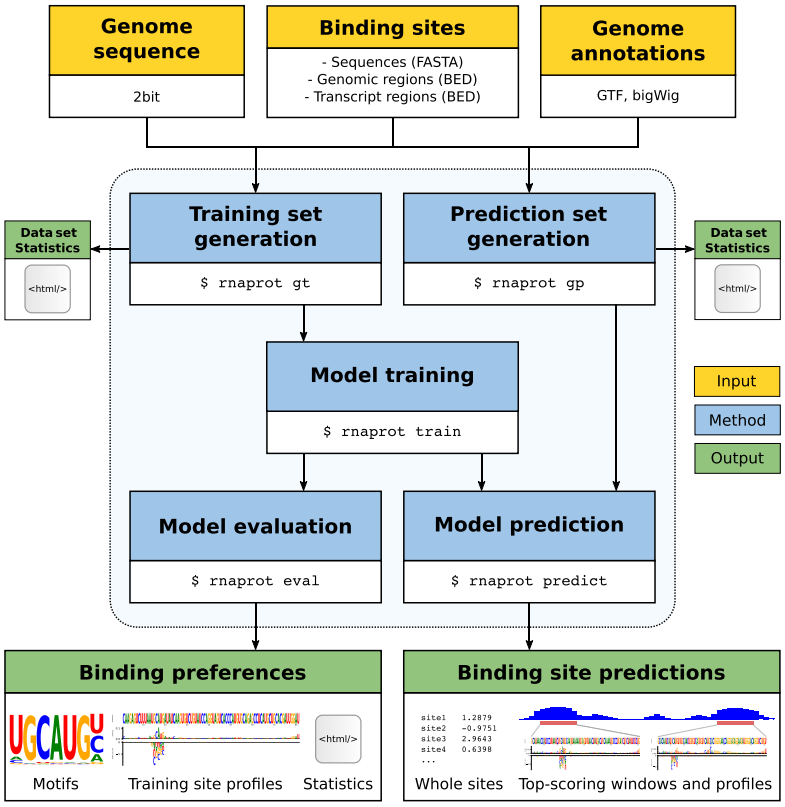


Figure 1: Overview of the RNAProt framework. The yellow boxes mark necessary framework inputs, the blue boxes mark the 5 program modes of RNAProt, and the green boxes mark the framework outputs. Arrows show the dependencies between inputs, modes, and outputs.

gion biotype, and overlap statistics, as well as additional statistics and visualizations for each selected feature. In the model evaluation mode, sequence and additional feature logos are output, as well as training site profiles for a subset of training sites to illustrate binding preferences. In the model prediction mode, whole site or moving window predictions are supported. In case of moving windows, position-wise scoring profiles are calculated and peak regions and top-scoring windows are extracted from the profiles. For a complete and up-to-date description, please refer to the online documentation on GitHub [24].

Model architecture

RNAProt features an RNN-based model for binary classification of input sequences, which can be further customized from the command line or optimized using state-of-the-art hyperparameter optimization by Bayesian Optimization and Hyperband (BOHB) [25]. RNN-based models are well suited to learn from linear sequence information: in particular to learn dependencies between near or distant parts in a given sequence. This has been demonstrated in a number of related tasks over the years, from natural language processing to the analysis of time-series data and biological sequences like DNA or RNA. The type of RNN network used by RNAProt can be adjusted (Long Short-Term Memory [LSTM] [26] or Gated Recurrent Unit [27]), as can the numbers of hidden and full connected layers and dimensions, use of bidirectional RNN, or an embedding layer instead of 1-hot encoding for the sequence feature. As the optimizer, RNAProt applies an improved version of the Adam optimizer, termed AdamW [28]. RNAProt's default hyperparameter setting was used to generate all the results presented in this work: a batch size of 50, learning rate of 0.001, weight decay of 0.0005, RNN model type of Gated Recurrent Unit, number of RNN layers set as 1, RNN layer dimensions set at 32, number of fully connected layers set as 1, dropout rate of 0.5, and no sequence embedding.

Program modes

RNAProt is logically split into 5 different program modes: training set generation (`rnaprot gt`), prediction set generation (`rnaprot gp`), model training (`rnaprot train`), model evaluation (`rnaprot eval`), and model prediction (`rnaprot predict`). Separating data set generation from training or prediction has the advantage that feature values of interest have to be calculated or extracted only once (e.g., secondary structure, conservation scores, region annotations). Since model training is fast, one can then quickly train several models to assess which features or settings in general work best and move on to predictions. In the following we briefly introduce the mode functionalities.

Training set generation

This mode (`rnaprot gt`) is used to generate a training data set from a given set of RBP binding sites, which can be sequences, genomic regions, or transcript regions. In case sequences (FASTA format) are given as input, negative training sequences can be supplied or generated by k-nucleotide shuffling of the positive

Table 1: RNAProts's 3 supported input types (sequences, genomic regions [Genomic], transcript regions [Transcript]) and the features available for them

	Input		
Feature	Sequences	Genomic	Transcript
structure	YES	YES	YES
conservation scores	NO	YES	YES
exon-intron regions	NO	YES	NO
transcript regions	NO	YES	YES
repeat regions	NO	YES	YES
user-defined	NO	YES	YES

input sequences. In case genomic or transcript regions (BED format) are given as input, negatives can be supplied or selected randomly from gene or transcript regions containing positive sites (i.e., RBP binding sites identified by CLIP-seq). In general, we recommend supplying BED regions, as this allows RNAProt to automatically generate a negative set by randomly sampling sites from the genome or transcriptome. By default, negative sites are sampled based on 2 criteria: (i) sampling only from gene regions covered by positive sites; and (ii) no overlap with any positive site. The number of generated negative sites can be further specified, as can regions from which not to extract them. Output site lengths can be of variable or fixed size, and various filtering options are available to filter the sites by score, sequence complexity, region, or length. Concerning site lengths, RNAProt can train and predict on sequences of variable length due to its solely RNN-based architecture. For CNN-based methods this is usually not the case (unless the method internally applies padding before training and predicting). To keep data sets compatible with other tools, RNAProt therefore offers both variable and fixed-size outputs. Depending on the input type (see Table 1), different additional features can be selected for annotating the positive and negative sites (see the "Supported features" section for more details). An HTML report can be generated, providing statistics and visualizations to compare the positive with the negative set. The whole training data set is stored in a folder that forms the main input to the model training mode.

Model training

After generating a training set, a model can be trained on the data set in model training mode (`rnaprot train`). By default, all features of the training set are used to train the model, but specific features can be selected as well. Cross-validation is supported to estimate generalization performance, as well as learning curve plots and hyperparameter optimization using BOHB [25]. Unless cross-validation is specified, a model is trained using the default hyperparameters (or if BOHB is enabled, using the optimized hyperparameters after BOHB has finished) and output data are stored in a new folder, which serves as input to the model evaluation and model prediction modes.

Model evaluation

This mode (`rnaprot eval`) is used to visualize binding preferences of the model trained with `rnaprot train`. Sequence and additional feature logos of various lengths can be output, as well as training site profiles for a user-defined subset of training sites (see "Visualization" section for more details).

Prediction set generation

The prediction set generation mode (`rnaprot gp`) resembles `rnaprot gt` but, instead of generating a training set containing positives and negatives, it generates a prediction set from a given set of sites or sequences. Note that the types of additional features that can be added to the prediction set are dictated by the types used to train the model. Its output folder forms the input of `rnaprot predict`.

Model prediction

Model prediction mode (`rnaprot predict`) is used to predict whole binding sites or peak regions and top-scoring windows from sliding window profiles for a given set of sequences, genomic sites, or transcript sites. The prediction data set needs to be generated by `rnaprot gp` beforehand, as does the model, which needs to be trained through `rnaprot train`. Profiles of top-scoring windows can also be plotted and the input sites on which to predict can be specified.

Supported features

RNAProt supports the following position-wise features, which can be utilized for training and prediction in addition to the sequence feature: secondary structure information (structural element probabilities), conservation scores (phastCons and phyloP), exon-intron annotation, transcript region annotation, and repeat region annotation. In addition, it also accepts user-defined region features (categorical or numerical; see documentation on GitHub [24] for details and examples), which no other tool so far offers. Table 1 lists the features available for each binding site input type.

Secondary structure information

RNAProt can include position-wise structure information, encoded as unpaired probabilities for different loop contexts (probabilities for the nucleotide being paired or inside external, hairpin, internal, or multi loops). ViennaRNA's RNAplfold [29] is used with its sliding window approach, with user-definable parameters (by default these are window size = 70, maximum base pair span length = 50, and probabilities for regions of length u = 3). Note that genomic or transcript input sites are automatically extended on both sides (by window size) to get the most accurate structure predictions. This important feature is also not offered by any related tool.

Conservation scores

RNAProt supports 2 scores measuring evolutionary conservation (phastCons and phyloP). Human conservation scores were downloaded from the University of California Santa Cruz (UCSC) Genome Browser website, using the phastCons and phyloP scores generated from multiple sequence alignments of 99 vertebrate genomes to the human genome (as described in the GitHub manual [24]). RNAProt accepts scores in .bigWig format. To assign conservation scores to transcript regions, transcript regions are first mapped to the genome using the provided GTF file.

Exon-intron annotation

Exon-intron annotation in the form of 1-hot encoded exon or intron labels can also be added. Labels are assigned to each input BED site position by overlapping the site with genomic exon regions using BEDTools [30]. To unambiguously assign labels, RNAProt by default uses the most prominent isoform for each gene. The most prominent isoform for each gene gets se-

lected through hierarchical filtering of the transcript information present in the input GTF file (for the benchmark results we used the Ensembl Genes 99 GRCh38.p13 version): given that the transcript is part of the GENCODE basic gene set, RNAProt selects transcripts based on their transcript support level (highest priority) and by transcript length (longer isoform preferred). The extracted isoform exons are then used for region type assignment. Alternatively, all exons can be used for labeling. Note that this feature is only available for genomic regions, as it is not informative for transcript regions, which would contain only exon labels. A user-defined isoform list can also be supplied, substituting the list of most prominent isoforms for annotation. Regions not overlapping with introns or exons can also be labeled separately (instead of labeled as intron).

Transcript region annotation

Similarly to the exon-intron annotation, binding regions can be labeled based on their overlap with transcript regions. Labels are assigned based on untranslated region (UTR) or coding region (CDS) overlap (5'UTR, CDS, 3'UTR, None), by taking the isoform information in the input GTF file. Again, the list of most prominent isoforms is used for annotation or, alternatively, a list of user-defined isoforms can be used. Additional annotation options include start and stop codon or transcript and exon border labeling.

Repeat region annotation

Repeat region annotation can also be added analogously to other region type annotations. This information is derived directly from the genomic sequence file (in .2bit format, from the UCSC website), where repeat regions identified by RepeatMasker and Tandem Repeats Finder are stored in lowercase letters.

Visualization

To better understand the sequence or additional feature preferences of a model, RNAProt can plot logos and whole-site profiles. Both show position-wise features for each position, and the profile plots also include a saliency map track, plus a track that visualizes the effects of single-position mutations (also known as *in silico* mutagenesis) on the whole site score. Saliency maps visualize the gradient with respect to the input for each sequence position, thus showing the importance the trained model attributes to each sequence position and also its influence on the network output [31]. In contrast, *in silico* mutagenesis treats the network as a black box, where the input sequence is mutated at each position (3 mutations possible at each position, since there are 3 non-wild-type nucleotides) and the mutated sequences are scored by the network. For example, given a sequence AC, the mutated sequences would be CC, GC, UC, AA, AG, and AU. For a sequence of length n, we thus need to generate $3*n$ mutated sequences for which to calculate scores. The score difference (mutated sequence score minus wild-type sequence score) is then plotted for each mutated nucleotide at each position, with the height of the nucleotide corresponding to the score difference. This difference can be positive (i.e., the mutation increases the whole-site score) or negative (i.e., the mutation decreases the whole-site score). This way, both visualizations help in understanding what parts in a given sequence the model regards as important.

To generate the logo, RNAProt extracts top saliency value positions from a specified number of top scoring sites, and extends them to a defined logo length. The extracted subsequences (weighted by saliency) are then converted into a weight matrix and plotted with Logomaker [32].

Tool comparison

Benchmark sets

For the tool comparison, we constructed 2 different benchmark sets. The first consists of 23 different PAR-CLIP, iCLIP, and High-throughput sequencing of RNA isolated by CLIP (HITS-CLIP) data sets (20 different RBPs) extracted from the original GraphProt publication. The second includes 30 eCLIP data sets (30 different RBPs) extracted from the Encyclopedia of DNA elements (ENCODE) website, [33]. For the GraphProt data sets, we defined a maximum number of positive and negative sites (each 5,000), and randomly selected these numbers for larger data sets. This was done since run times for DeepCLIP and DeepRAM can become very long as the number of sites increases (see the "Run time comparison" section for more details). For the eCLIP data sets, we aimed for 6,000 to 10,000 positive sites per data set during preprocessing and filtering. All sites were length-normalized to 81 nucleotides (nt) due to the fixed-size input required by DeepRAM. To generate the negative sets, we used RNAProt, which can automatically generate a set of random negative sites for a given set of positive input sites (i.e., RBP binding sites identified by CLIP-seq). By default, RNAProt randomly selects negative sites based on 2 criteria: (i) negative sites are sampled from gene regions containing positive sites; and (ii) a negative site should not overlap with any positive site. This setting was used to create the benchmark sets. The same number of random negative and positive instances was used throughout the benchmarks. More details on data preprocessing and data set construction can be found in the Supplementary Methods. For the run time comparison, we recorded single model training run times. Here, we randomly selected 5,000 positive and 5,000 negatives sites from the eCLIP RBFOX2 set, all with lengths of 81 nt, and trained each method 3 times on this set.

Tool setup and performance measurement

DeepCLIP, GraphProt, and RNAProt were benchmarked using their default parameters. For DeepRAM, we used their best-performing network architecture k-mer embedding with single layer CNN and bidirectional LSTM (ECBLSTM). The area under the receiver operating curve (AUC) was used in combination with 10-fold cross-validation to estimate and compare model generalization performances for the first 3 tools. Since DeepRAM does not offer a 10-fold cross-validation setting, we compared it separately to RNAProt using a hold-out setting (1 split with 90% of data for training and 10% for testing). For DeepCLIP, we set patience (early stopping) to 20 epochs and the maximum number of epochs to 200, which corresponds to the setting used for most data sets in the original publication. For RNAProt, we set the patience to 30 and the maximum number of epochs to 200 in cross-validation, while for the hold-out comparison we increased patience to 50, since we found that smaller data sets can sometimes benefit from increased patience. For the run time comparison, both DeepCLIP and RNAProt were set to a patience of 20 and a maximum number of epochs of 200. To signify differences in 10-fold cross-validation performance between the 3 methods, we calculated P-values using the 2-sided Wilcoxon test in R (version 3.6.2) for each data set and method combination. For comparing window prediction performances, we used the F-score (also known as F1 score or harmonic mean of precision and recall).

Computing benchmark results

To compute the benchmark results, we used 2 different desktop PCs: an AMD Ryzen7-2700X (32 GB RAM, GeForce RTX 2070 8 GB) and a Intel i7-8700k (32 GB RAM, Geforce GTX 1060 GPU 6 GB), both with Ubuntu 18.04 LTS installed. Tool run times were measured using solely the Intel i7, running single-model training 3 times and recording run times. In general, we found that RNAProt runs fine on a PC with 8 GB RAM and no GPU with the data set sizes found in the benchmark set. However, even an average consumer-grade GPU like the GTX 1060 drastically reduces run times (see the "Run time comparison" section results) and is thus recommended for on-the-fly model training (specifically an Nvidia card with $\geq$ 4 GB GPU RAM).

Results and discussion

Below, we demonstrate RNAProt's state-of-the-art performance and show its run time efficiency. In particular, we compared it to 2 recent deep-learning methods (DeepCLIP [34] and DeepRAM [35]), as well as GraphProt. We chose the first 2 because both provide usage instructions and are easy to install. Moreover, both compare favorably with many other methods in the field in their respective papers. As a reference, we also included the popular classical machine learning method GraphProt. Furthermore, we illustrate that RNAProt's built-in visualizations can uncover known RBP binding preferences, and show that additional built-in features can boost performance. Finally, we exemplify the benefits of including structure information by improving the binding site prediction quality of the stem loop binding RBP Roquin.

Cross-validation comparison

We first compared RNAProt in a standard 10-fold cross-validation setting with GraphProt and DeepCLIP, on 2 different sets of RBP data sets. The first set consists of 30 eCLIP data sets from 30 different RBPs, while the second set consists of 23 data sets from 20 RBPs, generated by various CLIP-seq protocols (see the "Benchmark sets" section for data set details). GraphProt is a popular classical machine learning method that uses a graph kernel with a Support Vector Machine classifier, while DeepCLIP is a recent deep-learning method featuring a combination of CNN and bidirectional LSTM. All 3 tools were trained using only sequence features.

Fig. 2a and b show the 10-fold cross-validation results over the 2 benchmark sets for GraphProt, DeepCLIP, and RNAProt. For both sets, RNAProt achieves the highest total average AUC (87.26% and 89.30%), followed by DeepCLIP (84.03% and 87.00%), and GraphProt (81.71% and 83.81%). We note that both deep-learning methods outperform GraphProt on both sets. To signify performance differences between 2 methods, we calculated the 2-sided Wilcoxon test on the AUC distributions for each method combination and each of the 53 data sets (see Supplementary Tables S3 and S4 for AUCs and *P*-values). Fig. 2c and d contrast the single data set AUCs of GraphProt with RNAProt (Fig. 2c) and DeepCLIP with RNAProt (Fig. 2d), coloring significantly better method AUCs (GraphProt: red; DeepCLIP: yellow; RNAProt: blue). We can see that RNAProt outperforms GraphProt in 49 cases and DeepCLIP in 42 cases, while DeepCLIP and GraphProt both only perform better on 2 data sets. The 2 data sets are the same for both methods (ALKBH5, C17ORF85), which are from the original GraphProt publication. We can only speculate here that RNAProt's lower performance might be due to some intrinsic incompatibilities of the data set and the utilized RNN network. As for the largely lower performances of DeepCLIP, we assume that it is possible to tune its hyperparameters (e.g., CNN filter or regularization settings) to increase its performance. Out of the box, however, RNAProt clearly outperforms DeepCLIP. Moreover, DeepCLIP has a clear disadvantage regarding run time (see the "Run time comparison" section below).

Hold-out validation comparison

We also compared results to DeepRAM, a tool which allows the testing of various deep neural network architectures to compare their performances on DNA or RNA sequence data derived from chromatin immunoprecipitation with high-throughput sequencing (ChIP-seq) or CLIP-seq. For the comparison, we chose their best-performing architecture (ECBLSTM), a Word2Vec embedding of the input sequence (k-mer length = 3, stride = 1), followed by 1 CNN layer and 1 bidirectional LSTM layer. Since DeepRAM does not support cross-validation, we used a hold-out setting (i.e., 1 train-test split) for comparison, where models were trained on 90% of the data and tested on the remaining 10% for each data set. Note that we ran RNAProt with default hyperparameters, while DeepRAM does not offer default hyperparameters and requires hyperparameter optimization for each training run. We therefore manually reduced the number of random search iterations from 40 to 20 inside the DeepRAM code, to make the comparison more fair and run times more bearable. By this, the run time for a data set with 10,000 instances (81 nt long) got reduced to 5–6 hours, while for the same set RNAProt needs 1–2 minutes.

Fig. 3 shows the hold-out results over the 2 benchmark sets for DeepRAM and RNAProt. As we can see, average hold-out AUC performances of the 2 methods are very close for the 2 sets (DeepRAM: 87.42% and 89.28%; RNAProt: 87.50% and 89.34%). Again, there are only 2 data sets (ALKBH5, C17ORF85) where RNAProt performance drops considerably compared to DeepRAM, consistent with the cross-validation results above. For the remaining 51 data sets, there can be differences of 2% to 3% (both ways) but in general the performance is very similar (for full results, see Supplementary Tables S5 and S6). We thus can conclude that for the given data sets, there is no real advantage of using a more complex architecture like DeepRAM's ECBLSTM.

As shown in the DeepRAM paper, more complex architectures like ECBLSTM can benefit from larger data sets (>10,000 positive instances). As our benchmark data sets contain between 1,338 and 9,206 positive sites (on average 6,389.4), ECBLSTM might perform better as data set sizes increase. However, >10,000 sites is often not a realistic estimate of the real number of RBP binding sites coming from a CLIP-seq experiment. For example, in order to get a high-confidence set of RBP binding sites from an eCLIP data set, the ENCODE consortium advises use of a strict filtering routine [36], leaving often only a few thousand sites, if not less, for subsequent analysis and model training. In addition, as pointed out in the DeepRAM paper, more complex models tend to be harder to interpret. On top of that, high test set performance does not guarantee that the model learned something biologically meaningful. We are also facing a trade-off between accuracy, interpretability, and run time. Depending on the application, the user might prefer a faster or a more accurate method, or they might care more about the interpretation of the prediction. In this regard, it would be interesting to explore in future studies whether ensemble predictions (including various more interpretable and more complex models) could help to combine individual model strengths.

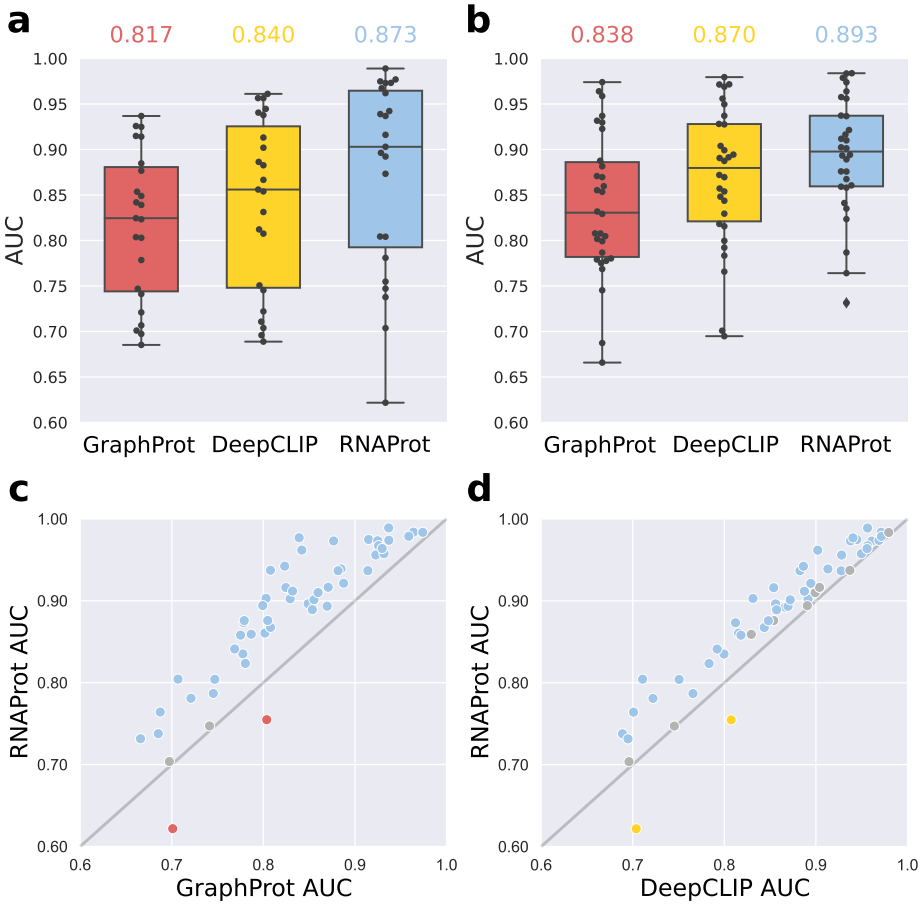


Figure 2: The 10-fold cross-validation results for GraphProt, DeepCLIP, and RNAProt. (a) Results for the first benchmark set contain 23 CLIP-seq data sets from 20 different RBPs and various CLIP-seq protocols (average method AUCs on top). (b) Results for the second benchmark set contain 30 eCLIP data sets from 30 different RBPs (average method AUCs on top). (c) Comparing single data set AUCs between GraphProt and RNAProt for all 53 data sets, the blue dots indicate a significantly better AUC for RNAProt ($n = 49$), the gray dots indicate no significant difference ($n = 2$), and the red dots indicate a significantly better AUC for GraphProt ($n = 2$). (d) Comparing single data set AUCs between DeepCLIP and RNAProt for all 53 data sets, the blue dots indicate a significantly better AUC for RNAProt ($n = 42$), the gray dots indicate no significant difference ($n = 9$), and the yellow dots indicate a significantly better AUC for DeepCLIP ($n = 2$). A 2-sided Wilcoxon test was used to calculate P-values (significance threshold = 0.05).

Run time comparison

Model training is known to be the computationally most expensive part of working with deep neural networks. We therefore compared the times it takes to train a single model with DeepCLIP, RNAProt, and, as a reference, the classical machine learning method GraphProt. Note that DeepRAM always runs a hyperparameter optimization for model training, making it unsuitable for this comparison. Specifically, we took 10,000 training instances (5,000 positives) of length 81 nt from the RBFOX2 eCLIP data set and trained a sequence model for all 3 methods (3 times each). We used default parameters for all methods, and for DeepCLIP and RNAProt set the patience and maximum number of epochs to 20 and 200, respectively (also see the "Computing benchmark results" section).

Fig. 4 shows the obtained average training times for DeepCLIP, RNAProt (CPU and GPU modes), and GraphProt (for full results, see Supplementary Table S7). We note that GraphProt model training is the fastest, at 40.3 seconds, followed by RNAProt (GPU) at 72 seconds, RNAProt (CPU) at 8 minutes, and DeepCLIP at 37.4 minutes. In other words, RNAProt GPU is 31 times faster (RNProt CPU 4.7 times faster) than DeepCLIP. This clearly shows RNAProt's ability for on-the-fly model training, as well as the benefit of using a GPU (even an average consumer-grade GPU as described here). Since RNAProt supports many different features and settings, fast model training allows the user to try different settings for a specific task in a short amount of time. As for the run time difference, it seems that DeepCLIP currently does not support GPU computing, or at least we could not find any hints in the code. This would explain the slow run time, which unfortunately makes it less useful for on-the-fly training and testing. Still, its run times are much more practical than the ones we got with DeepRAM: due to its hard-coded hyperparameter optimization, DeepRAM can easily take 12 hours for model training (with the default number of random search iterations and benchmark data set sizes), even though it uses GPU computing.

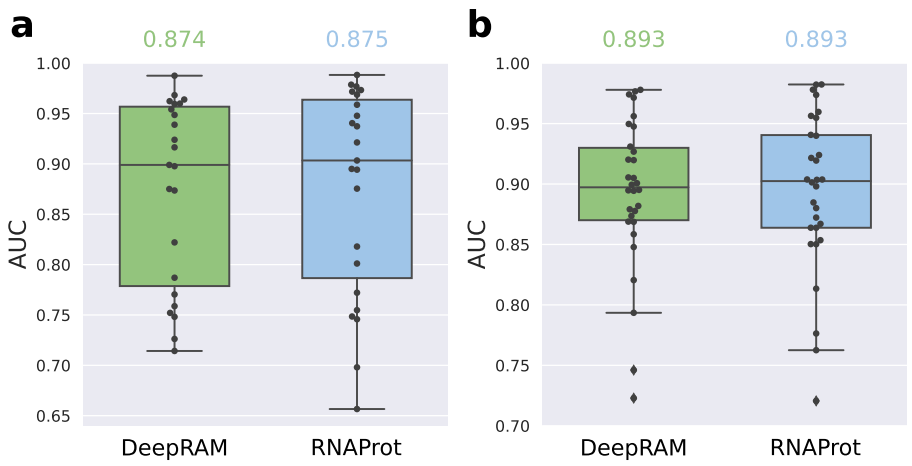


Figure 3: Hold-out validation results for DeepRAM and RNAProt. (a) Results for the first benchmark set contain 23 CLIP-seq data sets from 20 different RBPs and various CLIP-seq protocols. (b) Results for the second benchmark set contain 30 eCLIP data sets from 30 different RBPs. For both sets, we also report the average method AUC on top.

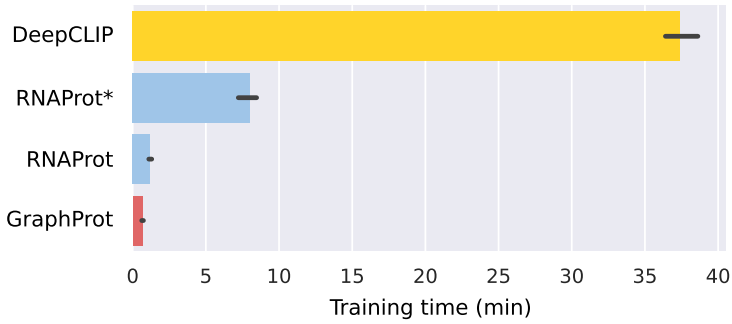


Figure 4: Model training time comparison. Training times are in minutes (averaged over 3 runs) for training a single model with 10,000 instances (81 nt) for GraphProt, RNAProt, and DeepCLIP. ∗RNAProt using CPU only for calculations (no GPU).

Visualizations capture known binding preferences

As deep-learning models are complex by design and thus hard to interpret, the development of visualizations that help to explain what is learned by a model is an important and active area of research. For RNAProt, we chose to visualize position-wise importances using 2 approaches: saliency maps and *in silico* mutagenesis (see the "Visualization" section for details).

To compare RNAProt sequence logos and profiles with known RBP binding preferences from the literature, we trained sequence models on 6 different RBP data sets with known binding preferences. Fig. 5 shows the obtained sequence logo and known preferences (based on RBP motifs listed in the ATtRACT database [37]), as well as the top scoring training site profile for each RBP. As we can see, the logos clearly capture the literature preferences, both for RBPs without a single dominant motif (hnRNPK, KHDRBS1, PTBP1, SRSF1) and for RBPs with strong individual motifs (QKI, RBFOX2). This shows that saliency can be used to extract meaningful logos, which provide a rough idea about global model preferences. In addition, the saliency and mutation tracks give clues to local position-wise preferences. As shown, both match literature knowledge, but can also give interesting new insights. For example, important positions for the first 3 RBPs are more scattered in the observed profiles, while for QKI and RBFOX2 the model pays much more attention to the precise binding motif locations, with other positions having little effect on the model prediction. Both tracks are thus helpful to understanding local model decisions, but they are only informative for individual sites. To better understand global model preferences, we hope to integrate new visualizations in the near future, since this is also a very active area of research, albeit less mature than work on local preferences [38].

Additional features boost performance

Since RNAProt supports various additional features on top of the sequence information, we also checked how including these features in training influences model performance. When generating training sets with RNAProt, the user can specify which features to compute and then, for training, can select which feature information the model should be trained on (see the "Supported features" section for details). For the comparison, we used RNA secondary structure, phastCons conservation scores, phyloP conservation scores, exon-intron annotation, and a combination of exon-intron and conservation scores.

Fig. 6 shows the 10-fold cross-validation results for the 2 benchmark sets, for each described feature. We observe that the conservation and exon-intron features can, depending on

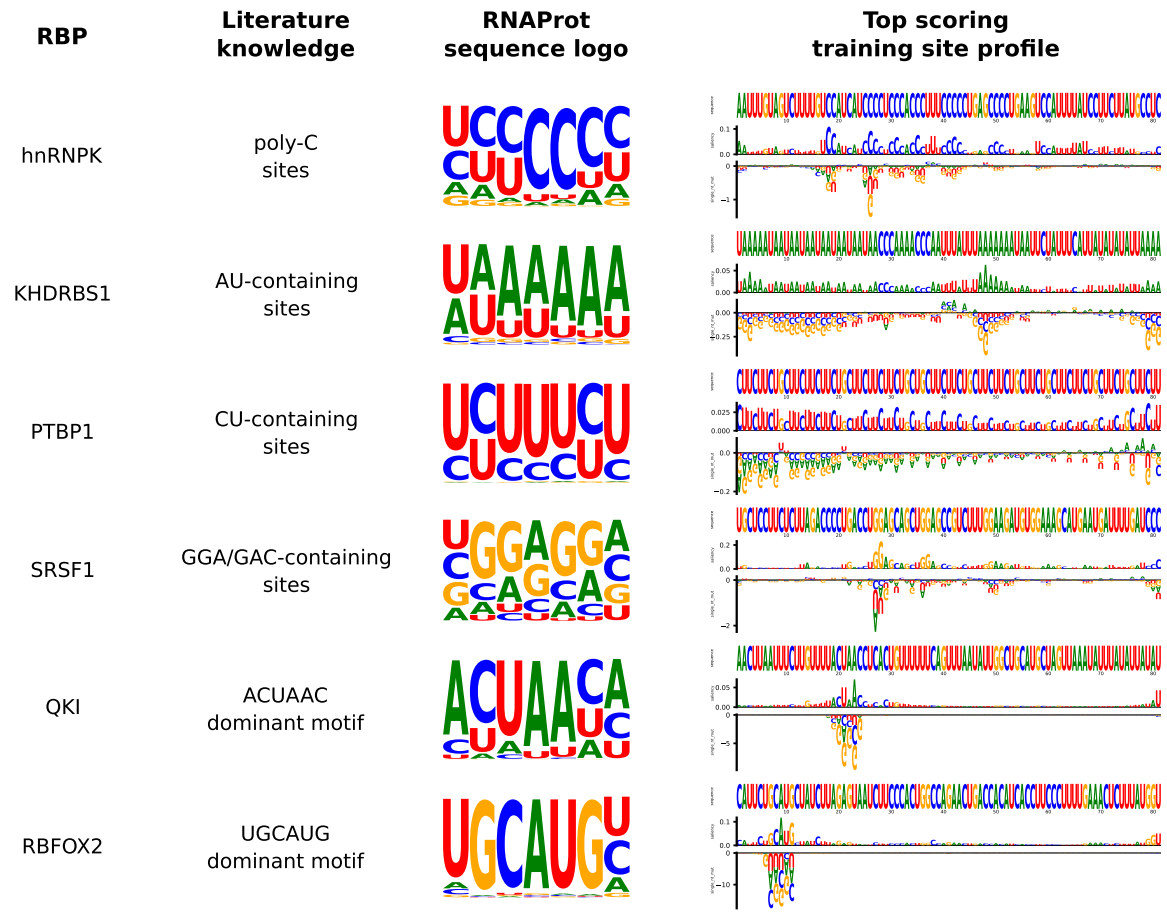


Figure 5: Comparison of RNAProt sequence logos and profiles with known RBP binding preferences. Literature knowledge was obtained from the ATtRACT database [37]. All models were trained using only the sequence feature. Logos were generated by extracting the top site saliency positions for each of the top 200 scoring training sites, and extending them by 3 on each side to generate logos of length 7. Logo character heights correspond to their respective saliency values at each of the 7 positions. On the right site, profiles for the top scoring training sites are shown, offering several tracks: the site nucleotide sequence, the position-wise saliency, and single mutation effects. The single mutations track shows how much every possible single nucleotide mutation at each position changes the total site score (positive or negative).

the data set, strongly boost model performance on the benchmark sets. As for the structure feature, individual data set performances are usually very similar between structure and sequence-only models (see Supplementary Tables S1 and S2 for full results), although for the eCLIP set the overall performance with structure is slightly higher (89.41% vs 89.30% for the sequence-only model). We assume that this can be further tuned on the data set level by changing the structure calculation settings of RNAProt (different modes available, plus RNAplfold settings for window length, maximum base pair span, and mean probability region length). As for region type and conservation features, these performances of course highly depend on the selected negative regions. For example, using exon-intron annotations with negative regions located only inside introns and positive regions with a high amount of exonic sites will naturally lead to higher performance. But this does not make the model more useful. Thus, what the focus of the prediction should be is important. If the prediction should be on transcripts only, then exon-intron distinction becomes meaningless. However, some intrinsic bias of an RBP regarding regions can also be natural and of interest, such as when predicting on gene sequences containing introns and exons. In this regard, RNAProt offers several options to control negatives selection: users can either supply their own negative regions or the sampling of negative regions can be further specified by excluding certain genomic or transcript regions (see documentation for details).

Regarding the tested features, note that we did not include transcript or repeat region annotations in the comparison. As for the first feature, our tests showed performances similar to exon-intron inclusion, but we think that this feature needs an accurate (i.e., condition-specific) CDS and UTR region annotation to make sense. In line with this, it has been shown that context choice (i.e., selecting the authentic transcript or genomic context surrounding binding sites) affects the performances of RBP binding site prediction tools [39]. As RNAProt supports both genomic and transcript region annotations, it can easily be combined with isoform detection tools in future workflows. Regarding repeat region annotations, it did not make sense to test this feature since the eCLIP pipeline that produced the benchmark set binding sites only considers uniquely mapped reads. How-

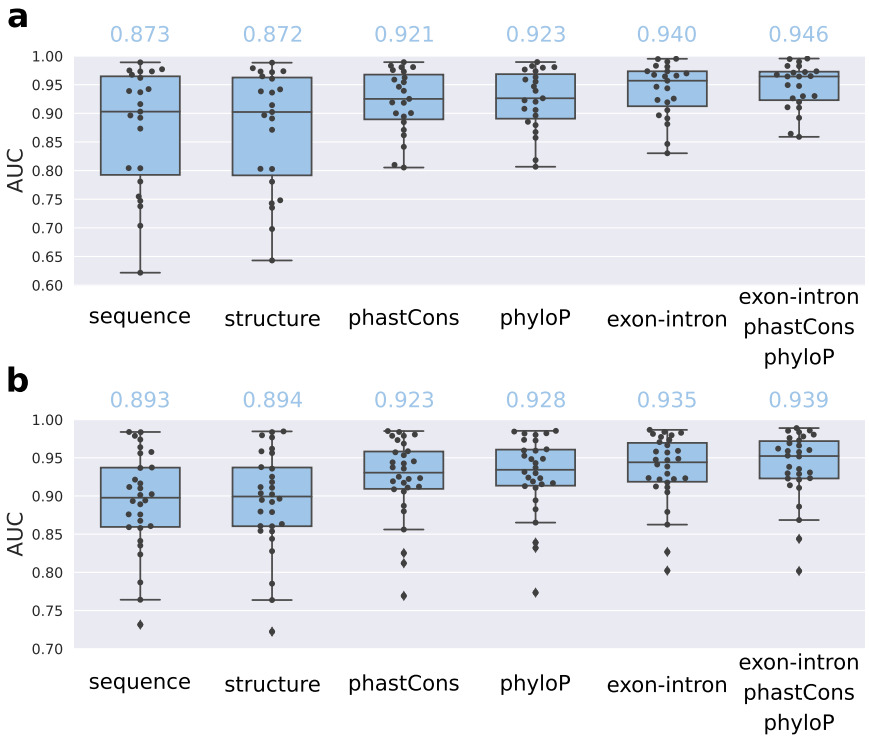


Figure 6: The 10-fold cross-validation results for RNAProt models trained with additional features. (a) Results for the first benchmark set contain 23 CLIP-seq data sets from 20 different RBPs and various CLIP-seq protocols. (b) Results for the second benchmark set contain 30 eCLIP data sets from 30 different RBPs. The "sequence" is included for reference, using only sequence information for training. For both sets, we report the average AUC with included additional feature(s) on top.

ever, a recent pipeline update [36] now also allows mapping to certain repeat elements and has already led to the discovery of many new RBP binding sites overlapping with these elements. Repeat region annotation could thus become an informative feature once these data sets are available.

Structure information can increase specificity

Given that additional features can increase predictive performance, we next checked whether they also can help in a more practical scenario. For this, we downloaded a data set consisting of predicted structurally conserved binding sites of the RBP Roquin (also termed constitutive decay elements [CDEs]) [40]. The CDEs were predicted using a biologically verified consensus structure consisting of a 6–8 bp long stem capped with a YRN (Y: C or U; R: A or G; N: any base) tri-nucleotide loop, including all human 3'UTRs as potential target regions. After preprocessing and training set generation (same number of random negatives; 81 nt site length), we trained a structure and a sequence model on the resulting 2,271 CDEs. For the structure prediction, we used an RNAplfold window length of 70 nt, a maximum base pair span of 50 nt, and a mean probability region length of 3 (see Supplementary Methods for more details).

Comparing the 10-fold cross-validation results of the 2 models, the sequence model achieves an average AUC of 79.22%, while the structure model performs almost 20% better (99.02%). We also note a high standard deviation for the individual sequence model AUC (7.66%), which is not the case for the structure model (0.43%). This means that the sequence model has problems with consistently classifying the test sites correctly, while the added structure information almost completely resolves this issue. We can thus conclude that the addition of structure information allows us to predict the given set of potential CDEs with high accuracy. As a reference, we also trained 2 GraphProt models (1 with sequence and 1 with structure information), which resulted in average AUCs of 70.81% and 78.49%, respectively.

To complete the use case, the authors also experimentally verified 2 CDEs in the 3'UTR of the UCP3 gene (transcript ID ENST00000314032.9; length 2,277 nt). We therefore trained another structure model, excluding the 2 sites from the training set, and ran RNAProt using its window (profile) prediction mode on the transcript. Fig. 7a shows the transcript, along with verified and predicted CDEs. We note that our model predicts 4 CDEs in total (all in the 3'UTR), with 2 of them perfectly overlapping the verified CDEs. Fig. 7b shows the profile of the second site (compare to the red hairpin in Fig. 1C of Braun et al. [40]), with saliencies and the single mutations track highlighting the hairpin loop portion and parts of the surrounding stem. The stem loop can also be recognized in the structural elements track on the bottom. The single mutations track (measuring effects of single nucleotide changes on the whole-site score) indicates that the loop nucleotides are a particularly important sequence feature. In contrast, the structure feature contributes more to the area surrounding the loop, by providing the stem information. This again matches what is known about Roquin binding, with few sequence preferences in the hairpin aside from the described loop preferences. As a reference, we also trained a se-

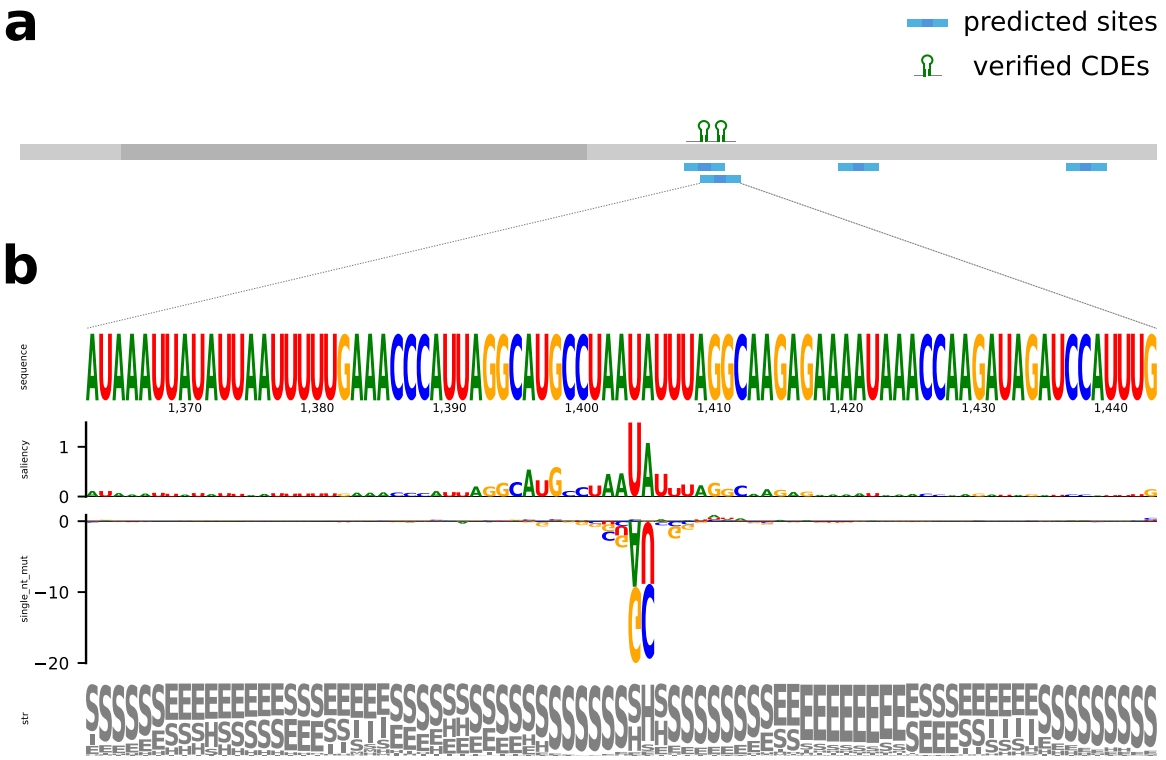


Figure 7: Roquin structure model predictions on the UCP3 gene transcript ENST00000314032.9. (a) The ENST00000314032.9 transcript (length 2,277 nt; 5' and 3' untranslated regions (UTRs) in light gray, coding sequence (CDS) in dark gray) is displayed together with verified and predicted Roquin binding sites (CDEs). (b) RNAProt site profile for the second verified CDE is shown with sequence, saliency map, single mutations, and structural elements tracks.

quence model (validation set AUC 94.73%) and predicted CDEs on the transcript. This resulted in 18 predictions, with only 1 overlapping the first verified site, despite the very good validation AUC. This clearly demonstrates how additional features like structure information can help to make predictions more specific (F-score, sequence model = 0.10; F-score, structure model = 0.67).

Conclusion

In this article we presented RNAProt, an RBP binding site prediction framework based on RNNs. Devised as an end-to-end method, RNAProt includes all necessary functionalities, from data set generation over model training to the evaluation of binding preferences and binding site prediction. We compared it to other popular tools in the field, showing its state-of-the-art performance and improved run time efficiency. The short training times allow for on-the-fly model training, which is great for quickly testing hypotheses regarding data set, parameter, or feature selections. Moreover, RNAProt is currently the most flexible method when it comes to supported position-wise features for learning, as well as input data types. RNAProt is easy to install and use, assisted by comprehensive documentation. Furthermore, it provides comprehensive statistics and visualizations, informing the user about data set characteristics and learned model properties. All this makes RNAProt a valuable tool to apply and include in RBP binding site analysis workflows.

Availability of source code and requirements

- Project name: RNAProt
- Project page: https://github.com/BackofenLab/RNAProt
- Operating system(s): Linux
- Programming language: Python
- Other requirements: Anaconda
- Installation: `conda install -c bioconda rnaprot`
- License: MIT
- biotools ID: biotools:rnaprot
- RRID: SCR_021218

Data Availability

All benchmark and training data sets used to create the reported results can be downloaded from Zenodo [41]. Supplementary Methods and Tables can be found on the *GigaScience* website and on GitHub [24]. A code snapshot as well as Supplementary Data are also available via GigaDB [42].

Additional Files

Supplementary Table S1: 10-fold cross validation results for GraphProt, DeepCLIP, RNAProt, and RNAProt with additional features. Results for the first benchmark set, containing 23 CLIP-seq datasets from 20 different RBPs and various CLIP-seq protocols.
Supplementary Table S2: 10-fold cross validation results for GraphProt, DeepCLIP, RNAProt, and RNAProt with additional fea-

tures. Results for the second benchmark set, containing 30 eCLIP datasets from 30 different RBPs.
Supplementary Table S3: 10-fold cross validation single fold AUC results for GraphProt, DeepCLIP, and RNAProt. Benchmark Set 1: Set from Table S1 (23 datasets). Benchmark Set 2: Set from Table S2 (30 datasets).
Supplementary Table S4: Two-sided Wilcoxon Test on Table S3 single fold AUCs, to determine significantly different AUCs between methods and single datasets. Calculated p-values for two method comparions are shown: RNAProt vs. GraphProt, and RNAProt vs. DeepCLIP.
Supplementary Table S5: Hold out validation results for DeepRAM and RNAProt. Results for the first benchmark set, containing 23 CLIP-seq datasets from 20 different RBPs and various CLIP-seq protocols.
Supplementary Table S6: Hold out validation results for DeepRAM and RNAProt. Results for the second benchmark set, containing 30 eCLIP datasets from 30 different RBPs.
Supplementary Table S7: Single model training runtime comparison for GraphProt, DeepCLIP, and RNAProt. Runtime is given in minutes (min), together with the mean runtime over three runs for each method.
Supplementary methods
Dataset construction
Cross validation comparison
Hold-out comparison
Roquin CDE dataset preparation and prediction
Runtime comparison

List of abbreviations

AUC: area under the receiver operating curve; eCLIP: enhanced CLIP; CDE: constitutive decay element; CDS: coding region; CLIP-seq: cross-linking and immunoprecipitation followed by next-generation sequencing; CNN: convolutional neural network; iCLIP: individual-nucleotide resolution UV cross-linking and immunoprecipitation; LSTM: Long Short-Term Memory; PAR-CLIP: photoactivatable-ribonucleoside-enhanced crosslinking and immunoprecipitation; RBP: RNA-binding protein; RNN: Recurrent neural network; UTR: untranslated region.

Acknowledgements

We thank Martin Raden for his invaluable suggestions on the topic.

