Analysis of high-throughput sequencing data related to small non-coding RNAs biogenesis and function

Dissertation zur Erlangung des Doktorgrades der
Technischen Fakultät der
Albert-Ludwigs-Universität Freiburg im Breisgau

vorgelegt von
Pavankumar Videm

Dekan:	Prof. Dr. Rolf Backofen
Erstgutachter:	Prof. Dr. Rolf Backofen
Zweitgutachter:	Prof. Dr. Wolfgang R. Hess
Vorsitz:	Prof. Dr. Gerald Urban
Beisitz:	Prof. Dr. Fabian Kuhn
Datum der Promotion:	23.07.2021

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisor Prof. Dr. Rolf Backofen for granting me the opportunity to work on exciting research topics and for his tremendous support throughout my Ph.D. career. Thank you Rolf for inspiring me by sharing your knowledge and expertise in RNA bioinformatics. I am highly grateful to Prof. Dr. Wolfgang R. Hess for his time and kindness to review my thesis. I cordially thank Prof. Dr. Gerald Urban and Prof. Dr. Fabian Kuhn for being part of my Ph.D. examination committee.

I would like to extend my sincere thanks to all my collaborators from Prof. Dr. Tanja Vogel's lab for the successful collaborations that made this thesis possible. My special thanks to Fabrizio Costa and Dominic Rose for their great support and encouragement during the early stages of my research career.

I can never enough appreciate my mentor and good friend Björn Grüning's outstanding support throughout my Ph.D. life. It has been a great pleasure working with him. I am deeply indebted to Steffen Heyne and Prof. Dr. Rolf Backofen for offering me the first-ever opportunity to work in the field of bioinformatics.

I have been fortunate to have friendly and helpful lab members. I take this opportunity to thank my fellow Ph.D. students and Postdocs, in particular Omer Alkhnbashi, Florian Eggenhofer, Torsten Houwaart, Milad Miladi, Teresa Müller, Michael Uhl and Patrick Wright for their companionship. I owe my gratitude to Simon Bray, Björn Grüning, Teresa Müller, Martin Raden, Mehmet Tekman and Michael Uhl for proofreading this thesis and their valuable comments. I would also like to thank Monika Degen-Hellmuth for her kind support with the administration works.

I would like to pay special regards to my parents for providing unconditional love and support at every stage of my life. Last but not least, I thank with love my wife and daughter for offering care and happiness with their beautiful smiles during the hardest times of my thesis.

Contents

Abstract

Genes are the functional regions of DNA that are transcribed into RNA. RNA can either encode for proteins or is on-itself a functional end-product. The RNAs that act as templates for protein biosynthesis are called protein-coding RNAs, whereas the remaining are called non-coding RNAs (ncRNAs). Surprisingly, the majority of transcribed RNA loci are actually non-coding, and only a minority encode proteins*. Although they do not code for proteins, they play key roles in diverse cellular processes such as transcription, splicing, translation and gene regulation. Moreover, they are implicated in many diseases. This thesis focuses on the analysis of ncRNAs that are short in length, called small ncRNAs. The overall contributions in this thesis cover three research problems related to small ncRNAs: (i) annotation of functional small ncRNAs (ii) detection and analysis of small ncRNA interactions and (iii) improving the accessibility of computational tools used in small ncRNA research.

Conventionally, RNAs are annotated using sequence and structure conservation. This thesis proposes a computational method called `BlockClust` that uses neither the sequence nor the structure of ncRNAs but the processing patterns occurring in their biogenesis. The functions of ncRNAs are often closely associated with their biogenesis. The patterns in the processing of small ncRNAs can be observed by aligning the RNA sequencing data to the reference genome. `BlockClust` encodes these patterns into graphs and uses a graph kernel for clustering and classification. With quasi-linear time complexity of the graph kernel, `BlockClust` gained a 60-fold speedup, while improving the clustering and classification performance compared to other approaches. With a consistent performance over different organisms, a variety of tissues and cell lines, `BlockClust` has proved to be a robust and bias-free approach.

There are two use-cases of the ncRNA interaction analysis that are presented in this thesis. The first use-case analyzes the role of a microRNA family (*miR200*) in a neurological disorder. It involves differential gene expression analysis from RNA sequencing data of Forkhead box G1 (*FOXG1*) gene knockout and *miR200* overexpression experiments. The study revealed an important pathway that causes Rett syndrome by *FOXG1* knockout. The results indicate that *FOXG1* affects the biogenesis of the *miR200* family whose target is the protein kinase type II-beta regulatory subunit (*PRKAR2B*). As a result, *miR200* upregulates *PRKAR2B*, which plays an important role in memory formation. The imbalance in its expression level may contribute to atypical Rett syndrome. The second use case offers a computational framework, `ChiRA`, for the analysis of genome-wide ncRNA interactions from RNA-RNA interactome experiments. These experiments generate chimeric sequences, each of which is a fusion of two interacting RNA sequences. Because of the short lengths, these

*When it comes to the number of transcribed RNA molecules, however, the majority does code for proteins.

sequences are often mapped to multiple reference locations, causing ambiguity in annotating the sequences. `ChiRA` deals with two important challenges in the data analysis, namely handling of multi-mapped sequences and accurately annotating them by quantification. It has been shown that `ChiRA` can identify the sequences that are multi-mapped to paralogous genes or gene families without requiring any information on gene relations. It is also an effective and sensitive approach that can detect new RNA-RNA interactions from published RNA-RNA interactome datasets.

The final objective of the thesis is to ensure the accessibility of the above-mentioned tools, analyses and long-term sustainability of small ncRNA research. RNA workbench, a Galaxy based framework for RNA-centric research was developed to achieve this goal. The RNA workbench comes in two flavors. A Docker-based RNA workbench can easily be deployed on a custom hardware infrastructure or even on a personal computer. A web-based alternative served on European Galaxy infrastructure has access to vast computational resources and is open to all users. The RNA workbench provides various workflows and hands-on tutorials for the analysis of small ncRNAs. Being part of the RNA workbench, all the computational tools and workflows developed during this thesis consequently feature long-term maintenance and support from the RNA community.

Zusammenfassung

Gene sind die funktionellen Bereiche der DNA, die in RNA umgeschrieben werden. RNA kann entweder für Proteine kodieren oder kann als selbstständiges funktionelles Endprodukt fungieren. Die RNAs, die als Vorlage für die Proteinbiosynthese dienen, werden als proteinkodierende RNAs bezeichnet, während die übrigen als nicht-kodierende RNAs (ncRNAs) bezeichnet werden. Überraschenderweise ist die Mehrheit der transkribierten RNA-Loci tatsächlich nicht-kodierend, und nur eine Minderheit ist Protein kodierend[†]. Obwohl ncRNAs nicht für Proteine kodieren, spielen sie Schlüsselrollen in verschiedenen zellulären Prozessen wie Transkription, Spleißen, Translation und Genregulation. Außerdem spielen sie eine entscheidende Rolle in vielen Krankheiten. Diese Arbeit konzentriert sich auf die Analyse einer speziellen Art von ncRNAs, die aufgrund ihre Länge kleine ncRNAs genannt werden. Die vorliegende Arbeit befasst sich mit drei Forschungsschwerpunkten im Zusammenhang mit kleinen ncRNAs: (i) Annotation von funktionalen kleinen ncRNAs, (ii) Detektion und Analyse von kleinen ncRNA-Interaktionen und (iii) Verbesserung der Zugänglichkeit von computergestützten Werkzeugen, die in der Erforschung von kleinen ncRNA verwendet werden.

Konventionell werden RNAs anhand von Sequenz- und Strukturerhaltung annotiert. In dieser Arbeit wird eine Berechnungsmethode namens `BlockClust` vorgeschlagen, die weder die Sequenz noch die Struktur von ncRNAs verwendet, sondern die Verarbeitungsmuster, die bei ihrer Biogenese auftreten. Die Funktionen von ncRNAs sind oft eng mit ihrer Biogenese verbunden. Die Muster in der Prozessierung von kleinen ncRNAs können durch Alignments der RNA-Sequenzierungsdaten an das Referenzgenom beobachtet werden. `BlockClust` kodiert diese Muster in Graphen und verwendet einen Graph-Kernel für das Clustering und die Klassifikation der ncRNAs. Durch die quasi-lineare Zeitkomplexität des Graph-Kernels erreicht `BlockClust` eine 60-fach schnellere Laufzeit und verbesserte zudem das Clustering und die Klassifizierung der ncRNAs im Vergleich zu bestehenden Ansätzen. Mit einer konsistenten Leistung über verschiedene Organismen, eine Vielzahl von Geweben und Zelllinien hat sich `BlockClust` als ein robuster und bias-freier Ansatz erwiesen.

Es gibt zwei Anwendungsfälle der ncRNA-Interaktionsanalyse, die in dieser Arbeit vorgestellt werden. Der erste Anwendungsfall befasst sich mit der Analyse einer microRNA-Familie (*miR200*) und deren Rolle in einer neurologischen Störung. Es handelt sich hierbei um eine differenzielle Genexpressionsanalyse von Forkhead Box G1 (*FOXG1*) Gen-Knockout- und *miR200*-Überexpressions-Experimenten. Die Arbeit entdeckte einen wichtigen Signalweg, der das Rett-Syndrom durch *FOXG1*-Knockout verursacht. Die Ergebnisse zeigen, dass *FOXG1* die Biogenese der *miR200*-Familie beeinflusst, deren Ziel die regulatorische Untereinheit der Proteinkinase Typ II-beta (*PRKAR2B*) ist. Infolgedessen wird *PRKAR2B*

[†]Wenn es um die Anzahl der transkribierten RNA-Moleküle geht, kodiert die Mehrheit jedoch für Proteine.

durch *miR200* hochreguliert, welches eine wichtige Rolle bei der Gedächtnisbildung spielt. Ein Expressionsungleichgewicht könnte zum atypischen Rett-Syndrom beitragen. Der zweite Anwendungsfall beschreibt ein computergestütztes Framework, `ChiRA`, für die Analyse von genomweiten ncRNA-Interaktionen aus RNA-RNA-Interaktomexperimenten. Diese Experimente erzeugen chimäre Sequenzen, die jeweils eine Fusion zweier interagierender RNA-Sequenzen darstellen. Aufgrund der kurzen Längen werden diese Sequenzen oft auf mehrere Referenzstellen gemappt, was zu Mehrdeutigkeit bei der Annotation der Sequenzen führt. `ChiRA` befasst sich mit zwei wichtigen Herausforderungen bei der Datenanalyse, nämlich dem Umgang mit mehrfach gemappten Sequenzen und deren genauer Annotation durch Quantifizierung. Es konnte gezeigt werden, dass `ChiRA` Sequenzen identifizieren kann, die gegen paraloge Gene oder Genfamilien mehrfach gemappt wurden, ohne dass Informationen über die Genbeziehungen erforderlich ist. `ChiRA` ist darüber hinaus ein effektiver und sensitiver Ansatz für die Identifizierung neuer RNA-RNA-Interaktionen aus bereits veröffentlichten RNA-RNA-Interaktomdatensätzen.

Ein weiteres Ziel dieser Arbeit ist es, die Zugänglichkeit der oben genannten Werkzeuge und Analysen sowie die langfristige Nachhaltigkeit der Forschung an kleinen ncRNAs zu gewährleisten. Um dieses Ziel zu erreichen, wurde die RNA-Workbench, ein Galaxy-basiertes Framework zur computergestützten Erforschung von RNAs, entwickelt. Die RNA-Workbench gibt es in zwei Ausprägungen. Die Docker-basierte RNA-Workbench kann auf einer kundenspezifischen Hardware-Infrastruktur oder auf einem Personal-Computer eingesetzt werden. Eine webbasierte Alternative, basierend auf dem europäischen Galaxy Server, stellt umfangreiche Rechenressourcen für alle Benutzer zur Verfügung. Die RNA-Workbench bietet verschiedene Workflows und praktische Tutorials für die Analyse von kleinen ncRNAs an. Als Teil der RNA-Workbench werden alle in dieser Arbeit entwickelten Methoden und Workflows von der RNA-Community langfristig gepflegt und unterstützt.

List of publications

This thesis is based on the following publications:

- [*P*1] **Pavankumar Videm**, Dominic Rose, Fabrizio Costa, and Rolf Backofen. BlockClust: efficient clustering and classification of non-coding RNAs from short read RNA-seq profiles. *ISMB 2014 proceedings'* special issue in *Bioinformatics*, 2014.

- [*P*2] Stefan C Weise*, Ganeshkumar Arumugam*, Alejandro Villarreal*, **Pavankumar Videm***, Stefanie Heidrich, Nils Nebel, Verónica I Dumit, Farahnaz Sananbenesi, Viktoria Reimann, Madeline Craske, et al. FOXG1 regulates PRKAR2B transcriptionally and posttranscriptionally via miR200 in the adult hippocampus. *Molecular neurobiology*, 2019.

- [*P*3] **Pavankumar Videm**, Anup Kumar, Oleg Zharkov, Björn A. Grüning, and Rolf Backofen. ChiRA: an integrated framework for chimeric read analysis from RNA-RNA interactome and RNA structurome data. *GigaScience*, 2021.

- [*P*4] Jörg Fallmann, **Pavankumar Videm**, Andrea Bagnacani, Bérénice Batut, Maria A Doyle, Tomas Klingstrom, Florian Eggenhofer, Peter F Stadler, Rolf Backofen, and Björn Grüning. The RNA workbench 2.0: next generation RNA data analysis. *Nucleic Acids Research*, 2019.

Further publications:

1. **Pavankumar Videm***, Deepika Gunasekaran*, Bernd Schröder, Bettina Mayer, Martin L Biniossek, and Oliver Schilling. Automated peptide mapping and protein-topographical annotation of proteomics data. *BMC bioinformatics*, 2014.

2. Tomasz Chelmicki, Friederike Dündar, Matthew James Turley, Tasneem Khanam, Tugce Aktas, Fidel Ramírez, Anne-Valerie Gendrel, Patrick Rudolf Wright, **Pavankumar Videm**, Rolf Backofen, et al. MOF-associated complexes ensure stem cell identity and Xist repression. *Elife*, 2014.

3. Björn A Grüning, Jörg Fallmann, Dilmurat Yusuf, Sebastian Will, Anika Erxleben, Florian Eggenhofer, Torsten Houwaart, Bérénice Batut, **Pavankumar Videm**, Andrea Bagnacani, et al. The RNA workbench: best practices for RNA and high-throughput sequencing bioinformatics in Galaxy. *Nucleic acids research*, 2017.

4. Bérénice Batut, Saskia Hiltemann, Andrea Bagnacani, Dannon Baker, Vivek Bhardwaj, Clemens Blank, Anthony Bretaudeau, Loraine Brillet-Guéguen, Martin Čech, John

[†]* Joint first authors

Chilton, et al. Community-driven data analysis training for biology. *Cell systems*, 2018.

5. Henriette Franz, Alejandro Villarreal, Stefanie Heidrich, **Pavankumar Videm**, Fabian Kilpert, Ivan Mestres, Federico Calegari, Rolf Backofen, Thomas Manke, and Tanja Vogel. DOT1L promotes progenitor proliferation and primes neuronal layer identity in the developing cerebral cortex. *Nucleic acids research*, 2019.

Part I

Introduction

Motivation

The genetic material of most of the life forms on planet Earth is composed of deoxyribonucleic acid (DNA). It contains hereditary information and also encodes for the functional biomolecules called proteins. According to the central dogma of molecular biology [1], the production of proteins from DNA involves another intermediate macromolecule called ribonucleic acid (RNA). DNA is first transcribed into RNAs that are then translated into proteins. However, RNAs are not merely a template for protein biosynthesis. There are also a class of RNA molecules (non-coding RNAs abbreviated as ncRNAs) that do not produce proteins but play important roles in diverse biological processes. In several viruses, RNA is the initial genetic material instead of DNA. RNAs are versatile molecules that like DNA can store genetic information, but like proteins can also catalyze biochemical reactions. These key findings led to the exploration of more RNA functions and also its role in the origin of life. The "RNA world" [2] is such a hypothesis that posits that self-replicating RNA is the origin of life. It theorizes that DNA is the result of evolution favoring stable storage of genetic material.

Projects like ENCODE [3] revealed that only 2% of the human RNAs produce proteins and the rest are non-coding. Based on length, ncRNAs are categorized into either small or long ncRNAs. Different classes of small ncRNAs such as micro RNAs, transfer RNAs, or small nucleolar RNAs are involved in diverse cellular processes and are also implicated in several diseases [4]. Small ncRNAs typically interact with other macromolecules to exert their specific functions. The regulatory mechanism of microRNA by interaction with its target protein-coding RNA is a prime example.

Traditionally, experiments in molecular biology are carried out at a small scale, often at a single-molecule level. The advent of DNA sequencing revolutionized experimental molecular biology. The Human Genome Project [5] was an exceptional feat performed at a grand scale, aimed at deciphering complete human genome of 3 billion base pairs. It used Sanger DNA sequencing, which though highly accurate, has limited throughput. It took about 10 years to conceptualize, sequence, assemble and annotate the first draft of the human genome and was estimated to cost ~$3 billion Dollars. Thanks to modern high-throughput, massively parallel, next-generation sequencing techniques, a decade after the release of the first human genome draft, we can today sequence a similar-sized genome in the span of a day for less than $2000. Apart from DNA sequencing, these techniques have also been adapted to RNA, allowing scientists to conduct genome-wide experiments to study RNAs and their interactions. With the aid of RNA sequencing, we can quantify the gene expression, assemble complete transcriptomes, detect novel RNAs, or identify the interactions of RNAs with other RNAs, proteins and DNA. Often a single sequencing experiment produces several millions of RNA sequences. Hence, there is a need for computational methods that can efficiently

process these huge amounts of data.

Objectives

Owing to their essential roles in cellular functions, gene regulation and diseases, the study of ncRNAs has now become a prominent area of research in molecular and computational biology. The main objective of this thesis is to utilize the data from high-throughput sequencing experiments and build efficient computational methods and workflows to predict the small ncRNAs and their interactions. In addition, this thesis also prioritizes transparent and accessible research. The following listing provides an overview of the work presented in this thesis to achieve the aforementioned objectives:

- Each class of ncRNAs has a distinct maturation process and comes in different lengths and characteristics. The traces of their processing are often noticeable in the RNA-Seq data. `BlockClust` from publication *P1* is a novel method that exploits the processing patterns from small RNA-Seq to cluster and annotate the sequenced ncRNA transcripts. It also provides supervised classification models to predict novel functional small ncRNAs. It uniquely encodes the processing patterns as graphs and utilizes graph kernels for fast and accurate clustering of small ncRNAs.

- RNA-Seq can also be used to study transcriptome-wide regulatory interactions of a gene by knockdown or knockout experiments. The work from publication *P2* is an effort to study miRNA interactions in a neurological disorder termed Rett syndrome using RNA-Seq data. Furthermore, `ChiRA` from publication *P3* provides a comprehensive solution for the analysis of genome-wide miRNA interactions from RNA-RNA interactome experiments.

- Owing to low sequencing costs, bioinformatics analysis is experiencing a bottleneck in drawing conclusions from large-scale experiments. The bioinformatics community is striving for solutions by providing easily accessible analysis workflows and training experimentalists to engage them in their own data analysis. RNA workbench from publication *P4* is one such effort to provide RNA-centric research.

Structure of the thesis

The remaining thesis is divided into four parts. Part II presents the biological and computational background required to understand the thesis. This includes the biogenesis of small ncRNAs, high throughput sequencing (HTS) protocols for RNA-Seq and RNA-RNA interactome experiments, general steps involved in the analysis of the HTS data analysis,

and an introduction to graph kernels. Part III gives an overview of the individual contributions related to this thesis. Chapter 3 summarizes the work from publication *P1* and Chapter 4 outlines the contributions from publications *P2* and *P3*. Each contribution is concisely presented in 3 sections, namely motivation, methods overview, and summary of results. Chapter 5 is based on the contribution from publication *P4*. Part IV concludes this thesis and delivers an outlook on the future of small ncRNA research. Part V contains the publications included in this thesis along with the statements of authors' contributions.

Part II

Background

Chapter 1

Biological background

All living organisms on earth consist of fundamental biological entities called cells. Depending on the structure of their cells, organisms can be categorized into prokaryotes and eukaryotes. Eukaryotes contain a special compartment within the cells, called the nucleus, to hold the genetic information, whereas prokaryotes do not. A cell can be viewed as a factory of life where essential functions such as respiration, growth, and reproduction are carried out. The genetic material within a cell is carried in the form of a double helix of nucleotide chains known as deoxyribonucleic acid or the familiar abbreviation "DNA". All the genetic information of an organism is collectively called a genome. Each nucleotide is composed of a phosphate group, deoxyribose (five-carbon sugar ring), and one of 4 primary nitrogenous nucleobases – adenine (A), cytosine (C), guanine (G), thymine (T). Nucleobases are often simply called "bases". The first carbon of the ribose is covalently bound to one of the bases, whereas the third and fifth carbons are connected to a hydroxyl group and a phosphate group, respectively. The $5'$ (five-prime) end of the ribose with the phosphate group of each nucleotide is covalently bound to the $3'$ (three-prime) end with the hydroxyl group of the next nucleotide to form the backbone of the DNA structure. Such a sugar-phosphate chain is known as a DNA strand. The two DNA strands are held together by hydrogen bonds between the bases resembling a twisted ladder, formally called a double helix structure. In general, G is bound to C by 3 hydrogen bonds ($G \equiv C$), and A is bound to T by 2 hydrogen bonds ($A = T$). These pairs of bases that form hydrogen bonds are known as complementary bases and bound together as base pairs.

1.1 The flow of genetic information

Parts of the DNA strands encode functional units called genes. These genes are usually inherited from the parent organism and present in various lengths. For example, human genes vary from tens of nucleotides (in short *nt*) to more than 2 million nucleotides long. The genes on the DNA are not immediately functional. They undergo further processing such as such as splicing and capping to generate functional end products called proteins. This flow of genetic information from DNA to proteins is explained by the *central dogma of molecular biology*, formulated by Francis Crick [1], according to which

i) the DNA within the cells is "replicated", which is an essential step for reproduction and inheritance. First, the DNA double helix structure is unwound by enzymes termed helicases. Then a family of enzymes known as DNA polymerases builds complementary DNA strands for each of the unwound strands by accommodating the complementary bases one after another.

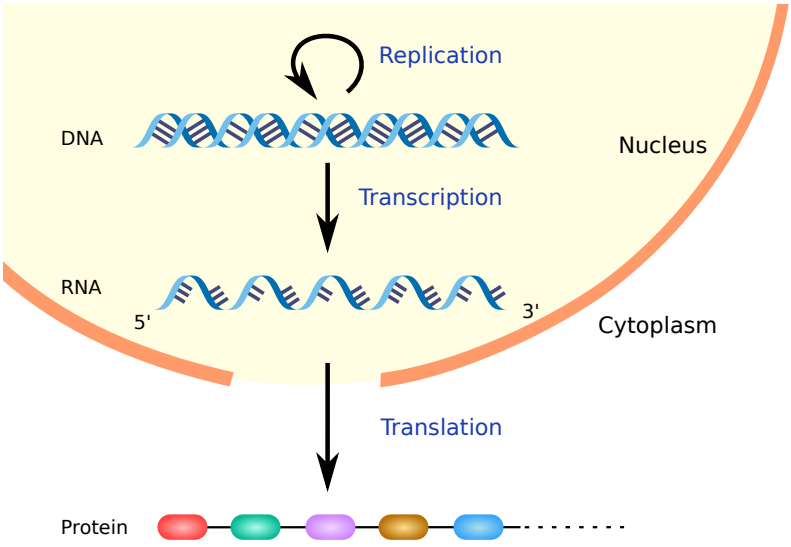


Figure 1.1: The central dogma of molecular biology. Double-stranded DNA is replicated which is essential for cell division and DNA repair. DNA is transcribed to single-stranded RNA, which produces coding and non-coding transcripts. The messenger RNA transcripts are then translated to proteins by ribosome complex with the help of non-coding transfer RNAs. Proteins, linear polymers built from aminoacids are the product of translation. They serve various functions in the cell.

ii) The genetic DNA segments are "transcribed" to another biopolymer known as ribonucleic acid (RNA). This process involves an RNA polymerase enzyme bound to the upstream of the gene (formally known as a promoter) and splits a portion of the DNA helix by breaking the hydrogen bonds among the bases. After that, RNA polymerase synthesizes the complementary RNA strand by adding the nucleotides. In the end, the DNA-RNA helix breaks apart, releasing the newly synthesized RNA free. Some of these RNAs can be functional at this stage, but some need to be processed further to produce functional end products. The RNA that is processed further is generally called messenger RNA (mRNA) or protein-coding RNA. The RNAs which are already functional after transcription are known as non-coding RNAs (ncRNA). As the name implies, they do not yield any protein. In eukaryotes, this processed RNA is transported from the nucleus to the cytoplasm. In contrast to the deoxyribose of DNA, RNA contains a ribose backbone with an additional hydroxyl group attached to the second carbon. Additionally, in RNA, uracil (U) is present instead of thymine (T). Unlike DNA, RNA is usually single-stranded, and is a less stable, more reactive molecule

compared to DNA.

iii) The mRNA undergoes further processing called "translation" to produce the functional end products called proteins. This step mainly involves a ribosome and transfer RNAs (tRNAs). Ribosome is a piece of cellular machinery largely composed of ribosomal RNA (rRNA). Each tRNA carries an "amino acid" which are the basic building blocks of the proteins. The ribosome reads three nucleotides (codon) of an RNA at a time and attaches a tRNA with a complementary anticodon. The amino acids fetched by the tRNAs are chemically attached together to produce a chain of amino acids called peptides and later proteins. Protein synthesis begins at a start codon and ends at a stop codon. The most common start codon contains a nucleotide series ATG. Figure 1.1 shows the process of protein biosynthesis from DNA. Proteins then fold in three-dimensional space due to hydrogen bonds, van der Waals forces, and various other chemical interactions between the amino acids; this folding is key to their biological function. Proteins are considered to be the key actors in cellular functions. Many of them function as enzymes, catalyzing many reactions in biological processes. Proteins also have a structural role in the organs and tissue elements. They play a vital role in coordinating multiple cells through cell signaling.

1.2 General introduction to the non-coding RNAs

Since proteins are involved in many biological processes, most of the transcribed non-coding RNA were initially believed to be transcriptional noise. However, later, the discovery of their functional importance [6, 7] provided a promising future for ncRNA research. The study of human DNA elements by the ENCODE project showed that ∼90% of the human genome is transcribed and only less than 2% of the transcribed RNA encodes for proteins [3]. The great diversity in ncRNAs can be seen from their sizes and functions. The most abundant class of ncRNAs are ribosomal RNAs and transfer RNAs. Although it sounds arbitrary, it is widely accepted that ncRNAs that are longer than $200nt$ are considered as long non-coding RNAs (lncRNAs), whereas shorter ones are classified as small non-coding RNA. LncRNAs that do not overlap with any protein-coding RNAs on the genome are classified as long intergenic non-coding RNAs (lincRNAs). The small ncRNAs include classes of RNAs such as microRNA (miRNA), small nuclear RNA (snRNA), small nucleolar RNA (snoRNA), small nucleolar RNA derived RNA (sdRNA), transfer RNA derived fragments (tRFs) and YRNAs. Each of these small ncRNAs serves a specific cellular function. The role of an RNA in a cellular process is often associated with its sequence and structure.

1.2.1 The structure of RNAs

We have learned that two strands of DNA wrap into a double helix by forming hydrogen bonds between the complementary bases. Despite having a single strand, an RNA molecule can also form hydrogen bonds among the complementary bases within itself. This process of building intramolecular interactions is commonly known as "RNA folding" and results in *RNA structure*. The RNA structure is considered to be one of the key factors that determine its function [8, 9]. Based on the level of abstraction and representation of molecular complexity, a structure of an RNA can be categorized into primary, secondary, and tertiary structure. The primary structure is defined as a sequence of ribonucleotides, which is simply the representation of an RNA sequence. Within the cell, RNA folding yields its native tertiary structure in the 3-dimensional form which is essential for its function. However, due to its intricacy, tertiary structure is not a popular choice for studying an RNA structure. A secondary structure is the representation of base pairs without a backbone and is most commonly used in RNA functional analysis.

Figure 1.2 illustrates the tRNA secondary structure and its main structural components. The tRNAs typically fold in the form of a cloverleaf as shown in the figure. The bases are represented in colored circles and hydrogen bonds with a thick line between the complementary bases. In some cases, atypical *G* and *U* base pairs are also formed with less stable hydrogen bonds. Generally, a stack of base pairs is known as a stem. A loop formed by unpaired bases with a closing stem is a hairpin or stem-loop. Some unpaired bases on one side of a stem form a bulge. An interior loop is formed by bulges on both sides of a stem. A multi-loop is formed by three or more neighboring stems. The top left corner in the figure shows a simplified schematic, only representing the structural elements. From here on, this type of abstract schematic is referred for RNA secondary structure. There are several computational tools available to predict an RNA secondary structure from its sequence [10, 11, 12]. Although throughout the fold-

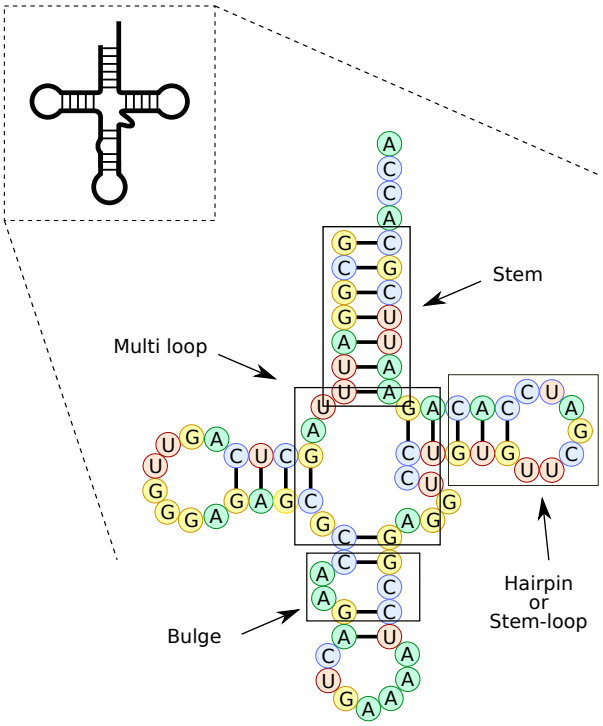


Figure 1.2: Secondary structure of a transfer RNA. Different structural components such as hairpin-loop, stem, bulge, multi-loop are marked in the boxes. An abstract representation is shown in the top left corner.

ing process an RNA folds into numerous structures, the structure prediction tools predict the minimum free energy (MFE) structure. An RNA structure with the lowest free energy is considered to be the most stable and probable structure. The free energy is measured in *kcal/mol*.

1.2.2 RNA-RNA interactions

Interactions among all types of genetic material are essential for cellular functions. Being functional RNA molecules, ncRNAs interact with other ncRNAs and mRNAs. These RNA-RNA interactions play a vital role in RNA processing, transcription, and translation. Similar to the DNA double helix, RNAs interact by hydrogen bonds between the complementary bases. One of such examples was described in Section 1.1. During translation, the anticodon site of a tRNA binds to the mRNA codon that is being translated to deliver the amino acid to the ribosome complex. Another type of interaction that is most relevant to this thesis is between miRNA and mRNA. miRNAs are on average $22nt$ long and are the important gene regulators. The three prime untranslated region (3′-UTR) on the mRNA is the most typical target site of miRNAs. 3′-UTR is the region that follows the stop codon of an mRNA. However, it has also been shown that the miRNAs bind to the coding sequence (CDS) and five prime untranslated region (5′-UTR) [13, 14, 15], i.e. the transcribed region upstream of the start codon. Perfect complementarity between the miRNA seed region (2nd to 7th nucleotide) and the target site is crucial for the miRNA target site recognition. However, several studies suggest that interactions with imperfect seed matches [16, 17] or even interactions at non-seed sites [18, 19] are quite common. Given two RNA sequences, computational methods can predict the interacting regions of those RNAs based on the MFE of the interaction site [20, 21]. These tools often take the "accessibility" of the interaction site into account, i.e. the energy required to break the intramolecular base pairs and make the bases available for an intermolecular interaction. Recent high-throughput experimental methods aim to predict genome-wide RNA-RNA interactions. (see Section 1.4 for more details).

1.3 Types and roles of small non-coding RNAs

To understand the type of information that has been analyzed in this thesis (extensively in Chapter 3), we need to gain insight into the biogenesis of small ncRNAs. This section briefly describes the biological processing and functions of the different ncRNAs, starting from the most abundant microRNAs and transfer RNA fragments followed by small nucleolar and small nuclear RNAs.

1.3.1 MicroRNAs

MicroRNAs are the most abundant small non-coding RNAs in the cell. The first miRNA, *lin-4*, was discovered in *Caenorhabditis elegans* in the early 1990s [22]. A decade after their discovery, their regulatory functions are widely studied in different organisms [23, 24, 25, 26, 27]. miRNAs are the important post-transcriptional gene expression regulators. They bind to the messenger RNAs and mediate the translational repression or degrade them. Deregulation of miRNAs is associated with various diseases ranging from several cancer types [28] to cardiovascular diseases [29] to neurodegenerative disorders [30]. The biogenesis of miRNA gives an insight into how these fascinating short 22 nucleotide molecules are produced.

A schematic representation of miRNA biogenesis is shown in Figure 1.3. First, a specific region of DNA that encodes for a miRNA gene is usually transcribed by *RNA polymerase II* and results in a primary miRNA (pri-miRNA) of more than 1kb in length. This pri-miRNA within the nucleus folds into one or several hairpin structures, each of approximately $70nt$ length. Then a protein complex called the microprocessor complex, composed of an RNA binding nuclear protein *DGCR8* and an enzyme *Drosha* cuts out and releases individual stem-loop structures known as precursor miRNAs (pre-miRNA). Based on the reading direction, pre-miRNA stems can be called $5'$ and $3'$ stems. There is an alternate pathway that produces these pre-miRNAs bypassing the microprocessor complex. Those pre-miRNAs originate from the intronic or seldomly from exonic parts of mRNAs through splicing. The pre-miRNA is then exported by *Exportin-5 (XPOS5)* protein into the cytoplasm where the miRNA maturation process is carried out.

Thereafter an enzyme known as *Dicer* cleaves the pre-miRNA near the hairpin loop leaving out the miRNA duplex. Subsequently, the miRNA-duplex is loaded into one of the *Argonaute (AGO)* family of proteins. *AGO* proteins are the core components of the miRNA-induced silencing complex (miRISC) where one of the duplex strands targets mRNA and the other strand is degraded. The strand that is involved in RNA interference is often referred to as mature miRNA or guide strand whereas the degraded strand is a called miRNA* or passenger strand. Later research revealed that often miRNA* is also functional in particular tissues and cell lines and bound to certain Argonates [31, 32]. Therefore, based on the stems from which these miRNA and miRNA* originate, they are named as miR-5p (from $5'$ stem) or miR-3p (from $3'$ stem) mature miRNAs. This new nomenclature is now widely accepted and used in the well-known miRNA database miRBase [33]. Usually, mRNAs are targeted at multiple sites by a single miRNA or several distinct miRNAs and act cooperatively for an effective translation repression [34]. Generally, over time, the poly(A) tails at the $3'$ end of the mRNAs are shortened. This process of deadenylation affects translational efficiency and leads to the degradation of the mRNA. Upon binding to the target mRNA, miRNA

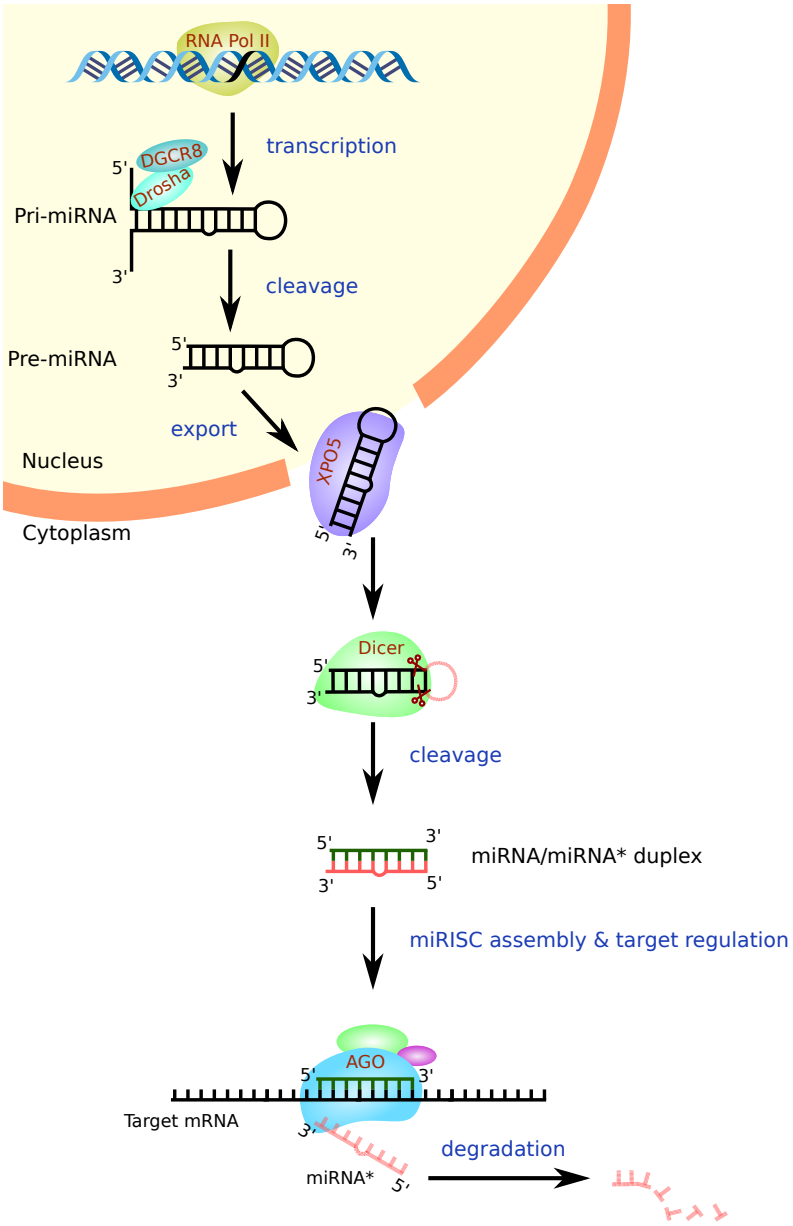


Figure 1.3: MicroRNA biogenesis. A miRNA gene is transcribed to pri-miRNA whose ends are then cleaved by *Drosha* to produce a stem-loop structured pre-miRNA. Pre-miRNA is translocated from the nucleus to the cytoplasm where its hairpin is cleaved by *Dicer* to produce a miRNA duplex. One of the miRNA strands interacts with mRNA and regulates its expression, whereas the other strand is degraded.

accelerates the process of deadenylation which speeds up the target mRNA decay. The repression of the target mRNA is achieved by either blocking the translation initiation or at the elongation stage [35]. Arguably, there is still not enough evidence of the direct effects of mRNA degradation on translation repression [36, 37].

1.3.2 Transfer RNA derived fragments

In Section 1.1, we have seen the importance of the tRNAs in protein synthesis. Transfer RNAs can produce short RNA fragments that are of similar length to mature miRNAs. These are commonly known as tRNA derived fragments (tRFs) [38, 39]. The tRNA derived fragments are the second most abundant small ncRNAs next to miRNAs. These tRFs serve important biological functions such as RNA processing, cell proliferation, and gene regulation [40, 41, 42]. It has been shown that the tRFs are also involved in miRNA like gene regulation [43, 44] and are implicated in several human diseases [45]. The complete processing of a tRNA reveals the generation of the tRFs.

Transfer RNA maturation is a relatively complex process in contrast to miRNA maturation. Here, only the major steps involved in tRNA maturation and tRNA fragment generation have been described. Unlike miRNA genes, tRNA genes are transcribed by *RNA polymerase III* resulting in precursor tRNA (pre-tRNA). The $5'$ and $3'$ trailers of the pre-tRNA are then cleaved by *ribonuclease P* and *ribonuclease Z* enzymes, respectively. Then *TRNT1* enzyme adds CCA sequence at the $3'$ terminal of the pri-tRNA which is essential for aminoacylation. Aminoacylation is the process of attaching an amino acid to a tRNA, which then transfers the amino acid to the ribosome during mRNA translation.

Often, tRNAs also undergo post-transcriptional chemical modifications that produce non-canonical nucleotides. These nucleotide modifications ensure the thermodynamically stable structure that makes it accessible to other enzymes and RNAs to interact [46]. Typically tRNAs form a cloverleaf shaped structure with a D-loop, an anticodon loop, and a T-loop. The D-loop contains a modified dihydrouridine base and it is important for aminoacylation. The anticodon loop contains a base triplet that is complementary to an mRNA codon and involved in base pairing during mRNA translation. The T-loop contains thymidine, modified uridine (pseudouridine) and it facilitates rRNA interaction during translation.

The tRNAs processed further in the cytoplasm to generate tRNA derived fragments [38, 39]. These tRFs were initially thought to be products of degradation but later found to be functional. *Dicer* dependent cleavage at the tRNA D-loop releases $5'$ tRFs. Either *Dicer* cleavage or a less frequent angiogenin cleavage at T-loop produces $3'$ CCA tRFs. The $3'$ trailers that are cleaved from pre-tRNA are sometimes exported to the cytoplasm. These fragments are referred to as $3'$-U tRFs due to the presence of a poly-U tail at the $3'$ end.

Another type of tRNA fragment is tRNA halves, which are generated by cleaving the tRNA at the anticodon site by anticodon ribonuclease (angiogenin *ACNase*). This *ACNase* is generally activated by oxidative stress and starvation [47]. This results in two tRNA halves, namely the $5'$ tRNA half and $3'$ tRNA half.

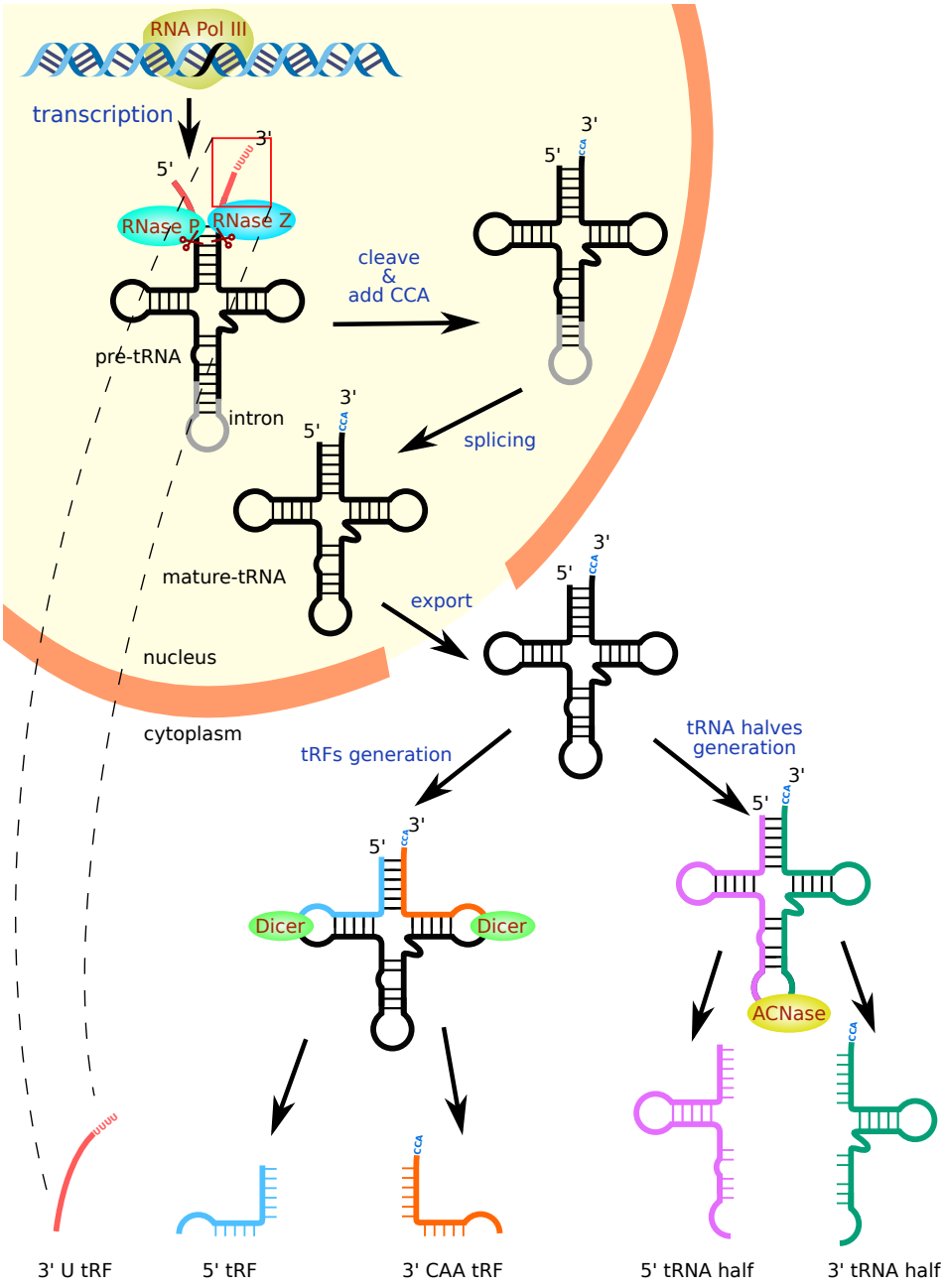


Figure 1.4: Generation of tRNA derived fragments. 5′ and 3′ ends of the transcribed pre-tRNA are cleaved by *RNase P* and *RNase Z* enzymes, respectively. Cleaved 3′ poly-U tail generates 3′ U tRF. At the 3′ end of the pre-tRNA, a CCA tail is added and introns are spliced to generate a mature tRNA. Mature tRNA is exported to cytoplasm and cleaved by *Dicer* at D-loop and T-loops to generate 5′ tRF and 3′ CAA tRF, respectively. Alternatively, it is cleaved at the anticodon site by angiogenin to generate 5′ and 3′ tRNA halves.

1.3.3 Small nucleolar RNAs

Small nucleolar RNAs (snoRNAs) are a type of small ncRNAs that are involved in the chemical modifications of other ncRNAs [48]. On the basis of secondary structure and specific sequence motifs, snoRNAs can further be classified into C/D box snoRNAs and H/ACA box snoRNAs [49]. The class of C/D box snoRNAs contains two common sequence motifs, namely box C(RUGAUGA) near the $5'$ end, and box D (CUGA) near the $3'$ end of the snoRNAs. With complementary bases upstream of the C box and downstream of the D box, snoRNAs form a typical stem-loop structure. H/ACA box snoRNAs contain box H (ANANNA), and ACA sequence motifs and form an evolutionarily conserved structure with a hairpin followed by unpaired bases containing box H element and a second hairpin followed by unpaired bases with ACA towards the $3'$ end. It was shown that snoRNAs can be processed to generate miRNA-like fragments [50, 51]. These sno-derived RNAs (sdRNAs) most commonly originate from the $5'$ end of the C/D box snoRNAs and $3'$ end of the H/ACA box snoRNAs [52]. Similarly to miRNAs, sdRNAs are processed by *Dicer* (but not *Drosha*). sdRNAs derived from H/ACA box snoRNAs are of similar length to miRNAs and are involved in miRNA like gene regulation in association with AGO protein complexes[51], whereas C/D box snoRNA originated sdRNAs are generally longer than $25nt$ and involved in alternative splicing mechanism [53].

1.3.4 Small nuclear RNAs

Small nuclear RNAs (snRNAs) play a vital role in messenger RNA maturation by catalyzing the splicing. Based on proteins they associate with and sequence features, snoRNAs can be divided into two major subclasses, namely Sm snRNAs and Lsm snRNAs [54]. The snRNAs associated with the Sm snRNA class are U1, U2, U4, U5, U11, U12, and U4atac. The class of Sm snRNA genes is transcribed by *RNA polymerase II* into pre-snRNAs and then exported to the cytoplasm. In the cytoplasm, the maturation takes place and assembled with specific Lsm proteins to form small nuclear ribonucleoproteins (snRNPs) [55, 56]. This RNA-protein complex is then transported back into the nucleus. In contrast to Sm snRNAs, Lsm snRNAs are transcribed by *RNA polymerase III* and the snRNAs associated with the Lsm snRNA class are U6 and U6atac. The complete Lsm snRNAs processing and snRNP assembly is carried out within the nucleus. Inside the nucleus, the snRNPs along with various helper proteins and pre-mRNA forms a spliceosome [57]. This spliceosome is responsible for the removal of introns from pre-mRNAs.

1.4 High-throughput sequencing of RNAs and their interactions

The invention of the first DNA sequencing methods dates back to the early 1970s [58]. Since then, several branches of science such as genomics, biotechnology, forensic, anthropology, and archaeology are taking advantage of DNA sequencing. There exist several conventional methods to sequence DNA, among which Sanger's chain termination method and Maxam & Gilbert chemical-degradation methods are notable [59, 60]. These methods are known as the "first generation" sequencing methods. The first complete DNA sequencing was carried out in 1977 on a bacteriophage ϕX174. It was found to have approximately 5,375 nucleotides at that time and was sequenced using "plus and minus method" [61]. Later in 2001, the first human genome draft was released. It was sequenced using the Sanger method [62, 5], took 10 years, and cost $3 billion to complete. Following that, several scalable and cost-effective approaches to whole-genome sequencing were introduced. These are collectively known as the "next-generation sequencing" (NGS) methods. One of the main differences to first-generation sequencing is that the genome is broken up into small "fragments", amplifying and sequencing them massively in parallel [63].

First, the DNA or RNA to be sequenced should be extracted from the biological sample (sample preparation) and it should be made compatible with the sequencing machine (library construction). Depending on the origin of genetic material, each sample preparation protocol differs. For example, RNA Sequencing (RNA-Seq) [64] can be used to determine the RNA transcript abundances or study alternative splicing. Genome-wide DNA-protein interactions and histone modifications can be studied using chromatin immunoprecipitation and sequencing (ChIP-Seq) [65]. Various crosslinking and immunoprecipitation (CLIP) based sequencing protocols [66, 67] can be used to identify genome-wide RNA-protein interactions or RNA modification sites. Crosslinking, ligation, and sequencing of hybrids (CLASH) [68, 17] and covalent ligation of endogenous argonaute-bound RNAs-CLIP (CLEAR-CLIP) [69] deals with the targeted RNA-RNA interactions, whereas protocols like psoralen analysis of RNA interactions and structures (PARIS) [70] and sequencing of psoralen crosslinked, ligated, and selected hybrids (SPLASH) [71] can be used to analyze genome-wide RNA-RNA interactions or RNA structures. Each of the aforementioned protocols generates fragments of DNA or RNA. The final sequencing step is independent of the sample preparation method.

One major aspect of my work is the analysis of RNA sequencing data and RNA-RNA interactome data. The following sections outline the steps involved in sample preparation of RNA sequencing and RNA-RNA interactome and structurome protocols followed by the Illumina sequencing approach.

1.4.1 RNA sequencing and small RNA sequencing

Before the invention of RNA sequencing, gene expression levels were measured by DNA microarray [72]. It is a chip with a surface attached with a large number of short DNA sequences known as probes. These probes are sections of genes that are specific to the genes. Probes are then used to hybridize the labeled target DNA or antisense RNA. Owing to the complementarity between the probes and DNA in the sample, the probes and target DNA hybridize. Upon hybridization, the abundance of target RNA or DNA in the sample is quantified. Despite its huge success [73, 74], there are several major drawbacks to this approach. As predetermined probes are used, it has no ability to detect novel transcripts. The range of gene expression measurements was limited, meaning that it fails to detect lowly expressed or very highly expressed genes. It can only measure the relative abundances of the same targets within two different conditions; for example, measuring the relative abundance of genes between normal and cancer cells. In 2008, the superior RNA-Seq protocol [75, 76] was used for the first time on different organisms to sequence the entire transcriptome [77, 64, 78]. RNA-Seq measures the absolute transcript abundances, which helps in differential expression analysis comparing samples of two different conditions. It can also be used to study alternative splicing events and detect single nucleotide variations. RNA-Seq is highly sensitive and can detect transcripts with a wider range of expression levels than microarrays. It is also capable of detecting novel transcripts and splicing variants.

Before RNA can be sequenced in a high-throughput manner, RNA in cells must be extracted and prepared. This step is generally referred to as sample preparation. Sample preparation of an RNA-Seq starts with all the transcribed RNA in the cell (known as total RNA). Sequencing total RNA is overwhelmed by the most abundant rRNA. Hence, for a cost-effective sequencing, total RNA is filtered and transcripts of interest are extracted. Early RNA-seq protocols used oligo (dT) primers to isolate mRNAs with poly-A tails. Although it is a cost-effective method, it fails to capture all other interesting long non-coding RNAs, tRNAs, and any other non-polyadenylated RNA. In certain cases with degraded RNA, this method yields low throughput. A better alternative is rRNA depletion. This method uses biotinylated oligo probes that are complementary to the parts of rRNAs and washes out all the captured transcripts [79]. As it only aims to remove the most abundant rRNA, ideally, the rest of the transcriptome is enriched. To acquire small ncRNA fragments such as miRNAs and tRNA fragments, snoRNA derived RNAs from the total RNA, RNAs are size selected [80, 81]. Sequencing these short functional RNA fragments is called small RNA sequencing (smRNA-Seq). The extracted RNA is sheared into short fragments (typically ranging from 50-300nt long). As smRNA-Seq RNA extraction already produces short RNAs, the fragmentation step is skipped. Being single-stranded and susceptible to hydrolysis, RNA is less stable than DNA. For this reason, RNA fragments are reverse transcribed to make

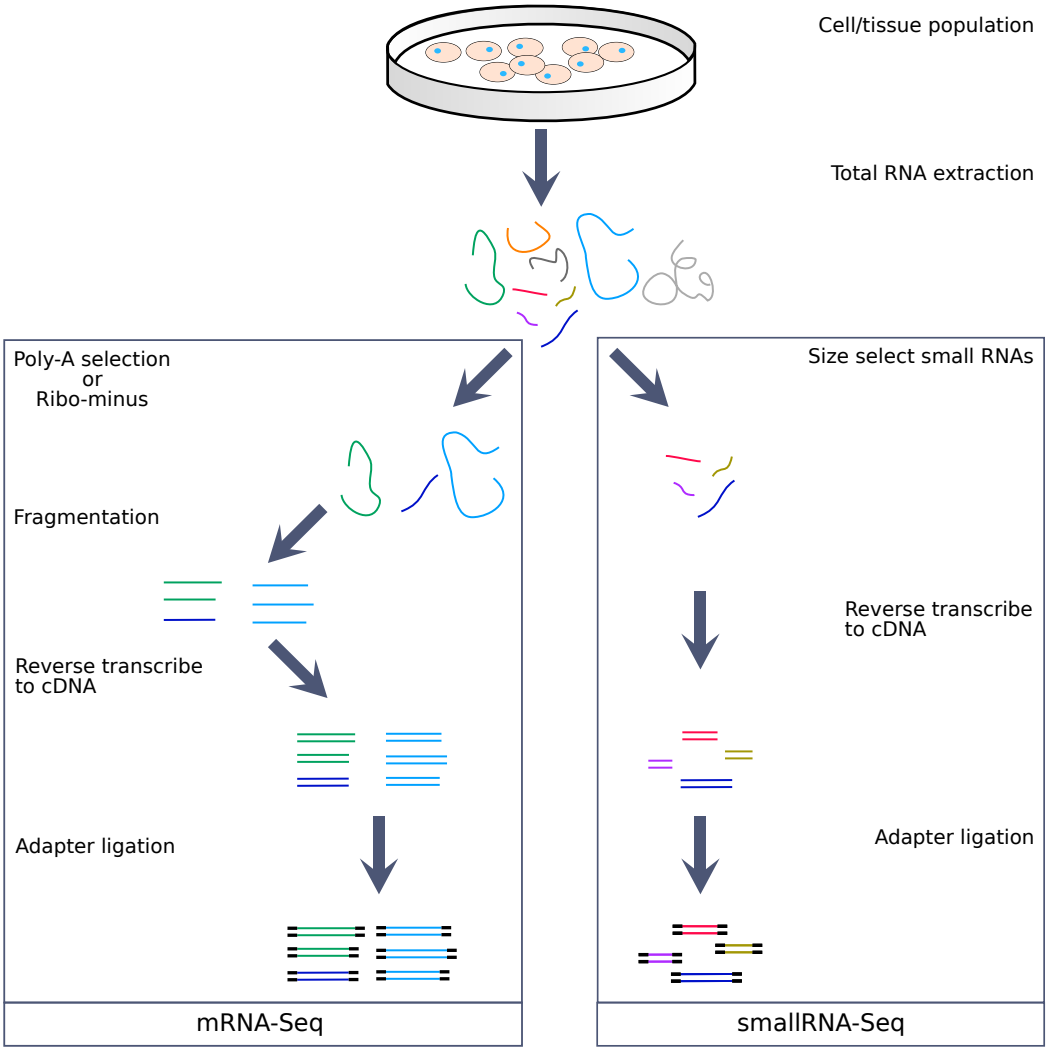


Figure 1.5: RNA-Seq sample preparation. Total RNA is extracted from the cell population. In (m)RNA-Seq, transcripts with poly-A are selected or ribosomal RNA is depleted and then fragmented. Small RNA-Seq aims to capture the small non-coding RNAs through the size selection of RNA transcripts. The RNA fragments are then reverse transcribed and adapters are ligated.

complementary DNA (cDNA). Then adapters are ligated to both 5′ and 3′ ends of the cDNA fragments. Adapters are short non-organism specific sequences that help sequencing machines to recognize the cDNA fragments and allow simultaneous sequencing of different samples (multiplexing). A strand-specific sequencing library uses different adapters for 5′ and 3′ ends in a predetermined orientation so that in the subsequent step, only one specific cDNA strand is sequenced. This enables accurate mapping of the read fragments to the reference genome (see section 2.1.2 for more details) which allows detection of antisense transcripts (transcribed from complementary DNA strand of an mRNA or ncRNA) and proper transcriptome assembly. Figure 1.5 depicts the steps involved in (m)RNA-Seq (on

the left) and smRNA-seq (on the right).

1.4.2 RNA interactome and RNA structurome protocols

Although CLIP protocols are designed to determine RNA-protein interactions, an elegant application enables RNA-RNA interaction predictions. For example, AGO-mediated HITS-CLIP [82] and PAR-CLIP [83] were previously used to predict transcriptome-wide miRNA interactions. The main disadvantage of these protocols is that miRNA and its target are not captured together, which leaves the overhead of dealing with ambiguous interactions through bioinformatics analysis. CLASH [68, 17] is the first protocol that ligates and sequences the miRNAs along with their targets. CLASH is a quite successful method, with a limitation that only AGO-based RNA-RNA interactions are predictable. Protein-independent crosslinking is achievable via other protocols like PARIS [70], SPLASH [71], and LIGR-Seq [84].

A summary of common steps in RNA-RNA interactome experiments is shown in Figure 1.6. The first step in the sample preparation is crosslinking interacting RNA strands. These RNA strands can be from different RNAs representing an intermolecular interaction or from a single RNA representing an intramolecular interaction (RNA structure). In protocols like CLASH and CLEAR-CLIP, interacting RNAs are crosslinked with proteins by UV irradiation. The protein-independent methods, such as PARIS, SPLASH, and LIGR-Seq, use psoralen derivatives for crosslinking. On UV irradiation, psoralen crosslinks between the pyrimidines of RNA strands. PARIS and LIGR-Seq use *4′-aminomethyltrioxsalen hydrochloride (AMT)* for crosslinking, whereas SPLASH uses a biotinylated *psoralen*. The advantage of *psoralen*-based crosslinking is that any prior knowledge of interacting proteins is not needed, allowing genome-wide RNA-RNA interactome and structurome sequencing. After crosslinking, the interacting RNAs (whole RNA-protein complex for CLASH) are purified with *RNase* digestion. In SPLASH protocol, crosslinked RNAs are fragmented and enriched using streptavidin beads. Following purification crosslinked RNA strands are proximity-ligated using *T4 RNA ligase I* (circRNA ligase in LIGR-Seq). This results in fragments with two interacting RNAs next to each other. These fragments are called chimeric fragments. Due to limited crosslinking efficiency, these protocols produce more fragments with singleton RNAs than chimeric. These fragments are then reverse transcribed, adapters and barcodes are attached and ready to be sequenced.

1.4.3 Sequencing by synthesis

As the data in this thesis is derived from Illumina sequencing machines, here sequencing by synthesis is described. After sample preparation, cDNA fragments are transferred to the flow cell [85], a tiny glass slide used in the sequencing machine to hold the fragments. A series of hundreds of millions of DNA oligonucleotides are immobilized to the surface of

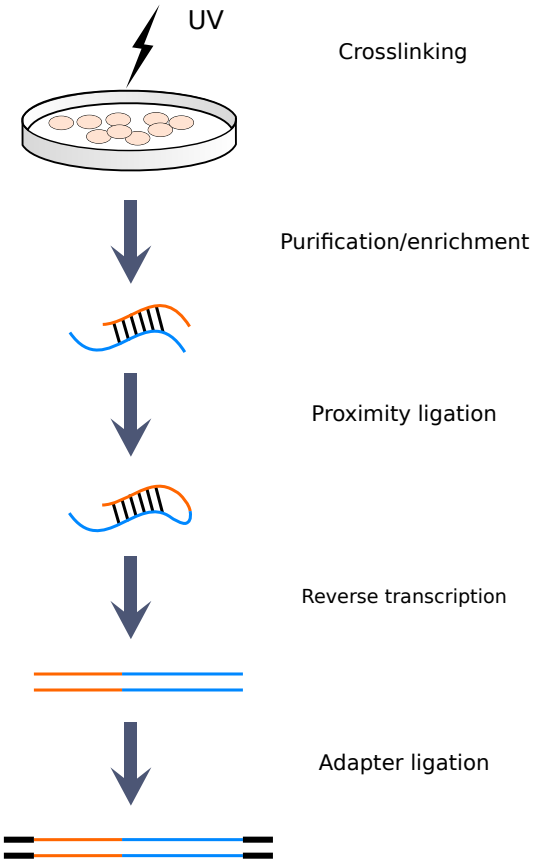


Figure 1.6: Steps involved in RNA interactome sample preparation. First the interacting RNAs are crosslinked by UV irradiation. Some methods use derivatives of psoralen to crosslink. Crosslinked RNAs are gel-purified or enriched on streptavidin beads. Then the ends of the RNAs are ligated, reverse transcribed and adapters are ligated.

the flow cell. These oligos are complementary to one of the adapters and thus one of the denatured cDNA strands adhere to the flow cell. To achieve a good sequencing precision, the cDNA strands are amplified by polymerase chain reaction (PCR) on the flow cell forming clusters of fragment copies. Following the amplification, sequencing is done by synthesizing the complementary strands to the clusters of fragments. Sequencing is done for a fixed number of cycles that define the target length of the sequences. In each cycle, one of the fluorescently labeled complementary base A, C, G or T is synthesized by the polymerase. Upon base incorporation, the fluorescent label is illuminated by light and the color of each cluster is recorded by the sequencing machine. These recorded colors convert back to letters A, C, G and T, which correspond to bases and depending on the color intensity each base is given a confidence score. The sequencing can be done in a single-end or paired-end layout. In a single-end layout, the sequencer synthesizes a fragment in only one direction, whereas in a paired-end layout, synthesis is done from both directions of a fragment. The final output

is a specially formatted text file containing a sequence of bases representing RNA fragments in the sample and a score for each base of the sequences. Each sequenced fragment is called a *read*. The file that contains reads comes in FASTQ format, which is the most common format to start the analysis with. A FASTQ file contains information about the sequenced fragments with nucleotide level quality scores encoded in ASCII characters [86].

Chapter 2

Computational background

2.1 Next generation sequencing data processing

In general, the analysis workflows for data from each sequencing protocol differ. The following are the common steps involved in various data analyses carried out in this thesis.

2.1.1 Pre-processing

The sequenced reads may contain adapters ligated during library preparation, low-quality bases, and even contain contaminants. Often sequencing facilities provide some statistics on which species reads are possibly coming from. One can foresee cross-species contamination at this step. An inefficient rRNA depletion leaves unwanted rRNA in the final sequenced reads. As the 5′ adapter contains the complementary bases to the flow cell, the sequencing starts following the 5′ adapter. Hence, generally, 5′ adapters are not expected in the reads. However, it can happen that 3′ adapters are present in the reads. If the RNA fragment being sequenced is shorter than the number of sequencing cycles, then the sequencer synthesizes the whole RNA fragment followed by a part or complete 3′ adapter. This produces reads with adapters at their 3′ end. For example, owing to short miRNAs and tRFs, it is common to have 3′ adapters in smRNA-seq data. During the sequencing, if a few fragments of a cluster miss a cycle, they become out of sync with the other synthesized strands, causing uncertainty on the sequenced base. Over the cycles, cumulatively, this uncertainty results in low-quality bases towards the 3′ ends of the reads. All these issues must be resolved before the actual analysis starts.

Quality control programs provide basic to very advanced statistics on data quality at each analysis step. FastQC [87] provides raw read quality statistics such as base qualities, GC content, read length distribution, sequence duplication levels, overrepresented sequences. RSeQC [88] is a comprehensive tool suite that can be used for quality control at different stages of RNA-Seq analysis.

Adapter trimming is one of the important steps that facilitates streamlined mapping of reads to the reference in subsequent analysis [89, 90]. Reads with low-quality ends and adapters can be trimmed using various bioinformatics tools [91, 92]. In small RNA-seq analysis and RNA-RNA interactome data analysis, adapter trimming is followed by read

deduplication. This deduplication can essentially remove the PCR duplicates and drastically reduces the number of reads, hence speeding up the subsequent analysis. A drawback of this process is that after deduplication we lose base-level quality scores. These quality scores can be used in mapping (Section 2.1.2) step to score the alignments, but this is not mandatory. One of the analyses where base level qualities are essential is the detection of single nucleotide variants. However, this type of analysis is not in the scope of this thesis and there is no added benefit from the preservation of base-level quality scores for my analysis. The latest library preparation methods attach a distinct short sequence called uniform molecular identifier (UMI) at the $5'$ end of each fragment before the PCR amplification. UMIs help in identifying the sequenced fragments uniquely. Identical RNA fragments with distinct UMIs are possibly from gene isoforms or paralogs, whereas identical fragments with the same UMI are likely PCR duplicates. UMI-based read deduplication improves the accuracy of RNA abundance estimation [93].

2.1.2 Read mapping

The reads do not carry any information on which transcripts they are coming from. Mapping is a process of inferring the origin of the reads. This is generally done by aligning the reads to the reference genome. A reference genome is a consensus genomic sequence of an organism that is built from several individuals of that organism. In general, alignment is done by matching individual bases of a read sequence to the reference sequence. Mismatches, deletions (matching a base on the reference to a newly introduced gap '_' character in the read), and insertions (matching a base on the read to a gap on the reference) are allowed. Reads that originate from the splice junctions (split reads) cannot be aligned linearly to the reference genome. In this scenario, using a transcriptome as a reference is an alternative. For extensively studied organisms, well-annotated transcriptomes are available. If a reference transcriptome is not available, one can use the genome and align the sub-sequences of the read. This process is generally called local alignment. Local alignment in combination with common splice junction sequence motives such as GU-AG solves the problem of split read mapping in the absence of a reference transcriptome. Chimeric reads from the RNA-RNA interactome protocols can be considered as a special type of split-reads with no constraint on the origin of the chimeric RNA fragments. If reads still contain any adapters or contamination, local alignment is a viable choice because it can align the only portion of the read that belongs to the organism.

A single read aligned to multiple reference locations results in multi-mapping. Usually, repetitive regions on the reference cause multi-mapping. Possible sources of these repetitive sequences are gene paralogs, gene families, and gene isoforms (if the transcriptome is used as reference). Despite the multi-mapping, the true origin of the read is only from one of

the transcribed loci. Figure 2.1 shows the alignment of a uniquely mapped read (green), multi-mapped read (blue), and a split or chimeric read (red).

There are numerous mapping algorithms published in the last decade [94, 95]. Splice-aware read mappers tailored for RNA-seq data can handle split-reads [96, 97]. As the chimeric reads are a special case of split reads, splice-aware read mappers like STAR with appropriate alignment settings can be used for chimeric read mapping. Most recently, an explicit chimeric read aligner for RNA-RNA interactome experiments was published [98]. A naive alignment of a single read to the reference can be achieved with a time complexity of $O(mn)$, where m is the length of the reference and n is the read length. Aligning millions of reads with this time complexity is practically unreasonable. Therefore mapping programs use algorithms such as Burrows-Wheeler transform and suffix arrays to reduce the time complexity to $O(n)$. Given any reference position, the number of aligned reads represents the read coverage or depth of that position. The read coverage along a reference transcript positions called as the *read coverage profile* or simply *read profile* of that transcript.

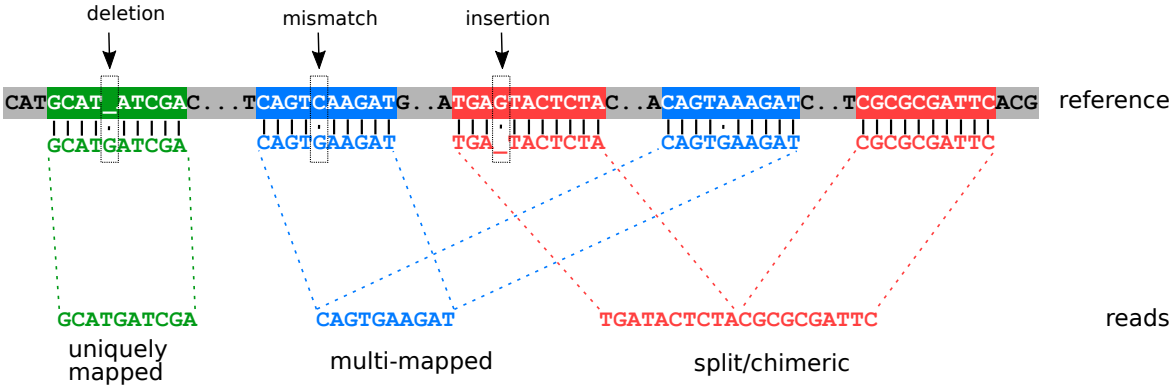


Figure 2.1: Mapping of uniquely mapped, multi-mapped, and chimeric reads. Insertion, deletion and mismatches are shown in boxes along with the alignments of reads against the reference.

2.1.3 Quantification

With alignments in hand, one can estimate gene or transcript expressions within a sample. This is equivalent to identifying which genes are transcribed to produce functional proteins or ncRNAs. Transcript or gene abundance is measured by a process called quantification. The term quantification can loosely be defined as counting the number of sequenced reads per gene. Because of multi-mapped reads, quantification can be challenging. Exons that are shared among the isoforms of a gene adds another layer of complexity for transcript-level quantification. There are several accepted ways of counting multi-mapped reads. Some programs follow simple approaches like discarding the multi-mapped reads or counting them to

all multi-mapped reference genes/transcripts [99] or fractionally divide each read contribution to each reference it mapped to [100]. There are also more sophisticated methods that probabilistically assign the multi-mapped reads [101, 102]. Even though modelling these methods can be quite complex, their abundance estimates are accurate [103]. The ultimate goal of quantification is to measure the relative abundances of different transcripts within a sample or to identify the genes/transcripts that are differentially expressed between two samples prepared from two different biological conditions.

2.2 Machine learning concepts

This section outlines the basics and some key concepts of machine learning that are used in this thesis. Machine learning provides computers with algorithms to solve problems by experience rather than by explicit programming. Machine learning is one of the rapidly growing fields of science that has broad-spectrum applications such as weather prediction, stock market trading, image and speech recognition, medical diagnosis, e-commerce, social media platforms, self-driving cars, and so on. Bioinformatics is not an exception and has also made use of this great resource. The main objective of machine learning is to discover intrinsic features in the data to achieve a certain task. The two most common learning techniques are supervised learning and unsupervised learning. In supervised learning, example input objects are provided along with the target class such that the algorithm learns the patterns in the input to map to the output. Classification is an instance of supervised learning in which the algorithm predicts a target class for any new input object based on the patterns it learned. In contrast, in unsupervised learning, only the input objects are provided and the algorithm learns the patterns to group similar objects in the input. Clustering can be achieved through unsupervised learning. The process of learning is also often called training and the data set with the examples to learn is called training data. The data set on which the learning is evaluated is called test data. The training process involves feature extraction. A feature is a specific characteristic of a training object. In general, several such features are needed to represent each training object. Therefore, each training object is denoted as an n-dimensional vector of features it represents. Intuitively, the features that can discriminate the different classes in training data are used.

Given a set of countable objects X along with their target classes T, to map a new object $y \notin X$ to a target label $t \in T$, we need a function that generalizes the mappings from X to T by measuring the similarities of all possible combinations of objects in X. Such a function is called a *kernel*, $\mathcal{K} : X \times X \to \mathbb{R}$ such that, $\forall x, y \in X \ \mathcal{K}(x, y) = \langle \phi(x), \phi(y) \rangle$, which is a dot product on vectors called *feature space* induced by ϕ.

Each object $x \in X$ can be a composite structure of P parts ($P \geq 1$), i.e., $x_1, ..., x_P$ such that $x_p \in X_p$ for $p = 1, ..., P$. This composite structure can be briefly represented as

$\vec{x}$. Haussler *et al.* [104] defined a relation R on the set $X_1 \times ... \times X_P \times X$ to represent the relation between x and $\vec{x}$. $R(\vec{x}, x)$ is true iff $x_1, ..., x_P$ are the parts of x. The parts of x can be generated by the same but inverse relation $R^{-1}(x) = \{\vec{x} : R(\vec{x}, x)\}$. For any two composite structures, $x, y \in X$, with parts $x_1, ..., x_p$ $(\vec{x})$ and $y_1, ..., y_p$ $(\vec{y})$, respectively, a kernel can be defined as the sum of products of kernels over the respective parts of x and y.

$$\mathcal{K}(x, y) = \sum_{\substack{\vec{x} \in R^{-1}(x) \\ \vec{y} \in R^{-1}(y)}} \prod_{p=1}^{P} \mathcal{K}_p(x_p, y_p) \tag{2.1}$$

The kernel defined above is called a *decomposition kernel.* Neighborhood Subgraph Pairwise Distance Kernel (NSPDK) is an instance of a decomposition kernel that was used in publication *P1*. To understand the NSPDK, we first need to learn the following terminology related to graphs and graph kernels in general.

Graph: A graph G is set of vertices V and set of edges $E \subseteq \{(u, v) | u \neq v \wedge u, v \in V\}$ that connects them. Formally, $G = (V, E)$ with $|E| \leq |V|(|V| - 1)/2$.

Directed graph: A directed graph is a graph with edges that are ordered pairs of vertices, i.e., $E \subseteq \{(u, v) | u \neq v \wedge u, v \in V^2\}$ with $|E| \leq |V|(|V| - 1)$.

Induced subgraph: An induced subgraph G' is a graph formed by subset of vertices $V' \subseteq V$ and edges connecting them $E' \subseteq E$, i.e., $G' = (V',\ E')$.

Vertex neighborhood: The neighborhood of vertex $v \in V$ with a radius r is the set of connected vertices with shortest path less than or equals r. The shortest path is also referred to as distance and the distance between any two vertices u and v is denoted by $\mathcal{D}(u, v)$. Hence, the vertex neighborhood $N_r(v) = \{V' \subseteq V \mid \forall u \in V'\ \mathcal{D}(u, v) \leq r\}$.

Neighborhood subgraph: A subgraph induced by vertex v with a neighborhood of radius r is called a neighborhood subgraph of v and is represented by $G^v_{N_r}$.

Graph Isomorphism: Two graphs are called isomorphic if there exists the same number of vertices and edges with an identical edge connectivity. Graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic (denoted by $G_1 \simeq G_2$) if $|V_1| = |V_2|$, $|E_1| = |E_2|$ and if there exists a bijective mapping f such that, $u, v \in V_1 \& \ (u, v) \in E_1 \Leftrightarrow (f(u), f(v)) \in E_2$.

Graph invariant: A property that is preserved between two isomorphic graphs.

Graph kernel: Given a set of graphs $\mathcal{G}$, a graph kernel is analogous to the definition of the kernel. i.e., $\mathcal{K} : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ such that $\forall G, G' \in \mathcal{G}$, $\mathcal{K}(G, G') = \langle \phi(G), \phi(G') \rangle$. Here the similarities measured from graph isomorphism.

2.2.1 Neighborhood Subgraph Pairwise Distance Kernel

NSPDK is a graph kernel and is an instance decomposition kernel. Using NSPDK, the similarity between any two graphs is measured by counting the number of identical neighborhood subgraph pairs. The selection of neighborhood subgraph pairs is carried out by a relation $R_{r,d}$ which selects all pairs of neighborhood subgraphs with radius r whose roots are separated by a distance d. The relation between two rooted graphs A_a, B_b and a graph G, $R_{r,d}(A_a, B_b, G)$ to be true iff both A_a and B_b are in neighborhood of some vertex in G with radius r and $\mathcal{D}(a, b) = d$. In NSPDK terminology, each subgraph pair is a feature.

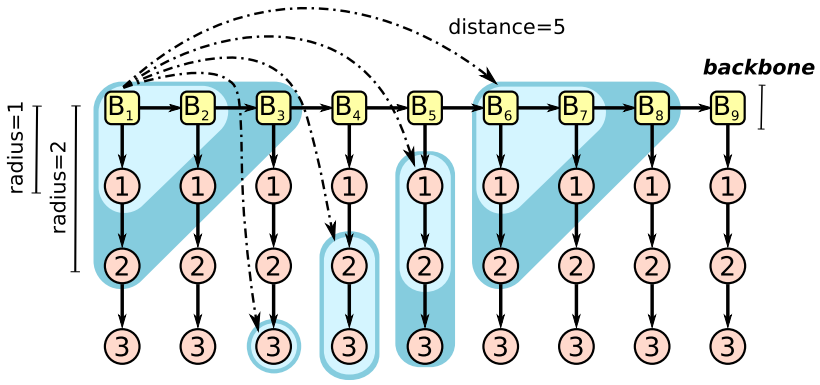


Figure 2.2: Feature extraction from a directed graph with maximum radius $R = 2$ and maximum distance $D = 5$. In each turn, one of the vertices is considered as a root. In this example, with B_1 as root, a neighborhood graph of radius $r = 1, ..., R$ is extracted. Then all possible pairs of neighborhood graphs of the same radius whose respective roots are exactly at distance $d = 1, ..., D$ are considered. Each such pair of subgraphs is an NSPDK feature. This figure is taken from Publication *P1*.

Following the generalized convolution, adhering to the kernel definition in equation 2.1, a decomposition kernel on graphs $G, G' \in \mathcal{G}$, $\mathcal{K}_{r,d} : \mathcal{G} \times \mathcal{G}$ on relation $R_{r,d}$ is defined as following:

$$\mathcal{K}_{r,d}(G, G') = \sum_{\substack{A_a, B_b \in R^{-1}_{r,d}(G) \\ A'_{a'}, B'_{b'} \in R^{-1}_{r,d}(G')}} \delta(A_a, A'_{a'})\delta(B_b, B'_{b'}) \tag{2.2}$$

Here $R^{-1}_{r,d}(G)$ yields the neighborhood subgraphs of G with all possible root vertices with vertex neighborhood of r with a distance d. The $\delta(x, x')$ is the *exact matching kernel* whose value is 1 if $x \simeq x'$ (if x and x' are isomorphic) and 0 otherwise. From equation 2.2, 1 is added to the kernel result if both the neighbourhood subgraphs are isomorphic. Hence, $\mathcal{K}_{r,d}$ counts the number of identical neighborhood subgraph pairs. The NSPDK is the sum over all possible radii and distances and is defined as:

$$K(G, G') = \sum_r \sum_d \mathcal{K}_{r,d}(G, G')$$

In practice, it is not efficient to extract all possible features, as the number of features increases exponentially with increasing radius and distance. Hence, upper bounds for the radius (R) and distance (D) are imposed. NSPDK extracts all the subgraph pairs, starting with root vertices separated by a distance $d = 0, ..., D$ and increasing the neighborhood with a radius $r = 0, ..., R$. Figure 2.2 illustrates the NSPDK feature extraction from a graph.

$$K(G, G') = \sum_{r=0}^{R} \sum_{d=0}^{D} \mathcal{K}_{r,d}(G, G')$$

In order to equally weight the different sizes of induced subgraph pairs, $\mathcal{K}_{r,d}$ is normalized as follows:

$$\hat{\mathcal{K}}_{r,d}(G, G') = \frac{\mathcal{K}_{r,d}(G,G')}{\sqrt{\mathcal{K}_{r,d}(G,G)\mathcal{K}_{r,d}(G',G')}}$$

For small values of R and D, NSPDK has linear time complexity in the size of the graph.

Part III

Overview of the individual contributions

Chapter 3

Clustering and classification of small non-coding RNAs

This chapter summarizes the work from the following publication:

- **Pavankumar Videm**, Dominic Rose, Fabrizio Costa, and Rolf Backofen. BlockClust: efficient clustering and classification of non-coding RNAs from short read RNA-seq profiles. *ISMB 2014 proceedings'* special issue in Bioinformatics, 2014 [105].

3.1 Motivation

As discussed in chapter 1, small ncRNAs play a crucial role in distinct cellular processes such as transcription, translation, RNA maturation and splicing. They are also implicated in several diseases. Studies like the ENCODE project revealed that the majority of the transcribed RNA is non-protein-coding [106]. On the basis of this pervasive transcription, there are over 450,000 predicted ncRNAs in the human genome [107]. The functional annotation of these predicted ncRNA transcripts is the subject of ongoing research. Conventionally, small ncRNAs are predicted based on evolutionarily conserved sequence and secondary structure information [108, 109]. However, post-transcriptional modifications during the maturation of ncRNAs result in sequence changes which can also influence their structure [46]. A better alternative that is independent of these issues are the traces of RNA biogenesis that are preserved in the read profiles from high-throughput sequencing data [110]. For example, mapped small RNA-Seq reads belonging to a precursor miRNA typically show two stacks of alignments separated at a certain distance on the reference genome. One stack with a relatively high number of alignments represents the reads from the expressed mature miRNA strand and the other one with a low number of alignments indicate the degraded strand. Figure 3.1A depicts the *miR-1* human miRNA secondary structure on the right side along with its read profile on the left side. The alignments were generated from smRNA-Seq data of the H1 embryonic stem cell line (GSM450239) [111] by mapping the reads against the human reference genome assembly (hg19). The read profile and the precursor miRNA annotation (red bar in the bottom) were taken from the UCSC genome browser [112]. The two miRNA strands *hsa-*

miR-1-5p and *hsa-miR-1-3p* are represented in blue and green colors, respectively. In this particular smRNA-Seq sample, the majority of the reads supports *hsa-miR-1-3p* to be the mature strand. This read profile represents the processed miRNA-miRNA* duplex, as there are no reads near the hairpin and the terminal regions of its precursor miRNA annotation.

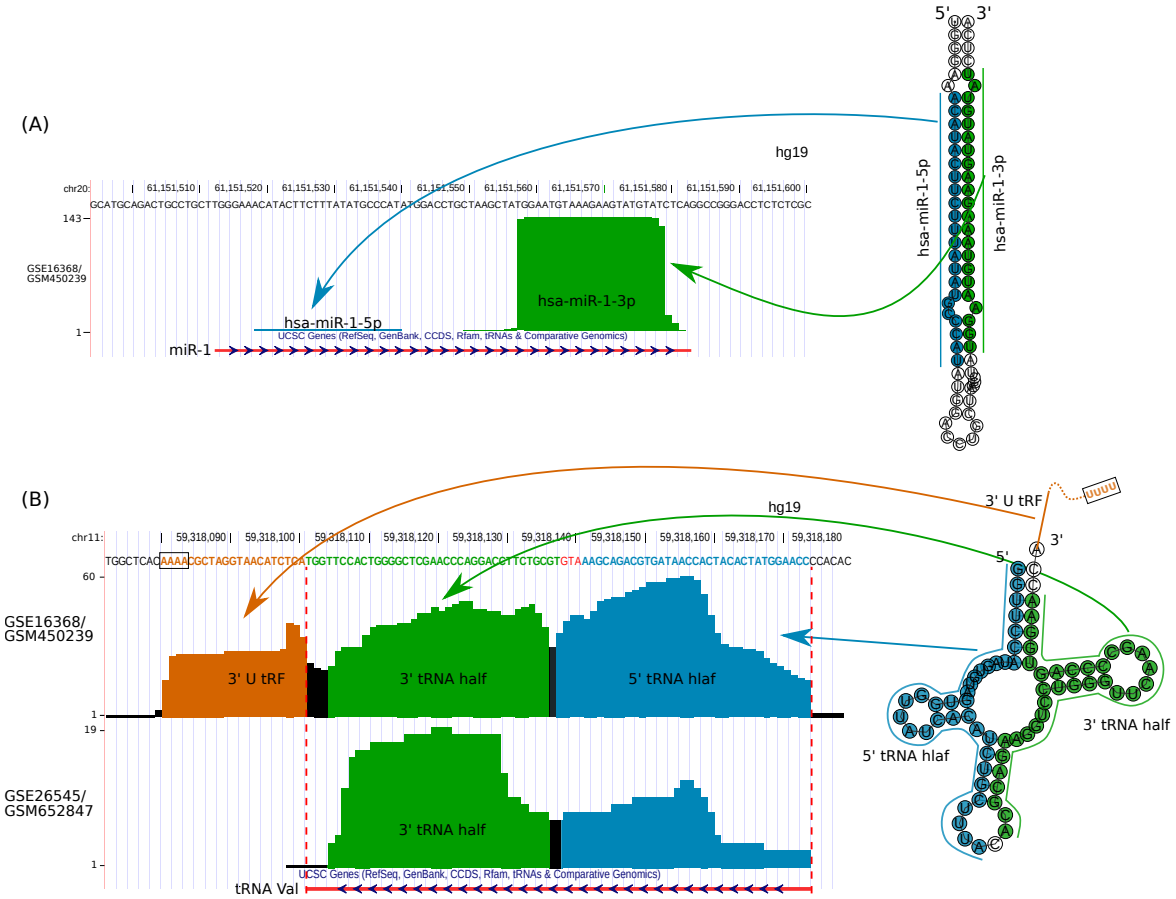


Figure 3.1: (A) Representation of *hsa-mir-1* miRNA read profile and its secondary structure. The expressed 3′ arm (green) is an indication of mature miRNA where as 5′ arm (blue) is degraded and hence not expressed. (B) Alternative processing of human *tRNA Valine* in two different samples. The sample GSM450239 has equally expressed 3′ and 5′ tRNA halves and an additional 3′ U tRF is produced by pre-tRNA. The sample GSM652847 is slightly different with less prominent 5′ tRNA half than 3′ tRNA half and there is no sign of other tRFs.

This interesting observation is not just limited to miRNAs. An example of tRNA processing is shown in Figure 3.1B. It shows the human *tRNA Valine* (anticodon TAC) secondary structure and its read profiles from two different smRNA-Seq data sets, H1 embryonic stem cell line (GSM450239) [111] and cerebellar cortex (GSM652847) [113]. The red bar with blue arrows at the bottom represents the region of the annotated mature tRNA on the reverse strand of the DNA. This tRNA produces nearly equal amount of 3′ and 5′ tRNA halves in GSM450239, whereas more 3′ tRNA half fragments compared to that of 5′ tRNA halfs

present in GSM652847. Additionally, the traces of *RNase Z* cleaved 3′U tRF are evident in GSM450239.

Processing patterns from read profiles were previously used in `DARIO` [114, 115] to classify and in `deepBlockAlign` [116] to cluster the small ncRNAs. Both of these tools utilize the output of `blockbuster`, a tool that groups alignments into *blocks* based on a Gaussian approximation of positional information. It further groups adjacent blocks into clusters which are referred to as *blockgroups* throughout this thesis. In general, each blockgroup corresponds to the read profile of a transcript. For instance, a typical miRNA blockgroup constitutes two blocks where the first block is an abstraction of reads from mature miRNA and the second block is from miRNA*.

`DARIO` built a random forest classifier using a handful of features computed from the relations among reads, blocks, and blockgroups. This classifier is then used to predict the miRNAs, tRNAs, and snoRNAs from the input small RNA-Seq samples. `deepBlockAlign` uses a variant of the Sankoff algorithm to align the blocks among the blockgroups. It computes all pairwise similarities among the blockgroups, and then it defines the clusters from the hierarchical clustering of blockgroups. With a time complexity of $O(n^6)$, `deepBlockAlign` is a computationally expensive algorithm. For a faster and efficient clustering as well as functional annotation of the small ncRNAs, a novel method called `BlockClust` is proposed in this thesis.

3.2 Methods overview

Data preprocessing in `BlockClust` is similar to that of `DARIO` and `deepBlockAlign`, and it also uses `blockbuster` output. Given alignments, `blockbuster` fits Gaussians over the alignments, starting from the position with the most number of aligned reads. Each Gaussian is controlled by the number of reads and a standard deviation for each read. The computed Gaussians are called blocks. Blocks that are at a certain distance are further grouped into blockgroups. Each `blockbuster` blockgroup corresponds to a transcribed small ncRNA. The novelty of `BlockClust` is its representation of the blockgroups in the form of graphs and the usage of graph kernels to identify similar read profiles. Each blockgroup is encoded as a graph with two disconnected components. Refer to Figure 3.2 for a graphical illustration. The first component (COMPONENT 1) is used to represent blockgroup attributes. The second one (COMPONENT 2) is to represent the attributes of the blocks and their relationship to the adjacent blocks. All the computed attributes are then discretized into 3 values (low, mid, and high) using an equal-frequency binning algorithm. The discretized attributes are then used as node labels in the graph. Each rectangle node represents an attribute and the colored circle denotes its discretized value. NSPDK [117] is used to compute the similarity between any two graphs as a fraction of common neighborhood subgraph pairs. Each subgraph here

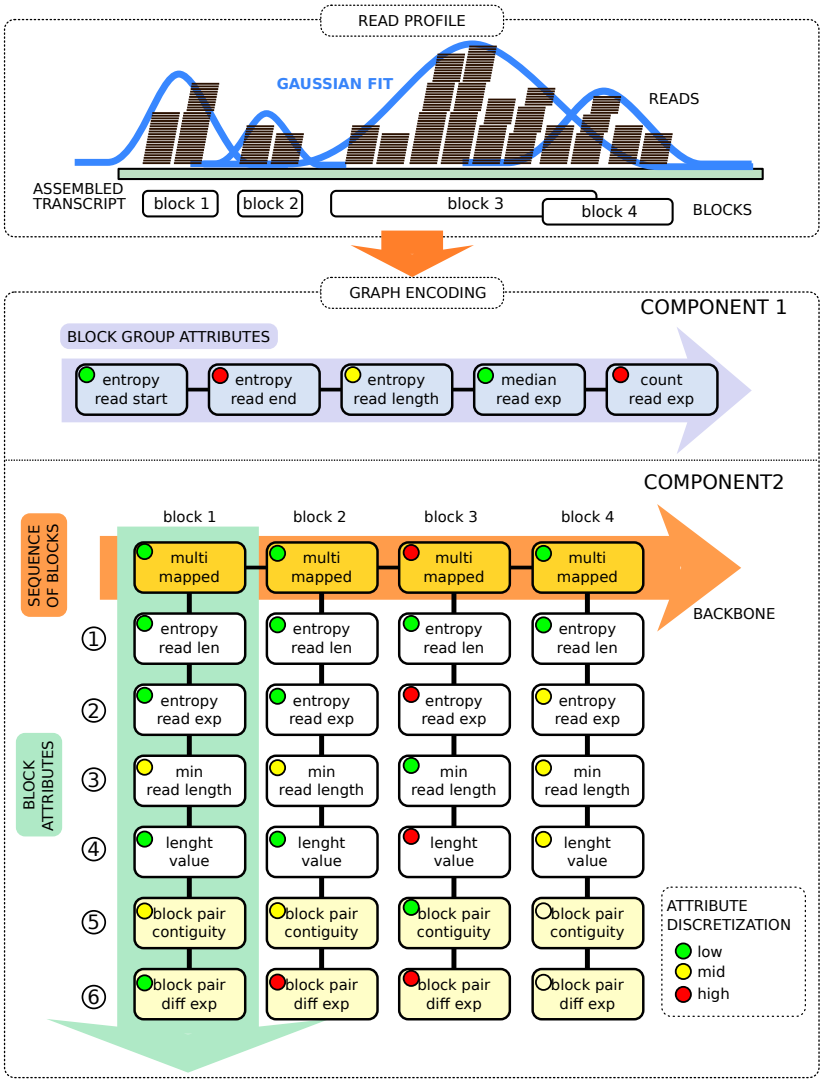


Figure 3.2: Encoding of a read profile as a graph. First, using `blockbuster`, blocks are defined by Gaussian approximation of overlapping alignments, and then neighboring blocks are further grouped into blockgroups. Each blockgroup that describes a small ncRNA transcript is then encoded as a graph with two components. The blockgroup attributes are used in the first component of the graph whereas block-specific attributes are used in the second component of the graph. The discretized values of the attributes build up the vertices. The first row of the block attributes acts as a backbone, starting from which the features are extracted. This figure is adapted from publication *P1*.

makes up a distinct intrinsic feature of a blockgroup. Unlike the handcrafted features in `DARIO`, NSPDK generates a large number of features to evaluate. The size of feature space is dependent on the graph size, radius, and distance parameters. Linear time enumeration of subgraphs by NSPDK makes this graph kernel approach suitable for this large-scale data analysis. NSPDK produces a matrix of pairwise similarities among the blockgroups which is further processed using the Markov Cluster Algorithm (MCL) [118] to generate clusters.

The final clusters represent the blockgroups of similarly processed transcripts, potentially indicating common functions. Additionally, precomputed SVM models available for miRNA, tRNA, and C/D box snoRNA classes can be used to classify the input read profiles.

3.3 Summary of results and discussion

To build and evaluate the SVM models as well as to optimize the tool parameters used in the analysis pipeline, data from human embryonic stem cells, H1 cell line, and IMR90 cell lines were used. In the following, this data is referred to as development data. To assess the robustness of `BlockClust` and compare the performance to the existing tools, benchmark data has been used. This data consists of 32 samples with a variety of cell lines and tissues from human, mouse, fly, chimp, worm, and plants. None of the 32 samples from benchmark data are used in the training process of `BlockClust`. The development data was split into train, validation, and test data sets of non-overlapping portions 35%, 35%, and 30%, respectively. Parameter optimization was done on train and validation sets, whereas the 10-fold classification performance was reported on the test set.

3.3.1 Evaluation of `BlockClust`'s performance

To evaluate the efficiency of `BlockClust`'s similarity notion, the area under the curve for the receiver operating characteristic (AUC-ROC) was computed. To assess the quality of clustering, the cluster purity was used as a metric. Additionally, the precision and recall values achieved by the SVM models of miRNA, tRNA, and C/D box snoRNAs are also reported (Table 3.1). MiRNAs and rRNAs indicate a very high grouping tendency whereas snRNAs and Y_RNAs show a moderate AUC-ROC value. MCL failed to produce any clusters of ncRNA classes with less than 10 transcripts. Except for C/D box snoRNAs, clusters of all other classes show a very high purity of at least 83%. Apart from low sensitivity in classification of C/D box snoRNAs, the SVM models produce high precision of approximately 0.9 for all three classes and good recall values for miRNA and tRNA classes.

3.3.2 Comparison with the other existing methods

A different but sophisticated way of evaluating `BlockClust` is to compare it with other established methods on the same data sets. The clustering performance was compared to `deepBlockAlign` and the classification performance to `DARIO`. The clustering comparison was carried out on the whole benchmark data of 32 samples. Owing to the lack of a standalone `DARIO` tool, one of the benchmark data sets (GSM769510) has been used to compare the predictive performance.

		Similarity notion	Clustering		Classification	
ncRNA class	**#transcripts**	**AUC ROC**	**#clusters**	**cluster purity**	**Precision**	**Recall**
miRNA	168	0.896	10	0.855	0.901	0.886
tRNA	173	0.741	17	0.837	0.899	0.796
C/D box snoRNA	78	0.731	7	0.683	0.870	0.474
rRNA	20	0.872	2	0.956	NA	NA
H/ACA box snoRNA	4	0.838	0	0	NA	NA
snRNA	7	0.637	0	0	NA	NA
Y_RNA	8	0.685	0	0	NA	NA

Table 3.1: Evaluation of similarity notion, clustering, and classification performance of `BlockClust` on development data test set. NA symbolizes the absence of an SVM model.

Similarity notions of `BlockClust` and `deepBlockAlign` were evaluated by computing AUC-ROC on their respective similarity matrices. Table 3.2 summarizes the AUC-ROC values of 7 ncRNA classes computed on 32 benchmark data sets, and an average that is normalized over the number of transcripts per class. Clearly, the results indicate that `BlockClust`'s graph similarity method is more accurate than `deepBlockAlign`'s block alignment method for any given ncRNA class. With an average AUC-ROC of 0.84 on benchmark data comprising a large variety of species, tissues, and cell lines, `BlockClust` turned out to be a quite reliable approach to cluster read profiles from smRNA-Seq data sets. Owing to the quasi-linear time complexity of NSPDK, `BlockClust` could achieve a 60-fold speed-up in processing a data set of 600 blockgroups in 50 seconds compared to 58 minutes using `deepBlockAlign`.

ncRNA class	**#transcripts**	`BlockClust` **AUC ROC**	`deepBlockAlign` **AUC ROC**
miRNA	3869	0.925	0.714
tRNA	4988	0.795	0.701
C/D box snoRNA	731	0.762	0.615
H/ACA box snoRNA	142	0.859	0.720
rRNA	770	0.873	0.759
snRNA	240	0.698	0.610
Y_RNA	244	0.694	0.656
Weighted average	11061	0.839	0.700

Table 3.2: Comparison of similarity notion of `BlockClust` and `deepBlockAlign` on benchmark data. `BlockClust`'s similarity metric results in a better AUC-ROC than `deepBlockAlign` for any ncRNA class. The exceptional performance of `BlockClust` on the comprehensive benchmark data with a variety of species, tissues and cell lines proves its robustness.

The clustering performance has been assessed by inspecting the clusters generated by

MCL from the similarity matrices of `BlockClust` and `deepBlockAlign` on the data set GSM450239. The *inflation* and *pre-inflation* parameters of the MCL affect the clusters size and their elements. Hence, for a fair comparison, the clustering was performed by varying *inflation* and *pre-inflation* parameters on a wide range of values. Figure 3.3A shows the number of clusters generated in each parameter setting on the x-axis and the median of precision over the clusters on y-axis. The red dots correspond to the `BlockClust` and the blue dots to the `deepBlockAlign`. Generally, with the increasing number of clusters, the average cluster size decreases. At the same time, there is a high probability that smaller clusters constitute read profiles of the same ncRNA class. In the worst-case scenario, each individual read profile forms a cluster, but obviously the median precision in this case is 1.0. Therefore, it is important to keep a nice balance between cluster size and precision. For any parameter setting, `BlockClust`'s similarity notion did not result in more than 35 clusters and precision was always higher than that of `deepBlockAlign`.

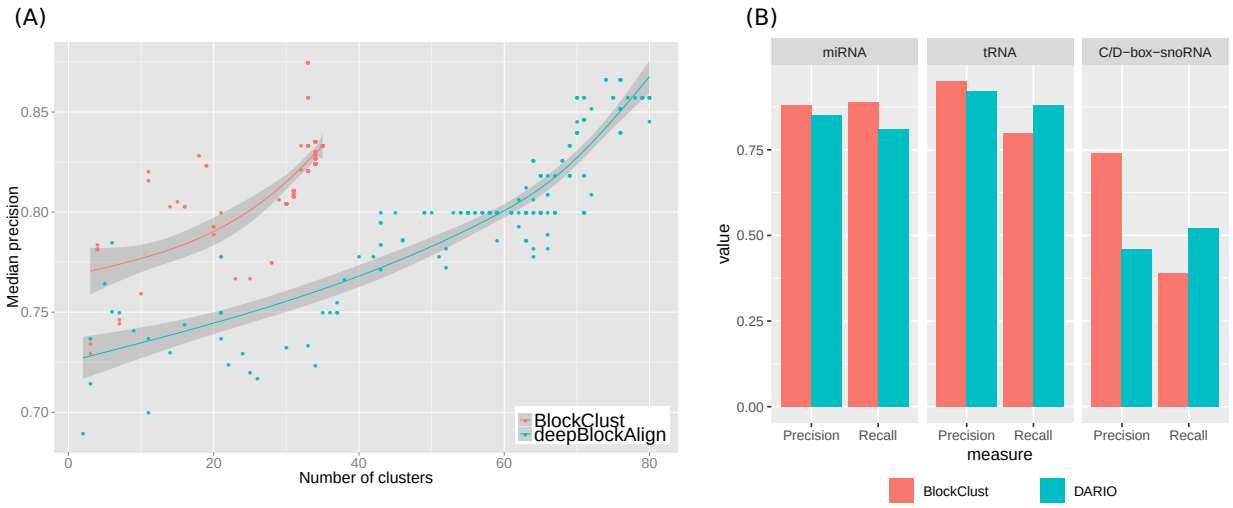


Figure 3.3: Comparison of clustering and classification performance. (A) MCL generated clusters from `BlockClust` and `deepBlockAlign`. For any cluster size, `BlockClust` clusters show a higher median of precision than `deepBlockAlign`. (B) Classification performance on 3 most abundant small ncRNA classes. Overall, the `BlockClust` shows better precision and recall compared to `DARIO`. This figure is adapted from publication *P1*

Figure 3.3B shows the classification performance of `BlockClust` and `DARIO` on data set GSM769510. `DARIO` showed better recall values for tRNAs and snoRNAs, but overall, `BlockClust` is better at predicting small ncRNAs.