Competing Interests

The authors declare that they have no competing interests.

Funding

MU was funded by Deutsche Forschungsgemeinschaft (DFG) grants BA 2168/11-1 SPP 1738 and BA2168/11-2 SPP 1738 and by the Bundesministerium für Bildung und Forschung, RNAProNet-031L0164B. VDT was funded by DFG grant BA 2168/3-3. FH was funded by DFG grant 322977937/GRK2344 2017 MeInBio–BioInMe Research Training Group. This study was supported by the DFG under Germany's Excellence Strategy (CIBSS - EXC-2189 - Project ID 390939984).

Author's Contributions

RB, VDT, and MU conceived the study. FH contributed Wilcoxon test *P*-values and DeepCLIP cross-validation results. VDT and MU implemented the network part. MU performed the remaining data analysis, wrote the draft, and implemented the tool. All authors reviewed and approved the final manuscript.

REFERENCES

1. Gerstberger S, Hafner M, Tuschl T. A census of human RNA-binding proteins. Nat Rev Genet 2014; **15**(12):829.
2. Brannan KW, Jin W, Huelga SC, et al. SONAR discovers RNA-binding proteins from analysis of large-scale protein-protein interactomes. Mol Cell 2016; **64**(2):282–93.
3. Hentze MW, Castello A, Schwarzl T, et al. A brave new world of RNA-binding proteins. Nat Rev Mol Cell Biol 2018; **19**(5):327.
4. Liu L, Li T, Song G, et al. Insight into novel RNA-binding activities via large-scale analysis of lncRNA-bound proteome and IDH1-bound transcriptome. Nucleic Acids Res 2019; **47**(5):2244–62.
5. Gerstberger S, Hafner M, Ascano M, et al. Evolutionary conservation and expression of human RNA-binding proteins and their role in human genetic disease. In: Yeo GW , ed. Systems biology of RNA binding proteins. Springer: New York; 2014:1–55.
6. Pereira B, Billaud M, Almeida R. RNA-binding proteins in cancer: old players and new actors. Trends Cancer 2017; **3**(7):506–28.
7. Conlon EG, Manley JL. RNA-binding proteins in neurodegeneration: mechanisms in aggregate. Genes Dev 2017; **31**(15):1509–28.
8. Licatalosi DD, Mele A, Fak JJ, et al. HITS-CLIP yields genome-wide insights into brain alternative RNA processing. Nature 2008; **456**(7221):464.
9. Hafner M, Landthaler M, Burger L, et al. Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP. Cell 2010; **141**(1):129–41.
10. König J, Zarnack K, Rot G, et al. iCLIP reveals the function of hnRNP particles in splicing at individual nucleotide resolution. Nat Struct Mol Biol 2010; **17**(7):909.
11. Van Nostrand EL, Pratt GA, Shishkin AA, et al. Robust transcriptome-wide discovery of RNA-binding protein binding sites with enhanced CLIP (eCLIP). Nat Methods 2016; **13**(6):508.
12. Uhl M, Houwaart T, Corrado G, et al. Computational analysis of CLIP-seq data. Methods 2017; **118**: 60–72.
13. Uren PJ, Bahrami-Samani E, Burns SC, et al. Site identification in high-throughput RNA–protein interaction data. Bioinformatics 2012; **28**(23):3013–20.
14. Lovci MT, Ghanem D, Marr H, et al. Rbfox proteins regulate alternative mRNA splicing through evolutionarily conserved RNA bridges. Nat Struct Mol Biol 2013; **20**: 1434.
15. Krakau S, Richard H, Marsico A. PureCLIP: capturing target-specific protein–RNA interaction footprints from single-nucleotide CLIP-seq data. Genome Biol 2017; **18**(1):240.
16. Kornienko AE, Dotter CP, Guenzl PM, et al. Long non-coding RNAs display higher natural expression variation than protein-coding genes in healthy humans. Genome Biol 2016; **17**(1):14.

17. Ferrarese R, Harsh GR, Yadav AK, et al. Lineage-specific splicing of a brain-enriched alternative exon promotes glioblastoma progression. J Clin Invest 2014; **124**(7): 2861–76.
18. Kazan H, Ray D, Chan ET, et al. RNAcontext: a new method for learning the sequence and structure binding preferences of RNA-binding proteins. PLoS Comput Biol 2010; **6**(7):e1000832.
19. Maticzka D, Lange SJ, Costa F, et al. GraphProt: modeling binding preferences of RNA-binding proteins. Genome Biol 2014; **15**(1):R17.
20. Alipanahi B, Delong A, Weirauch MT, et al. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. Nat Biotechnol 2015; **33**(8):831.
21. Pan X, Yang Y, Xia CQ, et al. Recent methodology progress of deep learning for RNA–protein interaction prediction. Wiley Interdiscip Rev RNA 2019; **10**(6):e1544.
22. Mölder F, Jablonski KP, Letcher B, et al. Sustainable data analysis with Snakemake. F1000Research 2021; **10**:33.
23. Afgan E, Baker D, Batut B, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res 2018; **46**(W1): W537–44.
24. Uhl M. RNAProt: an efficient and feature-rich RNA binding protein binding site predictor. GitHub repository 2021; https://github.com/BackofenLab/RNAProt. Accessed: 9 August 2021.
25. Falkner S, Klein A, Hutter F. BOHB: robust and efficient hyperparameter optimization at scale. In: Program and Abstracts of the International Conference on Machine Learning PMLR. Proceedings of the 35th International Conference on Machine Learning, PMLR 80: 10-15 July 2018, Stockholm: Sweden; 2018. p. 1437–46.
26. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997; **9**(8):1735–80.
27. Cho K, Van Merriënboer B, Bahdanau D, et al. On the properties of neural machine translation: encoder-decoder approaches. arXiv 2014; arXiv:1409.1259.
28. Loshchilov I, Hutter F. Decoupled weight decay regularization. arXiv 2017; arXiv:1711.05101.
29. Lorenz R, Bernhart SH, Zu Siederdissen CH, et al. ViennaRNA Package 2.0. Algorithms Mol Biol 2011; **6**(1):1–14.
30. Quinlan AR, Hall IM. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics 2010; **26**(6): 841–2.
31. Li J, Chen X, Hovy E, et al. Visualizing and understanding neural models in nlp. arXiv 2015; arXiv:1506.01066.
32. Tareen A, Kinney JB. Logomaker: beautiful sequence logos in Python. Bioinformatics 2020; **36**(7):2272–4.
33. Sloan CA, Chan ET, Davidson JM, et al. ENCODE data at the ENCODE portal. Nucleic Acids Res 2015; **44**(D1):D726–32.
34. Grønning AGB, Doktor TK, Larsen SJ, et al. DeepCLIP: predicting the effect of mutations on protein–RNA binding with deep learning. Nucleic Acids Res 2020; **48**(13):7099–118.
35. Trabelsi A, Chaabane M, Ben-Hur A. Comprehensive evaluation of deep learning architectures for prediction of DNA/RNA sequence binding specificities. Bioinformatics 2019; **35**(14):i269–77.
36. Van Nostrand EL, Pratt GA, Yee BA, et al. Principles of RNA processing from analysis of enhanced CLIP maps for 150 RNA binding proteins. Genome Biol 2020; **21**:1–26.
37. Giudice G, Sánchez-Cabo F, Torroja C, et al. ATtRACT–a database of RNA-binding proteins and associated motifs. Database (Oxford) 2016; **2016**:baw035.
38. Koo PK, Ploenzke M. Deep learning for inferring transcription factor binding sites. Curr Opin Syst Biol 2020; **19**: 16–23.
39. Uhl M, VD Tran, Backofen R. Improving CLIP-seq data analysis by incorporating transcript information. BMC Genomics 2020; **21**:894.
40. Braun J, Fischer S, Xu ZZ, et al. Identification of new high affinity targets for Roquin based on structural conservation. Nucleic Acids Res 2018; **46**(22):12109–25.
41. Uhl M. RNAProt: an efficient and feature-rich RNA binding protein binding site predictor. Zenodo 2021. http://doi.org/10.5281/zenodo.4647254.
42. Uhl M, Tran VD, Heyl F, et al. Supporting data for "RNAProt: An efficient and feature-rich RNA binding protein binding site predictor." GigaScience Database 2021. http://dx.doi.org/10.5524/100916.

[P4] Improving CLIP-seq data analysis by incorporating transcript information

Publication:

- [P4] **Michael Uhl**, Van Dinh Tran, and Rolf Backofen. **Improving CLIP-seq data analysis by incorporating transcript information**. *BMC Genomics*, 2020.

Contributions of individual authors:

"I am the main contributor to this work. I conceived the study together with Rolf Backofen. I performed the data analysis, implemented the software, wrote the online manual and the manuscript. Van Dinh Tran performed binding site predictions for DeepBind. All authors reviewed and approved the final manuscript."

Michael Uhl

The following co-authors confirm the above-stated contributions:

Uhl *et al. BMC Genomics* (2020) 21:894
https://doi.org/10.1186/s12864-020-07297-0

BMC Genomics

RESEARCH ARTICLE Open Access

Improving CLIP-seq data analysis by incorporating transcript information

Michael Uhl[1], Van Dinh Tran[1] and Rolf Backofen[1,2*]

Abstract

Background: Current peak callers for identifying RNA-binding protein (RBP) binding sites from CLIP-seq data take into account genomic read profiles, but they ignore the underlying transcript information, that is information regarding splicing events. So far, there are no studies available that closer observe this issue.

Results: Here we show that current peak callers are susceptible to false peak calling near exon borders. We quantify its extent in publicly available datasets, which turns out to be substantial. By providing a tool called CLIPcontext for automatic transcript and genomic context sequence extraction, we further demonstrate that context choice affects the performances of RBP binding site prediction tools. Moreover, we show that known motifs of exon-binding RBPs are often enriched in transcript context sites, which should enable the recovery of more authentic binding sites. Finally, we discuss possible strategies on how to integrate transcript information into future workflows.

Conclusions: Our results demonstrate the importance of incorporating transcript information in CLIP-seq data analysis. Taking advantage of the underlying transcript information should therefore become an integral part of future peak calling and downstream analysis tools.

Keywords: CLIP-seq, eCLIP, Peak calling, RBP binding site prediction

Background

Over the last decade, CLIP-seq (cross-linking and immunoprecipitation followed by next generation sequencing) [1] has become the state-of-the-art procedure to experimentally determine the precise transcriptome-wide binding locations of RNA-binding proteins (RBPs). Many variants have been introduced, out of which PAR-CLIP [2], iCLIP [3], and eCLIP [4] are currently the most widely used. Regardless of the variant, CLIP-seq is usually applied in vivo to a specific RBP, producing a library of reads bound by the RBP. Identification of binding sites is subsequently achieved by mapping the reads back to the corresponding reference genome and running a so called peak caller tool on the read profiles. A number of popular peak callers have emerged over the years, such as Piranha [5], CLIPper [6], PEAKachu [7], and PureCLIP [8].

While there exist various protocol-specific as well as more generic peak callers [9], none of the current tools takes into account the transcript information underlying the mapped reads. Instead, they extract binding regions directly from the genomic read profiles. This can be acceptable if the studied RBP binds intronic sequences or in general unspliced RNAs. However, if the RBP is actually predominantly binding to spliced RNAs, which should be true for most cytoplasmically active RBPs, ignoring transcript information potentially leads to false peak calling and the inclusion of non-authentic sequence context. This in turn can compromise the results of downstream analysis tools like motif finders or binding site predictors, which usually take the genomic sequence context for extending the binding sites as well.

Here we show that current peak callers indeed have problems with correctly defining binding sites for RBPs binding predominantly to exonic regions. We further

*Correspondence: backofen@informatik.uni-freiburg.de
[1]Bioinformatics Group, Department of Computer Science, University of Freiburg, Georges-Köhler-Allee 106, 79110 Freiburg, Germany
[2]Signalling Research Centres BIOSS and CIBSS, University of Freiburg, Schaenzlestr. 18, 79104 Freiburg, Germany

look at publicly available eCLIP datasets with binding sites identified by CLIPper and present comprehensive statistics regarding exonic binding frequencies. Focusing specifically on sites near exon borders, we report the extent of sites mostly affected by context sequence selection and false peak calling. To compare different sequence contexts, we implemented a tool called CLIPcontext. CLIPcontext automatically extracts the transcript and genomic context for a given set of transcript or genomic sites, and also offers other useful functions such as identifying sites at exon borders or motif search. We then trained three different binding site prediction tools on sites near exon borders, and demonstrate that sequence context choice can have a large impact on predictive performance. Moreover, we show for a selection of predominantly exon-binding RBPs that known motifs are enriched in transcript context sequences, enabling the identification of more authentic binding sites. In the end, we discuss possible ways on how to integrate transcript information in order to improve CLIP-seq data analysis workflows.

Results and discussion

Ignoring transcript information compromises peak calling

To illustrate the issues current peak callers have with predominantly exon-binding RBPs, we chose one out of many eCLIP RBP cell type combinations (YBX3 K562) with large amounts of exonic binding regions (see Table S1 for eCLIP overlap statistics). In this paper, we call or count peak regions as overlapping or exon binding if they have an overlap of ≥90% with exonic regions. 84.6% of YBX3 K562 merged peak sites overlap with exonic regions, out of which 51.0% are ≤50 nt away from exon borders. Figure 1 shows the YBX3 K562 genomic reads profile visualized via IGV (Integrative Genomics Viewer) [10] over two genomic regions, with added peak regions from CLIPper, CLIPper IDR, PEAKachu, and PureCLIP (see Methods section "Peak caller setup"). Figure 1a depicts a genomic region of 11 kb, containing the *PRDX6* gene. We can see that the read alignments clearly follow the exon annotations: most reads map to exons, including many intron-spanning ones (blue-gray lines), while only few reads map to introns. Not surprisingly, all three peak callers only report exonic peaks, often close or directly at exon borders. Given the alignment information, extending these peak regions with genomic context, as usually done prior to further analysis, is not correct. Instead, the transcript context of the spliced RNA should be used, which is where the actual RBP binding occurs. Zooming in on the matter, Fig. 1b shows a genomic region of 563 bp, comprising exon 6 and 7 of the *DDOST* gene. Again the mapped reads strongly suggest a spliced RNA context, given the many intron-spanning reads and almost no intron coverage. Keeping the intron therefore leads to an artificial split-up of peak regions spanning the exon border. Unaware of the split, peak callers might consequently call two peaks, whereas they should have treated the split peaks as one contiguous region.

In the Fig. 1b example, both CLIPper and PureCLIP call peaks at adjacent exon borders, while PEAKachu

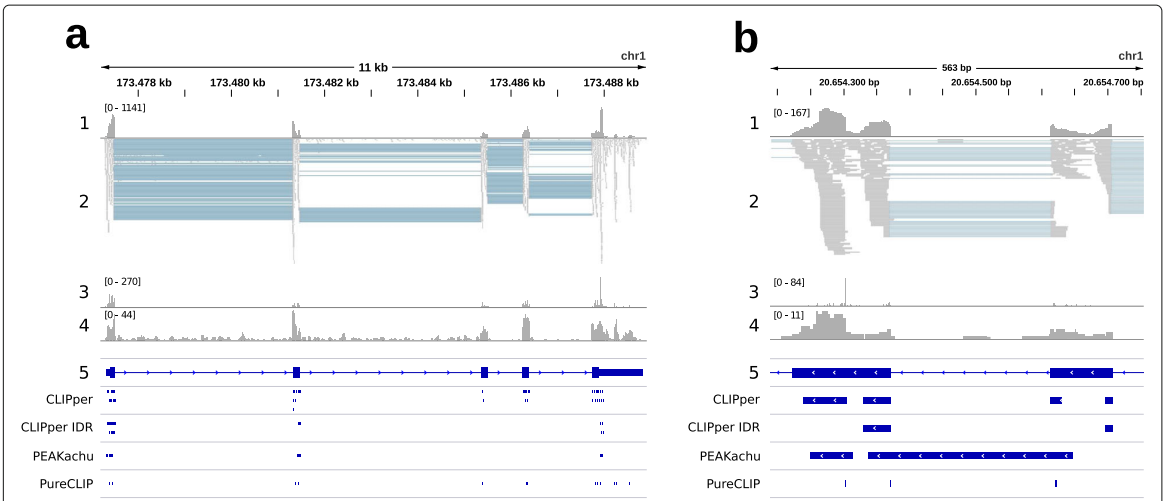


Fig. 1 IGV snapshot of two genomic regions with mapped YBX3 K562 eCLIP data. 1: read profile (coverage), 2: read alignments, 3: crosslink positions profile, 4: input control profile, 5: gene annotations (thick blue regions are exons, thin blue regions introns), CLIPper / CLIPper IDR: CLIPper replicate 1 and IDR peaks, PEAKachu: PEAKachu peaks, PureCLIP: PureCLIP peaks (nearby crosslink positions merged). For clarity only gene strand reads from replicate 1 are displayed. **a** *PRDX6* whole gene region (length 11 kb, maximum read coverage 1141). **b** *DDOST* gene exons 6 and 7 region (length 563 bp, maximum read coverage 167)

even calls a single peak over the entire intron. In general, PEAKachu and CLIPper define peak regions by fitting functions (Gaussian density versus splines) on the mapped reads. More precisely, CLIPper fits splines on the genomic read coverage profile counting each base of a read once, while PEAKachu replaces each read with a Gaussian, using the genomic mean of read start and end as the center of the Gaussian. Both methods thus have problems with split reads, leading to PEAKachu calling peaks over introns in the presence of intron-spanning reads, and CLIPper calling peaks at exon ends with shared read coverage. Using more robust peaks (like CLIPper IDR) is the recommended way to obtain high-confidence binding sites, but it does not solve the underlying issue (see also Fig. 3). In contrast, PureCLIP uses read starts to identify crosslink sites, which later can be merged into peak regions. This circumvents the described problems, as each read is considered only once at one genomic position. For example, Fig. 1b shows a peak called by CLIPper and CLIPper IDR at the start of exon 6 (downstream exon). But since there are no read starts (i.e., crosslink sites) present, PureCLIP does not call a peak here. On the other hand, it still can be fooled since intron-spanning reads are treated no different to contiguously aligned reads. For the YBX3 dataset and with default settings, PEAKachu tends to call broader peaks than CLIPper, while PureCLIP peaks are much shorter (see Table S2 for peak statistics).

Exon binding is substantial in public CLIP-seq data

To quantify the extent of exon and near exon border binding in eCLIP data, CLIPper peak regions from 223 eCLIP datasets were overlapped with exon regions featuring strong experimental evidence (see Methods section "Data preparation and exon overlap statistics"). As shown in Fig. 2a, 61 datasets (27.4%) feature ≥50% exonic sites, with 14 datasets (6.3%) even reaching ≥75% (see Table S1 for full statistics on each dataset). Table S1 also lists the ratios of sites near exon borders and pair sites, i.e., two sites located at adjacent exon borders. Looking closer at the 61 datasets, 63.3% of exonic sites lie within ≤50 nt to exon borders, and 20.7% form pairs (<10 nt distance of site ends to exon borders required for both sites of the pair). We thus have a substantial amount of sites susceptible to split peak calling and false sequence context choice. Since the selection procedure for splice isoforms (i.e., their exon regions) was quite strict, the actual percentages should be even higher. As the data features experiments from K562 and HepG2 cell lines, we also looked at the correlation of percentages for RBPs with experiments in both lines. Figure 2b shows the correlation plot of exon site ratios, resulting in an R^2 score of 0.76. This suggests a general agreement in the amount of exon binding across cell lines. On the other hand, it also shows that classifying RBPs into spliced or unspliced binding oversimplifies actual binding patterns. Instead, the correct site context needs to be determined directly from the mapped data. One might wonder whether potentially problematic pair sites could easily be filtered out based on their assigned scores (i.e., log_2 fold changes) prior to data analysis. As shown in Fig. 2c, this is not the case, with an average score of 2.47 for pair sites and 2.17 for all exonic sites.

Sequence context influences binding site prediction performances

Based on the considerable amount of sites near exon borders, we further investigated whether different sequence contexts could also influence the performances of binding site prediction tools. For this we constructed different context datasets for 6 RBPs, by focusing on RBPs with high amounts of exonic sites (≥80%) and choosing 5 RBPs randomly within this range (see "Methods" sections). Briefly, we kept only sites ≤10 nt from exon borders and extended the centered sites 80 nt up- and downstream with both genomic and transcript context (total length 161 nt, see Table S3 for dataset details). Note that this also includes sites at transcript ends, where full extension is only possible in the genomic context case. To assess any effects, three different prediction tools (DeepBind [11], GraphProt [12], and GraphProt2 [13]) were run on both context sets, using 10-fold cross validation and no additional features (i.e., only sequence information). Figure 2d shows the performance results as average accuracies over the 6 datasets, for both genomic and transcript context sets (see Table S4 for detailed results). As we can see, using the more authentic transcript context considerably improves accuracies for all three tools, showcasing that context sequence choice can have a large influence on predictive performance and thus on what is learned. One could argue that including large amounts of context sequence bears the risk of learning binding site-unspecific patterns. We acknowledge that this can influence predictions. Some bias from the negative set is also possible, although we tried to minimize this by random sampling from the whole gene sequence and no overlap with positive sites. On the other hand, intronic context near exon borders also harbors various recognizable regions, like the polypyrimidine tract, or splice donor and acceptor sites, which can lead to wrong conclusions for spliced RNA binding RBPs. Moreover, learning the transcript context for RBPs binding to spliced RNA can also be advantageous, especially when predicting on gene sequences that contain introns.

To check whether the trained models learned any RBP specific binding information or rather generic context features, we generated GraphProt sequence logos for each RBP-context combination (see Figure S1). Sequence logos are generated from the top 200 scoring sites (taking the highest scoring 8-mer sequence for each site) of each positive training set, therefore providing a visualization

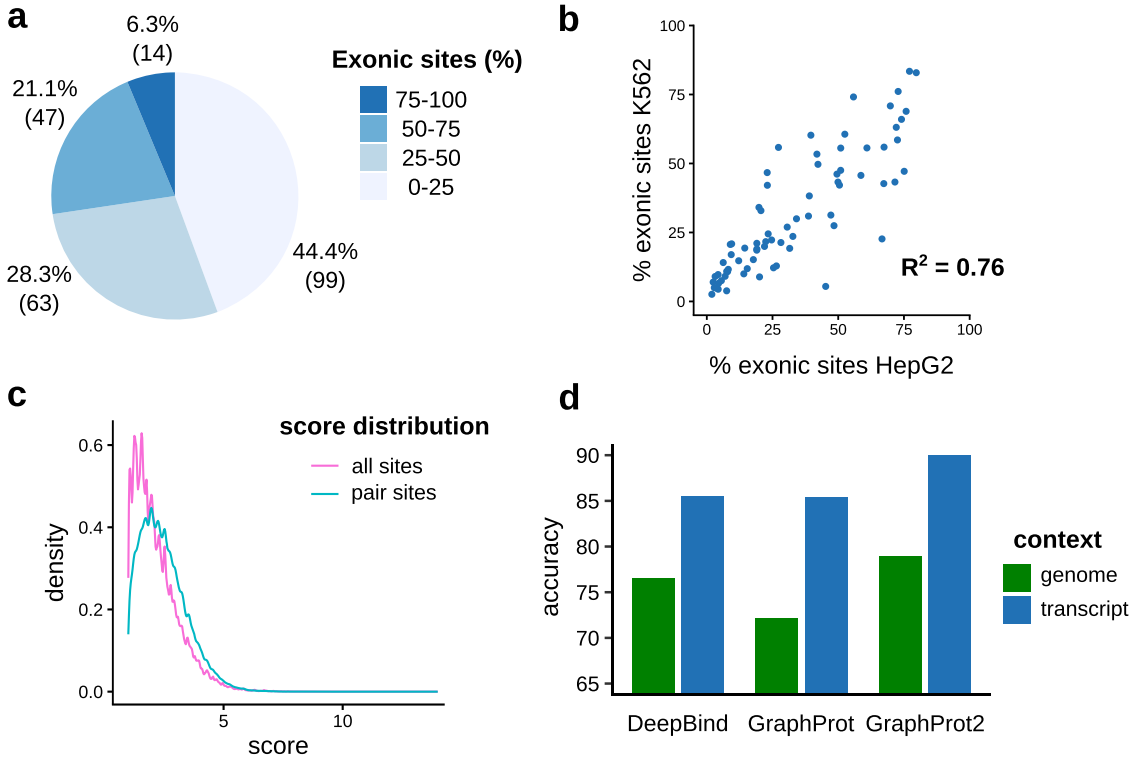


Fig. 2 Exon binding statistics of eCLIP datasets and prediction results for different sequence contexts. **a** Distribution of exonic site ratios for 223 eCLIP datasets over four percentage ranges. For each range, the percentage (number) of sets with ratios falling into this range is given. **b** Correlation plot of exonic site ratios for RBPs present in two cell lines (HepG2 and K562). **c** Site score distributions for all exonic sites and exonic sites that form pairs by being located at adjacent exon borders. log_2 fold change values of the sites determined by CLIPper were taken as site scores. Only pair sites with a distance of <10 nt to exon borders were considered. **d** Average classification accuracies over 6 eCLIP datasets for 3 RBP binding site prediction methods, comparing genome and transcript context

aid of what sequence information the model regards as most important. Comparing the generated sequence logos with known RBP binding preferences obtained from the ATtRACT database [14], we can see a general agreement (more or less pronounced depending on the RBP). For example, the Pumilio Response Element (PRE) of PUM2 (UGUANAUA) clearly shows up for both context sets, as well as the preference for CA-rich elements for IGF2BP1 and YBX3 or GA-rich elements for SRSF1. FMR1 and FXR2 are less distinguishable, although both RBPs are closely related and thus also might have common targets. This indicates that the models do not primarily pick up generic context information, but instead are capable of prioritizing RBP specific binding sites, independent of the context. Nevertheless, since we included a large amount of context (sequence lengths 161 nt), the context is expected to contribute to the increased performances for the transcript context sets. As discussed, this can be, depending on the prediction task, beneficial, as it can offer new insights into what other elements tend to be associated with core binding elements. In addition, choosing a more authentic context could also help to improve RNA secondary structure predictions, which often include hundreds of nucleotides of context.

Known motifs are enriched in transcript context

To check whether known binding motifs are more frequent in eCLIP sites with added transcript context compared to the respective sites with genomic context, we collected 28 motifs from 9 RBPs known to bind predominantly to spliced or exonic RNA (FMR1, FXR1, FXR2, IGF2BP1-3, PUM2, SRSF1, and YBX3) [15–20]. Since we could not find reported human motifs for YBX3, we used the corresponding mouse motif [21], as well as two human motifs from YBX1 and YBX2. We then took the CLIPper IDR peak regions (high-confidence reproducible peaks between replicates) of the respective eCLIP datasets, and used CLIPcontext to select sites near exon borders and to look for motifs in both genomic and transcript context sites. As shown in Table S5, there are 23 motifs that

have >10 hits in both genomic and transcript context sites (counting hits at same genomic or transcript positions only once). Out of these 23, 20 are 10 - 57% more frequent in transcript context sites, while the remaining three change by 2.8%, -2.3%, and -2.4%. The other five motifs with less than 10 hits are all enriched by 35% up to 709% (ratios calculated with counts normalized by total context set length).

On the one hand, when taking the transcript context, we expect higher frequencies for motifs that are commonly found in exonic regions. On the other hand, well-defined motifs like the PUM2 PRE (107 vs. 89 hits, 27.5%) or the extended compound motif for IGF2BP3 (7 vs. 1 hit, 709%) also show increased frequencies, indicating that more authentic binding sites are recovered by using the transcript context. To illustrate this (Fig. 3), we chose two example regions that contain IDR peaks as well as known binding motifs mapped by CLIPcontext of IGF2BP3 (the mentioned recently published compound motif GGC-N_{15-25}-CA-N_{7-20}-CA-N_{15-25}-GGC-N_{2-8}-$[CA]_4$) and PUM2 (the mentioned PRE UGUANAUA). As shown, the motifs are even split in these examples by the exon border, and the read profile accordingly suggests one split peak, although multiple CLIPper IDR peaks are reported, either in one of the two exons (IGF2BP3), or one at each adjacent exon end (PUM2). Naturally, we would expect the influence of context choice on recovering complete binding sites to be higher for multi-domain RBPs like IGF2BP1-3, which prefer to bind to several disconnected elements with long stretches of variable length in between. Since most RBPs in fact contain multiple RNA-binding domains and systematic studies on their combinatorial RNA recognition are still scarce [20], identifying the correct context in CLIP-seq studies could further help to uncover their combinatorial binding modes.

Strategies to improve CLIP-seq data analysis workflows

In this study we used CLIPcontext to extract the transcript context of genomic sites from a set of well annotated splice isoforms, completely ignoring the context information given in the eCLIP data. This is of course far from optimal, and future workflows should implement a more sophisticated, data-driven way to incorporate transcript information, in order to identify the most likely context and therefore increase the accuracy of peak calling and downstream processes. In this regard, one major factor will be the ability to correctly identify exon regions and their corresponding isoforms in a given sample, or at least the correct site neighborhood for accurate context extraction. The presence and quantity of split reads at exon borders therefore marks an important feature to decide which context is appropriate. Unfortunately, reference annotations often lag behind and do not cover

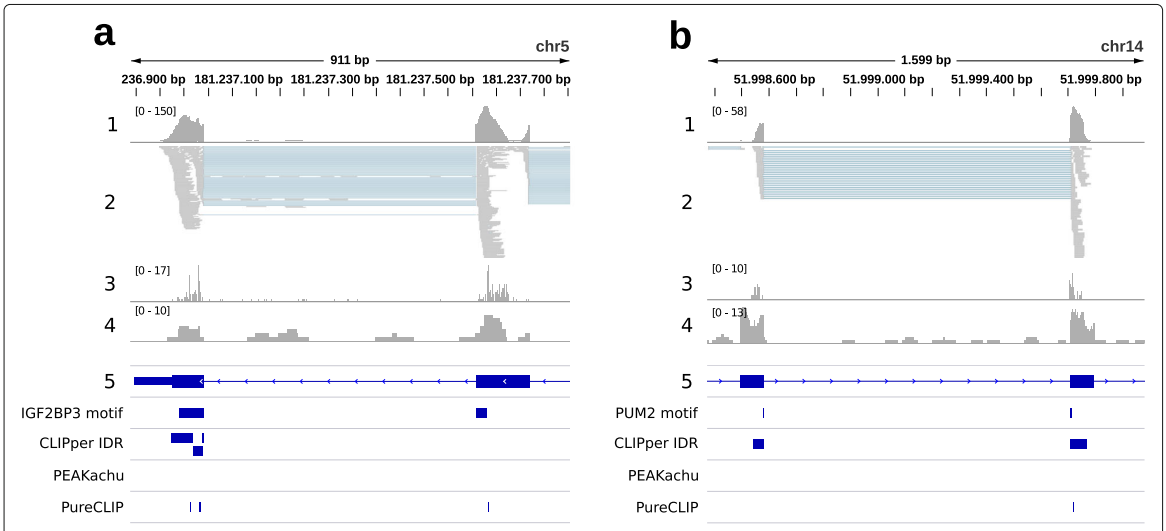


Fig. 3 IGV snapshot of two genomic regions with mapped IGF2BP3 K562 and PUM2 K562 eCLIP data. 1: read profile (coverage), 2: read alignments, 3: crosslink positions profile, 4: input control profile, 5: gene annotations (thick blue regions are exons, thin blue regions introns), IGF2BP3 / PUM2 motif: RBP motifs mapped with CLIPcontext, CLIPper IDR: CLIPper IDR peaks, PEAKachu: PEAKachu peaks, PureCLIP: PureCLIP peaks (nearby crosslink positions merged). For clarity only gene strand reads from replicate 1 are displayed. **a** *RACK1* gene exons 7 and 8 region (length 911 bp, maximum read coverage 150) with split IGF2BP3 motif hit. **b** *RTRAF* gene exons 4 and 5 region (length 1.599 bp, maximum read coverage 58) with split PUM2 motif hit

the present transcript diversity [22], which is why de novo transcriptome assemblies from RNA-seq data, e.g. by tools like Ryūtō [23], might be an interesting alternative to isoform detection or mapping approaches that rely on reference annotations. Since all these tools were developed for RNA-seq data, it will also be interesting to see whether it is possible to adapt them to work directly with CLIP-seq data, omitting the need to conduct additional RNA-seq experiments.

In any case, context selection should ideally be done on site level, as RBPs often have several biological roles and can bind to both contexts, depending on subcellular location and the time point in the RNA life cycle. In this regard, applying CLIP-seq to different subcellular fractions might be a way to further dissect binding events, as already done for some multi-function SR proteins [24]. In the presence of several likely contexts (i.e., for alternative splice isoforms), it is possible to keep all events if the goal is to learn general binding characteristics. This is because binding site prediction tools are typically robust when it comes to noisy data, as long as the principal binding preferences are still present in sufficient quantities. However, if the focus lies on specifically studying these events, it would be most convenient to label and output them separately.

An alternative approach could be to adapt or fine-tune peak calling based on specific features of the dataset at hand. These features could be learned from publically available CLIP-seq datasets, ideally produced with the same protocol (including read mapping), and possibly also the same cell type or condition. For example, dataset properties could be extracted and used as features, like exon-intron read distributions for typical exon-, intron-, or mixed context binding RBPs, either at defined genomic locations or over the whole genome. Additional labeled test data (either derived from CLIP-seq data or artificially constructed) could then be used to evaluate what features or strategies work best.

Conclusions

In this paper we raised the issue of ignoring transcript information in the process of peak calling and beyond. We showed that current peak callers by design are prone to false peak calling near exon borders, and that peak regions near exon borders are frequent in publicly available datasets. We also saw that sequence context choice has a profound effect on predicting sites near exon borders. Moreover, motif analysis confirmed that choosing the transcript context enriches for known RBP binding motifs, leading to the recovery of more authentic binding sites. Finally, we discussed ways on how to improve CLIP-seq analysis workflows in order to identify the correct site context.

Taken together, incorporating transcript information leads to more authentic results and thus should become an integral feature of future peak calling and downstream analysis methods.

Methods

Data preparation and exon overlap statistics

eCLIP datasets out of two cell lines (HepG2, K562) were downloaded from the ENCODE project website [25] (https://www.encodeproject.org, November 2018 release). Altogether the data covers 150 RBPs, divided into 103 HepG2 and 120 K562 sets, resulting in 223 datasets. We directly used the genomic binding regions (genome assembly GRCh38) determined by CLIPper, available in BED format for each replicate (2 replicates per dataset). For each RBP cell type combination, replicate binding sites were merged by keeping only the sites with the highest log_2 fold change (LFC) in case of overlapping sites. After filtering sites by LFC $\geq$1, sites were overlapped with exon regions of the most prominent transcripts using intersectBed (bedtools 2.29.0 [26]) and a required exon overlap $\geq$90% for a region to be counted as exon overlapping. We defined the most prominent isoform of a gene based on the information Ensembl (Ensembl Genes 97, GRCh38.p12) provides for each transcript through hierarchical filtering: APPRIS annotation [27] (highest priority, labels principal1-5), and transcript support level (TSL, labels 1-5). We considered only genes with isoforms featuring these labels and transcripts that belong to the GENCODE basic gene set, resulting in 29,798 isoforms and 238,271 exon regions. Exon overlap statistics for the 223 datasets are stored in Table S1.

Peak caller setup

To illustrate potential peak caller problems (Fig. 1), we chose an RBP cell type combination with a high amount of exonic peak regions (YBX3 K562, 84.6%), out of which 51.0% are close to exon borders (region ends $\leq$50 nt from exon borders, see Table S1 for statistics). To illustrate false peak calling at sites containing known motifs (Fig. 3), we chose the IGF2BP3 (HepG2) and PUM2 (K562) eCLIP sets. Mapped eCLIP reads in BAM format (replicate 1, size-matched input) and CLIPper peak regions (BED) for the three sets (ENCODE IDs ENCSR529FKI, ENCSR993OLA, ENCSR661ICQ) were obtained from the ENCODE website.

We collected peak regions identified by three peak callers: CLIPper, PEAKachu, and PureCLIP. For CLIPper, we took the peak regions called on replicate 1, filtered by a minimum LFC of 1. In addition, we also display the CLIPper IDR peaks (high-confidence peaks reproducible between replicates, Figs. 1 and 3). For PEAKachu and PureCLIP, we took the mapped reads (replicate 1, size-matched input), and used the R2 reads (second pair reads) as experiment and control libraries. PEAKachu was run

on Galaxy [28] (https://usegalaxy.eu, Galaxy tool version 0.1.0.2) with default settings and a fold threshold of 2. PureCLIP (version 1.3.1) was installed locally and run with default parameters, setting *−dm* 8 for merging called crosslink sites into peak regions.

Construction of sequence context sets

For comparing the effects of different sequence contexts on predictive performance, we chose 6 eCLIP sets from RBPs with documented binding preferences (IGF2BP1, FMR1, FXR2, PUM2, SRSF1, YBX3), which also feature relatively high percentages of exonic peak regions (from 40.23 to 84.06%, see Table S1). CLIPper replicate 1 peaks were obtained and filtered (maximum length of 80, minimum LFC of 3, maximum p-value of 0.01). We further selected all exonic sites within $\leq$10 nt of exon borders (`clipcontext exb`), and extracted their transcript and genomic context (`clipcontext g2t`), merging nearby sites (distance $\geq$10 nt) by selecting the site with the highest LFC, and extending sites to 161 nt length. To generate one negative set for both genome and transcript context sets, we used GraphProt2 (https://github.com/BackofenLab/GraphProt2) to randomly select genomic sites based on two criteria: 1) their location on genes covered by eCLIP peak regions and 2) no overlap with any eCLIP peak regions from the experiment. Sequence context set statistics are stored in Table S3.

Tool setup for context predictions

Three RBP binding site prediction tools (DeepBind, GraphProt, and GraphProt2) were trained on the described context sets (see previous Methods section). DeepBind models were trained using the DeepRAM [29] framework, which includes hyperparameter optimization. GraphProt and GraphProt2 models were trained using default parameters (no hyperparameter optimization). All three methods used only sequence features for classification. The accuracy measure, i.e., the proportion of correctly classified instances, was used in combination with 10-fold cross validation to measure model performances over 6 datasets. Accuracies are reported in Table S4, together with standard deviations from cross validation (except for GraphProt, since it does not output single accuracies during cross validation). GraphProt sequence logos for the top 100 scoring sites of each dataset-context combination are shown in Table S5, together with a description of known binding preferences.

Motif search

For the motif search, CLIPper IDR peaks for 9 RBPs were downloaded from ENCODE and filtered by a maximum length of 80. Sites near exon borders were selected and their transcript and genomic context was extracted as described in section "Construction of sequence context sets". CLIPcontext (`clipcontext mtf`) was then used to obtain motif frequencies in the transcript and genomic context sets, as well as to map the PUM2 and IGF2BP3 motifs to the genome, to generate the split motif annotations seen in Fig. 3.

CLIPcontext availability and documentation

CLIPcontext is available together with a comprehensive documentation on GitHub (https://github.com/BackofenLab/CLIPcontext), as well as on Bioconda (https://anaconda.org/bioconda/clipcontext). Besides mapping sites of interest in BED format (transcript or genomic coordinates) to a user-definable transcriptome or the genome, CLIPcontext also offers modes for the extraction of: sites near exon borders, a list of most prominent transcripts, intronic sites, or exon and intron regions for a given set of transcripts. Moreover, a motif search can be conducted on genomic and transcript regions (including split motif discovery) for comparative analysis.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12864-020-07297-0.

Additional file 1: Table S1: Exon overlap statistics of ENCODE eCLIP datasets (.xlsx)

Additional file 2: Supplementary tables S2-S4 and supplementary figure S1 (.pdf)

Additional file 3: Table S5: Motif search results for 9 RBPs and 28 binding motifs collected from various sources (.xlsx)

Abbreviations

CLIP-Seq: Cross-linking and immunoprecipitation followed by next generation sequencing; eCLIP: Enhanced CLIP; iCLIP: Individual-nucleotide CLIP; IGV: Integrative genomics viewer; PAR-CLIP: Photoactivatable-ribonucleoside-enhanced CLIP; RBP: RNA-binding protein; nt: Nucleotides; LFC: Log2 fold change; PRE: Pumilio response element

Acknowledgements

We thank Martin Raden and Gabriel Pratt for their invaluable suggestions on the topic.

Authors' contributions

RB and MU conceived the study. VDT and MU performed the binding site predictions. MU performed the remaining data analysis, wrote the draft, and implemented the software. RB, VDT, and MU contributed to and approved the final manuscript.

Funding

MU was funded by Deutsche Forschungsgemeinschaft (DFG) grant BA 2168/11-1 SPP 1738 and BA2168/11-2 SPP 1738. VDT was funded by DFG grant BA 2168/3-3. The article processing charge was funded by the University of Freiburg in the funding programme Open Access Publishing. Open Access funding enabled and organized by Projekt DEAL.

Availability of data and materials

CLIPcontext is available on GitHub (https://github.com/BackofenLab/CLIPcontext) and Bioconda (https://anaconda.org/bioconda/clipcontext). Supplementary data is also stored in the GitHub repository.

Ethics approval and consent to participate

Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 22 July 2020 Accepted: 2 December 2020
Published online: 17 December 2020

References

1. Licatalosi DD, Mele A, Fak JJ, Ule J, Kayikci M, Chi SW, Clark TA, Schweitzer AC, Blume JE, Wang X, et al. HITS-CLIP yields genome-wide insights into brain alternative RNA processing. Nature. 2008;456(7221):464.
2. Hafner M, Landthaler M, Burger L, Khorshid M, Hausser J, Berninger P, Rothballer A, Ascano Jr M, Jungkamp A-C, Munschauer M, et al. Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP. Cell. 2010;141(1):129–41.
3. König J, Zarnack K, Rot G, Curk T, Kayikci M, Zupan B, Turner DJ, Luscombe NM, Ule J. iCLIP reveals the function of hnRNP particles in splicing at individual nucleotide resolution. Nat Struct Mol Biol. 2010;17(7):909.
4. Van Nostrand EL, Pratt GA, Shishkin AA, Gelboin-Burkhart C, Fang MY, Sundararaman B, Blue SM, Nguyen TB, Surka C, Elkins K, et al. Robust transcriptome-wide discovery of RNA-binding protein binding sites with enhanced CLIP (eCLIP). Nat Methods. 2016;13(6):508.
5. Uren PJ, Bahrami-Samani E, Burns SC, Qiao M, Karginov FV, Hodges E, Hannon GJ, Sanford JR, Penalva LO, Smith AD. Site identification in high-throughput RNA–protein interaction data. Bioinformatics. 2012;28(23):3013–20.
6. Lovci MT, Ghanem D, Marr H, Arnold J, Gee S, Parra M, Liang TY, Stark TJ, Gehman LT, Hoon S, et al. Rbfox proteins regulate alternative mRNA splicing through evolutionarily conserved RNA bridges. Nat Struct Mol Biol. 2013;20:1434.
7. Bischler T, Maticzka D, Förstner KU, Wright PR. PEAKachu. https://github.com/tbischler/PEAKachu.
8. Krakau S, Richard H, Marsico A. PureCLIP: capturing target-specific protein–RNA interaction footprints from single-nucleotide CLIP-seq data. Genome Biol. 2017;18(1):240.
9. Uhl M, Houwaart T, Corrado G, Wright PR, Backofen R. Computational analysis of CLIP-seq data. Methods. 2017;118:60–72.
10. Thorvaldsdóttir H, Robinson JT, Mesirov JP. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. Brief Bioinforma. 2013;14(2):178–92.
11. Alipanahi B, Delong A, Weirauch MT, Frey BJ. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. Nat Biotechnol. 2015;33(8):831.
12. Maticzka D, Lange SJ, Costa F, Backofen R. GraphProt: modeling binding preferences of RNA-binding proteins. Genome Biol. 2014;15(1):17.
13. Uhl M, Tran VD, Heyl F, Backofen R. GraphProt2. https://github.com/BackofenLab/GraphProt2.
14. Giudice G, Sánchez-Cabo F, Torroja C, Lara-Pezzi E. ATtRACT - a database of RNA-binding proteins and associated motifs. Database. 2016;2016:. https://doi.org/10.1093/database/baw035.
15. Galgano A, Forrer M, Jaskiewicz L, Kanitz A, Zavolan M, Gerber AP. Comparative analysis of mRNA targets for human PUF-family proteins suggests extensive interaction with the miRNA regulatory system. PLoS ONE. 2008;3(9):3164.
16. Patel VL, Mitra S, Harris R, Buxbaum AR, Lionnet T, Brenowitz M, Girvin M, Levy M, Almo SC, Singer RH, et al. Spatial arrangement of an RNA zipcode identifies mRNAs under post-transcriptional control. Genes Dev. 2012;26(1):43–53.
17. Ascano M, Mukherjee N, Bandaru P, Miller JB, Nusbaum JD, Corcoran DL, Langlois C, Munschauer M, Dewell S, Hafner M, et al. FMRP targets distinct mRNA sequence elements to regulate protein expression. Nature. 2012;492(7429):382–6.
18. Ray D, Kazan H, Cook KB, Weirauch MT, Najafabadi HS, Li X, Gueroussov S, Albu M, Zheng H, Yang A, et al. A compendium of RNA-binding motifs for decoding gene regulation. Nature. 2013;499(7457):172–7.
19. Anczukow O, Akerman M, Clery A, Wu J, Shen C, Shirole NH, Raimer A, Sun S, Jensen MA, Hua Y, et al. SRSF1-regulated alternative splicing in breast cancer. Mol Cell. 2015;60(1):105–17.
20. Schneider T, Hung L-H, Aziz M, Wilmen A, Thaum S, Wagner J, Janowski R, Müller S, Schreiner S, Friedhoff P, et al. Combinatorial recognition of clustered RNA elements by the multidomain RNA-binding protein IMP3. Nat Commun. 2019;10(1):1–18.
21. Giorgini F, Davies HG, Braun RE. MSY2 and MSY4 bind a conserved sequence in the 3′ untranslated region of protamine 1 mRNA in vitro and in vivo. Mol Cell Biol. 2001;21(20):7010–9.
22. Morillon A, Gautheret D. Bridging the gap between reference and real transcriptomes. Genome Biol. 2019;20(1):1–7.
23. Gatter T, Stadler PF. Ryūtō: network-flow based transcriptome reconstruction. BMC Bioinformatics. 2019;20(1):190.
24. Brugiolo M, Botti V, Liu N, Müller-McNicoll M, Neugebauer KM. Fractionation iCLIP detects persistent SR protein binding to conserved, retained introns in chromatin, nucleoplasm and cytoplasm. Nucleic Acids Res. 2017;45(18):10452–65.
25. Sloan CA, Chan ET, Davidson JM, Malladi VS, Strattan JS, Hitz BC, Gabdank I, Narayanan AK, Ho M, Lee BT, et al. ENCODE data at the ENCODE portal. Nucleic Acids Res. 2015;44(D1):726–32.
26. Quinlan AR, Hall IM. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics. 2010;26(6):841–2.
27. Rodriguez JM, Maietta P, Ezkurdia I, Pietrelli A, Wesselink J-J, Lopez G, Valencia A, Tress ML. APPRIS: annotation of principal and alternative splice isoforms. Nucleic Acids Res. 2012;41(D1):110–7.
28. Afgan E, Baker D, Batut B, Van Den Beek M, Bouvier D, Čech M, Chilton J, Clements D, Coraor N, Grüning BA, et al. The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res. 2018;46(W1):537–44.
29. Trabelsi A, Chaabane M, Ben-Hur A. Comprehensive evaluation of deep learning architectures for prediction of DNA/RNA sequence binding specificities. Bioinformatics. 2019;35(14):269–77.

Publisher's Note

[P5] Peakhood: individual site context extraction for CLIP-seq peak regions

Publication:

- [P5] **Michael Uhl**, Dominik Rabsch, Florian Eggenhofer, and Rolf Backofen. **Peakhood: individual site context extraction for CLIP-seq peak regions**. *Bioinformatics*, 2021.

Contributions of individual authors:

"I am the main contributor to this work. I conceived the tool, implemented it, wrote the online manual and the manuscript draft. Dominik Rabsch generated the transcript context site collections and added multi-threading support to speed up batch processing. Florian Eggenhofer and Rolf Backofen both revised the draft and provided helpful comments. All authors reviewed and approved the final manuscript."

Michael Uhl

The following co-authors confirm the above-stated contributions:

Bioinformatics, 2021, 1–2
https://doi.org/10.1093/bioinformatics/btab755
Advance Access Publication Date: 4 November 2021
Applications Note

Sequence analysis

Peakhood: individual site context extraction for CLIP-seq peak regions

Michael Uhl [1,*], Dominik Rabsch [1], Florian Eggenhofer [1] and Rolf Backofen [1,2,*]

[1]Bioinformatics Group, Department of Computer Science, University of Freiburg, Freiburg im Breisgau, Germany and [2]Signalling Research Centres BIOSS and CIBSS, University of Freiburg, Freiburg im Breisgau, Germany

*To whom correspondence should be addressed.