In summary, with its class-specific discriminative attributes and novel way of encoding read profiles as graphs, `BlockClust` efficiently clusters as well as predicts the functional non-coding RNA classes. The quality of `BlockClust`'s similarity notion and classification models are comparably better and significantly faster than the existing methods. `BlockClust` can be considered as a complete package for predicting small ncRNAs of known classes or potentially

cluster novel ncRNAs based on their processing. To serve a broad range of users, `BlockClust` is available as a command line tool as well as a Galaxy tool. Separate ready-to-use Galaxy based workflows for clustering and classification are also available in the Galaxy tool shed. More information about the Galaxy tools and workflows can be found in section 5.3. The complete list of `BlockClust` predictions on benchmark data and links to their read profiles on UCSC genome browser are accessible at https://pavanvidem.github.io/blockclust_predictions/.

Chapter 4

Analysis of RNA-RNA interactions from high throughput sequencing data

---

This chapter is a summary of work from the following publications:

- Stefan C Weise, Ganeshkumar Arumugam, Alejandro Villarreal, Pavankumar Videm, Stefanie Heidrich, Nils Nebel, Verónica I Dumit, Farahnaz Sananbenesi, Viktoria Reimann, Madeline Craske, et al. FOXG1 regulates PRKAR2B transcriptionally and posttranscriptionally via mir200 in the adult hippocampus. *Molecular neurobiology*, 2019 [119].
- Pavankumar Videm, Anup Kumar, Oleg Zharkov, Björn A. Grüning, and Rolf Backofen. ChiRA: an integrated framework for chimeric read analysis from RNA-RNA interactome and RNA structurome data. *GigaScience*, 2021. [120]

4.1 Motivation

As mentioned in Section 1.2.2, RNA-RNA interactions are essential for cellular functions, particularly in gene regulation. Generally, genes are regulated through a complex network of RNA interactions. In addition to the direct RNA-RNA interactions, the secondary effects of RNA interactions can also lead to significant changes in the expression profile of the whole genome. In this chapter, two different use cases of the small ncRNA interaction analysis are addressed. The first use case involves the analysis of RNA-Seq data to study a very specific case of the *miR200* family interactions that play a key role in a complex neurodevelopmental disorder. The second use case deals with the analysis of the genome-wide direct RNA-RNA interactions from the RNA-RNA interactome experiments. It involves overcoming the challenges in several stages of miRNA interactome data analysis. The presented method is applicable for AGO-mediated miRNA interactome protocols like CLASH and CLEAR-CLIP as well as for genome-wide RNA-RNA interactome protocols like PARIS and SPLASH. Therefore, section 4.3 will be addressed as a generic RNA-RNA interactome analysis workflow.

4.2 MicroRNA interaction network analysis from RNA-Seq data

Rett syndrome is one of the major neurodevelopmental disorders that affects brain development in females. Typically, it is caused by a genetic mutation in the X-chromosomal gene methyl CpG binding protein 2. Less common causes are deficiency of cyclin-dependent kinase-like 5 or mutations in Forkhead box G1 (*FOXG1*). This project investigates the effects of the *FOXG1* in one of the atypical Rett syndromes. This study indicates that an overexpression of protein kinase type II-beta regulatory subunit (*PRKAR2B*) can possibly lead to Rett syndrome. *PRKAR2B* is a target of *miR200* family whose biogenesis is influenced by *FOXG1*.

4.2.1 Methods overview

4.2.1.1 Differential gene expression analysis

The *FOXG1* gene was knocked out ($Foxg1^{cre/+}$) from 6-week-old mice, the total RNA was extracted and ribosomal RNA was depleted. Then two separate (i) RNA-Seq and (ii) small RNA-Seq libraries were prepared and sequenced. From the analysis of the small RNA-Seq data, it was evident that $Foxg1^{cre/+}$ altered the expression of the *miR200* family (*miR200a*, *miR200b*, and *miR429*). To identify the putative targets of the *miR200* family an additional RNA-Seq experiment was performed, where the *miR200* family genes in $N2a$ cells were overexpressed (OE). All the RNA-Seq libraries were sequenced in a paired-end layout.

Most of the data analysis was carried out on the Galaxy platform [121]. Quality control on raw data was assessed using `FASTQC` [87]. The 3′-end bases with *Phred* score less than 28 were trimmed using `TrimGalore` [122]. Then a splice-aware aligner `TopHat2` [97] was used to map the reads to mouse genome build *mm10*. For $Foxg1^{cre/+}$ samples, the sum of cDNA fragment length and adapter length is less than the number of sequencing cycles. Consequently, the sequenced mate pairs are overlapping. For an accurate mapping of mate pairs, `-mate-inner-dist` was set to 0 and `-mate-std-dev` was set to 80. For all samples, `-library-type` was set to "fr-firststrand". Using `htseq-count` [99] and the gene annotation model from Ensembl release 79 [123] gene abundances were measured. Finally, differential gene expression analysis was carried out using `DESeq2` [124] by comparing $Foxg1^{cre/+}$ and *miR200* OE to their corresponding wild type samples. Genes with an adjusted p-value of 0.05 or less have been considered as significantly differentially expressed.

4.2.1.2 Transcription factor analysis

FOXG1 is a transcription factor that plays an important role in the mammalian embryonic telencephalon progenitor cell cycle [125, 126]. Therefore, its effect on transcription within the *miR200* family gene cluster has been investigated. First, the mouse FOX transcription factor binding motifs from JASPAR database [127] have been obtained as position weight matrices. Then `FIMO` [128] was used to search for individual motif occurrences in the promoter region (defined as the 1000*nt* upstream) of the *miR200* gene cluster.

Additionally, published *FOXG1* ChIP-Seq data from cortical tissue (at `http://www.activemotif.com/catalog/details/61211/foxg1-antibody-pab`) was used to identify putative *FOXG1* DNA binding sites. The data was obtained in BAM format, having only a single replicate and no control samples. For sensitive detection of binding sites, two tools were used for peak calling, namely `MACS` [129], and `findPeaks` from `HOMER` tool-suite [130]. The peaks produced by `MACS` are significantly broader compared to that of `HOMER`. Hence, all the `MACS` peaks that are completely overlapped with `HOMER` peaks were considered as the putative *FOXG1* binding sites.

4.2.2 Summary of results

The first major finding of this project is the regulation of the *miR200* family by *FOXG1*. The small RNA-seq of $Foxg1^{cre/+}$ revealed 10 significantly differentially expressed (DE) miRNAs. With at least 2-fold downregulation, *miR200* family genes are the most significantly altered among all the small RNAs. Previous studies suggested that the *miR200* family is a key actor in neural progenitor proliferation [131, 132]. In Figure 4.1, each bar represents the mean of fold changes calculated from qRT-PCR cycle threshold (CT) values along with the standard error of the mean (SEM). *miR200* family members are colored in black.

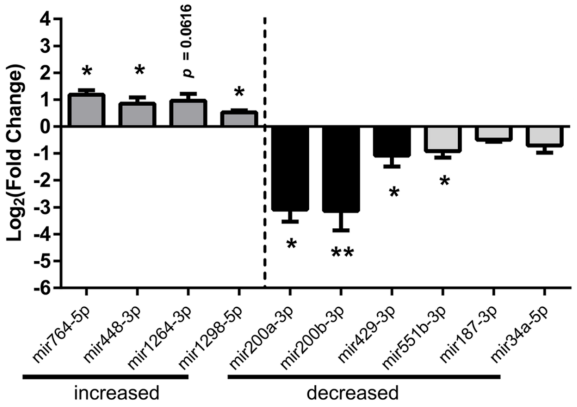


Figure 4.1: qRT-PCR validation of differentially expressed miRNAs in $Foxg1^{cre/+}$ small RNA-Seq. *miR200* family showed more than 2-fold downregulation in $Foxg1^{cre/+}$ samples. A single star (*) denotes a p-value<0.05, whereas a double star (**) signifies p<0.01. This figure is taken from publication *P2*.

Transcription factor analysis indicated no significant *FOX* motifs in the promoter region of the miRNA gene family cluster. Following this observation, the data from RNA-Seq experiments of (i) $Foxg1^{cre/+}$ and (ii) *miR200* OE in *N2a* cells have been an-

alyzed. The rationale behind this is to find the putative targets of the *miR200* family that have significant functions in brain development and that ultimately lead to the Rett syndrome. There are 382 and 2081 significantly DE genes in $Foxg1^{cre/+}$ and *miR200* OE RNA-Seq data sets, respectively. Out of these, the most interesting candidates are 35 genes that are DE in both the RNA-Seq experiments (Figure 4.2a). With a total of 43,629 annotated genes, a hypergeometric test on two independent gene sets of 382 and 2081 with an overlap of 35 genes resulted in a p-value of 0.0001 that signifies the overlap. Among these 35 genes, 25 were downregulated due to an OE of the *miR200* family, implying the putative direct targets of the *miR200* family genes. 12 out of 25 downregulated genes showed an

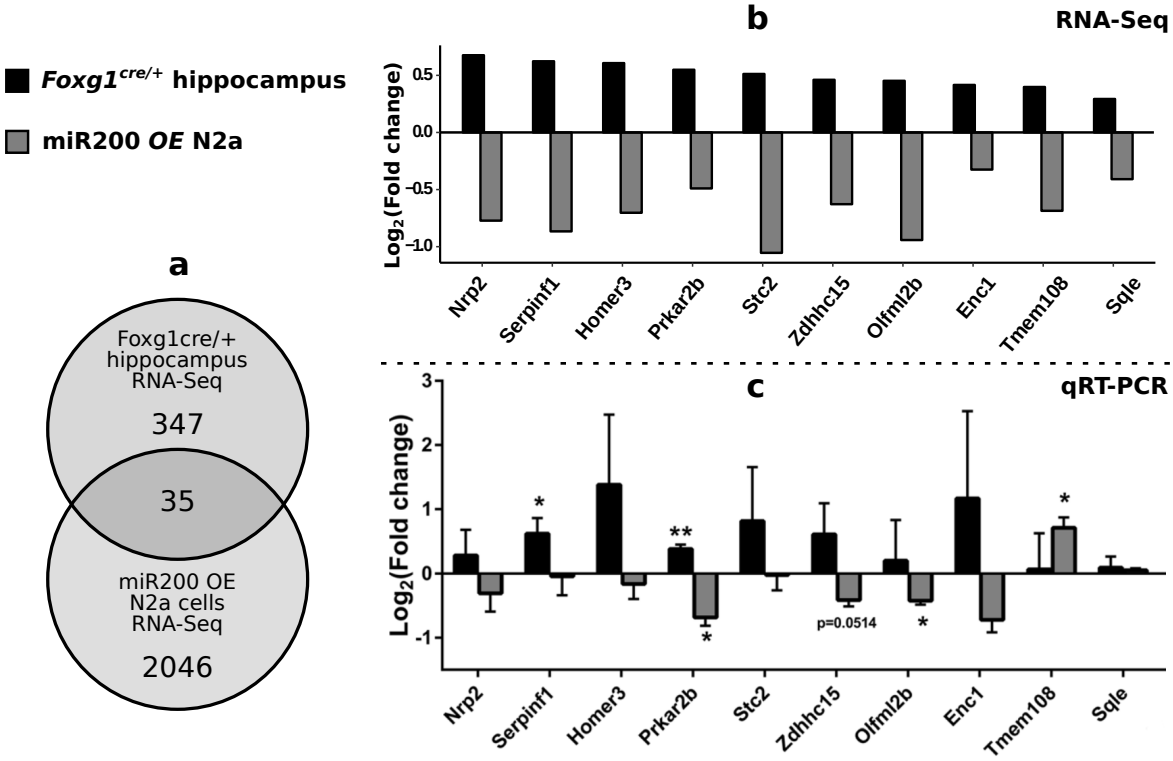


Figure 4.2: RNA-Seq analysis of $Foxg1^{cre/+}$ and *miR200* family OE indicates *PRKAR2B* as target of *miR200* family. (**a**) Overlap of significantly DE genes between both RNA-Seq experiments. (**b**) Log_2 fold changes calculated using separate DESeq2 analysis on $Foxg1^{cre/+}$ and *miR200* OE RNA-Seq data sets. Only the upregulated genes from the overlap set are shown. (**c**) qRT-PCR validation showed significant fold change of *PRKAR2B* in both conditions. Figures a and c are adapted from publication *P2*.

upregulation in $Foxg1^{cre/+}$, which further implies a possible secondary effects of *FOXG1* knockout. Figure 4.2b shows the log_2 fold change of 10 downregulated genes from the intersection. The log_2 fold changes were calculated by DESeq2 for both $Foxg1^{cre/+}$ (black bars) and *miR200* OE (gray bars) RNA-Seq data sets. The qRT-PCR validation of these genes (shown in Figure 4.2c) revealed *Serpinf1*, *PRKAR2B*, *Olfml2b*, and *Tmem108* with significant p-values in one of the RNA-Seq experiments. *PRKAR2B* is the only candidate that

has a p-value < 0.05 in both conditions. Furthermore, *PRKAR2B* is the only common gene that was predicted to be a target of *miR200b* and *miR429* by the miRNA target prediction algorithms TargetScan [133] and miRanda [134]. This analysis indicated that *PRKAR2B* is a target of the *miR200* gene family in the hippocampus. *PRKAR2B* is type II regulatory subunit gene of protein kinase A (*PKA*) complex. The right balance of *PKA* activity is the key to memory formation [135, 136]. The ChIP-Seq analysis that was initially aimed at finding the *FOXG1* binding at *miR200* family primary transcript showed no such evidence. Instead, interestingly, a peak was identified in the promoter region of the *PRKAR2B*. Later, an experimental validation through luciferase reporter assay revealed a direct regulation of *PRKAR2B* by *FOXG1*, in addition to the *miR200* family mediated regulation.

As a follow-up experiment, co-immunoprecipitation and quantitative mass spectrometry of overexpressed *FOXG1* in N2a cells resulted in *FOXG1* interaction with DDX5, which is a post-transcriptional gene expression regulator. Within the nucleus, this *FOXG1-DDX5* complex further interacts with the microprocessor complex which affects the maturation of the *miR200* family. Hence, mutations in *FOXG1* alter the expression of the *miR200* family through interacting with DDX5 and the microprocessor complex. Being a direct target of the *miR200* family, *PRKAR2B*'s expression is also altered. This imbalanced expression levels of *PRKAR2B* affects memory formation, causes neuronal dysfunction and ultimately, an atypical Rett syndrome.

4.3 Framework for analysis of direct RNA-RNA interactions from RNA-RNA interactome protocols

The evolution of high-throughput sequencing technologies led to rapid advancements in RNA-RNA interactome protocols in the past decade. Consequently, several databases have emerged [137, 138, 139] that compile and present RNA interactions from RNA interactome protocols and computational prediction methods such as TargetScan [133] and miRDB [140]. Each of the collected interactions are given with a confidence score based on experimental evidence, interaction frequency, and associations from literature mining. However, the interactions from RNA interactome protocols are taken straight from the published studies that were analyzed using different bioinformatics pipelines. A common scoring scheme for the interactions that were generated from different pipelines might affect the reliability of the ranking. An underlying reason for this could be the lack of an easy-to-use bioinformatics pipeline that applies to different RNA-RNA interactome protocols. Either it is a protein-mediated protocol (such as CLASH and CLEAR-CLIP) or protein-independent genome-wide RNA-RNA interaction detection protocol (such as PARIS and SPLASH), the sequenced reads are chimeric, representing intra- and intermolecular interactions. `ChiRA`, a bioinformatics tool

suite that is developed as a part of this thesis, and analyzes both types of interactome data. `ChiRA` provides solutions for two related problems in chimeric read analysis: (i) handling multi-mapped reads and (ii) correct annotation of the chimeric read segments. Based on the multi-mapped reads, it can effectively group reference loci that belong to gene families, isoforms and gene paralogs without requiring any prior annotation. Quantification of the reference loci helps in inferring the true origin of chimeric read segments.

4.3.1 Methods overview

A `ChiRA` Galaxy workflow has been built based on different tools in the `ChiRA` tool suite. Figure 4.3 is the illustration of different steps in `ChiRA` workflow. Each step in the workflow deals with a challenge in RNA-RNA interactome data analysis. Owing to short insert sizes, most of the reads from miRNA interactome experiments contain adapters. Therefore, the workflow starts with adapter clipping and low-quality base trimming using `cutadapt` [91]. Then the read deduplication step potentially reduces the number of reads by an order of magnitude and consequently speeds up the subsequent steps. The workflow also allows UMI-based read deduplication. After deduplication, reads are mapped to a reference genome/transcriptome through local alignment. A general-purpose local aligner `BWA-MEM` [95] or a chimeric read specific aligner `CLAN` [98] can be used for mapping. Mapping is the crucial step that possibly influences the composition of the interactions. In certain cases, a single mismatch allows aligning a read segment to multiple genes of a miRNA family. Hence, `ChiRA` used less stringent parameters to generate sub-optimal alignments for each read and processes them further to select the most probable alignment. For protein-mediated RNA-RNA interactions, often the characteristics of interacting RNAs are known. For example, AGO-mediated CLASH experiments predominantly produce miRNA-involved interactions. To comply with such protocols, the workflow accepts a “split reference”, i.e., two different reference FASTA files. For AGO CLASH or CLEAR-CLIP data, it is quite rational to provide a split reference of separate FASTA files containing (i) miRNAs and (ii) the rest of the transcriptome.

The next step in the workflow is to merge the significantly overlapping reference locations to generate "expressed loci", which are potential interaction sites. `ChiRA` offers two modes of merging, aimed at different sequencing depths. For samples with low to medium coverage, a simple overlap-based merging is appropriate whereas for high coverage samples, `blockbuster`-based merging is optimal. Similarly, aligned portions of the reads are merged to define “read segments”. A read with only a single mapped segment is classified as "singleton read" whereas a read with two non-overlapping mapping segments is classified as "chimeric read".

After merging, the workflow deals with the multi-mapped reads. Choosing the correct

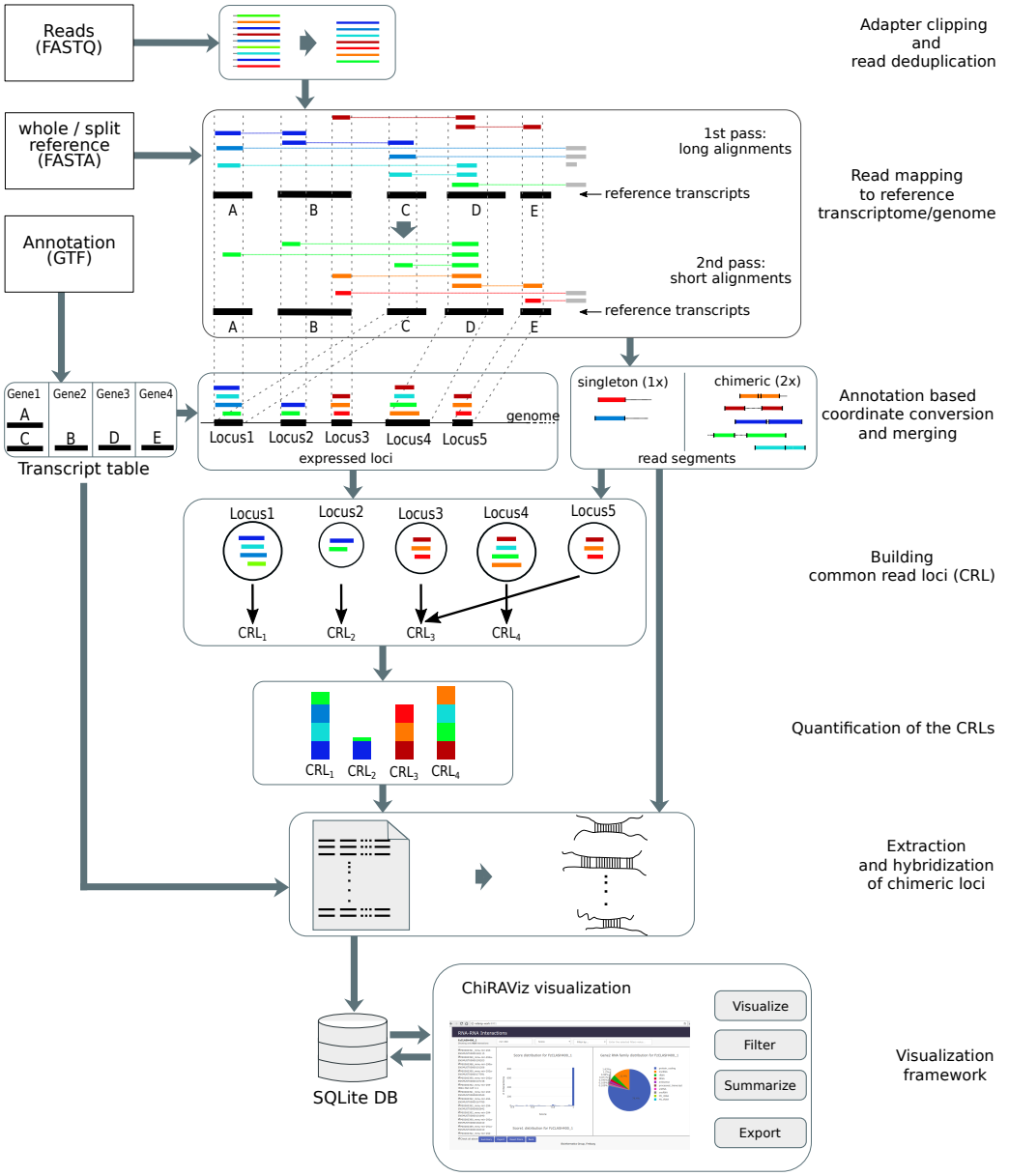


Figure 4.3: `ChiRA` workflow. First, the de-duplicated reads are mapped twice with different parameters to capture long and then short alignments. Then interaction sites are detected by merging the overlapping reference regions. Based on aligned read portions, chimeric split points are defined on each read. The expressed loci with consistently multi-mapped reads are then grouped into a common read loci. Then using an EM algorithm, the common read loci are quantified. Finally, the interactions are scored, hybridized using `IntaRNA`, and compiled into an SQLite database. `ChiRAViz` can be used to search, filter, visualize, and export the interactions. This figure is taken from publication *P3*.

alignment for a read segment that is multi-mapped to gene families or gene paralogs is a challenging task. It has already been shown that for RNA-Seq quantification it is sensible to consider reads that are consistently multi-mapped to gene families as uniquely mapped [141].

However, for most organisms, it is difficult to find a reliable annotation with information on gene paralogs or gene families. Hence, `ChiRA` tool suite provides an approach to cluster the reference loci that potentially belong to gene families without requiring any information on gene relations. Each such cluster of loci is called a common read loci (CRL). Starting with the highly expressed locus, using single linkage clustering, each expressed locus is merged into a CRL if it is similar to that CRL. The similarity between a CRL and an expressed locus is measured as the Jaccard index on aligned read segments of that CRL and the expressed locus. The resulting CRLs contain the loci that share common reads. If a locus has only uniquely-mapped reads or is not sharing a significant portion of multi-mapped reads with any other CRL, then it makes a new CRL. Algorithm 1 presents the pseudo-code for CRL generation by `ChiRA`.

Algorithm 1: `ChiRA`'s CRL creation based on expressed loci.

```
ℂ ← {} // set of CRLs
/* 𝕃 is the expressed loci                                        */
for L ∈ 𝕃 do
    match ← False;
    /* L_i is the set of read segments of an expressed locus L    */
    /* C_k is the set of read segments of a CRL C                 */
    for C ∈ ℂ do
        if (C_k ∩ L_i) / (C_k ∪ L_i) ≥ θ then
            C_k ← C_k ∪ L_i;
            C ← C ∪ L // add locus to an existing CRL
            match ← True
        end
    end
    if not match then
        ⋃{ℂ, L} // for a locus not sharing reads, create a new CRL
    end
end
```

The next step in the workflow is the quantification of the CRLs. Quantification allows the correct annotation of the read segments in the case of multi-mapping. Since the CRLs represent highly identical loci, all multi-mappings of the reads within a CRL are considered as unique mappings. The expression levels of the CRLs are estimated by using an expectation-maximization (EM) algorithm.

Let $\mathbb{S}$ be the set of read segments. The abundance of a CRL $c \in \mathbb{C}$ is estimated by computing the likelihood that a read segment s originate from a CRL c i.e., $\rho_c = Pr[s \in c]$.

An indicator matrix $Z = (z_{s,c})_{\substack{s\in\mathbb{S}\\ c\in\mathbb{C}}}$ denotes the selection of read segments from CRLs where each entry

$$z_{s,c} = \begin{cases} 1 & \text{if read segment } s \text{ originate from CRL } c \\ 0 & \text{else} \end{cases}$$

Each row Z^s of the above matrix contains exactly one entry with 1 representing the CRL that the segment s is stemming from. For uniquely mapped read segments, the matrix entries are evident but not for the read segments that are mapped to multiple CRLs. Since a read cannot be originated from more than a single CRL, another indicator matrix $Y = (y_{s,c})_{\substack{s\in\mathbb{S}\\ c\in\mathbb{C}}}$ is introduced with read segment mapping information.

$$y_{s,c} = \begin{cases} 1 & \text{if read segment } s \text{ maps to CRL } c \\ 0 & \text{else} \end{cases}$$

Unlike Z matrix, each row Y^s may contain multiple entries with 1 representing multi-mapping. With $\hat{\rho}$ as a vector of all ρ_c, the likelihood of the read segment origin $\mathcal{L}(\rho)$ is defined as follows:

$$\mathcal{L}(\rho) = \prod_s \sum_c y_{s,c}\rho_c.$$

EM algorithm has been used to solve this. Let $\rho^{(t)}$ be the vector of abundance estimates in t^{th} iteration of the EM-algorithm. In E-step, the expected values of the hidden variables (Z matrix) are determined as follows:

$$\begin{aligned} E[z_{s,c} \mid Y, \rho^{(t)}] &= Pr[z_{s,c} = 1 \mid \rho^{(t)}, Y] \\ &= \frac{\rho_c^{(t)}}{\sum_{c'} y_{s,c'}\rho_{c'}^{(t)}} \end{aligned} \tag{4.1}$$

From the hidden values of Z matrix, the M-step estimates the maximum likelihoods as follows:

$$\rho_c^{(t+1)} = \frac{\sum_s z_{s,c}^{(t+1)}}{|\mathbb{S}|} \tag{4.2}$$

The E and M steps are repeated until there are no changes in relative abundances of CRLs in 2 consecutive iterations, i.e., $\sum_{c=1}^{N} |\rho_c^{(t+1)} - \rho_c^t| \leq \varepsilon$. By default, a very low value of $1e^{-5}$ is used for ε.

Z matrix entries of the last iteration, $Pr[z_{s,c} = 1 \mid \hat{\rho}, Y]$ (which we calculated in equation (4.1)) are used to score the alignments in case of multi-mappings. Finally, the probability that a chimeric read with read segments $..s..s'..$ originated from the interaction between CRLs c and c' is denoted by

$$Pr[(s, s') \in c \leftrightarrow c'] = Pr[z_{s,c} = 1 \mid \hat{\rho}, Y] Pr[z_{s',c'} = 1 \mid \hat{\rho}, Y]$$

Based on the relative expression levels, the alignments are scored and the chimeric reads are annotated. After quantification, `IntaRNA` is used to hybridize the sequences of the loci

involved in the interaction. If gene annotation of the reference organism is provided, the final output file is also annotated with useful information like gene ids, gene symbols, biotypes, the regions of interaction (like CDS, UTRs), etc.

4.3.2 Summary of results

To evaluate the overall performance of `ChiRA` workflow, the published benchmark data sets of `CLAN` have been used. Each of these artificial data sets imitates the CLASH experimental data. Each read is a fusion of two chimeric arms containing a mature miRNA sequence and a random TargetScan target sequence. There are 5 samples with chimeric arms of lengths 10, 12, 15, 18, and 20nts and each containing 1 million reads (called as *noInsert*). In real CLASH experimental data, often there are short randomly floating sequences non-deliberately incorporated into the interactions during library preparation. To simulate this, 5 more samples were prepared from *noInsert* samples by incorporating a random 5nt sequence either in between or at the ends of the chimeric arms (referred to as *Insert*).

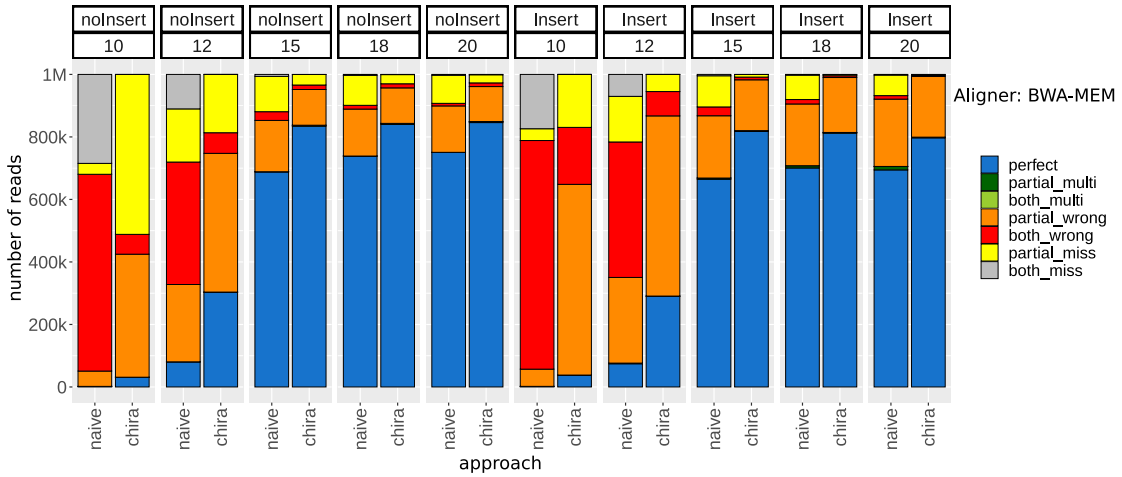


Figure 4.4: Performance of `ChiRA` workflow compared to *naive* mode on benchmark data. For any arm length, `ChiRA` could produce at least 10% more perfect hits (blue) compared to *naive* mode. This is an indication that `ChiRA`'s strategy is better at selecting the correct alignment from the sub-optimal alignments than simply choosing the longest alignment. This figure is taken from publication *P3*.

There are two modes on which the performance is evaluated, namely *naive* mode and *chira* mode. In *naive* mode, reads are mapped using `BWA-MEM` with sensible alignment parameters and the longest alignment is chosen in case of multi mapping. In *chira* mode, the complete `ChiRA` workflow is used with the same alignment parameters as in *naive* mode, but the best alignment for each multi mapped read is chosen based on the final alignment scores. The performance results on overall reads of benchmark data using `BWA-MEM` as the aligner are summarized in Figure 4.4. The blue bars ("perfect" reads) denote the number of reads

in agreement with the ground truth. Although *chira* mode produced a 3-fold increment in "perfect" reads for arm lengths $10nt$ and $12nt$, they are still less than 40%. This observation suggests that capturing alignments shorter than $15nts$ results in ambiguous or wrong alignments. With arm lengths of $15nt$ and above, all the runs on *noInsert* data produced at least 70% of "perfect" hits. The *chira* mode produced at least 10% more perfect hits in all cases. The superior performance of *chira* mode compared to *naive* mode signifies that the longest alignment is not necessarily the true alignment. Quantification of CRLs and alignment scoring by `ChiRA` results in the accurate annotation of the chimeric arms.

From the procedure of CRL creation, it is evident that loci belonging to each CRL have a large portion of reads in common. Furthermore, the reference loci tend to be very similar. To analyze the sequence identities of the loci belong to CRLs, a total of 46 published data sets from CLASH, CLEAR-CLIP, PARIS, and SPLASH protocols have been used. The average pairwise sequence identities (APSIs) of all the loci within the CRLs for each data set have been computed. As a control for each CRL size, the reference loci have been randomly sampled. Figure 4.5A shows the APSIs of CRLs over all the samples (green) in each protocol along with APSIs of the randomly sampled reference loci (blue). With all the medians well above 90%, it indicates that CRLs are comprised of highly identical reference sequences.

Further investigation of the genes that constitute the CRLs revealed more remarkable results. This analysis was also carried out on the above-mentioned data. For annotation of the genes, Rfam family, Ensembl protein family, and KEGG pathway information have been obtained from the Ensembl biomart [142]. Then for each CRL, calculated the percentage of genes belong to the same gene family or share the same pathways. Again, as a control for each CRL, genes have been randomly sampled. The results are plotted in Figure 4.5B. Box plots in green color are presenting the CRLs by `ChiRA`, whereas the box plots in blue color are from randomly sampled genes. The y-axis represents what percentage of genes associated with the CRLs share a common KEGG pathway or belong to the same protein family. Compared to random sampling, the genes of the CRLs more often belong to the same protein family or KEGG pathway. These observations indicate that the CRL creation by `ChiRA` is a process to group reference loci into biologically relevant clusters independent of any annotation.

To measure the sensitivity of `ChiRA` in detecting new interactions, published interactions from CLASH and CLEAR-CLIP have been used. The published interactions of CLASH were generated by a bioinformatics pipeline called `hyb` [143], whereas CLEAR-CLIP interactions were generated using custom-written scripts. Re-analysis of CLASH and CLEAR-CLIP data using `ChiRA` workflow and comparison to the published interactions (Figure 4.6) showed that a large portion of published interactions (83% for CLASH and 73% for CLEAR-CLIP) could be detected by `ChiRA`. In addition, `ChiRA` detected several interactions that were previously not published.

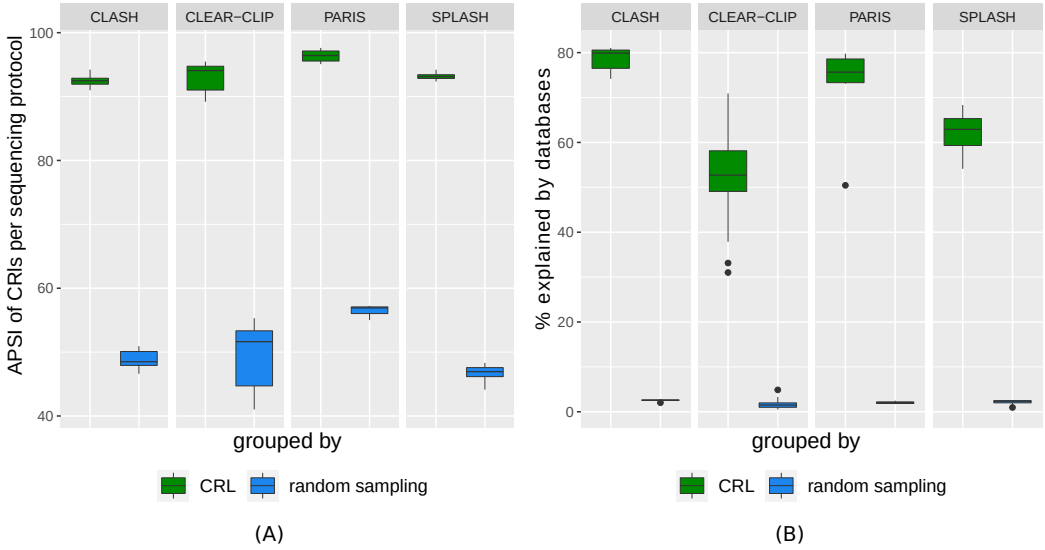


Figure 4.5: (A) Protocol-wise average pairwise sequence identities (APSIs) of reference loci belong to CRLs compared to randomly sampled reference loci. With a very high APSI, loci belonging to the CRLs are very identical. (B) Validation of the CRLs based on RNA family, Ensembl protein family and KEGG pathway information suggest that most of the genes associated with the CRLs are biologically relevant. This figure is adapted from publication *P3*.

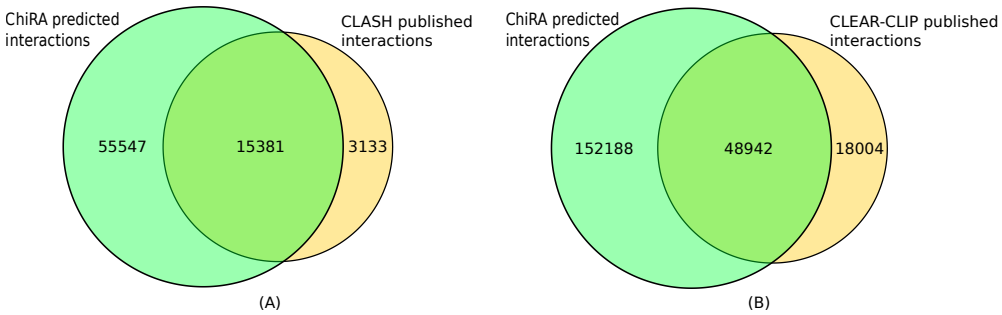


Figure 4.6: Number of interactions that were detected by `ChiRA` compared to published interactions in (A) CLASH and (B) CLEAR-CLIP data sets. `ChiRA` detects novel miRNA interactions from published miRNA interactome datasets. This figure is adapted from publication *P3*.

In conclusion, the `ChiRA` tool suite provides a complete analysis framework for RNA-RNA interactome datasets. Without requiring any information on gene relations, `ChiRA` can group reference loci that belong to gene families of gene paralogs. A ready-to-use Galaxy-based workflow, along with training material, helps the users to learn, understand and easily analyze large RNA interactome datasets.

Chapter 5

Fostering sustainable bioinformatics research

This chapter is based on the work from the following publication:

- Jörg Fallmann, Pavankumar Videm, Andrea Bagnacani, Bérénice Batut, Maria A Doyle, Tomas Klingstrom, Florian Eggenhofer, Peter F Stadler, Rolf Backofen, and Björn Grüning. The RNA workbench 2.0: next generation RNA data analysis. *Nucleic Acids Research*, 47(W1):W511-W515, 2019 [144].

5.1 Motivation

This chapter discusses an important aspect of my research - "sustainable bioinformatics". To cope with the recent advances in high-throughput sequencing methods, there is a rapid growth in bioinformatics tools and workflows every year [145, 146]. Accessibility and sustainability are often underestimated necessities of a good tool. After a bioinformatics method is published, these two important aspects of the research are often ignored. Bioinformatics tools can be provided as standalone programs or web servers. A standalone program is in the best case, portable and can run on a single computer, whereas a web server requires a network connection to access. Web servers are black boxes but are ideal for researchers without programming knowledge. For skilled bioinformaticians, command-line tools give a great deal of leverage in tool usage. A tool that comes as a command-line and as a web server is the best of both worlds. Another advantage of a web server is that the user does not need to care about the installation. But these benefits come at a cost of additional development. It includes setting up dedicated hardware, web services, wrapping the tool in a user interface, and most importantly maintaining it. When researchers publish their web server, they need to establish this infrastructure but eventually, they are unmaintained and become unusable. Comprehensive studies exist on the availability of published bioinformatics programs, web services and databases. Schultheiss *et al.* [147] focused mainly on web servers published in *Nucleic Acids Research* (NAR) web server issue from the year 2003 to 2009. On average 9% of the web services were not reachable and 20% were located at a different web address than

that referred to in the corresponding article. 50% of them not shipped with any example data and 33% of web servers did not provide "fair" testing possibility. A recent extended survey from Kern *et al.* [148] screened 2396 web tools published in 10 different journals from 2010 to 2020. There is a linear correlation between the accessibility and age of the web server. In 2020, ∼90% of the web servers published in 2019 and 2020 were accessible, whereas only ∼50% of the web servers published in 2010 were still working. Wren *et al.* [149] also showed a similar decay of accessibility for a wide selection of bioinformatics programs, web servers and databases published between 2000 and 2015.

Maintenance is the primary reason behind the inaccessibility of bioinformatics resources. All the aforementioned surveys in common have discovered a strong association between the maintainability of the web server and the number of citations it got. Sustaining the compatibility between the tool and its dependencies is also another challenging task. Over the years, incompatibilities build up when updating the dependent tools or operating system libraries. Tools that use APIs to access remote web services or that use remote databases must always stay up-to-date to prevent any possible failures. There are also non-technical reasons for this. For example, tool developers moving to a different job leaving the tool at the lab where it was developed or non-availability of consistent funding for the projects. Not having good documentation or user manuals can also leave the tools with no successors to maintain. Pooling resources and putting many services into a single web service or tool collection has many advantages. Generally, standalone tools or web servers that are part of a large tool collection are more likely to be well maintained and continuously updated than individual tools or web servers. Even upon the inclusion of a new tool or update of an existing tool, the overall integrity of the tool collection is verified. On the contrary, single tool web servers often cannot afford long-term dedicated maintenance infrastructure or personnel. Freiburg RNA Tools [150], Vienna RNA websuite [151], and EMBL-EBI bioinformatics tools framework [152] are good examples of tool/web server collections that are being maintained successfully since their publication.

A major portion of this thesis deals with the analysis of HTS data from complex workflows that are built by combining individual tools. The workflows that require minimum programming knowledge to create and run are more usable. Workflow management systems come in handy for this [153]. Galaxy [121] not only offers a great and easy to use workflow management system, its large development and support community also ensures the long-term sustainability of the tools.

5.2 A voyage to the Galaxy in the world of scientific data analysis

Galaxy is an open-source software framework, a tool and workflow management system that was initially developed to analyze genomics data. Its high usability and huge developer community attracted diverse fields of science such as proteomics [154], transcriptomics [155], computational chemistry [156], clinical research [157], drug discovery [158], image analysis [159], climate science [160], ecology [161], biodiversity analysis [162], natural language processing [163], and machine learning [164]. Accessible, reproducible, and transparent research is the motto of the Galaxy project. Using Galaxy, researchers can access computational tools and build complex workflows effortlessly without the need of programming skills. In a survey of 1,576 researchers from different fields of science, 90% believed that there is a need for strengthening reproducible research [165]. There are published studies that attempt to reveal the reproducibility crisis also in bioinformatics [166]. Galaxy was built in a way that each tool version, computing steps, and parameter details can be tracked back, which makes any analysis on Galaxy reproducible. Galaxy allows us to share analysis steps, workflows, and data to other researchers or publicly. The concept of sharing allows a transparent review of analysis and reproducibility. These key concepts of Galaxy are in harmony with the goal of sustainable research stated in section 5.1. Galaxy can be used from one of the publicly available servers (like https://usegalaxy.org or https://usegalaxy.eu) or on a personal computer. To use a public Galaxy server, all that is needed is a computer with a web browser.

5.2.1 Introduction to Galaxy framework

Any computational tool can be integrated into the Galaxy framework by writing an interface between the tool and Galaxy in a custom extensible markup language (XML) file. This interface file is known as a Galaxy tool or informally a Galaxy wrapper. A Galaxy tool contains the definition of all tool parameters and a command to invoke the tool with the parameters. These defined parameters are parsed by Galaxy and transformed into a user-friendly hypertext markup language (HTML) form. The whole command execution is translated into a single “Execute” button in the HTML form. However, Galaxy tool development requires some understanding of the underlying tool and basic command-line interface usage. These are commonly written by bioinformaticians. Galaxy also allows users to build workflows from the tools. Users can select the tools from the tool panel and connect them onto the workflow canvas. The association between any two tools is possible only if the output file type of the first tool is the same (or compatible) as the input file type of the second tool. Building workflows require no computational knowledge. Each step of the data analysis is stored in a separate “history”, meaning each entry in the history represents a computational step of

the analysis. Each history item also records the tool version, the complete set of parameters along with input data sets used to produce it. The workflows and histories can be shared with an individual user or among the whole Galaxy user community. The beneficiary can adapt the shared workflow and apply it to their data as needed.

5.3 Leveraging Galaxy and training into small non-coding RNA research

In my research career, I worked as a Galaxy tool developer, and a trainer for Galaxy-based HTS data analysis. Most of my contributions in Galaxy are in the RNA field. This led to my co-authorship in the publication related to the Galaxy RNA workbench [155, 144] and Galaxy training [161].

The RNA workbench is one of the finest resources for reproducible RNA research. An update of the workbench, *The RNA workbench 2.0* was released in 2019. Although I was involved in the development of both versions of the workbench, in this thesis I focus on version 2.0 into which the majority of my contributions went. It offers more than 100 tools and 25 carefully tailored analysis specific workflows for RNA-centric research. These workflows can be used for the analysis of RNA sequence alignments, RNA secondary structures, RNA-protein, RNA-RNA interactions, ribosome footprinting, RNA sequencing, RNA annotation, and so on. Figure 5.1 shows an overview of some major RNA research topics that the RNA workbench serves. RNA workbench was implemented based on the Galaxy framework. The tool dependencies in Galaxy are resolved via *Bioconda* [167], a channel for the *Conda* package manager, dedicated to bioinformatics software. Bioconda is a result of world-wide bioinformatics community efforts. Writing a Conda recipe for a tool requires knowledge of the installation and basic usage of that tool. Every tool in Bioconda undergoes continuous integration and is rigorously tested which guarantees the functioning of the tools. On top of it, Docker-based *BioContainers* [168] facilitate continuous deployment. In addition to the RNA analysis tools, the RNA workbench offers a wide variety of text manipulation, file conversion tools, and a great visualization framework to produce publication-ready plots.

The first version of the workbench was provided as a Galaxy framework in a containerized *Docker* image [169]. Docker-based workbench comes with the advantages of easy deployment and minimal maintenance. Flexible deployment on local hardware infrastructure benefits medical facilities in dealing with sensitive patient data. In addition to the Docker container, the RNA workbench 2.0 was launched in a web server at https://rna.usegalaxy.eu. By being part of the European Galaxy server, workbench users were able to leverage the huge storage space and high-performance computing environments. The number of usable tools and workflows were doubled in the latest workbench, and over the years, the close alliance

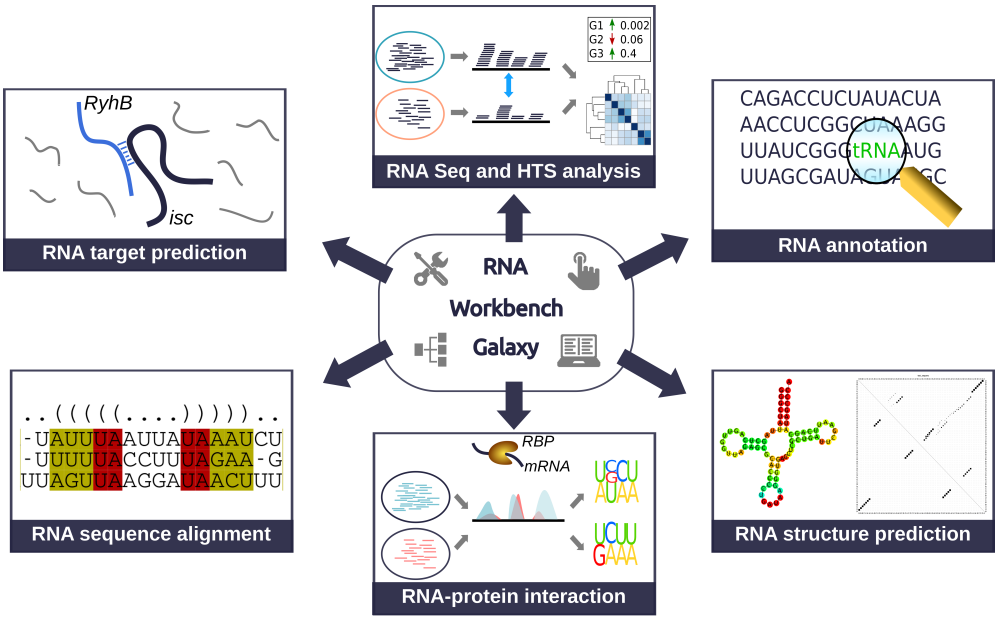


Figure 5.1: The RNA workbench serves classical RNA bioinformatics such as RNA structure prediction, sequence alignment, target prediction and provides analysis workflows for high-throughput RNA-Seq analysis, CLIP-Seq analysis, RNA-RNA interactome data analysis, etc. This figure is taken from publication *P4*.

between the RNA workbench community and the Galaxy training community led to the development of more and more training materials for various types of RNA based analysis workflows. Along with the workflows, training has now become an integral part of the RNA workbench.

The vast majority of Galaxy users are biologists that have little to no experience in bioinformatic analysis. Galaxy training sessions held by the field experts deliver a hands-on experience in data analysis. Eventually, lack of trainers and training resources could not serve the increased demand for training. To supplement the training sessions, the global Galaxy training community brought up a unified training material database. Now there are more than 100 different tutorials from 16 different topics. Additionally, there are more than 50 tutorials for Galaxy administrators, developers, and instructors.

Each training lesson can be viewed as a written transcript of a training session held by the field experts. Each training lesson is a collection of 6 components (see Figure 5.2), namely, slides, input data, hands-on tutorial, workflows, Galaxy history, and Galaxy interactive tours. Slides contain motivation and useful background information for each topic. The input data used in the lesson is usually stored on *Zenodo* (`https://zenodo.org/`). Each data set corresponding to the training lesson gets a citable unique digital object identifier. The hands-on provides step-by-step instructions of the complete analysis along with some perks in the form of useful tips and thought-provoking questions.

These tips and questions are derived from the experiences of real data analysts and from frequently asked questions from the users. Each training is provided with an example workflow related to the training topic. The provision of a Galaxy history generated by running the workflow on the input data allows the trainees to verify their analysis steps. Galaxy interactive tours are the analysis walkthroughs on the Galaxy interface that is aimed at novice users. With 23 tutorials, transcriptomics is the most established topic in the current Galaxy training material databases (https://training.galaxyproject.org/training-material/). Unsurprisingly, the RNA workbench community developed a major portion of it.

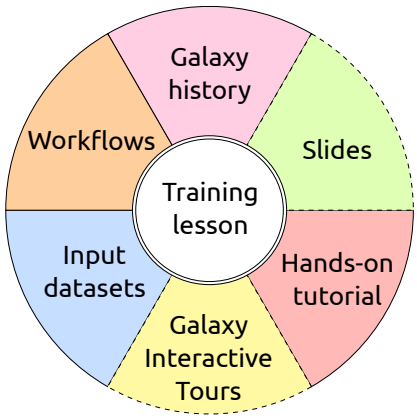


Figure 5.2: Components of each training lesson. Solid lined components are mandatory to make up an effective training lesson.

As I worked mostly in the transcriptomics field, all the tools that I implemented in my research career are part of the RNA workbench. The workflows related to small non-coding RNA analysis using `BlockClust` (publication *P1*) and RNA-RNA interactome analysis using `ChiRA` (publication *P3*) tool suites are available on the workbench. Detailed training materials for each of the analysis workflows also have been developed. As a part of work related publication *P2*, the first stable version of the Galaxy wrapper for `DESeq2` has been developed. It is now evolved to be the most popular Galaxy tool for differential gene expression analysis, widely used in Galaxy training network and RNA workbench. Being part of the workbench team also involves checking the integrity of several other tools and workflows to support my goal of accessible and reproducible research.

Part IV

Conclusion and outlook

Conclusion

This thesis presents novel computational methods for the prediction and annotation of small non-coding RNAs and their interactions. `BlockClust` is an efficient approach to predict the small ncRNAs from small RNA-Seq data. From the results on the benchmark data, it is evident that `BlockClust`'s graph similarity notion yields better clustering with a significant speed-up compared to an existing clustering approach called `deepBlockAlign`. It also showed a competitive predictive performance for miRNAs, tRNAs and snoRNAs compared to `DARIO`'s state-of-the-art classification method. With a consistent performance on different organisms and a variety of tissues and cell lines, `BlockClust` proved to be a robust and bias-free approach that can reliably annotate read profiles. Because the clustering is purely based on the features of read profiles and it does not make any assumptions regarding the ncRNA classes, `BlockClust` is suitable for clustering ncRNAs with similar processing patterns and potentially to predict novel ncRNA classes. The implementation of `BlockClust` allows easy incorporation of new attributes that makes it readily usable to solve a different analysis problem than ncRNA clustering. For example, previously, `BlockClust` was adapted with the inclusion of attributes that are relevant for a specific Ribo-Seq experiment to characterize ribosomal footprints in different experimental conditions in yeast.

In addition to the prediction of ncRNAs, this thesis also contributed to the identification and analysis of their interactions that helps in understanding their function. It has been shown using RNA-Seq data analysis that the FOXG1 interactions with the miR200 family and the regulation of PRKAR2B by miR200 helped in unveiling an important pathway in atypical Rett syndrome. `ChiRA` tool suite provides comprehensive bioinformatics solutions for detection and analysis of miRNA interactions from genome-wide RNA-RNA interaction experiments. Benchmarking of the `ChiRA` workflow demonstrated that considering the sub-optimal alignments, building CRLs from consistently multi-mapped reads in combination with quantification of the reference loci enabled accurate annotation of sequenced chimeric fragments. Genes that belong to CRLs also implied that `ChiRA` can cluster the reference loci that belong to gene families without requiring any gene annotation.

All the tools and workflows that are developed for this thesis are part of the RNA workbench. It also includes tutorials for small ncRNA prediction and RNA interactome data analysis. A continually growing developer base of the RNA workbench ensures the long-term sustainability of these works.

Outlook

Studies show that miRNAs, tRNAs and snoRNAs are expressed differently in different cell types and tissues [170, 171, 172]. Consequently, their biogenesis is affected and produced

functional mature miRNAs, tRFs and sdRNAs are also cell-type specific. For example, tissue-specific miRNA arm switching alters the gene regulatory landscape [31]. To my knowledge, there are only two computational tools available for the detection of differentially processed small ncRNAs and the agreement between their predictions is moderate [173, 174]. There is still a need for more sophisticated bioinformatics methods for comparing processing patterns of small ncRNAs among different tissues and cell types.

Rapidly developing single-cell RNA-Seq (scRNA-Seq) enables a deeper understanding of cell heterogeneity, behavior and development [175]. Currently, there are only very few scRNA-Seq protocols available that deliberately sequence small ncRNAs [176, 177]. Single-cell sequencing of small ncRNA transcriptome indicated that the miRNA expression levels yielded better clustering of cell types than mRNA expressions [178]. However, in the same experiment, tRNAs and snoRNAs did not show any pronounced cell-type-specific expression levels. Performing an additional clustering based on the processing patterns of ncRNAs may reveal some interesting insight into their biogenesis and function in different cell types.

Not only are miRNAs enriched in specific cell types, but also their targets. For instance, cell-type-specific target selection of miRNAs during various developmental stages of the brain suggests their important role in cell proliferation and differentiation [179]. A recent single-cell experiment analyzed the genome-wide miRNA regulatory mechanism by co-sequencing miRNAs and mRNAs [180]. However, the protocol is not designed to capture the interactions among the miRNAs and mRNAs. Therefore, miRNA-target relations had to be derived computationally, using TargetScan. High-throughput experimental protocols to sequence and analyze direct miRNA-target interactions at single-cell resolution are yet to be designed. Such protocols will help to understand cell-type specific miRNA regulatory networks and potentially contribute to effective miRNA therapeutics and disease control.

Part V

Publications

Publications

This part contains all the publications on which this thesis is based. The publications are in the order of contributions described in Part III. Each publication is prepended with a statement of contribution of all the associated authors.

[P1] BlockClust: efficient clustering and classification of non-coding RNAs from short read RNA-seq profiles

Pavankumar Videm, Dominic Rose, Fabrizio Costa, and Rolf Backofen. *ISMB 2014 proceedings'* special issue in *Bioinformatics*, 2014. DOI: 10.1093/bioinformatics/btu270

Contributions of individual authors:
I am the major contributor to this work. I developed the idea for the paper, implemented the non-coding RNA prediction tool BlockClust, integrated it into the Galaxy framework, analyzed the data, benchmarked the tools and interpreted the results. I wrote and revised the manuscript. Fabrizio costa helped in graph kernel usage and manuscript revision. Dominic Rose provided consultation and revised the manuscript. Rolf Backofen provided general consultation and involved in writing the introduction section.

Pavankumar Videm

The following authors confirm the above stated contributions.

- Pavankumar Videm
- Dominic Rose
- Fabrizio Costa
- Rolf Backofen

BIOINFORMATICS

Vol. 30 ISMB 2014, pages i274–i282
doi:10.1093/bioinformatics/btu270

BlockClust: efficient clustering and classification of non-coding RNAs from short read RNA-seq profiles

Pavankumar Videm[1], Dominic Rose[1,2], Fabrizio Costa[1,*] and Rolf Backofen[1,3,4,5,*]

[1]Bioinformatics Group, Department of Computer Science, University of Freiburg, [2]Munich Leukemia Laboratory (MLL), Munich, [3]Centre for Biological Signalling Studies (BIOSS), [4]Centre for Biological Systems Analysis (ZBSA), University of Freiburg, Germany and [5]Centre for Non-coding RNA in Technology and Health, Bagsvaerd, Denmark

ABSTRACT

Summary: Non-coding RNAs (ncRNAs) play a vital role in many cellular processes such as RNA splicing, translation, gene regulation. However the vast majority of ncRNAs still have no functional annotation. One prominent approach for putative function assignment is clustering of transcripts according to sequence and secondary structure. However sequence information is changed by post-transcriptional modifications, and secondary structure is only a proxy for the true 3D conformation of the RNA polymer. A different type of information that does not suffer from these issues and that can be used for the detection of RNA classes, is the pattern of processing and its traces in small RNA-seq reads data. Here we introduce `BlockClust`, an efficient approach to detect transcripts with similar processing patterns. We propose a novel way to encode expression profiles in compact discrete structures, which can then be processed using fast graph-kernel techniques. We perform both unsupervised clustering and develop family specific discriminative models; finally we show how the proposed approach is scalable, accurate and robust across different organisms, tissues and cell lines.

Availability: The whole `BlockClust` galaxy workflow including all tool dependencies is available at http://toolshed.g2.bx.psu.edu/view/rnateam/blockclust_workflow.

Contact: backofen@informatik.uni-freiburg.de; costa@informatik.uni-freiburg.de

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

1 INTRODUCTION

Genome-wide sequencing revealed that DNA is pervasively transcribed, with the majority of the DNA encoding for non-coding RNAs (ncRNAs) (Jacquier, 2009). ncRNAs are important parts of cellular regulation that were long ignored but have received an increasing level of attention over the past decade. There have been reports of up to 450 000 predicted ncRNAs in the human genome alone (Rederstorff *et al*., 2010), the vast majority of them having no functional annotation. While the exact numbers, and even the magnitude, of regulators and interactions are of course a matter of discussion, they reflect the current challenge for the analysis of whole-transcriptome data.

Comparatively assigning a putative function to ncRNAs requires the detection of RNA families or classes with a common function. RNA families contain sequences that are related via evolution, whereas the members of RNA classes are defined only by a common function without evolutionary relationship, with miRNAs and snoRNAs being well-known examples. RFAM (Burge *et al*., 2013) is the largest known collection of known RNA families. However, only a minor part of the transcriptome is covered by those examples. For that reason, Will *et al*. (2007) and Torarinsson *et al*. (2007) introduced clustering of transcripts according to sequence and structure as a mean to assign functions. This is now used as a standard tool for the detection and analysis of ncRNA in genomic and metagenomic data [see e.g. Parker *et al*. (2011), Saito *et al*. (2011), Weinberg *et al*. (2009) or Shi *et al*. (2009)].

There are, however, several caveats if one relies only on the genomic sequence and its predicted secondary structure. First, the genomic sequence is often changed by post-transcriptional modifications. The database of RNA modification pathways [MODOMICS, see Machnicka *et al*. (2013)] lists 144 types of modifications, from methylation of RNA bases to editing events like C-to-U or A-to-I editing [see e.g. Su and Randau (2011) or Nishikura (2010)]. Second, the reliability of the classification depends on the quality of secondary-structure prediction, which is often low [see e.g. Mathews *et al*. (2004)]. The reason is not only that the energy model for secondary structure is incomplete, but RNA modifications and the influence of RNA-binding proteins also add layers of complexity. In the case of transcriptome data, an additional problem is that often the full transcript is not seen in the deep sequencing. This implies that one has to perform *local* secondary-structure prediction, which is an even harder task (Lange *et al*., 2012). Third, relying on structure is optimal for structured ncRNA, but would miss many long ncRNAs that often do not have a conserved structure [for a review, see e.g. Rinn and Chang (2012)].

There is, however, a similarity other than the genomic sequence and its predicted secondary structure that can be used for the detection of RNA classes, namely the pattern of processing and its traces in small RNA-seq reads data (Findeiss *et al*., 2011). The reason is simply that these processing patterns depend on the functional molecule and its 3D-structure, and thus should carry information not only about the structure of the polymer but also about all modifications and processing of the RNA molecule. This is well understood for prominent examples like miRNA, where most pre-miRNA have a hairpin structure with a 2-nt 3′ overhang that are processed into a double-stranded RNA consisting of the miRNA and its complement miRNA* [see e.g. Gan *et al*. (2008), for alternative processing modes see e.g. Ando *et al*. (2011)]. Computational approaches for finding new miRNAs in deep sequencing data such as `miRDeep` rely on the detection of traces of this process (Friedlander *et al*., 2008).

It has now become clear that this is not limited to miRNA. Instead, class specific slicing of widely expressed ncRNAs (but

*To whom correspondence should be addressed.

no mRNAs) into smaller RNAs is a widespread regulatory mechanism (Li *et al*., 2012). Examples are tRNA, where there are several species of tRNA-derived fragments such as tRNA halves, 5′ tRF, 3′ U tRF or 3′ CCA tRF are known (Gebetsberger and Polacek, 2013). Similarly, snoRNA-derived (sdRNAs) fragments are specific for the snoRNA class and the size and position distribution of the sdRNAs are conserved across species (Taft *et al*., 2009).

In this article, we introduce `BlockClust`, an efficient approach to detect transcripts with similar processing patterns. We propose a novel way to encode expression profiles in compact discrete structures, which can then be processed using fast graph-kernel techniques.

Note that in this work we do not deal with long RNA sequences such as messenger RNAs which require to deal with exon boundaries or with extreme variability in length and expression levels, rather we consider the transcripts that are retrieved from small RNA-seq protocols and we therefore optimize `BlockClust` to process transcripts characteristic of small ncRNAs of length 50–200 nt.

We perform both unsupervised clustering and develop ncRNA family specific discriminative models; finally we show how the proposed approach is scalable, accurate and robust across different organisms or experimental protocols.

2 MATERIALS AND METHODS

The core idea of the `BlockClust` method is to characterize transcribed loci using the expression profiles obtained from deep sequencing experimental protocols. To do so, we extract characteristic attributes from the expression profiles, such as the entropy of the read length or the normalized read expression. We then encode the sequence of several of these attributes in compact discrete structures, which we then process using fast graph-kernel techniques.

More specifically, in order to achieve high computational efficiency, we do not use alignment-based techniques as done e.g. in `deepBlockAlign` (Langenberger *et al*., 2012), and we do not resort to a set of handcrafted measurements or features to describe the entire profile as done e.g. in `DARIO` (Fasold *et al*., 2011). Instead in `BlockClust` we partition the reads of an expression profile in a sequence of blocks. We then discretize the statistics of the reads distribution in each block and we encode the result in a discrete data structure. Such representations can be processed by high-performance machine-learning techniques such as kernelized Support Vector Machines (Joachims, 1999) to build classification models or by Locality Sensitive Hashing techniques (Heyne *et al*., 2012) to obtain fast clustering approaches.

In summary, the two key components in `BlockClust` are: (i) the expression profile encoding with discretized attributes, and (ii) the combinatorial feature generation from the sequence of attributes.

2.1 Expression profiles encoding

With the term *expression profile* we denote the set of assembled read sequences relative to a given transcript. In order to extract these profiles, read sequences from the deep sequencing experiments are aligned (or mapped) against their corresponding reference genome in order to get the chromosomal coordinates (note that `BlockClust` can in principle work on reads that are mapped to assembled transcripts, when there is no reference genome). The information about mapped reads is generally stored using the Sequence Alignment/Map (SAM) format or a compressed binary version of the SAM format (BAM). BAM files can be converted to a six-column Browser Extensible Data (BED) file of *tags* (a *tag* is a unique read sequence in a deep-sequencing library). The BED format provides information on the *normalized expression* of each tag, i.e. the ratio of the read count per tag to the number of mappings on the reference genome. Considering tags instead of reads allows a high lossless compression of the original data. In `BlockClust` we further represent this information by (i) grouping tags into 'blocks' and sequences of blocks into 'block groups', (ii) extracting several statistics from the read signal within each block and globally over the whole block group and (iii) discretizing these statistics. In this way we can represent hundreds of thousands of reads over regions spanning hundreds of nucleotides with few bytes. More in details, the expression-profile-encoding phase is composed of the following steps: (i) conversion of BAM file to BED file of normalized tag expressions, (ii) block and block groups extraction, (iii) statistics extraction for each block and block group, (iv) discretization of the statistics, (v) graph encoding of the block group and of the associated discretized statistics. The novel contribution of this work lies in the details of phases 3 and 5.

2.1.1 Blocks and block groups In order to enhance computational performance, we compress expression profiles by grouping reads into blocks. Because of biological noise and sequencing errors, the read positions do not respect any exact boundary notion, and one cannot therefore assume that blocks should be non-overlapping. For this reason we use the `blockbuster` tool (Langenberger *et al*., 2009) to identify blocks. The idea is to perform peak detection on the signal obtained by counting the number of reads per nucleotide. This signal, spanning adjacent loci, is then modeled with a mixture of Gaussians. An iterative greedy procedure is then used to collect reads that belong to the same *block*, starting from the largest Gaussian component, and removing them in successive iterations. The tool further assembles a sequence of adjacent blocks into a *block group* if the blocks are either overlapping or are at a distance smaller than a user-defined threshold. Finally, in `BlockClust` we assume that a gene can span at most a single block group.

2.1.2 Blocks and block groups attributes To identify patterns in expression profiles we partition the reads into blocks and block groups and then describe each block and the entire group of blocks with a set of statistics and measures. Note that it is not possible to characterize different ncRNA families using only simple statistics on the overall distribution of reads. To increase the discriminative power we therefore consider the exact sequence of blocks, making use of attributes relative to each individual block and relative to the relations between adjacent blocks. More precisely we define three types of descriptive attributes: (i) block group attributes, (ii) individual block attributes and (iii) block edge attributes, i.e. measures about the relation between two adjacent blocks in a block group. The block group attributes are: *entropy of read starts*, *entropy of read ends*, *entropy of read lengths*, *median of normalized read expressions* and *normalized read expression levels in first quantile*. The block attributes are: *number of multi-mapped reads*, *entropy of read lengths*, *entropy of read expressions*, *minimum read length* and *block length*. The block edge attributes are: *contiguity* and *difference in median read expressions*. The entropy of read starts is defined as $-\sum_i q_i \log_2 q_i$, where q_i is the fraction of reads in a given block group starting at position i. The other entropies are defined correspondingly. The overall *expression* is defined as the sum over all tag expressions per block. The *block contiguity* is defined as the overlap fraction or the fractional distance between two consecutive blocks. For more details see Supplementary Material Section S.4.

2.1.3 Attribute discretization To identify patterns in large collections of sequences of blocks we propose to discretize the attributes, treating the resulting intervals as nominal values. This achieves the combined result of reducing data storage requirements and it allows us to use powerful machine-learning techniques that work on discrete data structures. Discretization methods can be divided into those that choose the intervals taking the class information into account and those that are class-blind.

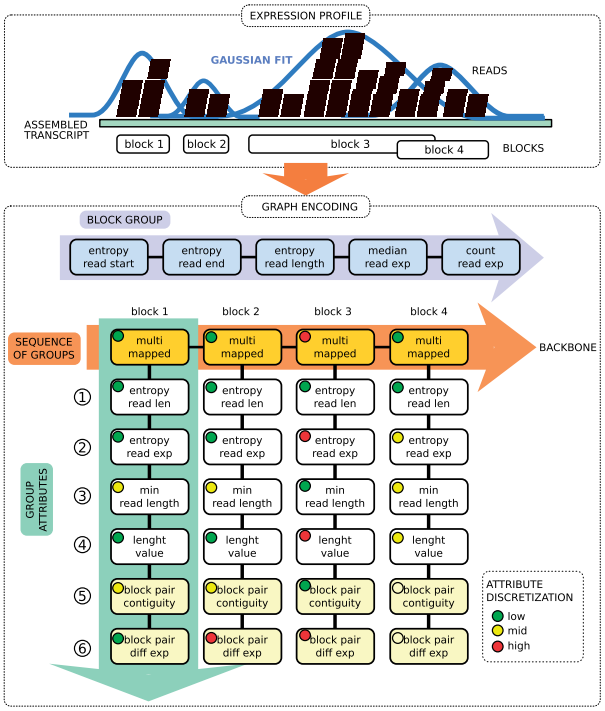


Fig. 1. Read profile encoding. (Top) Read profile, and successive partition of reads in blocks (`blockbuster`). (Bottom) The block partitioned reads are encoded as a graph with two disconnected components: (i) BLOCK GROUP: which contains statistics and attributes of the global distribution of reads; (ii) SEQUENCE OF BLOCKS: which encodes a list of attributes for each individual block. The *backbone* is the sequence of the most discriminate type of block attribute. The discretized value of each attribute is depicted by a color-coded circle in the corresponding box

As we seek an approach that is ultimately capable of novel discoveries, we opt for the latter. Between the two main class-blind approaches, the equal-width and the equal-frequency discretization, we observe that the first method can yield empty intervals and lose a large amount of information. We chose therefore the equal-frequency algorithm, which sorts all values and then divides the range into a user-defined number of intervals so that every interval contains the same number of values (note that special care must be taken to treat identical values, which can potentially spread over several intervals. We adjust the procedure so that duplicate values belong to a single interval).

2.1.4 Graph encoding Since the number of blocks is variable we cannot use a simple vector or matrix encoding of the attributes. Instead we encode the sequence of blocks with attributes as a graph with discrete labels. We adopt an encoding similar in spirit to the one used in Kundu *et al.* (2013). In `BlockClust` we encode a single non-protein-coding gene with a graph made of two disconnected components: in the first one we represent the overall block group information, while the second one is used to represent the sequence of individual blocks and their relationships. The first component is modeled as a position specific sequence of discretized attributes, that is, each block group attribute type appears consistently at the same position in the modeling sequence (see Figure 1 at the top of the Graph Encoding box). The second component is modeled as a sequence of vertices, each representing a single block, called the *backbone* sequence. The discretized block attributes are represented as a position-specific sequence and this sequence is connected to the correspondent block vertex. Two blocks that appear subsequently in the assembled transcript are encoded as adjacent vertices. The block edge attributes between two adjacent blocks with starting coordinates i and j respectively, with $i<j$, are appended at the end of the attributes for the block with starting coordinates i. Note that in practice we collapse the vertex representing the block together with the vertex representing the first attribute. The final structure for the second component is therefore a sort of 'comb'-shaped graph as shown in Figure 1. Note that the order of the attributes affects the discriminative capacity of the encoding and it therefore needs to be optimized (see Section S.4 in Supplementary Material).

The reason for this type of more complex modeling, as compared to a simple sequential encoding, becomes clearer in the next section, where we introduce how the actual features are extracted in a combinatorial way from the graph encoding.

2.2 Combinatorial feature generation

In `BlockClust` we do not employ alignment-based techniques to compare block groups, as we would incur in high computational costs. Instead given the graph encoding of a block group we extract an explicit feature representation that can be processed more efficiently. The type of features considered are those developed for a recently proposed graph kernel called Neighborhood Subgraph Pairwise Distance Kernel (`NSPDK`) (Costa and Grave, 2010) and used for the efficient clustering of ncRNA molecular graphs in Heyne *et al.* (2012).

`NSPDK` is a fast graph kernel based on exact matching between pairs of small subgraphs. One can view the similarity notion expressed by `NSPDK` as a generalization of the k-mer substring kernel for strings (with gaps) to the case of graphs. The idea is to decompose a graph into a set of smaller fragments and express the similarity between two graphs as the fraction of common fragments. In `NSPDK` the fragments are pairs of neighborhood subgraphs for a small radius (parametrized by the maximum allowed radius R) at increasing distances (parametrized by the maximum allowed distance D). Intuitively the radius parameter controls the complexity of the features, while the distance parameter controls the range of locality for the non-linear interactions. A neighborhood graph is a subgraph specified by a root vertex v and a radius R, consisting of all vertices that are at a distance (the distance between two vertices v and u on a graph is defined as the number of edges in the shortest path between v and u) not greater than D from v. All pairs of neighborhood subgraphs can be efficiently enumerated in near linear time and hashing techniques can be used to extract quasi-canonical identifiers from these pairs (see Supplementary Material Section S.1 for a formal introduction and additional details). As shown in Heyne *et al.* (2012) we can use these identifiers to build feature indices and represent graphs as vectors in a very high-dimensional vector space. Differently from what is done in Costa and Grave (2010) and Heyne *et al.* (2012) here we make use of the notion of *viewpoint* [first introduced in Frasconi *et al.* (2012)]. A viewpoint is an additional information that is placed on specific vertices in the graph encoding. The intended effect is to constrain the feature-generation mechanism in such a way that at least one of the two subgraph root is a viewpoint. In this way we can choose which specific vertices are more relevant in a given domain. In our case we place viewpoints on the backbone, i.e. the chain of vertices representing the blocks. In this way we generate features that at the same time (i) take into account an incremental amount of attributes, but (ii) that work on a much smaller subset of the exponentially large set of possible combinations. Since the sequential order in which we encode the attributes determines the combinations generated, we need to determine the optimal order (see Section 3.1.3 for further details on the parameters optimization step). The features obtained following the `NSPDK` approach contain pairs, triplets and higher order combinations of the original attributes. Having these complex features allow linear models to express complex classification decisions that are non-linear with respect to the original sequential

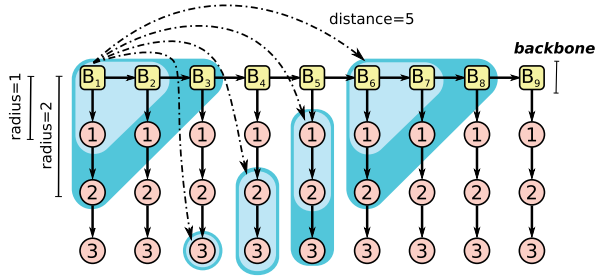


Fig. 2. Combinatorial features. Given a directed graph, the NSPDK approach constructs a large number of features taking only specific subgraphs into account. The procedure is parametrized by the maximum *radius* R and the maximum *distance* D. Each vertex is considered in turn as a *root*. A neighborhood graph of radius $r = 1, \ldots, R$ is extracted around each root. All possible pairs of neighborhood graphs of the same size r are considered, provided that their respective roots are exactly at distance $d = 1 \ldots D$. Viewpoints are used to constrain at least one of the roots to be on the backbone. The graph shows a specific case of combinatorial feature construction with $r = 1, 2$ and $d = 5$ with the viewpoint in $v = B_1$

information. Figure 2 depicts how the features are generated with a given radius and distance. The technique that we present here allows therefore to combine the benefits of large-scale efficiency provided by linear methods and locality sensitive hashing with accurate non-linear modeling.

2.3 ncRNA expression profile clustering

The similarity notion of NSPDK can be used directly by clustering algorithms that make use of pairwise similarity or distance information. More specifically the similarity between two expression profiles is equivalently defined as the dot product of the corresponding high-dimensional vector representations. Note that in large-scale settings, when a quadratic complexity in space or time is unfeasible, we can avoid to materialize the pairwise similarity matrix and resort instead to more efficient locality sensitive hashing techniques (see Supplementary Material Section S.2 for details) as introduced in Heyne *et al.* (2012). This technique allows us to extract the approximate nearest neighbors in linear time complexity.

As a clustering algorithm BlockClust uses the Markov Cluster Process (MCL) algorithm (Enright *et al.*, 2002). Given a weighted nearest neighbor graph G between the instances to be clustered, the MCL algorithm applies a parametrized algebraic process to the matrix of random walks on G. The underlying idea is to characterize clusters as subgraphs such that a random walk on the graph will infrequently go from one subgraph to another. The MCL was chosen as it produces balanced non-hierarchical clusters and it does neither need seeding information nor a user-defined number of clusters. Moreover it can be employed in large-scale settings as it can work with sparse graph/matrix implementations. In our application setting, the *inflation* parameter, which affects the cluster granularity, was selected to retain relatively small clusters.

2.4 ncRNA expression profile classification

In addition to unsupervised clustering, BlockClust provides a supervised classification mode. Given a set of expression profiles for a known ncRNA family or class and a set of negative examples, i.e. expression profiles of ncRNAs with a different or unknown function, BlockClust can efficiently build a discriminative linear binary classifier. As in the unsupervised clustering mode, we first extract explicit high-dimensional vector representations from the expression profile encodings. Subsequently BlockClust uses fast and scalable linear techniques such as Stochastic Gradient Descent Support Vector Machines (Bottou, 2010) to induce a discriminative model. Note that even if we use linear models to allow scaling to genome wide data settings, the resulting classifier is in fact non-linear in the original attribute space.

The resulting models are precise and surprisingly robust: a model for the identification of tRNA genes can be trained on human data with reads extracted under a specific experimental protocol (say Illumina GAII) and it can be used to reliably annotate expression profiles across diverse organisms (e.g. fly or plants), from data produced by different experimental protocols (see Section 3.2).

3 RESULTS AND DISCUSSION

In order to evaluate the BlockClust approach we formulate and analyze the following questions.

- Q1: Is the BlockClust encoding of expression profiles informative enough to be used in clustering procedures to detect specific ncRNA classes?
- Q2: Is the BlockClust encoding of expression profiles robust across different sequencing platforms, organisms, tissues, cell lines?
- Q3: Can BlockClust be used for the annotation of known ncRNAs classes?
- Q4: How does BlockClust compare to other tools for clustering or classification of expression profiles?

3.1 Q1: clustering ncRNAs with encoded expression profiles

3.1.1 Performance measures Given the graph encoding of two expression profiles, BlockClust can compute a similarity score between the corresponding high-dimensional feature representations. Formally, given two expression profiles a and b, if x_a and x_b are the corresponding vector representations, then their (cosine) similarity is defined as $S(a,b) = \frac{\langle x_a, x_b \rangle}{\sqrt{\langle x_a, x_a \rangle \langle x_b, x_b \rangle}}$. To assess the quality of this similarity notion, we measure the grouping tendency for profiles of functionally identical RNAs. As a measure we chose the Area Under the Curve for the Receiver Operating Characteristic (AUC ROC), which is defined as the integral of the fraction of true positives out of the total actual positives with respect to the fraction of false positives out of the total actual negatives at various threshold settings. More precisely, given a profile a we sort all other profiles by decreasing similarities with respect to a. Ideally, we expect the neighboring instances to share the same ncRNA class, i.e. we expect the same class to appear preferentially at the beginning of the sorted list. We consider the class of profile a as the positive class and all other classes as the negative class. Given this assignment we can compute the AUC ROC as if a was a classifier. The overall performance is then computed as the average AUC ROC over all instances. As a general rule of thumb, AUC ROC values >0.9 are excellent, ~0.8–0.9 are indicative of good performance, 0.7–0.8 indicate a somewhat sufficient quality, while 0.5 is the baseline for pure random performance.

Since the similarity score can be used for clustering purposes we also need a performance measure for the final cluster quality. We do not resort to measures such as the *adjusted Rand index* or the *F1 score* since we expect the same ncRNA class to be

Table 1. Parameter optimization

Component	Parameter	Interval	Step	Optimum
`blockbuster`	Cluster distance	20–100	10	40
`blockbuster`	Scale of standard deviation	0.2–0.8	0.1	0.5
Encoding	Discretization bins	3, 5, 7	2	3
`NSPDK`	Radius R	1, 3, 5, 7	2	5
`MCL`	Inflation	1–30	0.3	20
`MCL`	Pre-inflation	1–30	0.3	20

Overview of the parameters value ranges, search step size and the selected optimal values. Note that D is set as a function of R: $D = 2 \times R + 1$.

partitioned in several highly similar sub-groups, a situation that would be penalized by such measures. We therefore score the cluster *purity* via the *average precision per cluster*, defined as the fraction of instances belonging to the majority class present in each cluster.

3.1.2 Datasets We used NGS data generated by Illumina sequencing of human embryoid body (EB) and embryonic stem cells (hESC) (Morin *et al*., 2008), H1 cell line (H1) and IMR90 cell line (Bernstein *et al*., 2010), referred to as `Development Data` in the following. In order to evaluate robustness of `BlockClust` we used a comprehensive collection of datasets (see Supplementary Table S1), which we refer to as `Benchmark Data`, that comprises 32 samples, out of which 13 from human, seven from mouse, five from fly (*Drosophila melanogaster*), two from chimp, two from worm (*Caenorhabditis elegans*) and three samples from plant (*Arabidopsis Thaliana*). The sequencing machines were: Illumina GAII, Illumina GAIIx, Illumina GA, Illumina HiSeq 2K. Cell types, tissues and organisms include human and chimp brain tissues, human skin, embryos of worm, head and body of fly, testis and uterus of mouse and leaves and seeds of plant. Cell lines range from the H1 cell line to breast cancer cell line MCF-7 from human, from S2 to KC cells for the fly.

3.1.3 Parameters optimization The `BlockClust` system configuration comprises several parametric choices for each phase: (i) block identification, (ii) graph encoding and (iii) clustering or classification (see Table 1). For the `blockbuster` module used in phase (i) we need to specify the desired grain resolution in terms of *cluster distance* and *standard deviation*; in (ii) for the attribute construction phase we need to specify the discretization resolution and for the feature construction phase the complexity of the features via the maximal radius and distance needs to be set; finally in the clustering phase (iii) we need to specify the desired grain resolution for the clusters via the *inflation* and *pre-inflation* parameters and the regularization trade-off in the classification phase.

We optimized all parameters using a 35/35/30%-dataset split of `Development Data` into train/validation/test set, respectively. In the remainder of the article all performance measures are reported on the test set, while the quality of the parametric choices are evaluated on the validation set. Note that the parameters are not optimized on each ncRNA class separately.

Table 2. Clustering performance of `BlockClust` averaged over 10 random test splits of `Development Data`

ncRNA class	Number of transcripts	AUC	Number of clusters	Precision
miRNA	168	0.896	10	0.855
tRNA	173	0.741	17	0.837
C/D-box snoRNA	78	0.731	7	0.683
H/ACA-box snoRNA	4	0.838	0	0
rRNA	20	0.872	2	0.956
snRNA	7	0.637	0	0
Y_RNA	8	0.685	0	0
Weighted average	458	0.805	36	0.813

AUC ROC was measured from the expression profile similarities and precision from the clusters generated by the `MCL` algorithm. Note that due to the very low number of transcripts for the classes H/ACA-box snoRNAs, snRNAs and Y_RNAs we could not retrieve any significant clusters.

One initial difficulty to overcame is represented by the circular notion that (i) we would like to partition the dataset without splitting the profiles that belong to a unique ncRNA, but at the same time and (ii) we need to have a parametric method to identify the extent of the underlying ncRNAs and the parameters have to be determined on a valid dataset partition. We therefore break the circularity by employing the conservative and non-informed notion of *read stretches*, defined as a series of sorted reads separated by a maximum distance d. With the exception of a few ribosomal RNAs, most of the classic short ncRNAs are not longer than 500 nt. Finally, we set d to 500 and partition the set of the resulting read stretches in train, validation and test sets.

In order to evaluate the quality of the pre-processing step, i.e. the extraction of blocks and block groups, we measured the fraction of retrieved annotations. An annotation is considered retrieved if there is a reciprocal overlap of at least 70% between the annotation and the block group. All the transcripts failing this criteria and which are also not in the length range of 50–200 nt were discarded. For further details on the parameter optimization phase refer to Supplementary Section S.3.

3.1.4 Performance results In Table 2 we report for each ncRNA class the average AUC and the overall weighted mean on all classes averaged over 10 random test splits of `Development Data`. In our sample we observed seven ncRNA classes, namely: miRNAs, tRNAs and C/D-box snoRNAs (which contribute to the majority of the profiles) and rRNAs, snRNAs, Y_RNAs and HACA-box snoRNAs. Overall, we observed good average performance results (AUC ROC $\approx$ 0.8). Best results were with miRNA (AUC ROC $\approx$ 0.9), H/ACA-box snoRNA (AUC ROC $\approx$ 0.9) and rRNA (AUC ROC $\approx$ 0.85), good results were obtained with tRNA (AUC ROC $\approx$ 0.75) and C/D-box snoRNAs, while snRNA and YRNA performed poorly (AUC ROC $\approx$ 0.6). These last ones are also the least represented having only $\sim$10 instances each.

In Table 2 we report also details on the precision clustering performance for the four ncRNA classes with the largest number of instances: tRNA, miRNA, rRNA and C/D-box snoRNA. We used the `MCL`-clustering algorithm with inflation parameter set to 20 to capture also small clusters. The clusters obtained for

tRNA, miRNA and rRNA are quite consistent, containing <20% extraneous material on average, while for C/D-box snoRNAs the precision is ≈68%. For the remaining three classes with low number of instances, `MCL` could not identify any cluster.

3.2 Q2: robustness and range of applicability

A desirable property for a parametric computational approach is to require little to no parameter re-configuration when the data changes in ways that are marginal with respect to task at hand. In our case we would like the parametrization of `BlockClust` to be insensitive to the sequencing machine type and to factors like the cell line, tissue and organism. In order to evaluate the extent of `BlockClust` robustness we applied `BlockClust` on `Benchmark Data`.

Supplementary Table S4 shows that the clustering performance measured via the AUC ROC is good and consistent with the performance measured on the `Benchmark Data`. This result tells us that the parameters for `blockbuster` and the encoding parameters (such as the type of attributes and their discretization levels, and the values for the maximum radius and the maximum distance in NSPDK) are indeed insensitive to the variation of sequencing machine or organism.

3.3 Q3: annotation of known ncRNAs with encoded expression profiles

`BlockClust` can be used to extract expression-profile models for each ncRNA class and thus provides a way to automatically classify and annotate unknown deep sequencing data into known ncRNA functions. Our models have linear complexity, are extremely efficient and can be used to scan genomes and metagenomes, achieving a speed of 2–4 million reads per minute on standard hardware (Intel Core i3-2100 at 3.10 GHz) if we start from BED files as input and exclude the genome mapping phase. We tested the robustness and accuracy of these models on the major ncRNA classes: miRNAs, tRNAs and C/D-box snoRNAs. Table 3 shows results for supervised classification task averaged over 10 random test splits of `Development Data`. In Supplementary Table S5 we report very good results across a variety of conditions present in `Benchmark Data`. The classifiers that we build exhibit the same robustness that was found for the clustering task in Section 3.2. That is, models trained on the processing traces of ncRNAs in human, can be used without any re-calibration to identify the same type of ncRNA class in distant organisms such as worm, fly and plant, irrespective of changes in the sequencing machine type or the cell line or tissue. These models maintain generally a high precision (≈0.9) for tRNAs and miRNAs while they suffer from a more severe drop in recall (≈0.8 for miRNA and 0.65 for tRNA). In the case of C/D-box snoRNAs results are more variable and exhibit in general quite poor recall rates.

3.4 Q4: Performance comparison

Other approaches known in literature that can process expression profiles derived from deep sequencing data are `deepBlockAlign` (Langenberger *et al.*, 2012) and `DARIO` (Fasold *et al.*, 2011).

Table 3. Classification performance of `BlockClust` averaged over 10 random test splits of `Development Data`

ncRNA class	Number of transcripts	PPV	Recall
miRNA	168	0.901	0.886
tRNA	173	0.899	0.796
C/D-box snoRNA	78	0.870	0.474

Table 4. Metric performance: `BlockClust` versus `deepBlockAlign`

ncRNA class	Number of instances	`BlockClust` AUC ROC	`deepBlockAlign` AUC ROC
miRNA	3869	0.925	0.714
tRNA	4988	0.795	0.701
C/D-box snoRNA	731	0.762	0.615
H/ACA-box snoRNA	142	0.859	0.720
rRNA	770	0.873	0.759
snRNA	240	0.698	0.610
YRNA	244	0.694	0.656
Weighted average	11061	0.839	0.700

Comparison on `Benchmark Data`. The AUC ROC results across different species, tissues and cell lines are averaged with weight proportional to the number of instances per class.

3.4.1 Clustering performance comparison Since currently there are no available tools that can cluster expression profiles, we compare against `deepBlockAlign` even though this tool aims at solving a different problem. `deepBlockAlign` is a tool to align expression profiles which also uses `blockbuster` to generate block groups. `deepBlockAlign` uses a variant of the Sankoff algorithm to obtain an optimal alignment and computes a corresponding pairwise similarity score between expression profiles. Finally these similarities can be used to cluster expression profiles.

We applied both tools to the `Benchmark Data`. We evaluated both the quality of the similarity notion generated by the tools as well as the quality of the clusters that can be obtained under the respective similarities. In Table 4 we report the average AUC ROC for each individual ncRNA class. The class specific and weighted-averaged ROC scores indicates that `BlockClust` is highly competitive.

An additional advantage of `BlockClust` is its computational complexity and wall clock runtime. Since `deepBlockAlign` is designed with the purpose of actually generating the alignments of the read profiles, it has a quadratic complexity in the number of profiles. `BlockClust` on the other hand, is designed to solve the clustering problem and, by exploiting the hashed approximate nearest neighbors query technique, it can achieve a quasi-linear runtime. Moreover, `deepBlockAlign` uses computationally expensive algorithms like Needleman–Wunsch ($O(m^2)$) for block alignment with $m \in 15 \ldots 30$ nucleotides, and a variant of the Sankoff algorithm for block group alignment ($O(n^6)$), where $n \in 1 \ldots 5$ is the number of blocks. In contrast

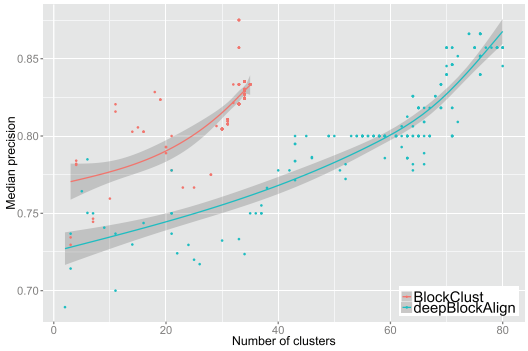


Fig. 3. Clustering performance: `BlockClust` versus `deepBlockAlign`. Comparison of median precision with respect to number of clusters on the GSM450239 dataset when the `MCL` clustering algorithm uses the expression profile similarity scores produced by `BlockClust` (red) or by `deepBlockAlign` (blue)

`BlockClust` uses explicit graph kernels with a linear complexity ($O(n)$) since it performs a simple dot product and a preprocessing attribute extraction phase that runs in $O(m)$. Not surprisingly, given the different problems that are solved, `BlockClust` achieves a speedup of 60-fold, with a wall clock runtime of 50 s as compared to 58 min for `deepBlockAlign` on a dataset of ≈600 profiles.

In addition, we evaluated the quality of the clusters that can be obtained using the `deepBlockAlign` similarity score. As we have done with `BlockClust`, we applied the `MCL` algorithm to the neighborhood graph obtained with the `deepBlockAlign` similarity score. We then compared the resulting clusters obtained varying the inflation and the pre-inflation parameters of `MCL`. In Figure 3 we report the median cluster precision versus the resulting number of clusters for different `MCL` parameter settings. We observe that `BlockClust` tends to produce larger clusters with a higher precision on average than `deepBlockAlign`.

As a final remark, note that `deepBlockAlign` was developed and optimized to identify similar processing patterns, even for different RNAs (e.g. between miRNA and some tRNAs) and it therefore might give suboptimal results when used to cluster the ncRNAs into the families of their primary function.

3.4.2 Classification performance comparison The `DARIO` tool is used in a supervised setting to classify expression profiles into known ncRNA classes. The tool also uses `blockbuster` in a pre-processing phase to identify block groups. Given a block group `DARIO` extracts a set of attributes without any need for discretization. Random forests are then employed as the underlying predictive system. Note that, differently from `BlockClust`, `DARIO` does not explicitly take the sequential arrangement of the blocks into account.

In Table 5 we report the classification performance for both tools: clearly, for all three classes `BlockClust` has better precision than `DARIO`. `BlockClust` shows higher recall for miRNAs where as `DARIO`s recall is higher for remaining two classes.

Table 5. Classification performance: `BlockClust` versus `DARIO`

	miRNA		tRNA		snoRNA C/D-box	
	PPV	Recall	PPV	Recall	PPV	Recall
`BlockClust`	0.88	0.89	0.95	0.80	0.74	0.39
`DARIO`	0.85	0.81	0.92	0.88	0.46	0.52

Comparison on the GSM769510 dataset.

Since `DARIO` cannot be run as a standalone tool, and it is accessible only via a web interface, we could not reliably compare the respective run times.

3.5 Analysis of known ncRNA clusters

To validate our approach, we clustered all block groups from the data set GSM768988. The MCL clustering produced several small clusters. We analyzed the clusters from each ncRNA class that achieved the highest precision. In Figure 4, we show the dendrogram together with representative read profiles for the selected clusters.

For tRNAs, we see very different profiles composed of a mixture of tRNA halves and 5′- or 3′-derived fragments. This can already be seen in the two examples shown in Figure 4. The left profile corresponds to a tRNA having mainly 3′-derived fragments, whereas 5′-derived fragments dominate the right profile. MicroRNAs exhibit the typical block-like structure, either with only one solid block for the miRNA, or with two blocks for miRNA and miRNA* [see Fasold *et al.* (2011) for more details and illustrations].

When examining snoRNAs, we found an even more interesting processing pattern with a step-wise extension. For that reason, we investigate the snoRNA cluster in more detail. The cluster with the highest precision contains only C/D-box snoRNAs. According to the literature (Taft *et al.*, 2009), snoRNA-derived fragments from C/D-box snoRNAs are predominantly stemming from the 5′-end. Thus, according to the literature, the profiles for the snoRNAs shown in Figure 5 should be prototypical examples. However, to our surprise, our C/D-box snoRNA cluster contained mostly 3′-derived fragments with quite some variation in length.

Finally, we examined the tRNA that was clustered together with the miRNAs (marked with a star in Figure 4). When analyzing the read profile of this tRNA we could only find very precisely cut 5′-derived fragments (see Figure 6). It is very conceivable that this tRNA might actually be processed by Dicer and/or is associated with the Argonaute complex. First, the 5′-derived fragment has a length of ≈26 nt, which is compatible with the possible lengths for miRNAs. Second, it is known that 5′-tRFs are likely to be processed by Dicer (Gebetsberger and Polacek, 2013). A miRNA-like function has been investigated in detail by Maute *et al.* (2013). Finally, it has been shown that only the 5′-derived fragments but not the 3′-derived ones are inhibiting translation and are associated with the Argonaute complex (Ivanov *et al.*, 2011).

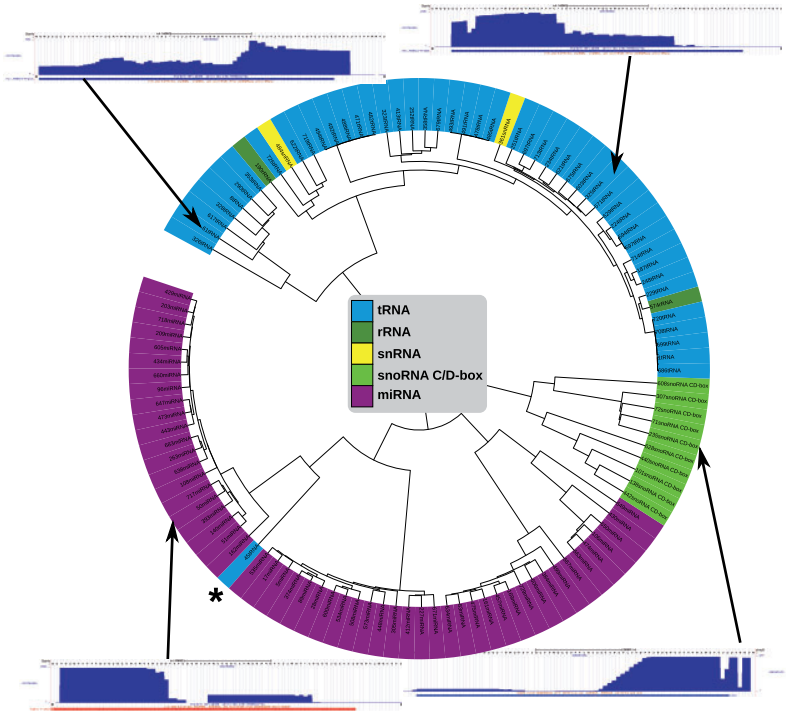


Fig. 4. Dendrogram of clusters with highest precision. One representative read profile for miRNAs and snoRNAs, and two for tRNAs are shown. The horizontal bar under each profile represents the annotation of the corresponding ncRNA

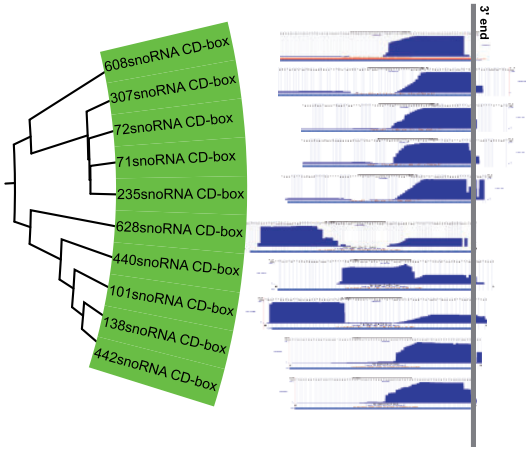


Fig. 5. Read profiles of the snoRNA cluster. All profiles are having the same scale with respect to the genomic sequence. Furthermore, they are aligned at the 3′-end of the annotated transcript

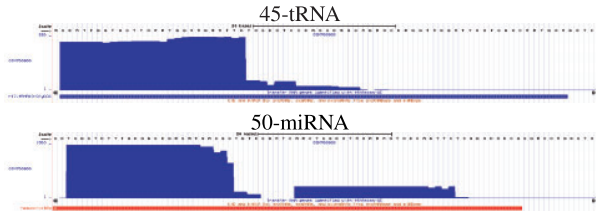


Fig. 6. Read profiles of a tRNA clustered within miRNAs and a close-by miRNA. The horizontal bar under each profile represents the annotation of the corresponding ncRNA

4 CONCLUSION

We have introduced `BlockClust`, an efficient approach to detect transcript with similar processing patterns. The procedure that we have proposed is stable with respect to changes in sequencing machines, cell lines and organisms and can be used to reliably cluster and annotate sequencing output at increasing depths. Differently from other methods, in `BlockClust` we encode expression profiles with discrete structures that can be processed efficiently and, at the same time, can retain most of the information content of the profiles.

In future work we will present the application of `BlockClust` to large deep sequencing datasets to discover novel classes of functional ncRNAs.

`BlockClust`, including all tool dependencies, is available at the Galaxy tool shed (Goecks *et al.*, 2010), and can easily be installed and used via a web interface.

Funding: German Research Foundation (DFG-grant SFB 992/1 and BA 2168/3-1 to R.B.).

Conflict of Interest: none declared.

REFERENCES

Ando,Y. *et al.* (2011) Two-step cleavage of hairpin RNA with 5′ overhangs by human DICER. *BMC Mol. Biol.*, **12**, 6.

Bernstein,B.E. *et al.* (2010) The nih roadmap epigenomics mapping consortium. *Nat. Biotechnol.*, **28**, 1045–1048.

Bottou,L. (2010) Large-Scale Machine Learning with Stochastic Gradient Descent. In: *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'2010)*. Springer, pp. 177–187.

Burge,S.W. *et al.* (2013) Rfam 11.0: 10 years of RNA families. *Nucleic Acids Res.*, **41**, D226–D232.

Costa,F. and Grave,K.D. (2010) Fast neighborhood subgraph pairwise distance kernel. In: *Proceedings of the 26th International Conference on Machine Learning*. Omnipress, pp. 255–262.

Enright,A.J. *et al.* (2002) An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res*, **30**, 1575–1584.

Fasold,M. *et al.* (2011) DARIO: a ncRNA detection and analysis tool for next-generation sequencing experiments. *Nucleic Acids Res.*, **39**, W112–W117.

Findeiss,S. *et al.* (2011) Traces of post-transcriptional RNA modifications in deep sequencing data. *Biol. Chem.*, **392**, 305–313.

Frasconi,P. *et al.* (2012) klog: A language for logical and relational learning with kernels. *CoRR*, 1205.3981.

Friedlander,M.R. *et al.* (2008) Discovering microRNAs from deep sequencing data using miRDeep. *Nat. Biotechnol.*, **26**, 407–415.

Gan,J. *et al.* (2008) A stepwise model for double-stranded RNA processing by ribonuclease III. *Mol. Microbiol.*, **67**, 143–154.

Gebetsberger,J. and Polacek,N. (2013) Slicing tRNAs to boost functional ncRNA diversity. *RNA Biol.*, **10**, 0–8.

Goecks,J. *et al.* (2010) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.*, **11**, R86.

Heyne,S. *et al.* (2012) GraphClust: alignment-free structural clustering of local RNA secondary structures. *Bioinformatics*, **28**, i224–i232.

Ivanov,P. *et al.* (2011) Angiogenin-induced tRNA fragments inhibit translation initiation. *Mol. Cell*, **43**, 613–23.

Jacquier,A. (2009) The complex eukaryotic transcriptome: unexpected pervasive transcription and novel small RNAs. *Nat. Rev. Genet.*, **10**, 833–844.

Joachims,T. (1999) Making large-scale support vector machine learning practical. In: Schölkopf,B. *et al.* (eds) *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, MA, pp. 169–184.

Kundu,K. *et al.* (2013) A graph kernel approach for alignment-free domain-peptide interaction prediction with an application to human SH3 domains. *Bioinformatics*, **29**, i335–i343.

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

Langenberger,D. *et al.* (2009) Evidence for human microRNA-offset RNAs in small RNA sequencing data. *Bioinformatics*, **25**, 2298–2301.

Langenberger,D. *et al.* (2012) deepBlockAlign: a tool for aligning RNA-seq profiles of read block patterns. *Bioinformatics*, **28**, 17–24.

Li,Z. *et al.* (2012) Extensive terminal and asymmetric processing of small RNAs from rRNAs, snoRNAs, snRNAs, and tRNAs. *Nucleic Acids Res.*, **40**, 6787–6799.

Machnicka,M.A. *et al.* (2013) MODOMICS: a database of RNA modification pathways–2013 update. *Nucleic Acids Res.*, **41**, D262–D267.

Mathews,D.H. *et al.* (2004) Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proc. Natl Acad. Sci. USA*, **101**, 7287–7292.