Associate Editor: Can Alkan

Received on August 9, 2021; revised on October 13, 2021; editorial decision on October 27, 2021; accepted on October 29, 2021

Abstract

Motivation: CLIP-seq is by far the most widely used method to determine transcriptome-wide binding sites of RNA-binding proteins (RBPs). The binding site locations are identified from CLIP-seq read data by tools termed peak callers. Many RBPs bind to a spliced RNA (i.e. transcript) context, but all currently available peak callers only consider and report the genomic context. To accurately model protein binding behavior, a tool is needed for the individual context assignment to CLIP-seq peak regions.

Results: Here we present Peakhood, the first tool that utilizes CLIP-seq peak regions identified by peak callers, in tandem with CLIP-seq read information and genomic annotations, to determine which context applies, individually for each peak region. For sites assigned to transcript context, it further determines the most likely splice variant, and merges results for any number of datasets to obtain a comprehensive collection of transcript context binding sites.

Availability and implementation: Peakhood is freely available under MIT license at: https://github.com/BackofenLab/Peakhood.

Contact: uhlm@informatik.uni-freiburg.de or backofen@informatik.uni-freiburg.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

CLIP-seq (cross-linking and immunoprecipitation followed by next generation sequencing) (Licatalosi *et al.*, 2008) is the most widely used procedure to experimentally determine the exact transcriptome-wide binding locations of RNA-binding proteins (RBPs). The most popular protocol variants are PAR-CLIP (Hafner *et al.*, 2010), iCLIP (König *et al.*, 2010) and eCLIP (Van Nostrand *et al.*, 2016). CLIP-seq is usually performed *in vivo* for a specific RBP, resulting in a library of reads bound by the target RBP. Binding sites are subsequently identified by mapping the reads back to the reference genome, and analyzing the read profiles with tools referred to as peak callers. A number of peak callers have been popular over the years, such as Piranha (Uren *et al.*, 2012), CLIPper (Lovci *et al.*, 2013) or PureCLIP (Krakau *et al.*, 2017).

Calling peaks in the genomic context, as done by all currently available peak callers, is unbiased for RBPs that predominantly bind to unspliced RNA. However, for RBPs that predominantly bind in a spliced (i.e. transcript) context, this is clearly suboptimal. Indeed, a recent study (Uhl *et al.*, 2020) has demonstrated this to be a substantial problem, and that the inclusion of transcript context can improve the identification of authentic binding sites. Peak callers applied to CLIP-seq data have produced millions of publicly available binding sites, e.g. from ENCODE (Van Nostrand *et al.*, 2020b). Consequently, a tool is required that can analyze CLIP-seq peak regions to extract the individual site context for each peak region.

Here, we present Peakhood, the first tool capable of extracting the most likely site context, individually for each CLIP-seq peak region. The necessary information are extracted directly from the CLIP-seq read profiles, in combination with a genomic annotations file (both reference and custom annotations are supported). For sites assigned to transcript context, Peakhood further determines the most likely splice variant. In addition, Peakhood can merge extracted transcript context sets into comprehensive transcript context site collections. Peakhood also supports batch processing, i.e. context extraction of multiple datasets and merging in one run. As a supplement, we provide four precomputed transcript context site collections, using eCLIP datasets of 49 RBPs with known roles in posttranscriptional gene regulation (see Data availability section).

2 Approach

Here, we briefly describe how Peakhood works. A detailed description can be found in the Online Supplementary (Section 1.2). For full details, please check out the comprehensive manual on GitHub. Peakhood first extracts the site context for each input peak region. Figure 1a shows two peak regions inside a typical transcript context. Peakhood uses the given exon annotations (GTF) and CLIP-seq

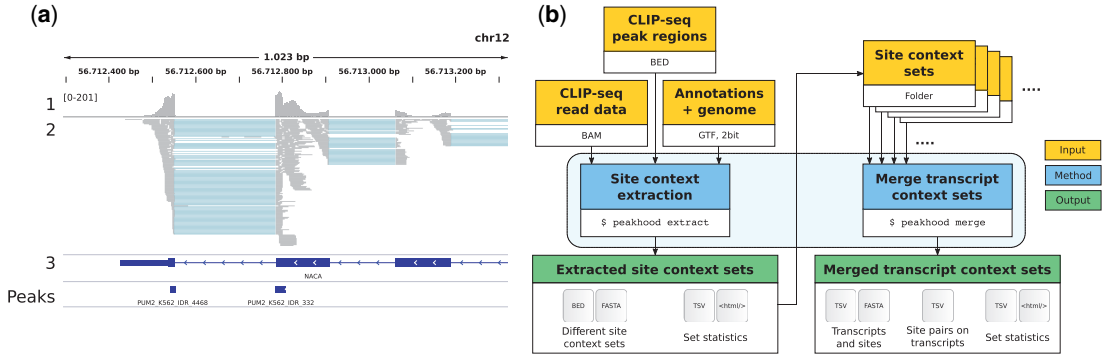


Fig. 1. (**a**) Genomic region (IGV screenshot) with mapped PUM2 K562 eCLIP data (details in Supplementary Section 1.4). 1: Read profile (coverage range in brackets), 2: read alignments, 3: gene annotations (thick blue regions are exons, thin blue regions introns), Peaks: peaks called by CLIPper IDR method (high-confidence peaks reproducible between replicates). Example transcript context region for the predominantly spliced RNA-binding RBP PUM2, where an exon border site is falsely split in two peaks. (**b**) Overview of the Peakhood workflow for the two main program modes extract and merge. Yellow boxes mark necessary inputs, blue boxes the two program modes and green boxes the outputs. Arrows show the dependencies between inputs, modes and outputs

read information (BAM), essentially looking for differences in exon and surrounding intron coverage, as well as coverage drops at exon borders. If these differences exceed the configured thresholds, the site is assigned to transcript context, otherwise to genomic context (Supplementary Fig. S1 example). In addition, sites at exon borders connected by intron-spanning reads are merged into single sites (as in Fig. 1a). For sites assigned to transcript context, Peakhood further selects the most likely site-transcript combination, using various read, site and transcript statistics. Moreover, Peakhood can merge single datasets into comprehensive transcript context site collections (see Fig. 1b for the extraction and merge workflow). The collections also include tabular data, e.g. to identify which sites on transcripts are in close distance, or if site distances decreased compared to the original genomic context. Percentages of extracted transcript context sites agree with known RBP roles (see Supplementary Section 1.3 and Fig. S2). Peakhood requires a Linux operating system and is easy to install, e.g. via Conda (Conda package available). The tool was tested (Intel i7-8700k, Ubuntu 18.04 LTS), with single dataset site context extraction (example dataset with 2146 input peak regions, see Supplementary Section 1.6) taking about 2 min and 30 s.

3 Conclusion

Here we presented Peakhood, the first tool capable of extracting the most likely site context, individually for each CLIP-seq peak region. Peakhood is easy to install and use, thanks to its comprehensive online manual, and it works with standardized file formats (BAM, BED, GTF, 2 bit). We demonstrated Peakhood's capabilities with eCLIP data and peak regions obtained from ENCODE (Van Nostrand *et al.*, 2020b). However, it is not limited to this type of data, and should work fine with other HTS peak data (iCLIP, PAR-CLIP, OOPS), as well as other peak caller outputs, e.g. from PureCLIP. The flexibility is further increased through Peakhood's various command line parameters, to adapt it for individual datasets or new input types. Summing up, Peakhood allows for an improved modeling of protein binding behavior, by providing a more authentic sequence and structure context, especially for spliced RNA-binding proteins.

Acknowledgement

The authors thank the reviewers for their constructive comments.

Funding

M.U. was funded by Deutsche Forschungsgemeinschaft (DFG) [BA 2168/11-1 SPP 1738, BA2168/11-2 SPP 1738]. D.R. was funded by the Bundesministerium für Bildung und Forschung (BMBF) [RNAProNet-031L0164B]. The study was further supported by the DFG under Germany's Excellence Strategy (CIBSS—EXC-2189-Project ID 390939984).

Conflict of Interest: none declared.

Data availability

The transcript context site collections generated by Peakhood from eCLIP datasets of 49 RBPs (first collection with 36 RBPs from HepG2, second collection with 40 RBPs from K562) with known roles in posttranscriptional gene regulation (mRNA stability and decay, translational regulation; information taken from Van Nostrand *et al.*, 2020a, Supplementary Data 1 Table) can be downloaded from Zenodo (https://doi.org/10.5281/zenodo.5557101).

References

Hafner,M. *et al.* (2010) Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP. *Cell*, **141**, 129–141.
König,J. *et al.* (2010) iCLIP reveals the function of hnRNP particles in splicing at individual nucleotide resolution. *Nat. Struct. Mol. Biol.*, **17**, 909–915.
Krakau,S. *et al.* (2017) PureCLIP: capturing target-specific protein–RNA interaction footprints from single-nucleotide CLIP-seq data. *Genome Biol.*, **18**, 240.
Licatalosi,D.D. *et al.* (2008) HITS-CLIP yields genome-wide insights into brain alternative RNA processing. *Nature*, **456**, 464–469.
Lovci,M.T. *et al.* (2013) Rbfox proteins regulate alternative mRNA splicing through evolutionarily conserved RNA bridges. *Nat. Struct. Mol. Biol.*, **20**, 1434–1442.
Uhl,M. *et al.* (2020) Improving CLIP-seq data analysis by incorporating transcript information. *BMC Genomics*, **21**, 1–8.
Uren,P.J. *et al.* (2012) Site identification in high-throughput RNA–protein interaction data. *Bioinformatics*, **28**, 3013–3020.
Van Nostrand,E.L. *et al.* (2016) Robust transcriptome-wide discovery of RNA-binding protein binding sites with enhanced CLIP (eCLIP). *Nat. Methods*, **13**, 508–514.
Van Nostrand,E.L. *et al.* (2020a) A large-scale binding and functional map of human RNA-binding proteins. *Nature*, **583**, 711–719.
Van Nostrand,E.L. *et al.* (2020b) Principles of RNA processing from analysis of enhanced CLIP maps for 150 RNA binding proteins. *Genome Biol.*, **21**, 26.

Bibliography

[1] Allison Piovesan, Francesca Antonaros, Lorenza Vitale, Pierluigi Strippoli, Maria Chiara Pelleri, and Maria Caracausi. Human protein-coding genes and gene feature statistics in 2019. *BMC research notes*, 12(1):1–5, 2019. (Cited on pages 1, 10 and 14.)

[2] Francis S Collins. Medical and societal consequences of the Human Genome Project. *New England Journal of Medicine*, 341(1):28–37, 1999. (Cited on page 1.)

[3] Eric S Lander, Lauren M Linton, Bruce Birren, Chad Nusbaum, Michael C Zody, Jennifer Baldwin, Keri Devon, Ken Dewar, Michael Doyle, William FitzHugh, et al. Initial sequencing and analysis of the human genome. 2001. (Cited on pages 1 and 10.)

[4] Elizabeth Pennisi. Shining a light on the genome's' dark matter', 2010. (Cited on page 1.)

[5] Joseph R Ecker, Wendy A Bickmore, Inês Barroso, Jonathan K Pritchard, Yoav Gilad, and Eran Segal. ENCODE explained. *Nature*, 489(7414):52–54, 2012. (Cited on page 1.)

[6] Kevin V Morris and John S Mattick. The rise of regulatory RNA. *Nature Reviews Genetics*, 15(6):423–437, 2014. (Cited on pages 2, 10, 12 and 17.)

[7] Giacomo Cavalli and Edith Heard. Advances in epigenetics link genetics to the environment and disease. *Nature*, 571(7766):489–499, 2019. (Cited on page 2.)

[8] Barbara Uszczynska-Ratajczak, Julien Lagarde, Adam Frankish, Roderic Guigó, and Rory Johnson. Towards a complete map of the human long non-coding RNA transcriptome. *Nature Reviews Genetics*, 19(9):535–548, 2018. (Cited on pages 2 and 10.)

[9] Matthias W Hentze, Alfredo Castello, Thomas Schwarzl, and Thomas Preiss. A brave new world of RNA-binding proteins. *Nature reviews Molecular cell biology*, 19(5):327–341, 2018. (Cited on pages 2 and 25.)

[10] Kotb Abdelmohsen, Amaresh C Panda, Min-Ju Kang, Rong Guo, Jiyoung Kim, Ioannis Grammatikakis, Je-Hyun Yoon, Dawood B Dudekula, Ji Heon Noh, Xiaoling Yang, et al. 7SL RNA represses p53 translation by competing with HuR. *Nucleic acids research*, 42(15):10099–10111, 2014. (Cited on pages 2 and 58.)

[11] Fátima Gebauer, Thomas Schwarzl, Juan Valcárcel, and Matthias W Hentze. RNA-binding proteins in human genetic disease. *Nature Reviews Genetics*, 22(3):185–198, 2021. (Cited on pages 2, 25 and 26.)

[12] Erin G Conlon and James L Manley. RNA-binding proteins in neurodegeneration: mechanisms in aggregate. *Genes & development*, 31(15):1509–1528, 2017. (Cited on page 2.)

[13] Bruno Pereira, Marc Billaud, and Raquel Almeida. RNA-binding proteins in cancer: old players and new actors. *Trends in cancer*, 3(7):506–528, 2017. (Cited on page 2.)

[14] Elaine R Mardis. A decade's perspective on DNA sequencing technology. *Nature*, 470(7333):198–203, 2011. (Cited on page 2.)

[15] Stephan C Schuster. Next-generation sequencing transforms today's biology. *Nature methods*, 5(1):16–18, 2008. (Cited on page 2.)

[16] Michael L Metzker. Sequencing technologies - the next generation. *Nature reviews genetics*, 11(1):31–46, 2010. (Cited on page 3.)

[17] Zhong Wang, Mark Gerstein, and Michael Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57–63, 2009. (Cited on pages 3 and 27.)

[18] Flora CY Lee and Jernej Ule. Advances in CLIP technologies for studies of protein-RNA interactions. *Molecular cell*, 69(3):354–369, 2018. (Cited on pages 3, 31 and 34.)

[19] Kenneth Katz, Oleg Shutov, Richard Lapoint, Michael Kimelman, J Rodney Brister, and Christopher O'Sullivan. The Sequence Read Archive: a decade more of explosive growth. *Nucleic acids research*, 2021. (Cited on page 3.)

[20] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. Deep learning in bioinformatics. *Briefings in bioinformatics*, 18(5):851–869, 2017. (Cited on page 3.)

[21] Emmanuelle J Javaux. Challenges in evidencing the earliest traces of life. *Nature*, 572(7770):451–460, 2019. (Cited on page 7.)

[22] Martina Preiner, Silke Asche, Sidney Becker, Holly C Betts, Adrien Boniface, Eloi Camprubi, Kuhan Chandru, Valentina Erastova, Sriram G Garg, Nozair Khawaja, et al. The future of origin of life research: bridging decades-old divisions. *Life*, 10(3):20, 2020. (Cited on page 7.)

[23] Rosalind E Franklin and Raymond G Gosling. Molecular configuration in sodium thymonucleate. *Nature*, 171(4356):740–741, 1953. (Cited on page 8.)

[24] James D Watson and Francis HC Crick. Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738, 1953. (Cited on page 8.)

[25] Sandy L Klemm, Zohar Shipony, and William J Greenleaf. Chromatin accessibility and the regulatory epigenome. *Nature Reviews Genetics*, 20(4):207–220, 2019. (Cited on pages 8 and 13.)

[26] ChemAxon. Marvin - a full featured chemical editor for making science accessible on all platforms. https://chemaxon.com/products/marvin. (Cited on pages 9 and 22.)

[27] Klas Hatje, Stefanie Mühlhausen, Dominic Simm, and Martin Kollmar. The Protein-Coding Human Genome: Annotating High-Hanging Fruits. *BioEssays*, 41(11):1900066, 2019. (Cited on page 9.)

[28] Sarah Djebali, Carrie A Davis, Angelika Merkel, Alex Dobin, Timo Lassmann, Ali Mortazavi, Andrea Tanzer, Julien Lagarde, Wei Lin, Felix Schlesinger, et al. Landscape of transcription in human cells. *Nature*, 489(7414):101–108, 2012. (Cited on page 10.)

[29] Luisa Statello, Chun-Jie Guo, Ling-Ling Chen, and Maite Huarte. Gene regulation by long non-coding RNAs and its biological functions. *Nature Reviews Molecular Cell Biology*, 22(2):96–118, 2021. (Cited on pages 10, 12 and 20.)

[30] Seo-Won Choi, Hyun-Woo Kim, and Jin-Wu Nam. The small peptide world in long non-coding RNAs. *Briefings in bioinformatics*, 20(5):1853–1864, 2019. (Cited on page 10.)

[31] Alexander F Palazzo and Eugene V Koonin. Functional long non-coding RNAs evolve from junk transcripts. *Cell*, 2020. (Cited on page 10.)

[32] Ryan J Taft, Michael Pheasant, and John S Mattick. The relationship between non-protein-coding DNA and eukaryotic complexity. *Bioessays*, 29(3):288–299, 2007. (Cited on pages 10 and 16.)

[33] Hyunmin Lee, Zhaolei Zhang, and Henry M Krause. Long noncoding RNAs and repetitive elements: junk or intimate evolutionary partners? *TRENDS in Genetics*, 35(12):892–902, 2019. (Cited on pages 10 and 16.)

[34] Francis HC Crick. On protein synthesis. In *Symp Soc Exp Biol*, volume 12, page 8, 1958. (Cited on page 10.)

[35] Francis Crick. Central dogma of molecular biology. *Nature*, 227(5258):561–563, 1970. (Cited on page 10.)

[36] John M Coffin. 50th anniversary of the discovery of reverse transcriptase. *Molecular Biology of the Cell*, 32(2):91–97, 2021. (Cited on page 11.)

[37] Liguo Zhang, Alexsia Richards, M Inmaculada Barrasa, Stephen H Hughes, Richard A Young, and Rudolf Jaenisch. Reverse-transcribed SARS-CoV-2 RNA can integrate into the genome of cultured human cells and can be expressed in patient-derived tissues. *Proceedings of the National Academy of Sciences*, 118(21), 2021. (Cited on page 11.)

[38] Nathan Smits, Jay Rasmussen, Gabriela O Bodea, Alberto A Amarilla, Patricia Gerdes, Francisco J Sanchez-Luque, Prabha Ajjikuttira, Naphak Modhiran, Benjamin

Liang, Jamila Faivre, et al. No evidence of human genome integration of SARS-CoV-2 found by long-read DNA sequencing. *Cell Reports*, page 109530, 2021. (Cited on page 11.)

[39] Anna D Senft and Todd S Macfarlan. Transposable elements shape the evolution of mammalian development. *Nature Reviews Genetics*, pages 1–21, 2021. (Cited on page 11.)

[40] Caitlin M Roake and Steven E Artandi. Regulation of human telomerase in homeostasis and disease. *Nature Reviews Molecular Cell Biology*, 21(7):384–397, 2020. (Cited on page 11.)

[41] Natalia Pinzón, Stéphanie Bertrand, Lucie Subirana, Isabelle Busseau, Hector Escrivá, and Hervé Seitz. Functional lability of RNA-dependent RNA polymerases in animals. *PLoS genetics*, 15(2):e1007915, 2019. (Cited on page 11.)

[42] Irene Bozzoni. Widespread occurrence of circular RNA in eukaryotes. *Nature Reviews Genetics*, pages 1–1, 2021. (Cited on page 12.)

[43] Noa Gil and Igor Ulitsky. Regulation of gene expression by cis-acting long non-coding RNAs. *Nature Reviews Genetics*, 21(2):102–117, 2020. (Cited on pages 12 and 20.)

[44] Robert G Roeder. 50+ years of eukaryotic transcription: an expanding universe of factors and mechanisms. *Nature structural & molecular biology*, 26(9):783–791, 2019. (Cited on page 12.)

[45] Allison C Schier and Dylan J Taatjes. Structure and mechanism of the RNA polymerase II transcription machinery. *Genes & development*, 34(7-8):465–488, 2020. (Cited on page 12.)

[46] Heena Khatter, Matthias K Vorlaender, and Christoph W Mueller. RNA polymerase I and III: similar yet unique. *Current opinion in structural biology*, 47:88–94, 2017. (Cited on page 12.)

[47] Vanja Haberle and Alexander Stark. Eukaryotic core promoters and the functional basis of transcription initiation. *Nature reviews Molecular cell biology*, 19(10):621–637, 2018. (Cited on page 13.)

[48] Fei Xavier Chen, Edwin R Smith, and Ali Shilatifard. Born to run: control of transcription elongation by RNA polymerase II. *Nature reviews Molecular cell biology*, 19(7):464–478, 2018. (Cited on page 13.)

[49] Joshua D Eaton and Steven West. Termination of transcription by RNA polymerase II: BOOM! *Trends in Genetics*, 36(9):664–675, 2020. (Cited on page 13.)

[50] Takashi Fukaya, Bomyi Lim, and Michael Levine. Enhancer control of transcriptional bursting. *Cell*, 166(2):358–368, 2016. (Cited on page 13.)

[51] Vittorio Sartorelli and Shannon M Lauberth. Enhancer RNAs are an important regulatory layer of the epigenome. *Nature structural & molecular biology*, 27(6):521–528, 2020. (Cited on page 13.)

[52] Jesse R Raab and Rohinton T Kamakaka. Insulators and promoters: closer than we think. *Nature Reviews Genetics*, 11(6):439–446, 2010. (Cited on page 13.)

[53] Moyra Lawrence, Sylvain Daujat, and Robert Schneider. Lateral thinking: how histone modifications regulate gene expression. *Trends in Genetics*, 32(1):42–56, 2016. (Cited on page 13.)

[54] Maxim VC Greenberg and Deborah Bourc'his. The diverse roles of DNA methylation in mammalian development and disease. *Nature reviews Molecular cell biology*, 20(10):590–607, 2019. (Cited on page 13.)

[55] Lydia Herzel, Diana SM Ottoz, Tara Alpert, and Karla M Neugebauer. Splicing and transcription touch base: co-transcriptional spliceosome assembly and function. *Nature reviews Molecular cell biology*, 18(10):637–650, 2017. (Cited on pages 14 and 15.)

[56] Bhupendra Verma, Maureen V Akinyi, Antto J Norppa, and Mikko J Frilander. Minor spliceosome and disease. In *Seminars in cell & developmental biology*, volume 79, pages 103–112. Elsevier, 2018. (Cited on page 14.)

[57] Manuel Irimia and Scott William Roy. Origin of spliceosomal introns and alternative splicing. *Cold Spring Harbor perspectives in biology*, 6(6):a016071, 2014. (Cited on page 14.)

[58] Alan M Lambowitz and Marlene Belfort. Mobile bacterial group II introns at the crux of eukaryotic evolution. *Microbiology spectrum*, 3(1):3–1, 2015. (Cited on page 14.)

[59] Marta Melé, Kaia Mattioli, William Mallard, David M Shechner, Chiara Gerhardinger, and John L Rinn. Chromatin environment, transcriptional regulation, and splicing distinguish lincRNAs and mRNAs. *Genome research*, 27(1):27–37, 2017. (Cited on page 14.)

[60] Claire M Smathers and Aaron R Robart. The mechanism of splicing as told by group II introns: Ancestors of the spliceosome. *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, 1862(11-12):194390, 2019. (Cited on page 15.)

[61] Max E Wilkinson, Clément Charenton, and Kiyoshi Nagai. RNA Splicing by the Spliceosome. *Annual review of biochemistry*, 89:359–388, 2020. (Cited on pages 14 and 15.)

[62] Xiaofeng Zhang, Xiechao Zhan, Chuangye Yan, Wenyu Zhang, Dongliang Liu, Jianlin Lei, and Yigong Shi. Structures of the human spliceosomes before and after release of the ligated exon. *Cell research*, 29(4):274–285, 2019. (Cited on page 14.)

[63] Yodfat Leader, Galit Lev Maor, Matan Sorek, Ronna Shayevitch, Maram Hussein, Ofir Hameiri, Luna Tammer, Jonathan Zonszain, Ifat Keydar, Dror Hollander, et al. The upstream 5' splice site remains associated to the transcription machinery during intron synthesis. *Nature Communications*, 12(1):1–11, 2021. (Cited on page 15.)

[64] Saurabh Chaudhary, Waqas Khokhar, Ibtissam Jabre, Anireddy SN Reddy, Lee J Byrne, Cornelia M Wilson, and Naeem H Syed. Alternative splicing and protein diversity: plants versus animals. *Frontiers in plant science*, 10:708, 2019. (Cited on page 16.)

[65] Jernej Ule and Benjamin J Blencowe. Alternative splicing regulatory networks: functions, mechanisms, and evolution. *Molecular cell*, 76(2):329–345, 2019. (Cited on page 16.)

[66] Yoseph Barash, John A Calarco, Weijun Gao, Qun Pan, Xinchen Wang, Ofer Shai, Benjamin J Blencowe, and Brendan J Frey. Deciphering the splicing code. *Nature*, 465(7294):53–59, 2010. (Cited on page 16.)

[67] Francisco E Baralle and Jimena Giudice. Alternative splicing as a regulator of development and tissue identity. *Nature reviews Molecular cell biology*, 18(7):437–451, 2017. (Cited on page 16.)

[68] Pavel V Mazin, Philipp Khaitovich, Margarida Cardoso-Moreira, and Henrik Kaessmann. Alternative splicing during mammalian organ development. *Nature Genetics*, 53(6):925–934, 2021. (Cited on page 16.)

[69] Marina M Scotti and Maurice S Swanson. RNA mis-splicing in disease. *Nature Reviews Genetics*, 17(1):19–32, 2016. (Cited on page 16.)

[70] Lu Chen, Stephen J Bush, Jaime M Tovar-Corona, Atahualpa Castillo-Morales, and Araxi O Urrutia. Correcting for differential transcript coverage reveals a strong relationship between alternative splicing and organism complexity. *Molecular biology and evolution*, 31(6):1402–1413, 2014. (Cited on page 16.)

[71] Ashley R Jurado, Dazhi Tan, Xinfu Jiao, Megerditch Kiledjian, and Liang Tong. Structure and function of pre-mRNA 5'-end capping quality control and 3'-end processing. *Biochemistry*, 53(12):1882–1898, 2014. (Cited on page 16.)

[72] Ian A Roundtree, Molly E Evans, Tao Pan, and Chuan He. Dynamic RNA modifications in gene expression regulation. *Cell*, 169(7):1187–1200, 2017. (Cited on page 16.)

[73] Eli Eisenberg and Erez Y Levanon. A-to-I RNA editing-immune protector and transcriptome diversifier. *Nature Reviews Genetics*, 19(8):473–490, 2018. (Cited on page 16.)

[74] Jiani Ma, Lin Zhang, Shutao Chen, and Hui Liu. A brief review of RNA modification related database resources. *Methods*, 2021. (Cited on page 16.)

[75] Krysta L Engel, Ankita Arora, Raeann Goering, Hei-Yong G Lo, and J Matthew Taliaferro. Mechanisms and consequences of subcellular RNA localization across diverse cell types. *Traffic*, 21(6):404–418, 2020. (Cited on pages 17 and 81.)

[76] Sulagna Das, Maria Vera, Valentina Gandin, Robert H Singer, and Evelina Tutucci. Intracellular mRNA transport and localized translation. *Nature Reviews Molecular Cell Biology*, 22(7):483–504, 2021. (Cited on page 17.)

[77] Mary Catherine Bridges, Amanda C Daulagala, and Antonis Kourtidis. LNCcation: lncRNA localization and function. *Journal of Cell Biology*, 220(2):e202009045, 2021. (Cited on page 17.)

[78] Giuseppe Nicastro, Adela M Candel, Michael Uhl, Alain Oregioni, David Hollingworth, Rolf Backofen, Stephen R Martin, and Andres Ramos. Mechanism of β-actin mRNA Recognition by ZBP1. *Cell reports*, 18(5):1187–1199, 2017. (Cited on page 17.)

[79] Carrie L Simms, Erica N Thomas, and Hani S Zaher. Ribosome-based quality control of mRNA and nascent peptides. *Wiley Interdisciplinary Reviews: RNA*, 8(1):e1366, 2017. (Cited on page 17.)

[80] Sara Andjus, Antonin Morillon, and Maxime Wery. From Yeast to Mammals, the Nonsense-Mediated mRNA Decay as a Master Regulator of Long Non-Coding RNAs Functional Trajectory. *Non-coding RNA*, 7(3):44, 2021. (Cited on page 17.)

[81] Sandra L Wolin and Lynne E Maquat. Cellular RNA surveillance in health and disease. *Science*, 366(6467):822–827, 2019. (Cited on pages 17 and 18.)

[82] KA Tatosyan, IG Ustyantsev, and DA Kramerov. RNA degradation in eukaryotic cells. *Molecular Biology*, 54(4):485–502, 2020. (Cited on page 18.)

[83] Luca FR Gebert and Ian J MacRae. Regulation of microRNA function in animals. *Nature reviews Molecular cell biology*, 20(1):21–37, 2019. (Cited on page 18.)

[84] William F Marzluff, Eric J Wagner, and Robert J Duronio. Metabolism and regulation of canonical histone mRNAs: life without a poly (A) tail. *Nature Reviews Genetics*, 9(11):843–854, 2008. (Cited on page 18.)

[85] Aaron C Goldstrohm, Traci M Tanaka Hall, and Katherine M McKenney. Post-transcriptional regulatory functions of mammalian pumilio proteins. *Trends in Genetics*, 34(12):972–990, 2018. (Cited on pages 18, 26 and 57.)

[86] Jesse Stombaugh, Craig L Zirbel, Eric Westhof, and Neocles B Leontis. Frequency and isostericity of RNA base pairs. *Nucleic acids research*, 37(7):2294–2312, 2009. (Cited on page 19.)

[87] Gabriele Varani and William H McClain. The G · U wobble base pair. *EMBO reports*, 1(1):18–23, 2000. (Cited on page 19.)

[88] Matthew G Seetin and David H Mathews. RNA structure prediction: an overview of methods. *Bacterial regulatory RNA*, pages 99–122, 2012. (Cited on page 19.)

[89] Laura R Ganser, Megan L Kelly, Daniel Herschlag, and Hashim M Al-Hashimi. The roles of structural dynamics in the cellular functions of RNAs. *Nature reviews Molecular cell biology*, 20(8):474–489, 2019. (Cited on page 19.)

[90] Ling X Shen and Ignacio Tinoco Jr. The structure of an RNA pseudoknot that causes efficient frameshifting in mouse mammary tumor virus. *Journal of molecular biology*, 247(5):963–978, 1995. (Cited on page 19.)

[91] Peter Kerpedjiev, Stefan Hammer, and Ivo L Hofacker. Forna (force-directed RNA): Simple and effective online RNA secondary structure diagrams. *Bioinformatics*, 31(20):3377–3379, 2015. (Cited on page 19.)

[92] Yikrazuul. X-ray structure of the phenylalanine tRNA from yeast. *Wikimedia Commons*, 2010. https://commons.wikimedia.org/wiki/File:TRNA-Phe_yeast_1ehz.png. (Cited on page 19.)

[93] Creative Commons organization. Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0) license. *Creative Commons Website*, 2021. https://creativecommons.org/licenses/by-sa/3.0. (Cited on page 19.)

[94] Cornelis WA Pleij. Pseudoknots: a new motif in the RNA game. *Trends in biochemical sciences*, 15(4):143–147, 1990. (Cited on page 19.)

[95] Robert T Batey, Robert P Rambo, and Jennifer A Doudna. Tertiary motifs in RNA structure and folding. *Angewandte Chemie International Edition*, 38(16):2326–2343, 1999. (Cited on page 20.)

[96] Samuel E Butcher and Anna Marie Pyle. The molecular interactions that stabilize RNA tertiary structure: RNA motifs, patterns, and networks. *Accounts of chemical research*, 44(12):1302–1311, 2011. (Cited on page 20.)

[97] Jessica A Brown, Max L Valenstein, Therese A Yario, Kazimierz T Tycowski, and Joan A Steitz. Formation of triple-helical structures by the 3'-end sequences of MALAT1 and MENβ noncoding RNAs. *Proceedings of the National Academy of Sciences*, 109(47):19202–19207, 2012. (Cited on page 20.)

[98] Jordan A Ramilowski, Chi Wai Yip, Saumya Agrawal, Jen-Chien Chang, Yari Ciani, Ivan V Kulakovskiy, Mickaël Mendez, Jasmine Li Ching Ooi, John F Ouyang, Nick Parkinson, et al. Functional annotation of human long noncoding RNAs via molecular phenotyping. *Genome research*, 30(7):1060–1072, 2020. (Cited on page 20.)

[99] Andreas Adam Greifenstein, SoYoung Jo, and Holger Bierhoff. RNA: DNA triple helices: from peculiar structures to pervasive chromatin regulators. *Essays in Biochemistry*, 2021. (Cited on page 20.)

[100] Xiaona Zhang, Yanchun Zhou, Shaoying Chen, Wei Li, Weibing Chen, and Wei Gu. LncRNA MACC1-AS1 sponges multiple miRNAs and RNA-binding protein PTBP1. *Oncogenesis*, 8(12):1–13, 2019. (Cited on page 20.)

[101] Joana Carlevaro-Fita and Rory Johnson. Global positioning system: understanding long noncoding RNAs through subcellular localization. *Molecular cell*, 73(5):869–883, 2019. (Cited on page 20.)

[102] Adam M Schmitt and Howard Y Chang. Long noncoding RNAs in cancer pathways. *Cancer cell*, 29(4):452–463, 2016. (Cited on page 20.)

[103] Katharina Jonas, George A Calin, and Martin Pichler. RNA-binding proteins as important regulators of long non-coding RNAs in cancer. *International journal of molecular sciences*, 21(8):2969, 2020. (Cited on page 21.)

[104] Matthew Cobb. A breakthrough from 60 years ago: "General nature of the genetic code for proteins"(1961). *Natural Sciences*, page e10018, 2021. (Cited on page 21.)

[105] Jamie A Kelly, Alexandra N Olson, Krishna Neupane, Sneha Munshi, Josue San Emeterio, Lois Pollack, Michael T Woodside, and Jonathan D Dinman. Structural and functional conservation of the programmed -1 ribosomal frameshift signal of SARS coronavirus 2 (SARS-CoV-2). *Journal of Biological Chemistry*, 295(31):10741–10748, 2020. (Cited on page 21.)

[106] Alexandre Ambrogelly, Sotiria Palioura, and Dieter Söll. Natural expansion of the genetic code. *Nature chemical biology*, 3(1):29–35, 2007. (Cited on page 21.)

[107] Patrick J Keeling. Genomics: evolution of the genetic code. *Current Biology*, 26(18):R851–R853, 2016. (Cited on page 21.)

[108] Gloria A Brar. Beyond the triplet code: context cues transform translation. *Cell*, 167(7):1681–1692, 2016. (Cited on pages 21 and 22.)

[109] Daniel N Wilson and Jamie H Doudna Cate. The structure and function of the eukaryotic ribosome. *Cold Spring Harbor perspectives in biology*, 4(5):a011536, 2012. (Cited on page 21.)

[110] Gavin Hanson and Jeff Coller. Codon optimality, bias and usage in translation and mRNA decay. *Nature reviews Molecular cell biology*, 19(1):20–30, 2018. (Cited on pages 22 and 23.)

[111] Kathrin Leppek, Rhiju Das, and Maria Barna. Functional 5' UTR mRNA structures in eukaryotic translation regulation and how to find them. *Nature reviews Molecular cell biology*, 19(3):158–174, 2018. (Cited on page 23.)

[112] John WB Hershey, Nahum Sonenberg, and Michael B Mathews. Principles of translational control. *Cold Spring Harbor perspectives in biology*, 11(9):a032607, 2019. (Cited on page 23.)

[113] Stijn Sonneveld, Bram MP Verhagen, and Marvin E Tanenbaum. Heterogeneity in mRNA translation. *Trends in Cell Biology*, 30(8):606–618, 2020. (Cited on page 23.)

[114] Robert F Harvey, Tom S Smith, Thomas Mulroney, Rayner ML Queiroz, Mariavittoria Pizzinga, Veronica Dezi, Eneko Villenueva, Manasa Ramakrishna, Kathryn S Lilley, and Anne E Willis. Trans-acting translational regulatory RNA binding proteins. *Wiley Interdisciplinary Reviews: RNA*, 9(3):e1465, 2018. (Cited on page 23.)

[115] Jacob O'Brien, Heyam Hayder, Yara Zayed, and Chun Peng. Overview of microRNA biogenesis, mechanisms of actions, and circulation. *Frontiers in endocrinology*, 9:402, 2018. (Cited on page 23.)

[116] Ping Song, Fan Yang, Hongchuan Jin, and Xian Wang. The regulation of protein translation and its implications for cancer. *Signal Transduction and Targeted Therapy*, 6(1):1–9, 2021. (Cited on page 23.)

[117] Valentina Iadevaia and André P Gerber. Combinatorial control of mRNA fates by RNA-binding proteins and non-coding RNAs. *Biomolecules*, 5(4):2207–2222, 2015. (Cited on page 23.)

[118] Erin L Sternburg, Jason A Estep, Daniel K Nguyen, Yahui Li, and Fedor V Karginov. Antagonistic and cooperative AGO2-PUM interactions in regulating mRNAs. *Scientific reports*, 8(1):1–13, 2018. (Cited on page 23.)

[119] Sukjun Kim, Soyoung Kim, Hee Ryung Chang, Doyeon Kim, Junehee Park, Narae Son, Joori Park, Minhyuk Yoon, Gwangung Chae, Young-Kook Kim, et al. The regulatory impact of RNA-binding proteins on microRNA targeting. *Nature communications*, 12(1):1–15, 2021. (Cited on page 23.)

[120] David Balchin, Manajit Hayer-Hartl, and F Ulrich Hartl. In vivo aspects of protein folding and quality control. *Science*, 353(6294), 2016. (Cited on page 23.)

[121] Diana Ekman, Åsa K Björklund, Johannes Frey-Skött, and Arne Elofsson. Multi-domain proteins in the three kingdoms of life: orphan domains and other unassigned regions. *Journal of molecular biology*, 348(1):231–243, 2005. (Cited on page 23.)

[122] Jung-Hoon Han, Sarah Batey, Adrian A Nickson, Sarah A Teichmann, and Jane Clarke. The folding and evolution of multidomain proteins. *Nature Reviews Molecular Cell Biology*, 8(4):319–330, 2007. (Cited on page 23.)

[123] Peter E Wright and H Jane Dyson. Intrinsically disordered proteins in cellular signalling and regulation. *Nature reviews Molecular cell biology*, 16(1):18–29, 2015. (Cited on page 23.)

[124] Peter Tompa, Norman E Davey, Toby J Gibson, and M Madan Babu. A million peptide motifs for the molecular biologist. *Molecular cell*, 55(2):161–169, 2014. (Cited on page 23.)

[125] Peter D Sun, Christine E Foster, and Jeffrey C Boyington. Overview of protein structural and functional folds. *Current protocols in protein science*, 35(1):17–1, 2004. (Cited on page 23.)

[126] Chantal Christis, Nicolette H Lubsen, and Ineke Braakman. Protein folding includes oligomerization–examples from the endoplasmic reticulum and cytosol. *The FEBS journal*, 275(19):4700–4727, 2008. (Cited on pages 23 and 24.)

[127] Shahin Ramazi and Javad Zahiri. Posttranslational modifications in proteins: resources, tools and prediction methods. *Database*, 2021, 2021. (Cited on page 24.)

[128] Alejandro Velázquez-Cruz, Blanca Baños-Jaime, Antonio Díaz-Quintana, Miguel A De la Rosa, and Irene Díaz-Moreno. Post-translational control of RNA-binding proteins and disease-related dysregulation. *Frontiers in molecular biosciences*, 8, 2021. (Cited on page 24.)

[129] Stefanie Gerstberger, Markus Hafner, and Thomas Tuschl. A census of human RNA-binding proteins. *Nature Reviews Genetics*, 15(12):829–845, 2014. (Cited on page 24.)

[130] Junichiro Sonoda and Robin P Wharton. Recruitment of Nanos to hunchback mRNA by Pumilio. *Genes & development*, 13(20):2704–2712, 1999. (Cited on page 24.)

[131] Sarah F Mitchell and Roy Parker. Principles and properties of eukaryotic mRNPs. *Molecular cell*, 54(4):547–558, 2014. (Cited on pages 25 and 49.)

[132] Eckhard Jankowsky and Michael E Harris. Specificity and nonspecificity in RNA-protein interactions. *Nature reviews Molecular cell biology*, 16(9):533–544, 2015. (Cited on pages 25 and 81.)

[133] Lyudmila Dimitrova-Paternoga, Kevin Haubrich, Mai Sun, Anne Ephrussi, Janosch Hennig, et al. Validation and classification of RNA binding proteins identified by mRNA interactome capture. *RNA*, 27(10):1173–1185, 2021. (Cited on page 25.)

[134] Yue Ren, Yue Huo, Weiqian Li, Manman He, Siqi Liu, Jiabin Yang, Hongmei Zhao, Lingjie Xu, Yuehong Guo, Yanmin Si, et al. A global screening identifies chromatin-enriched RNA-binding proteins and the transcriptional regulatory activity of QKI5 during monocytic differentiation. *Genome biology*, 22(1):1–32, 2021. (Cited on page 25.)

[135] Vivek L Patel, Somdeb Mitra, Richard Harris, Adina R Buxbaum, Timothée Lionnet, Michael Brenowitz, Mark Girvin, Matthew Levy, Steven C Almo, Robert H Singer, et al. Spatial arrangement of an RNA zipcode identifies mRNAs under post-transcriptional control. *Genes & development*, 26(1):43–53, 2012. (Cited on page 26.)

[136] Jiaxu Wang, Tong Zhang, Zhang Yu, Wen Ting Tan, Ming Wen, Yang Shen, Finnlay RP Lambert, Roland G Huber, and Yue Wan. Genome-wide RNA structure changes during human neurogenesis modulate gene regulatory networks. *Molecular Cell*, 81(23):4942–4953, 2021. (Cited on page 26.)

[137] Cole JT Lewis, Tao Pan, and Auinash Kalsotra. RNA modifications and structures cooperate to guide RNA–protein interactions. *Nature reviews Molecular cell biology*, 18(3):202–210, 2017. (Cited on page 26.)

[138] Johannes Braun, Sandra Fischer, Zhenjiang Z Xu, Hongying Sun, Dalia H Ghoneim, Anna T Gimbel, Uwe Plessmann, Henning Urlaub, David H Mathews, and Julia E Weigand. Identification of new high affinity targets for Roquin based on structural conservation. *Nucleic acids research*, 46(22):12109–12125, 2018. (Cited on pages 26 and 68.)

[139] Dazhi Tan, William F Marzluff, Zbigniew Dominski, and Liang Tong. Structure of histone mRNA stem-loop, human stem-loop binding protein, and 3' hExo ternary complex. *Science*, 339(6117):318–321, 2013. (Cited on page 26.)

[140] Arttu Jolma, Jilin Zhang, Estefania Mondragón, Ekaterina Morgunova, Teemu Kivioja, Kaitlin U Laverty, Yimeng Yin, Fangjie Zhu, Gleb Bourenkov, Quaid Morris, et al. Binding specificities of human RNA-binding proteins toward structured and linear RNA sequences. *Genome research*, 30(7):962–973, 2020. (Cited on pages 26 and 49.)

[141] Grégoire Masliah, Pierre Barraud, and Frédéric H-T Allain. RNA recognition by double-stranded RNA binding domains: a matter of shape and sequence. *Cellular and Molecular Life Sciences*, 70(11):1875–1895, 2013. (Cited on page 26.)

[142] Jacki E Heraud-Farlow and Michael A Kiebler. The multifunctional Staufen proteins: conserved roles from neurogenesis to synaptic plasticity. *Trends in neurosciences*, 37(9):470–479, 2014. (Cited on page 26.)

[143] J Matthew Taliaferro, Nicole J Lambert, Peter H Sudmant, Daniel Dominguez, Jason J Merkin, Maria S Alexis, Cassandra A Bazile, and Christopher B Burge. RNA sequence context effects measured in vitro predict in vivo protein binding and regulation. *Molecular cell*, 64(2):294–306, 2016. (Cited on page 26.)

[144] Daniel Dominguez, Peter Freese, Maria S Alexis, Amanda Su, Myles Hochman, Tsultrim Palden, Cassandra Bazile, Nicole J Lambert, Eric L Van Nostrand, Gabriel A Pratt, et al. Sequence, structure, and context preferences of human RNA binding proteins. *Molecular cell*, 70(5):854–867, 2018. (Cited on page 26.)

[145] Donghee Kang, Yerim Lee, and Jae-Seon Lee. RNA-binding proteins in cancer: functional and therapeutic perspectives. *Cancers*, 12(9):2699, 2020. (Cited on page 26.)

[146] National Human Genome Research Institute. The Cost of Sequencing a Human Genome. https://www.genome.gov/sequencingcosts. (Cited on page 26.)

[147] Rory Stark, Marta Grzelak, and James Hadfield. RNA sequencing: the teenage years. *Nature Reviews Genetics*, 20(11):631–656, 2019. (Cited on pages 27, 29 and 81.)

[148] Morgane Boone, Andries De Koker, and Nico Callewaert. Capturing the 'ome': the expanding molecular toolbox for RNA and DNA library construction. *Nucleic Acids Research*, 46(6):2701–2721, 2018. (Cited on pages 27 and 28.)

[149] Radmila Hrdlickova, Masoud Toloue, and Bin Tian. RNA-Seq methods for transcriptome analysis. *Wiley Interdisciplinary Reviews: RNA*, 8(1):e1364, 2017. (Cited on page 28.)

[150] Kary Mullis, Fred Faloona, Stephen Scharf, Randall Saiki, Glenn Horn, and Henry Erlich. Specific enzymatic amplification of DNA in vitro: the polymerase chain reaction. In *Cold Spring Harbor symposia on quantitative biology*, volume 51, pages 263–273. Cold Spring Harbor Laboratory Press, 1986. (Cited on page 28.)

[151] Wilhelm J Ansorge. Next-generation DNA sequencing techniques. *New biotechnology*, 25(4):195–203, 2009. (Cited on page 28.)

[152] Nicholas Stoler and Anton Nekrutenko. Sequencing error profiles of Illumina sequencing instruments. *NAR genomics and bioinformatics*, 3(1):lqab019, 2021. (Cited on page 29.)

[153] Fadia Ibrahim, Jan Oppelt, Manolis Maragkakis, and Zissimos Mourelatos. TERA-Seq: true end-to-end sequencing of native RNA molecules for transcriptome characterization. *Nucleic acids research*, 49(20):e115–e115, 2021. (Cited on page 29.)

[154] Nicola De Maio, Liam P Shaw, Alasdair Hubbard, Sophie George, Nicholas D Sanderson, Jeremy Swann, Ryan Wick, Manal AbuOun, Emma Stubberfield, Sarah J Hoosdally, et al. Comparison of long-read sequencing technologies in the hybrid assembly of complex bacterial genomes. *Microbial genomics*, 5(9), 2019. (Cited on page 29.)

[155] Muthukumar Ramanathan, Douglas F Porter, and Paul A Khavari. Methods to study RNA–protein interactions. *Nature methods*, 16(3):225–234, 2019. (Cited on pages 29, 30 and 81.)

[156] Emiliano P Ricci, Alper Kucukural, Can Cenik, Blandine C Mercier, Guramrit Singh, Erin E Heyer, Ami Ashar-Patel, Lingtao Peng, and Melissa J Moore. Staufen1 senses overall transcript secondary structure to regulate translation. *Nature structural & molecular biology*, 21(1):26–35, 2014. (Cited on pages 30 and 49.)

[157] Jesse M Engreitz, Amy Pandya-Jones, Patrick McDonel, Alexander Shishkin, Klara Sirokman, Christine Surka, Sabah Kadri, Jeffrey Xing, Alon Goren, Eric S Lander,

et al. The Xist lncRNA exploits three-dimensional genome architecture to spread across the X chromosome. *Science*, 341(6147):1237973, 2013. (Cited on page 30.)

[158] Colleen A McHugh and Mitchell Guttman. RAP-MS: a method to identify proteins that interact directly with a specific RNA molecule in cells. In *RNA Detection*, pages 473–488. Springer, 2018. (Cited on page 30.)

[159] Muthukumar Ramanathan, Karim Majzoub, Deepti S Rao, Poornima H Neela, Brian J Zarnegar, Smarajit Mondal, Julien G Roth, Hui Gai, Joanna R Kovalski, Zurab Siprashvili, et al. RNA–protein interaction detection in living cells. *Nature methods*, 15(3):207–212, 2018. (Cited on page 30.)

[160] Baekgyu Kim and V Narry Kim. fCLIP-seq for transcriptomic footprinting of dsRNA-binding proteins: Lessons from DROSHA. *Methods*, 152:3–11, 2019. (Cited on page 30.)

[161] Jing Zhao, Toshiro K Ohsumi, Johnny T Kung, Yuya Ogawa, Daniel J Grau, Kavitha Sarma, Ji Joon Song, Robert E Kingston, Mark Borowsky, and Jeannie T Lee. Genome-wide identification of polycomb-associated RNAs by RIP-seq. *Molecular cell*, 40(6):939–953, 2010. (Cited on page 30.)

[162] Markus Hafner, Markus Landthaler, Lukas Burger, Mohsen Khorshid, Jean Hausser, Philipp Berninger, Andrea Rothballer, Manuel Ascano Jr, Anna-Carina Jungkamp, Mathias Munschauer, et al. Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP. *Cell*, 141(1):129–141, 2010. (Cited on pages 30, 31, 33 and 56.)

[163] Zhi-Ren Liu, Arlene M Wilkie, Michael J Clemens, and CW Smith. Detection of double-stranded RNA-protein interactions by methylene blue-mediated photo-crosslinking. *Rna*, 2(6):611–621, 1996. (Cited on page 30.)

[164] Aoife C McMahon, Reazur Rahman, Hua Jin, James L Shen, Allegra Fieldsend, Weifei Luo, and Michael Rosbash. TRIBE: hijacking an RNA-editing enzyme to identify cell-specific targets of RNA-binding proteins. *Cell*, 165(3):742–753, 2016. (Cited on page 31.)

[165] Jernej Ule, Kirk B Jensen, Matteo Ruggiu, Aldo Mele, Aljaz Ule, and Robert B Darnell. CLIP identifies Nova-regulated RNA networks in the brain. *Science*, 302(5648):1212–1215, 2003. (Cited on page 31.)

[166] Donny D Licatalosi, Aldo Mele, John J Fak, Jernej Ule, Melis Kayikci, Sung Wook Chi, Tyson A Clark, Anthony C Schweitzer, John E Blume, Xuning Wang, et al. HITS-CLIP yields genome-wide insights into brain alternative RNA processing. *Nature*, 456(7221):464–469, 2008. (Cited on page 31.)

[167] Julian König, Kathi Zarnack, Gregor Rot, Tomaž Curk, Melis Kayikci, Blaž Zupan, Daniel J Turner, Nicholas M Luscombe, and Jernej Ule. iCLIP reveals the function of

hnRNP particles in splicing at individual nucleotide resolution. *Nature structural & molecular biology*, 17(7):909–915, 2010. (Cited on pages 31 and 33.)

[168] Eric L Van Nostrand, Gabriel A Pratt, Alexander A Shishkin, Chelsea Gelboin-Burkhart, Mark Y Fang, Balaji Sundararaman, Steven M Blue, Thai B Nguyen, Christine Surka, Keri Elkins, et al. Robust transcriptome-wide discovery of RNA-binding protein binding sites with enhanced CLIP (eCLIP). *Nature methods*, 13(6):508–514, 2016. (Cited on pages 31 and 34.)

[169] Aleksandra Helwak, Grzegorz Kudla, Tatiana Dudnakova, and David Tollervey. Mapping the human miRNA interactome by CLASH reveals frequent noncanonical binding. *Cell*, 153(3):654–665, 2013. (Cited on page 31.)

[170] Yoichiro Sugimoto, Alessandra Vigilante, Elodie Darbo, Alexandra Zirra, Cristina Militti, Andrea D'Ambrogio, Nicholas M Luscombe, and Jernej Ule. hiCLIP reveals the in vivo atlas of mRNA secondary structures recognized by Staufen 1. *Nature*, 519(7544):491–494, 2015. (Cited on page 31.)

[171] Michael J Moore, Troels KH Scheel, Joseph M Luna, Christopher Y Park, John J Fak, Eiko Nishiuchi, Charles M Rice, and Robert B Darnell. miRNA–target chimeras reveal miRNA 3'-end pairing as a major determinant of Argonaute target specificity. *Nature communications*, 6(1):1–17, 2015. (Cited on page 31.)

[172] Alexander G Baltz, Mathias Munschauer, Björn Schwanhäusser, Alexandra Vasile, Yasuhiro Murakawa, Markus Schueler, Noah Youngs, Duncan Penfold-Brown, Kevin Drew, Miha Milek, et al. The mRNA-bound proteome and its global occupancy profile on protein-coding transcripts. *Molecular cell*, 46(5):674–690, 2012. (Cited on page 31.)

[173] Markus Hafner, Maria Katsantoni, Tino Köster, James Marks, Joyita Mukherjee, Dorothee Staiger, Jernej Ule, and Mihaela Zavolan. CLIP and complementary methods. *Nature Reviews Methods Primers*, 1(1):1–23, 2021. (Cited on page 32.)

[174] Kendric C Smith and Robin T Aplin. A mixed photoproduct of uracil and cysteine (5-S-cysteine-6-hydrouracil). A possible model for the in vivo cross-linking of deoxyribonucleic acid and protein by ultraviolet light. *Biochemistry*, 5(6):2125–2130, 1966. (Cited on page 31.)

[175] Katharina Kramer, Timo Sachsenberg, Benedikt M Beckmann, Saadia Qamar, Kum-Loong Boon, Matthias W Hentze, Oliver Kohlbacher, and Henning Urlaub. Photo-cross-linking and high-resolution mass spectrometry for assignment of RNA-binding sites in RNA-binding proteins. *Nature methods*, 11(10):1064–1070, 2014. (Cited on page 31.)

[176] Stephen M Testa, Matthew D Disney, Douglas H Turner, and Ryszard Kierzek. Thermodynamics of RNA-RNA duplexes with 2-or 4-thiouridines: implications for antisense

design and targeting a group I intron. *Biochemistry*, 38(50):16655–16662, 1999. (Cited on page 33.)

[177] Shivendra Kishore, Lukasz Jaskiewicz, Lukas Burger, Jean Hausser, Mohsen Khorshid, and Mihaela Zavolan. A quantitative analysis of CLIP methods for identifying binding sites of RNA-binding proteins. *Nature methods*, 8(7):559–564, 2011. (Cited on page 33.)

[178] Anna-Carina Jungkamp, Marlon Stoeckius, Desirea Mecenas, Dominic Grün, Guido Mastrobuoni, Stefan Kempa, and Nikolaus Rajewsky. In vivo and transcriptome-wide identification of RNA binding protein target sites. *Molecular cell*, 44(5):828–840, 2011. (Cited on page 33.)

[179] Dimitrios G Anastasakis, Alexis Jacob, Parthena Konstantinidou, Kazuyuki Meguro, Duncan Claypool, Pavol Cekan, Astrid D Haase, and Markus Hafner. A non-radioactive, improved PAR-CLIP and small RNA cDNA library preparation protocol. *Nucleic Acids Research*, 49(8):e45–e45, 2021. (Cited on page 33.)

[180] Neelanjan Mukherjee, Hans-Hermann Wessels, Svetlana Lebedeva, Marcin Sajek, Mahsa Ghanbari, Aitor Garzia, Alina Munteanu, Dilmurat Yusuf, Thalia Farazi, Jessica I Hoell, et al. Deciphering human ribonucleoprotein regulatory networks. *Nucleic acids research*, 47(2):570–581, 2019. (Cited on page 33.)

[181] Yoichiro Sugimoto, Julian König, Shobbir Hussain, Blaž Zupan, Tomaž Curk, Michaela Frye, and Jernej Ule. Analysis of CLIP and iCLIP methods for nucleotide-resolution studies of protein-RNA interactions. *Genome biology*, 13(8):1–13, 2012. (Cited on page 33.)

[182] Georges Martin and Mihaela Zavolan. Redesigning CLIP for efficiency, accuracy and speed. *Nature methods*, 13(6):482–483, 2016. (Cited on pages 33, 34 and 83.)

[183] Andreas Buchbender, Holger Mutter, FX Reymond Sutandy, Nadine Körtel, Heike Hänel, Anke Busch, Stefanie Ebersberger, and Julian König. Improved library preparation with the new iCLIP2 protocol. *Methods*, 178:33–48, 2020. (Cited on page 33.)

[184] Flora CY Lee, Anob M Chakrabarti, Heike Hänel, Elisa Monzón-Casanova, Martina Hallegger, Cristina Militti, Federica Capraro, Christoph Sadée, Patrick Toolan-Kerr, Oscar Wilkins, et al. An improved iCLIP protocol. *bioRxiv*, 2021. (Cited on page 33.)

[185] Eric L Van Nostrand, Peter Freese, Gabriel A Pratt, Xiaofeng Wang, Xintao Wei, Rui Xiao, Steven M Blue, Jia-Yu Chen, Neal AL Cody, Daniel Dominguez, et al. A large-scale binding and functional map of human RNA-binding proteins. *Nature*, 583(7818):711–719, 2020. (Cited on page 34.)

[186] Ana Conesa, Pedro Madrigal, Sonia Tarazona, David Gomez-Cabrero, Alejandra Cervera, Andrew McPherson, Michał Wojciech Szcześniak, Daniel J Gaffney, Laura L

Elo, Xuegong Zhang, et al. A survey of best practices for RNA-seq data analysis. *Genome biology*, 17(1):1–19, 2016. (Cited on pages 34, 36 and 51.)

[187] Michael Uhl, Torsten Houwaart, Gianluca Corrado, Patrick R Wright, and Rolf Backofen. Computational analysis of CLIP-seq data. *Methods*, 118:60–72, 2017. (Cited on pages 34, 36 and 50.)

[188] Florian Heyl, Daniel Maticzka, Michael Uhl, and Rolf Backofen. Galaxy CLIP-Explorer: a web server for CLIP-Seq data analysis. *GigaScience*, 9(11):giaa108, 2020. (Cited on pages 34, 50 and 52.)

[189] Tom Smith, Andreas Heger, and Ian Sudbery. UMI-tools: modeling sequencing errors in Unique Molecular Identifiers to improve quantification accuracy. *Genome research*, 27(3):491–499, 2017. (Cited on page 35.)

[190] Alexander Dobin, Carrie A Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R Gingeras. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21, 2013. (Cited on page 35.)

[191] Timothy L Bailey, James Johnson, Charles E Grant, and William S Noble. The MEME suite. *Nucleic acids research*, 43(W1):W39–W49, 2015. (Cited on page 36.)

[192] Hilal Kazan, Debashish Ray, Esther T Chan, Timothy R Hughes, and Quaid Morris. RNAcontext: a new method for learning the sequence and structure binding preferences of RNA-binding proteins. *PLoS computational biology*, 6(7):e1000832, 2010. (Cited on page 36.)

[193] Daniel Maticzka, Sita J Lange, Fabrizio Costa, and Rolf Backofen. GraphProt: modeling binding preferences of RNA-binding proteins. *Genome biology*, 15(1):1–18, 2014. (Cited on pages 36 and 52.)

[194] Debashish Ray, Hilal Kazan, Kate B Cook, Matthew T Weirauch, Hamed S Najafabadi, Xiao Li, Serge Gueroussov, Mihai Albu, Hong Zheng, Ally Yang, et al. A compendium of RNA-binding motifs for decoding gene regulation. *Nature*, 499(7457):172–177, 2013. (Cited on pages 36 and 49.)

[195] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature biotechnology*, 33(8):831–838, 2015. (Cited on page 36.)

[196] Xiaoyong Pan, Yang Yang, Chun-Qiu Xia, Aashiq H Mirza, and Hong-Bin Shen. Recent methodology progress of deep learning for RNA–protein interaction prediction. *Wiley Interdisciplinary Reviews: RNA*, 10(6):e1544, 2019. (Cited on page 36.)

[197] Mahsa Ghanbari and Uwe Ohler. Deep neural networks for interpreting RNA-binding protein target preferences. *Genome research*, 30(2):214–226, 2020. (Cited on page 37.)

[198] Peter K Koo and Matt Ploenzke. Deep learning for inferring transcription factor binding sites. *Current opinion in systems biology*, 19:16–23, 2020. (Cited on pages 37 and 66.)

[199] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. (Cited on pages 37 and 39.)

[200] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. (Cited on page 39.)

[201] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. (Cited on page 40.)

[202] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. (Cited on page 40.)

[203] Stefan Falkner, Aaron Klein, and Frank Hutter. BOHB: Robust and efficient hyperparameter optimization at scale. In *International Conference on Machine Learning*, pages 1437–1446. PMLR, 2018. (Cited on pages 41 and 61.)

[204] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997. (Cited on page 42.)

[205] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994. (Cited on page 42.)

[206] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. (Cited on page 42.)

[207] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014. (Cited on pages 42 and 61.)

[208] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. (Cited on page 42.)

[209] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015. (Cited on page 44.)

[210] Sabrina Krakau, Hugues Richard, and Annalisa Marsico. PureCLIP: capturing target-specific protein–RNA interaction footprints from single-nucleotide CLIP-seq data. *Genome biology*, 18(1):1–17, 2017. (Cited on pages 49 and 52.)

[211] Cricket A Sloan, Esther T Chan, Jean M Davidson, Venkat S Malladi, J Seth Strattan, Benjamin C Hitz, Idan Gabdank, Aditi K Narayanan, Marcus Ho, Brian T Lee, et al. ENCODE data at the ENCODE portal. *Nucleic acids research*, 44(D1):D726–D732, 2016. (Cited on pages 51 and 57.)

[212] Yashar S Niknafs, Sumin Han, Teng Ma, Corey Speers, Chao Zhang, Kari Wilder-Romans, Matthew K Iyer, Sethuramasundaram Pitchiaya, Rohit Malik, Yasuyuki Hosono, et al. The lncRNA landscape of breast cancer reveals a role for DSCAM-AS1 in breast cancer progression. *Nature communications*, 7(1):1–13, 2016. (Cited on page 51.)

[213] Creative Commons organization. Attribution 4.0 International (CC BY 4.0) license. *Creative Commons Website*, 2021. https://creativecommons.org/licenses/by/4.0. (Cited on page 51.)

[214] Michael Daume, Michael Uhl, Rolf Backofen, and Lennart Randau. RIP-Seq suggests translational regulation by L7Ae in Archaea. *MBio*, 8(4):e00730–17, 2017. (Cited on page 52.)

[215] Guramrit Singh, Emiliano P Ricci, and Melissa J Moore. RIPiT-Seq: a high-throughput approach for footprinting RNA: protein complexes. *Methods*, 65(3):320–332, 2014. (Cited on page 52.)

[216] Enis Afgan, Dannon Baker, Bérénice Batut, Marius Van Den Beek, Dave Bouvier, Martin Čech, John Chilton, Dave Clements, Nate Coraor, Björn A Grüning, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic acids research*, 46(W1):W537–W544, 2018. (Cited on page 52.)

[217] Bora Uyar, Dilmurat Yusuf, Ricardo Wurmus, Nikolaus Rajewsky, Uwe Ohler, and Altuna Akalin. RCAS: an RNA centric annotation system for transcriptome-wide regions of interest. *Nucleic acids research*, 45(10):e91–e91, 2017. (Cited on page 52.)

[218] Martin Mann, Patrick R Wright, and Rolf Backofen. IntaRNA 2.0: enhanced and customizable prediction of RNA–RNA interactions. *Nucleic acids research*, 45(W1):W435–W439, 2017. (Cited on page 53.)

[219] Alexander R Gawronski, Michael Uhl, Yajia Zhang, Yen-Yi Lin, Yashar S Niknafs, Varune R Ramnarine, Rohit Malik, Felix Feng, Arul M Chinnaiyan, Colin C Collins, et al. MechRNA: prediction of lncRNA mechanisms from RNA–RNA and RNA–protein interactions. *Bioinformatics*, 34(18):3101–3110, 2018. (Cited on pages 54, 56 and 59.)

[220] Christopher W Schultz, Ranjan Preet, Teena Dhir, Dan A Dixon, and Jonathan R Brody. Understanding and targeting the disease-related RNA binding protein human antigen R (HuR). *Wiley Interdisciplinary Reviews: RNA*, 11(3):e1581, 2020. (Cited on page 57.)

[221] John R Prensner, Wei Chen, Matthew K Iyer, Qi Cao, Teng Ma, Sumin Han, Anirban Sahu, Rohit Malik, Kari Wilder-Romans, Nora Navone, et al. PCAT-1, a long noncoding RNA, regulates BRCA2 and controls homologous recombination in cancer. *Cancer research*, 74(6):1651–1660, 2014. (Cited on pages 58 and 59.)

[222] Michael Uhl, Van Dinh Tran, Florian Heyl, and Rolf Backofen. RNAProt: an efficient and feature-rich RNA binding protein binding site predictor. *GigaScience*, 10(8):giab054, 2021. (Cited on pages 62, 65, 66, 67 and 68.)

[223] Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and Understanding Neural Models in NLP. *arXiv preprint arXiv:1506.01066*, 2015. (Cited on page 63.)

[224] Alexander Gulliver Bjørnholt Grønning, Thomas Koed Doktor, Simon Jonas Larsen, Ulrika Simone Spangsberg Petersen, Lise Lolle Holm, Gitte Hoffmann Bruun, Michael Birkerod Hansen, Anne-Mette Hartung, Jan Baumbach, and Brage Storstein Andresen. DeepCLIP: predicting the effect of mutations on protein–RNA binding with deep learning. *Nucleic acids research*, 48(13):7099–7118, 2020. (Cited on page 64.)

[225] Ameni Trabelsi, Mohamed Chaabane, and Asa Ben-Hur. Comprehensive evaluation of deep learning architectures for prediction of DNA/RNA sequence binding specificities. *Bioinformatics*, 35(14):i269–i277, 2019. (Cited on page 64.)

[226] Girolamo Giudice, Fátima Sánchez-Cabo, Carlos Torroja, and Enrique Lara-Pezzi. ATtRACT - a database of RNA-binding proteins and associated motifs. *Database*, 2016, 2016. (Cited on page 67.)

[227] Michael Uhl, Van Dinh Tran, and Rolf Backofen. Improving CLIP-seq data analysis by incorporating transcript information. *BMC genomics*, 21(1):1–8, 2020. (Cited on pages 70, 71 and 73.)

[228] Tim Schneider, Lee-Hsueh Hung, Masood Aziz, Anna Wilmen, Stephanie Thaum, Jacqueline Wagner, Robert Janowski, Simon Müller, Silke Schreiner, Peter Friedhoff, et al. Combinatorial recognition of clustered RNA elements by the multidomain RNA-binding protein IMP3. *Nature communications*, 10(1):1–18, 2019. (Cited on page 73.)

[229] Michael Uhl, Dominik Rabsch, Florian Eggenhofer, and Rolf Backofen. Peakhood: individual site context extraction for CLIP-seq peak regions. *Bioinformatics*, 2021. (Cited on pages 75, 76 and 78.)

[230] Chenyu Lin and Wayne O Miles. Beyond CLIP: advances and opportunities to measure RBP–RNA and RNA–RNA interactions. *Nucleic acids research*, 47(11):5490–5501, 2019. (Cited on page 81.)

[231] FX Reymond Sutandy, Stefanie Ebersberger, Lu Huang, Anke Busch, Maximilian Bach, Hyun-Seo Kang, Jörg Fallmann, Daniel Maticzka, Rolf Backofen, Peter F Stadler, et al. In vitro iCLIP-based modeling uncovers how the splicing factor U2AF2

relies on regulation by cofactors. *Genome research*, 28(5):699–713, 2018. (Cited on page 81.)

[232] Lisa M Strittmatter, Charlotte Capitanchik, Andrew J Newman, Martina Hallegger, Christine M Norman, Sebastian M Fica, Chris Oubridge, Nicholas M Luscombe, Jernej Ule, and Kiyoshi Nagai. psiCLIP reveals dynamic RNA binding by DEAH-box helicases before and after exon ligation. *Nature communications*, 12(1):1–15, 2021. (Cited on page 81.)

[233] Mattia Brugiolo, Valentina Botti, Na Liu, Michaela Müller-McNicoll, and Karla M Neugebauer. Fractionation iCLIP detects persistent SR protein binding to conserved, retained introns in chromatin, nucleoplasm and cytoplasm. *Nucleic acids research*, 45(18):10452–10465, 2017. (Cited on page 81.)

[234] Daniel Benhalevy, Dimitrios G Anastasakis, and Markus Hafner. Proximity-CLIP provides a snapshot of protein-occupied RNA elements in subcellular compartments. *Nature methods*, 15(12):1074–1082, 2018. (Cited on page 81.)

[235] Lei Sun, Furqan M Fazal, Pan Li, James P Broughton, Byron Lee, Lei Tang, Wenze Huang, Eric T Kool, Howard Y Chang, and Qiangfeng Cliff Zhang. RNA structure maps across mammalian cellular compartments. *Nature structural & molecular biology*, 26(4):322–330, 2019. (Cited on page 81.)

[236] Marina Garcia-Jove Navarro, Shunnichi Kashida, Racha Chouaib, Sylvie Souquere, Gerard Pierron, Dominique Weil, and Zoher Gueroui. RNA is a critical element for the sizing and the composition of phase-separated RNA–protein condensates. *Nature communications*, 10(1):1–13, 2019. (Cited on page 81.)

[237] Diana Kwon et al. The secret lives of cells - as never seen before. *Nature*, 598(7882):558–560, 2021. (Cited on page 81.)

[238] Deepak Sharma, Leah L Zagore, Matthew M Brister, Xuan Ye, Carlos E Crespo-Hernández, Donny D Licatalosi, and Eckhard Jankowsky. The kinetic landscape of an RNA-binding protein in cells. *Nature*, 591(7848):152–156, 2021. (Cited on page 81.)

[239] Joe G Greener, Shaun M Kandathil, Lewis Moffat, and David T Jones. A guide to machine learning for biologists. *Nature Reviews Molecular Cell Biology*, 23(1):40–55, 2022. (Cited on page 82.)

[240] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020. (Cited on page 82.)

[241] Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. Text data augmentation for deep learning. *Journal of big Data*, 8(1):1–34, 2021. (Cited on page 82.)

[242] Gherman Novakovsky, Manu Saraswat, Oriol Fornes, Sara Mostafavi, and Wyeth W Wasserman. Biologically relevant transfer learning improves transcription factor binding prediction. *Genome biology*, 22(1):1–25, 2021. (Cited on page 82.)

[243] Romain Lopez, Adam Gayoso, and Nir Yosef. Enhancing scientific discoveries in molecular biology with deep generative models. *Molecular Systems Biology*, 16(9):e9198, 2020. (Cited on page 82.)

[244] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020. (Cited on page 82.)

[245] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. (Cited on page 82.)

[246] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big Bird: Transformers for Longer Sequences. In *NeurIPS*, 2020. (Cited on page 82.)

[247] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021. (Cited on page 82.)

[248] Michael Eisenstein et al. Artificial intelligence powers protein-folding predictions. *Nature*, 599(7886):706–708, 2021. (Cited on page 82.)

[249] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019. (Cited on page 82.)

[250] Britta AM Bouwman, Nicola Crosetto, and Magda Bienko. RNA gradients: Shapers of 3D genome architecture. *Current Opinion in Cell Biology*, 74:7–12, 2022. (Cited on page 82.)

[251] Irene Farabella, Marco Di Stefano, Paula Soler-Vila, Maria Marti-Marimon, and Marc A Marti-Renom. Three-dimensional genome organization via triplex-forming RNAs. *Nature Structural & Molecular Biology*, 28(11):945–954, 2021. (Cited on page 82.)

[252] Kevin Michael Creamer, Heather Jill Kolpa, and Jeanne Bentley Lawrence. Nascent RNA scaffolds contribute to chromosome territory architecture and counter chromatin compaction. *Molecular Cell*, 81(17):3509–3525, 2021. (Cited on page 82.)

[253] Stephanie C Hicks, F William Townes, Mingxiang Teng, and Rafael A Irizarry. Missing data and technical variability in single-cell RNA-sequencing experiments. *Biostatistics*, 19(4):562–578, 2018. (Cited on page 83.)

[254] David Lähnemann, Johannes Köster, Ewa Szczurek, Davis J McCarthy, Stephanie C Hicks, Mark D Robinson, Catalina A Vallejos, Kieran R Campbell, Niko Beerenwinkel, Ahmed Mahfouz, et al. Eleven grand challenges in single-cell data science. *Genome biology*, 21(1):1–35, 2020. (Cited on page 83.)

[255] Yukie Kashima, Yoshitaka Sakamoto, Keiya Kaneko, Masahide Seki, Yutaka Suzuki, and Ayako Suzuki. Single-cell sequencing techniques from individual to multiomics analyses. *Experimental & Molecular Medicine*, 52(9):1419–1427, 2020. (Cited on page 83.)

[256] Jeffrey M Perkel. Single-cell proteomics takes centre stage. *Nature*, 597(7877):580–582, 2021. (Cited on page 83.)

[257] Kristopher W Brannan, Isaac A Chaim, Ryan J Marina, Brian A Yee, Eric R Kofman, Daniel A Lorenz, Pratibha Jagannatha, Kevin D Dong, Assael A Madrigal, Jason G Underwood, et al. Robust single-cell discovery of RNA targets of RNA-binding proteins and ribosomes. *Nature Methods*, 18(5):507–519, 2021. (Cited on page 83.)

[258] Yves Van de Peer, Eshchar Mizrachi, and Kathleen Marchal. The evolutionary significance of polyploidy. *Nature Reviews Genetics*, 18(7):411–424, 2017. (Cited on page 83.)

[259] Rachel M Sherman and Steven L Salzberg. Pan-genomics in the human genome era. *Nature Reviews Genetics*, 21(4):243–254, 2020. (Cited on page 83.)

[260] Alice M Kaye and Wyeth W Wasserman. The genome atlas: navigating a new era of reference genomes. *Trends in Genetics*, 2021. (Cited on page 83.)

[261] Glennis A Logsdon, Mitchell R Vollger, and Evan E Eichler. Long-read human genome sequencing and its applications. *Nature Reviews Genetics*, 21(10):597–614, 2020. (Cited on page 83.)

[262] Wouter De Coster, Matthias H Weissensteiner, and Fritz J Sedlazeck. Towards population-scale long-read sequencing. *Nature Reviews Genetics*, pages 1–16, 2021. (Cited on page 83.)

[263] Jouni Sirén, Jean Monlong, Xian Chang, Adam M Novak, Jordan M Eizenga, Charles Markello, Jonas A Sibbesen, Glenn Hickey, Pi-Chuan Chang, Andrew Carroll, et al. Pangenomics enables genotyping of known structural variants in 5202 diverse genomes. *Science*, 374(6574):abg8871, 2021. (Cited on page 83.)

Index

Appendix - List of abbreviations

Table 1: List of abbreviations used in this thesis and their meanings.

Abbreviation	Meaning
3D	three-dimensional
A	adenine
ANN	artifical neural network
AS	alternative splicing
AUC	area under the curve (performance metric)
BOHB	Bayesian optimization and hyperband method
bp	base pair(s)
BPS	branch point sequence
C	cytosine
CDE	constitutive decay element
cDNA	complementary DNA
CDS	coding DNA sequence
CLIP	cross-linking and immunoprecipitation
CLIP-seq	CLIP followed by high-throughput sequencing
CNN	convolutional neural network
DL	deep learning
DNA	deoxyribonucleic acid
DNN	deep neural network
dsRNA	double-stranded RNA
ECDF	empirical cumulative density function
eCLIP	enhanced CLIP
ER	endoplasmatic reticulum
eRNA	enhancer RNA
G	guanine
GRU	gated recurrent unit (a type of RNN)
h	hours
HITS-CLIP	high-throughput sequencing of RNA isolated by CLIP
HPO	hyperparameter optimization
HTS	high-throughput sequencing
iCLIP	individual-nucleotide CLIP
IDR	intrinsically disordered region
IP	immunoprecipitation
lncRNA	long non-coding RNA
LSTM	long short-term memory (a type of RNN)

min	minutes
miRNA	microRNA
ML	machine learning
MLP	multilayer perceptron
mRNA	messenger RNA
ncRNA	non-coding RNA
NGS	next-generation sequencing
NMD	nonsense-mediated decay
nm	nanometer
NN	neural network
nt	nucleotide(s)
ORF	open reading frame
PAR-CLIP	photoactivatable-ribonucleoside-enhanced CLIP
PCR	polymerase chain reaction
Pol I	RNA Polymerase I
Pol II	RNA Polymerase II
Pol III	RNA Polymerase III
PPT	polypyrimidine tract
PTM	post-translational modification
PTM	post-translational modification
qPCR	real-time quantitative PCR
RBD	RNA-binding domain
RBP	RNA-binding protein
RIP-seq	RNA immunoprecipitation sequencing
RNA	ribonucleic acid
RNN	recurrent neural network
RNP	ribonucleoprotein
rRNA	ribosomal RNA
RT	reverse transcription
sec	seconds
SGD	stochastic gradient descent
snRNA	small nuclear RNA
SS	splice site
tRNA	transfer RNA
TSS	transcription start site
T	thymine
UMI	unique molecular identifier
UTR	untranslated region
U	uracil
UV	ultraviolet (radiation)

Appendix - Supplementary Material

[P2] MechRNA: prediction of lncRNA mechanisms from RNA-RNA and RNA-protein interactions

Supplementary material for publication:

- [P2] Alexander R. Gawronski, **Michael Uhl**, Yajia Zhang, Yen-Yi Lin, Yashar S. Niknafs, Varune R. Ramnarine, Rohit Malik, Felix Feng, Arul M. Chinnaiyan, Colin C. Collins, S. Cenk Sahinalp, and Rolf Backofen. **MechRNA: prediction of lncRNA mechanisms from RNA-RNA and RNA-protein interactions**. *Bioinformatics*, 2018.

Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: 2 April 2015
Applications Note

OXFORD

RNA Biology

Supplementary Methods and Discussion

Alexander R. Gawronski [1,*], Michael Uhl [3], S. Cenk Sahinalp [2,4,*], Rolf Backofen [3,*]

[1] Computing Science, Simon Fraser University, Burnaby, Canada
[2] Vancouver Prostate Centre, Vancouver, BC, Canada
[3] Institut für Informatik, University of Freiburg, Freiburg im Breisgau, Germany
[4] Department of Computer Science, Indiana University, Bloomington, USA

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

1 Methodological Details of IntaRNA2

The next stage is the prediction of RNA-RNA interactions using IntaRNA2 (Mann *et al.*, 2017) with the modifications outlined above. IntaRNA is a popular accessibility-based tool known for its highly competitive performance (Lai and Meyer, 2016). The hybridization calculation follows that of RNAHybrid (Rehmsmeier *et al.*, 2004) with a time and space complexity of $O(nm)$. The accessibility is calculated in $O(nL^2)$ using RNAplfold (Bernhart *et al.*, 2006), an algorithm that computes accessibility in a locally folded region of length L. Both energy contributions are calculated for every combination of intervals on both sequences requiring a time and space complexity of $O(n^2m^2)$. Using the same restriction on interaction length w as RNAup (Muckstein *et al.*, 2006), the time and space complexity is $O(nmw^2)$. By using sparsification (Figure 1), this complexity is further reduced to $O(nm)$ space and $O(n\bar{m})$ time where $\bar{m} = max(m, L^3)$.

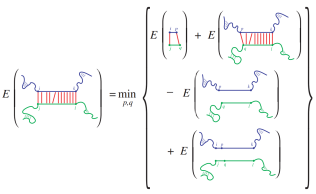


Fig. 1. Heuristic for reducing time complexity of IntaRNA (figure taken from (Busch et al., 2008)). The top energies are of the hybridization and the two bottom energies are for the accessibilities. The accessibilities are not additive so the contribution needs to be subtracted and then added back with the extended region.

2 P-value Computation for Predicted Interactions

2.1 RNA-RNA Interactions

In a previous work, RNA-RNA interaction energies were fitted to a generalized extreme value (GEV) distribution in order to compute interaction p-values (Wright *et al.*, 2014). From our recent experience we found that a gamma distribution fits the data better(data not shown), so it was used for all experiments. Regardless, we support a CopraRNA-style GEV approach through a user-specified parameter. We first compute a background gamma cumulative distribution function (CDF), which has two parameters: shape (α) and rate (β) (Equation 1- 3). The background values are obtained by assuming that a top percent (default 3%) are true interactions and the rest are background. The parameters of the function are estimated using maximum-likelihood fitting. This is done using the "fit" function in the python "stats" package from the scipy library (Jones *et al.*, 01). With these estimated α and β parameters, the p-values for each energy value (x) can be computed using the survival function ($1 - cdf(x)$).

$$F(x;\alpha,\beta) = \int_0^x f(u;\alpha,\beta)du \tag{1}$$

$$f(x;\alpha,\beta) = \frac{\beta^\alpha x^{\alpha-1}e^{-x\beta}}{\Gamma(\alpha)} \tag{2}$$

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1}e^{-x}dx \tag{3}$$

2.2 RNA-Protien Interactions

P-values for each peak score were calculated based on position-wise score data from 5000 randomly selected transcripts, using R's empirical cumulative distribution function (ECDF). The function returns the p-value of a given score based on the constructed ECDF and the ecdf() object can be stored on disk for subsequent recalculations (found together with models in Supplementary file 1). We chose this non-parametric approach since the scores did not show a clear unimodal distribution for most models,

Table 1. Used GraphProt models with model parameters, training set information and filter p-values. GraphProt model parameters are: epochs, lambda, R, D, bitsize, abstraction. PMID: data source pubmed ID, method: CLIP-seq protocol, filter_p: p-value used for filtering predicted sites, pos_tr: number of positive training sites, neg_tr: number of negative training sites

RBP	PMID	method	model_type	filter_p	epochs	lambda	R	D	bitsize	abstraction	pos_tr	neg_tr	ROC	APR
AGO1-4	20371350	PARCLIP	structure	0.02376089	20	0.000001	4	1	14	3	36802	31310	0.85584	0.86766
ELAVL1	21723170	PARCLIP	sequence	0.001640548	10	0.001	3	5	14	-	7747	7750	0.92887	0.94365
EWSR1	20371350	PARCLIP	sequence	0.005115178	50	0.001	1	2	14	-	16292	14720	0.94345	0.9496
FMR1	27018577	eCLIP	structure	0.04819012	40	0.0001	4	5	14	3	2587	2587	0.88109	0.87115
FUS	22081015	PARCLIP	sequence	0.003591709	40	0.0001	1	1	14	-	34581	31480	0.96988	0.97034
HNRNPC	27018577	eCLIP	sequence	0.0006383588	50	0.001	3	6	14	-	2511	2511	0.95636	0.95178
HNRNPK	27018577	eCLIP	sequence	0.0011904	10	0.001	2	1	14	-	2674	2673	0.9823	0.98059
IGF2BP1-3	20371350	PARCLIP	structure	0.01519445	50	0.0001	4	0	14	3	8539	6838	0.88223	0.89533
KHDRBS1	27018577	eCLIP	structure	0.003200621	40	0.001	3	2	14	3	2552	2552	0.9234	0.92122
MOV10	22844102	PARCLIP	sequence	0.02331425	20	0.001	4	2	14	-	13793	12987	0.79824	0.7715
PUM2	27018577	PARCLIP	sequence	0.002040983	40	0.001	4	4	14	-	9116	8227	0.94144	0.95158
QKI	27018577	eCLIP	structure	0.0006862552	40	0.000001	4	2	14	3	2650	2650	0.94722	0.95187
SND1	27018577	eCLIP	structure	0.04999487	50	0.0001	3	4	14	3	2413	2413	0.89622	0.88589
TAF15	22081015	PARCLIP	sequence	0.003209317	50	0.001	3	2	14	-	7298	6606	0.96794	0.964
TARDBP	27018577	eCLIP	sequence	0.0003341065	30	0.001	4	5	14	-	2752	2752	0.98524	0.98712
TIA1	27018577	eCLIP	sequence	0.009658455	30	0.001	2	5	14	-	3073	3073	0.84148	0.86061
TNRC6A	27018577	eCLIP	structure	0.04627634	50	0.001	3	0	14	3	2653	2653	0.83569	0.85761

which prevented the use of conventional fitting procedures for unimodal distributions. For each model, we then calculated the top position-wise score of each positive training site to construct a second ECDF. To get a threshold for filtering the peak score p-values, the score at 50 % of the distribution was taken and inserted into the first ECDF to get its p-value. This way we obtain an individual p-value threshold for each RBP model, allowing us to select binding sites with scores comparable to the scores found in the respective positive training sites. The obtained filter p-values for each model can be found in Supplementary Table 1.

3 Challenges and Limitations

Predicting combined interactions between lncRNAs, RBPs and target RNAs on a transcriptome-wide scale is an inherently difficult task, due to several reasons: firstly, the limited number of known lncRNA mechanism cases makes it difficult to tune the model. Specifically, the selection of various parameters in terms of distances between interactions and various cutoffs becomes nearly *ad hoc*. Moreover, it is unknown to what extent the studied cases occur in the cell or whether they are typical representatives of a certain class of interactions. Secondly, even with the careful filtering applied in this work, RNA-RNA and RNA-protein predictions are fairly non-specific. With thousands of predicted targets, it is likely that many are false positives. Given that only the most significant interaction combinations are included, it is difficult to determine which are true predictions since they are all plausible. Despite these difficulties, the presented work provides a solid starting point for further experimental investigation.

One way to improve the current approach would be the development of more realistic interaction models. As for the RBP-target prediction, information on RBP affinities for a range of target RNAs as well as the relative importance of target sequence, structure and context should help to design more accurate models. So far, detailed affinity distributions have only been reported for the *E. coli* C6 protein, utilizing the high-throughput sequencing kinetics (HiTS-KIN) protocol (Lin *et al.*, 2016). Lately, a more simple affinity approach was combined with estimating the sequential and structural binding properties of 78 human RBPs, using an RNA Bind-n-Seq variant with 5 different protein concentrations (Dominguez *et al.*, 2017). In order to improve prediction specificity, it is also possible to use CLIP data to cluster RBPs with common binding sites and to learn properties from these sites, as shown by Li et al. (Li *et al.*, 2017). As for the lncRNA-target prediction, integrating protein binding information directly into the RNA-RNA interaction calculation might lead to the prediction of more realistic hybrids. Moreover, incorporating RNA structure probing data of the involved RNAs, e.g. determined by selective 2-hydroxyl acylation and profiling (SHAPE), could improve the hybrid prediction. As the number of studied lncRNA mechanisms gradually increases, machine learning approaches could further help to improve model performance by learning optimal parameter combinations from the data.

Another more immediate extension of this work would be the incorporation of additional data, such as new RBP predictions or miRNA interaction information. It is conceivable to assume that lncRNAs might block or sequester miRNAs, just as they do RBPs. Inclusion of miRNA target sites would therefore broaden the scope of mechanisms MechRNA can predict. The modular nature of MechRNA makes such extensions possible, which might open exciting new avenues for lncRNA research.

References

Bernhart, S. H., Hofacker, I. L., and Stadler, P. F. (2006). Local RNA base pairing probabilities in large sequences. *Bioinformatics*, **22**(5), 614–615.

Busch, A., Richter, A. S., and Backofen, R. (2008). IntaRNA: efficient prediction of bacterial sRNA targets incorporating target site accessibility and seed regions. *Bioinformatics*, **24**(24), 2849–2856.

Dominguez, D., *et al.* (2017). Sequence, structure and context preferences of human rna binding proteins. *bioRxiv*, page 201996.

Jones, E., *et al.* (2001–). SciPy: Open source scientific tools for Python. [Online; accessed <today>].

Lai, D. and Meyer, I. M. (2016). A comprehensive comparison of general rna–rna interaction prediction methods. *Nucleic acids research*, **44**(7), e61–e61.

Li, Y. E., *et al.* (2017). Identification of high-confidence rna regulatory elements by combinatorial classification of rna–protein binding sites. *Genome biology*, **18**(1), 169.

Lin, H.-C., *et al.* (2016). Analysis of the rna binding specificity landscape of c5 protein reveals structure and sequence preferences that direct rnase p specificity. *Cell chemical biology*, **23**(10), 1271–1281.

Mann, M., Wright, P. R., and Backofen, R. (2017). IntaRNA 2.0: enhanced and customizable prediction of RNA-RNA interactions. *Nucleic Acids Res.*

Muckstein, U., *et al.* (2006). Thermodynamics of RNA-RNA binding. *Bioinformatics*, **22**(10), 1177–1182.

Rehmsmeier, M., *et al.* (2004). Fast and effective prediction of microRNA/target duplexes. *RNA*, **10**(10), 1507–1517.

Wright, P. R., *et al.* (2014). CopraRNA and IntaRNA: predicting small RNA targets, networks and interaction domains. *Nucleic Acids Res.*, **42**(Web Server issue), W119–123.

Table 2. Selected LncRNAs for MechRNA analysis. The lncRNAs vary in terms of what is known about their mechanisms, allowing MechRNA to be tested with various amounts of a priori data. PCAT1 has a question mark indicating that competitive binding is the hypothesis not been validated yet.

	Protein Binding		**RNA-RNA Interaction**				
TP53 Transcript	**HuR S**	**HuR E**	**TP53 S**	**TP53 E**	**7SL S**	**7SL E**	**FE**
ENST00000618944	1950	1971	1980	2022	256	298	-51.563
ENST00000504937	1817	1838	1847	1889	256	298	-51.563
ENST00000445888	2071	2092	2101	2143	256	298	-51.563
ENST00000420246	2201	2222	2231	2273	256	298	-51.563
ENST00000269305	**2125**	**2146**	**2155**	**2197**	**256**	**298**	**-51.563**
ENST00000610292	2185	2206	2215	2257	256	298	-51.563
ENST00000620739	2125	2146	2155	2197	256	298	-51.563
ENST00000455263	2128	2149	2158	2200	256	298	-51.563
ENST00000610623	1877	1898	1907	1949	256	298	-51.563
ENST00000504290	1877	1898	1907	1949	256	298	-51.563
ENST00000610538	2128	2149	2158	2200	256	298	-51.563
ENST00000619485	2071	2092	2101	2143	256	298	-51.563
ENST00000510385	1950	1971	1980	2022	256	298	-51.563
ENST00000622645	2201	2222	2231	2273	256	298	-51.563
ENST00000619186	1817	1838	1847	1889	256	298	-51.563
ENST00000617185	2270	2291	2300	2342	256	298	-51.563

[P3] RNAProt: an efficient and feature-rich RNA binding protein binding site predictor

Supplementary material for publication:

- [P3] **Michael Uhl**, Van Dinh Tran, Florian Heyl, and Rolf Backofen. **RNAProt: an efficient and feature-rich RNA binding protein binding site predictor**. *GigaScience*, 2021.

RNAProt: An efficient and feature-rich RNA binding protein binding site predictor Supplementary Material

Michael Uhl, Van Dinh Tran, Florian Heyl, and Rolf Backofen

April 2, 2021

Supplementary methods

Dataset construction

For the tool comparison we constructed two different benchmark sets: the first one includes 23 different PAR-CLIP, iCLIP, and HITS-CLIP datasets (20 different RBPs) extracted from the original GraphProt publication [3]. The second one consists of 30 eCLIP datasets (30 different RBPs) extracted from ENCODE.

For the first set, CLIP-Seq datasets used for benchmarking GraphProt were obtained from here. Sets for hyperparameter optimization and training were merged, hg19 genomic regions (corresponding to uppercase sequence parts, also termed viewpoint regions) were extracted from the FASTA headers, and lifted over to hg38, using the UCSC's liftOver command line tool. Viewpoint regions were filtered by a maximum length of 60 nt and extended to a new constant length of 81 nt, for CNN method compatibility. For each of the 24 datasets, we randomly selected a maximum of 5,000 positive and negative sites each, in order to keep model training times for DeepRAM [5] and DeepCLIP [2] reasonable. Note that we removed the PTB dataset from the benchmark set, as the sampled 10,000 sites showed to be non-informative (resulting AUCs of 50% for all methods). This led us to the final benchmark set size of 23 datasets.

For the eCLIP set we extracted data out of two cell lines (HepG2, K562) from ENCODE [4] (November 2018 release). We directly used the genomic binding regions (genome assembly GRCh38) identified by ENCODE's in-house peak caller CLIPper, which are available in BED format for each RBP and each replicate, often for both cell lines (thus 4 replicate BED files per RBP). Binding sites were further filtered by their log2 fold change (FC) to obtain $\sim$ 6,000 to 10,000 binding regions for each replicate. We next removed sites with length $>$ 0.75 percentile length and selected for each RBP the replicate set that contained the most regions, centered the sites, and extended them to make all sites of equal length. We chose a binding site length of 81 nt (40 nt extension up- and downstream of center position) and selected 30 RBP sets (some based on previous knowledge about binding preferences for comparison, the remaining ones random). To generate the eCLIP negative sets, RNAProt randomly selected sites based on two criteria: 1) their location on genes covered by eCLIP peak regions and 2) no overlap with

any eCLIP peak regions from the experiment. The same number of random negative and positive instances was used throughout the benchmarks.

Cross validation comparison

All three methods (GraphProt, DeepCLIP, RNAProt) were run using default parameters. For DeepCLIP we set patience (early stopping) to 20 and the maximum number of epochs to 200, since this setting was used the most in the DeepCLIP paper. For RNAProt we used a patience of 30 and maximum number of epochs to 200, which also is the tool default. An example call for DeepCLIP thus looked like this:

```
./DeepCLIP.py --runmode cv -n runtime_test_model
-P runtime_test_model_pred_fct
--predict_PFM_file pfms.json --sequences positives.5000.fa
--background_sequences negatives.5000.fa --num_epochs 200
--early_stopping 20 > runtime_test_model.log.txt
```

Likewise, an RNAProt call looks the following, with the dataset generated by `rnaprot gt` stored in `data_gt_out` used as input for training:

```
rnaprot train --in data_gt_out --out data_cv_train_out
--verbose-train --cv --only-seq
```

Hold-out comparison

For the hold-out comparison, DeepRAM was executed with its highest performing setting (ECBLSTM). This was achieved by calling DeepRAM with the following parameters (replace "data" with specific dataset ID):

```
python deepRAM.py --train_data data.train.gz --test_data data.test.gz
--data_type RNA --train True --evaluate_performance True
--model_path data.model.pkl --out_file data.predictions.txt --Conv True
--conv_layers 1 --Embedding True --RNN True --RNN_type BiLSTM --kmer_len 3
--stride 1 --word2vec_train True
--word2vec_model data.word2vec_train.model
```

We used 90% of a dataset for training, and the remaining 10% for testing. The same split was used for DeepRAM and RNAProt. For RNAProt, we used its option `--test-ids` to provide the same test IDs as used for DeepRAM for model training:

```
rnaprot train --in data_gt_out --out data_train_out --only-seq
--verbose-train --test-ids hold_out/data.test_ids
--val-size 0.2 --patience 50
```

Roquin CDE dataset preparation and prediction

To further assess the impact of adding structure information on RNAProt's predictive performance, we downloaded a dataset consisting of genomic regions containing potential human CDEs (constitutive decay elements) identified by [1] (Supplementary Table 6, table

"all"). A CDE consists of a short single hairpin with a tri-nucleotide loop that is preferably bound by the RBP Roquin. We then filtered the CDE containing sites by a minimum folding probability of 0.15, centered and extended them to 81 nt, and ran `rnaprot gt` with RNAplfold settings `--plfold-l 50`, `--plfold-w 70`, and `--plfold-u 3` to focus more on local hairpin structures. Finally we calculated the average model AUC with 10-fold cross validation, for both the sequence-only set and the sequence set with added structure information. We chose the most basic GRU model architecture (non-bidirectional GRU with one GRU layer, RNAProt default setting), corresponding to the following set parameters for training (`rnaprot train`):

```
--str-mode 1 --patience 30 --epochs 300 --batch-size 50
--lr 0.001 --weight-decay 0.0005 --n-rnn-layers 1 --n-hidden-dim 32
--dr 0.5 --model-type 1 --n-fc-layers 1
```

Note that we increased the maximum number of epochs from 200 to 300, which can help with smaller datasets like the described CDE set. We also ran GraphProt on the same set using the Galaxy version, to train a sequence and a structure model. For GraphProt we used the Galaxy default parameters.

For the window prediction, we used the UCP3 gene transcript (ENST00000314032.9). We trained a sequence model and a structure model, after excluding the CDE site on the UCP3 gene from the training set. For reporting peak regions, we used threshold levels `--thr 2` for the sequence and `--thr 1` for the structure model.

Runtime comparison

For the runtime comparison we took 5,000 positive and 5,000 negative training sequences, all with a length of 81 nt. Each tool was run three times using only the sequence information in train mode with default parameters. For GraphProt (sequence model mode), these are R: 1, D: 4, bitsize: 14, epochs: 10, and lambda: 0.001. For DeepCLIP and RNAProt, we used the default parameters together with a patience (early stopping) of 20 and a maximum number of 200 training epochs. For DeepCLIP, an example single model training call looked like this:

```
./DeepCLIP.py --runmode train -n TEST_MODEL
-P TEST_MODEL_PREDICTION_FUNCTION
--sequences positives.5000.fa --background_sequences negatives.5000.fa
--num_epochs 200 --early_stopping 20
```

References

[1] Johannes Braun, Sandra Fischer, Zhenjiang Z Xu, Hongying Sun, Dalia H Ghoneim, Anna T Gimbel, Uwe Plessmann, Henning Urlaub, David H Mathews, and Julia E Weigand. Identification of new high affinity targets for roquin based on structural conservation. *Nucleic acids research*, 46(22):12109–12125, 2018.