Maute,R.L. *et al.* (2013) tRNA-derived microRNA modulates proliferation and the DNA damage response and is down-regulated in B cell lymphoma. *Proc. Natl Acad. Sci. USA*, **110**, 1404–1409.

Morin,R.D. *et al.* (2008) Application of massively parallel sequencing to microRNA profiling and discovery in human embryonic stem cells. *Genome Res.*, **18**, 610–621.

Nishikura,K. (2010) Functions and regulation of RNA editing by ADAR deaminases. *Ann. Rev. Biochem.*, **79**, 321–349.

Parker,B.J. *et al.* (2011) New families of human regulatory RNA structures identified by comparative analysis of vertebrate genomes. *Genome Res.*, **21**, 1929–1943.

Rederstorff,M. *et al.* (2010) RNPomics: defining the ncRNA transcriptome by cDNA library generation from ribonucleo-protein particles. *Nucleic Acids Res.*, **38**, e113.

Rinn,J.L. and Chang,H.Y. (2012) Genome regulation by long noncoding RNAs. *Ann. Rev. Biochem.*, **81**, 145–166.

Saito,Y. *et al.* (2011) Fast and accurate clustering of noncoding RNAs using ensembles of sequence alignments and secondary structures. *BMC Bioinform.*, **12** (**Suppl 1**), S48.

Shi,Y. *et al.* (2009) Metatranscriptomics reveals unique microbial small RNAs in the ocean's water column. *Nature*, **459**, 266–269.

Su,A.A.H. and Randau,L. (2011) A-to-I and C-to-U editing within transfer RNAs. *Biochemistry (Mosc)*, **76**, 932–937.

Taft,R.J. *et al.* (2009) Small RNAs derived from snoRNAs. *RNA*, **15**, 1233–1240.

Torarinsson,E. *et al.* (2007) Multiple structural alignment and clustering of RNA sequences. *Bioinformatics*, **23**, 926–932.

Weinberg,Z. *et al.* (2009) Exceptional structured noncoding RNAs revealed by bacterial metagenome analysis. *Nature*, **462**, 656–659.

Will,S. *et al.* (2007) Inferring non-coding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comput. Biol.*, **3**, e65.

[P2] FOXG1 Regulates PRKAR2B Transcriptionally and Posttranscriptionally via miR200 in the Adult Hippocampus

Stefan C. Weise*, Ganeshkumar Arumugam*, Alejandro Villarreal*, **Pavankumar Videm***, Stefanie Heidrich, Nils Nebel, Verónica I. Dumit, Farahnaz Sananbenesi, Viktoria Reimann, Madeline Craske, Oliver Schilling, Wolfgang R. Hess, Andre Fischer, Rolf Backofen & Tanja Vogel. *Molecular Neurobiology*, 2019. DOI: 10.1007/s12035-018-1444-7

Contributions of individual authors:
I have made an important contribution to this work, which led to shared first authorship. I analyzed the data from Foxg1$^{cre/+}$ and miR-200 over-expression RNA-Seq experiments. I made the sequencing data available on sequence read archive and gene expression omnibus. I also performed the transcription factor motif analysis and Foxg1 ChIP-Seq data analysis to identify putative Foxg1 binding sites. I wrote the bioinformatics methods related to all my data analysis and revised the manuscript. Farahnaz Sananbenesi and Andre Fischer analyzed the miRNA-Seq data. The other authors, namely Stefan C. Weise, Ganeshkumar Arumugam, Alejandro Villarreal, Stefanie Heidrich, Nils Nebel, Verónica I. Dumit, Viktoria Reimann, Madeline Craske, Oliver Schilling, Wolfgang R. Hess and Tanja Vogel designed, conducted the wet lab experiments. Rolf Backofen and Tanja Vogel provided general project consultation. All the authors were involved in the assessment of results and manuscript revision.
Pavankumar Videm

The following authors confirm the above stated contributions.

- Stefan C. Weise
- Ganeshkumar Arumugam
- Alejandro Villarreal
- Pavankumar Videm
- Nils Nebel
- Stefanie Heidrich
- Madeline Craske
- Viktoria Reimann
- Wolfgang R. Hess
- Oliver Schilling
- Rolf Backofen
- Andre Fischer
- Tanja Vogel

* Joint first authors

Molecular Neurobiology (2019) 56:5188–5201
https://doi.org/10.1007/s12035-018-1444-7

FOXG1 Regulates PRKAR2B Transcriptionally and Posttranscriptionally via miR200 in the Adult Hippocampus

Stefan C. Weise[1,2,3] · Ganeshkumar Arumugam[1,2,4] · Alejandro Villarreal[1] · Pavankumar Videm[5] · Stefanie Heidrich[1] · Nils Nebel[1] · Verónica I. Dumit[6] · Farahnaz Sananbenesi[7] · Viktoria Reimann[8] · Madeline Craske[9] · Oliver Schilling[10,11] · Wolfgang R. Hess[8,12] · Andre Fischer[7,13,14] · Rolf Backofen[5,6,10,15] · Tanja Vogel[1]

Received: 7 May 2018 / Accepted: 30 November 2018 / Published online: 11 December 2018

Abstract

Rett syndrome is a complex neurodevelopmental disorder that is mainly caused by mutations in *MECP2*. However, mutations in *FOXG1* cause a less frequent form of atypical Rett syndrome, called FOXG1 syndrome. FOXG1 is a key transcription factor crucial for forebrain development, where it maintains the balance between progenitor proliferation and neuronal differentiation. Using genome-wide small RNA sequencing and quantitative proteomics, we identified that FOXG1 affects the biogenesis of miR200b/a/429 and interacts with the ATP-dependent RNA helicase, DDX5/p68. Both FOXG1 and DDX5 associate with the microprocessor complex, whereby DDX5 recruits FOXG1 to DROSHA. RNA-Seq analyses of *Foxg1*$^{cre/+}$ hippocampi and N2a cells overexpressing miR200 family members identified cAMP-dependent protein kinase type II-beta regulatory subunit (PRKAR2B) as a target of miR200 in neural cells. PRKAR2B inhibits postsynaptic functions by attenuating protein kinase A (PKA) activity; thus, increased PRKAR2B levels may contribute to neuronal dysfunctions in FOXG1 syndrome. Our data suggest that FOXG1 regulates PRKAR2B expression both on transcriptional and posttranscriptional levels.

Keywords DROSHA · Atypical Rett syndrome · MECP2 · Neurogenesis · PKA

Introduction

Rett syndrome (RTT) is a progressive neurodevelopmental disorder that affects one in 10,000 females, and it is the second leading cause of female intellectual deficiency. RTT patients show symptoms such as microcephaly, seizures, indifference to visual/auditory stimuli, and severe cognitive dysfunction [1]. There are two forms of RTT, namely typical RTT (tRTT) and atypical RTT-like (atRTT). About 70–90% of cases are due to tRTT, which results from mutations in the Methyl CpG binding Protein 2 (*MECP2*) gene. Different subvariants of atRTT have been defined, one of which is caused by mutations in the Forkhead box G1 (*FOXG1*) gene (FOXG1 syndrome, OMIM#164874). Surprisingly, both loss- and gain-of-function mutations result in clinical phenotypes, which encompass common (e.g. seizures) but also unique features (e.g. spasms). Rett-like syndrome and epilepsy have been associated with *FOXG1*-haploinsufficiency [2]. Loss- and gain-of-function mutations are reported also for both tRTT and atRTT, whereby gain-of-function is caused by gene duplication of either *MECP2* or *FOXG1* [3, 4]. Therefore, both gene products seem to be associated with functions that are dosage sensitive, although these functions are so far ill defined, especially for FOXG1.

FOXG1 plays a central role in forebrain development as its complete absence results in anencephaly [5]. FOXG1 influences proliferation as well as differentiation of neural stem cells, and it is involved in migration and integration of pyramidal neurons into the cortical plate [6]. FOXG1-deficient stem cells differentiate prematurely to Cajal–Retzius neurons, whereas overexpression of FOXG1 increases the stem cell pool and delays neurogenesis [7, 8]. On a molecular level, FOXG1 represses expression of cyclin-dependent kinase inhibitor 1A (*Cdkn1a*) and thereby prevents cell cycle exit of progenitor cells and promotes stem cell pool expansion [9, 10]. Cell cycle regulation through FOXG1 is mediated by its

Stefan C. Weise, Ganeshkumar Arumugam, Alejandro Villarreal and Pavankumar Videm contributed equally to this work.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s12035-018-1444-7) contains supplementary material, which is available to authorized users.

✉ Tanja Vogel
tanja.vogel@anat.uni-freiburg.de

Extended author information available on the last page of the article

binding to Forkhead box O− (FOXO−) and SMAD− (SMA and MAD related−) protein complexes and by antagonising TGFβ-induced neuronal differentiation [11]. In addition, FOXG1 deficiency results in the loss of the ventral telencephalon through impaired expression of ventralising signals [12]. Also, FOXG1 interacts with one of two MECP2-isoforms (MECP2-e2), which prevents cell death of cerebellar neurons [13]. Several mouse models were used to study the molecular basis of tRTT and atRTT, and some of these studies included non-coding RNA (ncRNA), such as miRNAs. Whereas altered expression of ncRNA is involved in *MECP2*-mediated RTT [14–16], a comprehensive expression study of the misregulated coding and non-coding transcriptome is missing for FOXG1 haploinsufficient adult brains.

Here, we report on altered expression of members of the miRNA200 family, namely miR200a, miR200b and miR429 in the adult $Foxg1^{cre/+}$ hippocampus. Stable isotope labelling with amino acids in cell culture (SILAC) followed by quantitative mass spectrometry revealed that FOXG1 associates with the RNA helicase DDX5 (DEAD (Asp-Glu-Ala-Asp) box polypeptide 5, p68). DDX5 recruits FOXG1 to the DROSHA complex, and FOXG1 overexpression alongside with reduced levels of DDX5 affects biogenesis of miR200 family members. Decreased expression of FOXG1 and overexpression of miR200 result in altered expression levels of protein kinase cAMP-dependent regulatory type II beta (PRKAR2B). As PRKAR2B influences synaptic function, our results reveal a novel candidate gene, whose altered expression might be implicated in FOXG1 syndrome. Additionally, we establish that FOXG1 has functions in post-transcriptional regulation besides its known role as transcription factor [6, 15].

Material and Methods

Information on cell culture conditions, transfections and plasmids used in this study, on cell fractionation, mouse hippocampus dissection, culture of neurons and viral transduction, RNA immunoprecipitation (RIP), RNA isolation, reverse transcription and quantitative real-time PCR (qRT-PCR) and luciferase assays, as well as on miRNA analysis by Northern hybridization are found in the Supplementary Material and Methods.

Mice

The animal welfare committees of the University of Freiburg and local authorities approved all mouse experiments, registered under the licence G14-096 or X14/04H. $Foxg1^{cre/+}$ mice [17] were maintained in a C56BL/6 background. For experiments with wild-type (WT) mice, NMRI was used either at E13.5 or adult stages.

SILAC and Mass Spectrometry

N2a cells were cultured in SILAC DMEM without arginine and lysine (#89985 ThermoScientific, Bremen, Germany) supplemented with Lys0/Arg0 or Lys8/Arg10 (0.398 mM L-arginine $^{13}C_6{}^{15}N_4$, 0.798 mM L-lysine $^{13}C_6{}^{15}N_2$, Euriso-Top, Saarbrücken, Germany) and 10% dialysed FCS (ThermoScientific). N2a cells were labelled for 12 passages with SILAC medium. Further processing is described in the Supplementary Material and Methods.

Analysis of the protein groups was done with the Perseus software [18]. Only proteins with two or more unique identified peptides, which were enriched more than 2-fold in both experiments, were considered. All raw data and original result files were deposited to the ProteomeXchange Consortium (http://proteomecentral.proteomexchange.org) via the PRIDE partner repository with the dataset identifier PXD007040.

Immunoprecipitation

Immunoprecipitation (IP) was performed according to standard procedures as outlined in the Supplementary Material and Methods. For co-immunoprecipitations (co-IPs) with tagged FOXG1 (either FOXG1-Au1 or FOXG1-HA), mock conditions were either untransfected, empty vector or FOXG1 with the other tag. The following antibodies were used for IP: anti-Au1-tag (MMs-130R, Covance, Koblenz, Germany), anti-HA-tag (#3724, Cell Signaling), anti-DDX5 (rabbit, ab126730, abcam, Cambridge, UK), anti-FOXG1 (rabbit, ab18259, abcam) and anti-DROSHA (rabbit, ab12286, abcam).

Immunoblotting

Immunoblotting was performed according to standard procedures as outlined in the Supplementary Material and Methods. The following antibodies were used for immunoblots: anti-Au1-tag (1:1000, MMs-130R, Covance), anti-HA-tag (1:1000, #3724, Cell Signaling), anti-DDX5 (1:2000, rabbit, ab126730, abcam), anti-FOXG1 (1:1000, rabbit, ab18259, abcam), anti-DROSHA (1:1000, rabbit, ab12286, abcam), anti-DGCR8 (1:1000, rabbit, ab191875, abcam), anti-H3 (1:1000, goat, ab12079, abcam), anti-NPM1 (1:1000, mouse, ab10530, abcam), anti-PRKAR2B (1:1000, DAKO) and anti-GAPDH (1:3000, ab8245, abcam). Densitometric analyses were done with ImageJ.

RNA-Seq and Small RNA-Seq

Total RNA was prepared from the hippocampus with RNeasy kits (Qiagen), including on-column DNAse digestion. Samples were depleted from rRNA using RiboZero Gold kit (Illumina) before sequencing. Quality of the RNA was

assessed with QIAxcel (#9001941, Qiagen, Hilden, Germany). Samples were prepared and analysed with Illumina HiSeq2500 (paired end, multiplexing run, 75 Mio/reads per sample). Bioinformatics analysis was performed using the Freiburger Galaxy Server [19, 20] as described in the Supplementary Material and Methods. For small RNA-Seq, 6-week-old *Foxg1*$^{cre/+}$ mice hippocampi were used ($n = 9$). Raw data were deposited at the GEO database under the following accession numbers: small-RNAseq: GSE104169, mir-200 OE RNA-Seq: GSE106802 and *Foxg1*$^{cre/+}$ hippocampus RNA-Seq: GSE106801.

Proximity Ligation Assay

Proximity ligation assay (PLA) was performed with the Duolink starter kit reagents (DUO92103, SIGMA) according to the manufacturer's instruction as outlined in the Supplementary Material and Methods. The following antibodies were used: anti-Au1 (1:2000, mouse, Covance) and anti-DDX5 (1:200, goat, ab10261, abcam) for PLA and anti-Lamin B1 (1:200, rabbit, ab133741, abcam). Images were taken with a confocal microscope and analysed with the LASX software (SP8, Leica, Jena, Germany).

Statistical Analysis

GraphPad Prism software was used for statistical analyses. Statistical tests are indicated in the respective figure legends. Values in bar charts are expressed as average ± SEM. In in vivo experiments, each independent *N* is a different animal, and in in vitro experiments, each *N* is a different passage of cells. One sample Student's *t* test was performed (e.g. on ΔΔCt values of qRT-PCRs) if measured variables could be paired (e.g. control and treatment of the same passage of cells). Unpaired Student's *t* test (equal variances) or Welch's *t* test (unequal variances) was used if variables were not paired (using in these cases for example ΔCt values for each sample group).

Final figures were prepared using FIJI (ImageJ, v. 2.0.0-rc-43/1.51d [21]) and Inkscape (v. 0.91).

Results

Foxg1$^{cre/+}$ Animals Express Reduced Levels of Mature and Precursor miR200b/a/429 in the Hippocampus

As altered expression of miRNAs has been identified in tRTT [14, 15], we aimed to determine if miRNA expression was altered in a FOXG1 syndrome mouse model. We used 6-week-old *Foxg1*$^{cre/+}$ mice, which expressed approximately half the amount of FOXG1 protein in the hippocampus compared to WT littermates (Fig. 1a) and performed small RNA sequencing (RNA-Seq) with hippocampi of *Foxg1*$^{cre/+}$ and WT mice. This experiment revealed in total 11 small RNAs, including ten miRNAs, which were significantly altered with a fold change of at least ± 1.5 (Fig. 1b). Altered expression levels of these ten miRNAs were validated by qRT-PCR (Fig. 1c). Seven miRNAs, namely miR200a, miR200b, miR429, miR448, miR764, miR1264 and miR1298, had more than 2-fold and significantly altered expression levels in hippocampi of *Foxg1*$^{cre/+}$ mice (Fig. 1b). Out of these, miR200b/a/429, which were decreased in *Foxg1*$^{cre/+}$ hippocampus, derived from a single transcript (*Gm13648*). miR448/764/1264/1298, which were increased in *Foxg1*$^{cre/+}$ hippocampi, derived from the 5-hydroxytryptamine receptor 2C (*Htr2c*) transcript. Several reports suggested that miR200 family members control similar processes as FOXG1 in the developing cerebral cortex [5, 9, 22–28]. However, the functions of miR448/764/1264/1298 from the *Htr2c* gene are not known yet. We therefore decided to study the influence of FOXG1 on miR200 in more detail and performed qRT-PCR for precursor (pre-) miRNAs in 6-week-old male *Foxg1*$^{cre/+}$ mice hippocampi. Pre-miR200a transcripts were significantly reduced in the hippocampus of the *Foxg1*$^{cre/+}$ mice, whereas reduced levels of pre-miR200b did not reach significance and pre-miR429 was not detected (Fig. 1d). Expression of the primary transcript was neither detected by RNA-Seq nor by qRT-PCR in vivo (data not shown), suggesting very low expression levels and/or high turnover, e.g. by co-transcriptional processing [29]. Together, these data indicated that reduced levels of FOXG1 occurred alongside with reduced levels of pre-miR200a as well as mature miRNA 200b/a/429 levels in the adult hippocampus.

Prkar2b Is a Target of FOXG1 and miR200 Family

To identify miR200 targets with a putative role in FOXG1 syndrome, we performed RNA-Seq after overexpressing miR200 family members in N2a cells and compared it with RNA-Seq data obtained from adult *Foxg1*$^{cre/+}$ hippocampus. In total, we identified 2081 differentially expressed genes after overexpressing miR200 family members and 382 genes in *Foxg1*$^{cre/+}$ hippocampus. Intersection of the two RNA-Seq datasets revealed 35 genes shared between both datasets (Fig. 2a). The intersection of two independent datasets within a finite population size of 43,629 genes can be modelled as a hypergeometric distribution. We performed a hypergeometric test, which revealed a p value of 0.0001 that suggested that the overlap of 35 out of 382 *Foxg1*$^{cre/+}$ and 2081 miR200 differentially expressed genes is not by chance but rather significant.

Gene ontology (GO) term analysis revealed that this set of 35 genes is classified to processes like neuronal projection and development (Fig. 2b). As the *Foxg1*$^{cre/+}$ hippocampus expressed less mature miR200b/a/429, we focused on genes

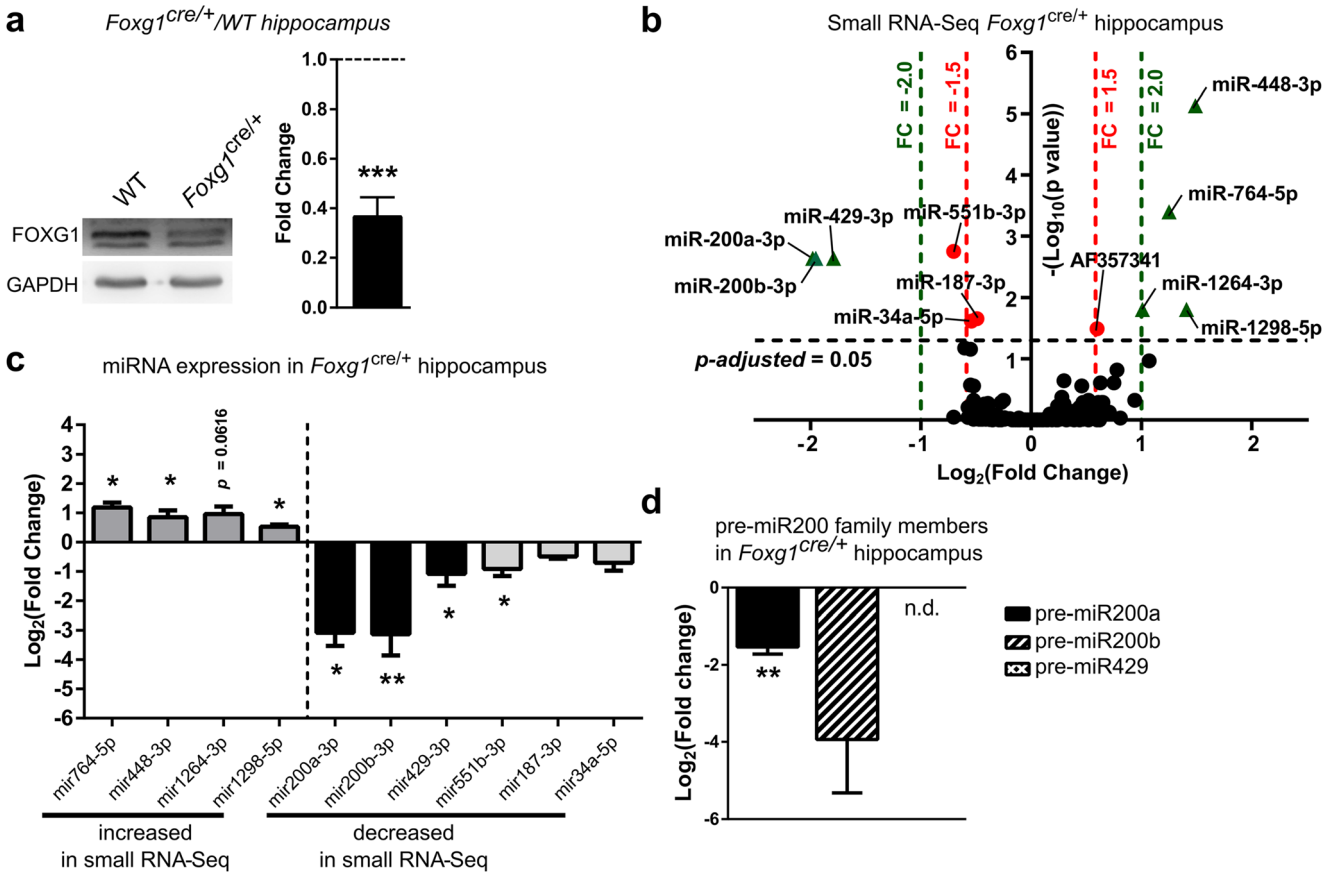
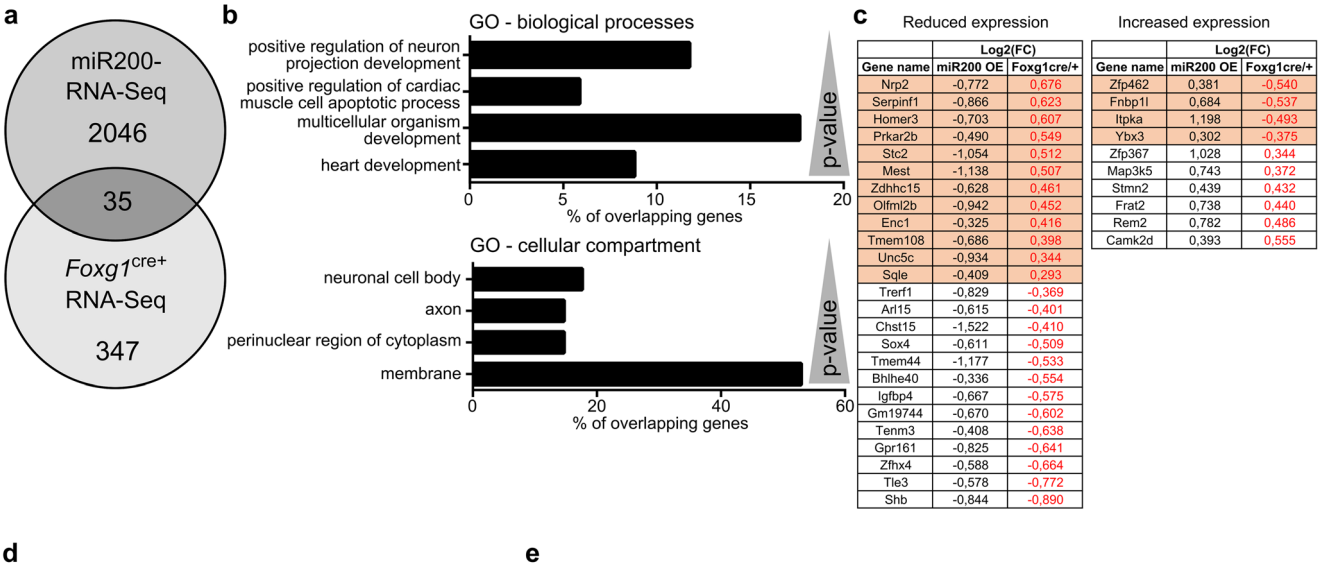


Fig. 1 *Foxg1*$^{cre/+}$ adult hippocampus expresses altered miRNA levels of miR200b/a/429 and *Htr2c* families. **a** Representative immunoblot and quantification of FOXG1 protein levels in *Foxg1*$^{cre/+}$ and WT hippocampus show that FOXG1 protein is reduced by 60% in *Foxg1*$^{cre/+}$ compared to control levels (dashed line). Mean with SEM, ***$p < 0.001$, one-sample Student's *t* test. $n = 6$. **b** Volcano plot of small RNA-Seq of *Foxg1*$^{cre/+}$ compared to WT hippocampus indicates that expression of miR200b/a/429 family is significantly decreased, whereas miRNAs from *Htr2c* gene significantly increased (dashed lines: DEseq2 *p*-adjusted value = 0.05 (black); FC = ± 2.0 (green); FC = ± 1.5 (red)). Values for adjusted *p* value were plotted on the *y*-axis. $n = 9$. **c** qRT-PCR validation confirms significantly altered expression levels of miRNAs of miR200b/a/429 and *Htr2c* families in 6-week-old hippocampus of *Foxg1*$^{cre/+}$ animals. Mean with SEM, *$p < 0.05$, **$p < 0.01$, unpaired Student's *t* test. $n = 3$. **d** Expression of precursor transcripts of miR200b/a/429 in *Foxg1*$^{cre/+}$ hippocampus using qRT-PCR reveals decreased expression of pre-miR200b and pre-miR200a, while expression of pre-miR429 was not detectable (n.d.). Mean with SEM, *$p < 0.05$, **$p < 0.01$, unpaired Student's *t* test. $n = 3$

with opposing expression levels after miR200 overexpression compared to *Foxg1*$^{cre/+}$ RNA-Seq to identify putative targets. Twelve of these genes showed reduced levels after miR200 overexpression, whereas four increased in expression (highlighted in Fig. 2c). We used different target prediction tools to analyse miR200 seed sequences on the mRNAs of these 16 genes. Three of the four prediction algorithms identified putative miR200 binding sites in the 3′-untranslated region (UTR) of *Prkar2b* (Fig. 2d). We subsequently assessed altered expression of 13 candidates by qRT-PCR in vitro and in vivo. *Prkar2b* mRNA levels decreased upon overexpression of the miR200 family members and increased in *Foxg1*$^{cre/+}$ hippocampus (Fig. 2e) as predicted. Other candidates, i.e. *Serpinf1*, *Olfml2b* and *Tmem108*, were either significantly altered in only one condition or were altered in an opposing direction compared to the RNA-Seq data. These expression analyses therefore rendered *Prkar2b* as the best candidate for altered expression through FOXG1 and/or miR200 family.

Next, we analysed altered protein levels of PRKAR2B in the *Foxg1*$^{cre/+}$ hippocampus using immunoblotting. As anticipated, protein levels of PRKAR2B increased with reduced levels of FOXG1 in vivo (Fig. 3a, c) and decreased after overexpression of miR200 compared to a miR200 sponge in N2a cells (Fig. 3b, c). To further show that the *Prkar2b* 3′UTR was targeted by miR200b/a/429, we used a luciferase reporter assay. Overexpression of miR200b/a/429 together with a plasmid carrying luciferase followed by the WT 3′UTR of *Prkar2b* reduced the luciferase signal. In contrast, inverted or T_7 replaced seed sequences in the 3′UTR of *Prkar2b* did not affect luciferase expression (Fig. 3d). We therefore concluded that *Prkar2b* was a direct target of miR200 family members.

As FOXG1 is described as a transcription factor, we analysed FOXG1 chromatin-immunoprecipitation sequencing



Reduced expression	Log2(FC)	
Gene name	miR200 OE	Foxg1cre/+
Nrp2	-0,772	0,676
Serpinf1	-0,866	0,623
Homer3	-0,703	0,607
Prkar2b	-0,490	0,549
Stc2	-1,054	0,512
Mest	-1,138	0,507
Zdhhc15	-0,628	0,461
Olfml2b	-0,942	0,452
Enc1	-0,325	0,416
Tmem108	-0,686	0,398
Unc5c	-0,934	0,344
Sqle	-0,409	0,293
Trerf1	-0,829	-0,369
Arl15	-0,615	-0,401
Chst15	-1,522	-0,410
Sox4	-0,611	-0,509
Tmem44	-1,177	-0,533
Bhlhe40	-0,336	-0,554
Igfbp4	-0,667	-0,575
Gm19744	-0,670	-0,602
Tenm3	-0,408	-0,638
Gpr161	-0,825	-0,641
Zfhx4	-0,588	-0,664
Tle3	-0,578	-0,772
Shb	-0,844	-0,890

Increased expression	Log2(FC)	
Gene name	miR200 OE	Foxg1cre/+
Zfp462	0,381	-0,540
Fnbp1l	0,684	-0,537
Itpka	1,198	-0,493
Ybx3	0,302	-0,375
Zfp367	1,028	0,344
Map3k5	0,743	0,372
Stmn2	0,439	0,432
Frat2	0,738	0,440
Rem2	0,782	0,486
Camk2d	0,393	0,555

Gene Name	TargetScan	miRanda	miRDB	Microcosm
Nrp2	miR200b/a/429	miR200b/a/429	X	X
Serpinf1	X	X	X	X
Homer3	X	X	X	X
Prkar2b	miR200b/429	miR200b/429	X	miR200b/429
Stc2	X	X	X	X
Mest	X	miR200a	X	X
Zdhhc15	X	X	miR200b/429	X
Olfml2b	X	X	X	X
Enc1	X	miR200a	X	X
Tmem108	X	X	X	X
Unc5c	X	X	miR200a	miR200a/429
Sqle	X	miR200b/429	X	miR200b/429
Zfp462	X	miR200a	X	X
Fnbp1l	X	X	X	X
Itpka	X	X	X	X
Ybx3	X	X	X	X

Fig. 2 RNA-Seq after miR200 overexpression and of *Foxg1*$^{cre/+}$ hippocampus identify *Prkar2b* as a miR200 target in the hippocampus. **a** Venn diagram depicting the overlap of 35 differentially expressed genes from RNA-Seq of N2a cells overexpressing miR200 family and from RNA-Seq of *Foxg1*$^{cre/+}$ hippocampus. $n = 2$ and $n = 3$, respectively. **b** Bar chart of a DAVID GO term analysis for biological processes and cellular compartments of the 35 overlapping genes displayed in **a**. *p* values (as reported by DAVID) are given in the range of 0.0016 to 0.075. **c** Thirty-five overlapping genes with reduced or increased expression after miRNA200 family overexpression. Given is the $\log_2$(fold change) of both RNA-Seq datasets. Highlighted are genes, which show opposing expression level changes after miR200 family overexpression and in condition of less FOXG1 expression in *Foxg1*$^{cre/+}$ hippocampus. **d** Table showing which of the 16 genes highlighted in **c** might be putative targets of miR200 family using four different prediction algorithms. Three out of four prediction algorithms identify miR200b and miR429 seed sequences in *Prkar2b*. **e** qRT-PCR validation of putative miR200 family target genes from **c**, together with *Zeb1*, which served as control for miR200 overexpression. miR200 family overexpression in N2a decreases *Prkar2b* levels compared to untransfected N2a cells, which are in turn increased in *Foxg1*$^{cre/+}$ hippocampus with reduced levels of miR200 family member expression compared to wild type. Mean with SEM, $*p < 0.05$, $**p < 0.01$, Student's *t* test (hippocampus samples) and one-sample Student's *t* test (miR200 overexpression in N2a cells). $n = 3$–4

(ChIP-Seq) from E14.5 cortical tissue (available through the Active Motif web site) and own ChIP-Seq data from adult hippocampus (data not shown) for enriched genomic regions around the *Prkar2b* and *Gm13648* genes. We identified a peak at the 5′ end of *Prkar2b*, whereas no enrichment was observed in *Gm13648* (Fig. 4a, b). We therefore analysed whether FOXG1 suppressed *Prkar2b* expression through regulative sequences at the 5′ end. First, overexpression of full-length FOXG1 in N2a cells resulted in decreased transcription of *Prkar2b*. Interestingly, overexpression of a Forkhead box-deficient variant of FOXG1 also led to a reduction, albeit smaller than in the presence of full-length FOXG1 (Fig. 4c). We next used a luciferase assay to verify putative regulative sequences within the 5′ end of *Prkar2b*. We cloned the 5′ region of the *Prkar2b* gene containing a predicted Forkhead box binding site (Fig. 4a) upstream of a luciferase reporter. In this assay, overexpression of FOXG1 reduced luciferase activity compared to the empty vector control (Fig. 4d). These results indicated that FOXG1 suppressed *Prkar2b* transcription directly in addition to its degradation activities via miR200 family members. Together, the data provided strong evidence that PRKAR2B is a novel candidate protein, misexpression of which might be implicated in FOXG1 syndrome.

To investigate if levels of miR200 family members or *Prkar2b* were altered in other regions than the hippocampus of *Foxg1*$^{cre/+}$ animals, we assessed the respective transcript levels in samples derived from the cerebral cortex and

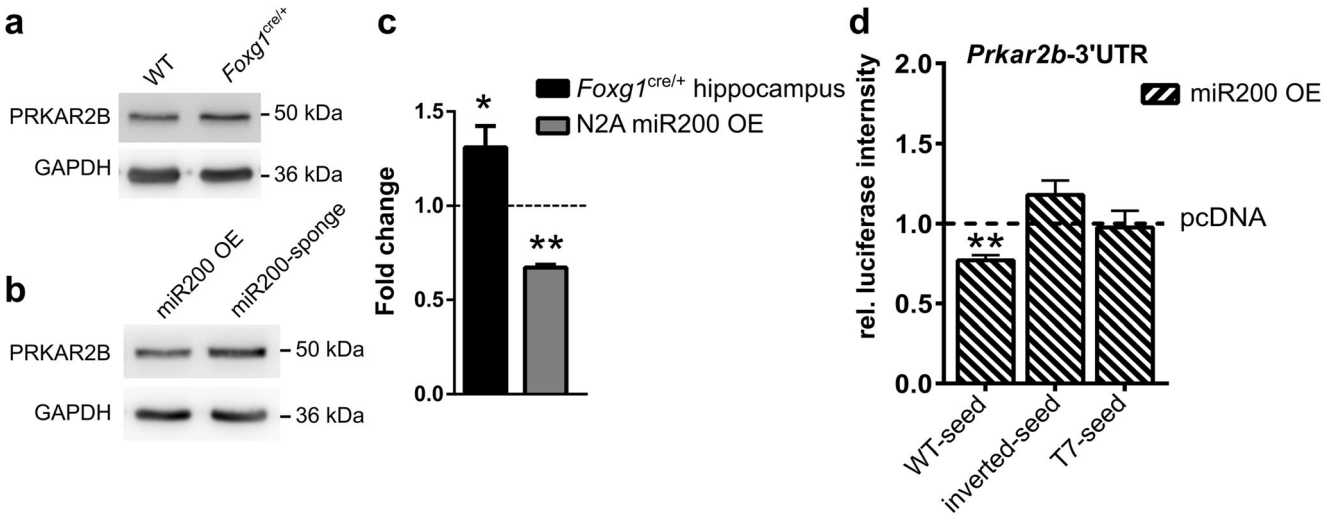


Fig. 3 miR200b/a/429 overexpression reduces PRKAR2B protein levels. **a** Immunoblots of WT and *Foxg1*$^{cre/+}$ hippocampus using anti-PRKAR2B antibodies show increased expression in the adult hippocampus of *Foxg1*$^{cre/+}$ animals. $n = 7$. **b** Immunoblots of N2a cells overexpressing miR200 family or miR200 sponge plasmids. Overexpression of miR200 family reduces PRKAR2B levels compared to miR200 sponge-transfected N2a cells. $n = 3$. **c** Densitometric quantification of **a** and **b**. Reduced levels of FOXG1 result in increased levels of PRKAR2B in the adult hippocampus. miR200 family member overexpression reduces PRKAR2B levels significantly when compared to miR200 sponge (dashed line represents control expression levels). Mean with SEM, $*p < 0.05$, $**p < 0.01$, one-sample Student's *t* test. $n = 7$. *Foxg1*$^{cre/+}$ hippocampus, and $n = 3$ for miR200 OE. **d** Results of luciferase assays to detect degradation of a transcript containing a wild type (WT), inverted or T_7 seed sequence in the *Prakr2b*-3′UTR. Overexpression of miR200b/a/429 degrades transcripts with WT-seed but not with inverted or T_7 seed in the *Prkar2b*-3′UTR, when compared to control vector expression (dashed line). Mean with SEM, $**p < 0.01$, one-sample Student's *t* test. $n = 3$–5

olfactory bulb. Neither mature miR200b/a/429 nor *Prkar2b* levels changed in the cerebral cortex or olfactory bulb in *Foxg1*$^{cre/+}$ compared to WT animals (Fig. S1a–c).

FOXG1 Interacts with DDX5 and DROSHA Microprocessor Complex

As our data indicated that FOXG1 was not directly involved in transcriptional control of the pri-miR200 transcript, we aimed to identify protein interaction partners of FOXG1 that might explain the posttranscriptional effects observed. We overexpressed FOXG1 in SILAC-labelled N2a cells and performed FOXG1 co-IP followed by quantitative mass spectrometry (MS). MS analysis identified 701 proteins with at least two unique identified peptides, which were enriched more than 2-fold by FOXG1 co-IP (Fig. 5a). We analysed the MS dataset using *Database for Annotation, Visualization and Integrated Discovery* (DAVID) which revealed *Kyoto Encyclopedia of Genes and Genomes* (KEGG) pathway terms related to RNA metabolism, i.e. spliceosome, RNA transport and RNA degradation (Fig. 5b). These data

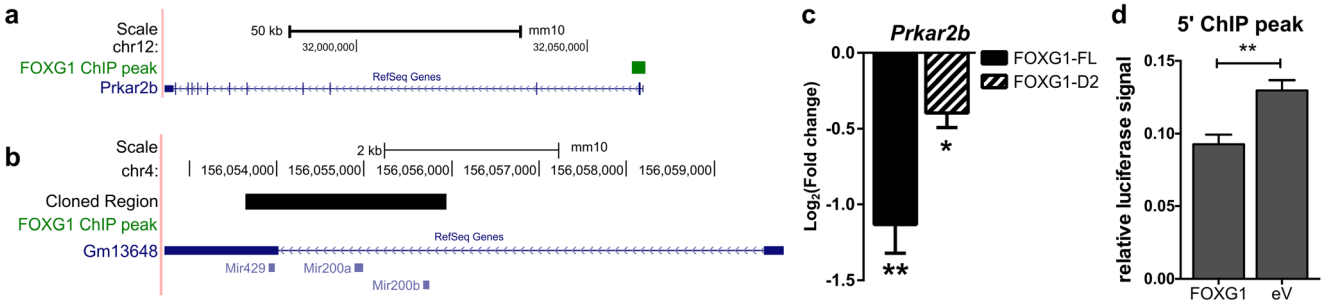


Fig. 4 FOXG1 suppresses *Prkar2b* expression through direct and indirect mechanisms. **a** UCSC Genome browser view of *Prkar2b* gene region indicating FOXG1 binding site (FOXG1 ChIP peak, shown in green) from FOXG1 adult hippocampus ChIP-Seq data. **b** UCSC genome browser view of miR200b/a/429 gene showing the cloned region in the black bar, which does not include promoter, 5′ and 3′UTR regions. No FOXG1 binding sites were predicted in the miR200b/a/429 gene. **c** Transcript levels of *Prkar2b* are reduced after FOXG1-FL and FOXG1-D2 expression in N2a cells compared to empty vector-transfected cells. Mean with SEM, $*p < 0.05$, $**p < 0.01$. $n = 5$. **d** Results of luciferase assays to detect transcriptional influence of FOXG1 on the region around the peak enriched after FOXG1–ChIP-Seq in the 5′ region of the *Prkar2b* gene. FOXG1 suppresses luciferase activity significantly, when compared to control. Mean with SEM, $**p < 0.01$, unpaired Student's *t* test. $n = 5$

strongly suggested that FOXG1 might be involved in RNA metabolism in addition to its transcriptional repressor activity. The spliceosome pathway had a p value of 8.71E-25, and one of the strongest enriched proteins overall and among RNA binding proteins was the ATP-dependent RNA helicase DDX5 (Fig. 5a). DDX5 regulates posttranscriptional control of gene expression at various levels [30]. Posttranscriptional control is affected in MECP2-mediated Rett syndrome [31] as well as in other autism spectrum disorders [32]. Therefore, we decided to focus on the interaction between FOXG1 and DDX5 to elucidate a novel function for FOXG1 in posttranscriptional RNA regulation. We confirmed this novel interaction between FOXG1 and DDX5 after overexpression of HA- or Au1-tagged FOXG1 in N2a cells (Fig. 5c), or with endogenous FOXG1 in adult mouse hippocampus (Fig. 5d) using co-IP followed by immunoblotting. FOXG1 interacted with DDX5 both in vitro and in vivo. Since DDX5 associates with the microprocessor complex [33] and has an important role in miRNA maturation [34], we probed FOXG1-co-IP samples for DROSHA and DiGeorge syndrome critical region gene 8 (DGCR8). FOXG1 interacted with both DROSHA and DGCR8 in vitro and in vivo (Fig. 5c, d), suggesting that FOXG1 may influence miRNA maturation.

FOXG1–DDX5 Complex Interacts with the Microprocessor in the Nucleus

The maturation process of miRNAs is spatially separated. The nuclear DROSHA microprocessor excises the pre-miRNA from the primary transcript, whereas subsequent cleavage of mature miRNAs in the form of the stem loop occurs through the cytoplasmic DICER complex. To identify the cellular localization of the FOXG1–DDX5 complex, we used co-IPs from fractionated cells. In N2a cells, DROSHA localised not exclusively to the nucleus but was detected at significant levels in the cytoplasm. Cytoplasmic localisation of DROSHA depends on phosphorylation [35, 36] and splicing [37]. Despite the enriched cytoplasmic localisation of DROSHA, we precipitated FOXG1 along with DDX5, DROSHA and DGCR8 from the nucleoplasmic and chromatin fraction (Fig. 6a). We used PLA and confirmed that the FOXG1–DDX5 complex localised to the nucleus in vivo in N2a cells (Fig. 6b). Confocal imaging and 3D stacking of FOXG1–DDX5 PLA in N2a cells revealed that the PLA signal of FOXG1–DDX5 localised to the proximity of the inner nuclear membrane (Fig. 6c). This result suggested that nuclear FOXG1–DDX5 complexes did not directly affect DICER-mediated processing, as their interaction was restricted to the nuclear compartment.

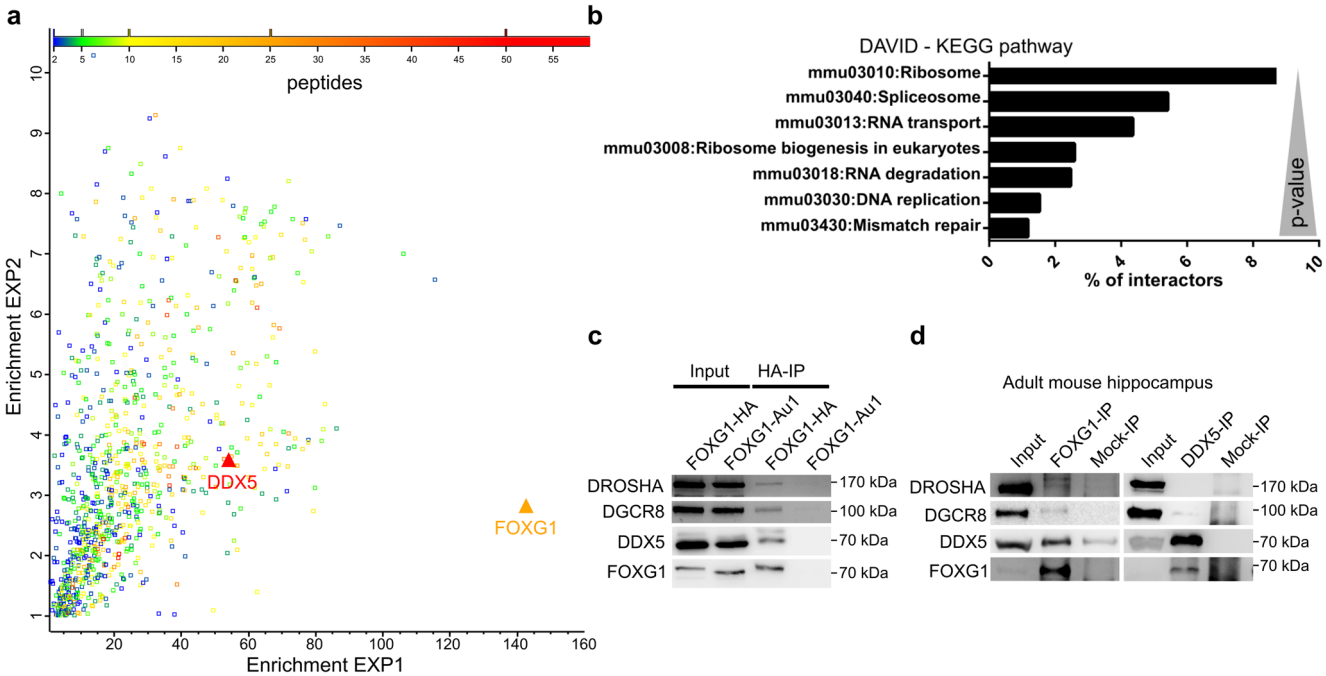


Fig. 5 Mass spectrometry and Co-IP reveal interaction of FOXG1 with DDX5 and the microprocessor. **a** Scatterplot of the FOXG1 interactome highlights its interaction with DDX5. The fold enrichment of two independent FOXG1-Au1 co-IP replicates of the MS analysis are plotted on the x- and y-axis, respectively, and detected proteins are colour coded with regard to the number of individual peptides mapping into the identified protein. $n = 2$. **b** DAVID KEGG pathway analysis with the most significant pathways. p values < 0.049. **c** Representative immunoblot after FOXG1-HA co-IP using anti-HA, anti-DDX5, anti-DROSHA and anti-DGCR8 shows that DDX5 and the microprocessor proteins interact with FOXG1 in N2a cells. Overexpression of FOXG1-Au1 serves as control for specificity of the HA-co-IP. **d** Representative immunoblot using anti-FOXG1, anti-DDX5, anti-DROSHA and anti-DGCR8 antibodies after co-IP of endogenous FOXG1 and DDX5, respectively, from protein extracts of adult mouse hippocampus. $n = 3$

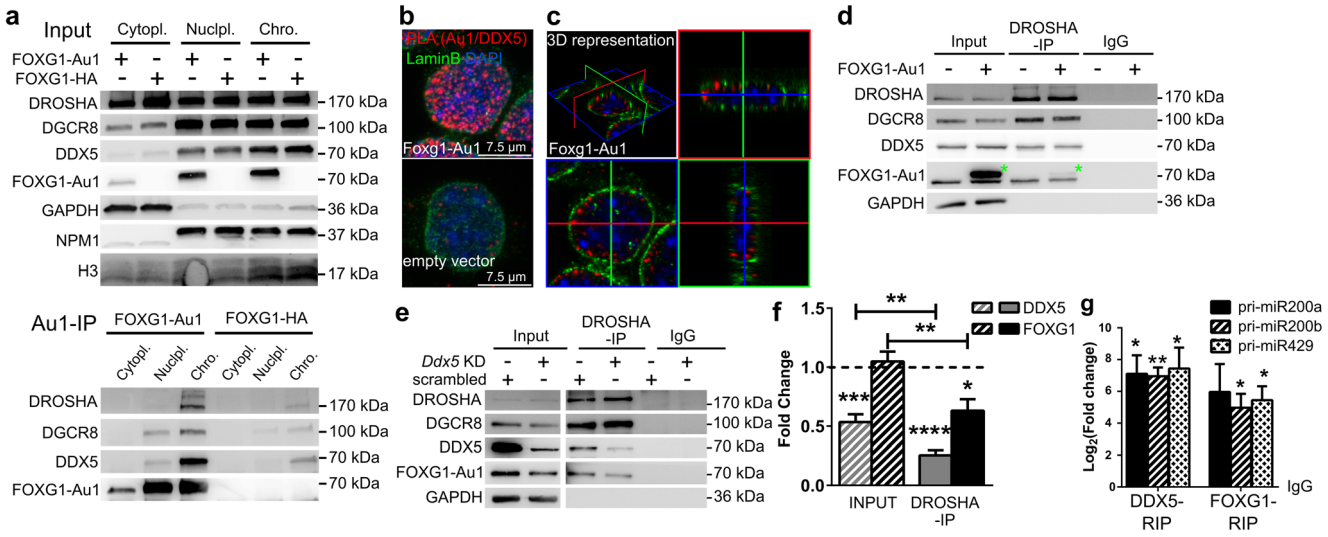


Fig. 6 FOXG1 and DDX5 interact in the nucleus and DDX5 recruits FOXG1 to DROSHA. **a** Immunoblots of cytoplasmic (cytopl), nucleoplasmic (nuclpl) and chromatin (chro) fractions after cell fractionation of FOXG1-Au1 or FOXG1-HA expressing N2a cells using anti-Au1, anti-DDX5 and microprocessor antibodies (upper panel, input samples). GAPDH, NPM1 and H3 are used as controls for fractionation. Au1-co-IP of FOXG1-Au1 or FOXG1-HA expressing N2a cells and immunoblots with anti-Au1, anti-DDX5 and microprocessor proteins showing that interactions take place in the nucleoplasm and chromatin fraction (lower panel, Au1-IP samples). $n = 3$. b Confocal imaging of PLA of FOXG1-Au1 and DDX5 after FOXG1-Au1 overexpression in N2a cells shows that FOXG1/DDX5 localises near Lamin B-positive immunostaining inside the nucleus. Scale bar 7.5 µm. $n = 2$. **c** 3D representation of **b**. **d** Immunoblot after DROSHA co-IP of FOXG1-Au1 overexpressing or empty vector-transfected N2a cells using anti-Au1, anti-DDX5 and anti-microprocessor antibodies. DDX5 co-precipitates with DROSHA in the presence and absence of FOXG1. Green asterisks indicate FOXG1-Au1 band. $n = 2$. **e** Immunoblot of DROSHA co-IP after *Ddx5* KD or scrambled control transfection in FOXG1 overexpressing N2a cells. Antibodies as in **d**. DROSHA co-precipitates less FOXG1 in conditions of decreased DDX5 expression. $n = 6$. **f** Densitometric analysis of **e**. Input FOXG1 and DDX5 are normalised to GAPDH, and in DROSHA-IP, FOXG1 and DDX5 are normalised to DROSHA (dashed line). Ratios of *Ddx5* KD to scrambled control are represented. Additional comparison of FOXG1 and DDX5 levels between DROSHA-IP and input revealed statistical significant reduction of FOXG1 and DDX5 after *Ddx5* KD (represented by the straight horizontal line bars). Mean with SEM, $*p < 0.05$, $**p < 0.01$ $***p < 0.001$, $****p < 0.0001$, one-sample Student's *t* test. $n = 6$. **g** qRT-PCR analyses of pri-miR200 family members after native DDX5 and FOXG1-Au1 RIP showing that FOXG1 and DDX5 co-precipitate pri-miR200 transcripts normalised to IgG control. Mean with SEM, $*p < 0.05$, $**p < 0.01$, unpaired Student's *t* test. $n = 3$

We next investigated if FOXG1 mediated the binding of DDX5 to the microprocessor in neural cells, or vice versa. We performed DROSHA co-IP after overexpressing FOXG1 in N2a cells and assessed DDX5 levels. DROSHA co-precipitated similar levels of DDX5, irrespective of the presence of FOXG1 (Fig. 6d). Next, we investigated if knock-down (KD) of DDX5 affected FOXG1 binding to DROSHA. By reducing expression of DDX5, we observed decreased FOXG1 recruitment to DROSHA (Fig. 6e, f). Together, these data suggested that DDX5 associated to the microprocessor independent of the presence of FOXG1, but that FOXG1 required DDX5 to bind to DROSHA.

FOXG1 did not seem to alter pri-miR200b/a/429 transcription and it associated with the microprocessor in the nucleus. We therefore hypothesised that a FOXG1–DDX5 complex might affect miR200 biogenesis and assessed whether FOXG1 or DDX5 would bind the pri-miR200b/a/429 transcript. We applied native RIP with antibodies against the Au1-tag after FOXG1-Au1 overexpression in N2a cells, or with anti-DDX5, respectively. RIP was followed by qRT-PCR, which revealed that both FOXG1 and DDX5 precipitated the pri-miR200b/a/429 (Fig. 6g).

Decreased Levels of DDX5 in FOXG1 Overexpressing Cells Reduce Mature miR200 Levels but Increase DROSHA Processivity

We next aimed to elucidate the molecular level at which FOXG1 and DDX5 affected miR200 biogenesis. First, we assessed whether overexpression of FOXG1 alone would be sufficient to affect miR200 biogenesis. Using pCX-miR200b/a/429 plasmid to express miR200b/a/429 family in N2a cells did not increase the expression of the primary transcript after FOXG1 overexpression (Fig. 7a), confirming that FOXG1 did not influence transcription of the parental gene. We next determined the levels of pre-miR200 and mature miR200 after FOXG1 and miR200b/a/429 family overexpression using Northern blots. Overexpression of FOXG1 in these conditions did neither affect precursor nor mature miR200 levels (Fig. 7b, c).

We next used a pmiRGLO-miR200b/a/429 plasmid to overexpress the miR200b/a/429 family and assessed whether levels of primary miR200 transcripts changed after FOXG1 overexpression. In these conditions, the presence of FOXG1 significantly increased pri-miR200b/

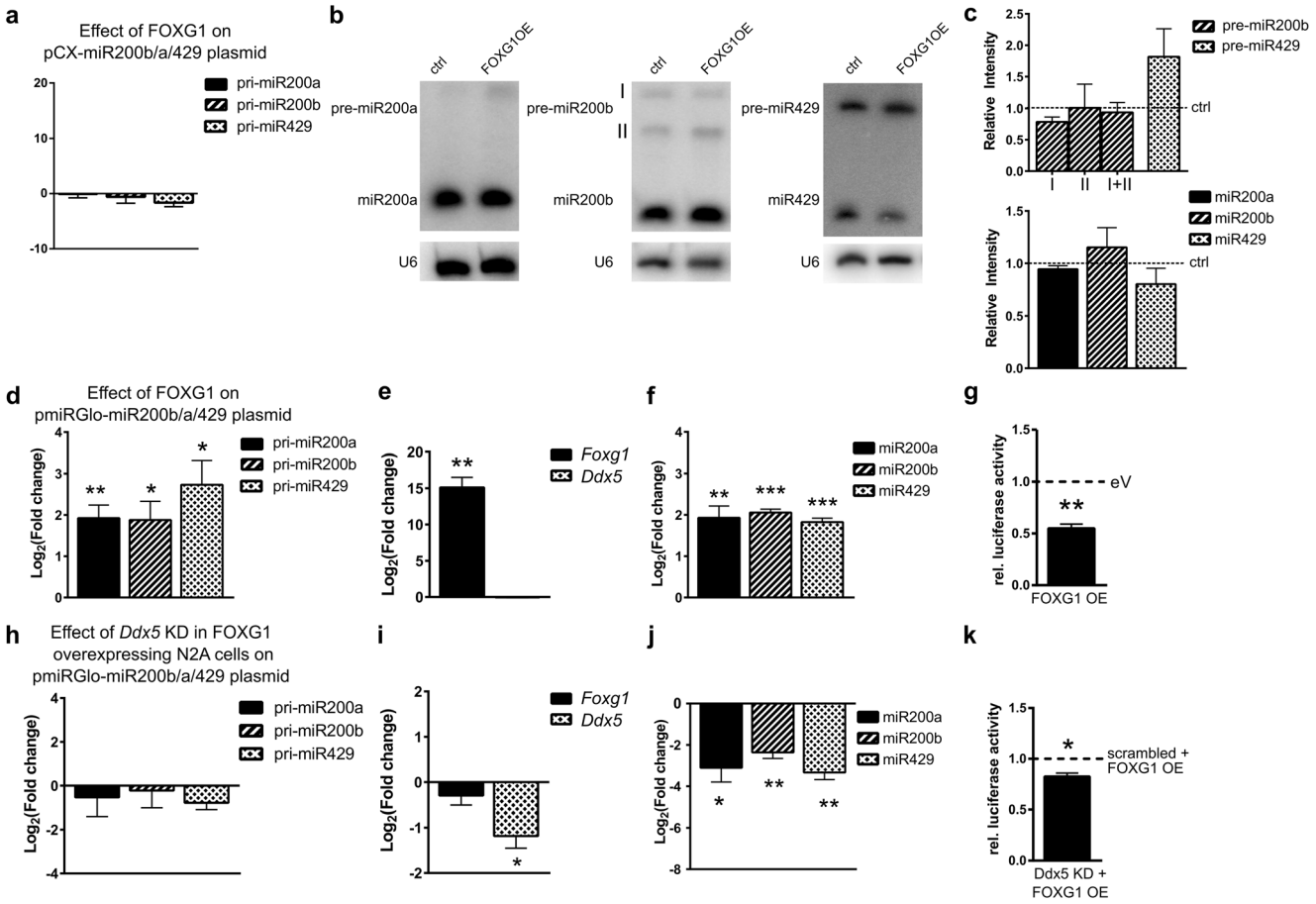


Fig. 7 FOXG1 overexpression affects miR200 family levels in conditions of increased expression of pri-miR200 in a DDX5-dependent manner. **a** qRT-PCRs of pri-miR200b/a/429 of N2a cells overexpressing FOXG1 and miR200 family using pCX-miR200b/a/429 compared to an empty vector. One-sample Student's *t* test. $n = 3$. **b** Representative northern blots of precursor (pre-) and mature miR200b/a/429 in N2a cells using pCX-miR200b/a/429. Shown are control-transfected and FOXG1 overexpression in N2a cells probed for *pre/miR200a* (upper panel), *pre-miR200b* (middle panel) and pre-miR429 (lower panel) as well as U6 loading control. **c** Densitometric quantification of the northern blot bands for precursors of miR200b (isoforms I and II) and miR429 (left panels) and mature miR200b/a/429 (right panels). Dashed lines indicate expression levels of control cells. FOXG1 overexpression does not alter expression levels of precursor and mature miR200 family members. Mean with SEM, $^{**}p < 0.01$, one-sample Student's *t* test. $n = 3$. **d** qRT-PCRs of pri-miR200b/a/429 of N2a cells overexpressing miR200 family using pmiRGlo-miR200b/a/429 and FOXG1 compared to empty vector. pri-miR200 levels increase upon FOXG1 overexpression. Mean with SEM, $^{*}p < 0.05$, $^{**}p < 0.01$, one-sample Student's *t* test. $n = 3$–6. **e** qRT-PCR results confirm increased *Foxg1* and unchanged *Ddx5* expression. Mean with SEM, $^{**}p < 0.01$, one-sample Student's *t* test. $n = 4$. **f** qRT-PCRs of mature miR200b/a/429 of N2a cells overexpressing miR200 family from pmiRGlo-miR200b/a/429 plasmid and FOXG1 compared to empty vector expression. Levels of mature miR200 increase statistically significant. Mean with SEM, $^{**}p < 0.01$, $^{***}p < 0.001$, one-sample Student's *t* test. $n = 5$–6. **g** Reduced luciferase activity indicates an increased turnover of pri-miR200b/a/429 in the presence of FOXG1. Mean with SEM, $^{**}p < 0.01$, one-sample Student's *t* test. $n = 4$. **h** qRT-PCRs of pri-miR200b/a/429 of N2a cells overexpressing miR200 family using pmiRGlo-miR200b/a/429 and FOXG1 as well as simultaneous *Ddx5* KD, compared to FOXG1 overexpression and scrambled control. pri-miR200 levels are unaffected by *Ddx5* KD. Mean with SEM, one-sample Student's *t* test. $n = 3$–6. *i* qRT-PCR results confirm decreased *Ddx5* levels after *Ddx5* KD and unchanged *Foxg1* expression. Mean with SEM, $^{*}p < 0.05$, one-sample Student's *t* test. $n = 4$. **j** qRT-PCRs of mature miR200b/a/429 in N2a cells after overexpressing miR200 family from pmiRGlo-miR200b/a/429 plasmid and FOXG1 as well as simultaneous *Ddx5* KD compared to FOXG1 overexpression and scrambled control. Levels of mature miR200 decrease significantly after reduction of DDX5 in FOXG1 expressing cells. Mean with SEM, $^{*}p < 0.05$, $^{**}p < 0.01$, one-sample Student's *t* test. $n = 5$–6. **k** Reduced luciferase activity indicates an increased turnover of pri-miR200b/a/429 after *Ddx5* KD and FOXG1 overexpression when compared to cells overexpressing FOXG1 and scrambled control. Mean with SEM, $^{*}p < 0.05$, one-sample Student's *t* test. $n = 4$

a/429 levels (Fig. 7d, e), which resulted in significantly increased levels of mature miR200 family expression (Fig. 7f). Moreover, the N2a cells transfected with pmiRGlo-miR200b/a/429 along with FOXG1 allowed us to observe an increased turnover of the pri-miR200b/a/429 in a luciferase reporter assay (Fig. 7g). As we observed altered mature miR200 levels only when we used pmiRGLO-miR200b/a/429, which led to increased levels of the primary transcript, but not with the pCX-miR200b/a/429 plasmid that did not change pri-miR200 levels, we concluded that FOXG1 alone is probably not sufficient to affect miR200 biogenesis. Instead,

significantly altered levels of the primary transcript are necessary to observe altered expression of mature miR200 family members.

To further address whether DDX5 would play a role in the FOXG1-mediated increase in mature miR200 levels as observed in Fig. 7f, we knocked-down (KD) *Ddx5* in FOXG1 and pri-miR200b/a/429 (using pmiRGLO-miR200b/a/429) expressing cells (Fig. 7i). Under these conditions, pri-miR200b/a/429 levels did not change (Fig. 7h); however, the levels of the mature miR200 family decreased significantly (Fig. 7j), despite increased activity of the microprocessor (Fig. 7k). These findings indicated that DDX5 is involved in FOXG1-mediated processing of the miR200 family members.

As the experimental conditions in N2a cells required overexpression of FOXG1 and increased levels of pri-miR200b/a/429, we aimed to study the influence of FOXG1 and DDX5 in primary hippocampal neurons. KD of FOXG1 resulted in increased levels of DDX5, without significant effect on miR200 maturation (Fig. 8a, b). In contrast, KD of DDX5 resulted in concomitant increase of FOXG1 expression (Fig. 8c), as well as significantly decreased expression of miR200b (Fig. 8d). This result mimicked and confirmed our observation in N2a cells, in which decreased levels of DDX5 and concomitant increased levels of FOXG1 affected miR200 maturation. Taken together, these data showed that FOXG1 and DDX5 were involved in miR200 biogenesis and that the underlying molecular mechanism depended on a crucial balance between the levels of FOXG1, DDX5 and pri-miR200b/a/429.

Discussion

In this study, we identified that FOXG1 affects PRKAR2B expression at multiple levels, on one hand through direct transcriptional repression and on the other hand through the miR200 family that directly targets *Prkar2b* transcripts. Thereby, we discovered important new roles of FOXG1 for the function of neural cells and of hippocampal neurons in the adult CNS, independent on proliferative effects (Fig. S2). Patients with mutations in *FOXG1* show impaired neuronal function and the symptoms are comparable to Rett syndrome. However, the role of FOXG1 in mature neurons and in the adult CNS is not entirely clear. In the adult hippocampus, FOXG1 prevents depletion of the progenitor pool in the dentate gyrus [38]. In cerebellar granule neurons, FOXG1 promotes survival [39]. Also, FOXG1 binds to a spliced, proapoptotic version of MECP2 thereby preventing neuronal death [13].

Apart from influencing gene transcription, Pancrazi et al. showed that FOXG1 localised within the mitochondrial matrix and regulates the mitochondrial membrane potential, mitochondrial fission and mitosis [40]. This finding gave indication that FOXG1 might have additional functions than regulation of transcription, for example in posttranscriptional control. Here, we describe a novel interaction between FOXG1, DDX5 and the microprocessor complex that can affect maturation of miR200b/a/429. Our data show that the PKA regulator PRKAR2B is a direct target of FOXG1 as well as of miR200 family. The biogenesis of miR200 can also be influenced by FOXG1, probably in a context-dependent manner that depends on the levels of the different players. Our data robustly show that increased levels of FOXG1 in *Ddx5* KD N2a and hippocampal cells decreased mature miR200. However, at least in N2a cells, we observed decreased miR200 levels despite a significantly increased turnover of the primary transcript. Therefore, we hypothesise further regulative layers downstream of DROSHA that might be affected by FOXG1. Our MS data propose different pathways that might act downstream of DROSHA, as the data suggest that FOXG1 associates for example with the exosome and therefore confers degradation of pre-miR200 family members. It is also possible that FOXG1 interacts with nuclear envelope proteins and affects the transport of pre-miR200 into the cytoplasm. The PLA signal suggests that FOXG1/DDX5 complexes localise near the membrane, which might corroborate such interpretation. However, to shed more light on FOXG1's implication in posttranscriptional control and FOXG1/DDX5-mediated effects downstream of DROSHA, much more research is needed.

Our finding that FOXG1 influences posttranscriptional maturation of miRNAs reflects a shared cellular function between FOXG1 and MECP2, as the latter also associates to DROSHA and DGCR8 to regulate miRNA processing in the adult mouse hippocampus [41]. However, while the involvement of FOXG1 and MECP2 in miRNA biogenesis is conserved, targets are different. MECP2 suppresses miR134, miR383, miR382 and miR182 maturation in the hippocampus [41]. Our study revealed fewer numbers of differently expressed miRNAs that are affected in *Foxg1*$^{cre/+}$ hippocampus compared to the data reported for MECP2 deficiency. And, we identified no commonly misregulated miRNAs between the two forms of RTT. Thus, although both MECP2 and FOXG1 associate with the microprocessor complex, they seemingly affect maturation of different miRNAs.

miR200b/a/429 transcription is regulated by different factors in a tissue-specific manner. TP53 is a transcription factor necessary for miR200b/a/429 gene expression [42, 43], whereas ZEB1 and ZEB2 repress expression of the miR200b/a/429 gene. Ovarian tumours induce miR200 expression upon DNA damage involving another RNA helicase, namely DDX1 [44]. Control of expression and biogenesis of the miR200 is necessary as this family is implicated in diverse cellular processes, ranging from neurodegeneration, eye development, adipocyte differentiation, taste bud and tooth development to maintenance of stem cell identity (reviewed in [45]). The miR200 family has important regulative functions

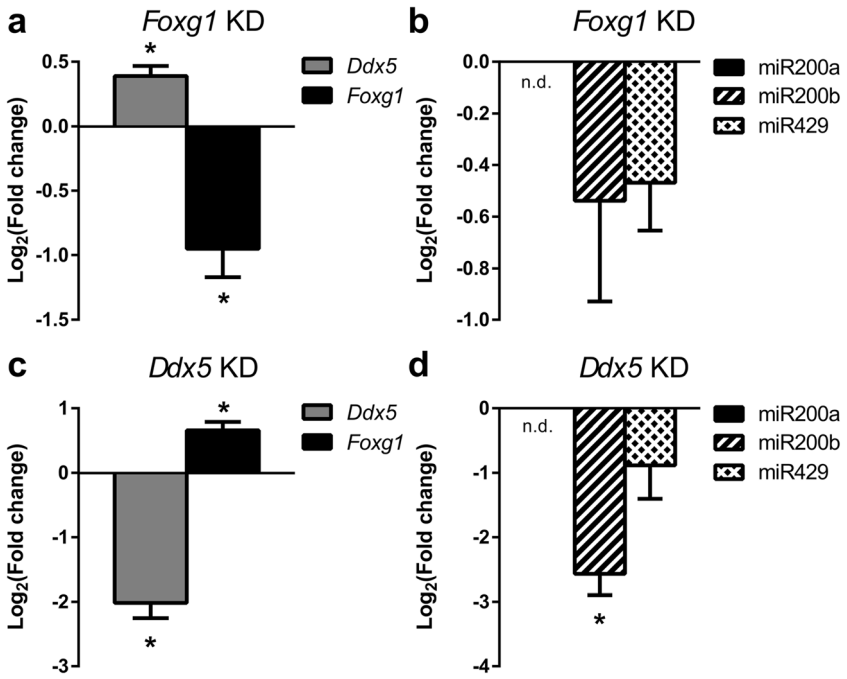


Fig. 8 Increased levels of *Foxg1* and *Ddx5* KD decrease miR200b expression in primary hippocampal neurons. **a** qRT-PCR of *Ddx5* and *Foxg1* in *Foxg1* KD primary hippocampal cells, *Foxg1* KD increases expression of *Ddx5*. **b** qRT-PCR of mature miR200 family expression after KD of *Foxg1*. Expression levels of mature miRNA200 family do not change. **c** qRT-PCR of *Ddx5* and *Foxg1* in *Ddx5* KD primary hippocampal cells. *Ddx5* KD results in increased expression of *Foxg1*. **d** qRT-PCR of mature miR200 family expression after KD of *Ddx5*. Reduced expression of *Ddx5* with concomitant increase in *Foxg1* expression causes significant reduction of mature miR200b expression in primary hippocampal neurons. All data are represented as mean with SEM, $*p < 0.05$, $**p < 0.01$, one-sample Student's *t* test. $n = 3$

in cancer [46] and neural differentiation in humans [28], zebrafish [26], *Drosophila* [47] and neural PC12 cells [27], as well as in mice [24, 25]. Altogether, miR200 controls similar processes as FOXG1 in the developing cerebral cortex [5, 9, 22, 23]. Interestingly, *Foxg1* has been proposed as miR200 target in different model systems [26, 48, 49], and varying FOXG1 expression levels might thus be regulated through a miR200-dependent feedback loop. Such feedback loop might account for the transiently reduced FOXG1 expression in differentiating progenitors in the cerebral cortex and the reinitiation of its expression in differentiated projection neurons [6]. It is therefore tempting to speculate further that the biogenesis of miR200, influenced by FOXG1, is not restricted to the adult hippocampus but also takes place in the developing cerebral cortex. This notion is supported by our identification of FOXG1–DDX5 interaction in E13.5 embryonic cortical tissue (data not shown).

Another important finding of our study is that FOXG1 and miR200 target PRKAR2B in the CNS. PRKAR2B is the target of miR200b in platelets [50]. PRKAR2B is a regulative type 2 subunit of protein kinase A (PKA), which is expressed in different tissues but has the highest expression in the CNS [51]. PKA signalling is critically implicated in memory formation, and this function is evolutionary highly conserved (reviewed by [52]). For example, PKA regulates synaptic plasticity by phosphorylating AMPA receptor subunits [53], or the GluN2B subunit of NMDAR during emotional response to stress [54]. Interference with PKA signalling in the mouse hippocampus impairs long-term spatial memory formation and elicits long-term memory deficits in contextual fear conditioning [55]. In the latter, freezing behaviour reduces after interference with PKA signalling. $Foxg1^{cre/+}$ mice show a similar behaviour of decreased contextual fear response [56]. Thus, increasing levels of the repressive PRKAR2B subunit after decreased expression of miR200 family members might interfere with effective PKA signalling in $Foxg1^{cre/+}$ mice. Strikingly, MECP2-deficient mice show the same decreased freezing behaviour [57], and increased levels of PRKAR2B (data not shown and [58]). Published data from iPS-derived neurons show that *PRKAR2B* expression levels are altered in some FOXG1 syndrome patients (log_2FC of *PRKAR2B* of 1.14 with a FDR of 8E-04 [59]). In all, impaired PKA signalling might be a common feature in tRTT and atRTT. Altered levels of PRKAR2B might also be responsible for other phenotypic alterations in Rett syndrome, such as altered motor behaviour [60], or impaired vision [61–64].

In summary, our data indicate that FOXG1 associates to the microprocessor complex with DDX5 and that it affects mature miR200 levels. We further suggest that FOXG1 and miR200 family are both part of a multilevel network that balances the expression of a regulative subunit of PKA, PRKAR2B.

Thereby, FOXG1 and also MECP2 may affect a common PKA-dependent pathway to adjust neuronal function in the hippocampus.

Acknowledgements The authors thank Harold Cremer (IBDM, Aix-Marseille Université, France) for pCX-miRNA200 family and pCX-D2eGFP-miR200 sponge plasmids and Nicole Gensch (BIOSS tool box, University of Freiburg) for further plasmids used in this study; Michael Müller (University Medicine Göttingen, Germany) for providing *Mecp2*$^{-/y}$ mice hippocampal tissue; Ute Baur and Monika Pätzold for technical support; and Marco Ell for his experiments during the early phase of the project.

Funding Information This work was funded by the Deutsche Forschungsgemeinschaft DFG SPP1738 through grants to T. Vogel, R. Backofen and A. Fischer, by the DFG Research Training Group GRK2344 (to T. Vogel, R. Backofen, and W.R. Hess) and by the Collaborative Research Centre 992 Medical Epigenetics (T. Vogel and R. Backofen) and the German Federal Ministry of Education and Research (BMBF) through grants to R. Backofen (031A538A de. NBI) and W.R. Hess (031L0106B de. NBI Partner).

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Bienvenu T, Chelly J (2006) Molecular genetics of Rett syndrome: when DNA methylation goes unrecognized. Nat Rev Genet 7:415–426. https://doi.org/10.1038/nrg1878
2. Jacob FD, Ramaswamy V, Andersen J, Bolduc FV (2009) Atypical Rett syndrome with selective FOXG1 deletion detected by comparative genomic hybridization: case report and review of literature. Eur J Hum Genet 17:1577–1581. https://doi.org/10.1038/ejhg.2009.95
3. del Gaudio D, Fang P, Scaglia F, Ward PA, Craigen WJ, Glaze DG, Neul JL, Patel A et al (2006) Increased MECP2 gene copy number as the result of genomic duplication in neurodevelopmentally delayed males. Genet Med 8:784–792
4. Brunetti-Pierri N, Paciorkowski AR, Ciccone R, Mina ED, Bonaglia MC, Borgatti R, Schaaf CP, Sutton VR et al (2011) Duplications of *FOXG1* in 14q12 are associated with developmental epilepsy, mental retardation, and severe speech impairment. Eur J Hum Genet 19:102–107. https://doi.org/10.1038/ejhg.2010.142
5. Xuan S, Baptista CA, Balas G, Tao W, Soares VC, Lai E (1995) Winged helix transcription factor BF-1 is essential for the development of the cerebral hemispheres. Neuron 14:1141–1152. https://doi.org/10.1016/0896-6273(95)90262-7
6. Miyoshi G, Fishell G (2012) Dynamic FoxG1 expression coordinates the integration of multipolar pyramidal neuron precursors into the cortical plate. Neuron 74:1045–1058. https://doi.org/10.1016/j.neuron.2012.04.025
7. Hanashima C, Li SC, Shen L et al (2004) Foxg1 suppresses early cortical cell fate. Science 303:56–59. https://doi.org/10.1126/science.1090674
8. Brancaccio M, Pivetta C, Granzotto M, Filippis C, Mallamaci A (2010) Emx2 and Foxg1 inhibit gliogenesis and promote neuronogenesis. Stem Cells 28(7):1206–1218. https://doi.org/10.1002/stem.443
9. Seoane J, Le H-V, Shen L et al (2004) Integration of Smad and forkhead pathways in the control of neuroepithelial and glioblastoma cell proliferation. Cell 117:211–223
10. Siegenthaler JA, Tremper-Wells BA, Miller MW (2008) Foxg1 haploinsufficiency reduces the population of cortical intermediate progenitor cells: effect of increased p21 expression. Cereb Cortex 18:1865–1875. https://doi.org/10.1093/cercor/bhm209
11. Vezzali R, Weise SC, Hellbach N, Machado V, Heidrich S, Vogel T (2016) The FOXG1/FOXO/SMAD network balances proliferation and differentiation of cortical progenitors and activates Kcnh3 expression in mature neurons. Oncotarget 7:37436–37455
12. Manuel MN, Martynoga B, Molinek MD, Quinn JC, Kroemmer C, Mason JO, Price DJ (2011) The transcription factor Foxg1 regulates telencephalic progenitor proliferation cell autonomously, in part by controlling Pax6 expression levels. Neural Dev 6:9. https://doi.org/10.1186/1749-8104-6-9
13. Dastidar SG, Narayanan S, Stifani S, D'Mello SR (2012) Transducin-like enhancer of Split-1 (TLE1) combines with Forkhead box protein G1 (FoxG1) to promote neuronal survival. J Biol Chem 287:14749–14759. https://doi.org/10.1074/jbc.M111.328336
14. Urdinguio RG, Fernández AF, Lopez-Nieva P, Rossi S, Huertas D, Kulis M, Liu CG, Croce CM et al (2010) Disrupted microRNA expression caused by Mecp2 loss in a mouse model of Rett syndrome. Epigenetics 5:656–663. https://doi.org/10.4161/epi.5.7.13055
15. Wu H, Tao J, Chen PJ, Shahab A, Ge W, Hart RP, Ruan X, Ruan Y et al (2010) Genome-wide analysis reveals methyl-CpG–binding protein 2–dependent regulation of microRNAs in a mouse model of Rett syndrome. PNAS 107:18161–18166. https://doi.org/10.1073/pnas.1005595107
16. Petazzi P, Sandoval J, Szczesna K, Jorge OC, Roa L, Sayols S, Gomez A, Huertas D et al (2013) Dysregulation of the long noncoding RNA transcriptome in a Rett syndrome mouse model. RNA Biol 10:1197–1203. https://doi.org/10.4161/rna.24286
17. Hébert JM, McConnell SK (2000) Targeting of cre to the Foxg1 (BF-1) locus mediates loxP recombination in the telencephalon and other developing head structures. Dev Biol 222:296–306. https://doi.org/10.1006/dbio.2000.9732
18. Tyanova S, Temu T, Sinitcyn P, Carlson A, Hein MY, Geiger T, Mann M, Cox J (2016) The Perseus computational platform for comprehensive analysis of (prote) omics data. Nat Meth 13:731–740. https://doi.org/10.1038/nmeth.3901
19. Afgan E, Baker D, van den Beek M, Blankenberg D, Bouvier D, Čech M, Chilton J, Clements D et al (2016) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. Nucleic Acids Res 44:W3–W10. https://doi.org/10.1093/nar/gkw343
20. Grüning BA, Fallmann J, Yusuf D, Will S, Erxleben A, Eggenhofer F, Houwaart T, Batut B et al (2017) The RNA workbench: best practices for RNA and high-throughput sequencing bioinformatics in Galaxy. Nucleic Acids Res 45:W560–W566. https://doi.org/10.1093/nar/gkx409
21. Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, Preibisch S, Rueden C et al (2012) Fiji: an open-source

platform for biological-image analysis. Nat Methods 9:676–682. https://doi.org/10.1038/nmeth.2019
22. Dou C, Lee J, Liu B, Liu F, Massague J, Xuan S, Lai E (2000) BF-1 interferes with transforming growth factor beta signaling by associating with Smad partners. Mol Cell Biol 20:6201–6211
23. Hanashima C, Shen L, Li SC, Lai E (2002) Brain factor-1 controls the proliferation and differentiation of neocortical progenitor cells through independent mechanisms. J Neurosci 22:6526–6536
24. Beclin C, Follert P, Stappers E, Barral S, Coré N, de Chevigny A, Magnone V, Lebrigand K et al (2016) miR-200 family controls late steps of postnatal forebrain neurogenesis via Zeb2 inhibition. Sci Rep 6:35729. https://doi.org/10.1038/srep35729
25. Peng C, Li N, Ng Y-K, Zhang J, Meier F, Theis FJ, Merkenschlager M, Chen W et al (2012) A unilateral negative feedback loop between miR-200 microRNAs and Sox2/E2F3 controls neural progenitor cell-cycle exit and differentiation. J Neurosci 32:13292–13308. https://doi.org/10.1523/JNEUROSCI.2124-12.2012
26. Choi PS, Zakhary L, Choi W-Y, Caron S, Alvarez-Saavedra E, Miska EA, McManus M, Harfe B et al (2008) Members of the miRNA-200 family regulate olfactory neurogenesis. Neuron 57:41–55. https://doi.org/10.1016/j.neuron.2007.11.018
27. Pandey A, Singh P, Jauhari A, Singh T, Khan F, Pant AB, Parmar D, Yadav S (2015) Critical role of the miR-200 family in regulating differentiation and proliferation of neurons. J Neurochem 133:640–652. https://doi.org/10.1111/jnc.13089
28. Du Z-W, Ma L-X, Phillips C, Zhang S-C (2013) miR-200 and miR-96 families repress neural induction from human embryonic stem cells. Development 140:2611–2618. https://doi.org/10.1242/dev.092809
29. Liu H, Liang C, Kollipara RK, Matsui M, Ke X, Jeong BC, Wang Z, Yoo KS et al (2016) HP1BP3, a chromatin retention factor for co-transcriptional microRNA processing. Mol Cell 63:420–432. https://doi.org/10.1016/j.molcel.2016.06.014
30. Bourgeois CF, Mortreux F, Auboeuf D (2016) The multiple functions of RNA helicases as drivers and regulators of gene expression. Nat Rev Mol Cell Biol 17:426–438. https://doi.org/10.1038/nrm.2016.50
31. Ehrhart F, Coort SLM, Cirillo E, Smeets E, Evelo CT, Curfs LMG (2016) Rett syndrome—biological pathways leading from MECP2 to disorder phenotypes. Orphanet Journal of Rare Diseases 11(158):158. https://doi.org/10.1186/s13023-016-0545-5
32. Smith RM, Sadee W (2011) Synaptic signaling and aberrant RNA splicing in autism spectrum disorders. Front Synaptic Neurosci 3. https://doi.org/10.3389/fnsyn.2011.00001
33. Denli AM, Tops BBJ, Plasterk RHA, Ketting RF, Hannon GJ (2004) Processing of primary microRNAs by the microprocessor complex. Nature 432:231–235. https://doi.org/10.1038/nature03049
34. Hong S, Noh H, Chen H, Padia R, Pan ZK, Su SB, Jing Q, Ding HF et al (2013) Signaling by p38 MAPK stimulates nuclear localization of the microprocessor component p68 for processing of selected primary microRNAs. Sci Signal 6:ra16–ra16. https://doi.org/10.1126/scisignal.2003706
35. Tang X, Zhang Y, Tucker L, Ramratnam B (2010) Phosphorylation of the RNase III enzyme Drosha at Serine300 or Serine302 is required for its nuclear localization. Nucleic Acids Res 38:6610–6619. https://doi.org/10.1093/nar/gkq547
36. Yang Q, Li W, She H, Dou J, Duong DM, du Y, Yang SH, Seyfried NT et al (2015) Stress induces p38 MAPK-mediated phosphorylation and inhibition of Drosha-dependent cell survival. Mol Cell 57:721–734. https://doi.org/10.1016/j.molcel.2015.01.004
37. Link S, Grund SE, Diederichs S (2016) Alternative splicing affects the subcellular localization of Drosha. Nucleic Acids Res 44:5330–5343. https://doi.org/10.1093/nar/gkw400
38. Tian C, Gong Y, Yang Y, Shen W, Wang K, Liu J, Xu B, Zhao J et al (2012) Foxg1 has an essential role in postnatal development of the dentate gyrus. J Neurosci 32:2931–2949. https://doi.org/10.1523/JNEUROSCI.5240-11.2012
39. Dastidar SG, Landrieu PMZ, D'Mello SR (2011) FoxG1 promotes the survival of postmitotic neurons. J Neurosci 31:402–413. https://doi.org/10.1523/JNEUROSCI.2897-10.2011
40. Pancrazi L, Benedetto GD, Colombaioni L et al (2015) Foxg1 localizes to mitochondria and coordinates cell differentiation and bioenergetics. PNAS 112:13910–13915. https://doi.org/10.1073/pnas.1515190112
41. Cheng T-L, Wang Z, Liao Q, Zhu Y, Zhou WH, Xu W, Qiu Z (2014) MeCP2 suppresses nuclear microRNA processing and dendritic growth by regulating the DGCR8/Drosha complex. Dev Cell 28:547–560. https://doi.org/10.1016/j.devcel.2014.01.032
42. Kim T, Veronese A, Pichiorri F, Lee TJ, Jeon YJ, Volinia S, Pineau P, Marchio A et al (2011) p53 regulates epithelial–mesenchymal transition through microRNAs targeting ZEB1 and ZEB2. J Exp Med 208:875–883. https://doi.org/10.1084/jem.20110235
43. Kolesnikoff N, Attema JL, Roslan S et al (2014) Specificity protein 1 (Sp1) maintains basal epithelial expression of the miR-200 family: implications for epithelial-mesenchymal transition. J Biol Chem 289:11194–11205. https://doi.org/10.1074/jbc.M113.529172
44. Han C, Liu Y, Wan G, Choi HJ, Zhao L, Ivan C, He X, Sood AK et al (2014) The RNA-binding protein DDX1 promotes primary microRNA maturation and inhibits ovarian tumor progression. Cell Rep 8:1447–1460. https://doi.org/10.1016/j.celrep.2014.07.058
45. Trümbach D, Prakash N (2015) The conserved miR-8/miR-200 microRNA family and their role in invertebrate and vertebrate neurogenesis. Cell Tissue Res 359:161–177. https://doi.org/10.1007/s00441-014-1911-z
46. Brabletz S, Bajdak K, Meidhof S, Burk U, Niedermann G, Firat E, Wellner U, Dimmler A et al (2011) The ZEB1/miR-200 feedback loop controls Notch signalling in cancer cells. EMBO J 30:770–782. https://doi.org/10.1038/emboj.2010.349
47. Morante J, Vallejo DM, Desplan C, Dominguez M (2013) Conserved miR-8/miR-200 defines a glial niche that controls neuroepithelial expansion and neuroblast transition. Dev Cell 27:174–187. https://doi.org/10.1016/j.devcel.2013.09.018
48. Garaffo G, Conte D, Provero P, Tomaiuolo D, Luo Z, Pinciroli P, Peano C, D'Atri I et al (2015) The Dlx5 and Foxg1 transcription factors, linked via miRNA-9 and -200, are required for the development of the olfactory and GnRH system. Mol Cell Neurosci 68:103–119. https://doi.org/10.1016/j.mcn.2015.04.007
49. Zeng F, Xue M, Xiao T, Li Y, Xiao S, Jiang B, Ren C (2016) MiR-200b promotes the cell proliferation and metastasis of cervical cancer by inhibiting FOXG1. Biomed Pharmacother 79:294–301. https://doi.org/10.1016/j.biopha.2016.02.033
50. Nagalla S, Shaw C, Kong X, Kondkar AA, Edelstein LC, Ma L, Chen J, McKnight GS et al (2011) Platelet microRNA-mRNA coexpression profiles correlate with platelet reactivity. Blood 117:5189–5197. https://doi.org/10.1182/blood-2010-09-299719
51. Skalhegg BS, Tasken K (2000) Specificity in the cAMP/PKA signaling pathway. Differential expression,regulation, and subcellular localization of subunits of PKA. Front Biosci 5:D678–D693
52. Kandel ER (2012) The molecular biology of memory: cAMP, PKA, CRE, CREB-1, CREB-2, and CPEB. Molecular Brain 5:14. https://doi.org/10.1186/1756-6606-5-14
53. Esteban JA, Shi S-H, Wilson C, Nuriya M, Huganir RL, Malinow R (2003) PKA phosphorylation of AMPA receptor subunits controls synaptic trafficking underlying plasticity. Nat Neurosci 6:136–143. https://doi.org/10.1038/nn997
54. Murphy JG, Sanderson JL, Gorski JA, Scott JD, Catterall WA, Sather WA, Dell'Acqua ML (2014) AKAP-anchored PKA maintains neuronal L-type calcium channel activity and NFAT

transcriptional signaling. Cell Rep 7:1577–1588. https://doi.org/10.1016/j.celrep.2014.04.027
55. Abel T, Nguyen PV, Barad M, Deuel TAS, Kandel ER, Bourtchouladze R (1997) Genetic demonstration of a role for PKA in the late phase of LTP and in hippocampus-based long-term memory. Cell 88:615–626. https://doi.org/10.1016/S0092-8674(00)81904-2
56. Shen L, Nam H-S, Song P, Moore H, Anderson SA (2006) FoxG1 haploinsufficiency results in impaired neurogenesis in the postnatal hippocampus and contextual memory deficits. Hippocampus 16:875–890. https://doi.org/10.1002/hipo.20218
57. Moretti P, Levenson JM, Battaglia F, Atkinson R, Teague R, Antalffy B, Armstrong D, Arancio O et al (2006) Learning and memory and synaptic plasticity are impaired in a mouse model of Rett syndrome. J Neurosci 26:319–327. https://doi.org/10.1523/JNEUROSCI.2623-05.2006
58. Balakrishnan S, Niebert M, Richter DW (2016) Rescue of cyclic AMP mediated long term potentiation impairment in the hippocampus of Mecp2 knockout (Mecp2−/y) mice by rolipram. Front Cell Neurosci 10. https://doi.org/10.3389/fncel.2016.00015
59. Mariani J, Coppola G, Zhang P, Abyzov A, Provini L, Tomasini L, Amenduni M, Szekely A et al (2015) FOXG1-dependent dysregulation of GABA/glutamate neuron differentiation in autism spectrum disorders. Cell 162:375–390. https://doi.org/10.1016/j.cell.2015.06.034
60. Brandon EP, Logue SF, Adams MR, Qi M, Sullivan SP, Matsumoto AM, Dorsa DM, Wehner JM et al (1998) Defective motor behavior and neural gene expression in RIIβ-protein kinase A mutant mice. J Neurosci 18:3639–3649
61. Fischer QS, Beaver CJ, Yang Y, Rao Y, Jakobsdottir KB, Storm DR, McKnight G, Daw NW (2004) Requirement for the RIIβ isoform of PKA, but not calcium-stimulated adenylyl cyclase, in visual cortical plasticity. J Neurosci 24:9049–9058. https://doi.org/10.1523/JNEUROSCI.2409-04.2004
62. Krishnan Y, Li Y, Zheng R, Kanda V, McDonald TV (2012) Mechanisms underlying the protein-kinase mediated regulation of the HERG potassium channel synthesis. Biochimica et Biophysica Acta (BBA) - Molecular Cell Research 1823:1273–1284. https://doi.org/10.1016/j.bbamcr.2012.05.012
63. LeBlanc JJ, DeGregorio G, Centofante E et al (2015) Visual evoked potentials detect cortical processing deficits in Rett syndrome. Ann Neurol 78:775–786. https://doi.org/10.1002/ana.24513
64. Boggio EM, Pancrazi L, Gennaro M, Lo Rizzo C, Mari F, Meloni I, Ariani F, Panighini A et al (2016) Visual impairment in FOXG1-mutated individuals and mice. Neuroscience 324:496–508. https://doi.org/10.1016/j.neuroscience.2016.03.027

Affiliations

Stefan C. Weise[1,2,3] · Ganeshkumar Arumugam[1,2,4] · Alejandro Villarreal[1] · Pavankumar Videm[5] · Stefanie Heidrich[1] · Nils Nebel[1] · Verónica I. Dumit[6] · Farahnaz Sananbenesi[7] · Viktoria Reimann[8] · Madeline Craske[9] · Oliver Schilling[10,11] · Wolfgang R. Hess[8,12] · Andre Fischer[7,13,14] · Rolf Backofen[5,6,10,15] · Tanja Vogel[1]

1 Institute of Anatomy and Cell Biology, Department of Molecular Embryology, Faculty of Medicine, Albert-Ludwigs-University Freiburg, 79104 Freiburg, Germany

2 Faculty of Biology, Albert-Ludwigs-University Freiburg, 79104 Freiburg, Germany

3 Institute of Anatomy and Cell Biology, Department of Neuroanatomy, Faculty of Medicine, Albert-Ludwigs-University Freiburg, 79104 Freiburg, Germany

4 Spemann Graduate School of Biology and Medicine, Albert-Ludwigs-University Freiburg, 79104 Freiburg, Germany

5 Bioinformatics Group, Department of Computer Science, Albert-Ludwigs-University Freiburg, 79110 Freiburg, Germany

6 Centre for Biological Systems Analysis (ZBSA), Albert-Ludwigs-University Freiburg, 79104 Freiburg, Germany

7 Group for Genome Dynamics in Neurodegenerative Diseases, German Center for Neurodegenerative Diseases (DZNE), 37075 Göttingen, Germany

8 Genetics and Experimental Bioinformatics, Faculty of Biology, Albert-Ludwigs-University Freiburg, 79104 Freiburg, Germany

9 Active Motif Incorporation, Carlsbad, CA, USA

10 Centre for Biological Signalling Studies (BIOSS), Albert-Ludwigs-University Freiburg, 79104 Freiburg, Germany

11 Institute of Molecular Medicine and Cell Research, Faculty of Medicine, Albert-Ludwigs-University Freiburg, 79104 Freiburg, Germany

12 Freiburg Institute for Advanced Studies, Albert-Ludwigs-University Freiburg, 79104 Freiburg, Germany

13 Department for Epigenetics and Systems Medicine in Neurodegenerative Diseases, German Center for Neurodegenerative Diseases (DZNE), 37075 Göttingen, Germany

14 Department for Psychiatry and Psychotherapy, University Medical Center, Georg August University Göttingen, 37075 Göttingen, Germany

15 Center for non-coding RNA in Technology and Health, University of Copenhagen, 1870 Frederiksberg C, Denmark

[P3] ChiRA: an integrated framework for chimeric read analysis from RNA-RNA interactome and RNA structurome data

Pavankumar Videm, Anup Kumar, Björn Andreas Grüning, and Rolf Backofen. *GigaScience*, 2021. DOI: 10.1093/gigascience/giaa158

Contributions of individual authors:
I am the main contributor to this work. I proposed the idea and implemented the ChiRA tool suite. I analyzed several RNA-RNA interactome data sets, benchmarked the workflow performance, integrated the tool suite into the Galaxy framework, developed the Galaxy workflow and training material. I wrote the major portion of the manuscript and revised it. Anup Kumar developed the ChiRAViz Galaxy visualization and was involved in writing corresponding sections of the manuscript. Björn Andreas Grüning and Oleg Zharkov supported galaxy integration and deployment. Rolf Backofen provided general consultation. All the authors were involved in revising the manuscript.

Pavankumar Videm

The following authors confirm the above stated contributions.

- Pavankumar Videm
- Anup Kumar
- Oleg Zharkov
- Björn Andreas Grüning
- Rolf Backofen

GigaScience, 10, 2021, 1–13

doi: 10.1093/gigascience/giaa158
Technical Note

TECHNICAL NOTE

ChiRA: an integrated framework for chimeric read analysis from RNA-RNA interactome and RNA structurome data

Pavankumar Videm [1], Anup Kumar [1], Oleg Zharkov[1], Björn Andreas Grüning [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. 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/0000-0001-8231-3323

Abstract

Background: With the advances in next-generation sequencing technologies, it is possible to determine RNA-RNA interaction and RNA structure predictions on a genome-wide level. The reads from these experiments usually are chimeric, with each arm generated from one of the interaction partners. Owing to short read lengths, often these sequenced arms ambiguously map to multiple locations. Thus, inferring the origin of these can be quite complicated. Here we present `ChiRA`, a generic framework for sensitive annotation of these chimeric reads, which in turn can be used to predict the sequenced hybrids. **Results:** Grouping reference loci on the basis of aligned common reads and quantification improved the handling of the multi-mapped reads in contrast to common strategies such as the selection of the longest hit or a random choice among all hits. On benchmark data `ChiRA` improved the number of correct alignments to the reference up to 3-fold. It is shown that the genes that belong to the common read loci share the same protein families or similar pathways. In published data, `ChiRA` could detect 3 times more new interactions compared to existing approaches. In addition, `ChiRAViz` can be used to visualize and filter large chimeric datasets intuitively. **Conclusion:** `ChiRA` tool suite provides a complete analysis and visualization framework along with ready-to-use Galaxy workflows and tutorials for RNA-RNA interactome and structurome datasets. Common read loci built by `ChiRA` can rescue multi-mapped reads on paralogous genes without requiring any information on gene relations. We showed that `ChiRA` is sensitive in detecting new RNA-RNA interactions from published RNA-RNA interactome datasets.

Keywords: microRNA; chimeric read; RNA-RNA interactome; structurome; visualization; CLASH; CLEAR-CLIP; PARIS; SPLASH; Galaxy workflow

Introduction

Many non-coding RNAs (ncRNAs) regulate gene expression, post-transcriptionally, via mechanisms such as activation or inhibition of translation, destabilization, localization, and processing. For example, a microRNA (miRNA) can downregulate target expression via translational inhibition or transcript destabilization, initiated by the formation of base pairs between the mature miRNA (~22 nt long) and the target RNA transcript [1]. For successful regulation, not only the intermolecular structure (i.e., the RNA-RNA interaction) but also the structure of the ncRNA itself (i.e., the intramolecular RNA structure) is key to

Received: 20 August 2020; **Revised:** 26 November 2020; **Accepted:** 15 December 2020

the regulatory process [2–4] because it influences the parts of the ncRNA that are accessible for RNA-RNA interactions. Computationally, the prediction of both inter- and intramolecular structure is non-trivial and results can be unreliable [5]. To support computational methods, several transcriptome-wide experimental protocols have been developed recently to detect both inter- and intramolecular RNA structure [6–10]. Although these protocols vary in their application-specific details, they currently all involve ligating the 2 RNA interaction partners together and subsequently sequencing the resulting chimeric RNA molecules using high-throughput sequencing technology. Chimeric RNAs from gene fusions by trans-splicing or chromosomal rearrangements can also be seen in RNA sequencing data. Such chimeric RNAs are often associated with specific cancer types [11, 12] and considered to be potential biomarkers [13, 14].

MicroRNAs have been a subject of avid research in the past decade owing mostly to 2 reasons: (i) it is proposed that each miRNA can regulate up to several hundred targets and that a substantial proportion of protein-coding genes are targeted by miRNAs at some stage [15] and (ii) individual miRNAs have been implicated in several notorious human diseases, such as different cancer types and neurodegenerative illnesses [16–18]. Therefore, accurate identification of miRNA targets is highly sought after. Despite numerous attempts, computational prediction approaches still deliver poor results with generally high false-positive rates, with no significant improvement observed in the past decade (see review [19]). Therefore, considerable effort has also gone into developing high-throughput experimental protocols, specifically designed to detect miRNA-target interactions (reviewed in [20]). The most recent line of development has been to ligate the miRNA to the site-specific interaction region of the target, selecting these interactions via cross-linking to 1 of the Argonaute proteins required for miRNA-based regulation, and to sequence the resulting chimeric RNA molecule, e.g., CLASH [6] and CLEAR-CLIP protocols [7]. Going beyond miRNAs, these protocols can obviously be applied to RNA interactions that involve a regulatory protein other than Argonaute. To generalize even further, researchers have applied the same idea to the detection of all transcriptome-wide RNA-RNA interactions. This includes both inter- and intramolecular base-pairing without the necessity of choosing a specific regulatory protein for cross-linking, as done, e.g., in PARIS [8], SPLASH [9], and LIGR-Seq [10]. Regardless of the protocol, the sequenced reads are chimeric; i.e., a fusion of 2 different RNA fragments corresponds either to intermolecular interaction or to 2 distinct parts of a single RNA molecule from its intramolecular structure.

Two main computational challenges arise from such chimeric-read data: (i) mapping the chimeric reads to 2 different locations on reference transcript annotations and (ii) dealing with the fact that these short RNA segments map to multiple locations, i.e., specifically dealing with multi-mapped reads. State-of-the-art mapping software, such as `Bowtie2` [21], `BWA-MEM` [22], and `STAR` [23], can both map chimeric reads and allow for multiple mapping locations, given the appropriate parameter settings. Subsequent to mapping, however, there are no satisfactory or standard solutions for correctly quantifying multi-mapped reads. Multi-mapped reads are either ignored or incorrectly assigned and/or quantified. Three common approaches exist for assigning multi-mapped reads: (i) they are not assigned but simply discarded; (ii) a read is assigned to each of the multi-mapped locations with equal distribution (e.g., with a count of 1 divided by the number of locations); and (iii) the true expression level is estimated by assigning the read to a multi-mapped location proportionally to the number of uniquely mapped reads in the vicinity of that location. The ability of the resulting read counts to capture expression levels or RNA interaction events increases with each approach. Obviously, discarding multi-mapped reads is a poor solution and definitely not an option when dealing with chimeric reads. Distributing counts equally under- or overestimates the actual expression in all locations in comparison to regions with uniquely mapping reads. The third approach can deliver accurate results but fails when it comes to distributing reads among gene families with very similar sequences, e.g., for miRNA gene families.

Existing software solutions that take the raw data input from RNA-interactome protocols and deliver quality interaction annotations are currently application or protocol specific. Most of them were released along with their corresponding published experimental protocols, and none of them has become a readily usable bioinformatic pipeline. There also exist generic standalone pipelines like `Hyb` [24], which was developed and demonstrated to deal with miRNA-specific data. From the computational side, there is thus still a major hurdle to overcome before such protocols can be broadly applied in practice: the availability of easy-to-use software that can process the raw data to produce accurate annotation and quantification of the identified RNA-RNA interactions. Here we present a method to resolve multi-mapping to very similar reference sequences from possible gene families and paralogs without requiring any prior annotation. Our method determines the best alignment for each multi-mapped read by an elegant quantification and scores them on the basis of the abundance of reference loci. Our `ChiRA` tool suite, Galaxy [25] workflows, and visualization provide a complete analysis framework for chimeric reads from RNA-RNA interactome and RNA structurome protocols. Thus, we aim to strengthen a weak link in the search for transcriptome-wide RNA interactions/structures.