[2] Alexander Gulliver Bjørnholt Grønning, Thomas Koed Doktor, Simon Jonas Larsen, Ulrika Simone Spangsberg Petersen, Lise Lolle Holm, Gitte Hoffmann Bruun, Michael Birkerod Hansen, Anne-Mette Hartung, Jan Baumbach, and Brage Storstein Andresen. Deepclip: predicting the effect of mutations on protein–rna binding with deep learning. *Nucleic acids research*, 48(13):7099–7118, 2020.

[3] Daniel Maticzka, Sita J Lange, Fabrizio Costa, and Rolf Backofen. Graphprot: modeling binding preferences of rna-binding proteins. *Genome biology*, 15(1):R17, 2014.

[4] Cricket A Sloan, Esther T Chan, Jean M Davidson, Venkat S Malladi, J Seth Strattan, Benjamin C Hitz, Idan Gabdank, Aditi K Narayanan, Marcus Ho, Brian T Lee, et al. Encode data at the encode portal. *Nucleic acids research*, 44(D1):D726–D732, 2015.

[5] Ameni Trabelsi, Mohamed Chaabane, and Asa Ben-Hur. Comprehensive evaluation of deep learning architectures for prediction of dna/rna sequence binding specificities. *Bioinformatics*, 35(14):i269–i277, 2019.

[P3] Table S1: 10-fold cross validation results for GraphProt, DeepCLIP, RNAProt, and RNAProt with additional features. Results for the first benchmark set, containing 23 CLIP-seq datasets from 20 different RBPs and various CLIP-seq protocols.

	GraphProt		DeepCLIP		RNAProt		RNAProt structure		RNAProt exon-intron		RNAProt phastCons		RNAProt phyloP		RNAProt eia+con	
Dataset_ID	**AUC**	**STDEV**	**AUC**	**STDEV**	**AUC**	**STDEV**	**AUC**	**STDEV**	**AUC**	**STDEV**	**AUC**	**STDEV**	**AUC**	**STDEV**	**AUC**	**STDEV**
ALKBH5_Baltz2012	70.10%	2.49%	70.38%	2.73%	62.17%	2.53%	64.31%	3.03%	83.03%	2.54%	80.53%	2.44%	81.83%	2.82%	86.43%	2.29%
C17ORF85_Baltz2012	80.37%	3.03%	80.75%	2.21%	75.48%	3.02%	74.81%	4.36%	89.12%	1.79%	90.04%	0.88%	90.58%	1.07%	91.06%	1.04%
C22ORF28_Baltz2012	74.12%	2.01%	74.54%	1.61%	74.71%	1.32%	73.51%	2.08%	88.12%	1.14%	87.10%	0.91%	88.52%	1.09%	90.99%	0.98%
CAPRIN1_Baltz2012	68.52%	1.36%	68.88%	1.54%	73.77%	2.62%	74.29%	1.66%	90.55%	0.67%	86.18%	0.60%	86.73%	0.80%	91.97%	0.95%
CLIPSEQ_AGO2	72.10%	1.26%	72.21%	1.57%	78.09%	1.60%	78.06%	1.85%	95.69%	0.59%	90.00%	0.97%	90.79%	1.07%	96.44%	0.60%
CLIPSEQ_ELAVL1	91.49%	0.65%	94.46%	1.72%	97.48%	0.45%	97.35%	0.45%	99.52%	0.14%	98.31%	0.36%	98.26%	0.38%	99.56%	0.15%
CLIPSEQ_SFRS1	85.36%	1.00%	86.66%	1.01%	89.23%	0.67%	89.10%	0.80%	92.34%	0.67%	91.90%	0.59%	92.27%	0.43%	92.99%	0.69%
ICLIP_HNRNPC	92.48%	0.48%	93.78%	1.35%	97.32%	0.37%	97.30%	0.44%	97.33%	0.53%	97.28%	0.41%	97.36%	0.39%	97.21%	0.44%
ICLIP_TDP43	84.89%	1.07%	85.60%	1.00%	89.63%	0.67%	89.67%	0.61%	89.65%	0.49%	89.44%	0.81%	89.61%	0.85%	89.21%	1.01%
ICLIP_TIA1	82.45%	1.58%	85.39%	2.26%	91.61%	0.74%	91.43%	0.66%	94.64%	0.45%	93.93%	0.54%	93.93%	0.56%	94.93%	0.45%
ICLIP_TIAL1	80.29%	1.67%	83.13%	2.02%	90.30%	0.90%	90.23%	1.04%	92.57%	1.00%	91.83%	1.16%	92.02%	0.71%	93.02%	0.89%
PARCLIP_AGO1234	70.67%	1.91%	71.08%	1.58%	80.44%	0.97%	80.28%	1.57%	91.92%	0.82%	88.46%	1.19%	87.94%	1.22%	92.62%	0.78%
PARCLIP_ELAVL1A	87.66%	1.03%	96.11%	1.43%	97.30%	0.55%	97.18%	0.46%	99.47%	0.10%	98.07%	0.21%	98.07%	0.30%	99.47%	0.09%
PARCLIP_ELAVL1	91.41%	1.14%	88.26%	2.07%	93.68%	0.72%	93.62%	0.90%	96.45%	0.65%	94.63%	0.93%	94.70%	0.86%	96.50%	0.75%
PARCLIP_EWSR1	82.33%	1.64%	88.64%	1.31%	94.22%	0.66%	94.18%	0.51%	96.72%	0.57%	95.88%	0.68%	95.90%	0.52%	96.75%	0.50%
PARCLIP_FUS	84.19%	0.88%	90.20%	3.10%	96.19%	0.70%	96.04%	0.52%	96.58%	0.55%	96.24%	0.55%	96.32%	0.69%	96.43%	0.63%
PARCLIP_HUR	93.68%	0.73%	95.64%	1.07%	98.90%	0.49%	98.84%	0.43%	99.02%	0.42%	98.93%	0.52%	98.95%	0.52%	99.02%	0.47%
PARCLIP_IGF2BP123	77.85%	0.91%	81.22%	2.48%	87.33%	0.68%	87.12%	0.95%	96.97%	0.69%	92.52%	0.73%	92.63%	0.52%	97.10%	0.63%
PARCLIP_MOV10	74.69%	1.52%	75.06%	1.88%	80.41%	1.60%	80.31%	1.35%	94.36%	0.72%	84.15%	1.29%	85.73%	1.49%	94.78%	0.57%
PARCLIP_PUM2	88.47%	1.27%	91.31%	1.10%	93.89%	0.83%	93.83%	0.68%	97.35%	0.42%	95.44%	0.53%	95.53%	0.61%	97.34%	0.52%
PARCLIP_QKI	92.58%	0.76%	95.65%	0.88%	96.73%	0.62%	96.46%	0.60%	98.30%	0.51%	97.55%	0.49%	97.64%	0.43%	98.27%	0.40%
PARCLIP_TAF15	83.90%	0.82%	94.05%	3.01%	97.70%	0.53%	97.87%	0.45%	98.14%	0.43%	98.04%	0.49%	97.96%	0.47%	98.24%	0.29%
ZC3H7B_Baltz2012	69.73%	1.22%	69.59%	1.13%	70.37%	1.40%	69.80%	1.87%	84.66%	1.12%	81.00%	1.14%	80.67%	1.40%	85.89%	1.11%
Mean	81.71%		84.03%		87.26%		87.20%		94.02%		92.06%		92.34%		94.62%	

[P3] Table S1 legend:

AUC	Area under the receiver operating characteristic curve
STDEV	Standard deviation
RNAProt structure	RNAProt with secondary structure feature
RNAProt exon-intron	RNAProt with exon-intron annotations feature
RNAProt phastCons	RNAProt with phastCons convervation scores feature
RNAProt phyloP	RNAProt with phyloP convervation scores feature
RNAProt eia+con	RNAProt with exon-intron annotations + conservation scores (phastCons + phyloP) features

[P3] Table S2: 10-fold cross validation results for GraphProt, DeepCLIP, RNAProt, and RNAProt with additional features. Results for the second benchmark set, containing 30 eCLIP datasets from 30 different RBPs.

	GraphProt		DeepCLIP		RNAProt		RNAProt structure		RNAProt exon-intron		RNAProt phastCons		RNAProt phyloP		RNAProt eia+con	
Dataset_ID	**AUC**	**STDEV**	**AUC**	**STDEV**	**AUC**	**STDEV**	**AUC**	**STDEV**	**AUC**	**STDEV**	**AUC**	**STDEV**	**AUC**	**STDEV**	**AUC**	**STDEV**
AGGF1_HepG2	80.16%	1.19%	81.56%	2.00%	86.06%	1.45%	86.33%	1.23%	92.92%	1.02%	92.23%	0.93%	92.99%	0.71%	93.53%	0.79%
BUD13_K562	78.67%	0.94%	82.95%	3.73%	85.91%	0.98%	86.03%	0.99%	94.75%	0.44%	94.26%	0.69%	94.87%	0.64%	95.76%	0.45%
CSTF2T_HepG2	92.27%	0.42%	92.81%	1.25%	95.59%	0.48%	95.62%	0.45%	95.81%	0.52%	95.72%	0.52%	96.11%	0.50%	96.20%	0.49%
DDX55_HepG2	74.53%	0.59%	76.58%	1.31%	78.68%	0.93%	78.52%	1.18%	92.35%	0.54%	88.71%	0.79%	89.42%	0.73%	92.88%	0.52%
EFTUD2_HepG2	86.96%	0.81%	86.96%	0.99%	89.33%	0.81%	89.48%	0.81%	91.76%	0.77%	91.92%	0.76%	92.33%	0.73%	93.08%	0.61%
EWSR1_K562	85.52%	1.00%	87.20%	2.40%	90.13%	1.14%	90.35%	1.03%	90.49%	1.12%	90.58%	1.07%	91.28%	1.10%	91.40%	1.10%
FASTKD2_HepG2	80.78%	0.84%	84.37%	1.66%	86.75%	0.90%	86.06%	0.62%	95.69%	0.49%	95.34%	0.52%	95.96%	0.41%	96.60%	0.39%
FMR1_K562	82.92%	1.32%	89.14%	1.12%	90.24%	0.86%	90.21%	0.91%	97.05%	0.49%	96.86%	0.44%	97.27%	0.27%	97.59%	0.35%
FUS_HepG2	80.47%	1.03%	85.39%	3.57%	87.59%	0.81%	87.89%	0.74%	87.91%	0.74%	87.99%	0.90%	88.26%	0.69%	88.61%	0.74%
FXR2_K562	85.97%	1.05%	89.92%	1.26%	91.00%	1.02%	91.08%	0.89%	98.13%	0.28%	97.88%	0.32%	98.21%	0.32%	98.89%	0.26%
HNRNPA1_K562	88.15%	0.90%	92.76%	0.51%	93.67%	0.56%	93.78%	0.35%	93.86%	0.52%	93.59%	0.53%	93.71%	0.48%	93.82%	0.44%
HNRNPC_HepG2	93.70%	0.35%	96.89%	0.50%	97.39%	0.38%	97.67%	0.30%	97.35%	0.30%	97.32%	0.32%	97.35%	0.34%	97.28%	0.36%
HNRNPK_HepG2	96.39%	0.30%	97.14%	0.69%	98.38%	0.31%	98.45%	0.41%	98.39%	0.37%	98.38%	0.36%	98.43%	0.31%	98.35%	0.40%
IGF2BP1_HepG2	77.91%	1.98%	84.81%	2.56%	87.58%	0.69%	87.96%	1.12%	97.73%	0.55%	93.75%	0.73%	94.31%	0.73%	97.79%	0.51%
KHDRBS1_K562	87.08%	1.22%	90.39%	1.51%	91.65%	0.95%	91.82%	0.74%	92.21%	0.90%	91.73%	0.87%	91.90%	0.73%	92.14%	0.85%
LIN28B_K562	77.76%	1.69%	79.96%	2.12%	83.51%	1.49%	85.37%	0.94%	94.87%	0.67%	92.52%	0.74%	93.15%	0.55%	95.28%	0.39%
PCBP2_HepG2	95.87%	0.58%	97.16%	0.42%	97.88%	0.40%	97.95%	0.25%	97.96%	0.34%	97.88%	0.33%	98.01%	0.33%	98.03%	0.37%
PTBP1_HepG2	93.16%	0.56%	94.97%	0.56%	95.76%	0.80%	95.85%	0.61%	95.91%	0.71%	95.86%	0.72%	95.93%	0.70%	95.89%	0.73%
PUM2_K562	66.58%	1.15%	69.48%	1.37%	73.15%	1.52%	72.23%	1.15%	80.22%	0.95%	76.92%	0.71%	77.34%	0.76%	80.17%	1.10%
QKI_HepG2	83.18%	2.13%	88.74%	1.65%	91.18%	0.65%	91.16%	0.56%	91.13%	0.74%	91.22%	0.78%	91.07%	0.62%	91.06%	0.75%
RBFOX2_K562	76.86%	1.18%	79.22%	1.95%	84.11%	0.97%	84.39%	1.18%	86.25%	0.89%	85.61%	0.83%	86.51%	0.74%	86.85%	0.81%
SF3B4_K562	77.51%	1.41%	81.82%	2.21%	85.80%	0.94%	85.43%	1.23%	94.13%	0.77%	94.40%	0.32%	95.26%	0.47%	96.02%	0.56%
SFPQ_HepG2	78.03%	1.08%	78.33%	1.54%	82.35%	1.12%	82.77%	0.76%	82.67%	1.00%	82.52%	0.96%	83.90%	1.08%	84.39%	0.96%
SMNDC1_K562	85.34%	0.70%	85.71%	0.75%	88.91%	0.64%	89.22%	0.71%	91.24%	0.67%	90.87%	0.65%	91.68%	0.72%	92.28%	0.62%
SRSF1_HepG2	92.97%	0.83%	95.59%	0.62%	96.38%	0.35%	96.19%	0.28%	98.27%	0.22%	98.05%	0.20%	98.17%	0.19%	98.53%	0.27%
TAF15_HepG2	88.76%	0.47%	89.44%	1.19%	92.14%	0.74%	92.52%	0.60%	92.17%	0.64%	92.33%	0.62%	92.41%	0.58%	92.33%	0.53%
TARDBP_K562	97.40%	0.41%	97.95%	0.55%	98.35%	0.37%	98.36%	0.25%	98.66%	0.29%	98.50%	0.18%	98.53%	0.30%	98.57%	0.29%
TIA1_K562	79.92%	1.13%	89.06%	1.05%	89.41%	0.73%	89.64%	0.63%	92.32%	0.53%	91.10%	0.70%	91.54%	0.62%	92.97%	0.62%
U2AF2_HepG2	80.77%	1.50%	93.71%	0.82%	93.73%	0.42%	93.58%	0.51%	94.69%	0.56%	94.54%	0.65%	94.83%	0.58%	95.17%	0.48%
UPF1_K562	68.73%	1.18%	70.09%	1.73%	76.41%	0.88%	76.37%	1.01%	96.64%	0.38%	81.19%	1.46%	83.20%	1.12%	96.91%	0.30%
Mean	83.81%		87.00%		89.30%		89.41%		93.45%		92.32%		92.80%		93.94%	

[P3] Table S2 legend:

AUC	Area under the receiver operating characteristic curve
STDEV	Standard deviation
RNAProt structure	RNAProt with secondary structure feature
RNAProt exon-intron	RNAProt with exon-intron annotations feature
RNAProt phastCons	RNAProt with phastCons convervation scores feature
RNAProt phyloP	RNAProt with phyloP convervation scores feature
RNAProt eia+con	RNAProt with exon-intron annotations + conservation scores (phastCons + phyloP) features

[P3] Table S3: 10-fold cross validation (CV) single fold AUC results for GraphProt, DeepCLIP, and RNAProt. Benchmark Set 1: Set from Table S1 (23 datasets). Benchmark Set 2: Set from Table S2 (30 datasets).

Benchmark Set	Dataset ID	Method	CV Fold	Fold AUC
1	ALKBH5_Baltz2012	DeepCLIP	1	0.695965335169881
1	ALKBH5_Baltz2012	DeepCLIP	2	0.685663452708907
1	ALKBH5_Baltz2012	DeepCLIP	3	0.70830463728191
1	ALKBH5_Baltz2012	DeepCLIP	4	0.726842286501377
1	ALKBH5_Baltz2012	DeepCLIP	5	0.702823691460055
1	ALKBH5_Baltz2012	DeepCLIP	6	0.745781680440771
1	ALKBH5_Baltz2012	DeepCLIP	7	0.69751492194674
1	ALKBH5_Baltz2012	DeepCLIP	8	0.721705693296602
1	ALKBH5_Baltz2012	DeepCLIP	9	0.709653351698806
1	ALKBH5_Baltz2012	DeepCLIP	10	0.643910697887971
1	C17ORF85_Baltz2012	DeepCLIP	1	0.813108254235252
1	C17ORF85_Baltz2012	DeepCLIP	2	0.774720653610477
1	C17ORF85_Baltz2012	DeepCLIP	3	0.781040490207858
1	C17ORF85_Baltz2012	DeepCLIP	4	0.787372341703713
1	C17ORF85_Baltz2012	DeepCLIP	5	0.840105731106572
1	C17ORF85_Baltz2012	DeepCLIP	6	0.807365132764628
1	C17ORF85_Baltz2012	DeepCLIP	7	0.809671993271657
1	C17ORF85_Baltz2012	DeepCLIP	8	0.827802475069086
1	C17ORF85_Baltz2012	DeepCLIP	9	0.801069325964196
1	C17ORF85_Baltz2012	DeepCLIP	10	0.833089030397693
1	C22ORF28_Baltz2012	DeepCLIP	1	0.729921487603306
1	C22ORF28_Baltz2012	DeepCLIP	2	0.774834710743802
1	C22ORF28_Baltz2012	DeepCLIP	3	0.730621900826446
1	C22ORF28_Baltz2012	DeepCLIP	4	0.763669421487603
1	C22ORF28_Baltz2012	DeepCLIP	5	0.727047520661157
1	C22ORF28_Baltz2012	DeepCLIP	6	0.756719008264463
1	C22ORF28_Baltz2012	DeepCLIP	7	0.738188016528926
1	C22ORF28_Baltz2012	DeepCLIP	8	0.735276859504132
1	C22ORF28_Baltz2012	DeepCLIP	9	0.752675619834711
1	C22ORF28_Baltz2012	DeepCLIP	10	0.745107438016529
1	CAPRIN1_Baltz2012	DeepCLIP	1	0.677022727272727
1	CAPRIN1_Baltz2012	DeepCLIP	2	0.70326652892562
1	CAPRIN1_Baltz2012	DeepCLIP	3	0.685545454545455
1	CAPRIN1_Baltz2012	DeepCLIP	4	0.694409090909091
1	CAPRIN1_Baltz2012	DeepCLIP	5	0.657012396694215
1	CAPRIN1_Baltz2012	DeepCLIP	6	0.689204545454545
1	CAPRIN1_Baltz2012	DeepCLIP	7	0.708123966942149
1	CAPRIN1_Baltz2012	DeepCLIP	8	0.705904958677686
1	CAPRIN1_Baltz2012	DeepCLIP	9	0.685871900826446
1	CAPRIN1_Baltz2012	DeepCLIP	10	0.681927685950413
1	CLIPSEQ_AGO2	DeepCLIP	1	0.695427685950413
1	CLIPSEQ_AGO2	DeepCLIP	2	0.730280991735537
1	CLIPSEQ_AGO2	DeepCLIP	3	0.725568181818182
1	CLIPSEQ_AGO2	DeepCLIP	4	0.721710743801653
1	CLIPSEQ_AGO2	DeepCLIP	5	0.735088842975207
1	CLIPSEQ_AGO2	DeepCLIP	6	0.726270661157025
1	CLIPSEQ_AGO2	DeepCLIP	7	0.694760330578512
1	CLIPSEQ_AGO2	DeepCLIP	8	0.744345041322314
1	CLIPSEQ_AGO2	DeepCLIP	9	0.723876033057851
1	CLIPSEQ_AGO2	DeepCLIP	10	0.723545454545454
1	CLIPSEQ_ELAVL1	DeepCLIP	1	0.923099173553719
1	CLIPSEQ_ELAVL1	DeepCLIP	2	0.964935950413223
1	CLIPSEQ_ELAVL1	DeepCLIP	3	0.94870041322314
1	CLIPSEQ_ELAVL1	DeepCLIP	4	0.970613636363636
1	CLIPSEQ_ELAVL1	DeepCLIP	5	0.949824380165289
1	CLIPSEQ_ELAVL1	DeepCLIP	6	0.934309917355372
1	CLIPSEQ_ELAVL1	DeepCLIP	7	0.926909090909091
1	CLIPSEQ_ELAVL1	DeepCLIP	8	0.950092975206612
1	CLIPSEQ_ELAVL1	DeepCLIP	9	0.922231404958678
1	CLIPSEQ_ELAVL1	DeepCLIP	10	0.955262396694215
1	CLIPSEQ_SFRS1	DeepCLIP	1	0.862460743801653
1	CLIPSEQ_SFRS1	DeepCLIP	2	0.862097107438017
1	CLIPSEQ_SFRS1	DeepCLIP	3	0.876731404958678
1	CLIPSEQ_SFRS1	DeepCLIP	4	0.862477272727273
1	CLIPSEQ_SFRS1	DeepCLIP	5	0.890884297520661
1	CLIPSEQ_SFRS1	DeepCLIP	6	0.863030991735537
1	CLIPSEQ_SFRS1	DeepCLIP	7	0.854607438016529
1	CLIPSEQ_SFRS1	DeepCLIP	8	0.866099173553719
1	CLIPSEQ_SFRS1	DeepCLIP	9	0.863929752066116

1	CLIPSEQ_SFRS1	DeepCLIP	10	0.86401652892562
1	ICLIP_HNRNPC	DeepCLIP	1	0.925673553719008
1	ICLIP_HNRNPC	DeepCLIP	2	0.95846694214876
1	ICLIP_HNRNPC	DeepCLIP	3	0.931295454545455
1	ICLIP_HNRNPC	DeepCLIP	4	0.959148760330578
1	ICLIP_HNRNPC	DeepCLIP	5	0.93654958677686
1	ICLIP_HNRNPC	DeepCLIP	6	0.924675619834711
1	ICLIP_HNRNPC	DeepCLIP	7	0.947526859504132
1	ICLIP_HNRNPC	DeepCLIP	8	0.938628099173554
1	ICLIP_HNRNPC	DeepCLIP	9	0.920756198347107
1	ICLIP_HNRNPC	DeepCLIP	10	0.93500826446281
1	ICLIP_TDP43	DeepCLIP	1	0.865297520661157
1	ICLIP_TDP43	DeepCLIP	2	0.849667355371901
1	ICLIP_TDP43	DeepCLIP	3	0.838014462809917
1	ICLIP_TDP43	DeepCLIP	4	0.858493801652893
1	ICLIP_TDP43	DeepCLIP	5	0.855092975206611
1	ICLIP_TDP43	DeepCLIP	6	0.848318181818182
1	ICLIP_TDP43	DeepCLIP	7	0.873929752066116
1	ICLIP_TDP43	DeepCLIP	8	0.85879132231405
1	ICLIP_TDP43	DeepCLIP	9	0.850644628099174
1	ICLIP_TDP43	DeepCLIP	10	0.861462809917355
1	ICLIP_TIA1	DeepCLIP	1	0.868694214876033
1	ICLIP_TIA1	DeepCLIP	2	0.823855371900826
1	ICLIP_TIA1	DeepCLIP	3	0.895382231404959
1	ICLIP_TIA1	DeepCLIP	4	0.875183884297521
1	ICLIP_TIA1	DeepCLIP	5	0.824318181818182
1	ICLIP_TIA1	DeepCLIP	6	0.861547520661157
1	ICLIP_TIA1	DeepCLIP	7	0.842805785123967
1	ICLIP_TIA1	DeepCLIP	8	0.859721074380165
1	ICLIP_TIA1	DeepCLIP	9	0.843010330578512
1	ICLIP_TIA1	DeepCLIP	10	0.844607438016529
1	ICLIP_TIAL1	DeepCLIP	1	0.818371900826446
1	ICLIP_TIAL1	DeepCLIP	2	0.841609504132232
1	ICLIP_TIAL1	DeepCLIP	3	0.817504132231405
1	ICLIP_TIAL1	DeepCLIP	4	0.875729338842975
1	ICLIP_TIAL1	DeepCLIP	5	0.832150826446281
1	ICLIP_TIAL1	DeepCLIP	6	0.847638429752066
1	ICLIP_TIAL1	DeepCLIP	7	0.818051652892562
1	ICLIP_TIAL1	DeepCLIP	8	0.83695867768595
1	ICLIP_TIAL1	DeepCLIP	9	0.817090909090909
1	ICLIP_TIAL1	DeepCLIP	10	0.807743801652893
1	PARCLIP_AGO1234	DeepCLIP	1	0.71025
1	PARCLIP_AGO1234	DeepCLIP	2	0.708464876033058
1	PARCLIP_AGO1234	DeepCLIP	3	0.694588842975207
1	PARCLIP_AGO1234	DeepCLIP	4	0.686780991735537
1	PARCLIP_AGO1234	DeepCLIP	5	0.706840909090909
1	PARCLIP_AGO1234	DeepCLIP	6	0.694590909090909
1	PARCLIP_AGO1234	DeepCLIP	7	0.725423553719008
1	PARCLIP_AGO1234	DeepCLIP	8	0.734584710743802
1	PARCLIP_AGO1234	DeepCLIP	9	0.726950413223141
1	PARCLIP_AGO1234	DeepCLIP	10	0.719202479338843
1	PARCLIP_ELAVL1A	DeepCLIP	1	0.95879132231405
1	PARCLIP_ELAVL1A	DeepCLIP	2	0.951619834710744
1	PARCLIP_ELAVL1A	DeepCLIP	3	0.975431818181818
1	PARCLIP_ELAVL1A	DeepCLIP	4	0.966688016528926
1	PARCLIP_ELAVL1A	DeepCLIP	5	0.971022727272727
1	PARCLIP_ELAVL1A	DeepCLIP	6	0.926152892561984
1	PARCLIP_ELAVL1A	DeepCLIP	7	0.960221074380165
1	PARCLIP_ELAVL1A	DeepCLIP	8	0.971572314049587
1	PARCLIP_ELAVL1A	DeepCLIP	9	0.95971694214876
1	PARCLIP_ELAVL1A	DeepCLIP	10	0.96996694214876
1	PARCLIP_ELAVL1	DeepCLIP	1	0.885322314049587
1	PARCLIP_ELAVL1	DeepCLIP	2	0.922103305785124
1	PARCLIP_ELAVL1	DeepCLIP	3	0.871382231404959
1	PARCLIP_ELAVL1	DeepCLIP	4	0.857382231404959
1	PARCLIP_ELAVL1	DeepCLIP	5	0.877256198347107
1	PARCLIP_ELAVL1	DeepCLIP	6	0.859847107438016
1	PARCLIP_ELAVL1	DeepCLIP	7	0.881049586776859
1	PARCLIP_ELAVL1	DeepCLIP	8	0.891801652892562
1	PARCLIP_ELAVL1	DeepCLIP	9	0.870039256198347
1	PARCLIP_ELAVL1	DeepCLIP	10	0.910037190082645
1	PARCLIP_EWSR1	DeepCLIP	1	0.881078512396694
1	PARCLIP_EWSR1	DeepCLIP	2	0.879477272727273

1	PARCLIP_EWSR1	DeepCLIP	3	0.88104132231405
1	PARCLIP_EWSR1	DeepCLIP	4	0.903373966942149
1	PARCLIP_EWSR1	DeepCLIP	5	0.89074173553719
1	PARCLIP_EWSR1	DeepCLIP	6	0.867378099173554
1	PARCLIP_EWSR1	DeepCLIP	7	0.883888429752066
1	PARCLIP_EWSR1	DeepCLIP	8	0.889836776859504
1	PARCLIP_EWSR1	DeepCLIP	9	0.911471074380165
1	PARCLIP_EWSR1	DeepCLIP	10	0.875311983471074
1	PARCLIP_FUS	DeepCLIP	1	0.899227272727273
1	PARCLIP_FUS	DeepCLIP	2	0.876642561983471
1	PARCLIP_FUS	DeepCLIP	3	0.902487603305785
1	PARCLIP_FUS	DeepCLIP	4	0.956111570247934
1	PARCLIP_FUS	DeepCLIP	5	0.874822314049587
1	PARCLIP_FUS	DeepCLIP	6	0.912051652892562
1	PARCLIP_FUS	DeepCLIP	7	0.954210743801653
1	PARCLIP_FUS	DeepCLIP	8	0.875239669421488
1	PARCLIP_FUS	DeepCLIP	9	0.875340909090909
1	PARCLIP_FUS	DeepCLIP	10	0.893526859504132
1	PARCLIP_HUR	DeepCLIP	1	0.962828512396694
1	PARCLIP_HUR	DeepCLIP	2	0.955045454545454
1	PARCLIP_HUR	DeepCLIP	3	0.95021694214876
1	PARCLIP_HUR	DeepCLIP	4	0.963890495867769
1	PARCLIP_HUR	DeepCLIP	5	0.975235537190083
1	PARCLIP_HUR	DeepCLIP	6	0.946530991735537
1	PARCLIP_HUR	DeepCLIP	7	0.965669421487603
1	PARCLIP_HUR	DeepCLIP	8	0.955231404958677
1	PARCLIP_HUR	DeepCLIP	9	0.938431818181818
1	PARCLIP_HUR	DeepCLIP	10	0.950770661157025
1	PARCLIP_IGF2BP123	DeepCLIP	1	0.837014462809917
1	PARCLIP_IGF2BP123	DeepCLIP	2	0.823111570247934
1	PARCLIP_IGF2BP123	DeepCLIP	3	0.782440082644628
1	PARCLIP_IGF2BP123	DeepCLIP	4	0.817824380165289
1	PARCLIP_IGF2BP123	DeepCLIP	5	0.830964876033058
1	PARCLIP_IGF2BP123	DeepCLIP	6	0.77750826446281
1	PARCLIP_IGF2BP123	DeepCLIP	7	0.82201652892562
1	PARCLIP_IGF2BP123	DeepCLIP	8	0.830855371900826
1	PARCLIP_IGF2BP123	DeepCLIP	9	0.82804958677686
1	PARCLIP_IGF2BP123	DeepCLIP	10	0.771894628099173
1	PARCLIP_MOV10	DeepCLIP	1	0.762588842975207
1	PARCLIP_MOV10	DeepCLIP	2	0.77246694214876
1	PARCLIP_MOV10	DeepCLIP	3	0.763657024793388
1	PARCLIP_MOV10	DeepCLIP	4	0.761535123966942
1	PARCLIP_MOV10	DeepCLIP	5	0.764646694214876
1	PARCLIP_MOV10	DeepCLIP	6	0.730944214876033
1	PARCLIP_MOV10	DeepCLIP	7	0.749510330578512
1	PARCLIP_MOV10	DeepCLIP	8	0.737371900826446
1	PARCLIP_MOV10	DeepCLIP	9	0.711799586776859
1	PARCLIP_MOV10	DeepCLIP	10	0.751867768595041
1	PARCLIP_PUM2	DeepCLIP	1	0.919359504132231
1	PARCLIP_PUM2	DeepCLIP	2	0.907832644628099
1	PARCLIP_PUM2	DeepCLIP	3	0.902925619834711
1	PARCLIP_PUM2	DeepCLIP	4	0.919884297520661
1	PARCLIP_PUM2	DeepCLIP	5	0.924334710743802
1	PARCLIP_PUM2	DeepCLIP	6	0.918045454545454
1	PARCLIP_PUM2	DeepCLIP	7	0.915382231404959
1	PARCLIP_PUM2	DeepCLIP	8	0.892628099173554
1	PARCLIP_PUM2	DeepCLIP	9	0.927276859504132
1	PARCLIP_PUM2	DeepCLIP	10	0.9035
1	PARCLIP_QKI	DeepCLIP	1	0.955012396694215
1	PARCLIP_QKI	DeepCLIP	2	0.973130165289256
1	PARCLIP_QKI	DeepCLIP	3	0.956665289256198
1	PARCLIP_QKI	DeepCLIP	4	0.953576446280992
1	PARCLIP_QKI	DeepCLIP	5	0.957518595041322
1	PARCLIP_QKI	DeepCLIP	6	0.95
1	PARCLIP_QKI	DeepCLIP	7	0.954654958677686
1	PARCLIP_QKI	DeepCLIP	8	0.940347107438016
1	PARCLIP_QKI	DeepCLIP	9	0.95704132231405
1	PARCLIP_QKI	DeepCLIP	10	0.966789256198347
1	PARCLIP_TAF15	DeepCLIP	1	0.945986818998822
1	PARCLIP_TAF15	DeepCLIP	2	0.950288205276532
1	PARCLIP_TAF15	DeepCLIP	3	0.971850918331508
1	PARCLIP_TAF15	DeepCLIP	4	0.973288845732702
1	PARCLIP_TAF15	DeepCLIP	5	0.889667995785385

1	PARCLIP_TAF15	DeepCLIP	6	0.954542073838398
1	PARCLIP_TAF15	DeepCLIP	7	0.900450385306696
1	PARCLIP_TAF15	DeepCLIP	8	0.907202032931843
1	PARCLIP_TAF15	DeepCLIP	9	0.955236245687251
1	PARCLIP_TAF15	DeepCLIP	10	0.956678305063736
1	ZC3H7B_Baltz2012	DeepCLIP	1	0.682989669421487
1	ZC3H7B_Baltz2012	DeepCLIP	2	0.704981404958678
1	ZC3H7B_Baltz2012	DeepCLIP	3	0.687650826446281
1	ZC3H7B_Baltz2012	DeepCLIP	4	0.703907024793388
1	ZC3H7B_Baltz2012	DeepCLIP	5	0.71251652892562
1	ZC3H7B_Baltz2012	DeepCLIP	6	0.682927685950413
1	ZC3H7B_Baltz2012	DeepCLIP	7	0.696161157024793
1	ZC3H7B_Baltz2012	DeepCLIP	8	0.703727272727273
1	ZC3H7B_Baltz2012	DeepCLIP	9	0.681365702479339
1	ZC3H7B_Baltz2012	DeepCLIP	10	0.702995867768595
1	ALKBH5_Baltz2012	RNAProt	1	0.671473
1	ALKBH5_Baltz2012	RNAProt	2	0.620525
1	ALKBH5_Baltz2012	RNAProt	3	0.625856
1	ALKBH5_Baltz2012	RNAProt	4	0.620581
1	ALKBH5_Baltz2012	RNAProt	5	0.6281
1	ALKBH5_Baltz2012	RNAProt	6	0.642128
1	ALKBH5_Baltz2012	RNAProt	7	0.611491
1	ALKBH5_Baltz2012	RNAProt	8	0.600317
1	ALKBH5_Baltz2012	RNAProt	9	0.622138
1	ALKBH5_Baltz2012	RNAProt	10	0.574514
1	C17ORF85_Baltz2012	RNAProt	1	0.746041
1	C17ORF85_Baltz2012	RNAProt	2	0.734188
1	C17ORF85_Baltz2012	RNAProt	3	0.746105
1	C17ORF85_Baltz2012	RNAProt	4	0.713947
1	C17ORF85_Baltz2012	RNAProt	5	0.777528
1	C17ORF85_Baltz2012	RNAProt	6	0.752332
1	C17ORF85_Baltz2012	RNAProt	7	0.762515
1	C17ORF85_Baltz2012	RNAProt	8	0.811864
1	C17ORF85_Baltz2012	RNAProt	9	0.784016
1	C17ORF85_Baltz2012	RNAProt	10	0.719181
1	C22ORF28_Baltz2012	RNAProt	1	0.758924
1	C22ORF28_Baltz2012	RNAProt	2	0.760816
1	C22ORF28_Baltz2012	RNAProt	3	0.733948
1	C22ORF28_Baltz2012	RNAProt	4	0.745224
1	C22ORF28_Baltz2012	RNAProt	5	0.746432
1	C22ORF28_Baltz2012	RNAProt	6	0.752184
1	C22ORF28_Baltz2012	RNAProt	7	0.722864
1	C22ORF28_Baltz2012	RNAProt	8	0.760972
1	C22ORF28_Baltz2012	RNAProt	9	0.756056
1	C22ORF28_Baltz2012	RNAProt	10	0.733748
1	CAPRIN1_Baltz2012	RNAProt	1	0.750992
1	CAPRIN1_Baltz2012	RNAProt	2	0.759872
1	CAPRIN1_Baltz2012	RNAProt	3	0.68384
1	CAPRIN1_Baltz2012	RNAProt	4	0.763152
1	CAPRIN1_Baltz2012	RNAProt	5	0.720044
1	CAPRIN1_Baltz2012	RNAProt	6	0.744432
1	CAPRIN1_Baltz2012	RNAProt	7	0.76816
1	CAPRIN1_Baltz2012	RNAProt	8	0.731292
1	CAPRIN1_Baltz2012	RNAProt	9	0.742484
1	CAPRIN1_Baltz2012	RNAProt	10	0.713072
1	CLIPSEQ_AGO2	RNAProt	1	0.799356
1	CLIPSEQ_AGO2	RNAProt	2	0.78198
1	CLIPSEQ_AGO2	RNAProt	3	0.752816
1	CLIPSEQ_AGO2	RNAProt	4	0.806516
1	CLIPSEQ_AGO2	RNAProt	5	0.772572
1	CLIPSEQ_AGO2	RNAProt	6	0.775868
1	CLIPSEQ_AGO2	RNAProt	7	0.795964
1	CLIPSEQ_AGO2	RNAProt	8	0.779056
1	CLIPSEQ_AGO2	RNAProt	9	0.768304
1	CLIPSEQ_AGO2	RNAProt	10	0.776428
1	CLIPSEQ_ELAVL1	RNAProt	1	0.978216
1	CLIPSEQ_ELAVL1	RNAProt	2	0.972724
1	CLIPSEQ_ELAVL1	RNAProt	3	0.965664
1	CLIPSEQ_ELAVL1	RNAProt	4	0.97236
1	CLIPSEQ_ELAVL1	RNAProt	5	0.978996
1	CLIPSEQ_ELAVL1	RNAProt	6	0.9718
1	CLIPSEQ_ELAVL1	RNAProt	7	0.978248
1	CLIPSEQ_ELAVL1	RNAProt	8	0.97332

1	CLIPSEQ_ELAVL1	RNAProt	9	0.980952
1	CLIPSEQ_ELAVL1	RNAProt	10	0.975888
1	CLIPSEQ_SFRS1	RNAProt	1	0.886392
1	CLIPSEQ_SFRS1	RNAProt	2	0.890536
1	CLIPSEQ_SFRS1	RNAProt	3	0.893044
1	CLIPSEQ_SFRS1	RNAProt	4	0.891524
1	CLIPSEQ_SFRS1	RNAProt	5	0.891112
1	CLIPSEQ_SFRS1	RNAProt	6	0.902168
1	CLIPSEQ_SFRS1	RNAProt	7	0.90234
1	CLIPSEQ_SFRS1	RNAProt	8	0.893736
1	CLIPSEQ_SFRS1	RNAProt	9	0.892012
1	CLIPSEQ_SFRS1	RNAProt	10	0.879652
1	ICLIP_HNRNPC	RNAProt	1	0.973276
1	ICLIP_HNRNPC	RNAProt	2	0.976224
1	ICLIP_HNRNPC	RNAProt	3	0.971496
1	ICLIP_HNRNPC	RNAProt	4	0.971404
1	ICLIP_HNRNPC	RNAProt	5	0.973904
1	ICLIP_HNRNPC	RNAProt	6	0.972724
1	ICLIP_HNRNPC	RNAProt	7	0.968132
1	ICLIP_HNRNPC	RNAProt	8	0.980944
1	ICLIP_HNRNPC	RNAProt	9	0.974692
1	ICLIP_HNRNPC	RNAProt	10	0.969192
1	ICLIP_TDP43	RNAProt	1	0.891928
1	ICLIP_TDP43	RNAProt	2	0.895744
1	ICLIP_TDP43	RNAProt	3	0.88938
1	ICLIP_TDP43	RNAProt	4	0.908696
1	ICLIP_TDP43	RNAProt	5	0.90494
1	ICLIP_TDP43	RNAProt	6	0.896848
1	ICLIP_TDP43	RNAProt	7	0.897088
1	ICLIP_TDP43	RNAProt	8	0.888088
1	ICLIP_TDP43	RNAProt	9	0.890616
1	ICLIP_TDP43	RNAProt	10	0.899716
1	ICLIP_TIA1	RNAProt	1	0.921588
1	ICLIP_TIA1	RNAProt	2	0.907312
1	ICLIP_TIA1	RNAProt	3	0.913384
1	ICLIP_TIA1	RNAProt	4	0.921816
1	ICLIP_TIA1	RNAProt	5	0.918488
1	ICLIP_TIA1	RNAProt	6	0.92972
1	ICLIP_TIA1	RNAProt	7	0.908756
1	ICLIP_TIA1	RNAProt	8	0.917008
1	ICLIP_TIA1	RNAProt	9	0.90618
1	ICLIP_TIA1	RNAProt	10	0.916816
1	ICLIP_TIAL1	RNAProt	1	0.904104
1	ICLIP_TIAL1	RNAProt	2	0.912764
1	ICLIP_TIAL1	RNAProt	3	0.899492
1	ICLIP_TIAL1	RNAProt	4	0.90546
1	ICLIP_TIAL1	RNAProt	5	0.88866
1	ICLIP_TIAL1	RNAProt	6	0.89194
1	ICLIP_TIAL1	RNAProt	7	0.914472
1	ICLIP_TIAL1	RNAProt	8	0.908528
1	ICLIP_TIAL1	RNAProt	9	0.909888
1	ICLIP_TIAL1	RNAProt	10	0.894196
1	PARCLIP_AGO1234	RNAProt	1	0.804044
1	PARCLIP_AGO1234	RNAProt	2	0.820364
1	PARCLIP_AGO1234	RNAProt	3	0.792572
1	PARCLIP_AGO1234	RNAProt	4	0.81052
1	PARCLIP_AGO1234	RNAProt	5	0.802304
1	PARCLIP_AGO1234	RNAProt	6	0.807448
1	PARCLIP_AGO1234	RNAProt	7	0.80488
1	PARCLIP_AGO1234	RNAProt	8	0.812796
1	PARCLIP_AGO1234	RNAProt	9	0.786304
1	PARCLIP_AGO1234	RNAProt	10	0.802364
1	PARCLIP_ELAVL1A	RNAProt	1	0.979088
1	PARCLIP_ELAVL1A	RNAProt	2	0.967804
1	PARCLIP_ELAVL1A	RNAProt	3	0.975376
1	PARCLIP_ELAVL1A	RNAProt	4	0.981136
1	PARCLIP_ELAVL1A	RNAProt	5	0.970176
1	PARCLIP_ELAVL1A	RNAProt	6	0.976244
1	PARCLIP_ELAVL1A	RNAProt	7	0.967904
1	PARCLIP_ELAVL1A	RNAProt	8	0.976916
1	PARCLIP_ELAVL1A	RNAProt	9	0.971388
1	PARCLIP_ELAVL1A	RNAProt	10	0.964232
1	PARCLIP_ELAVL1	RNAProt	1	0.93812

1	PARCLIP_ELAVL1	RNAProt	2	0.922324
1	PARCLIP_ELAVL1	RNAProt	3	0.928512
1	PARCLIP_ELAVL1	RNAProt	4	0.94532
1	PARCLIP_ELAVL1	RNAProt	5	0.939104
1	PARCLIP_ELAVL1	RNAProt	6	0.943384
1	PARCLIP_ELAVL1	RNAProt	7	0.93552
1	PARCLIP_ELAVL1	RNAProt	8	0.935172
1	PARCLIP_ELAVL1	RNAProt	9	0.93644
1	PARCLIP_ELAVL1	RNAProt	10	0.94418
1	PARCLIP_EWSR1	RNAProt	1	0.947188
1	PARCLIP_EWSR1	RNAProt	2	0.945832
1	PARCLIP_EWSR1	RNAProt	3	0.936344
1	PARCLIP_EWSR1	RNAProt	4	0.947436
1	PARCLIP_EWSR1	RNAProt	5	0.942368
1	PARCLIP_EWSR1	RNAProt	6	0.945584
1	PARCLIP_EWSR1	RNAProt	7	0.944404
1	PARCLIP_EWSR1	RNAProt	8	0.943456
1	PARCLIP_EWSR1	RNAProt	9	0.943892
1	PARCLIP_EWSR1	RNAProt	10	0.925788
1	PARCLIP_FUS	RNAProt	1	0.9641
1	PARCLIP_FUS	RNAProt	2	0.958076
1	PARCLIP_FUS	RNAProt	3	0.961832
1	PARCLIP_FUS	RNAProt	4	0.95502
1	PARCLIP_FUS	RNAProt	5	0.969548
1	PARCLIP_FUS	RNAProt	6	0.9536
1	PARCLIP_FUS	RNAProt	7	0.95326
1	PARCLIP_FUS	RNAProt	8	0.961384
1	PARCLIP_FUS	RNAProt	9	0.96938
1	PARCLIP_FUS	RNAProt	10	0.9729
1	PARCLIP_HUR	RNAProt	1	0.978412
1	PARCLIP_HUR	RNAProt	2	0.994096
1	PARCLIP_HUR	RNAProt	3	0.992792
1	PARCLIP_HUR	RNAProt	4	0.991736
1	PARCLIP_HUR	RNAProt	5	0.990992
1	PARCLIP_HUR	RNAProt	6	0.986708
1	PARCLIP_HUR	RNAProt	7	0.993028
1	PARCLIP_HUR	RNAProt	8	0.983684
1	PARCLIP_HUR	RNAProt	9	0.987616
1	PARCLIP_HUR	RNAProt	10	0.990632
1	PARCLIP_IGF2BP123	RNAProt	1	0.868612
1	PARCLIP_IGF2BP123	RNAProt	2	0.87558
1	PARCLIP_IGF2BP123	RNAProt	3	0.864236
1	PARCLIP_IGF2BP123	RNAProt	4	0.88156
1	PARCLIP_IGF2BP123	RNAProt	5	0.876564
1	PARCLIP_IGF2BP123	RNAProt	6	0.871012
1	PARCLIP_IGF2BP123	RNAProt	7	0.886092
1	PARCLIP_IGF2BP123	RNAProt	8	0.866108
1	PARCLIP_IGF2BP123	RNAProt	9	0.869652
1	PARCLIP_IGF2BP123	RNAProt	10	0.873532
1	PARCLIP_MOV10	RNAProt	1	0.809648
1	PARCLIP_MOV10	RNAProt	2	0.799192
1	PARCLIP_MOV10	RNAProt	3	0.814072
1	PARCLIP_MOV10	RNAProt	4	0.772404
1	PARCLIP_MOV10	RNAProt	5	0.814704
1	PARCLIP_MOV10	RNAProt	6	0.811008
1	PARCLIP_MOV10	RNAProt	7	0.814176
1	PARCLIP_MOV10	RNAProt	8	0.777816
1	PARCLIP_MOV10	RNAProt	9	0.814588
1	PARCLIP_MOV10	RNAProt	10	0.813436
1	PARCLIP_PUM2	RNAProt	1	0.935624
1	PARCLIP_PUM2	RNAProt	2	0.931008
1	PARCLIP_PUM2	RNAProt	3	0.94436
1	PARCLIP_PUM2	RNAProt	4	0.946616
1	PARCLIP_PUM2	RNAProt	5	0.940932
1	PARCLIP_PUM2	RNAProt	6	0.92826
1	PARCLIP_PUM2	RNAProt	7	0.944904
1	PARCLIP_PUM2	RNAProt	8	0.928228
1	PARCLIP_PUM2	RNAProt	9	0.952452
1	PARCLIP_PUM2	RNAProt	10	0.93668
1	PARCLIP_QKI	RNAProt	1	0.973776
1	PARCLIP_QKI	RNAProt	2	0.9757
1	PARCLIP_QKI	RNAProt	3	0.960444
1	PARCLIP_QKI	RNAProt	4	0.96808