Methods

We built a complete workflow that takes raw sequencing reads as input and outputs a comprehensive list of annotated interacting regions. This involves read deduplication, mapping, quantification (including multiple mapped reads) of reference loci to infer the correct locations on the basis of their expression, and hybridization of interacting reference loci. To offer a convenient interface on top of `ChiRA` output an interactive visualization, `ChiRAViz`, was developed. Fig. 1 shows the complete workflow built from the `ChiRA` and `ChiRAViz` tool suite. Each of the following sections corresponds to the steps represented (listed on the right side) in the figure.

Adapter clipping and read deduplication

Quality and adapter trimming, in general, are crucial for RNA-RNA interactome data but essential for small RNA-related interactome data. Mature miRNAs that interact with the targets are only ∼18–22 nt in length. Depending on the captured target sequence, chimeric reads often have adapters in them. In our analysis, ≥80% of sequenced reads from CLASH and CLEAR-CLIP datasets contained adapters. For our analysis we trimmed low-quality ends and adapters from the reads using `cutadapt` [26]. Reads that were shorter than 10 nucleotides were discarded, and the remaining reads were deduplicated to eliminate possible PCR duplicates. In general, not all identical reads are PCR duplicates. Gene isoforms or gene paralogs also result in duplicate RNA fragments. To uniquely identify the RNA fragments,

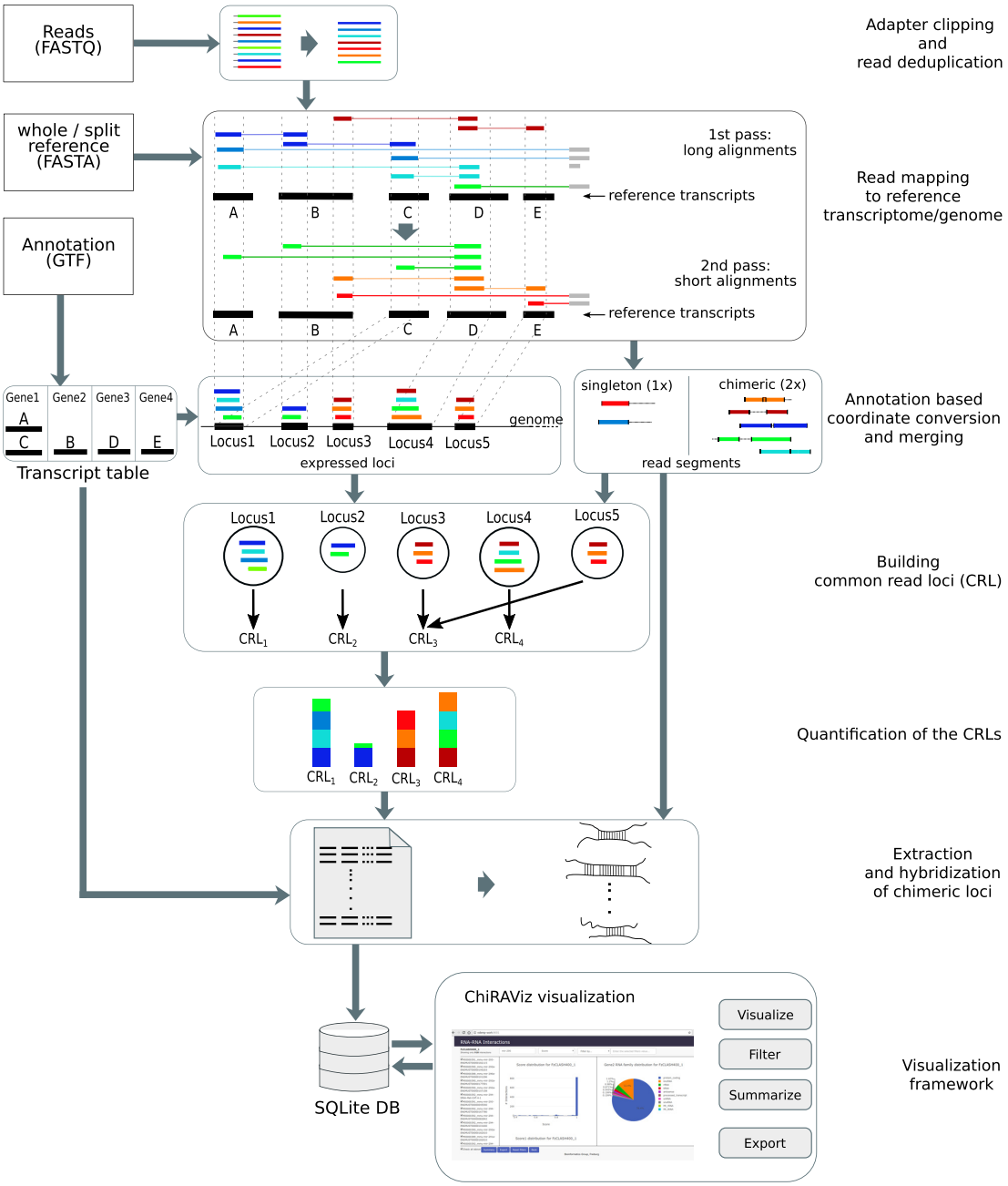


Figure 1: ChiRA workflow. First, the reads are deduplicated and mapped to reference sequences. Then the overlapping reference regions are merged into expressed loci. Given an annotation file, transcriptomic alignment positions are converted into genomic positions. Common read loci are built on the basis of the reads that are consistently multi-mapped among the expressed loci. The quantification is carried out at the common read loci level, and the interactions are scored and hybridized. With the visualization, users can search, filter, and export preferred interactions. Here transcripts A and C are 2 isoforms of a gene Gene1. Because of shared exons, multi-mappings to A and C can be collapsed into a single genomic locus Locus1. As Locus3 and Locus5 share all their read segments, they are merged into a single CRL. Owing to the quantification based on expectation maximization, the multi-mapped green read segment is counted more towards CRL_1 than CRL_2.

unique molecular identifiers (UMIs) are used. UMIs are short sequences of a specific length that are usually attached at the 5′ end of the RNA fragments during library preparation. We also deduplicate reads based on UMIs if they are present in the library. We consider identical reads with the same UMI as PCR duplicates, whereas identical reads with different UMIs are considered unique. The deduplication step may reduce the number of reads by orders of magnitude, which in turn can speed up the subsequent steps.

Read mapping to reference transcriptome/genome

In this step, we align the reads to the reference transcriptome or genome. For well-annotated organisms, we recommend using the transcriptome for the following reasons. (i) When mapped against a transcriptome, reads can be mapped linearly across the splice junctions. Especially, in the case of these small read fragments, it can be extremely difficult to map across the splice junctions when mapped to the genome. (ii) There is less chance of getting random false-positive hits for short read fragments

on transcriptome than on whole genome. Unfortunately, except for some model organisms, reference annotations are either incomplete or unreliable. In that case, using the whole genome sequence as a reference is a good choice for the following reasons. (i) An unreliable annotation leads to false conclusions on the type of detected interactions. (ii) An incomplete annotation results in false-negative interactions. Consider an example of CLASH data that predominantly contain miRNA and 3′ untranslated region (UTR) interactions. Mapping to a reference transcriptome with an incomplete 3′ UTR annotation fails to capture the most important category of interactions.

Currently, we support mapping with `BWA-MEM` [22] and `CLAN` [27]. `CLAN` is a recent exclusive chimeric read mapper and outputs the chimeric alignments in tabular format. `BWA-MEM` is also capable of producing chimeric reads by local alignment. With a high dynamic range in read lengths, it is not always possible to accurately map chimeric reads of different lengths with a single parameter setting. Hence, when `BWA-MEM` is used as the aligner, we do a 2-pass alignment. The first pass targets mapping long chimeric read segments whereas the second pass targets short ones. In the first pass, we use high alignment score thresholds and allow gaps and hence achieve long gapped chimeric alignments. In the second pass, we use a lower alignment score cut-off and do not allow any insertions or deletions. Therefore the second pass rescues short chimeric read segments with perfect matches on the reference. The default alignment settings were optimized on the miRNA interactome data from CLASH and CLEAR-CLIP protocols. The complete list of alignment settings can be found in the provided Galaxy histories (see Supplementary Section S4 for more details). `BWA-MEM` can output the alignments in Sequence Alignment/Map (SAM) format. We convert it into Binary sequence Alignment/Map (BAM) and use pysam [28] for further processing. It is important to consider that `BWA-MEM` randomly chooses 1 of the alignments as primary and writes all the alternative hits to the `XA` tag of the alignment. The true alignment can also be hidden under the `XA` tag and buried in the BAM file. `BWA-MEM` has an option (-h) that controls the writing of these suboptimal alignments to the output BAM file. In the second pass, we set it to a high number (default 100) so that we do not miss any of the equally good alternative alignments. The idea is to get as many multi-hits as possible and let `ChiRA` pick the best one in subsequent steps. In the end, we combine the alignments from both the alignment steps, parse the BAM file using pysam, and write them to a Browser Extensible Data (BED) file. In this step, we only keep the alignments that are mapped on the sense reference strand. If there is an `XA` tag for an alignment, we keep all the alternative alignments with the highest read coverage. In the end, we remove any duplicate hits in the second pass of the 2-pass alignment.

Because each chimeric read often contains 2 RNA fragments originating from 2 different RNA types, we allow mapping to 2 different reference transcriptomes (”split reference”). For example, for CLASH data, we encourage the use of a split reference, one containing miRNAs and the other containing the rest of the transcriptome, which restricts the output to miRNA-based interactions. The parameters such as seed lengths and alignment scores are dependent on the type of the data or expected length of chimeric arms. In our experience, the default settings work well with the miRNA interactome data.

Annotation-based coordinate conversion and merging

Given an annotation file in Gene Transfer Format (GTF), we convert transcriptome locations to genomic locations because working on the genomic locations is less ambiguous. The main problem with transcript locations is that the reads mapped to the exons that are shared among the isoforms appear to be multi-mapped. But at the genomic level, these are uniquely mapped. In the absence of a GTF file `ChiRA` can still work with transcriptome locations.

Merge reference positions to define interaction sites

Because the experimental protocols may generate several reads covering different parts of an interaction site, we have to define an interaction site by combining overlapping alignments. This step separates alignments stemming from the same interaction sites from alignments that cover a completely different interaction site on the same transcript. For example, 2 different miRNAs may target a single mRNA at 2 different locations such as coding sequence and 3′ UTR. In more detail, we merge the significantly overlapping alignments based on the reference mapping locations to generate so-called ”expressed loci.” A single transcript may have multiple such expressed loci. For an alignment to merge into an existing expressed locus, both the alignment and the locus must reciprocally overlap >70% (default value) in length.

While this approach works well with interaction sites that have a low to medium coverage, it might fail in the case of sites with high coverage because the likelihood of finding 2 alignments with 70% overlap at random increases. For this purpose, we have an alternative merging mechanism using `blockbuster` [29]. `blockbuster` defines the blocks of alignments based on a Gaussian approximation of the read coverage. Subsequently on the basis of the `-distance` parameter, it places adjacent read blocks into clusters. However, we ignore this cluster information and work further on the block level. We merge any overlapping blocks to define potential interaction loci. This approach is thus similar to (but also simpler than) the one introduced and successfully applied for cross-linking immunoprecipitation sequencing (CLIP-seq) peak calling in Holmqvist et al. [30].

Merge read positions to define chimeric arms

In this step, we identify all chimeric and non-chimeric (singleton) aligned reads. A chimeric read has ≥2 non-overlapping portions on the read mapped to distinct reference loci. If a sequenced read is chimeric and it is uniquely mapped to the reference, then we have ≤2 alignments each belonging to 1 chimeric arm. If a sequenced read is a singleton and mapped uniquely, then we have maximally 1 alignment. We call each aligned portion of the read a ”read segment.” In later steps, during quantification, a singleton read will be treated as 1 read whereas a chimeric read will be treated as 2 (1 for each segment) separate reads. Hence it is crucial to define the chimeric split points of the reads. A chimeric split point can be identified by its non-overlapping segments. Owing to local alignment and repetitive parts on the reference sequences, some overlapping segments multi-map with few bases shifted. Considering each such highly conserved read segment separately penalizes the overall read segment contribution in quantification. Hence, we further merge read segments that overlap ≥70% (default value) of their length into a single segment. In theory, there are only 2 interacting read segments because there are maximally 2 interacting RNA fragments captured in the interactome experiments. Owing to sensitive alignment settings, some reads also result in >2 segments. After a subsequent quantification step, only the 2 most probable chimeric arms will be considered for each read.

Building common read loci

There are cases where read segments map to the gene families or paralogous loci sharing the common sequences. It is huge a challenge to find a decent annotation that carries gene family or paralog information. It was shown by Robert and Watson [31] that grouping of genes based on multi-mapped reads resulted in groups of gene families and analyzing the RNA-seq data at this group level was biologically relevant. Similarly, we propose a method to group multi-mapped loci that does not depend on any annotation. If 2 loci share a large portion of their multi-mapped reads, their sequences tend to be very similar or originate from the same gene families or paralogs or have similar pathways (see Results and Discussion). Hence, we group expressed loci into common read loci (CRL) if they share a significant number of multi-mapped reads. Here we use single-linkage clustering with the Jaccard index to measure the similarity between the expressed loci. To merge an expressed locus into an existing CRL, the Jaccard index of sets of reads between that locus and the CRL should be greater than a user-defined threshold (default of 0.7). We merge the loci in order by size. If a locus failed to share a significant portion of multi-mapped reads with any other CRL, then it gets its own CRL. If the reads were mapped to transcriptome and the user does not provide any gene annotation file, CRLs are well capable of grouping multi-mapped reads that map to gene isoforms. See Algorithm 1 for CRL creation pseudo code.

Result: List of CRLs
$\mathbb{C} \leftarrow \{\}$;
for $L_i \in \mathbb{L}$ **do**
  `match` $\leftarrow$ *False*;
  for $C_k \in \mathbb{C}$ **do**
    if $\frac{C_k \cap L_i}{C_k \cup L_i} \geq \theta$ **then**
      $C_k \leftarrow C_k \cup L_i$;
      `match` $\leftarrow$ *True*
    end
  end
  if not `match` **then**
    $\bigcup\{\mathbb{C}, L_i\}$;
  end
end

Algorithm 1: CRL creation from expressed loci. $\mathbb{C}$ is the list of CRLs; $\mathbb{L}$ is the list of expressed loci; L_i is the set of read segments of an expressed locus i and C_k is the set of read segments of a CRL k.

Quantification of the CRLs

To score the mapped chimeric reads, we first need to estimate the expression of the CRLs by quantification. Quantification helps to assess the true origin of a read segment in the case of multi-mapping. It has been shown that proper quantification of multi-mapped reads led to the discovery of novel protein-RNA interactions from CLIP-seq data [32, 33]. A study on RNA-seq data revealed that the expression of genes with multi-mapped reads was underestimated by common quantification methods [31]. There exist comprehensive studies on methods [34, 35] and metrics [36] for quantification of RNA-seq data, but direct application of these methods to our data is not possible for the following reasons. First, it is hard to supply our pre-built locus-CRL relations to the quantification tools on the fly. Second, unlike our short reference loci, the reference RNAs in RNA-seq have multiple exons and are much longer. In RNA-seq, often the quantification is done at the isoform level, where exons that are unique to that isoform help to resolve the multi-mapping by estimating the total maximum likelihood for that isoform. But in interactome data, there is only a part of the interacting exons that is captured and the rest is missing. If this interacting part of an exon is shared among the isoforms, the read segments mapped are still called multi-mapped and each transcript gets an equal share from the read segment. Therefore we implemented an approach to quantify the CRLs based on the expectation-maximization (EM) algorithm. In this quantification, all multi-mapped reads that map to different expressed loci of a CRL are considered as uniquely mapped to that CRL.

Let $\mathbb{S}$ be the set of all read segments with $N = |\mathbb{S}|$ and $\mathbb{C}$ be the set of all CRLs with $K = |\mathbb{C}|$. We follow Xing et al. [37] in the annotation, where we estimate the CRL abundance by determining the likelihood $\rho_c = \Pr[s \in c]$ that a read segment s actually stemmed from CRL c. We denote with ρ the vector of all ρ_c. Note that when the CRLs have a similar length as in our case, length normalization can be omitted; i.e., ρ_c are then direct estimates for CRL abundances. In the case of multiple mapping, we define 2 indicator variable matrices to model the read segment selection process. We have an $N \times K$ indicator matrix $Z = (z_{s,c})_{\substack{s \in \mathbb{S} \\ c \in \mathbb{C}}}$ with

$$z_{s,c} = \begin{cases} 1 & \text{if read segment } s \text{ is from CRL } c \\ 0 & \text{else} \end{cases}$$

However, this is not directly observable in the case that the reads map to different CRLs. This can be overcome by introducing another matrix $Y = (y_{s,c})_{\substack{s \in \mathbb{S} \\ c \in \mathbb{C}}}$ with

$$y_{s,c} = \begin{cases} 1 & \text{if read segment } s \text{ maps to CRL } c \\ 0 & \text{else} \end{cases}$$

Note that we have in each row of Z exactly 1 entry with 1, whereas in Y we can have several such entries. Furthermore, $y_{s,c} = 0$ implies $z_{s,c} = 0$. We call Z the committed categorization and Y the uncommitted categorization. In the case of multiple mappings, we have many different Z-matrices that are compatible with Y (meaning that each row in Z has sum 1, and $y_{s,c} = 0$ implies $z_{s,c} = 0$) and are unobservable. Then, the likelihood of the observation (i.e., read segments) $\mathcal{L}(\rho)$ is defined as follows:

$$\mathcal{L}(\rho) = \prod_s \sum_c y_{s,c}\rho_c.$$

However, this maximum likelihood solution for $\mathcal{L}(\rho)$ cannot be obtained in closed form. Hence, we apply the following EM algorithm to determine the maximal likelihood estimates $\hat{\rho}$.

E-Step

Let $\rho^{(t)}$ be the vector of abundance estimates $\rho_c^{(t)}$ in round t of the EM algorithm. The E-Step consists of the determination of the expected values for the hidden variables:

$$E\left[z_{s,c} \mid Y, \rho^{(t)}\right] = \Pr\left(z_{s,c} = 1 \mid \rho^{(t)}, Y\right) = \frac{\rho_c^{(t)}}{\sum_{c'} y_{s,c'}\rho_{c'}^{(t)}}. \quad (1)$$

Note that we are interested not only in determining the abundances of the CRLs but also in the likelihood that a read segment s is from a CRL c, i.e., in $\Pr(z_{s,c} = 1 \mid \hat{\rho}, Y)$, for which we can use the values calculated in equation (1) in the last E-Step of the EM algorithm. From these likelihoods, we can calculate the probability $\Pr[(s, s') \in c \leftrightarrow c']$ that a chimeric read ..s..s'.. is an interaction between CRLs c and c':

$$\Pr[(s, s') \in c \leftrightarrow c'] = \Pr(z_{s,c} = 1 \mid \hat{\rho}, Y)\Pr(z_{s',c'} = 1 \mid \hat{\rho}, Y).$$

Note that the relative abundance of the transcript does not influence this probability because we consider only the read segment s (respectively s') and $\sum_c y_{s,c}\Pr(z_{s,c} = 1 \mid \hat{\rho}, Y) = 1$ [respectively $\sum_c y_{s',c}\Pr(z_{s',c} = 1 \mid \hat{\rho}, Y) = 1$].

M-Step

The M-Step is simply the maximum likelihood estimate, given the hidden values z:

$$\rho_c^{(t+1)} = \frac{\sum_s z_{s,c}^{(t+1)}}{N}. \quad (2)$$

We repeat the E and M steps until the sum of differences between the relative abundances of CRLs in 2 consecutive iterations is not higher than a user-defined value ϵ, i.e., $\sum_{c=1}^{K} |\rho_c^{(t+1)} - \rho_c^t| \leq \epsilon$. The default value for ϵ that we use is $1e^{-5}$. The expression levels of the CRLs are reported in transcripts per million (TPM). Calculation of TPM is explained in Supplementary Section S3.

Extraction and hybridization of chimeric loci

In this final step, we extract the 2 most probable chimeric arms for each chimeric read along with their alignment and sequence information. If a GTF file is provided, we annotate the interacting regions with gene IDs, symbols, biotypes, and so forth. For protein-coding genes, the biotypes are further categorized into 5′ UTR, coding sequence, and 3′ UTR. For hybridization of chimeric arms we use `IntaRNA` [38]. Occasionally, the real interaction is in the vicinity of the sequenced arms. For this reason, we hybridize the reference loci sequences from the output instead of the aligned read sequences. These reference loci are merged from multiple overlapping alignments and already contain some context of mapped arm locations.

Visualization framework

Motivation

`ChiRAViz` visualizer is developed in JavaScript (JS) to summarize, filter, and visualize the output of `ChiRA`. The output of `ChiRA` is a tabular file with each record containing interacting positions of a read on the reference with their annotation information (in case GTF was provided during the analysis) such as gene IDs, biotypes, gene symbols, alignment information, and so fort. Each such record contains >30 columns, and depending on the library size and the complexity of the interactome there can be millions of records in a single output file. Working with such large data is hard, especially extracting elements of significant interactions from their native tabular form. Therefore, to summarize the complete data, a visualizer is needed where information can be filtered and shown in the form of various charts that are easier to understand.

Datatype

The visualizer is integrated into Galaxy as a native visualization for chira.sqlite datatype. Using a database allows SQLite queries to be formulated and executed to fetch a subset of data by applying filters on its columns.

User Interface

The user interface (UI) of the visualizer is created using JS and multiple JS-related packages such as UnderscoreJS, Bootstrap, and jQuery. UnderscoreJS methods are used for better manipulation of JS arrays and dictionaries. Bootstrap is used for styling the UI and jQuery for document object model manipulation and asynchronous methods to fetch data from the database file.

Results and Discussion

Data

We applied `ChiRA` on a custom-made benchmark dataset to assess the performance, and on published RNA-RNA interactome and structurome datasets to validate the approach and showcase the functionality.

Benchmark data

Based on the benchmark data provided by the `CLAN` publication, we produced our benchmark data to test the performance of `ChiRA`. The reads were unchanged, but we modified the reference sequences. The reads imitate CLASH experimental data. Each read is a direct fusion of (sub)sequences of human hg38 miRBase [39] mature miRNAs and a random TargetScan [40] target sequence (i.e., the target sequence is not necessarily a true target of this miRNA). The reads are in FASTA format and contain 1 million reads per sample. There are 5 different samples of simulated chimeric reads, each containing a specific chimeric arm length (10, 12, 15, 18, and 20). These datasets are called "noInsert" data. There is a second set of data with the same arm lengths but a random 5-nucleotide sequence inserted either between or at the ends of the arms of each chimeric read. This dataset is called "Insert" data. In both cases, if the reference miRNA or reference TargetScan target is shorter than the arm length, the whole reference sequence was used.

As a reference database, we used miRBase mature miRNAs together with TargetScan target sites. The reference sequences used in the `CLAN` publication were very short in length, with a mean length of 21 nt for miRNAs and 14 nt for target reference sequences. Using those short TargetScan targets only as a reference is not realistic. Moreover, the TargetScan target sequences were predicted by a computational approach and generally not used as a reference database. With very short target sequences it is fairly easy for the aligners to map the reads to exact locations uniquely. Adding some context poses an additional challenge to the aligners and results in multi- or wrong alignments. Hence, to test the potential of our workflow on more complicated and near real-world reference sequences, we modified the target reference data as follows. First, we sorted all the target genomic regions and then extended each region until the next target region was within a 200-nt range. In the end, we extracted the sequences of these positions. This procedure results in target sequences of various lengths. Similar to the real reference database, there is also a fair chance of having multiple target sites on a single reference sequence. In the original `CLAN` benchmark data, there were duplicate reference sequences. These were coming from the same duplicated targets

of different miRNAs. All these duplicated reference sequences have been removed from our benchmark data.

Published data

To show the functionality of ChiRA, we applied ChiRA also on published datasets. We analyzed human miRNA interactome data from CLASH and mouse interactome data from CLEAR-CLIP protocols. For RNA-RNA interactome and structurome data, we used polyA enriched SPLASH samples from lymphoblastoid cells, human embryonic stem cells (ES), and human retinoic acid differentiated cells, as well as mouse ES and human HEK293T samples from PARIS protocol. For CLASH and CLEAR-CLIP we built the reference databases as explained in the methods from their respective articles. For SPLASH and PARIS datasets we used the complementary DNA sequences of hg38 and mm10 genome builds from Ensembl revision 100. A summary of published data and their processing is provided in the Supplementary Section S2.

Performance on the benchmark data

We chose the same terminology as in the CLAN article to categorize the reads on the basis of alignment types. An "arm" is an arm of chimeric read segments, and an "agreed arm" is an arm that has an alignment with ≥80% overlap on correct reference location. The categories are defined as follows: "perfect": has both uniquely mapped agreed arms; "partial_multi": has 1 uniquely mapped agreed arm and 1 multi-mapped agreed arm; "both_multi": both arms are multi-mapped agreed arms; "partial_wrong": has 1 uniquely mapped agreed arm and 1 wrongly mapped; "both_wrong": both arms are wrongly mapped; "partial_miss": has 1 mapped and 1 unmapped arm; "both_miss": both arms are unmapped. We carried out 2 separate runs of ChiRA using BWA-MEM and CLAN aligners. Figs. 2 and 3 show their respective performance. Each bar in the plot represents the result of 1 of the 2 modes "naive" or "chira." The naive mode involves running the alignment tool (BWA-MEM or CLAN depending on the run) on the single reference database obtained by concatenating both mature miRNAs and TargetScan targets together, resulting in a gapped alignment. The reads are then directly categorized into 1 of the 7 aforementioned categories. When using BWA-MEM in naive mode, we considered only the longest alignment for each arm. In cases of multiple longest alignments, we considered all of them. In the chira mode, the ChiRA workflow with the corresponding aligner was used to obtain the results. In this mode, we used a split reference, i.e., the 2 separate reference databases for mature miRNAs and target sequences. We also enabled CRL creation while quantifying. The bars are then grouped horizontally on the basis of the arm lengths and then furthermore grouped by whether the reads contain inserts.

The most challenging cases are with arm lengths of 10 and 12 nt. Being very short sequences, these cases tend to result in a lot more multi-mappings than the others. In naive mode, for an arm length of 10 nt there are a negligible number of perfect reads. The chira mode could detect some perfect reads, but they are still <10% in any case. Considering the short length of the arms, it is clear that these generally map to multiple or wrong locations. For an arm length of 12 nt, there is >2.5-fold increment in perfect reads from naive to chira mode. At this arm length there is still not an acceptable number of perfect reads except for the CLAN aligner on noInsert data. The percentages of perfect reads are consistently ∼70% for arms of lengths ≥15 nt for both the aligners in naive mode. This observation indicates that the sequenced RNA fragments must be ≥15 nt long to be uniquely identified at an acceptable rate. Despite being a chimeric read aligner, CLAN produced a significant amount of ambiguous partial_multi and both_multi alignments in naive mode (Fig. 3). ChiRA sensitive mapping combined with CRL quantification is good at picking the correct alignments. For this reason, in chira mode there are ≥10% more perfect reads in all samples.

There is a decreasing trend in perfect reads for CLAN-based results on reads of lengths 15–20 nt with inserts, whereas it is more stable for BWA-MEM–based results. As this trend can also be seen in naive mode, it is likely more of a flaw of the aligner than ChiRA processing. For BWA-MEM–based alignments we consider an arm to be unmapped if it has no alignment on the sense strand. For this reason, there are many reads in the partial_miss and both_miss categories for BWA-MEM–based results even though there might be wrong alignments on the anti-sense strand.

For reads with shorter arms, even with very sensitive alignment settings, both aligners struggled to map to correct locations. Hence, we suggest tweaking the alignment settings of the aligners to capture read segments of ≥15 nt long. Shorter alignments often tend to be from ambiguous or wrong locations and eventually lead to false-positive interactions.

Inferring CRL significance from published data

For the analysis of all published datasets, we used BWA-MEM to map the reads to reference databases and enabled CRL creation. From the process of creating CRLs, it is noticeable that the loci of a CRL share a common reference sequence. In this section we show that CRLs are not just random groups but have high sequence identity and that genes associated with the loci of a CRL implicate common annotations and functions.

CRLs and sequence identity

To determine the extent of the similarity among the CRL member loci, we computed the sequence identities. Each locus within a CRL is unique and does not contain any duplicate regions from gene isoforms. While running the workflow we used the default value of 0.7 for the option `--crl_share_threshold`. With this option loci having ≥70% of reads in common are grouped into a CRL. First, for each CRL we computed all pairwise global alignments among the loci using the Biopython module `pairwise2` [41] with default alignment parameters. With no gap or mismatch penalties in default parameters, we essentially counted the number of matching bases. We then calculated the average of pairwise sequence identities (APSI) per CRL and a final mean per sample overall CRLs normalized by the CRL size. Pairwise sequence identity is the ratio of the alignment score to the average sequence length of the sequences. As a baseline, for each CRL size, we randomly sampled loci and computed the APSIs.

Fig. 4 shows the box plots of the APSIs over all the samples in each sequencing protocol. Notably, with a default value of 0.7 for CRL share, we see that all the protocols have a median of ≥90% APSI s, whereas the APSIs for randomly sampled loci are only ∼50%. This similarity among the CRL loci is compelling considering that the global alignment is used. It is also consistent across different sequencing protocols.

Biological relevance of CRLs

Robert and Watson [31] showed for a handful of genes that the groups of genes that are consistently multi-mapped are from gene families. Similarly, here on a large scale, we analyzed whether the genes that constitute the CRLs share biologically

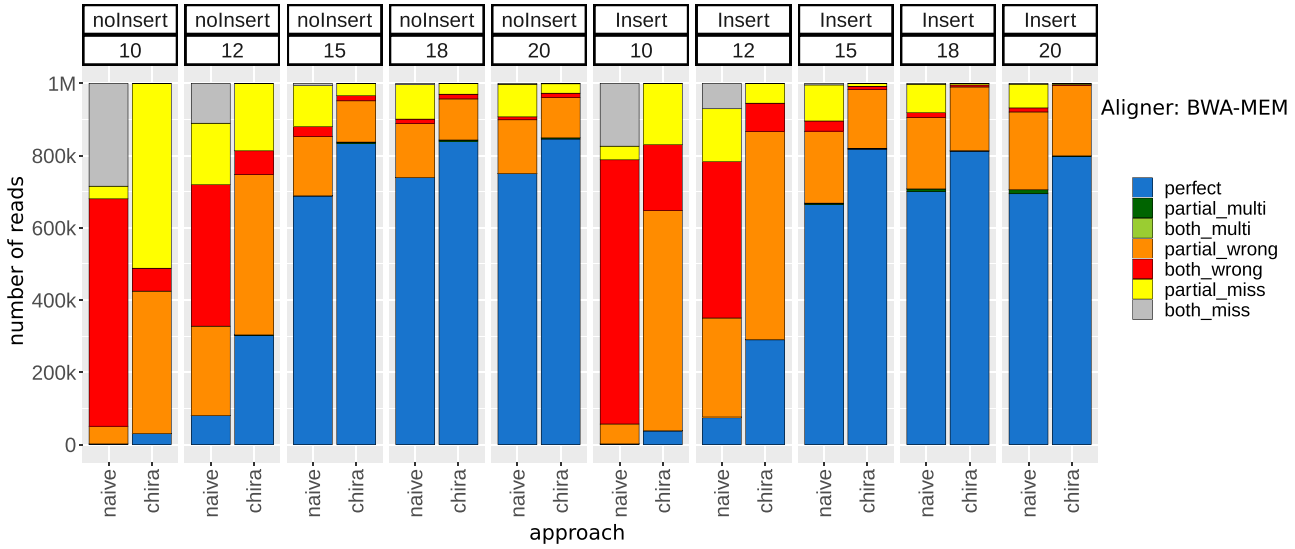


Figure 2: Performance of BWA-MEM–based ChiRA compared to naive approach on benchmark data. ChiRA-based results have ≥10% more perfect hits compared to naive mode for any arm length.

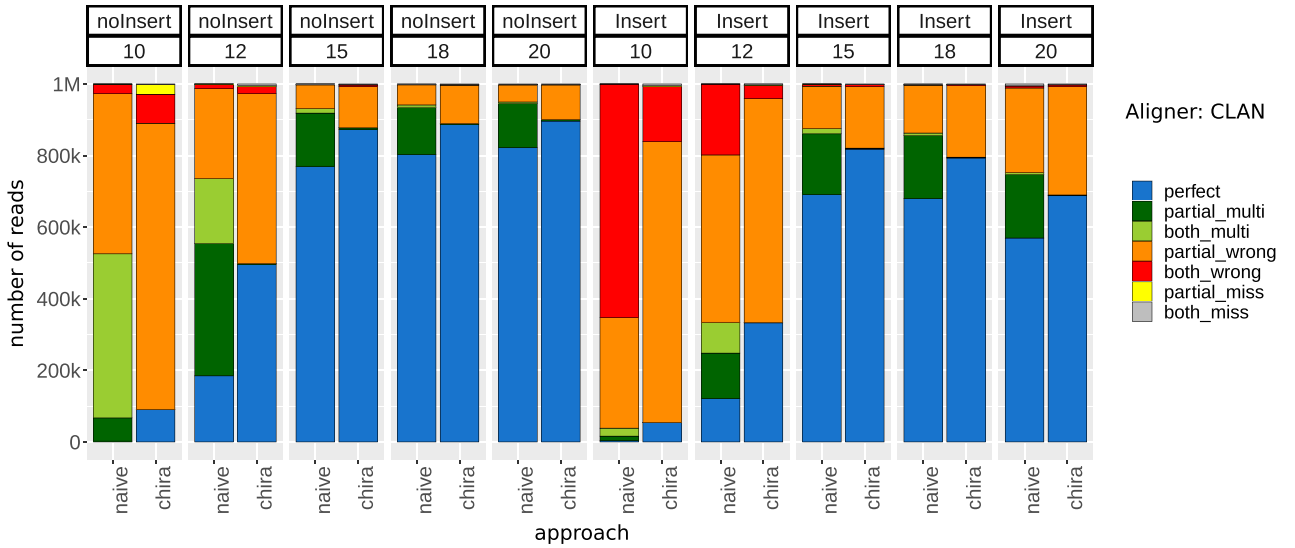


Figure 3: Performance of CLAN-based ChiRA compared to naive approach on benchmark data. Being a chimeric read aligner CLAN produced fewer wrong hits and more multi-mapped hits that contain the true alignment. CRL-based ChiRA could pick the correct reference from the multi-mapped hits for any arm length. Note that although there are more multi-hits (green) in naive mode compared to chira mode, the origin of these reads is still uncertain.

relevant information. We created an annotation database by extracting Rfam family, Ensembl protein family, and KEGG pathway information from Ensembl biomart [42]. We excluded all the CRLs from the analysis that do not contain ≥2 annotated genes in the database. For each CRL, we counted the number of genes with the same protein family or the same KEGG pathway or enzyme ID. We then calculated the ratio of this number to the total number of genes per CRL. In the end, we computed a weighted average over all the samples for each experimental protocol. As a control for each CRL, we randomly sampled the same number of genes out of the databases and calculated the percentage of those genes sharing a protein family or KEGG ID. Fig. 5 shows the box plots for the above explained values for CRL genes and randomly sampled genes for each experimental protocol. In all cases, it is evident that for most of the CRLs gene constitution is explainable compared to random gene constitution. Although not all of the CRLs have explainable sources (e.g., CLEAR-CLIP and SPLASH), overall the genes from a CRL more often belong to the same gene family or KEGG pathway than randomly sampled genes. Note that the CRLs are built from the short loci, which are just tiny portions of the genes. But here we are evaluating them at the level of the whole gene to which they belong. Although the loci are highly similar, the gene-level assessment might not necessarily explain all the CRLs.

Sensitive chimeric read detection using ChiRA

Finally, we tested the sensitivity of ChiRA in detecting interactions by analyzing all CLASH and CLEAR-CLIP mouse datasets and subsequently comparing them with the published inter-

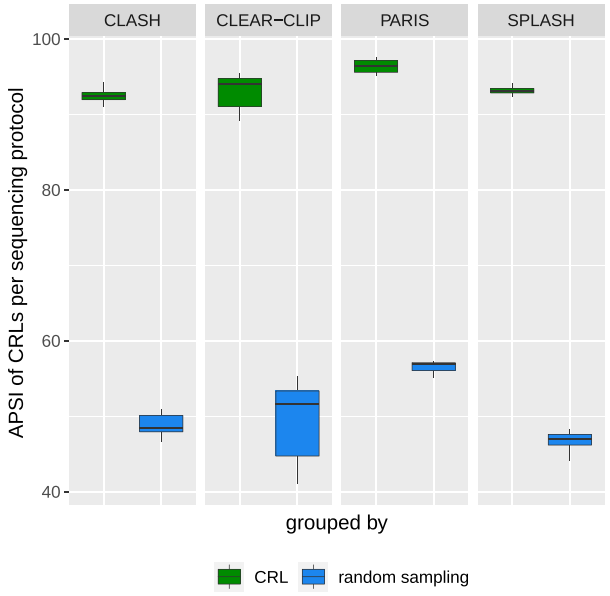


Figure 4: Box plots showing the average pairwise sequence identities (APSIs) among the member loci of the CRLs and those calculated from randomly sampled loci for each sequencing protocol. The loci sequences belonging to the CRLs show higher median of APSI than random sampling.

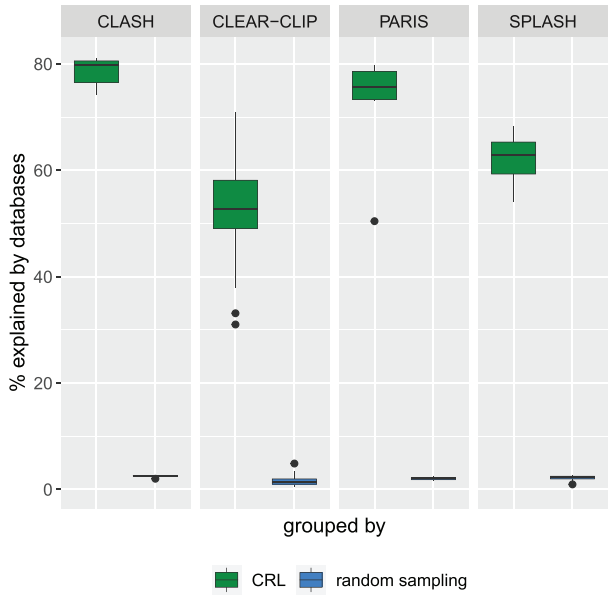


Figure 5: Validation of the CRLs from the different experimental protocols comparing the percentage of genes belonging to the CRLs that belong to the same protein family or have a similar KEGG identifier vs those when the genes were randomly sampled. In all datasets, it is clear that the genes constituting CRLs are found to be related in at least one of the annotation databases.

actions. To be consistent with the published interactions, for CLASH we considered miRNA IDs with their target transcript positions and for CLEAR-CLIP miRNA IDs with their target genomic positions. Because we used the transcriptomic database for mapping, we ignored the intronic and intergenic target sites from CLEAR-CLIP published interactions. From `ChiRA` output, we selected chimeric reads with a final probability of $\geq$0.5 and the detected interacting loci that could be hybridized by `IntaRNA`.

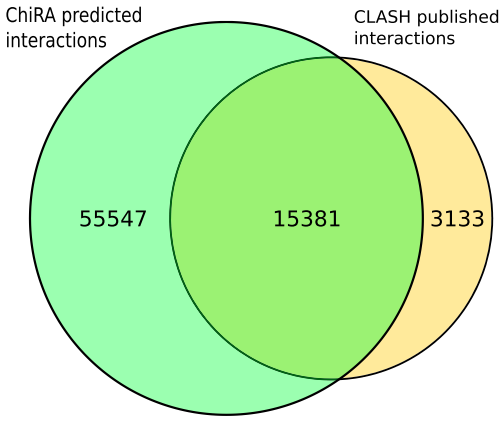


Figure 6: Number of interactions that were detected by `ChiRA` compared to published interactions in CLASH datasets.

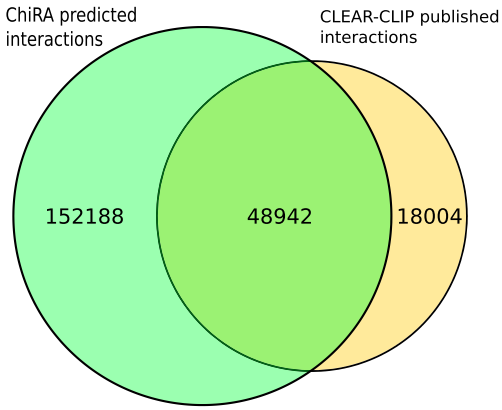


Figure 7: Number of interactions that were detected by `ChiRA` compared to published interactions in CLEAR-CLIP datasets.

Figs. 6 and 7 show Venn diagrams intersecting the published interactions and interactions predicted by `ChiRA` for the CLASH and CLEAR-CLIP, datasets respectively. There is a large overlap of 83% with CLASH and 73% with CLEAR-CLIP published interactions despite using different aligners. Compared to the published dataset(s), `ChiRA` on average detects 3 times more interactions. Given our analysis of benchmark data (Figs. 2 and 3), and supported by `IntaRNA` hybridization of interacting loci, we assume that the majority of these detected interactions are true-positive results.

Visualization of chimeric reads

The visualization has 3 views. The first page, shown in Fig. 8A, displays numerous plots to summarize the complete data. Two pie charts show the RNA biotype distribution of interacting transcripts. Another pie chart shows the distribution of interactions. Moreover, there is a bar plot that lists the gene symbols of top interacting transcripts sorted in decreasing order of their respective loci expressions. At the top of the page, there are 2 select boxes for choosing the interacting RNA types. When an interacting pair is chosen from these select boxes, it redirects to the second page (Fig. 8B), which shows all the interactions that involve these selected RNA biotypes. On the left, there is a list of unique combinations of gene symbols that represent unique RNA-RNA interactions. At the top of this page, there are several filters such as search and sort, which facilitate fetching data in the desired

Figure 8: `ChiRAViz` Galaxy visualization. (A) The home page of the visualization. The plots on this page summarize the RNA biotypes of left and right chimeric arms, types of interactions, and highly abundant genes within the sample. (B) The second page shows the interactions of selected biotypes. On this page, users can further search, sort, and filter the interactions and obtain a deep summary of filtered interactions. (C) Interaction information page that shows such useful information as gene symbol, transcript IDs, gene IDs, expression level, biotypes, a depiction of interaction reference regions at transcript level, an illustration of the aligned read positions, and `IntaRNA` predicted hybrid.

way. All or some of these entries can be selected together and a summary can be seen in the form of pie charts, histograms, and transcript-level alignment positions. The pie charts show distributions of the gene symbols and biotypes, and the histograms show the distributions of alignment scores and their loci expressions. The alignment regions on each interacting transcript are also depicted with the start, end, and length of the alignment. All the selected interactions can be exported as a tab-separated value file to the local computer. The pagination shown at the top left corner enables navigation through all the interactions and displays a small number of interactions (50) at a time, which simplifies the UI. All the unique reads associated with each interaction can be seen by clicking on the "+" icon adjacent to the interactions themselves. Clicking on any of these single records displays the interaction summary page, as shown in Fig. 8C. This page shows all the information related to interacting partners such as gene ID, gene symbol, biotype, alignment start and end positions, transcript length, CIGAR string of the read alignment, and the expression of its corresponding locus in TPM. If there is an `IntaRNA` predicted hybrid, it is shown at the bottom of this page.

Integration into Galaxy framework and tutorial

Galaxy [25] has been one of the most popular resources for reproducible research. It allows easy execution of tools and complex workflows on a web-based graphical user interface. With public Galaxy servers, users also get access to huge computing resources. We integrated all of our tools intoGalaxy. The whole Python suite is available through Bioconda [43] and BioContainers [44] for easy installation. Galaxy Training Network (GTN) is a Galaxy community aimed at developing analysis-specific training material [45]. We developed training material for RNA-RNA interactome data analysis that includes a step-by-

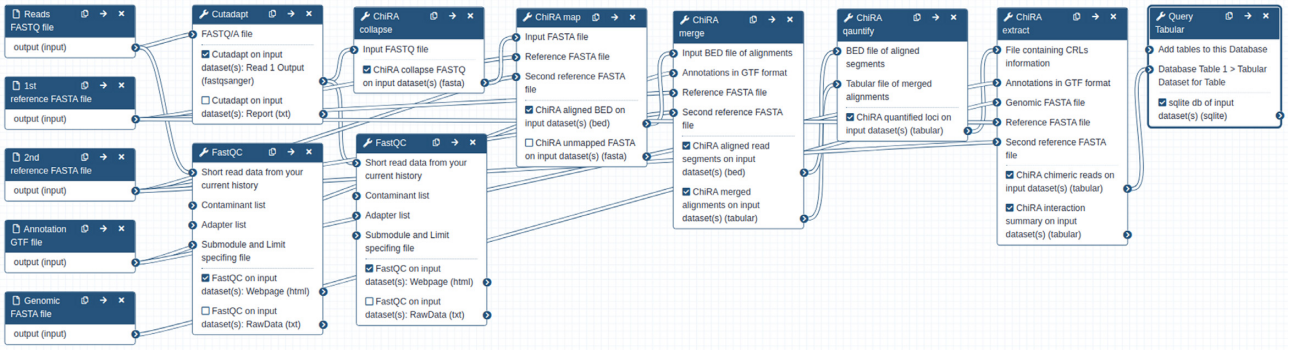


Figure 9: ChiRA Galaxy workflow. The workflow takes the FASTQ files that contain raw sequencing reads, process them, and produces a tabular and an SQLite database of interactions that are ready to be visualized by ChiRAViz.

step guide to hands-on Galaxy analysis workflows with example datasets, ready-to-use Galaxy workflows, and an example Galaxy history. The training material also deals with the visualization framework. Being nicely coupled into the Galaxy ecosystem, ChiRA is now part of RNA workbench [46], a large comprehensive Galaxy-based web server for RNA-based research. All the data and ChiRA analysis discussed in this article are available through RNA workbench. Fig. 9 shows the ChiRA Galaxy workflow that uses split reference.

Conclusion

In this article, we presented a comprehensive solution for RNA-RNA interactome and RNA structurome data analysis. Our method of creating CRLs from loci with consistent multi-mapped reads and quantification proved to rescue more reads from benchmark data. We also showed that the loci within a CRL have high sequence identities and the genes that constitute the CRLs originate from the same protein families or share common functional pathways, revealing that it is sensible to group consistently multi-mapped loci into CRLs. To our knowledge, ChiRA along with ChiRAViz is the only tool suite that makes analysis of RNA-interactome and structurome datasets easily accessible to users through Bioconda and Galaxy.

Availability of Source Code and Requirements

- Project name: ChiRA
- Project home page: https://github.com/pavanvidem/chira
- Visualization: https://github.com/galaxyproject/galaxy/tree/dev/config/plugins/visualizations/chiraviz
- Operating system(s): Platform independent
- Programming language: Python
- Other requirements: Anaconda
- Installation: `conda install -c conda-forge -c bioconda chira`
- License: GNU General Public License Version 3
- Galaxy tool suite: https://github.com/galaxyproject/tools-iuc/tree/master/tools/chira
- Galaxy training tutorial: https://galaxyproject.github.io/training-material/topics/transcriptomics/tutorials/rna-interactome/tutorial.html
- Galaxy workflows: https://rna.usegalaxy.eu/u/videmp/w/rna-rna-interactome-analysis (using BWA-MEM) https://rna.usegalaxy.eu/u/videmp/w/rna-rna-interactome-analysis-using-clan (using CLAN)
- BiotoolsID: chira
- RRID:SCR_019219

Data Availability

The benchmark data that were used to evaluate the performance of ChiRA can be obtained from Zenodo [47]. Snapshots of our code and other data are openly available in the GigaScience repository, GigaDB [48].

Additional Files

Supplementary Section S2. Data and pre-processing.
Supplementary Section S3. Calculation of Transcripts per Million.
Supplementary Section S4. Data availability.

Abbreviations

BAM: Binary sequence Alignment/Map; BED: Browser Extensible Data; BWA: Burrows-Wheeler Aligner; CIGAR: Compact Idiosyncratic Gapped Alignment Report; CLASH: Cross-linking Ligation and Sequencing of Hybrids; CLIP-seq: cross-linking immunoprecipitation sequencing; CRL: common read loci; EM: expectation-maximization; GTF: Gene Transfer Format; JS: JavaScript; KEGG: Kyoto Encyclopedia of Genes and Genomes; LIGR-Seq: ligation of interacting RNA followed by high-throughput sequencing; miRNA: microRNA; ncRNA: non-coding RNA; nt: nucleotides; PARIS: Psoralen Analysis of RNA Interactions and Structures; PSI: pairwise sequence identities; SAM: Sequence Alignment/Map; SPLASH: Sequencing of Psoralen crosslinked, Ligated, and Selected Hybrids; UI: user interface; UTR: untranslated region.

Competing Interests

The authors declare that they have no competing interests.

Funding

This work was supported by the German Research Foundation (DFG) grant eCLASH: Definition des Interactomes kleiner RNA [2168/14-1 awarded to R.B.] and the DFG-funded Collaborative Research Centre 992 Medical Epigenetics [SFB 992/1 2012 awarded to R.B.]. The article processing charge was funded by the Baden-Württemberg Ministry of Science, Research and Art and the University of Freiburg in the funding programme Open Access Publishing.

Authors' Contributions

P.V. implemented the `ChiRA` tool suite, integrated into Galaxy, created training material, analyzed the data, and wrote the major portion of the manuscript. A.K. developed the `ChiRAViz` Galaxy visualization and was involved in writing corresponding sections of the manuscript. B.A.G. and O.Z. supported in Galaxy integration and deployment. All authors were involved in reviewing the manuscript.

Acknowledgments

We thank Sita J. Saunders for fruitful discussions and support in writing the biological introduction. We are grateful to Michael Uhl for thorough revision of the manuscript and his constructive comments. The authors also acknowledge the support of the Freiburg Galaxy Team: Prof. Rolf Backofen, Bioinformatics, University of Freiburg, Germany funded by Collaborative Research Centre 992 Medical Epigenetics (DFG grant SFB 992/1 2012) and German Federal Ministry of Education and Research (BMBF grant 031 A538A de.NBI-RBC).