1	PARCLIP_QKI	RNAProt	5	0.971432
1	PARCLIP_QKI	RNAProt	6	0.965848
1	PARCLIP_QKI	RNAProt	7	0.972836
1	PARCLIP_QKI	RNAProt	8	0.957104
1	PARCLIP_QKI	RNAProt	9	0.965496
1	PARCLIP_QKI	RNAProt	10	0.962004
1	PARCLIP_TAF15	RNAProt	1	0.980596
1	PARCLIP_TAF15	RNAProt	2	0.975508
1	PARCLIP_TAF15	RNAProt	3	0.971112
1	PARCLIP_TAF15	RNAProt	4	0.981196
1	PARCLIP_TAF15	RNAProt	5	0.97776
1	PARCLIP_TAF15	RNAProt	6	0.98022
1	PARCLIP_TAF15	RNAProt	7	0.977044
1	PARCLIP_TAF15	RNAProt	8	0.980196
1	PARCLIP_TAF15	RNAProt	9	0.964724
1	PARCLIP_TAF15	RNAProt	10	0.981263
1	ZC3H7B_Baltz2012	RNAProt	1	0.690232
1	ZC3H7B_Baltz2012	RNAProt	2	0.694452
1	ZC3H7B_Baltz2012	RNAProt	3	0.716808
1	ZC3H7B_Baltz2012	RNAProt	4	0.719332
1	ZC3H7B_Baltz2012	RNAProt	5	0.71634
1	ZC3H7B_Baltz2012	RNAProt	6	0.706228
1	ZC3H7B_Baltz2012	RNAProt	7	0.713876
1	ZC3H7B_Baltz2012	RNAProt	8	0.711316
1	ZC3H7B_Baltz2012	RNAProt	9	0.685012
1	ZC3H7B_Baltz2012	RNAProt	10	0.683536
1	ALKBH5_Baltz2012	GraphProt	1	0.678038379530917
1	ALKBH5_Baltz2012	GraphProt	2	0.723207271911121
1	ALKBH5_Baltz2012	GraphProt	3	0.703288070923578
1	ALKBH5_Baltz2012	GraphProt	4	0.728537762316238
1	ALKBH5_Baltz2012	GraphProt	5	0.692234317136124
1	ALKBH5_Baltz2012	GraphProt	6	0.74189204354169
1	ALKBH5_Baltz2012	GraphProt	7	0.676803950173942
1	ALKBH5_Baltz2012	GraphProt	8	0.703188602442334
1	ALKBH5_Baltz2012	GraphProt	9	0.699983040307536
1	ALKBH5_Baltz2012	GraphProt	10	0.662728249194415
1	C17ORF85_Baltz2012	GraphProt	1	0.813224620605147
1	C17ORF85_Baltz2012	GraphProt	2	0.762324441511924
1	C17ORF85_Baltz2012	GraphProt	3	0.75541084537059
1	C17ORF85_Baltz2012	GraphProt	4	0.780108927302865
1	C17ORF85_Baltz2012	GraphProt	5	0.838290314941984
1	C17ORF85_Baltz2012	GraphProt	6	0.804404451811508
1	C17ORF85_Baltz2012	GraphProt	7	0.800402557423632
1	C17ORF85_Baltz2012	GraphProt	8	0.825076959507459
1	C17ORF85_Baltz2012	GraphProt	9	0.810632251953587
1	C17ORF85_Baltz2012	GraphProt	10	0.846744020838267
1	C22ORF28_Baltz2012	GraphProt	1	0.724242
1	C22ORF28_Baltz2012	GraphProt	2	0.759692
1	C22ORF28_Baltz2012	GraphProt	3	0.73311
1	C22ORF28_Baltz2012	GraphProt	4	0.781328
1	C22ORF28_Baltz2012	GraphProt	5	0.712068
1	C22ORF28_Baltz2012	GraphProt	6	0.744456
1	C22ORF28_Baltz2012	GraphProt	7	0.729476
1	C22ORF28_Baltz2012	GraphProt	8	0.731128
1	C22ORF28_Baltz2012	GraphProt	9	0.756092
1	C22ORF28_Baltz2012	GraphProt	10	0.74014
1	CAPRIN1_Baltz2012	GraphProt	1	0.689732
1	CAPRIN1_Baltz2012	GraphProt	2	0.6885
1	CAPRIN1_Baltz2012	GraphProt	3	0.675188
1	CAPRIN1_Baltz2012	GraphProt	4	0.680048
1	CAPRIN1_Baltz2012	GraphProt	5	0.655592
1	CAPRIN1_Baltz2012	GraphProt	6	0.680014
1	CAPRIN1_Baltz2012	GraphProt	7	0.69066
1	CAPRIN1_Baltz2012	GraphProt	8	0.69868
1	CAPRIN1_Baltz2012	GraphProt	9	0.704244
1	CAPRIN1_Baltz2012	GraphProt	10	0.689548
1	CLIPSEQ_AGO2	GraphProt	1	0.691756
1	CLIPSEQ_AGO2	GraphProt	2	0.731388
1	CLIPSEQ_AGO2	GraphProt	3	0.730616
1	CLIPSEQ_AGO2	GraphProt	4	0.714992
1	CLIPSEQ_AGO2	GraphProt	5	0.728888
1	CLIPSEQ_AGO2	GraphProt	6	0.731288
1	CLIPSEQ_AGO2	GraphProt	7	0.711244

1	CLIPSEQ_AGO2	GraphProt	8	0.729272
1	CLIPSEQ_AGO2	GraphProt	9	0.7193
1	CLIPSEQ_AGO2	GraphProt	10	0.720874
1	CLIPSEQ_ELAVL1	GraphProt	1	0.914366
1	CLIPSEQ_ELAVL1	GraphProt	2	0.913888
1	CLIPSEQ_ELAVL1	GraphProt	3	0.909156
1	CLIPSEQ_ELAVL1	GraphProt	4	0.916188
1	CLIPSEQ_ELAVL1	GraphProt	5	0.916856
1	CLIPSEQ_ELAVL1	GraphProt	6	0.92654
1	CLIPSEQ_ELAVL1	GraphProt	7	0.91994
1	CLIPSEQ_ELAVL1	GraphProt	8	0.907148
1	CLIPSEQ_ELAVL1	GraphProt	9	0.919404
1	CLIPSEQ_ELAVL1	GraphProt	10	0.905156
1	CLIPSEQ_SFRS1	GraphProt	1	0.851756
1	CLIPSEQ_SFRS1	GraphProt	2	0.862892
1	CLIPSEQ_SFRS1	GraphProt	3	0.856764
1	CLIPSEQ_SFRS1	GraphProt	4	0.846304
1	CLIPSEQ_SFRS1	GraphProt	5	0.87144
1	CLIPSEQ_SFRS1	GraphProt	6	0.842176
1	CLIPSEQ_SFRS1	GraphProt	7	0.844928
1	CLIPSEQ_SFRS1	GraphProt	8	0.842892
1	CLIPSEQ_SFRS1	GraphProt	9	0.864308
1	CLIPSEQ_SFRS1	GraphProt	10	0.852328
1	ICLIP_HNRNPC	GraphProt	1	0.92134
1	ICLIP_HNRNPC	GraphProt	2	0.922112
1	ICLIP_HNRNPC	GraphProt	3	0.929112
1	ICLIP_HNRNPC	GraphProt	4	0.922832
1	ICLIP_HNRNPC	GraphProt	5	0.926884
1	ICLIP_HNRNPC	GraphProt	6	0.922844
1	ICLIP_HNRNPC	GraphProt	7	0.923944
1	ICLIP_HNRNPC	GraphProt	8	0.933984
1	ICLIP_HNRNPC	GraphProt	9	0.917116
1	ICLIP_HNRNPC	GraphProt	10	0.928096
1	ICLIP_TDP43	GraphProt	1	0.850252
1	ICLIP_TDP43	GraphProt	2	0.845876
1	ICLIP_TDP43	GraphProt	3	0.834964
1	ICLIP_TDP43	GraphProt	4	0.848368
1	ICLIP_TDP43	GraphProt	5	0.847716
1	ICLIP_TDP43	GraphProt	6	0.839356
1	ICLIP_TDP43	GraphProt	7	0.8741
1	ICLIP_TDP43	GraphProt	8	0.855392
1	ICLIP_TDP43	GraphProt	9	0.842096
1	ICLIP_TDP43	GraphProt	10	0.850902
1	ICLIP_TIA1	GraphProt	1	0.816096
1	ICLIP_TIA1	GraphProt	2	0.823352
1	ICLIP_TIA1	GraphProt	3	0.815076
1	ICLIP_TIA1	GraphProt	4	0.846392
1	ICLIP_TIA1	GraphProt	5	0.823424
1	ICLIP_TIA1	GraphProt	6	0.789724
1	ICLIP_TIA1	GraphProt	7	0.8309
1	ICLIP_TIA1	GraphProt	8	0.83676
1	ICLIP_TIA1	GraphProt	9	0.824136
1	ICLIP_TIA1	GraphProt	10	0.839024
1	ICLIP_TIAL1	GraphProt	1	0.81781
1	ICLIP_TIAL1	GraphProt	2	0.778216
1	ICLIP_TIAL1	GraphProt	3	0.798048
1	ICLIP_TIAL1	GraphProt	4	0.791628
1	ICLIP_TIAL1	GraphProt	5	0.816502
1	ICLIP_TIAL1	GraphProt	6	0.780398
1	ICLIP_TIAL1	GraphProt	7	0.80719
1	ICLIP_TIAL1	GraphProt	8	0.829228
1	ICLIP_TIAL1	GraphProt	9	0.813128
1	ICLIP_TIAL1	GraphProt	10	0.7973
1	PARCLIP_AGO1234	GraphProt	1	0.737648
1	PARCLIP_AGO1234	GraphProt	2	0.68928
1	PARCLIP_AGO1234	GraphProt	3	0.701574
1	PARCLIP_AGO1234	GraphProt	4	0.691216
1	PARCLIP_AGO1234	GraphProt	5	0.69142
1	PARCLIP_AGO1234	GraphProt	6	0.689328
1	PARCLIP_AGO1234	GraphProt	7	0.734464
1	PARCLIP_AGO1234	GraphProt	8	0.726604
1	PARCLIP_AGO1234	GraphProt	9	0.70678
1	PARCLIP_AGO1234	GraphProt	10	0.699172

1	PARCLIP_ELAVL1A	GraphProt	1	0.918198
1	PARCLIP_ELAVL1A	GraphProt	2	0.894212
1	PARCLIP_ELAVL1A	GraphProt	3	0.92646
1	PARCLIP_ELAVL1A	GraphProt	4	0.92038
1	PARCLIP_ELAVL1A	GraphProt	5	0.924456
1	PARCLIP_ELAVL1A	GraphProt	6	0.89714
1	PARCLIP_ELAVL1A	GraphProt	7	0.91562
1	PARCLIP_ELAVL1A	GraphProt	8	0.920168
1	PARCLIP_ELAVL1A	GraphProt	9	0.91984
1	PARCLIP_ELAVL1A	GraphProt	10	0.904424
1	PARCLIP_ELAVL1	GraphProt	1	0.885584
1	PARCLIP_ELAVL1	GraphProt	2	0.880868
1	PARCLIP_ELAVL1	GraphProt	3	0.877632
1	PARCLIP_ELAVL1	GraphProt	4	0.85862
1	PARCLIP_ELAVL1	GraphProt	5	0.880568
1	PARCLIP_ELAVL1	GraphProt	6	0.862504
1	PARCLIP_ELAVL1	GraphProt	7	0.881648
1	PARCLIP_ELAVL1	GraphProt	8	0.892348
1	PARCLIP_ELAVL1	GraphProt	9	0.870068
1	PARCLIP_ELAVL1	GraphProt	10	0.876332
1	PARCLIP_EWSR1	GraphProt	1	0.806944
1	PARCLIP_EWSR1	GraphProt	2	0.828436
1	PARCLIP_EWSR1	GraphProt	3	0.81732
1	PARCLIP_EWSR1	GraphProt	4	0.847428
1	PARCLIP_EWSR1	GraphProt	5	0.816844
1	PARCLIP_EWSR1	GraphProt	6	0.821012
1	PARCLIP_EWSR1	GraphProt	7	0.803072
1	PARCLIP_EWSR1	GraphProt	8	0.807152
1	PARCLIP_EWSR1	GraphProt	9	0.836214
1	PARCLIP_EWSR1	GraphProt	10	0.848244
1	PARCLIP_FUS	GraphProt	1	0.851916
1	PARCLIP_FUS	GraphProt	2	0.835232
1	PARCLIP_FUS	GraphProt	3	0.833256
1	PARCLIP_FUS	GraphProt	4	0.830524
1	PARCLIP_FUS	GraphProt	5	0.845296
1	PARCLIP_FUS	GraphProt	6	0.844092
1	PARCLIP_FUS	GraphProt	7	0.85832
1	PARCLIP_FUS	GraphProt	8	0.83426
1	PARCLIP_FUS	GraphProt	9	0.841308
1	PARCLIP_FUS	GraphProt	10	0.84484
1	PARCLIP_HUR	GraphProt	1	0.935776
1	PARCLIP_HUR	GraphProt	2	0.93492
1	PARCLIP_HUR	GraphProt	3	0.926512
1	PARCLIP_HUR	GraphProt	4	0.941332
1	PARCLIP_HUR	GraphProt	5	0.932716
1	PARCLIP_HUR	GraphProt	6	0.941828
1	PARCLIP_HUR	GraphProt	7	0.952896
1	PARCLIP_HUR	GraphProt	8	0.936592
1	PARCLIP_HUR	GraphProt	9	0.929662
1	PARCLIP_HUR	GraphProt	10	0.93576
1	PARCLIP_IGF2BP123	GraphProt	1	0.7891
1	PARCLIP_IGF2BP123	GraphProt	2	0.79134
1	PARCLIP_IGF2BP123	GraphProt	3	0.786036
1	PARCLIP_IGF2BP123	GraphProt	4	0.763652
1	PARCLIP_IGF2BP123	GraphProt	5	0.781024
1	PARCLIP_IGF2BP123	GraphProt	6	0.772724
1	PARCLIP_IGF2BP123	GraphProt	7	0.776328
1	PARCLIP_IGF2BP123	GraphProt	8	0.781384
1	PARCLIP_IGF2BP123	GraphProt	9	0.776036
1	PARCLIP_IGF2BP123	GraphProt	10	0.766956
1	PARCLIP_MOV10	GraphProt	1	0.75242
1	PARCLIP_MOV10	GraphProt	2	0.772204
1	PARCLIP_MOV10	GraphProt	3	0.750744
1	PARCLIP_MOV10	GraphProt	4	0.757194
1	PARCLIP_MOV10	GraphProt	5	0.754772
1	PARCLIP_MOV10	GraphProt	6	0.73686
1	PARCLIP_MOV10	GraphProt	7	0.739712
1	PARCLIP_MOV10	GraphProt	8	0.730856
1	PARCLIP_MOV10	GraphProt	9	0.719436
1	PARCLIP_MOV10	GraphProt	10	0.7552
1	PARCLIP_PUM2	GraphProt	1	0.891888
1	PARCLIP_PUM2	GraphProt	2	0.884252
1	PARCLIP_PUM2	GraphProt	3	0.880012

1	PARCLIP_PUM2	GraphProt	4	0.883924
1	PARCLIP_PUM2	GraphProt	5	0.899016
1	PARCLIP_PUM2	GraphProt	6	0.894496
1	PARCLIP_PUM2	GraphProt	7	0.884052
1	PARCLIP_PUM2	GraphProt	8	0.855796
1	PARCLIP_PUM2	GraphProt	9	0.897748
1	PARCLIP_PUM2	GraphProt	10	0.876032
1	PARCLIP_QKI	GraphProt	1	0.919164
1	PARCLIP_QKI	GraphProt	2	0.941688
1	PARCLIP_QKI	GraphProt	3	0.927496
1	PARCLIP_QKI	GraphProt	4	0.929772
1	PARCLIP_QKI	GraphProt	5	0.920684
1	PARCLIP_QKI	GraphProt	6	0.926424
1	PARCLIP_QKI	GraphProt	7	0.933156
1	PARCLIP_QKI	GraphProt	8	0.916908
1	PARCLIP_QKI	GraphProt	9	0.920664
1	PARCLIP_QKI	GraphProt	10	0.921752
1	PARCLIP_TAF15	GraphProt	1	0.840484
1	PARCLIP_TAF15	GraphProt	2	0.846024
1	PARCLIP_TAF15	GraphProt	3	0.851888
1	PARCLIP_TAF15	GraphProt	4	0.832872
1	PARCLIP_TAF15	GraphProt	5	0.829184
1	PARCLIP_TAF15	GraphProt	6	0.842212
1	PARCLIP_TAF15	GraphProt	7	0.833604
1	PARCLIP_TAF15	GraphProt	8	0.841504
1	PARCLIP_TAF15	GraphProt	9	0.826428
1	PARCLIP_TAF15	GraphProt	10	0.846228456913827
1	ZC3H7B_Baltz2012	GraphProt	1	0.67914
1	ZC3H7B_Baltz2012	GraphProt	2	0.708596
1	ZC3H7B_Baltz2012	GraphProt	3	0.683972
1	ZC3H7B_Baltz2012	GraphProt	4	0.696876
1	ZC3H7B_Baltz2012	GraphProt	5	0.710076
1	ZC3H7B_Baltz2012	GraphProt	6	0.685844
1	ZC3H7B_Baltz2012	GraphProt	7	0.687612
1	ZC3H7B_Baltz2012	GraphProt	8	0.708116
1	ZC3H7B_Baltz2012	GraphProt	9	0.703076
1	ZC3H7B_Baltz2012	GraphProt	10	0.709972
2	AGGF1_HepG2	DeepCLIP	1	0.813963210702341
2	AGGF1_HepG2	DeepCLIP	2	0.839787469187951
2	AGGF1_HepG2	DeepCLIP	3	0.821732765504392
2	AGGF1_HepG2	DeepCLIP	4	0.852424749163879
2	AGGF1_HepG2	DeepCLIP	5	0.780299961810418
2	AGGF1_HepG2	DeepCLIP	6	0.809326937542674
2	AGGF1_HepG2	DeepCLIP	7	0.802882445521982
2	AGGF1_HepG2	DeepCLIP	8	0.810725775653563
2	AGGF1_HepG2	DeepCLIP	9	0.820657960213399
2	AGGF1_HepG2	DeepCLIP	10	0.804311661709736
2	BUD13_K562	DeepCLIP	1	0.800243344606313
2	BUD13_K562	DeepCLIP	2	0.874049745635334
2	BUD13_K562	DeepCLIP	3	0.87017646548734
2	BUD13_K562	DeepCLIP	4	0.800314704012024
2	BUD13_K562	DeepCLIP	5	0.797921002428026
2	BUD13_K562	DeepCLIP	6	0.857793350387328
2	BUD13_K562	DeepCLIP	7	0.785012176263152
2	BUD13_K562	DeepCLIP	8	0.860008201815239
2	BUD13_K562	DeepCLIP	9	0.789770674355417
2	BUD13_K562	DeepCLIP	10	0.859411131344664
2	CSTF2T_HepG2	DeepCLIP	1	0.923782190939054
2	CSTF2T_HepG2	DeepCLIP	2	0.925688509021842
2	CSTF2T_HepG2	DeepCLIP	3	0.924741834055559
2	CSTF2T_HepG2	DeepCLIP	4	0.956675405204817
2	CSTF2T_HepG2	DeepCLIP	5	0.914336389826586
2	CSTF2T_HepG2	DeepCLIP	6	0.925022145610381
2	CSTF2T_HepG2	DeepCLIP	7	0.920135148076324
2	CSTF2T_HepG2	DeepCLIP	8	0.943456869927458
2	CSTF2T_HepG2	DeepCLIP	9	0.926827711141437
2	CSTF2T_HepG2	DeepCLIP	10	0.920130160326239
2	DDX55_HepG2	DeepCLIP	1	0.761969555162194
2	DDX55_HepG2	DeepCLIP	2	0.761996683554091
2	DDX55_HepG2	DeepCLIP	3	0.762068743345066
2	DDX55_HepG2	DeepCLIP	4	0.768146350892185
2	DDX55_HepG2	DeepCLIP	5	0.757394182316358
2	DDX55_HepG2	DeepCLIP	6	0.754876328443441

2	DDX55_HepG2	DeepCLIP	7	0.745032960996155
2	DDX55_HepG2	DeepCLIP	8	0.787837324597991
2	DDX55_HepG2	DeepCLIP	9	0.775863530624563
2	DDX55_HepG2	DeepCLIP	10	0.782432840274811
2	EFTUD2_HepG2	DeepCLIP	1	0.888228605930981
2	EFTUD2_HepG2	DeepCLIP	2	0.852305975439777
2	EFTUD2_HepG2	DeepCLIP	3	0.86996764755359
2	EFTUD2_HepG2	DeepCLIP	4	0.87242261847544
2	EFTUD2_HepG2	DeepCLIP	5	0.874993241132366
2	EFTUD2_HepG2	DeepCLIP	6	0.859195514515044
2	EFTUD2_HepG2	DeepCLIP	7	0.866595723589349
2	EFTUD2_HepG2	DeepCLIP	8	0.874749170912237
2	EFTUD2_HepG2	DeepCLIP	9	0.863352969095453
2	EFTUD2_HepG2	DeepCLIP	10	0.873985418869557
2	EWSR1_K562	DeepCLIP	1	0.869257779030545
2	EWSR1_K562	DeepCLIP	2	0.90789019288673
2	EWSR1_K562	DeepCLIP	3	0.846763971827673
2	EWSR1_K562	DeepCLIP	4	0.833999436254791
2	EWSR1_K562	DeepCLIP	5	0.856024700253685
2	EWSR1_K562	DeepCLIP	6	0.879112119961994
2	EWSR1_K562	DeepCLIP	7	0.854373598803815
2	EWSR1_K562	DeepCLIP	8	0.899092407546719
2	EWSR1_K562	DeepCLIP	9	0.88226965314736
2	EWSR1_K562	DeepCLIP	10	0.891045038015468
2	FASTKD2_HepG2	DeepCLIP	1	0.82073606301128
2	FASTKD2_HepG2	DeepCLIP	2	0.848145707011353
2	FASTKD2_HepG2	DeepCLIP	3	0.85661465584263
2	FASTKD2_HepG2	DeepCLIP	4	0.842427420163412
2	FASTKD2_HepG2	DeepCLIP	5	0.864970192258699
2	FASTKD2_HepG2	DeepCLIP	6	0.868584596621625
2	FASTKD2_HepG2	DeepCLIP	7	0.847198959893096
2	FASTKD2_HepG2	DeepCLIP	8	0.824360329323355
2	FASTKD2_HepG2	DeepCLIP	9	0.828641237082901
2	FASTKD2_HepG2	DeepCLIP	10	0.834917545886828
2	FMR1_K562	DeepCLIP	1	0.894409098040511
2	FMR1_K562	DeepCLIP	2	0.873720097768016
2	FMR1_K562	DeepCLIP	3	0.8881489310752
2	FMR1_K562	DeepCLIP	4	0.899428585584172
2	FMR1_K562	DeepCLIP	5	0.877499938069247
2	FMR1_K562	DeepCLIP	6	0.887504851242331
2	FMR1_K562	DeepCLIP	7	0.889174917219227
2	FMR1_K562	DeepCLIP	8	0.89552488377662
2	FMR1_K562	DeepCLIP	9	0.913581827039792
2	FMR1_K562	DeepCLIP	10	0.895260645896468
2	FUS_HepG2	DeepCLIP	1	0.872265775298192
2	FUS_HepG2	DeepCLIP	2	0.870332339361293
2	FUS_HepG2	DeepCLIP	3	0.803094339168911
2	FUS_HepG2	DeepCLIP	4	0.884312475952289
2	FUS_HepG2	DeepCLIP	5	0.873741102347056
2	FUS_HepG2	DeepCLIP	6	0.865922590419392
2	FUS_HepG2	DeepCLIP	7	0.789174923047326
2	FUS_HepG2	DeepCLIP	8	0.878405155829165
2	FUS_HepG2	DeepCLIP	9	0.882697672181608
2	FUS_HepG2	DeepCLIP	10	0.819156406310119
2	FXR2_K562	DeepCLIP	1	0.901286255429764
2	FXR2_K562	DeepCLIP	2	0.893356449020077
2	FXR2_K562	DeepCLIP	3	0.904194599309095
2	FXR2_K562	DeepCLIP	4	0.89321856722646
2	FXR2_K562	DeepCLIP	5	0.909893713445292
2	FXR2_K562	DeepCLIP	6	0.909010842425693
2	FXR2_K562	DeepCLIP	7	0.891644149536546
2	FXR2_K562	DeepCLIP	8	0.90862284947156
2	FXR2_K562	DeepCLIP	9	0.870072596367616
2	FXR2_K562	DeepCLIP	10	0.910788341827137
2	HNRNPA1_K562	DeepCLIP	1	0.929803691968591
2	HNRNPA1_K562	DeepCLIP	2	0.934290535886486
2	HNRNPA1_K562	DeepCLIP	3	0.928666482986637
2	HNRNPA1_K562	DeepCLIP	4	0.925641961702714
2	HNRNPA1_K562	DeepCLIP	5	0.917540983606557
2	HNRNPA1_K562	DeepCLIP	6	0.933818019010883
2	HNRNPA1_K562	DeepCLIP	7	0.931998209119713
2	HNRNPA1_K562	DeepCLIP	8	0.924717591954815
2	HNRNPA1_K562	DeepCLIP	9	0.925118473618956

2	HNRNPA1_K562	DeepCLIP	10	0.9243725030996
2	HNRNPC_HepG2	DeepCLIP	1	0.976428690181392
2	HNRNPC_HepG2	DeepCLIP	2	0.968424000070002
2	HNRNPC_HepG2	DeepCLIP	3	0.968236745622709
2	HNRNPC_HepG2	DeepCLIP	4	0.966339700567888
2	HNRNPC_HepG2	DeepCLIP	5	0.971398195707148
2	HNRNPC_HepG2	DeepCLIP	6	0.96902513934706
2	HNRNPC_HepG2	DeepCLIP	7	0.959825170847808
2	HNRNPC_HepG2	DeepCLIP	8	0.976076931827131
2	HNRNPC_HepG2	DeepCLIP	9	0.964078646867863
2	HNRNPC_HepG2	DeepCLIP	10	0.969519526088745
2	HNRNPK_HepG2	DeepCLIP	1	0.966860386992744
2	HNRNPK_HepG2	DeepCLIP	2	0.975308123562961
2	HNRNPK_HepG2	DeepCLIP	3	0.967926968556839
2	HNRNPK_HepG2	DeepCLIP	4	0.976197408104154
2	HNRNPK_HepG2	DeepCLIP	5	0.981633894711697
2	HNRNPK_HepG2	DeepCLIP	6	0.961994983427394
2	HNRNPK_HepG2	DeepCLIP	7	0.968606296097226
2	HNRNPK_HepG2	DeepCLIP	8	0.980139934008182
2	HNRNPK_HepG2	DeepCLIP	9	0.962362641464362
2	HNRNPK_HepG2	DeepCLIP	10	0.973122704470125
2	IGF2BP1_HepG2	DeepCLIP	1	0.864793026053959
2	IGF2BP1_HepG2	DeepCLIP	2	0.813108831968161
2	IGF2BP1_HepG2	DeepCLIP	3	0.815058251191633
2	IGF2BP1_HepG2	DeepCLIP	4	0.869174880836688
2	IGF2BP1_HepG2	DeepCLIP	5	0.829271079179971
2	IGF2BP1_HepG2	DeepCLIP	6	0.863702623906706
2	IGF2BP1_HepG2	DeepCLIP	7	0.824404761904762
2	IGF2BP1_HepG2	DeepCLIP	8	0.852672786801796
2	IGF2BP1_HepG2	DeepCLIP	9	0.863403269471054
2	IGF2BP1_HepG2	DeepCLIP	10	0.885896790689065
2	KHDRBS1_K562	DeepCLIP	1	0.912188186296384
2	KHDRBS1_K562	DeepCLIP	2	0.901138322352591
2	KHDRBS1_K562	DeepCLIP	3	0.893958404423795
2	KHDRBS1_K562	DeepCLIP	4	0.912945026292534
2	KHDRBS1_K562	DeepCLIP	5	0.896848395761696
2	KHDRBS1_K562	DeepCLIP	6	0.909704176181852
2	KHDRBS1_K562	DeepCLIP	7	0.867865361227043
2	KHDRBS1_K562	DeepCLIP	8	0.920030448591752
2	KHDRBS1_K562	DeepCLIP	9	0.910878371875301
2	KHDRBS1_K562	DeepCLIP	10	0.913888232669239
2	LIN28B_K562	DeepCLIP	1	0.783524020561058
2	LIN28B_K562	DeepCLIP	2	0.799666862629826
2	LIN28B_K562	DeepCLIP	3	0.822445318741615
2	LIN28B_K562	DeepCLIP	4	0.821230347156273
2	LIN28B_K562	DeepCLIP	5	0.825981700055774
2	LIN28B_K562	DeepCLIP	6	0.820505283468246
2	LIN28B_K562	DeepCLIP	7	0.784161654532025
2	LIN28B_K562	DeepCLIP	8	0.779306290417402
2	LIN28B_K562	DeepCLIP	9	0.769453865750162
2	LIN28B_K562	DeepCLIP	10	0.790076727113764
2	PCBP2_HepG2	DeepCLIP	1	0.968460717271859
2	PCBP2_HepG2	DeepCLIP	2	0.972336172658863
2	PCBP2_HepG2	DeepCLIP	3	0.970888825848065
2	PCBP2_HepG2	DeepCLIP	4	0.978857165253225
2	PCBP2_HepG2	DeepCLIP	5	0.974711090539895
2	PCBP2_HepG2	DeepCLIP	6	0.974635503762542
2	PCBP2_HepG2	DeepCLIP	7	0.965924360965122
2	PCBP2_HepG2	DeepCLIP	8	0.973444778726708
2	PCBP2_HepG2	DeepCLIP	9	0.971755740862398
2	PCBP2_HepG2	DeepCLIP	10	0.964970661132346
2	PTBP1_HepG2	DeepCLIP	1	0.950021908497291
2	PTBP1_HepG2	DeepCLIP	2	0.950004809182332
2	PTBP1_HepG2	DeepCLIP	3	0.959017216872749
2	PTBP1_HepG2	DeepCLIP	4	0.947100063053724
2	PTBP1_HepG2	DeepCLIP	5	0.944803411313334
2	PTBP1_HepG2	DeepCLIP	6	0.945169977877761
2	PTBP1_HepG2	DeepCLIP	7	0.942513171816054
2	PTBP1_HepG2	DeepCLIP	8	0.948898697245941
2	PTBP1_HepG2	DeepCLIP	9	0.949954579944641
2	PTBP1_HepG2	DeepCLIP	10	0.95940195145932
2	PUM2_K562	DeepCLIP	1	0.684246535791715
2	PUM2_K562	DeepCLIP	2	0.694859200646734

2	PUM2_K562	DeepCLIP	3	0.70198534699827
2	PUM2_K562	DeepCLIP	4	0.72048795510659
2	PUM2_K562	DeepCLIP	5	0.704052760605228
2	PUM2_K562	DeepCLIP	6	0.698921410285798
2	PUM2_K562	DeepCLIP	7	0.674760755712728
2	PUM2_K562	DeepCLIP	8	0.697619543401687
2	PUM2_K562	DeepCLIP	9	0.695219663789378
2	PUM2_K562	DeepCLIP	10	0.676021501801441
2	QKI_HepG2	DeepCLIP	1	0.902052869730888
2	QKI_HepG2	DeepCLIP	2	0.874261728983091
2	QKI_HepG2	DeepCLIP	3	0.906968325791855
2	QKI_HepG2	DeepCLIP	4	0.888525839485592
2	QKI_HepG2	DeepCLIP	5	0.874934508216242
2	QKI_HepG2	DeepCLIP	6	0.875169087878066
2	QKI_HepG2	DeepCLIP	7	0.911705167897118
2	QKI_HepG2	DeepCLIP	8	0.85994403429388
2	QKI_HepG2	DeepCLIP	9	0.893294832102882
2	QKI_HepG2	DeepCLIP	10	0.887389854727316
2	RBFOX2_K562	DeepCLIP	1	0.815162068300212
2	RBFOX2_K562	DeepCLIP	2	0.800923210495852
2	RBFOX2_K562	DeepCLIP	3	0.751194289021802
2	RBFOX2_K562	DeepCLIP	4	0.804189658498939
2	RBFOX2_K562	DeepCLIP	5	0.771378545244067
2	RBFOX2_K562	DeepCLIP	6	0.781450897163805
2	RBFOX2_K562	DeepCLIP	7	0.800793941732587
2	RBFOX2_K562	DeepCLIP	8	0.799289021802045
2	RBFOX2_K562	DeepCLIP	9	0.809806096855103
2	RBFOX2_K562	DeepCLIP	10	0.787416554119236
2	SF3B4_K562	DeepCLIP	1	0.823942802203672
2	SF3B4_K562	DeepCLIP	2	0.828462567593002
2	SF3B4_K562	DeepCLIP	3	0.827124696689914
2	SF3B4_K562	DeepCLIP	4	0.767583506713941
2	SF3B4_K562	DeepCLIP	5	0.845180866919997
2	SF3B4_K562	DeepCLIP	6	0.832020897238288
2	SF3B4_K562	DeepCLIP	7	0.826413369891631
2	SF3B4_K562	DeepCLIP	8	0.793741511132815
2	SF3B4_K562	DeepCLIP	9	0.823194171020258
2	SF3B4_K562	DeepCLIP	10	0.81414107501064
2	SFPQ_HepG2	DeepCLIP	1	0.800241901100569
2	SFPQ_HepG2	DeepCLIP	2	0.776583828938526
2	SFPQ_HepG2	DeepCLIP	3	0.769800119956931
2	SFPQ_HepG2	DeepCLIP	4	0.781381202893419
2	SFPQ_HepG2	DeepCLIP	5	0.765339673225757
2	SFPQ_HepG2	DeepCLIP	6	0.813303476583106
2	SFPQ_HepG2	DeepCLIP	7	0.780070528894445
2	SFPQ_HepG2	DeepCLIP	8	0.766381166761813
2	SFPQ_HepG2	DeepCLIP	9	0.788851412384469
2	SFPQ_HepG2	DeepCLIP	10	0.791452887999249
2	SMNDC1_K562	DeepCLIP	1	0.849927928668516
2	SMNDC1_K562	DeepCLIP	2	0.860294294813557
2	SMNDC1_K562	DeepCLIP	3	0.852615898779819
2	SMNDC1_K562	DeepCLIP	4	0.847640354843236
2	SMNDC1_K562	DeepCLIP	5	0.847295943170657
2	SMNDC1_K562	DeepCLIP	6	0.86978049779534
2	SMNDC1_K562	DeepCLIP	7	0.860349570761008
2	SMNDC1_K562	DeepCLIP	8	0.862554230665111
2	SMNDC1_K562	DeepCLIP	9	0.863402503858686
2	SMNDC1_K562	DeepCLIP	10	0.857284590341449
2	SRSF1_HepG2	DeepCLIP	1	0.950728719556306
2	SRSF1_HepG2	DeepCLIP	2	0.96139956595129
2	SRSF1_HepG2	DeepCLIP	3	0.959733783457921
2	SRSF1_HepG2	DeepCLIP	4	0.966014950566675
2	SRSF1_HepG2	DeepCLIP	5	0.949306004340487
2	SRSF1_HepG2	DeepCLIP	6	0.955578490475042
2	SRSF1_HepG2	DeepCLIP	7	0.954891728960694
2	SRSF1_HepG2	DeepCLIP	8	0.960515071135761
2	SRSF1_HepG2	DeepCLIP	9	0.945552929828792
2	SRSF1_HepG2	DeepCLIP	10	0.955081745840366
2	TAF15_HepG2	DeepCLIP	1	0.890354608527941
2	TAF15_HepG2	DeepCLIP	2	0.887798053783343
2	TAF15_HepG2	DeepCLIP	3	0.885716261318049
2	TAF15_HepG2	DeepCLIP	4	0.893229112003151
2	TAF15_HepG2	DeepCLIP	5	0.890645392706812

2	TAF15_HepG2	DeepCLIP	6	0.886247822205539
2	TAF15_HepG2	DeepCLIP	7	0.892596918798981
2	TAF15_HepG2	DeepCLIP	8	0.912720912630158
2	TAF15_HepG2	DeepCLIP	9	0.884924167684309
2	TAF15_HepG2	DeepCLIP	10	0.919501555170587
2	TARDBP_K562	DeepCLIP	1	0.984751559377821
2	TARDBP_K562	DeepCLIP	2	0.984091845507872
2	TARDBP_K562	DeepCLIP	3	0.982692632877567
2	TARDBP_K562	DeepCLIP	4	0.979717966366501
2	TARDBP_K562	DeepCLIP	5	0.975327415755491
2	TARDBP_K562	DeepCLIP	6	0.970497452837603
2	TARDBP_K562	DeepCLIP	7	0.981741168288959
2	TARDBP_K562	DeepCLIP	8	0.971372704993581
2	TARDBP_K562	DeepCLIP	9	0.986154344465906
2	TARDBP_K562	DeepCLIP	10	0.978745067031216
2	TIA1_K562	DeepCLIP	1	0.880887227350949
2	TIA1_K562	DeepCLIP	2	0.898441036596221
2	TIA1_K562	DeepCLIP	3	0.877731152988649
2	TIA1_K562	DeepCLIP	4	0.893657333231607
2	TIA1_K562	DeepCLIP	5	0.892398463925129
2	TIA1_K562	DeepCLIP	6	0.900322134906708
2	TIA1_K562	DeepCLIP	7	0.904972745691446
2	TIA1_K562	DeepCLIP	8	0.872714325678391
2	TIA1_K562	DeepCLIP	9	0.889046565448488
2	TIA1_K562	DeepCLIP	10	0.896186939972703
2	U2AF2_HepG2	DeepCLIP	1	0.940906513252517
2	U2AF2_HepG2	DeepCLIP	2	0.939846311896445
2	U2AF2_HepG2	DeepCLIP	3	0.948987877542634
2	U2AF2_HepG2	DeepCLIP	4	0.938690774604479
2	U2AF2_HepG2	DeepCLIP	5	0.94644832545716
2	U2AF2_HepG2	DeepCLIP	6	0.934776248202178
2	U2AF2_HepG2	DeepCLIP	7	0.923152660776659
2	U2AF2_HepG2	DeepCLIP	8	0.938588041914938
2	U2AF2_HepG2	DeepCLIP	9	0.925550030819807
2	U2AF2_HepG2	DeepCLIP	10	0.933897678241216
2	UPF1_K562	DeepCLIP	1	0.685387861617612
2	UPF1_K562	DeepCLIP	2	0.714188843825371
2	UPF1_K562	DeepCLIP	3	0.672631232871083
2	UPF1_K562	DeepCLIP	4	0.686235478145621
2	UPF1_K562	DeepCLIP	5	0.714140041661637
2	UPF1_K562	DeepCLIP	6	0.700449750466831
2	UPF1_K562	DeepCLIP	7	0.69705414728493
2	UPF1_K562	DeepCLIP	8	0.733572292699967
2	UPF1_K562	DeepCLIP	9	0.701353874763374
2	UPF1_K562	DeepCLIP	10	0.703489611560462
2	AGGF1_HepG2	RNAProt	1	0.853701
2	AGGF1_HepG2	RNAProt	2	0.86463
2	AGGF1_HepG2	RNAProt	3	0.869311
2	AGGF1_HepG2	RNAProt	4	0.845606
2	AGGF1_HepG2	RNAProt	5	0.849559
2	AGGF1_HepG2	RNAProt	6	0.83565
2	AGGF1_HepG2	RNAProt	7	0.877478
2	AGGF1_HepG2	RNAProt	8	0.858047
2	AGGF1_HepG2	RNAProt	9	0.871095
2	AGGF1_HepG2	RNAProt	10	0.880425
2	BUD13_K562	RNAProt	1	0.870656
2	BUD13_K562	RNAProt	2	0.854339
2	BUD13_K562	RNAProt	3	0.849152
2	BUD13_K562	RNAProt	4	0.863991
2	BUD13_K562	RNAProt	5	0.854536
2	BUD13_K562	RNAProt	6	0.864529
2	BUD13_K562	RNAProt	7	0.844847
2	BUD13_K562	RNAProt	8	0.850729
2	BUD13_K562	RNAProt	9	0.863746
2	BUD13_K562	RNAProt	10	0.874557
2	CSTF2T_HepG2	RNAProt	1	0.954988
2	CSTF2T_HepG2	RNAProt	2	0.950453
2	CSTF2T_HepG2	RNAProt	3	0.953535
2	CSTF2T_HepG2	RNAProt	4	0.958816
2	CSTF2T_HepG2	RNAProt	5	0.950734
2	CSTF2T_HepG2	RNAProt	6	0.95511
2	CSTF2T_HepG2	RNAProt	7	0.956441
2	CSTF2T_HepG2	RNAProt	8	0.966616

2	CSTF2T_HepG2	RNAProt	9	0.952799
2	CSTF2T_HepG2	RNAProt	10	0.959619
2	DDX55_HepG2	RNAProt	1	0.791566
2	DDX55_HepG2	RNAProt	2	0.791855
2	DDX55_HepG2	RNAProt	3	0.782849
2	DDX55_HepG2	RNAProt	4	0.781657
2	DDX55_HepG2	RNAProt	5	0.792056
2	DDX55_HepG2	RNAProt	6	0.770434
2	DDX55_HepG2	RNAProt	7	0.79856
2	DDX55_HepG2	RNAProt	8	0.780573
2	DDX55_HepG2	RNAProt	9	0.799451
2	DDX55_HepG2	RNAProt	10	0.778878
2	EFTUD2_HepG2	RNAProt	1	0.888508
2	EFTUD2_HepG2	RNAProt	2	0.904184
2	EFTUD2_HepG2	RNAProt	3	0.886408
2	EFTUD2_HepG2	RNAProt	4	0.893422
2	EFTUD2_HepG2	RNAProt	5	0.881384
2	EFTUD2_HepG2	RNAProt	6	0.905182
2	EFTUD2_HepG2	RNAProt	7	0.901726
2	EFTUD2_HepG2	RNAProt	8	0.893985
2	EFTUD2_HepG2	RNAProt	9	0.891402
2	EFTUD2_HepG2	RNAProt	10	0.887217
2	EWSR1_K562	RNAProt	1	0.906963
2	EWSR1_K562	RNAProt	2	0.893386
2	EWSR1_K562	RNAProt	3	0.900523
2	EWSR1_K562	RNAProt	4	0.914654
2	EWSR1_K562	RNAProt	5	0.891752
2	EWSR1_K562	RNAProt	6	0.891573
2	EWSR1_K562	RNAProt	7	0.918433
2	EWSR1_K562	RNAProt	8	0.907535
2	EWSR1_K562	RNAProt	9	0.905702
2	EWSR1_K562	RNAProt	10	0.882434
2	FASTKD2_HepG2	RNAProt	1	0.870994
2	FASTKD2_HepG2	RNAProt	2	0.871385
2	FASTKD2_HepG2	RNAProt	3	0.847306
2	FASTKD2_HepG2	RNAProt	4	0.85671
2	FASTKD2_HepG2	RNAProt	5	0.879699
2	FASTKD2_HepG2	RNAProt	6	0.868465
2	FASTKD2_HepG2	RNAProt	7	0.869428
2	FASTKD2_HepG2	RNAProt	8	0.870597
2	FASTKD2_HepG2	RNAProt	9	0.870473
2	FASTKD2_HepG2	RNAProt	10	0.86947
2	FMR1_K562	RNAProt	1	0.890927
2	FMR1_K562	RNAProt	2	0.902266
2	FMR1_K562	RNAProt	3	0.905021
2	FMR1_K562	RNAProt	4	0.907039
2	FMR1_K562	RNAProt	5	0.908555
2	FMR1_K562	RNAProt	6	0.883618
2	FMR1_K562	RNAProt	7	0.903366
2	FMR1_K562	RNAProt	8	0.905954
2	FMR1_K562	RNAProt	9	0.912195
2	FMR1_K562	RNAProt	10	0.90469
2	FUS_HepG2	RNAProt	1	0.869512
2	FUS_HepG2	RNAProt	2	0.875199
2	FUS_HepG2	RNAProt	3	0.888433
2	FUS_HepG2	RNAProt	4	0.87162
2	FUS_HepG2	RNAProt	5	0.859778
2	FUS_HepG2	RNAProt	6	0.878289
2	FUS_HepG2	RNAProt	7	0.881512
2	FUS_HepG2	RNAProt	8	0.883561
2	FUS_HepG2	RNAProt	9	0.878853
2	FUS_HepG2	RNAProt	10	0.872518
2	FXR2_K562	RNAProt	1	0.903459
2	FXR2_K562	RNAProt	2	0.916091
2	FXR2_K562	RNAProt	3	0.905216
2	FXR2_K562	RNAProt	4	0.906803
2	FXR2_K562	RNAProt	5	0.893128
2	FXR2_K562	RNAProt	6	0.912073
2	FXR2_K562	RNAProt	7	0.901734
2	FXR2_K562	RNAProt	8	0.911837
2	FXR2_K562	RNAProt	9	0.928562
2	FXR2_K562	RNAProt	10	0.921193
2	HNRNPA1_K562	RNAProt	1	0.937501

2	HNRNPA1_K562	RNAProt	2	0.940886
2	HNRNPA1_K562	RNAProt	3	0.937795
2	HNRNPA1_K562	RNAProt	4	0.938914
2	HNRNPA1_K562	RNAProt	5	0.925591
2	HNRNPA1_K562	RNAProt	6	0.936279
2	HNRNPA1_K562	RNAProt	7	0.931477
2	HNRNPA1_K562	RNAProt	8	0.937839
2	HNRNPA1_K562	RNAProt	9	0.934038
2	HNRNPA1_K562	RNAProt	10	0.94652
2	HNRNPC_HepG2	RNAProt	1	0.980129
2	HNRNPC_HepG2	RNAProt	2	0.976512
2	HNRNPC_HepG2	RNAProt	3	0.972743
2	HNRNPC_HepG2	RNAProt	4	0.969151
2	HNRNPC_HepG2	RNAProt	5	0.971324
2	HNRNPC_HepG2	RNAProt	6	0.975249
2	HNRNPC_HepG2	RNAProt	7	0.97496
2	HNRNPC_HepG2	RNAProt	8	0.977686
2	HNRNPC_HepG2	RNAProt	9	0.972937
2	HNRNPC_HepG2	RNAProt	10	0.968153
2	HNRNPK_HepG2	RNAProt	1	0.982736
2	HNRNPK_HepG2	RNAProt	2	0.985045
2	HNRNPK_HepG2	RNAProt	3	0.987351
2	HNRNPK_HepG2	RNAProt	4	0.989522
2	HNRNPK_HepG2	RNAProt	5	0.980138
2	HNRNPK_HepG2	RNAProt	6	0.984945
2	HNRNPK_HepG2	RNAProt	7	0.981901
2	HNRNPK_HepG2	RNAProt	8	0.983724
2	HNRNPK_HepG2	RNAProt	9	0.979531
2	HNRNPK_HepG2	RNAProt	10	0.983091
2	IGF2BP1_HepG2	RNAProt	1	0.87074
2	IGF2BP1_HepG2	RNAProt	2	0.885933
2	IGF2BP1_HepG2	RNAProt	3	0.884697
2	IGF2BP1_HepG2	RNAProt	4	0.868739
2	IGF2BP1_HepG2	RNAProt	5	0.883662
2	IGF2BP1_HepG2	RNAProt	6	0.874693
2	IGF2BP1_HepG2	RNAProt	7	0.872547
2	IGF2BP1_HepG2	RNAProt	8	0.871946
2	IGF2BP1_HepG2	RNAProt	9	0.877855
2	IGF2BP1_HepG2	RNAProt	10	0.866766
2	KHDRBS1_K562	RNAProt	1	0.919074
2	KHDRBS1_K562	RNAProt	2	0.912072
2	KHDRBS1_K562	RNAProt	3	0.925576
2	KHDRBS1_K562	RNAProt	4	0.912208
2	KHDRBS1_K562	RNAProt	5	0.918088
2	KHDRBS1_K562	RNAProt	6	0.905741
2	KHDRBS1_K562	RNAProt	7	0.911789
2	KHDRBS1_K562	RNAProt	8	0.909903
2	KHDRBS1_K562	RNAProt	9	0.938403
2	KHDRBS1_K562	RNAProt	10	0.911805
2	LIN28B_K562	RNAProt	1	0.829168
2	LIN28B_K562	RNAProt	2	0.851151
2	LIN28B_K562	RNAProt	3	0.837563
2	LIN28B_K562	RNAProt	4	0.807007
2	LIN28B_K562	RNAProt	5	0.850777
2	LIN28B_K562	RNAProt	6	0.83831
2	LIN28B_K562	RNAProt	7	0.824533
2	LIN28B_K562	RNAProt	8	0.833244
2	LIN28B_K562	RNAProt	9	0.82392
2	LIN28B_K562	RNAProt	10	0.855171
2	PCBP2_HepG2	RNAProt	1	0.977661
2	PCBP2_HepG2	RNAProt	2	0.970131
2	PCBP2_HepG2	RNAProt	3	0.977618
2	PCBP2_HepG2	RNAProt	4	0.977978
2	PCBP2_HepG2	RNAProt	5	0.981006
2	PCBP2_HepG2	RNAProt	6	0.979527
2	PCBP2_HepG2	RNAProt	7	0.978636
2	PCBP2_HepG2	RNAProt	8	0.977
2	PCBP2_HepG2	RNAProt	9	0.982784
2	PCBP2_HepG2	RNAProt	10	0.985241
2	PTBP1_HepG2	RNAProt	1	0.96837
2	PTBP1_HepG2	RNAProt	2	0.942114
2	PTBP1_HepG2	RNAProt	3	0.962348
2	PTBP1_HepG2	RNAProt	4	0.951491

2	PTBP1_HepG2	RNAProt	5	0.960935
2	PTBP1_HepG2	RNAProt	6	0.967956
2	PTBP1_HepG2	RNAProt	7	0.958462
2	PTBP1_HepG2	RNAProt	8	0.956332
2	PTBP1_HepG2	RNAProt	9	0.955583
2	PTBP1_HepG2	RNAProt	10	0.952087
2	PUM2_K562	RNAProt	1	0.711181
2	PUM2_K562	RNAProt	2	0.740589
2	PUM2_K562	RNAProt	3	0.709025
2	PUM2_K562	RNAProt	4	0.725998
2	PUM2_K562	RNAProt	5	0.749344
2	PUM2_K562	RNAProt	6	0.716488
2	PUM2_K562	RNAProt	7	0.729325
2	PUM2_K562	RNAProt	8	0.743384
2	PUM2_K562	RNAProt	9	0.746556
2	PUM2_K562	RNAProt	10	0.743113
2	QKI_HepG2	RNAProt	1	0.897736
2	QKI_HepG2	RNAProt	2	0.913076
2	QKI_HepG2	RNAProt	3	0.908768
2	QKI_HepG2	RNAProt	4	0.909082
2	QKI_HepG2	RNAProt	5	0.918698
2	QKI_HepG2	RNAProt	6	0.910587
2	QKI_HepG2	RNAProt	7	0.920328
2	QKI_HepG2	RNAProt	8	0.917133
2	QKI_HepG2	RNAProt	9	0.908413
2	QKI_HepG2	RNAProt	10	0.913773
2	RBFOX2_K562	RNAProt	1	0.842158
2	RBFOX2_K562	RNAProt	2	0.827617
2	RBFOX2_K562	RNAProt	3	0.825756
2	RBFOX2_K562	RNAProt	4	0.845179
2	RBFOX2_K562	RNAProt	5	0.842518
2	RBFOX2_K562	RNAProt	6	0.855372
2	RBFOX2_K562	RNAProt	7	0.843959
2	RBFOX2_K562	RNAProt	8	0.832072
2	RBFOX2_K562	RNAProt	9	0.846309
2	RBFOX2_K562	RNAProt	10	0.850253
2	SF3B4_K562	RNAProt	1	0.859502
2	SF3B4_K562	RNAProt	2	0.870749
2	SF3B4_K562	RNAProt	3	0.846293
2	SF3B4_K562	RNAProt	4	0.851742
2	SF3B4_K562	RNAProt	5	0.841469
2	SF3B4_K562	RNAProt	6	0.870853
2	SF3B4_K562	RNAProt	7	0.858903
2	SF3B4_K562	RNAProt	8	0.859915
2	SF3B4_K562	RNAProt	9	0.858971
2	SF3B4_K562	RNAProt	10	0.861991
2	SFPQ_HepG2	RNAProt	1	0.825707
2	SFPQ_HepG2	RNAProt	2	0.839411
2	SFPQ_HepG2	RNAProt	3	0.824836
2	SFPQ_HepG2	RNAProt	4	0.836155
2	SFPQ_HepG2	RNAProt	5	0.815469
2	SFPQ_HepG2	RNAProt	6	0.834349
2	SFPQ_HepG2	RNAProt	7	0.81438
2	SFPQ_HepG2	RNAProt	8	0.81218
2	SFPQ_HepG2	RNAProt	9	0.806136
2	SFPQ_HepG2	RNAProt	10	0.826035
2	SMNDC1_K562	RNAProt	1	0.887304
2	SMNDC1_K562	RNAProt	2	0.888073
2	SMNDC1_K562	RNAProt	3	0.896902
2	SMNDC1_K562	RNAProt	4	0.898097
2	SMNDC1_K562	RNAProt	5	0.877747
2	SMNDC1_K562	RNAProt	6	0.894258
2	SMNDC1_K562	RNAProt	7	0.889942
2	SMNDC1_K562	RNAProt	8	0.881965
2	SMNDC1_K562	RNAProt	9	0.886071
2	SMNDC1_K562	RNAProt	10	0.891111
2	SRSF1_HepG2	RNAProt	1	0.965327
2	SRSF1_HepG2	RNAProt	2	0.963777
2	SRSF1_HepG2	RNAProt	3	0.959609
2	SRSF1_HepG2	RNAProt	4	0.95958
2	SRSF1_HepG2	RNAProt	5	0.960553
2	SRSF1_HepG2	RNAProt	6	0.963382
2	SRSF1_HepG2	RNAProt	7	0.962422

2	SRSF1_HepG2	RNAProt	8	0.965732
2	SRSF1_HepG2	RNAProt	9	0.967328
2	SRSF1_HepG2	RNAProt	10	0.970565
2	TAF15_HepG2	RNAProt	1	0.921458
2	TAF15_HepG2	RNAProt	2	0.922439
2	TAF15_HepG2	RNAProt	3	0.926178
2	TAF15_HepG2	RNAProt	4	0.924318
2	TAF15_HepG2	RNAProt	5	0.929484
2	TAF15_HepG2	RNAProt	6	0.928782
2	TAF15_HepG2	RNAProt	7	0.925556
2	TAF15_HepG2	RNAProt	8	0.906682
2	TAF15_HepG2	RNAProt	9	0.912488
2	TAF15_HepG2	RNAProt	10	0.916414
2	TARDBP_K562	RNAProt	1	0.980408
2	TARDBP_K562	RNAProt	2	0.991677
2	TARDBP_K562	RNAProt	3	0.981342
2	TARDBP_K562	RNAProt	4	0.983115
2	TARDBP_K562	RNAProt	5	0.984958
2	TARDBP_K562	RNAProt	6	0.981208
2	TARDBP_K562	RNAProt	7	0.983431
2	TARDBP_K562	RNAProt	8	0.984747
2	TARDBP_K562	RNAProt	9	0.978256
2	TARDBP_K562	RNAProt	10	0.986119
2	TIA1_K562	RNAProt	1	0.896467
2	TIA1_K562	RNAProt	2	0.888961
2	TIA1_K562	RNAProt	3	0.895587
2	TIA1_K562	RNAProt	4	0.888512
2	TIA1_K562	RNAProt	5	0.882988
2	TIA1_K562	RNAProt	6	0.896022
2	TIA1_K562	RNAProt	7	0.903493
2	TIA1_K562	RNAProt	8	0.891318
2	TIA1_K562	RNAProt	9	0.890568
2	TIA1_K562	RNAProt	10	0.907333
2	U2AF2_HepG2	RNAProt	1	0.939694
2	U2AF2_HepG2	RNAProt	2	0.935132
2	U2AF2_HepG2	RNAProt	3	0.939556
2	U2AF2_HepG2	RNAProt	4	0.934077
2	U2AF2_HepG2	RNAProt	5	0.936482
2	U2AF2_HepG2	RNAProt	6	0.930495
2	U2AF2_HepG2	RNAProt	7	0.943387
2	U2AF2_HepG2	RNAProt	8	0.942016
2	U2AF2_HepG2	RNAProt	9	0.9325
2	U2AF2_HepG2	RNAProt	10	0.939215
2	UPF1_K562	RNAProt	1	0.767225
2	UPF1_K562	RNAProt	2	0.766626
2	UPF1_K562	RNAProt	3	0.759644
2	UPF1_K562	RNAProt	4	0.761995
2	UPF1_K562	RNAProt	5	0.772848
2	UPF1_K562	RNAProt	6	0.746114
2	UPF1_K562	RNAProt	7	0.778927
2	UPF1_K562	RNAProt	8	0.758209
2	UPF1_K562	RNAProt	9	0.766629
2	UPF1_K562	RNAProt	10	0.762816
2	AGGF1_HepG2	GraphProt	1	0.816942763503764
2	AGGF1_HepG2	GraphProt	2	0.815343228822944
2	AGGF1_HepG2	GraphProt	3	0.808164897484368
2	AGGF1_HepG2	GraphProt	4	0.795363027259203
2	AGGF1_HepG2	GraphProt	5	0.776539905985129
2	AGGF1_HepG2	GraphProt	6	0.807982218786239
2	AGGF1_HepG2	GraphProt	7	0.798768087931295
2	AGGF1_HepG2	GraphProt	8	0.792387242791438
2	AGGF1_HepG2	GraphProt	9	0.801980919088195
2	AGGF1_HepG2	GraphProt	10	0.802983703355126
2	BUD13_K562	GraphProt	1	0.791531355993361
2	BUD13_K562	GraphProt	2	0.802334560150105
2	BUD13_K562	GraphProt	3	0.781027639460201
2	BUD13_K562	GraphProt	4	0.786898318539366
2	BUD13_K562	GraphProt	5	0.79733889009165
2	BUD13_K562	GraphProt	6	0.773812874359529
2	BUD13_K562	GraphProt	7	0.777611495994804
2	BUD13_K562	GraphProt	8	0.777664718192971
2	BUD13_K562	GraphProt	9	0.785391859709894
2	BUD13_K562	GraphProt	10	0.793633115100012

2	CSTF2T_HepG2	GraphProt	1	0.924120977062154
2	CSTF2T_HepG2	GraphProt	2	0.922567628449981
2	CSTF2T_HepG2	GraphProt	3	0.924195412430706
2	CSTF2T_HepG2	GraphProt	4	0.930163953693365
2	CSTF2T_HepG2	GraphProt	5	0.918483477307007
2	CSTF2T_HepG2	GraphProt	6	0.925662572721396
2	CSTF2T_HepG2	GraphProt	7	0.916968748283627
2	CSTF2T_HepG2	GraphProt	8	0.921097850905068
2	CSTF2T_HepG2	GraphProt	9	0.92607238974021
2	CSTF2T_HepG2	GraphProt	10	0.917525833862957
2	DDX55_HepG2	GraphProt	1	0.749655747270151
2	DDX55_HepG2	GraphProt	2	0.755630574310244
2	DDX55_HepG2	GraphProt	3	0.739349837459365
2	DDX55_HepG2	GraphProt	4	0.7488272068017
2	DDX55_HepG2	GraphProt	5	0.743313328332083
2	DDX55_HepG2	GraphProt	6	0.747598566308244
2	DDX55_HepG2	GraphProt	7	0.735210469283988
2	DDX55_HepG2	GraphProt	8	0.74029674085188
2	DDX55_HepG2	GraphProt	9	0.747899264867897
2	DDX55_HepG2	GraphProt	10	0.745221808251374
2	EFTUD2_HepG2	GraphProt	1	0.877399235069138
2	EFTUD2_HepG2	GraphProt	2	0.851403353927626
2	EFTUD2_HepG2	GraphProt	3	0.866630626654898
2	EFTUD2_HepG2	GraphProt	4	0.880928949691086
2	EFTUD2_HepG2	GraphProt	5	0.872359517505148
2	EFTUD2_HepG2	GraphProt	6	0.866219476316564
2	EFTUD2_HepG2	GraphProt	7	0.866444542512504
2	EFTUD2_HepG2	GraphProt	8	0.871612238893792
2	EFTUD2_HepG2	GraphProt	9	0.86777875845837
2	EFTUD2_HepG2	GraphProt	10	0.875125188519182
2	EWSR1_K562	GraphProt	1	0.863431506985793
2	EWSR1_K562	GraphProt	2	0.875480256281598
2	EWSR1_K562	GraphProt	3	0.844384284728348
2	EWSR1_K562	GraphProt	4	0.839662783557257
2	EWSR1_K562	GraphProt	5	0.855975647089298
2	EWSR1_K562	GraphProt	6	0.849319535041274
2	EWSR1_K562	GraphProt	7	0.860736999795308
2	EWSR1_K562	GraphProt	8	0.854087227764202
2	EWSR1_K562	GraphProt	9	0.855811712366113
2	EWSR1_K562	GraphProt	10	0.853557733817517
2	FASTKD2_HepG2	GraphProt	1	0.804871108901695
2	FASTKD2_HepG2	GraphProt	2	0.808430249671313
2	FASTKD2_HepG2	GraphProt	3	0.802714202549696
2	FASTKD2_HepG2	GraphProt	4	0.817806005978506
2	FASTKD2_HepG2	GraphProt	5	0.807953571732262
2	FASTKD2_HepG2	GraphProt	6	0.810064690837972
2	FASTKD2_HepG2	GraphProt	7	0.798451082868375
2	FASTKD2_HepG2	GraphProt	8	0.820972684637071
2	FASTKD2_HepG2	GraphProt	9	0.813152877074326
2	FASTKD2_HepG2	GraphProt	10	0.793651001568584
2	FMR1_K562	GraphProt	1	0.82974042214404
2	FMR1_K562	GraphProt	2	0.813598416704018
2	FMR1_K562	GraphProt	3	0.822126847877921
2	FMR1_K562	GraphProt	4	0.834826371620197
2	FMR1_K562	GraphProt	5	0.80973433189452
2	FMR1_K562	GraphProt	6	0.824333575185181
2	FMR1_K562	GraphProt	7	0.847776807637599
2	FMR1_K562	GraphProt	8	0.827502244303159
2	FMR1_K562	GraphProt	9	0.851188353398838
2	FMR1_K562	GraphProt	10	0.831496898967415
2	FUS_HepG2	GraphProt	1	0.812006089788982
2	FUS_HepG2	GraphProt	2	0.82006207808148
2	FUS_HepG2	GraphProt	3	0.801220318179672
2	FUS_HepG2	GraphProt	4	0.807122456686966
2	FUS_HepG2	GraphProt	5	0.798707100033045
2	FUS_HepG2	GraphProt	6	0.802815937308219
2	FUS_HepG2	GraphProt	7	0.782539064344049
2	FUS_HepG2	GraphProt	8	0.807101803332861
2	FUS_HepG2	GraphProt	9	0.80093986294896
2	FUS_HepG2	GraphProt	10	0.814416351606805
2	FXR2_K562	GraphProt	1	0.857067787703626
2	FXR2_K562	GraphProt	2	0.85287849594208
2	FXR2_K562	GraphProt	3	0.850324049746015

2	FXR2_K562	GraphProt	4	0.875916472737282
2	FXR2_K562	GraphProt	5	0.85336019151048
2	FXR2_K562	GraphProt	6	0.865400495458299
2	FXR2_K562	GraphProt	7	0.863784915967791
2	FXR2_K562	GraphProt	8	0.875292358581977
2	FXR2_K562	GraphProt	9	0.859547804804704
2	FXR2_K562	GraphProt	10	0.843481656587256
2	HNRNPA1_K562	GraphProt	1	0.896588465699769
2	HNRNPA1_K562	GraphProt	2	0.887416428544005
2	HNRNPA1_K562	GraphProt	3	0.870680198980963
2	HNRNPA1_K562	GraphProt	4	0.869804721135968
2	HNRNPA1_K562	GraphProt	5	0.879019519659516
2	HNRNPA1_K562	GraphProt	6	0.880346790691552
2	HNRNPA1_K562	GraphProt	7	0.886142632274624
2	HNRNPA1_K562	GraphProt	8	0.891678641357107
2	HNRNPA1_K562	GraphProt	9	0.872413329750067
2	HNRNPA1_K562	GraphProt	10	0.881391605284952
2	HNRNPC_HepG2	GraphProt	1	0.936101151673186
2	HNRNPC_HepG2	GraphProt	2	0.940447088222512
2	HNRNPC_HepG2	GraphProt	3	0.929835193937418
2	HNRNPC_HepG2	GraphProt	4	0.938696830182529
2	HNRNPC_HepG2	GraphProt	5	0.93911954041721
2	HNRNPC_HepG2	GraphProt	6	0.938499904932638
2	HNRNPC_HepG2	GraphProt	7	0.935963643524554
2	HNRNPC_HepG2	GraphProt	8	0.936453001840932
2	HNRNPC_HepG2	GraphProt	9	0.933053312232593
2	HNRNPC_HepG2	GraphProt	10	0.941340055652919
2	HNRNPK_HepG2	GraphProt	1	0.968308992910339
2	HNRNPK_HepG2	GraphProt	2	0.967792737384355
2	HNRNPK_HepG2	GraphProt	3	0.962948881384236
2	HNRNPK_HepG2	GraphProt	4	0.960682575537316
2	HNRNPK_HepG2	GraphProt	5	0.962225750900186
2	HNRNPK_HepG2	GraphProt	6	0.96170763163584
2	HNRNPK_HepG2	GraphProt	7	0.967300830123324
2	HNRNPK_HepG2	GraphProt	8	0.963837006181134
2	HNRNPK_HepG2	GraphProt	9	0.959958867090686
2	HNRNPK_HepG2	GraphProt	10	0.964665636477649
2	IGF2BP1_HepG2	GraphProt	1	0.757619856091842
2	IGF2BP1_HepG2	GraphProt	2	0.753961516694963
2	IGF2BP1_HepG2	GraphProt	3	0.793654643521245
2	IGF2BP1_HepG2	GraphProt	4	0.762257025051107
2	IGF2BP1_HepG2	GraphProt	5	0.80069068443704
2	IGF2BP1_HepG2	GraphProt	6	0.782926095608311
2	IGF2BP1_HepG2	GraphProt	7	0.756510597436253
2	IGF2BP1_HepG2	GraphProt	8	0.79104481928826
2	IGF2BP1_HepG2	GraphProt	9	0.785326715720302
2	IGF2BP1_HepG2	GraphProt	10	0.806628603822481
2	KHDRBS1_K562	GraphProt	1	0.887354733015922
2	KHDRBS1_K562	GraphProt	2	0.858628986842779
2	KHDRBS1_K562	GraphProt	3	0.855617924729805
2	KHDRBS1_K562	GraphProt	4	0.886002819428425
2	KHDRBS1_K562	GraphProt	5	0.858305779828271
2	KHDRBS1_K562	GraphProt	6	0.866287325387671
2	KHDRBS1_K562	GraphProt	7	0.863011662181212
2	KHDRBS1_K562	GraphProt	8	0.882272630184972
2	KHDRBS1_K562	GraphProt	9	0.880111068392499
2	KHDRBS1_K562	GraphProt	10	0.870179845358623
2	LIN28B_K562	GraphProt	1	0.746428050523614
2	LIN28B_K562	GraphProt	2	0.770960590414515
2	LIN28B_K562	GraphProt	3	0.794830955923106
2	LIN28B_K562	GraphProt	4	0.777057262040197
2	LIN28B_K562	GraphProt	5	0.784589131005513
2	LIN28B_K562	GraphProt	6	0.804690644963682
2	LIN28B_K562	GraphProt	7	0.78395029316531
2	LIN28B_K562	GraphProt	8	0.757177295638834
2	LIN28B_K562	GraphProt	9	0.776334283000949
2	LIN28B_K562	GraphProt	10	0.780021915406531
2	PCBP2_HepG2	GraphProt	1	0.954189250059074
2	PCBP2_HepG2	GraphProt	2	0.964441310255198
2	PCBP2_HepG2	GraphProt	3	0.958779386924376
2	PCBP2_HepG2	GraphProt	4	0.966051929591375
2	PCBP2_HepG2	GraphProt	5	0.958958148476782
2	PCBP2_HepG2	GraphProt	6	0.959762274475007

2	PCBP2_HepG2	GraphProt	7	0.949336364980775
2	PCBP2_HepG2	GraphProt	8	0.964636017450458
2	PCBP2_HepG2	GraphProt	9	0.960653283052351
2	PCBP2_HepG2	GraphProt	10	0.950227373558119
2	PTBP1_HepG2	GraphProt	1	0.928248433900841
2	PTBP1_HepG2	GraphProt	2	0.932706610340929
2	PTBP1_HepG2	GraphProt	3	0.94145296602606
2	PTBP1_HepG2	GraphProt	4	0.928848077291396
2	PTBP1_HepG2	GraphProt	5	0.92515375284629
2	PTBP1_HepG2	GraphProt	6	0.929590772701155
2	PTBP1_HepG2	GraphProt	7	0.927306292694346
2	PTBP1_HepG2	GraphProt	8	0.928269243046541
2	PTBP1_HepG2	GraphProt	9	0.933796897278202
2	PTBP1_HepG2	GraphProt	10	0.94106105424443
2	PUM2_K562	GraphProt	1	0.669065462135989
2	PUM2_K562	GraphProt	2	0.679598077763215
2	PUM2_K562	GraphProt	3	0.670067123982347
2	PUM2_K562	GraphProt	4	0.674736506674707
2	PUM2_K562	GraphProt	5	0.671848873386146
2	PUM2_K562	GraphProt	6	0.666833420136509
2	PUM2_K562	GraphProt	7	0.66109498780187
2	PUM2_K562	GraphProt	8	0.669788725638003
2	PUM2_K562	GraphProt	9	0.656196890162002
2	PUM2_K562	GraphProt	10	0.638820550423508
2	QKI_HepG2	GraphProt	1	0.832702351412029
2	QKI_HepG2	GraphProt	2	0.844698097601323
2	QKI_HepG2	GraphProt	3	0.838291386033321
2	QKI_HepG2	GraphProt	4	0.83638662412856
2	QKI_HepG2	GraphProt	5	0.787418173224625
2	QKI_HepG2	GraphProt	6	0.851716885265272
2	QKI_HepG2	GraphProt	7	0.856481153255347
2	QKI_HepG2	GraphProt	8	0.839725865532317
2	QKI_HepG2	GraphProt	9	0.827002245066761
2	QKI_HepG2	GraphProt	10	0.804009467455621
2	RBFOX2_K562	GraphProt	1	0.786744560838034
2	RBFOX2_K562	GraphProt	2	0.771607668234545
2	RBFOX2_K562	GraphProt	3	0.753815377696157
2	RBFOX2_K562	GraphProt	4	0.772754738224808
2	RBFOX2_K562	GraphProt	5	0.769418277350347
2	RBFOX2_K562	GraphProt	6	0.749849878025896
2	RBFOX2_K562	GraphProt	7	0.766027397260274
2	RBFOX2_K562	GraphProt	8	0.767380371551886
2	RBFOX2_K562	GraphProt	9	0.785651154062676
2	RBFOX2_K562	GraphProt	10	0.762949896791143
2	SF3B4_K562	GraphProt	1	0.788647259084277
2	SF3B4_K562	GraphProt	2	0.802670279534033
2	SF3B4_K562	GraphProt	3	0.781976522979094
2	SF3B4_K562	GraphProt	4	0.765509358054345
2	SF3B4_K562	GraphProt	5	0.787615011265397
2	SF3B4_K562	GraphProt	6	0.767770708644745
2	SF3B4_K562	GraphProt	7	0.765615229625512
2	SF3B4_K562	GraphProt	8	0.766308357567998
2	SF3B4_K562	GraphProt	9	0.761807864124466
2	SF3B4_K562	GraphProt	10	0.762718868896475
2	SFPQ_HepG2	GraphProt	1	0.793129680704075
2	SFPQ_HepG2	GraphProt	2	0.764407025693134
2	SFPQ_HepG2	GraphProt	3	0.774779618249885
2	SFPQ_HepG2	GraphProt	4	0.785179092093497
2	SFPQ_HepG2	GraphProt	5	0.772665281321892
2	SFPQ_HepG2	GraphProt	6	0.795649981516773
2	SFPQ_HepG2	GraphProt	7	0.773505101729528
2	SFPQ_HepG2	GraphProt	8	0.77431351859628
2	SFPQ_HepG2	GraphProt	9	0.793996360778234
2	SFPQ_HepG2	GraphProt	10	0.775498777953638
2	SMNDC1_K562	GraphProt	1	0.84774038247597
2	SMNDC1_K562	GraphProt	2	0.858300051174795
2	SMNDC1_K562	GraphProt	3	0.858876462931648
2	SMNDC1_K562	GraphProt	4	0.839817328230687
2	SMNDC1_K562	GraphProt	5	0.846341694326257
2	SMNDC1_K562	GraphProt	6	0.855793059856096
2	SMNDC1_K562	GraphProt	7	0.849545287474104
2	SMNDC1_K562	GraphProt	8	0.859163724354548
2	SMNDC1_K562	GraphProt	9	0.859597413735493

2	SMNDC1_K562	GraphProt	10	0.859071835111715
2	SRSF1_HepG2	GraphProt	1	0.917904436211646
2	SRSF1_HepG2	GraphProt	2	0.939089959152655
2	SRSF1_HepG2	GraphProt	3	0.930018048826826
2	SRSF1_HepG2	GraphProt	4	0.943588866723663
2	SRSF1_HepG2	GraphProt	5	0.918461100028498
2	SRSF1_HepG2	GraphProt	6	0.931045882017669
2	SRSF1_HepG2	GraphProt	7	0.926795719381688
2	SRSF1_HepG2	GraphProt	8	0.929510582639714
2	SRSF1_HepG2	GraphProt	9	0.924794292508918
2	SRSF1_HepG2	GraphProt	10	0.935705112960761
2	TAF15_HepG2	GraphProt	1	0.889247773353339
2	TAF15_HepG2	GraphProt	2	0.88729284491382
2	TAF15_HepG2	GraphProt	3	0.879397987842413
2	TAF15_HepG2	GraphProt	4	0.893688442597722
2	TAF15_HepG2	GraphProt	5	0.888512571175746
2	TAF15_HepG2	GraphProt	6	0.886392928593413
2	TAF15_HepG2	GraphProt	7	0.888927362265312
2	TAF15_HepG2	GraphProt	8	0.888174726839027
2	TAF15_HepG2	GraphProt	9	0.880723178693165
2	TAF15_HepG2	GraphProt	10	0.893990602191538
2	TARDBP_K562	GraphProt	1	0.97450569758143
2	TARDBP_K562	GraphProt	2	0.975202211960817
2	TARDBP_K562	GraphProt	3	0.973645021563804
2	TARDBP_K562	GraphProt	4	0.978544073882118
2	TARDBP_K562	GraphProt	5	0.969973663782894
2	TARDBP_K562	GraphProt	6	0.968687340088976
2	TARDBP_K562	GraphProt	7	0.979369572405836
2	TARDBP_K562	GraphProt	8	0.967409224476969
2	TARDBP_K562	GraphProt	9	0.97566415264254
2	TARDBP_K562	GraphProt	10	0.977000911320099
2	TIA1_K562	GraphProt	1	0.794103613502416
2	TIA1_K562	GraphProt	2	0.796857254222025
2	TIA1_K562	GraphProt	3	0.79779977554216
2	TIA1_K562	GraphProt	4	0.795221702519438
2	TIA1_K562	GraphProt	5	0.812240008601977
2	TIA1_K562	GraphProt	6	0.822920236823182
2	TIA1_K562	GraphProt	7	0.798030418494014
2	TIA1_K562	GraphProt	8	0.782709301680066
2	TIA1_K562	GraphProt	9	0.802544743717207
2	TIA1_K562	GraphProt	10	0.789802520351053
2	U2AF2_HepG2	GraphProt	1	0.792557737505584
2	U2AF2_HepG2	GraphProt	2	0.807426723384418
2	U2AF2_HepG2	GraphProt	3	0.839874327447896
2	U2AF2_HepG2	GraphProt	4	0.814208845885804
2	U2AF2_HepG2	GraphProt	5	0.809893680815546
2	U2AF2_HepG2	GraphProt	6	0.806765692614382
2	U2AF2_HepG2	GraphProt	7	0.806196820148782
2	U2AF2_HepG2	GraphProt	8	0.803597996790976
2	U2AF2_HepG2	GraphProt	9	0.78230498695321
2	U2AF2_HepG2	GraphProt	10	0.81444385018071
2	UPF1_K562	GraphProt	1	0.675274895850024
2	UPF1_K562	GraphProt	2	0.705545986220157
2	UPF1_K562	GraphProt	3	0.683608145729851
2	UPF1_K562	GraphProt	4	0.694689102307323
2	UPF1_K562	GraphProt	5	0.692815153821503
2	UPF1_K562	GraphProt	6	0.681786772953052
2	UPF1_K562	GraphProt	7	0.674640231933985
2	UPF1_K562	GraphProt	8	0.670724593414517
2	UPF1_K562	GraphProt	9	0.694627763980131
2	UPF1_K562	GraphProt	10	0.699482436959618

[P3] Table S4: Two-sided Wilcoxon Test on Table S3 single fold AUCs, to determine significantly different AUCs between methods and single datasets. Calculated p-values for two method comparions are shown: RNAProt vs. GraphProt, and RNAProt vs. DeepCLIP.

P-value not significant, i.e., AUC distributions not significantly different
RNAProt significantly better than GraphProt/DeepCLIP
GraphProt significantly better than RNAProt
DeepCLIP significantly better than RNAProt

	Test p-value	
Dataset ID	**DeepCLIP vs. RNAProt**	**RNAProt vs. GraphProt**
ALKBH5_Baltz2012	2.16501764489381E-05	2.16501764489381E-05
C17ORF85_Baltz2012	0.001050033557774	0.003886206672584
C22ORF28_Baltz2012	0.739364350819459	0.279861005867198
CAPRIN1_Baltz2012	0.000487128970101	0.000324752646734
CLIPSEQ_AGO2	1.0825088224469E-05	1.0825088224469E-05
CLIPSEQ_ELAVL1	2.16501764489381E-05	1.0825088224469E-05
CLIPSEQ_SFRS1	7.57756175712832E-05	1.0825088224469E-05
ICLIP_HNRNPC	1.0825088224469E-05	1.0825088224469E-05
ICLIP_TDP43	1.0825088224469E-05	1.0825088224469E-05
ICLIP_TIA1	1.0825088224469E-05	1.0825088224469E-05
ICLIP_TIAL1	1.0825088224469E-05	1.0825088224469E-05
PARCLIP_AGO1234	1.0825088224469E-05	1.0825088224469E-05
PARCLIP_ELAVL1A	0.018543376128515	1.0825088224469E-05
PARCLIP_ELAVL1	1.0825088224469E-05	1.0825088224469E-05
PARCLIP_EWSR1	1.0825088224469E-05	1.0825088224469E-05
PARCLIP_FUS	0.000205676676265	1.0825088224469E-05
PARCLIP_HUR	1.0825088224469E-05	1.0825088224469E-05
PARCLIP_IGF2BP123	1.0825088224469E-05	1.0825088224469E-05
PARCLIP_MOV10	2.16501764489381E-05	1.0825088224469E-05
PARCLIP_PUM2	1.0825088224469E-05	1.0825088224469E-05
PARCLIP_QKI	0.005196042347745	1.0825088224469E-05
PARCLIP_TAF15	0.000129901058694	1.0825088224469E-05
ZC3H7B_Baltz2012	0.105122431747819	0.217562623135379
AGGF1_HepG2	0.000129901058694	1.0825088224469E-05
BUD13_K562	0.19031587607439	1.0825088224469E-05
CSTF2T_HepG2	0.000487128970101	1.0825088224469E-05
DDX55_HepG2	0.001504687263201	1.0825088224469E-05
EFTUD2_HepG2	7.57756175712832E-05	1.0825088224469E-05
EWSR1_K562	0.003886206672584	1.0825088224469E-05
FASTKD2_HepG2	0.000487128970101	1.0825088224469E-05
FMR1_K562	0.028805559765312	1.0825088224469E-05
FUS_HepG2	0.217562623135379	1.0825088224469E-05
FXR2_K562	0.075256013336509	1.0825088224469E-05
HNRNPA1_K562	0.001504687263201	1.0825088224469E-05
HNRNPC_HepG2	0.035462989023361	1.0825088224469E-05
HNRNPK_HepG2	0.000129901058694	1.0825088224469E-05
IGF2BP1_HepG2	0.002089242027323	1.0825088224469E-05
KHDRBS1_K562	0.105122431747819	1.0825088224469E-05
LIN28B_K562	0.000324752646734	1.0825088224469E-05
PCBP2_HepG2	0.002879473467709	1.0825088224469E-05
PTBP1_HepG2	0.023230639329711	1.0825088224469E-05
PUM2_K562	7.57756175712832E-05	1.0825088224469E-05
QKI_HepG2	0.000487128970101	1.0825088224469E-05
RBFOX2_K562	1.0825088224469E-05	1.0825088224469E-05
SF3B4_K562	2.16501764489381E-05	1.0825088224469E-05
SFPQ_HepG2	4.33003528978761E-05	1.0825088224469E-05
SMNDC1_K562	1.0825088224469E-05	1.0825088224469E-05
SRSF1_HepG2	0.006841455757864	1.0825088224469E-05
TAF15_HepG2	0.000205676676265	1.0825088224469E-05
TARDBP_K562	0.19031587607439	4.33003528978761E-05
TIA1_K562	0.739364350819459	1.0825088224469E-05
U2AF2_HepG2	0.970512459676546	1.0825088224469E-05
UPF1_K562	1.0825088224469E-05	1.0825088224469E-05

[P3] Table S5: Hold out validation results for DeepRAM and RNAProt. Results for the first benchmark set, containing 23 CLIP-seq datasets from 20 different RBPs and various CLIP-seq protocols.

	DeepRAM	RNAProt
Dataset_ID	**AUC**	**AUC**
ALKBH5_Baltz2012	72.62%	65.66%
C17ORF85_Baltz2012	82.20%	77.21%
C22ORF28_Baltz2012	75.87%	74.85%
CAPRIN1_Baltz2012	75.21%	75.48%
CLIPSEQ_AGO2	74.82%	74.57%
CLIPSEQ_ELAVL1	96.39%	97.86%
CLIPSEQ_SFRS1	89.90%	89.50%
ICLIP_HNRNPC	95.41%	97.17%
ICLIP_TDP43	87.50%	89.42%
ICLIP_TIA1	91.64%	92.13%
ICLIP_TIAL1	89.77%	90.34%
PARCLIP_AGO1234	78.70%	81.79%
PARCLIP_ELAVL1A	96.22%	97.33%
PARCLIP_ELAVL1	92.39%	93.73%
PARCLIP_EWSR1	94.86%	94.78%
PARCLIP_FUS	96.82%	96.87%
PARCLIP_HUR	98.75%	98.83%
PARCLIP_IGF2BP123	87.36%	87.54%
PARCLIP_MOV10	77.04%	80.11%
PARCLIP_PUM2	93.89%	94.04%
PARCLIP_QKI	95.95%	95.87%
PARCLIP_TAF15	95.99%	97.69%
ZC3H7B_Baltz2012	71.43%	69.80%
Mean	87.42%	87.50%

[P3] Table S6: Hold out validation results for DeepRAM and RNAProt. Results for the second benchmark set, containing 30 eCLIP datasets from 30 different RBPs.

	DeepRAM	RNAProt
Dataset_ID	**AUC**	**AUC**
AGGF1_HepG2	86.89%	86.38%
BUD13_K562	87.91%	85.35%
CSTF2T_HepG2	94.97%	95.64%
DDX55_HepG2	79.35%	77.63%
EFTUD2_HepG2	90.07%	90.38%
EWSR1_K562	89.94%	89.81%
FASTKD2_HepG2	88.19%	86.70%
FMR1_K562	91.98%	90.35%
FUS_HepG2	87.77%	88.01%
FXR2_K562	93.10%	92.40%
HNRNPA1_K562	92.02%	94.00%
HNRNPC_HepG2	97.15%	97.38%
HNRNPK_HepG2	97.66%	98.23%
IGF2BP1_HepG2	85.84%	87.23%
KHDRBS1_K562	89.48%	92.16%
LIN28B_K562	86.87%	85.02%
PCBP2_HepG2	97.41%	97.81%
PTBP1_HepG2	95.61%	95.48%
PUM2_K562	72.28%	72.06%
QKI_HepG2	89.51%	90.14%
RBFOX2_K562	84.78%	85.03%
SF3B4_K562	87.36%	86.38%
SFPQ_HepG2	82.05%	81.34%
SMNDC1_K562	89.44%	88.48%
SRSF1_HepG2	94.76%	95.97%
TAF15_HepG2	90.50%	91.94%
TARDBP_K562	97.79%	98.24%
TIA1_K562	90.55%	90.37%
U2AF2_HepG2	92.68%	94.07%
UPF1_K562	74.61%	76.25%
Mean	89.28%	89.34%

[P3] Table S7: Single model training runtime comparison for GraphProt, DeepCLIP, and RNAProt. Runtime is given in minutes (min), together with the mean runtime over three runs for each method.

	GraphProt	DeepCLIP	RNAProt
Run	**Runtime (min)**	**Runtime (min)**	**Runtime (min)**
1	0.733333333333333	38.6166666666667	1.2
2	0.666666666666667	36.4	1.1
3	0.616666666666667	37.2166666666667	1.3
Mean	0.672222222222222	37.4111111111111	1.2

[P4] Improving CLIP-seq data analysis by incorporating transcript information

Supplementary material for publication:

- [P4] **Michael Uhl**, Van Dinh Tran, and Rolf Backofen. **Improving CLIP-seq data analysis by incorporating transcript information**. *BMC Genomics*, 2020.

[P4] Table S1: Exon overlap statistics for 223 ENCODE eCLIP datasets. The sets cover 150 RBPs, with 103 sets from HepG2 cells and 120 sets from K562 cells. Peak regions determined by CLIPper were downloaded and preprocessed as described in the methods section. Datasets are sorted by decending ex_ratio.

dataset_id	c_in	c_ex_ol	c_close	ex_ratio	close_ratio	c_exb_pairs_ext5	pair_ratio_ext5	c_exb_pairs_ext10	pair_ratio_ext10	avg_sc_all_ext5	avg_sc_pairs_ext5	avg_sc_all_ext10	avg_sc_pairs_ext10
FXR1_K562	42876	39548	29063	92.24%	67.78%	3862	19.53%	4684	23.69%	2.027368	2.223606	2.027368	2.241747
G3BP1_HepG2	102328	89664	63440	87.62%	62.00%	10821	24.14%	12794	28.54%	2.226725	2.465022	2.226725	2.4556
YBX3_K562	136203	115225	69421	84.60%	50.97%	10964	19.03%	12980	22.53%	2.151827	2.317076	2.151827	2.319678
NIP7_HepG2	15627	13200	8004	84.47%	51.22%	869	13.17%	991	15.02%	2.316207	2.394753	2.316207	2.382899
FXR2_K562	88390	74304	48518	84.06%	54.89%	6569	17.68%	7643	20.57%	2.504316	2.681836	2.504316	2.681516
PABPC4_K562	74186	60356	26193	81.36%	35.31%	3294	10.92%	3819	12.65%	2.781076	2.764855	2.781076	2.755974
FXR2_HepG2	86940	69886	48225	80.38%	55.47%	7135	20.42%	8443	24.16%	2.402319	2.607923	2.402319	2.595037
SERBP1_K562	3961	3177	1866	80.21%	47.11%	188	11.84%	209	13.16%	4.335801	3.809292	4.335801	3.81685
YBX3_HepG2	58775	45578	27152	77.55%	46.20%	3562	15.63%	4239	18.60%	1.740862	1.903106	1.740862	1.896541
ZNF800_K562	71381	55067	42127	77.15%	59.02%	6965	25.30%	8355	30.34%	2.531112	2.676961	2.531112	2.660587
RPS3_HepG2	72192	55591	37806	77.00%	52.37%	4962	17.85%	5748	20.68%	2.826514	3.043507	2.826514	3.029385
UCHL5_HepG2	114681	86755	66949	75.65%	58.38%	8205	18.92%	10729	24.73%	1.922036	2.138185	1.922036	2.128476
PUM1_K562	90799	68580	47220	75.53%	52.00%	6986	20.37%	8408	24.52%	2.311615	2.471084	2.311615	2.451113
IGF2BP1_K562	90314	68198	29103	75.51%	32.22%	4973	14.58%	5609	16.45%	2.276359	2.461296	2.276359	2.457866
SDAD1_HepG2	67800	50482	36707	74.46%	54.14%	4656	18.45%	5570	22.07%	2.923112	3.059941	2.923112	3.02749
UPF1_HepG2	50001	36917	6221	73.83%	12.44%	569	3.08%	679	3.68%	2.790531	2.058697	2.790531	2.065948
ZNF800_HepG2	111892	82338	61086	73.59%	54.59%	8505	20.66%	10412	25.29%	2.441855	2.651258	2.441855	2.63125
SND1_HepG2	55286	40642	26754	73.51%	48.39%	3853	18.96%	4439	21.84%	2.358222	2.764839	2.358222	2.760408
ABCF1_K562	22882	16819	10345	73.50%	45.21%	932	11.08%	1061	12.62%	2.282012	2.291283	2.282012	2.267647
FMR1_K562	96637	70756	46893	73.22%	48.52%	6381	18.04%	7347	20.77%	2.36974	2.516881	2.36974	2.513114
LIN28B_HepG2	61858	45250	26911	73.15%	43.50%	4099	18.12%	4564	20.17%	2.231164	2.491916	2.231164	2.485263
IGF2BP3_HepG2	62256	45434	18100	72.98%	29.07%	2052	9.03%	2359	10.38%	1.948257	2.042768	1.948257	2.038581
DDX24_K562	112795	82131	67547	72.81%	59.88%	11996	29.21%	14551	35.43%	3.085992	3.32024	3.085992	3.302587
GRWD1_K562	105234	75632	59348	71.87%	56.40%	8878	23.48%	10988	29.06%	2.047296	2.277933	2.047296	2.260716
IGF2BP2_K562	90856	65230	30773	71.79%	33.87%	4639	14.22%	5245	16.08%	2.068871	2.210175	2.068871	2.193717
PPIG_HepG2	175310	124528	100592	71.03%	57.38%	16705	26.83%	20878	33.53%	2.663431	3.009106	2.663431	2.982211
GRWD1_HepG2	189129	134123	101907	70.92%	53.88%	13474	20.09%	17993	26.83%	2.46591	2.707246	2.46591	2.687052
RPS3_K562	64493	45519	30846	70.58%	47.83%	3917	17.21%	4532	19.91%	2.99647	3.295472	2.99647	3.278385
RPS11_K562	17981	12477	8002	69.39%	44.50%	785	12.58%	896	14.36%	2.405243	2.473482	2.405243	2.45265
NOLC1_HepG2	42761	29195	17368	68.27%	40.62%	1908	13.07%	2185	14.97%	2.218158	2.421694	2.218158	2.406345
PCBP1_HepG2	37227	25341	13489	68.07%	36.23%	1181	9.32%	1392	10.99%	2.773932	2.582122	2.773932	2.571858
FTO_HepG2	52145	34932	24565	66.99%	47.11%	2791	15.98%	3310	18.95%	2.223239	2.337004	2.223239	2.325102
SDAD1_K562	20099	13425	8339	66.79%	41.49%	859	12.80%	996	14.84%	2.815193	2.472258	2.815193	2.482865
BCLAF1_HepG2	189044	125700	94999	66.49%	50.25%	15480	24.63%	19496	31.02%	2.770406	3.10433	2.770406	3.074721
NOL12_HepG2	76163	50142	34297	65.84%	45.03%	3279	13.08%	3980	15.87%	2.385431	2.496312	2.385431	2.479649
SUB1_HepG2	98354	64185	37502	65.26%	38.13%	4943	15.40%	5864	18.27%	2.621751	2.658771	2.621751	2.6421
LIN28B_K562	79363	51288	31578	64.62%	39.79%	5306	20.69%	5960	23.24%	2.274197	2.607313	2.274197	2.600878
METAP2_K562	110035	69471	42479	63.14%	38.60%	5507	15.85%	6661	19.18%	2.287653	2.386995	2.287653	2.378581
SBDS_K562	9795	6169	3482	62.98%	35.55%	298	9.66%	329	10.67%	2.118821	2.132128	2.118821	2.135882
ZNF622_K562	180616	113007	86240	62.57%	47.75%	14071	24.90%	17537	31.04%	2.369359	2.602152	2.369359	2.574997
DDX3X_K562	71236	44256	29528	62.13%	41.45%	4210	19.03%	4924	22.25%	2.505508	2.906072	2.505508	2.871941
PABPN1_HepG2	107849	66604	39391	61.76%	36.52%	4938	14.83%	5976	17.94%	2.165685	2.446407	2.165685	2.440729
UPF1_K562	92165	56553	14242	61.36%	15.45%	1458	5.16%	1748	6.18%	2.698008	2.146688	2.698008	2.139078
TRA2A_HepG2	54592	33473	25016	61.31%	45.82%	3062	18.30%	3625	21.66%	2.196142	2.470979	2.196142	2.44697
SRSF7_K562	62603	38149	28869	60.94%	46.11%	2694	14.12%	3291	17.25%	1.705653	1.932534	1.705653	1.915807
EIF3H_HepG2	142227	84686	52159	59.54%	36.67%	7619	17.99%	9093	21.47%	2.552285	2.663145	2.552285	2.639134
WDR43_HepG2	70346	41879	27029	59.53%	38.42%	2804	13.39%	3307	15.79%	2.219987	2.327137	2.219987	2.318145
APOBEC3C_K562	70078	41229	21565	58.83%	30.77%	2737	13.28%	3180	15.43%	2.000039	2.183544	2.000039	2.168582

DDX55_K562	98471	56710	27539	57.59%	27.97%	2714	9.57%	3297	11.63%	2.137552	2.240094	2.137552	2.216797
IGF2BP1_HepG2	104649	59673	23838	57.02%	22.78%	3469	11.63%	3967	13.30%	2.019779	2.144774	2.019779	2.122631
PCBP1_K562	45171	25746	12991	57.00%	28.76%	1154	8.96%	1340	10.41%	2.767668	2.710264	2.767668	2.696608
TRA2A_K562	58325	33214	24216	56.95%	41.52%	2971	17.89%	3510	21.14%	2.009865	2.320698	2.009865	2.317145
HLTF_K562	40534	22733	17159	56.08%	42.33%	2630	23.14%	2949	25.94%	1.653296	1.806027	1.653296	1.801402
DDX3X_HepG2	83195	45181	25894	54.31%	31.12%	3092	13.69%	3592	15.90%	2.797919	3.005128	2.797919	2.994501
LARP7_K562	17475	9448	5375	54.07%	30.76%	485	10.27%	553	11.71%	2.127963	2.125723	2.127963	2.113208
AKAP1_HepG2	77147	40593	11928	52.62%	15.46%	967	4.76%	1123	5.53%	2.340277	1.892786	2.340277	1.890883
DDX55_HepG2	127964	67120	32289	52.45%	25.23%	3204	9.55%	4038	12.03%	2.451025	2.538409	2.451025	2.50826
RBM15_K562	173586	90392	48436	52.07%	27.90%	3389	7.50%	4465	9.88%	2.301334	2.096248	2.301334	2.086693
SRSF1_HepG2	108111	55707	41646	51.53%	38.52%	3166	11.37%	4088	14.68%	1.85986	2.09535	1.85986	2.08329
DDX6_HepG2	95832	48987	20103	51.12%	20.98%	2053	8.38%	2361	9.64%	2.722469	2.676672	2.722469	2.672694
UTP18_HepG2	57454	28805	17382	50.14%	30.25%	1750	12.15%	2084	14.47%	2.33513	2.400843	2.33513	2.378666
LSM11_HepG2	137427	68385	35239	49.76%	25.64%	3989	11.67%	4838	14.15%	1.935197	2.102772	1.935197	2.093168
UCHL5_K562	232620	114745	82563	49.33%	35.49%	12846	22.39%	16148	28.15%	2.457281	2.69483	2.457281	2.672777
FAM120A_K562	103697	50811	9772	49.00%	9.42%	838	3.30%	955	3.76%	2.312904	1.894399	2.312904	1.884487
SRSF1_K562	119846	58317	43985	48.66%	36.70%	3221	11.05%	4340	14.88%	2.04066	2.259659	2.04066	2.252216
FASTKD2_HepG2	155090	74440	51013	48.00%	32.89%	6114	16.43%	7405	19.90%	1.983155	2.140801	1.983155	2.125012
DDX21_K562	57876	27556	14519	47.61%	25.09%	1260	9.15%	1431	10.39%	2.042193	2.127124	2.042193	2.111212
UTP18_K562	39679	18618	11604	46.92%	29.24%	1162	12.48%	1317	14.15%	2.329023	2.322975	2.329023	2.331786
WDR43_K562	59921	28063	16486	46.83%	27.51%	1982	14.13%	2296	16.36%	2.445895	2.467094	2.445895	2.45413
SND1_K562	149839	68273	43004	45.56%	28.70%	6518	19.09%	7730	22.64%	2.106734	2.443667	2.106734	2.436742
EXOSC5_HepG2	31517	14223	8876	45.13%	28.16%	917	12.89%	1037	14.58%	2.074931	2.103677	2.074931	2.09273
DDX6_K562	142417	64105	29307	45.01%	20.58%	3353	10.46%	4001	12.48%	2.290414	2.257308	2.290414	2.260796
GNL3_K562	47780	21440	12386	44.87%	25.92%	1397	13.03%	1592	14.85%	2.1419	2.193212	2.1419	2.174065
RBM15_HepG2	181621	80144	36676	44.13%	20.19%	2392	5.97%	3010	7.51%	2.260813	2.190111	2.260813	2.179846
PRPF4_HepG2	161949	71062	50533	43.88%	31.20%	4594	12.93%	5854	16.48%	1.924211	2.036221	1.924211	2.031062
NOLC1_K562	109392	47921	21599	43.81%	19.74%	1549	6.46%	1899	7.93%	1.907712	1.999683	1.907712	1.987895
AKAP1_K562	60535	26371	8792	43.56%	14.52%	677	5.13%	806	6.11%	2.433738	2.185164	2.433738	2.178563
NCBP2_K562	88878	38401	24066	43.21%	27.08%	1828	9.52%	2211	11.52%	2.137532	2.194221	2.137532	2.184031
LARP7_HepG2	20478	8685	4853	42.41%	23.70%	570	13.13%	650	14.97%	2.79127	2.79391	2.79127	2.824574
MTPAP_K562	166763	69187	46816	41.49%	28.07%	4813	13.91%	6071	17.55%	1.902661	2.043129	1.902661	2.032974
NPM1_K562	29113	12068	6917	41.45%	23.76%	694	11.50%	803	13.31%	2.231496	2.32131	2.231496	2.329326
BUD13_HepG2	201325	82299	56861	40.88%	28.24%	4374	10.63%	6082	14.78%	2.397343	2.628888	2.397343	2.603934
GEMIN5_K562	108146	44079	23530	40.76%	21.76%	1781	8.08%	2149	9.75%	2.640924	2.551084	2.640924	2.538515
LARP4_HepG2	157286	63818	27998	40.57%	17.80%	2525	7.91%	3099	9.71%	2.101303	2.299387	2.101303	2.290408
BUD13_K562	198860	80120	59191	40.29%	29.77%	5847	14.60%	8019	20.02%	2.150252	2.276719	2.150252	2.27567
PUM2_K562	103204	41514	14959	40.23%	14.49%	1756	8.46%	1982	9.55%	2.90275	2.774412	2.90275	2.765585
SRSF7_HepG2	76608	30335	20847	39.60%	27.21%	1032	6.80%	1307	8.62%	1.769734	1.886814	1.769734	1.879609
NIPBL_K562	129674	51325	28913	39.58%	22.30%	2907	11.33%	3511	13.68%	2.531015	2.53053	2.531015	2.510952
CPSF6_K562	83147	32844	18120	39.50%	21.79%	1818	11.07%	2167	13.20%	1.884345	2.075297	1.884345	2.062654
CPEB4_K562	55310	21392	9930	38.68%	17.95%	1103	10.31%	1289	12.05%	2.462405	2.390253	2.462405	2.417
SRSF9_HepG2	98718	37977	24953	38.47%	25.28%	2190	11.53%	2597	13.68%	1.833919	1.989541	1.833919	1.977894
PHF6_K562	51751	18727	9816	36.19%	18.97%	715	7.64%	845	9.02%	2.422461	2.385419	2.422461	2.362058
DKC1_HepG2	37603	13495	7433	35.89%	19.77%	706	10.46%	804	11.92%	1.915957	1.988029	1.915957	1.975757
UTP3_K562	70986	25138	13388	35.41%	18.86%	1512	12.03%	1775	14.12%	4.661748	4.661188	4.661748	4.639347
XRCC6_K562	151299	53319	32660	35.24%	21.59%	3382	12.69%	4161	15.61%	2.056103	2.041992	2.056103	2.02807
ZC3H11A_K562	139060	48137	16199	34.62%	11.65%	1553	6.45%	1877	7.80%	2.361979	2.260809	2.361979	2.269995
SLBP_K562	11709	4032	2136	34.44%	18.24%	208	10.32%	251	12.45%	2.736686	2.730849	2.736686	2.819039
SSB_HepG2	30559	10408	6352	34.06%	20.79%	340	6.53%	404	7.76%	1.459367	1.585167	1.459367	1.589952
GRSF1_HepG2	44753	15047	6878	33.62%	15.37%	739	9.82%	839	11.15%	1.884775	2.045244	1.884775	2.033019
AGGF1_HepG2	155040	51612	32461	33.29%	20.94%	3021	11.71%	3720	14.42%	1.744862	1.938899	1.744862	1.92588
EIF3G_K562	46297	15064	8172	32.54%	17.65%	792	10.52%	924	12.27%	1.842583	2.153443	1.842583	2.147938

EIF4G2_K562	72496	23483	14260	32.39%	19.67%	1535	13.07%	1742	14.84%	1.914714	1.932607	1.914714	1.927272
LARP4_K562	141576	45809	22651	32.36%	16.00%	2738	11.95%	3188	13.92%	2.31142	2.390509	2.31142	2.383095
FASTKD2_K562	87508	27978	15195	31.97%	17.36%	1573	11.24%	1787	12.77%	2.101812	2.056232	2.101812	2.047434
SMNDC1_HepG2	73375	23396	15452	31.89%	21.06%	954	8.16%	1181	10.10%	2.11157	2.267563	2.11157	2.251549
AARS_K562	94626	29573	16898	31.25%	17.86%	1887	12.76%	2191	14.82%	2.704416	2.734322	2.704416	2.731309
DHX30_HepG2	103084	32119	14307	31.16%	13.88%	1142	7.11%	1310	8.16%	1.956362	1.994604	1.956362	1.98866
SSB_K562	7491	2266	1111	30.25%	14.83%	60	5.30%	63	5.56%	1.537752	1.530664	1.537752	1.530284
EIF3D_HepG2	244466	72744	50124	29.76%	20.50%	5720	15.73%	7118	19.57%	1.849357	2.013681	1.849357	2.001383
U2AF1_HepG2	121076	34441	21916	28.45%	18.10%	876	5.09%	1183	6.87%	1.735404	1.843389	1.735404	1.852546
AQR_HepG2	141509	40084	28656	28.33%	20.25%	1205	6.01%	1652	8.24%	2.233538	2.356836	2.233538	2.334801
LSM11_K562	121236	34323	14892	28.31%	12.28%	1004	5.85%	1247	7.27%	1.971597	2.083928	1.971597	2.077845
HLTF_HepG2	271294	76301	44668	28.12%	16.46%	5345	14.01%	6598	17.29%	1.990481	2.074537	1.990481	2.057937
YWHAG_K562	62476	17536	8447	28.07%	13.52%	562	6.41%	655	7.47%	1.771216	1.904962	1.771216	1.904214
WRN_K562	70462	19615	8654	27.84%	12.28%	811	8.27%	952	9.71%	2.721054	2.626324	2.721054	2.650046
DHX30_K562	97046	27002	11762	27.82%	12.12%	813	6.02%	971	7.19%	2.43978	2.493698	2.43978	2.485659
PUS1_K562	45926	12680	5546	27.61%	12.08%	446	7.03%	517	8.15%	2.035007	2.136162	2.035007	2.148016
FUBP3_HepG2	72812	19801	2881	27.19%	3.96%	128	1.29%	148	1.49%	2.530361	2.006291	2.530361	1.981364
DDX51_K562	77265	20946	9906	27.11%	12.82%	681	6.50%	788	7.52%	2.342699	2.272662	2.342699	2.272827
TIA1_HepG2	68403	18400	2253	26.90%	3.29%	93	1.01%	100	1.09%	2.213177	2.122274	2.213177	2.114956
DDX42_K562	91243	24397	12655	26.74%	13.87%	646	5.30%	774	6.35%	1.931166	1.967016	1.931166	1.94996
ZC3H8_K562	47558	12371	6460	26.01%	13.58%	613	9.91%	742	12.00%	1.895616	1.996348	1.895616	1.992438
AKAP8L_K562	144983	37366	20666	25.77%	14.25%	1813	9.70%	2180	11.67%	2.002585	1.969738	2.002585	1.961093
WDR3_K562	32516	8316	3635	25.58%	11.18%	322	7.74%	386	9.28%	2.78084	2.831596	2.78084	2.882536
AATF_K562	46965	11729	5826	24.97%	12.40%	337	5.75%	408	6.96%	2.418423	2.422364	2.418423	2.368816
SUPV3L1_K562	14405	3594	1725	24.95%	11.98%	169	9.40%	199	11.07%	3.948659	3.860394	3.948659	3.946335
DGCR8_HepG2	170132	42209	24786	24.81%	14.57%	1797	8.51%	2230	10.57%	1.654405	1.757119	1.654405	1.753009
AGGF1_K562	142679	34492	19585	24.17%	13.73%	1713	9.93%	2040	11.83%	1.87102	1.952742	1.87102	1.945742
ZRANB2_K562	92688	22359	12652	24.12%	13.65%	701	6.27%	843	7.54%	2.295778	2.386069	2.295778	2.387506
SUPV3L1_HepG2	108089	25559	14026	23.65%	12.98%	845	6.61%	1007	7.88%	2.119184	2.238767	2.119184	2.213959
FTO_K562	126943	29887	17439	23.54%	13.74%	2345	15.69%	2687	17.98%	2.039177	2.20161	2.039177	2.190766
NCBP2_HepG2	157948	37035	25228	23.45%	15.97%	2395	12.93%	2818	15.22%	1.936608	2.050128	1.936608	2.022888
FAM120A_HepG2	158087	36466	7506	23.07%	4.75%	489	2.68%	546	2.99%	2.137887	1.826191	2.137887	1.81823
TARDBP_K562	75847	17256	10884	22.75%	14.35%	931	10.79%	1073	12.44%	1.844828	1.855863	1.844828	1.856069
TBRG4_HepG2	105800	24052	12141	22.73%	11.48%	1424	11.84%	1814	15.08%	3.258193	3.369326	3.258193	3.431399
DGCR8_K562	65509	14767	5721	22.54%	8.73%	297	4.02%	335	4.54%	1.798297	1.785166	1.798297	1.779806
PPIL4_K562	86805	19515	10794	22.48%	12.43%	806	8.26%	943	9.66%	1.534547	1.666217	1.534547	1.665657
SF3A3_HepG2	84733	19043	10044	22.47%	11.85%	259	2.72%	321	3.37%	2.195501	2.312704	2.195501	2.272547
TBRG4_K562	139694	31105	15825	22.27%	11.33%	1513	9.73%	1795	11.54%	1.997079	2.114986	1.997079	2.110536
TROVE2_HepG2	169712	37690	18597	22.21%	10.96%	1492	7.92%	1781	9.45%	1.954113	2.024625	1.954113	2.017843
U2AF1_K562	86034	18772	12281	21.82%	14.27%	240	2.56%	320	3.41%	1.827698	1.943479	1.827698	1.942197
DROSHA_K562	156240	33898	15372	21.70%	9.84%	1216	7.17%	1444	8.52%	2.379063	2.402503	2.379063	2.387991
ZC3H11A_HepG2	239628	51612	16393	21.54%	6.84%	1015	3.93%	1233	4.78%	1.835651	1.793552	1.835651	1.779342
DDX59_HepG2	153483	32920	16043	21.45%	10.45%	995	6.04%	1199	7.28%	1.976771	1.998519	1.976771	1.995698
RBFOX2_K562	189440	40506	23161	21.38%	12.23%	1950	9.63%	2407	11.88%	1.718739	1.825892	1.718739	1.813639
FUS_K562	131843	27616	10568	20.95%	8.02%	915	6.63%	1101	7.97%	1.733197	1.860835	1.733197	1.861174
GPKOW_K562	137894	28795	19415	20.88%	14.08%	1043	7.24%	1372	9.53%	2.292443	2.429397	2.292443	2.396474
CDC40_HepG2	219099	45611	23144	20.82%	10.56%	1067	4.68%	1395	6.12%	1.986581	2.133978	1.986581	2.110988
TROVE2_K562	127502	26237	12154	20.58%	9.53%	896	6.83%	1024	7.81%	1.895031	2.016943	1.895031	1.99242
RBM22_HepG2	187122	38339	19206	20.49%	10.26%	1224	6.39%	1474	7.69%	1.90237	1.940275	1.90237	1.925348
FKBP4_HepG2	72482	14791	6595	20.41%	9.10%	402	5.44%	454	6.14%	2.060258	2.002845	2.060258	1.991951
XRCC6_HepG2	97231	19293	8592	19.84%	8.84%	558	5.78%	663	6.87%	1.695434	1.719039	1.695434	1.709221
SMNDC1_K562	108852	21562	12528	19.81%	11.51%	346	3.21%	452	4.19%	1.846216	1.873103	1.846216	1.869856
SLTM_K562	118315	23236	8608	19.64%	7.28%	548	4.72%	632	5.44%	2.013038	1.888483	2.013038	1.892261

DDX52_HepG2	186924	35967	16935	19.24%	9.06%	1397	7.77%	1670	9.29%	1.81663	1.898391	1.81663	1.890472
DROSHA_HepG2	221598	42469	20853	19.16%	9.41%	1534	7.22%	1868	8.80%	1.674413	1.744867	1.674413	1.72964
DDX52_K562	98713	18771	8146	19.02%	8.25%	442	4.71%	520	5.54%	2.102049	2.006363	2.102049	2.006689
PTBP1_K562	118085	22423	12922	18.99%	10.94%	911	8.13%	1075	9.59%	1.868555	1.806133	1.868555	1.802244
PTBP1_HepG2	198904	37443	21210	18.82%	10.66%	1268	6.77%	1564	8.35%	1.796516	1.838135	1.796516	1.830152
TIAL1_HepG2	115966	21414	3264	18.47%	2.81%	101	0.94%	118	1.10%	2.019546	1.938451	2.019546	1.887744
EFTUD2_HepG2	201569	36645	22028	18.18%	10.93%	622	3.39%	846	4.62%	2.384744	2.482945	2.384744	2.462965
XRN2_K562	108740	19081	9826	17.55%	9.04%	212	2.22%	274	2.87%	1.909576	1.867875	1.909576	1.870783
NSUN2_K562	6406	1105	476	17.25%	7.43%	29	5.25%	31	5.61%	1.941558	2.191341	1.941558	2.188345
SF3B1_K562	70767	12037	5631	17.01%	7.96%	423	7.03%	528	8.77%	4.332806	4.353305	4.332806	4.41634
SF3B4_K562	128964	20878	13386	16.19%	10.38%	337	3.23%	484	4.64%	2.162975	2.196563	2.162975	2.177933
EFTUD2_K562	158493	25303	16612	15.96%	10.48%	653	5.16%	888	7.02%	2.184471	2.23195	2.184471	2.224184
NKRF_HepG2	220453	34898	13273	15.83%	6.02%	656	3.76%	818	4.69%	1.779856	1.729728	1.779856	1.71865
BCCIP_HepG2	57627	9121	4897	15.83%	8.50%	231	5.07%	254	5.57%	1.5437	1.609475	1.5437	1.613453
PRPF8_HepG2	168180	26473	18254	15.74%	10.85%	260	1.96%	368	2.78%	2.19639	2.317828	2.19639	2.330565
U2AF2_K562	89261	13602	9571	15.24%	10.72%	188	2.76%	246	3.62%	1.839187	2.206494	1.839187	2.157331
GTF2F1_HepG2	133766	20312	8915	15.18%	6.66%	437	4.30%	528	5.20%	1.849071	1.798863	1.849071	1.800198
CSTF2_HepG2	78957	11964	2451	15.15%	3.10%	83	1.39%	104	1.74%	1.802813	1.574541	1.802813	1.556848
AQR_K562	190362	28130	16940	14.78%	8.90%	404	2.87%	547	3.89%	2.281464	2.308159	2.281464	2.288771
SLTM_HepG2	131694	18993	4358	14.42%	3.31%	190	2.00%	207	2.18%	2.118893	1.778306	2.118893	1.79972
PCBP2_HepG2	150168	19781	5018	13.17%	3.34%	455	4.60%	509	5.15%	2.542195	2.503469	2.542195	2.514568
TIA1_K562	288350	37643	8042	13.05%	2.79%	382	2.03%	442	2.35%	2.001755	1.860152	2.001755	1.84928
RBM5_HepG2	204636	26040	12447	12.73%	6.08%	713	5.48%	845	6.49%	1.680124	1.713652	1.680124	1.711965
U2AF2_HepG2	225266	27451	16775	12.19%	7.45%	177	1.29%	278	2.03%	2.122451	2.297689	2.122451	2.235232
GTF2F1_K562	106817	12958	4640	12.13%	4.34%	222	3.43%	270	4.17%	1.960067	1.899501	1.960067	1.902695
SAFB_K562	133179	16065	5935	12.06%	4.46%	444	5.53%	534	6.65%	1.642373	1.718058	1.642373	1.723495
XPO5_HepG2	142146	17138	5775	12.06%	4.06%	295	3.44%	355	4.14%	2.000288	1.963387	2.000288	1.960378
HNRNPK_K562	96214	11010	2251	11.44%	2.34%	121	2.20%	144	2.62%	1.974375	1.985797	1.974375	1.986537
CSTF2T_K562	153043	16970	4410	11.09%	2.88%	136	1.60%	167	1.97%	1.905942	1.881079	1.905942	1.867915
POLR2G_HepG2	160963	17764	8856	11.04%	5.50%	542	6.10%	653	7.35%	1.638506	1.696298	1.638506	1.695476
PRPF8_K562	187403	19870	13257	10.60%	7.07%	208	2.09%	334	3.36%	1.851732	1.957281	1.851732	1.97059
QKI_K562	71671	7152	2503	9.98%	3.49%	192	5.37%	222	6.21%	2.139788	2.079976	2.139788	2.118728
STAU2_HepG2	201961	20067	6384	9.94%	3.16%	270	2.69%	333	3.32%	1.575533	1.527116	1.575533	1.517876
HNRNPC_K562	69359	6695	1152	9.65%	1.66%	41	1.22%	47	1.40%	1.63956	1.59421	1.63956	1.567768
RBM22_K562	45764	4361	1991	9.53%	4.35%	111	5.09%	127	5.82%	1.919876	2.01327	1.919876	2.009461
HNRNPUL1_K562	78331	7350	2281	9.38%	2.91%	85	2.31%	102	2.78%	1.593869	1.60974	1.593869	1.608842
XRN2_HepG2	211363	19826	5481	9.38%	2.59%	87	0.88%	107	1.08%	1.978506	1.799536	1.978506	1.833909
RBFOX2_HepG2	134551	12610	3763	9.37%	2.80%	144	2.28%	175	2.78%	1.94154	1.667786	1.94154	1.654443
FUS_HepG2	223994	19692	5952	8.79%	2.66%	317	3.22%	365	3.71%	1.64194	1.722518	1.64194	1.73255
KHDRBS1_K562	169322	14407	3193	8.51%	1.89%	211	2.93%	238	3.30%	1.640634	1.711684	1.640634	1.704708
SAFB_HepG2	167485	13523	3383	8.07%	2.02%	157	2.32%	187	2.77%	1.558606	1.569495	1.558606	1.577496
HNRNPA1_K562	143410	11184	3251	7.80%	2.27%	244	4.36%	270	4.83%	1.596814	1.643838	1.596814	1.634018
CSTF2T_HepG2	206660	15911	2908	7.70%	1.41%	26	0.33%	37	0.47%	1.879868	1.569028	1.879868	1.64943
HNRNPK_HepG2	153927	11638	1572	7.56%	1.02%	53	0.91%	56	0.96%	1.919198	1.649048	1.919198	1.647211
KHSRP_HepG2	323040	24226	6872	7.50%	2.13%	136	1.12%	187	1.54%	1.828877	1.654707	1.828877	1.66717
SF3B4_HepG2	77360	5585	2898	7.22%	3.75%	25	0.90%	35	1.25%	1.768827	1.631334	1.768827	1.68249
HNRNPM_K562	150895	10744	4046	7.12%	2.68%	213	3.97%	245	4.56%	1.842628	1.858093	1.842628	1.845263
SAFB2_K562	154499	10664	2550	6.90%	1.65%	170	3.19%	201	3.77%	1.804131	1.677048	1.804131	1.68894
TAF15_K562	148888	10172	3261	6.83%	2.19%	135	2.65%	159	3.13%	1.764523	1.770671	1.764523	1.748854
HNRNPUL1_HepG2	233396	15806	4468	6.77%	1.91%	186	2.35%	209	2.64%	1.767031	1.809731	1.767031	1.812858
EWSR1_K562	106301	6037	1473	5.68%	1.39%	12	0.40%	12	0.40%	2.084903	1.748293	2.084903	1.748293
EXOSC5_K562	124694	6783	2584	5.44%	2.07%	142	4.19%	157	4.63%	1.845797	1.820811	1.845797	1.806171
HNRNPA1_HepG2	62559	3332	486	5.33%	0.78%	24	1.44%	25	1.50%	1.522957	1.570181	1.522957	1.550936

HNRNPU_K562	119484	6240	1206	5.22%	1.01%	52	1.67%	59	1.89%	1.49585	1.569033	1.49585	1.556323
NONO_K562	209865	10770	3108	5.13%	1.48%	32	0.59%	40	0.74%	1.915214	1.388387	1.915214	1.380322
ILF3_K562	228091	11364	2544	4.98%	1.12%	131	2.31%	148	2.60%	1.636382	1.662335	1.636382	1.659228
HNRNPL_K562	193425	9052	1730	4.68%	0.89%	91	2.01%	96	2.12%	1.636862	1.611755	1.636862	1.605005
TAF15_HepG2	294909	12566	3136	4.26%	1.06%	153	2.44%	168	2.67%	1.520138	1.558341	1.520138	1.56555
KHSRP_K562	288127	12092	2414	4.20%	0.84%	10	0.17%	12	0.20%	1.913279	1.704802	1.913279	1.675221
HNRNPL_HepG2	274669	11428	862	4.16%	0.31%	34	0.60%	35	0.61%	1.841106	1.756588	1.841106	1.751017
QKI_HepG2	228943	9409	1990	4.11%	0.87%	95	2.02%	124	2.64%	1.857909	1.741317	1.857909	1.696284
ILF3_HepG2	182351	7007	835	3.84%	0.46%	48	1.37%	54	1.54%	1.848374	1.775099	1.848374	1.828319
HNRNPC_HepG2	348829	10667	958	3.06%	0.27%	22	0.41%	25	0.47%	1.719892	1.672577	1.719892	1.65368
MATR3_K562	125688	3558	533	2.83%	0.42%	11	0.62%	13	0.73%	1.985056	1.766817	1.985056	1.767949
HNRNPU_HepG2	178289	4778	522	2.68%	0.29%	5	0.21%	5	0.21%	1.525047	1.306024	1.525047	1.306024
HNRNPM_HepG2	227093	5030	620	2.21%	0.27%	5	0.20%	5	0.20%	1.995853	1.598302	1.995853	1.598302
SFPQ_HepG2	225115	4235	867	1.88%	0.39%	17	0.80%	19	0.90%	1.912816	1.799141	1.912816	1.871046
SUGP2_HepG2	353655	6293	1161	1.78%	0.33%	20	0.64%	24	0.76%	1.684531	1.31251	1.684531	1.330542
MATR3_HepG2	238389	4168	360	1.75%	0.15%	5	0.24%	6	0.29%	2.007953	1.584725	2.007953	1.635359

[P4] Table S1 legend:

c_in	Number of sites in dataset
c_ex_ol	Number of sites overlapping with exonic regions (>= 90 % exon overlap demanded)
c_close	Number of exonic sites close to exon borders (<= 50 nt from site ends)
ex_ratio	Ratio of sites overlapping with exonic regions
close_ratio	Ratio of exonic sites near exon borders
c_exb_pairs_ext5	Number of site pairs adjacent at exon borders (ends of both sites in pair < 5 nt away from exon border)
c_exb_pairs_ext10	Number of site pairs adjacent at exon borders (ends of both sites in pair < 10 nt away from exon border)
pair_ratio_ext5	Ratio of exonic sites that are part of a pair near adjacent exon borders (ends of both sites in pair < 5 nt away from exon border)
pair_ratio_ext10	Ratio of exonic sites that are part of a pair near adjacent exon borders (ends of both sites in pair < 10 nt away from exon border)
avg_sc_all_ext5	Average site log2 fold change value over all sites (ends of both sites in pair < 5 nt away from exon border)
avg_sc_pairs_ext5	Average site log2 fold change value over pair sites (ends of both sites in pair < 5 nt away from exon border)
avg_sc_all_ext10	Average site log2 fold change value over all sites (ends of both sites in pair < 10 nt away from exon border)
avg_sc_pairs_ext10	Average site log2 fold change value over pair sites (ends of both sites in pair < 10 nt away from exon border)

Improving CLIP-seq data analysis by incorporating transcript information

Supplementary Material

Michael Uhl, Van Dinh Tran, and Rolf Backofen

November 27, 2020

Supplementary tables

Table 1: Exon overlap statistics of ENCODE eCLIP datasets (see Additional File 1 in .xlsx format).

Table 2: Peak length statistics for CLIPper (replicate 1), CLIPper IDR, PEAKachu, and PureCLIP on YBX3 K562 replicate 1 eCLIP data. Peaks were called as described in supplementary methods section "Peak caller setup". Introns for determining overlapping sites were selected based on the set of exons extracted, as described in methods section "Data preparation and exon overlap statistics". A site is counted as intron-spanning if it completely overlaps with an intronic region.

Metric	CLIPper	CLIPper IDR	PEAKachu	PureCLIP
# sites	132,842	17,982	11,537	54,308
# sites > 500 nt	0	0	471	0
# intron-spanning sites	4	2	1,096	0
Minimum length	1	1	18	1
Maximum length	263	217	22,875	25
Mean length	37.9	28.0	112.4	1.6
Median length	34	27	48	1
25th percentile	19	13	42	1
75th percentile	51	50	64	2

Table 3: Dataset statistics for the 6 eCLIP sets used for genomic and transcript context comparison. A minimum log_2 fold change (LFC) of 3 and a maximum p-value (PV) of 0.01 was used for filtering initial CLIPper replicate 1 peak sites. Moreover, only exonic sites (overlapping $\geq 90\%$ with exons) near exon borders (≤ 10 nt away) were selected. In case of overlapping sites (≤ 10 nt distance), only the site with the highest LFC was kept. Positives: number of positive training instances. Negatives: number of negative training instances.

RBP	Cell type	LFC	PV	Positives	Negatives
FMR1	K562	3	0.01	2569	2569
FXR2	K562	3	0.01	3166	3166
IGF2BP1	K562	3	0.01	2199	2199
PUM2	K562	3	0.01	1136	1136
SRSF1	K562	3	0.01	1049	1049
YBX3	K562	3	0.01	4370	4370

Table 4: Performance results for 6 RBP eCLIP sets with genomic and transcript context. We report average accuracies obtained by 10-fold cross validation together with standard deviations (apart from GraphProt).

Methods	RBP	Cell line	Genomic context	Transcript context
DeepBind	FMR1	K562	80.63±1.58	88.22±1.99
	FXR2	K562	76.93±2.66	86.93±1.18
	IGF2BP1	K562	75.72±2.59	83.90±2.08
	PUM2	K562	70.05±2.94	80.69±2.31
	SRSF1	K562	79.39±4.64	85.98±3.07
	YBX3	K562	76.63±2.73	87.32±1.24
GraphProt	FMR1	K562	78.47	88.50
	FXR2	K562	75.71	86.73
	IGF2BP1	K562	66.24	84.18
	PUM2	K562	64.88	79.58
	SRSF1	K562	76.41	86.61
	YBX3	K562	71.63	86.61
GraphProt2	FMR1	K562	82.66±1.68	92.23±1.11
	FXR2	K562	81.51±1.57	91.09±0.84
	IGF2BP1	K562	75.58±1.81	88.54±1.83
	PUM2	K562	73.86±1.95	86.27±2.76
	SRSF1	K562	80.93±3.01	91.09±2.54
	YBX3	K562	79.22±1.20	90.86±0.53

Table 5: Motif search results for 9 RBPs and 28 binding motifs collected from various sources (see Additional File 3 in .xlsx format).

Supplementary figures

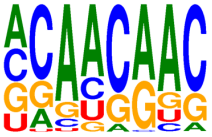
RBP	Literature knowledge	GraphProt sequence logo	
		Genomic context	Transcript context
FMR1	GAC-containing sites Similar to FXR1 and FXR2		
FXR2	GAC-containing sites Similar to FMR1 and FXR1		
IGF2BP1	CA(U)-rich sites*		
PUM2	UGUANAUA dominant motif		
SRSF1	GA(C)-containing sites		
YBX3	CA(U)-containing sites		

Figure 1: GraphProt sequence logos generated from models trained on the 6 eCLIP sets with genomic and transcript context (resulting in 12 models and 12 logos). Literature knowledge regarding RBP binding preferences was obtained from the ATtRACT database [1]. A logo is constructed for each RBP-context combination from the top 200 scoring sites (taking highest scoring 8-mer sequence for each site) of the positive set. *: note that IGF2BP1 binding sites are comprised of several parts, of which one dominant part are CA(U) rich sites.

.

References

[1] Giudice, G., Sánchez-Cabo, F., Torroja, C., Lara-Pezzi, E.: ATtRACT - a database of RNA-binding proteins and associated motifs. Database **2016** (2016)

[P4] Table S5: Motif search results for 9 RBPs and 28 binding motifs collected from various literature. IDR peak regions were processed as described in the Methods section to obtain sites with transcript and genomic context.

RBP	Dataset	Motif	Pubmed_ID	c_IDR_sites_near_exb	c_gen_hits	gen_hits_1000nt	c_tr_hits	tr_hits_1000nt	tr_gen_ratio
FMR1	K562	[GU]GACA[GA]G	23846655	747	55	0.45732	56	0.47025	1.02828
FMR1	K562	[AU]GGA	23235829	747	1630	13.55318	1991	16.71901	1.23359
FMR1	K562	ACU[GU]	23235829	747	855	7.10918	826	6.93616	0.97566
FXR1	K562	A[CU]GAC[AG]	23846655	348	28	0.49975	44	0.78532	1.57143
FXR2	K562	[GAU]GAC[AG][AG][AG]	23846655	1028	298	1.80052	340	2.07118	1.15032
FXR2	HepG2	[GAU]GAC[AG][AG][AG]	23846655	1064	312	1.82132	414	2.41697	1.32704
IGF2BP1	K562	CGGAC.{10,25}[CA]CA[CU]	22215810	535	2	0.02322	6	0.0701	3.01896
IGF2BP1	K562	[CA]CA[CU].{10,25}CGGAC	22215810	535	1	0.01161	3	0.03505	3.01896
IGF2BP2	K562	[GCA][AC]A[AUC][AU]CA	23846655	426	172	2.5078	217	3.19828	1.27533
IGF2BP3	HepG2	GGC.{15,25}CA.{7,20}CA.{15,25}GGC.{2,8}[CA]{4}	31118463	155	1	0.04007	7	0.28428	7.0941
IGF2BP3	HepG2	A[AC]A[ACU][AU]CA	23846655	155	28	1.12202	27	1.09649	0.97725
PUM2	K562	UGUA[ACGU]AUA	18776931	285	89	1.93963	107	2.4729	1.27493
SRSF1	K562	UCAGAGGA	26431027	342	2	0.03632	7	0.12787	3.52033
SRSF1	K562	G[GA]AGGA	23846655	342	199	3.61411	254	4.63978	1.2838
SRSF1	K562	GGAGGA	23846655	342	139	2.52443	181	3.3063	1.30972
SRSF1	K562	GGA[GC]G[GA][GCA]	23846655	342	183	3.32353	218	3.98217	1.19818
SRSF1	K562	AGGA[GC][AC]	23846655	342	209	3.79572	248	4.53018	1.1935
SRSF1	K562	GG[GA]GGA[GCA]	23846655	342	128	2.32465	157	2.86789	1.23369
SRSF1	HepG2	UCAGAGGA	26431027	319	3	0.05841	4	0.07911	1.3543
SRSF1	HepG2	G[GA]AGGA	23846655	319	172	3.34897	224	4.43003	1.3228
SRSF1	HepG2	GGAGGA	23846655	319	130	2.5312	164	3.24341	1.28137
SRSF1	HepG2	GGA[GC]G[GA][GCA]	23846655	319	200	3.89416	218	4.31137	1.10714
SRSF1	HepG2	AGGA[GC][AC]	23846655	319	198	3.85522	247	4.8849	1.26709
SRSF1	HepG2	GG[GA]GGA[GCA]	23846655	319	139	2.70644	155	3.06542	1.13264
YBX3*	K562	AACAUC[GAU]	23846655	3340	164	0.30498	214	0.39946	1.30978
YBX3*	K562	AACAUC	23846655	3340	215	0.39982	283	0.52825	1.32122
YBX3**	K562	AACA[AU]C[GAU]	23846655	3340	271	0.50396	357	0.66638	1.32229
YBX3***	K562	[ACU][AC]CA[CU]C[ACU]	11564883	3340	1901	3.53517	2085	3.8919	1.10091

[P4] Table S5 legend:

RBP	RBP name
Dataset	Dataset (K562 or HepG2 cell type)
Motif	Motif or regular expression found in the literature for the respective RBP
Pubmed ID	Literature reference (Pubmed ID) from which the motif information was obtained from
c_IDR_sites_near_exb	Number of IDR peak regions near exon borders (<= 10 nt away) used for motif search
c_gen_hits	Number of motif or regular expression hits in the c_IDR_sites_near_exb sites with genomic context. NOTE that motif positions in the genome that were counted > 1 were reduced to count = 1
gen_hits_1000nt	c_gen_hits normalized (hits per 1000 nt). Necessary since extracted transcript regions are sometimes truncated at transcript ends
c_tr_hits	Number of motif or regular expression hits in the c_IDR_sites_near_exb sites with transcript context. NOTE that motif positions on transcripts that were counted > 1 were reduced to count = 1
tr_hits_1000nt	c_tr_hits normalized (hits per 1000 nt). Necessary since extracted transcript regions are sometimes truncated at transcript ends
tr_gen_ratio	tr_hits_1000nt / gen_hits_1000nt ratio

[P5] Peakhood: individual site context extraction for CLIP-seq peak regions

Supplementary material for publication:

- [P5] **Michael Uhl**, Dominik Rabsch, Florian Eggenhofer, and Rolf Backofen. **Peakhood: individual site context extraction for CLIP-seq peak regions**. *Bioinformatics*, 2021.

Peakhood: individual site context extraction for CLIP-seq peak regions Supplementary Material

Michael Uhl, Dominik Rabsch, Florian Eggenhofer, and Rolf Backofen

October 28, 2021

1 Supplementary methods

1.1 Data availability

The transcript context site collections generated by `Peakhood` from eCLIP datasets of 49 RBPs (first collection with 36 RBPs from HepG2, second collection with 40 RBPs from K562) with known roles in post-transcriptional gene regulation (mRNA stability and decay, translational regulation; information taken from [1] Supplementary Data 1 table) can be downloaded from Zenodo [2].

1.2 How Peakhood works

Here we briefly describe how `Peakhood` works. For full details, please check out `Peakhood`'s comprehensive online manual at: https://github.com/BackofenLab/Peakhood

Site context extraction

To extract individual site context information for a CLIP-seq dataset, `Peakhood`'s input consists of the genomic CLIP-seq peak regions (BED), the mapped CLIP-seq reads (BAM), a genomic annotations file (GTF), and a genome sequence file (.2bit). `Peakhood` first intersects the peak regions with transcript and exon regions from the GTF file, to obtain exonic, intronic, and intergenic sites. Next it determines for each exonic site whether it is more likely embedded in a genomic context (introns included) or transcript context (mature or spliced RNA). For this `Peakhood` makes use of the exon-intron read coverage ratios in the site neighborhood, as well as over the whole transcript. This is based on the observation that an exonic site inside a transcript (spliced) context (Paper Fig. 1a) usually features considerably more reads in the exon region(s), as well as a pronounced coverage drop-off at the exon borders. Ideally this is true both locally (around the overlapping exon) and globally (on the whole transcript). However, due to how the CLIP-seq protocol works, read coverage is often concentrated at and around the binding site, so `Peakhood` weighs the local context information higher than the global one. In addition, intron-spanning reads receive more weight than continuously mapped reads, since they provide strong support for a transcript context. Sites which feature sufficiently high local

and global ratios (see online manual for more details on filter steps and thresholds) get assigned to transcript context. Exonic sites with lower ratios get assigned to genomic context (Fig. S1).

Choosing the most likely transcript

Since a gene usually has many transcript isoforms, and there can be several overlapping exons and transcripts which pass the filters, a transcript context site can have several possible site-transcript combinations. `Peakhood` thus also selects the most likely combination, based on number of informative filters: co-occurrence of other sites, read coverage, intron-spanning read numbers, and transcript support level. Filter order, choice of filters, and filter behavior (serial or majority vote) can be further customized. In addition, sites at exon borders connected by intron-spanning reads get merged into single sites (see Paper Fig. 1a example). Reference and custom transcript annotations are supported, which can be advantageous if created for the same cell types or conditions (see online manual on how to create custom annotations). Incorporation of RNA-seq data is also possible, to provide additional intron-spanning read information for transcript selection.

Merging transcript context sets

In addition to extracting transcript context sites for single CLIP-seq datasets, `Peakhood` can merge any number of transcript context sets into comprehensive transcript context site collections (see Paper Fig. 1b for general workflow). Output table files contain information on transcripts and overlapping sites, both for all and the most likely site-transcript combinations. Site pairs on transcripts and their genomic and transcript distances are also reported. This way, one e.g. can quickly filter for and spot interesting site pairs (same or different RBP), where the transcript site distance is lower than the original genomic distance.

1.3 Agreement with known RBP roles

When `Peakhood` performs the site context extraction, it reports (among other statistics) three informative percentages: the percentage of exonic sites (divided by all sites), the percentage of transcript context sites (divided by all exonic sites), and the percentage of exon border sites (divided by all transcript context sites). Fig. S2 shows these percentages for four eCLIP datasets from four different RBPs, obtained by running site context extraction (`peakhood extract`) with default parameters. We can see that for typical spliced RNA binding RBPs (IGF2BP1, PUM1, PUM2), most sites overlap with exons (>= 95%), and out of these >= 95% are assigned to transcript context. In contrast, for the splicing factor U2AF2 we get around 20% of exonic sites, and out of these only 5.9% get assigned to transcript context. This shows that `Peakhood`'s transcript context selection agrees with known RBP roles. We also see that the number of exon border sites can be quite substantial, as in the case of PUM1 (around 25%). Such sites at exon borders connected by intron-spanning reads need to be merged, and not taken as separate binding events (see Paper Fig. 1). This again showcases the importance of a proper site context selection as done by `Peakhood`.

1.4 Displaying genomic regions

To display the genomic regions in Figure 1a and Figure S1, BAM and IDR peak files from ENCODE were downloaded (dataset IDs ENCSR661ICQ (PUM2) and ENCSR893RAV (U2AF2)) and pre-processed as follows:

```
wget https://www.encodeproject.org/files/ENCFF880MWQ/@@download/ENCFF880MWQ.bed.gz
gunzip -c ENCFF880MWQ.bed.gz | awk '{print $1"\t"$2"\t"$3"\t"$4"\t"$7"\t"$6}' > PUM2_K562_IDR_peaks.bed
wget -O PUM2_K562_rep1.bam https://www.encodeproject.org/files/ENCFF231WHF/@@download/ENCFF231WHF.bam
wget -O PUM2_K562_rep2.bam https://www.encodeproject.org/files/ENCFF732EQX/@@download/ENCFF732EQX.bam
samtools merge -f PUM2_K562_rep12.bam PUM2_K562_rep1.bam PUM2_K562_rep2.bam
samtools view -hb -f 130 PUM2_K562_rep12.bam -o PUM2_K562_rep12.R2.bam
wget https://www.encodeproject.org/files/ENCFF290DFO/@@download/ENCFF290DFO.bed.gz
gunzip -c ENCFF290DFO.bed.gz | awk '{print $1"\t"$2"\t"$3"\t"$4"\t"$7"\t"$6}' > U2AF2_K562_IDR_peaks.bed
wget -O U2AF2_K562_rep1.bam https://www.encodeproject.org/files/ENCFF835KXL/@@download/ENCFF835KXL.bam
wget -O U2AF2_K562_rep2.bam https://www.encodeproject.org/files/ENCFF936JSP/@@download/ENCFF936JSP.bam
samtools merge -f U2AF2_K562_rep12.bam U2AF2_K562_rep1.bam U2AF2_K562_rep2.bam
samtools view -hb -f 130 U2AF2_K562_rep12.bam -o U2AF2_K562_rep12.R2.bam
```

The merged R2 read BAM and peak region BED files were then loaded up into IGV (Integrative Genome Viewer) for visualization [link].

1.5 Site context extraction percentages

To get the percentages of Figure S2, we used a custom GTF file generated as described in the online manual (see "Documentation" section, subsection "Custom GTF files", total RNA-seq ENCODE dataset ID: ENCSR792OIJ). The created custom GTF file for the K562 cell line can be downloaded from Zenodo [2]. The BAM and BED files were downloaded and pre-processed as described in the below section "Runtime measurement", encompassing the following four ENCODE eCLIP datasets: ENCSR975KIR (IGF2BP1), ENCSR308YNT (PUM1), ENCSR661ICQ (PUM2), and ENCSR893RAV (U2AF2). Using the custom GTF file, site context extraction was evoked to get the percentages from Figure S2 by the following commands:

```
peakhood extract --in IGF2BP1_K562_IDR_peaks.uniq_ids.bed --bam IGF2BP1_K562_rep12.R2.bam
--gtf K562_total_rnaseq_rep12_stringtie_gffcompare.gtf --gen hg38.2bit --out IGF2BP1_K562_IDR_pm_extract_out
--pre-merge --new-site-id IGF2BP1_K562_IDR
peakhood extract --in PUM1_K562_IDR_peaks.uniq_ids.bed --bam PUM1_K562_rep12.R2.bam
--gtf K562_total_rnaseq_rep12_stringtie_gffcompare.gtf --gen hg38.2bit --out PUM1_K562_IDR_pm_extract_out
--pre-merge --new-site-id PUM1_K562_IDR
peakhood extract --in PUM2_K562_IDR_peaks.uniq_ids.bed --bam PUM2_K562_rep12.R2.bam
--gtf K562_total_rnaseq_rep12_stringtie_gffcompare.gtf --gen hg38.2bit --out PUM2_K562_IDR_pm_extract_out
--pre-merge --new-site-id PUM2_K562_IDR
peakhood extract --in IGF2BP1_K562_IDR_peaks.uniq_ids.bed --bam U2AF2_K562_rep12.R2.bam
--gtf K562_total_rnaseq_rep12_stringtie_gffcompare.gtf --gen hg38.2bit --out U2AF2_K562_IDR_pm_extract_out
--pre-merge --new-site-id U2AF2_K562_IDR
```

1.6 Runtime measurement

For the runtime measurement (inside conda environment with `Peakhood` installed), we downloaded and pre-processed the eCLIP PUM1 data (K562 cell line) as described in `Peakhood`'s online manual on GitHub [3].

```
wget https://www.encodeproject.org/files/ENCFF094MQV/@@download/ENCFF094MQV.bed.gz
gunzip -c ENCFF094MQV.bed.gz | awk '{print $1"\t"$2"\t"$3"\t"$4"\t"$7"\t"$6}' > PUM1_K562_IDR_peaks.bed
bed_generate_unique_ids.py  --in PUM1_K562_IDR_peaks.bed --id PUM1_K562_IDR > PUM1_K562_IDR_peaks.uniq_ids.bed
wget -O PUM1_K562_rep1.bam https://www.encodeproject.org/files/ENCFF064COB/@@download/ENCFF064COB.bam
```

```
wget -O PUM1_K562_rep2.bam https://www.encodeproject.org/files/ENCFF583QFB/@@download/ENCFF583QFB.bam
wget https://hgdownload.cse.ucsc.edu/goldenpath/hg38/bigZips/hg38.2bit
wget http://ftp.ensembl.org/pub/release-103/gtf/homo_sapiens/Homo_sapiens.GRCh38.103.gtf.gz
```

Now we can run the site context extraction on the dataset:

```
peakhood extract --in PUM1_K562_IDR_peaks.uniq_ids.bed --bam PUM1_K562_rep1.bam PUM1_K562_rep2.bam
--bam-pp-mode 2 --gtf Homo_sapiens.GRCh38.103.gtf.gz --gen hg38.2bit --report
--out PUM1_K562_IDR_extract_out

peakhood extract --in PUM1_K562_IDR_peaks.uniq_ids.bed --bam PUM1_K562_rep1.bam PUM1_K562_rep2.bam
--bam-pp-mode 2 --gtf Homo_sapiens.GRCh38.103.gtf.gz --gen hg38.2bit --out PUM1_K562_IDR_pm_extract_out
--pre-merge --new-site-id PUM1_K562_IDR
```

On our test machine (Intel i7-8700k, 32 GB RAM, Ubuntu 18.04 LTS), this takes about 2 minutes and 30 seconds. In case the dataset is used more than once, we recommend to pre-merge the BAM files, as well as filter by R2 reads (in case of eCLIP data, as described in online manual), to speed up the run:

```
samtools merge -f PUM1_K562_rep12.bam PUM1_K562_rep1.bam PUM1_K562_rep2.bam
samtools view -hb -f 130 PUM1_K562_rep12.bam -o PUM1_K562_rep12.R2.bam
```

Running this again shortens to extraction time to about 1 minute and 30 seconds:

```
peakhood extract --in PUM1_K562_IDR_peaks.uniq_ids.bed --bam PUM1_K562_rep12.R2.bam
--gtf Homo_sapiens.GRCh38.103.gtf.gz --gen hg38.2bit --out PUM1_K562_IDR_pm_extract_out
--pre-merge --new-site-id PUM1_K562_IDR
```

Supplementary figures

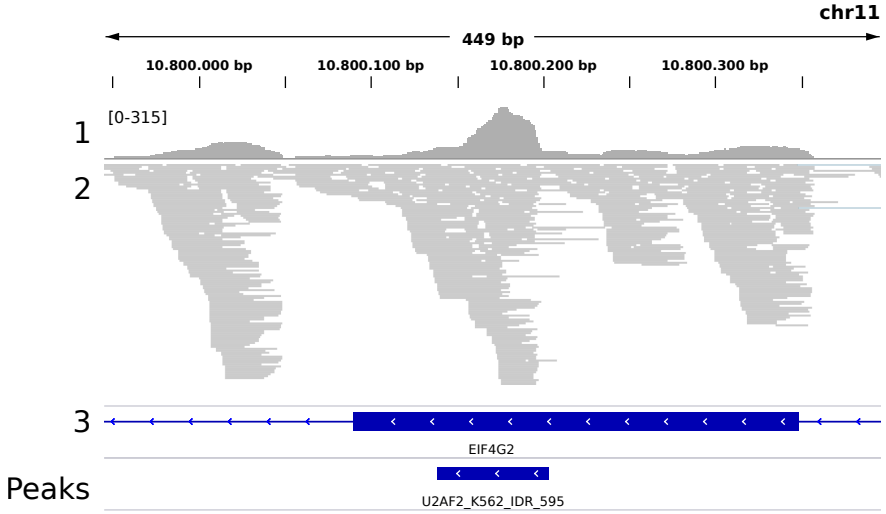


Figure 1: Genomic region (IGV snapshot) with mapped U2AF2 eCLIP data. 1: read profile (coverage range in brackets), 2: read alignments, 3: gene annotations (thick blue regions are exons, thin blue regions introns), Peaks: peaks called by CLIPper (IDR method). Example region for the splicing factor U2AF2, with higher read counts over exon borders and introns.

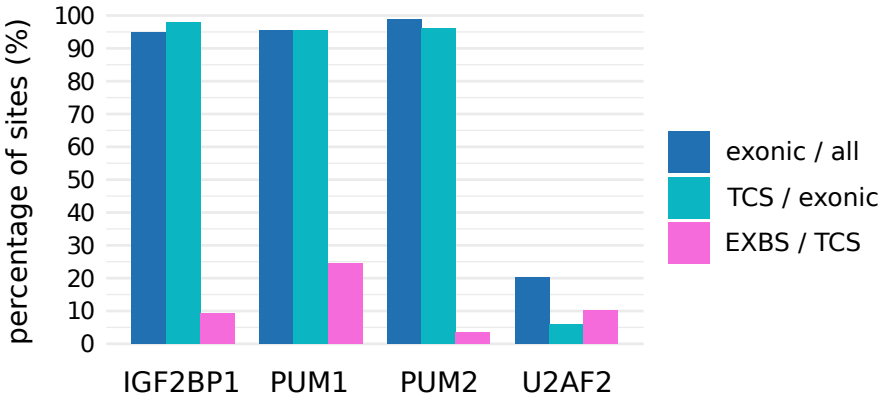


Figure 2: `Peakhood` site extraction results for four eCLIP datasets (K562 cell line, sites from CLIPper IDR) and four RBPs (number of all sites in brackets): IGF2BP1 (4776), PUM1 (2146), PUM2 (4578), and U2AF2 (3250). The plot shows percentages of exonic sites (exonic sites divided by all sites), transcript context sites (TCS) (TCS divided by all exonic sites), and exon border sites connected by intron-spanning reads (EXBS) (EXBS divided by TCS).

References

[1] Eric L Van Nostrand, Peter Freese, Gabriel A Pratt, Xiaofeng Wang, Xintao Wei, Rui Xiao, Steven M Blue, Jia-Yu Chen, Neal AL Cody, Daniel Dominguez, et al. A large-scale binding and functional map of human RNA-binding proteins. *Nature*, 583(7818):711–719, 2020.

[2] Michael Uhl. Peakhood: individual site context extraction for CLIP-seq peak regions. *Zenodo*, 2021. https://doi.org/10.5281/zenodo.5557101.

[3] Michael Uhl. Peakhood: individual site context extraction for CLIP-seq peak regions. *GitHub repository*, 2021. https://github.com/BackofenLab/Peakhood.