References

1. Ambros V. The functions of animal microRNAs. Nature 2004;**431**(7006):350–5.
2. Henras AK, Dez C, Henry Y. RNA structure and function in C/D and H/ACA s(no)RNPs. Curr Opin Struct Biol 2004;**14**(3):335–43.
3. Bartel DP. MicroRNAs: genomics, biogenesis, mechanism, and function. Cell 2004;**116**(2):281–97.
4. Mattick JS, Makunin IV. Non-coding RNA. Hum Mol Genet 2006;**15**(suppl_1):R17–R29.
5. Plotnikova O, Skoblov M. Efficiency of the miRNA–mRNA interaction prediction programs. Mol Biol 2018;**52**(3): 467–77.
6. Helwak A, Kudla G, Dudnakova T, et al. Mapping the human miRNA interactome by CLASH reveals frequent noncanonical binding. Cell 2013;**153**(3):654–65.
7. Moore MJ, Scheel TKH, Luna JM, et al. miRNA–target chimeras reveal miRNA 3'-end pairing as a major determinant of Argonaute target specificity. Nat Commun 2015;**6**:8864.
8. Lu Z, Zhang QC, Lee B, et al. RNA duplex map in living cells reveals higher-order transcriptome structure. Cell 2016;**165**(5):1267–79.
9. Aw JGA, Shen Y, Wilm A, et al. In vivo mapping of eukaryotic RNA interactomes reveals principles of higher-order organization and regulation. Mol Cell 2016;**62**(4):603–17.
10. Sharma E, Sterne-Weiler T, O'Hanlon D, et al. Global mapping of human RNA-RNA interactions. Mol Cell 2016;**62**(4):618–26.
11. Maher CA, Kumar-Sinha C, Cao X, et al. Transcriptome sequencing to detect gene fusions in cancer. Nature 2009;**458**(7234):97–101.
12. Kannan K, Wang L, Wang J, et al. Recurrent chimeric RNAs enriched in human prostate cancer identified by deep sequencing. Proc Natl Acad Sci U S A 2011;**108**(22):9172–7.
13. Asmann YW, Necela BM, Kalari KR, et al. Detection of redundant fusion transcripts as biomarkers or disease-specific therapeutic targets in breast cancer. Cancer Res 2012;**72**(8):1921–28.
14. Tandefelt DG, Boormans J, Hermans K, et al. ETS fusion genes in prostate cancer. Endocr Relat Cancer 2014;**21**(3):R143–52.
15. Lewis BP, Burge CB, Bartel DP. Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets. Cell 2005;**120**(1):15–20.
16. Esteller M. Non-coding RNAs in human disease. Nat Rev Genet 2011;**12**(12):861–74.
17. Mendell JT, Olson EN. MicroRNAs in stress signaling and human disease. Cell 2012;**148**(6):1172–87.
18. Coolen M, Bally-Cuif L. MicroRNAs in brain development and physiology. Curr Opin Neurobiol 2009;**19**(5):461–70.
19. Pinzón N, Li B, Martinez L, et al. microRNA target prediction programs predict many false positives. Genome Res 2017;**27**(2):234–45.
20. Broughton JP, Pasquinelli AE. A tale of two sequences: microRNA-target chimeric reads. Genet Sel Evol 2016;**48**(1):31.
21. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nat Methods 2012;**9**(4):357.
22. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv 2013:1303.3997.
23. Dobin A, Davis CA, Schlesinger F, et al. STAR: ultrafast universal RNA-seq aligner. Bioinformatics 2013;**29**(1):15–21.
24. Travis AJ, Moody J, Helwak A, et al. Hyb: a bioinformatics pipeline for the analysis of CLASH (crosslinking, ligation and sequencing of hybrids) data. Methods 2014;**65**(3): 263–73.
25. 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.
26. Martin M. Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet J 2011;**17**(1):10–2.
27. Zhong C, Zhang S. Accurate and efficient mapping of the cross-linked microRNA-mRNA duplex reads. iScience 2019;**18**:11–9.
28. Li H, Handsaker B, Wysoker A, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics 2009;**25**(16):2078–9.
29. Langenberger D, Bermudez-Santana C, Hertel J, et al. Evidence for human microRNA-offset RNAs in small RNA sequencing data. Bioinformatics 2009;**25**(18):2298–301.
30. Holmqvist E, Wright PR, Li L, et al. Global RNA recognition patterns of post-transcriptional regulators Hfq and CsrA revealed by UV crosslinking in vivo. EMBO J 2016;**35**(9):991–1011.
31. Robert C, Watson M. Errors in RNA-Seq quantification affect genes of relevance to human disease. Genome Biol 2015;**16**(1):177.
32. Zhang Z, Xing Y. CLIP-seq analysis of multi-mapped reads discovers novel functional RNA regulatory sites in the human transcriptome. Nucleic Acids Res 2017;**45**(16): 9260–71.
33. 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):90.
34. Teng M, Love MI, Davis CA, et al. A benchmark for RNA-seq quantification pipelines. Genome Biol 2016;**17**(1):74.
35. Pachter L. Models for transcript quantification from RNA-Seq. arXiv 2011:1104.3889.
36. Jin H, Wan YW, Liu Z. Comprehensive evaluation of RNA-seq quantification methods for linearity. BMC Bioinformatics 2017;**18**(4):117.
37. Xing Y, Yu T, Wu YN, et al. An expectation-maximization algorithm for probabilistic reconstructions of full-length isoforms from splice graphs. Nucleic Acids Res 2006;**34**(10):3150–60.

38. Mann M, Wright PR, Backofen R. IntaRNA 2.0: enhanced and customizable prediction of RNA–RNA interactions. Nucleic Acids Res 2017;**45**(W1):W435–9.
39. Griffiths-Jones S. miRBase: the microRNA sequence database. In: MicroRNA Protocols. Springer; 2006: 129–38.
40. Agarwal V, Bell GW, Nam JW, et al. Predicting effective microRNA target sites in mammalian mRNAs. Elife 2015;**4**:e05005.
41. Cock PJ, Antao T, Chang JT, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics 2009;**25**(11): 1422–3.
42. Kinsella RJ, Kähäri A, Haider S, et al. Ensembl BioMarts: a hub for data retrieval across taxonomic space. Database (Oxford) 2011;**2011**, doi:10.1093/database/bar030.
43. Grüning B, Dale R, Sjödin A, et al. Bioconda: sustainable and comprehensive software distribution for the life sciences. Nat Methods 2018;**15**(7):475–6.
44. da Veiga Leprevost F, Grüning BA, Alves Aflitos S, et al. BioContainers: an open-source and community-driven framework for software standardization. Bioinformatics 2017;**33**(16):2580–2.
45. Batut B, Hiltemann S, Bagnacani A, et al. Community-driven data analysis training for biology. Cell Syst 2018;**6**(6): 752–8.
46. Fallmann J, Videm P, Bagnacani A, et al. The RNA workbench 2.0: next generation RNA data analysis. Nucleic Acids Res 2019;**47**(W1):W511–5.
47. Videm P. Benchmark data used in the evaluation of ChiRA tool-suite. Zenodo 2020. https://doi.org/10.5281/zenodo.4289365.
48. Videm P, Kumar A, Zharkov O, et al. Supporting data for "ChiRA: an integrated framework for chimeric read analysis from RNA-RNA interactome and RNA structurome data." GigaScience Database; 2020. http://dx.doi.org/10.5524/100845.

[P4] The RNA workbench 2.0: next generation RNA data analysis

Jörg Fallmann, **Pavankumar Videm**, Andrea Bagnacani, Bérénice Batut, Maria A Doyle, Tomas Klingström, Florian Eggenhofer, Peter F Stadler, Rolf Backofen, Björn Grüning. *Nucleic Acids Research*, 2019. DOI: 10.1093/nar/gkz353

Contributions of individual authors:
I have made an important contribution in updating and checking the integrity of the tools and workflows, Galaxy tours, and histories on the webserver. I developed a new training material about small non-coding RNA clustering for the workbench. I also set up the home page for the workbench, prepared figure 1, written parts of the manuscript and revised it. Jörg Fallmann coordinated the efforts to initiate and writing the manuscript. The other co-authors, namely Andrea Bagnacani, Bérénice Batut, Maria A Doyle, Tomas Klingström and Florian Eggenhofer and Björn Grüning have contributed by adding and/or updating the tools, and testing the workflows in the RNA workbench. All the co-authors including Peter F Stadler and Rolf Backofen revised the manuscript.

Pavankumar Videm

The following authors confirm the above stated contributions.

- Jörg Fallmann
- Pavankumar Videm
- Bérénice Batut
- Tomas Klingström
- Peter F Stadler
- Björn Grüning
- Andrea Bagnacani
- Maria A Doyle
- Florian Eggenhofer
- Rolf Backofen

Published online 10 May 2019 *Nucleic Acids Research, 2019, Vol. 47, Web Server issue* ***W511–W515***
doi: 10.1093/nar/gkz353

The RNA workbench 2.0: next generation RNA data analysis

Jörg Fallmann [1,*], Pavankumar Videm [2], Andrea Bagnacani [3], Bérénice Batut [2], Maria A. Doyle [4,5], Tomas Klingstrom [6], Florian Eggenhofer [2], Peter F. Stadler [1,7,8], Rolf Backofen [2,9] and Björn Grüning [2,10,*]

[1]Bioinformatics Group, Department of Computer Science; Leipzig University, Härtelstraße 16-18, D-04107 Leipzig, [2]Bioinformatics Group, Department of Computer Science, Albert-Ludwigs-University Freiburg, Georges-Köhler-Allee 106, Freiburg 79110, Germany, [3]Department of Systems Biology and Bioinformatics, Institute of Computer Science, University of Rostock, Ulmenstr. 69, 18057 Rostock, Germany, [4]Research Computing Facility, Peter MacCallum Cancer Centre, Melbourne, Victoria 3000, Australia, [5]Sir Peter MacCallum Department of Oncology, The University of Melbourne, Victoria 3010, Australia, [6]SLU-Global Bioinformatics Centre, Department of Animal Breeding and Genetics, Swedish University of Agricultural Sciences, [7]Interdisciplinary Center of Bioinformatics; German Centre for Integrative Biodiversity Research (iDiv) Halle-Jena-Leipzig; Competence Center for Scalable Data Services and Solutions; and Leipzig Research Center for Civilization Diseases, Leipzig University, Härtelstraße 16-18, D-04107 Leipzig, [8]Max-Planck-Institute for Mathematics in the Sciences, Inselstraße 22, D-04103 Leipzig Inst. f. Theoretical Chemistry, University of Vienna, Währingerstraße 17, A-1090 Wien, Austria; Facultad de Ciencias, Universidad Nacional de Colombia, Sede Bogotá, Colombia Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM 87501, USA, [9]Signalling Research Centres BIOSS and CIBSS, Albert-Ludwigs-University Freiburg, Schänzlestr. 18, Freiburg 79104, Germany and [10] Center for Biological Systems Analysis (ZBSA), University of Freiburg, Habsburgerstr. 49, 79104 Freiburg, Germany

Received February 08, 2019; Revised April 11, 2019; Editorial Decision April 24, 2019; Accepted April 29, 2019

ABSTRACT

RNA has become one of the major research topics in molecular biology. As a central player in key processes regulating gene expression, RNA is in the focus of many efforts to decipher the pathways that govern the transition of genetic information to a fully functional cell. As more and more researchers join this endeavour, there is a rapidly growing demand for comprehensive collections of tools that cover the diverse layers of RNA-related research. However, increasing amounts of data, from diverse types of experiments, addressing different aspects of biological questions need to be consolidated and integrated into a single framework. Only then is it possible to connect findings from e.g. RNA-Seq experiments and methods for e.g. target predictions. To address these needs, we present the `RNA Workbench 2.0`, an updated online resource for RNA related analysis. With the `RNA Workbench` we created a comprehensive set of analysis tools and workflows that enables researchers to analyze their data without the need for sophisticated command-line skills. This update takes the established framework to the next level, providing not only a containerized infrastructure for analysis, but also a ready-to-use platform for hands-on training, analysis, data exploration, and visualization. The new framework is available at https://rna.usegalaxy.eu , and login is free and open to all users. The containerized version can be found at https://github.com/bgruening/galaxy-rna-workbench.

INTRODUCTION

Together with the focus on RNA as regulatory key player, the number and complexity of datasets ready for analysis is steadily increasing. Although many tools for the analysis of such data exist, they are often tailored to specific experiments and not always easy to install, adapt, and run appropriately. The challenge for the individual researcher remains to chain them into useful workflows and pipelines. Often this task is further complicated, as many tools are only available for the command line, limiting their user base to computer-savvy biologists and bioinformaticians.

Although pitfalls during the installation process of tools can be circumvented with package managers like *conda*

*To whom correspondence should be addressed. Tel: +49 341 97 16667; Fax: +49 341 97-16679; Email: fall@bioinf.uni-leipzig.de
Correspondence may also be addressed to Björn Grüning. Email: bjoern.gruening@gmail.com

https://conda.io and its *BioConda* (1) channel, or Docker containers, it remains with the user to set up the appropriate computational environment. Many of these needs were already addressed with the release of the *RNA Workbench* (2). Based on the framework (3), containerized in a *Docker* instance, the workbench guarantees simple access, easy extension and flexible adaption to personal and security needs. This enables users to run sophisticated analyses that are independent of command-line knowledge while utilizing's integrated and powerful workflow manager. With the current release of the *RNA Workbench 2.0* we now additionally provide the user with a pre-configured, ready-to-use compute environment, running on dedicated hardware, available at https://rna.usegalaxy.eu.

The *RNA Workbench 2.0* is developed and maintained by a community consisting of experts in RNA bioinformatics and, as well as a growing number of users, and tool developers. Our commitment to keep the workbench fit for future standards and needs is one of the reasons for the release of this update. We aim to provide researchers with an up-to-date reliable and robust framework for RNA data analysis. In this release, we integrated many new RNA-related tools, and updated well established suites, such as the *ViennaRNA* (4) package, covering a broad variety of use-cases.

Currently, we provide more than 100 bioinformatics tools that are dedicated to different research areas of RNA biology including RNA structure analysis, RNA alignment, RNA annotation, RNA-protein interaction, ribosome profiling, RNA-Seq pre-processing and analysis, as well as RNA target prediction. The complete list of tools can be found at https://rna.usegalaxy.eu or https://github.com/bgruening/galaxy-rna-workbench.

Taking advantage of Galaxy 's powerful workflow manager allows users to easily connect single tools into computational pipelines. For common RNA related tasks we provide >25 ready-to-use workflows combining, e.g. established tools for RNA-Seq processing and analysis. For each workflow we provide a dedicated training to guide researchers through the analysis. Training is a key aspect of our effort in bringing high-quality RNA bioinformatics to researchers. Thus, each training accompanying a workflow comes with a test dataset, allowing interested users to get hands-on experience with their tools and workflows of interest. Keeping such trainings up-to-date and functional is a cooperative endeavour together with the *Galaxy* Training Network , which hosts *Galaxy* Training Material (5), a collection of tutorials developed and maintained by the worldwide *Galaxy* community. In case a user requires a novel workflow to answer a research question that is not covered by existing ones or to incorporate specific tools, we encourage users to share these workflows and if possible adequate training data and material. This directly enables all users to benefit from contributions to our community, which distributes shared knowledge and in return helps to maintain and enhance workflows and trainings where possible.

GOALS

A main intention behind the development of the original RNA Workbench was the creation of an easy-to-use and deploy environment for training and self-empowerment of biologists in RNA bioinformatics. The RNA Workbench was downloaded >2000 times, used for research, training courses (e.g. within de.NBI (6)), and has even been integrated into the B3Africa toolset (7). The ongoing need for such a comprehensive collection of RNA bioinformatics tools, workflows and resources led to the development of RNA Workbench 2.0. Although the provision of RNA Workbench as in a *Docker* made it easy to maintain, deploy and use, we became aware that there is additional need for an instance with freely available compute resources. Our target audience, mainly RNA biologists, requested an even more easy-to-use and ready-to-go way of accessing this collection. With the realization of the European server (https://usegalaxy.eu), we gained access to an infrastructure that would allow exactly that. Thus, with RNA Workbench 2.0 we provide an updated and ready-to-use webserver, satisfying user requests and enabling even more scientists to participate in RNA research.

TOOLS AND IMPROVEMENTS

In addition to providing the *RNA Workbench 2.0* as a portable *Docker* container (https://github.com/bgruening/galaxy-rna-workbench), users can now directly use integrated tools, workflows and tutorials at a free online instance of *Galaxy*. This makes the use of the *RNA Workbench 2.0* even easier, and allows users to train and run data analysis workflows without the need to set up hardware, software environments, or even *Docker*. New workflows and tutorials ease introduction to the environment, and guide users through analysis tasks step by step. Continuous exchange of workflows, tours and training material with the *Galaxy* Training Network ensures that the *RNA Workbench 2.0* remains a state-of-the-art training and research resource. New and updated tools and workflows are continuously integrated and made available in close cooperation between the user and developer community. This includes also updates to the underlying packages in *BioConda*. Updated tools are for example *LocaRNA* (8), *RNAz* (9), (10), *AREsite2* (11) and *Infernal* (12). In addition new tools like *edgeR* (13), *CMV* (14), *RNAlien* (15), *MultiQC* (16) and *scPipe* (17) have been added. A complete list of available tools can be found at https://rna.usegalaxy.eu.

Figure 1 provides an overview of tools and workflows dedicated to specific topics of RNA research in version 2 of the *RNA Workbench*.

TRAINING

A key aspect behind the development of the original and now updated *RNA Workbench* was to provide an accessible platform, easing the process of gaining expertise in and applying bioinformatics. To this end, considerable effort went into extensive documentation and a large set of training material, empowering beginners and non-bioinformaticians to use, adapt, and apply workflows based on their needs and standards. The recently published *Galaxy* Training Material provides users with a collection of hands-on training material and data on many top-

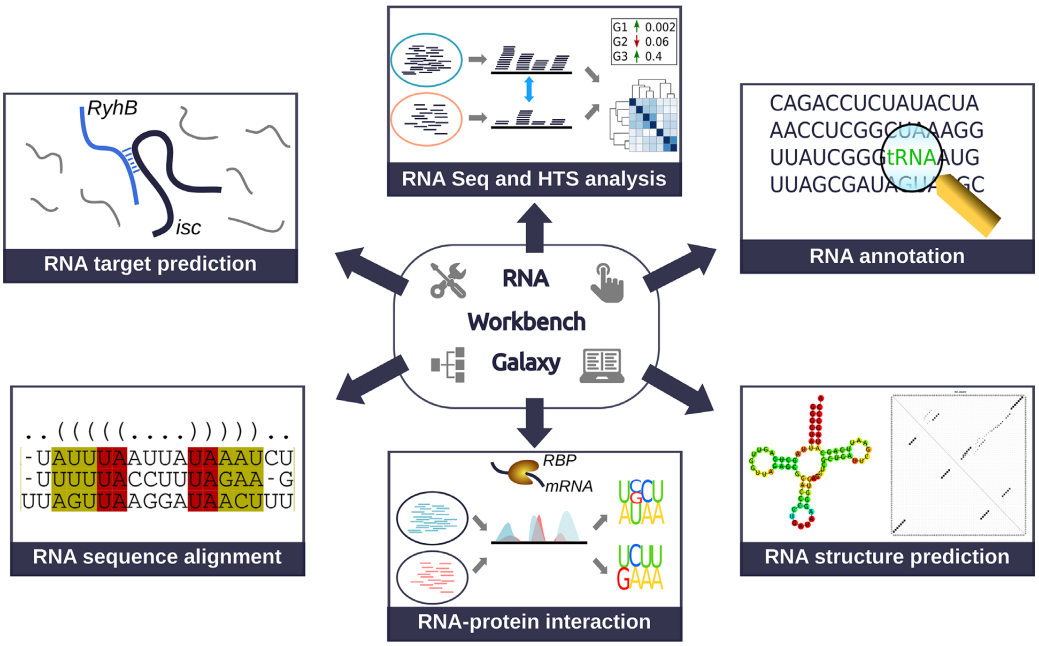


Figure 1. Overview of RNA research topics, dedicated tools and example workflows in *RNA Workbench 2.0* RNA target prediction enables to analyze potential interaction partners of RNA molecules. Included annotation tools allow the discovery of homologous sequences in genomes. The secondary structure of input RNA sequences can be predicted and visualized or for example used to create sequence-structure alignments. High-throughput and RNA sequencing data analysis can be performed with available tools and results directly intersected with *e.g.* databases for RNA-protein interactions.

ics of (not exclusively high-throughput sequencing (HTS)-related) life-science research. This collection is constantly improved and extended in an international community effort, including de.NBI, ELIXIR and EMBL. We tightly integrate *Galaxy* Training Material into the *RNA Workbench 2.0*, exchanging workflows and training material on various RNA related topics. As an example, for RNA-Seq data analyses we provide training instances as specific introduction to the topic. These consist of self-explanatory presentation slides, hands-on training documentation and a *Galaxy* Interactive Tour guiding through the analysis workflow with all required input files ready-to-use, hosted by *Zenodo*.

WORKFLOWS

One of the strengths of the framework is that users can easily create, customize and share their workflows with other users of the same or other instances. A workflow is not only a chain of tools applied to a fixed dataset, *Galaxy* workflows also save tool versions, required data formats and other metadata ensuring a maximum of reproducibility. The built-in graphical workflow editor facilitates repurposing or adaptation of workflows.

A set of >25 workflows dedicated to specific analysis goals is included in *RNA Workbench 2.0*. We provide for example a set of workflows for the analysis of non-coding RNA and cover a range of analysis tasks, from structure conservation and coding potential of homologous RNAs, based on *Locarna* (8) and *RNAz* (9), as well as automatic construction of RNA family models, based on *RNAlien* (14). The workbench features workflows for processing, analyzing and visualizing data from RNA-Seq, CLIP-Seq, RNA folding, network analysis, sRNA-Seq, RNA family model construction and more.

Datasets for analysis can be imported from a local source, from dedicated databases or via link, easing the integration of data from different sources. Training datasets can be imported directly from *Zenodo*.

TOURS

Another training aspect is provided via *Galaxy* Interactive Tours. These guide users through an entire analysis in an interactive and explorative way. In contrast to training videos, a *Galaxy* Interactive Tour can be easily created, updated and improved to guide the *Galaxy* user step-by-step, e.g. through a whole HTS analysis starting from uploading the data to using complex analysis tools. The *RNA workbench* currently integrates more than 15 *Galaxy* Interactive Tours. These range from general tours introducing new users to the *Galaxy* interface and its usage, with RNA-seq example datasets, to specialized tours, *e.g.* illustrating secondary structure prediction of RNA molecules using parts of the *ViennaRNA* package.

INPUTS AND OUTPUTS

Users of the *RNA Workbench 2.0* have access to a diverse set of *Galaxy* implemented data formats and format conversion tools. Common formats for sequence and/or structure information are readily accepted as input, generic data can be imported and converted to fit tool specifications, guaranteeing reproducibility and interoperability. Output data follows the same principle, defined by the analysis tool, but can be converted to a range of standard and

specific formats, including plots and figures. For the latter, the RNA Workbench 2.0 contains tools for visualizations of RNA-Seq related data (e.g. *mQC* (18), *MultiQC* (16), *sRNAPipe* (19)), RNA structure datasets, such as dot-bracket strings RNA 2D or 3D structures or RNA family models and alignments (cmv (15)).

COMMUNITY CONTRIBUTIONS

The RNA Workbench 2.0 is hosted on *GitHub* (https://github.com/bgruening/galaxy-rna-workbench) and users are welcome to suggest new tools, workflows and tours to be made available through *GitHub* and the workbench *Docker* container. Tools should be published to the *Galaxy* Tool Shed (20) via https://github.com/bgruening/galaxytools followed by a pull request at *GitHub*. After passing continuous integration tests and approval after manual review, new tools will be integrated into the RNA Workbench. More information about tool development can be found on the *Galaxy* community page. Workflows can easily be contributed by running them at https://rna.usegalaxy.eu and sharing them, ideally accompanied by test datasets and a shared history of the workflow run. A pull request adding them to the workflow folder of https://github.com/bgruening/galaxy-rna-workbench, will allow us to merge the workflow into the workbench. When contributing workflows, users should make sure that all tools needed for the workflow are integrated into the RNA Workbench 2.0. If not, please add these tools beforehand following above steps, or request them to be added by opening an appropriate issue at *GitHub*. *Galaxy* interactive tours can be contributed similarly, by opening a pull request and including tours in the tours folder of https://github.com/bgruening/galaxy-rna-workbench after approval.

USE CASES

de.NBI

The 'German Network for Bioinformatics Infrastructure–(de.NBI)' is an academic and non-profit infrastructure supported by the German Federal Ministry of Education and Research. As German partner of ELIXIR (https://www.elixir-europe.org) it provides bioinformatics services to users in life science research and biomedicine in Europe (6). The partners organize training events, courses and summer schools on tools, standards and compute services provided by de.NBI and ELIXIR to assist researchers to more effectively exploit their data. The RNA Workbench and also RNA Workbench 2.0 have in part been developed by researchers funded by de.NBI with the aim to generate a free and easy to use platform for training and education. As such, the RNA Workbench is ready for and has been used in de.NBI training courses. With the publication of RNA Workbench 2.0 this will become even easier, as trainers and trainees have access to a ready-to-use instance, including dedicated hardware, simply connecting via a web browser.

B3Africa

The Bridging Biomolecular Researcher and Biobanking in Africa (B3Africa) created the *eB3Kit*, an informatics platform for comprehensive management of samples and associated data (21) to support the establishment of research integrated biobanks (22). A key priority of the project is to strengthen the research capacity in resource constrained areas. As bioinformatics is a rapidly advancing field leading to constant changes in the demand for tools and procedures, the bioinformatics module has been designed to integrate a pre-existing platform satisfying the following key requirements. (i) An active community providing access to new tools, algorithms and training through a standardized interface, (ii) an accessible API enabling the B3Africa project to interact with the software without changing the supported codebase and (iii) the ability to download tools and databases for access without internet connection. Fulfilling these requirements, the RNA Workbench has been implemented in the *BIBBOX* appstore (7) and is used as the preferred solution to showcase both the *eB3Kit* and the *Galaksio* interface for simplified workflow management (23). Throughout the project successful showcases of the *eB3Kit* using the RNA Workbench have been conducted, e.g. in Lyon (France), Banjul (Gambia) and at Lake Naivasha in the Rift Valley region of Kenya.

DISCUSSION

An active community developing and applying the RNA workbench in training (e.g. within de.NBI, ELIXIR and B3Africa) and research, the RNA Workbench has become an important resource for best practices in RNA and high-throughput sequencing bioinformatics in *Galaxy*.

In this work, we present an update to this resource with the creation of the ready-to-use webserver instance. Users benefit from this setup as they can now directly browse to https://rna.usegalaxy.eu and use a pre-configured instance of RNA Workbench, without needing to have any software installed on their own system except for a browser. This enables researchers not only to become familiar with a set of RNA-related bioinformatics tasks, running one of the provided tutorials and/or accompanying workflows, but also to compute and analyze data on dedicated hardware. For users concerned with data regulations, e.g. when working on patient data, or users with their own dedicated hardware, we also provide an updated Docker container, similar to the first version of RNA Workbench. A RNA Workbench instance started with this container provides the same tools, workflows, trainings and tours as the online instance and can easily be extended with additional tools via the *Galaxy* Tool Shed. As for the first version of RNA Workbench, each tool in the workbench is also available as a *BioConda* package as well as a *Docker*/*rkt* container (*BioContainers*). The Docker container offers a comprehensive virtualized *RNA workbench* that can be deployed on every standard Linux, Windows and OSX computer, but can at the same time employ high-performance- or cloud-computing infrastructure.

Similar to the first version, this release is developed and maintained by a constantly growing RNA and Galaxy community. This community approach helps to keep the workbench up-to-date and valuable for research. Moreover, all components such as tools, workflows, visualizations, interactive tours and training material can be easily integrated

into any available *Galaxy* instance for teaching, learning or exploratory purposes. Every user is encouraged to contribute and add to this collection, which is tightly integrated into the *Galaxy* Training Material , providing state-of-the-art learning material.

To our knowledge, the *RNA workbench* is a unique suite without direct competitors. Existing workbenches, such as `miARma-Seq` (24), the `UEA Small RNA Workbench` (25) or the `NCBI genome workbench`, are all tailored to specific analysis tasks. In addition, our focus on accessibly, flexibility in workflow assembly and application, training and the interaction with the community are all major benefits of `RNA Workbench 2.0`.

ACKNOWLEDGEMENTS

We thank de.NBI and ELIXIR for supporting bioinformatics infrastructure. Thanks also to the *Galax*y community, especially to the Freiburg *Galaxy* Team, for developing, maintaining and supporting this great framework. We also like to acknowledge the *BioConda* and *BioContainers* community for setting new standards in reproducible software deployments. Furthermore, the authors acknowledge the support of many upstream developers like Chao Zhang, Steven Verbruggen, the *sRNAPipe* team Pierre Pouchin, Silke Jensen and Emilie Brasset that helped us to integrate their tools into the `RNA Workbench 2.0` and accepted patches and to whom we wish a wonderful day, every day.

FUNDING

German Federal Ministry of Education and Research [BMBF grants 031 A538A/A538C de.NBI-RBC awarded to P.F.S. and R.B., 031L0101C de.NBI-epi awarded to B.G., 031L0106 de.STAIR (de.NBI)]; German Research Foundation for the Collaborative Research Center 992 Medical Epigenetics [SFB 992/1 2012 and SFB 992/2 2016 awarded to R.B.]. Funding for open access charge: German Federal Ministry of Education and Research.
Conflict of interest statement. None declared.

REFERENCES

1. Grüning,B., Dale,R., Sjödin,A., Chapman,B.A., Rowe,J., Tomkins-Tinch,C.H., Valieris,R., Köster,J. and Bioconda,T. (2018) Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nat. Methods*, **15**, 475.
2. Grüning,B.A., Fallmann,J., Yusuf,D., Will,S., Erxleben,A., Eggenhofer,F., Houwaart,T., Batut,B., Videm,P., Bagnacani,A. *et al.* (2017) The RNA workbench: best practices for RNA and high-throughput sequencing bioinformatics in Galaxy. *Nucleic Acids Res.*, **45**, W560–W566.
3. Afgan,E., Baker,D., Batut,B., Van Den Beek,M., Bouvier,D., Ech,M., Chilton,J., Clements,D., Coraor,N., Grüning,B. *et al.* (2018) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic Acids Res.*, **46**, W537–W544.
4. Lorenz,R., Bernhart,S.H., Zu Siederdissen,C.H., Tafer,H., Flamm,C., Stadler,P.F. and Hofacker,I.L. (2011) ViennaRNA Package 2.0. *Alg. Mol. Biol.*, **6**, 26.
5. Batut,B., Hiltemann,S., Bagnacani,A., Baker,D., Bhardwaj,V., Blank,C., Bretaudeau,A., Brillet-Gueguen,L., čech,M., Chilton,J. *et al.* (2018) Community-Driven Data Analysis Training for Biology. *Cell Syst.*, **6**, 752–758.
6. Tauch,A. and Al-Dilaimi,A. (2019) Bioinformatics in Germany: toward a national-level infrastructure. *Brief. Bioinform.*, **20**, 370–374.
7. Müller,H., Malservet,N., Quinlan,P., Reihs,R., Penicaud,M., Chami,A., Zatloukal,K. and Dagher,G. (2017) From the evaluation of existing solutions to an all-inclusive package for biobanks. *Health Technol.*, **7**, 89–95.
8. Will,S., Joshi,T., Hofacker,I.L., Stadler,P.F. and Backofen,R. (2012) LocARNA-P: accurate boundary prediction and improved detection of structural RNAs. *RNA*, **18**, 900–914.
9. Gruber,A.R., Findeiß,S., Washietl,S., Hofacker,I.L. and Stadler,P.F. (2010) Rnaz 2.0: improved noncoding RNA detection. *Pac. Symp. Biocomput.*, **2010**, 69–79.
10. Videm,P., Rose,D., Costa,F. and Backofen,R. (2014) BlockClust: Efficient Clustering and Classification of Non-Coding RNAs from Short Read RNA-Seq Profiles. *Bioinform.*, **30**, i274–i282.
11. Fallmann,J., Sedlyarov,V., Tanzer,A., Kovarik,P. and Hofacker,I.L. (2016) AREsite2: an enhanced database for the comprehensive investigation of AU/GU/U-rich elements. *Nucleic Acids Res.*, **44**, D90–D95.
12. Nawrocki,E.P. and Eddy,S.R. (2013) Infernal 1.1: 100-fold faster RNA homology searches. *Bioinform.*, **29**, 2933–2935.
13. Robinson,M.D., McCarthy,D.J. and Smyth,G.K. (2010) edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinform.*, **26**, 139–140.
14. Eggenhofer,F., Hofacker,I.L. and zuSiederdissen,C.H. (2016) RNAlien-unsupervised RNA family model construction. *Nucleic Acids Res.*, **44**, 8433.
15. Eggenhofer,F., Hofacker,I.L. and Backofen,R. (2018) CMVVisualization for RNA and protein family models and their comparisons. *Bioinform.*, **1**, 3.
16. Ewels,P., Magnusson,M., Lundin,S. and Kaller,M. (2016) MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinform.*, **32**, 3047–3048.
17. Tian,L., Su,S., Dong,X., Amann-Zalcenstein,D., Biben,C., Seidi,A., Hilton,D.J., Naik,S.H. and Ritchie,M.E. (2018) scPipe: a flexible R/bioconductor preprocessing pipeline for single-cell RNA-sequencing data. *PLoS Comput. Biol.*, **14**, e1006361.
18. Verbruggen,S. and Menschaert,G. (2018) mQC: a post-mapping data exploration tool for ribosome profiling. *Comput. Methods Programs Biomed*, doi:10.1016/j.cmpb.2018.10.018.
19. Pogorelcnik,R., Vaury,C., Pouchin,P., Jensen,S. and Brasset,E. (2018) sRNAPipe: a Galaxy-based pipeline for bioinformatic in-depth exploration of small RNAseq data. *Mob DNA*, **9**, 25.
20. Blankenberg,D., Von Kuster,G., Bouvier,E., Baker,D., Afgan,E., Stoler,N., Taylor,J. and Nekrutenko,A. (2014) Dissemination of scientific software with Galaxy ToolShed. *Genome Biol.*, **15**, 403.
21. Klingström,T., Mendy,M., Meunier,D., Berger,A., Reichel,J., Christoffels,A., Bendou,H., Swanepoel,C., Smit,L., Mckellar-Basset,C. *et al.* (2016) Supporting the development of biobanks in low and medium income countries. In: *IST-Africa Week Conference*. IEEE, Durban, pp. 1–10.
22. Slokenberga,S., Reichel,J., Niringiye,R., Croxton,T., Swanepoel,C. and Okal,J. (2018) EU data transfer rules and African legal realities: is data exchange for biobank research realistic?. *Data Privacy Law Int.*, **9**, 30–48.
23. Klingström,T., Hernández-deDiego,R., Collard,T. and Bongcam-Rudloff,E. (2017) Galaksio, a user friendly workflow-centric front end for Galaxy. *EMBnet. J.*, **23**, e897.
24. Andres-Leon,E., Nunez-Torres,R. and Rojas,A.M. (2016) miARma-Seq: a comprehensive tool for miRNA, mRNA and circRNA analysis. *Sci. Rep.*, **6**, 25749.
25. Stocks,M.B., Mohorianu,I., Beckers,M., Paicu,C., Moxon,S., Thody,J., Dalmay,T. and Moulton,V. (2018) The UEA sRNA Workbench (version 4.4): a comprehensive suite of tools for analyzing miRNAs and sRNAs. *Bioinform.*, **34**, 3382–3384.

Appendix – Supplementary material

Supplementary material for

[P1] BlockClust: efficient clustering and classification of non-coding RNAs from short read RNA-seq profiles

S SUPPLEMENT

S.1 Graph Kernel

S.1.1 Graph Definitions and Notation A graph $G = (V, E)$ consists of two sets V and E. The notation $V(G)$ and $E(G)$ is used when G is not the only graph considered. The elements of V are called *vertices* and the elements of E are called *edges*. Each edge has a set of two elements in V associated with it, which are called its *endpoints*, which we denote by concatenating the vertices variables, e.g. we represent the edge between the vertices u and v with uv. An edge is said to *join* its endpoints. A vertex v is *adjacent* to a vertex u if they are joined by an edge. An edge and a vertex on that edge are called *incident*. The *degree* of a vertex is number of edges incident to it. A *multi-edge* is a collection of two or more edges having identical endpoints. A *self-loop* is an edge that joins a single endpoint to itself. A *simple graph* is a graph that has no self-loops nor multi-edges. In this work we consider only simple graphs. A graph is *complete* if every pair of vertices is joined by an edge. A graph is *rooted* when we distinguish one of its vertices, called *root*; we denote a rooted graph G with root vertex v with G^v. A *walk* in a graph G is a sequence of vertices $W = v_0, v_1, \ldots, v_n$ such that for $j = 1, \ldots, n$, the vertices v_{j-1} and v_j are adjacent. The *length* of a walk is the number of edges (counting repetitions). A *path* is a walk such that no vertex is repeated, except at most the initial (v_0) and the final (v_n) vertex (in this case it is called a *cycle*). The *distance* between two vertices, denoted $\mathcal{D}(u, v)$, is the length of the shortest path between them. A graph is *connected* if between each pair of vertices there exist a walk. In this work we consider only connected graphs. We denote the class of simple connected graphs with $\mathcal{G}$. The *neighborhood* of a vertex v is the set of vertices that are adjacent to v and is indicated with $N(v)$. The *neighborhood* of radius r of a vertex v is the set of vertices at a distance less than or equal to r from v and is denoted by $N_r(v)$. In a graph G, the *induced-subgraph* on a set of vertices $W = \{w_1, \ldots, w_k\}$ is a graph that has W as its vertex set and it contains every edge of G whose endpoints are in W. A subgraph H is a *spanning* subgraph of a graph G if $V(H) = V(G)$. The *neighborhood subgraph* of radius r of vertex v is the subgraph induced by the neighborhood of radius r of v and is denoted by $\mathcal{N}_r^v$. A *labeled graph* is a graph whose vertices and/or edges are labeled, possibly with repetitions, using symbols from a finite alphabet. We denote the function that maps the vertex/edge to the label symbol as $\mathcal{L}$. Two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are *isomorphic*, which we denote by $G_1 \simeq G_2$, if there is a bijection $\phi : V_1 \to V_2$, such that for any two vertices $u, v \in V_1$, there is an edge uv if and only if there is an edge $\phi(u)\phi(v)$ in G_2. An isomorphism is a structure-preserving bijection. Two labeled graphs are isomorphic if there is an isomorphism that preserves also the label information, i.e. $\mathcal{L}(\phi(v)) = \mathcal{L}(v)$. An *isomorphism invariant* or *graph invariant* is a graph property that is identical for two isomorphic graphs (e.g. the number of vertices and/or edges). A *certificate for isomorphism* is an isomorphism invariant that is identical for two graphs if and only if they are isomorphic.

S.1.2 Kernel Definition and Notation Given a set X and a function $K : X \times X \to \mathbb{R}$, we say that K is a *kernel on* $X \times X$ if K is symmetric, i.e. if for any x and $y \in X$ $K(x, y) = K(y, x)$, and if K is *positive-semidefinite*, i.e. if for any $N \geq 1$ and any $x_1, \ldots, x_N \in X$, the matrix K defined by $K_{ij} = K(x_i, x_j)$ is positive-semidefinite, that is $\sum_{ij} c_i c_j K_{ij} \geq 0$ for all $c_1, \ldots, c_N \in \mathbb{R}$ or equivalently if all its eigenvalues are nonnegative. It is easy to see that if each $x \in X$ can be represented as $\phi(x) = \{\phi_n(x)\}_{n \geq 1}$ such that K is the ordinary l_2 dot product $K(x, y) = \langle \phi(x), \phi(y) \rangle = \sum_n \phi_n(x)\phi_n(y)$ then K is a kernel. The converse is also true under reasonable assumptions (which are almost always verified) on X and K, that is, a given kernel K can be represented as $K(x, y) = \langle \phi(x), \phi(y) \rangle$ for some choice of ϕ. In particular it holds for any kernel K over $X \times X$ where X is a countable set. The vector space induced by ϕ is called the *feature space*. Note that it follows from the definition of positive-semidefinite that the *zero-extension* of a kernel is a valid kernel, that is, if $S \subseteq X$ and K is a kernel on $S \times S$ then K may be extended to be a kernel on $X \times X$ by defining $K(x, y) = 0$ if x or y is not in S. It is easy to show that kernels are closed under summation, i.e. a sum of kernels is a valid kernel.

Let now $x \in X$ be a *composite structure* such that we can define $x_1, \ldots, x_D$ as its parts[5]. Each part is such that $x_d \in X_d$ for $d = 1, \ldots, D$ with $D \geq 1$ where each X_d is a countable set. Let R be the relation defined on the set $X_1 \times \ldots \times X_D \times X$, such that $R(x_1, \ldots, x_D, x)$ is true iff $x_1, \ldots, x_D$ are the parts of x. We denote with $R^{-1}(x)$ the inverse relation that yields the parts of x, that is $R^{-1}(x) = \{x_1, \ldots, x_D : R(x_1, \ldots, x_D, x)\}$. In Haussler (1999) it is demonstrated that, if there exist a kernel K_d over $X_d \times X_d$ for each $d = 1, \ldots, D$, and if two instances $x, y \in X$ can be decomposed in $x_1, \ldots, x_d$ and $y_1, \ldots, y_d$, then the following generalized convolution:

$$K(x, y) = \sum_{\substack{x_1, \ldots, x_m \in R^{-1}(x) \\ y_1, \ldots, y_m \in R^{-1}(y)}} \prod_{m=1}^{M} K_m(x_m, y_m)$$

is a valid kernel called a *convolution* or *decomposition* kernel[6]. In words: a decomposition kernel is a sum (over all possible ways to decompose a structured instance) of the product of valid kernels over the parts of the instance.

S.1.3 The Neighborhood Subgraph Pairwise Distance Kernel Given the notation introduced in the previous sections, in the following we define the Neighborhood Subgraph Pairwise Distance Kernel (NSPDK) as an instance of a decomposition kernel.

We define the relation $R_{r,d}(A^v, B^u, G)$ between two rooted graphs A^v, B^u and a graph G to be true iff both A^v and B^u are in $\{\mathcal{N}_r^v : v \in V(G)\}$, where we require that A^v (B^u) be isomorphic to some $\mathcal{N}_r$ to verify the set inclusion, and that $\mathcal{D}(u, v) = d$. In words: the relation $R_{r,d}$ selects all pairs of neighborhood graphs of radius r whose roots are at distance d in a given graph G.

We define $\kappa_{r,d}$ over $\mathcal{G} \times \mathcal{G}$ as the decomposition kernel on the relation $R_{r,d}$, that is:

$$\kappa_{r,d}(G, G') = \sum_{\substack{A_v, B_u \in R_{r,d}^{-1}(G) \\ A'_{v'}, B'_{u'} \in R_{r,d}^{-1}(G')}} \delta(A_v, A'_{v'})\delta(B_u, B'_{u'})$$

[5] Note that the set of parts needs not be a partition for the composite structure, i.e. the parts may "overlap".

[6] To be precise, the valid kernel is the zero-extension of K to $X \times X$ since $R^{-1}(x)$ is not guaranteed to yield a non-empty set for all $x \in X$.

where the *exact matching kernel* $\delta(x, y)$ is 1 if $x \simeq y$ (i.e. if the graph x is isomorphic to y) and 0 otherwise. In words: $\kappa_{r,d}$ counts the number of identical pairs of neighboring graphs of radius r at distance d between two graphs.

The Neighborhood Subgraph Pairwise Distance Kernel is finally defined as:

$$K(G, G') = \sum_r \sum_d \kappa_{r,d}(G, G').$$

For efficiency reasons however, in this work we consider the zero-extension of K obtained by imposing an upper bound on the radius and the distance parameter: $K_{r^*,d^*}(G, G') = \sum_{r=0}^{r^*} \sum_{d=0}^{d^*} \kappa_{r,d}(G, G')$, that is, we are limiting NSPDK to the sum of the $\kappa_{r,d}$ kernels for all increasing values of the radius (distance) parameter up to a maximum given value r^* (d^*). Furthermore we consider a normalized version of $\kappa_{r,d}$, that is: $\hat{\kappa}_{r,d}(G, G') = \frac{\kappa_{r,d}(G,G')}{\sqrt{\kappa_{r,d}(G,G)\kappa_{r,d}(G',G')}}$, to ensure that relations of all orders are equally weighted regardless of the size of the induced part sets[7].

Finally, it is easy to show that the Neighborhood Subgraph Pairwise Distance Kernel is a valid kernel as: 1) it is built as a decomposition kernel over the countable space of all pairs of neighborhood subgraphs of graphs of finite size; 2) the kernel over parts (the exact matching kernel) is a valid kernel; 3) the zero-extension to bounded values for the radius and distance parameters preserves the kernel property; and 4) so does the normalization step.

S.1.4 Graph Invariant The NSPDK includes an exact matching kernel over two graphs which is equivalent to solving the graph isomorphism problem (ISO). Since the existence of (deterministic) polynomial algorithms for ISO is still an open problem, we have to resort to one of two strategies: 1) limit the class of graphs under consideration and solve ISO exactly; or 2) give an approximate (fast) solution of ISO on general graphs. Here we opt for the latter solution since we are mainly concerned with application domains where the number of graphs to be processed are in the range of tens to hundreds of thousands and application specific pre-processing might alter the class of the input graphs (making them non-outer planar for example).

In this work we implement the exact matching kernel $\delta(G_h, G'_{h'})$ in two steps: 1) we compute a fast graph invariant encoding for G_h and $G'_{h'}$ via a label function $\mathcal{L}^g : \mathcal{G}_h \to \Sigma^*$, where $\mathcal{G}_h$ is the set of rooted graphs and Σ^* is the set of strings over a finite alphabet Σ; 2) we make use of a hash function $H : \Sigma^* \to \mathbb{N}$ to confront $H(\mathcal{L}^g(G_h))$ and $H(\mathcal{L}^g(G'_{h'}))$. In words: we produce an efficient string encoding of graphs from which we obtain a unique identifier via a hashing function from strings to natural numbers. In this way the isomorphism test between two graphs is reduced to a fast numerical identity test. Note that we cannot hope to exhibit an efficient certificate for isomorphism in this way, but only an efficient graph invariant at most, i.e. there will be cases where two non-isomorphic graphs are assigned the same identifier.

The graph encoding $\mathcal{L}^g(G_h)$ that we propose is best described by introducing new label functions for vertices and edges, denoted $\mathcal{L}^n$ and $\mathcal{L}^e$ respectively. $\mathcal{L}^n(v)$ assigns to vertex v the concatenation of the lexicographically sorted listed of distance-label pairs $\langle \mathcal{D}(v, u), \mathcal{L}(u) \rangle$ for all $u \in G_h$. Since G_h is a rooted graph we can exploit the knowledge about the identity of the root vertex h and include, for each vertex v, the additional information of the distance from the root node $\mathcal{D}(v, h)$. $\mathcal{L}^e(uv)$ assigns to edge uv the label $\langle \mathcal{L}^n(u), \mathcal{L}^n(v), \mathcal{L}(uv) \rangle$. $\mathcal{L}^g(G_h)$ assigns to the rooted graph G_h the concatenation of the lexicographically sorted list of $\mathcal{L}^e(uv)$ for all $uv \in E(G_h)$. In words: we relabel each vertex with a string that encodes the vertex distance from all other labeled vertices (plus the distance from the root vertex); the graph encoding is obtained as the sorted edge list, where each edge is annotated with the endpoints' new labels.

We finally resort to a Merkle-Damgård construction based hashing function for variable-length data to map the graph encoding string to a 32-bit integer.

S.1.5 Kernel Algorithmic Complexity The time complexity of the NSPDK depends on two key procedures: 1) the extraction of all pairs of neighborhood graphs $\mathcal{N}_r^v$ at distance $d = 0, \ldots, d^*$, and 2) the computation of the graph invariant for those subgraphs. The first procedure can be efficiently implemented by factoring it into a) the extraction of $\mathcal{N}_r^v$ for all $v \in V(G)$ and b) the computation of distances between pairs of vertices whose pairwise distance is less than d^*. For this latter step we can repeat a breadth-first (BF) visit up to distance d^* for each vertex in $O(|V(G)||E(G)|)$. Note that, on graphs with bounded (low) degree, the complexity is more realistically modeled as a linear function of $|V(G)|$ since a small d^* implies, in practice, that each bounded BF visit can be performed in constant time. The complexity of point a) is linear in the number of edges in the neighborhood (constant in practice for small r). Finally, the complexity of point 2) (the computation of the graph invariant for neighborhood graphs) can be analyzed in terms of i) the computation of the string encoding $\mathcal{L}^g(G_h)$ and ii) the computation of the hash function $H(\mathcal{L}^g(G_h))$. Part i) is dominated by the computation of all pairwise distances in $O(|V(G_h)||E(G_h)|)$ and the sorting of the relabeled edges, which has complexity $O(|V(G_h)||E(G_h)| \log |E(G_h)|)$ since edges are relabeled with strings containing the distance information of the endpoints from all other vertices. The hash function complexity (part ii)) is linear in the size of the string. We conclude that the overall complexity $O(|V(G)||V(G_h)||E(G_h)| \log |E(G_h)|)$ is dominated by the repeated computation of the graph invariant for each vertex of the graph. Since this is a constant time procedure for small values of d^* and r^*, we conclude that the NSPDK complexity is in practice linear in the size of the graph.

Note finally that, to reduce space complexity, we do not manage the hash collisions, as this would force the algorithm to keep in memory all the encoding key - hashed value pairs.

S.2 Efficient Neighborhood graph extraction using Locality Sensitive Hashing

As datasets size increases, algorithms that directly make use of pairwise distance or similarity information become infeasible as they inevitably exhibit a quadratic complexity. The key idea then is to formulate the clustering problem in terms of approximate nearest neighbors queries which can be answered efficiently (sub-linearly). That is, given a set of n instances $P = \{p_1, \ldots, p_n\}$ in a metric space X with a distance function d, a neighborhood query is a

[7] As the number of neighborhood graphs grows exponentially with the radius, large (infrequent) subgraphs tend to dominate the kernel value with negative effects on the generalization performance of predictive systems.

procedure that returns the instance in P closest to a query instance $q \in X$. The *nearest neighbor search problem* is formulated as a dataset pre-processing that allows nearest neighbors queries to be answered efficiently. The key idea is to relax the requirements, ask for $\epsilon-$approximate nearest neighbor queries, and use *locality-sensitive hashing* techniques. The $\epsilon-$approximate nearest neighbor query returns an instance p for a given query q such that $\forall p' \in P, d(p,q) \leq (1+\epsilon)d(p',q)$. A locality-sensitive hash function is a hash function such that the probability of collision is higher for objects that are close to each other than for those that are far apart. As locality-sensitive hash function we choose the min-hash function Broder (1997) as it approximates the natural similarity notion defined by the Jaccard index. However these techniques require instances to be represented as sparse *binary* vectors rather than sparse *real* vectors. We therefore binarize all instances from $\mathbb{R}^m \mapsto \{0,1\}^m$ setting to 1 all non-null components. Let $x, z \in \{0,1\}^m$ be two instances; the Jaccard similarity between the two instances is defined as $s(x,z) = \frac{|x \cap z|}{|x \cup z|}$, i.e., the ratio of the number of features that the instances have in common over the overall number of features. We build a min-hash function starting from a set of random hash functions $f_i : \mathbb{N} \mapsto \mathbb{N}$, i.e., functions that map integers randomly (but consistently) to integers; in our case the domain/co-domain represent feature indicators. These functions must be independent and satisfy: $\forall x_j \neq x_k, f_i(x_j) \neq f_i(x_k)$, and $\forall x_j \neq x_k, P(f_i(x_j) \leq f_i(x_k)) = \frac{1}{2}$. The min-hash function derived from f_i is defined as $h_i(x) = \arg\min_{x_j \in x} f_i(x_j)$, i.e., the min-hash returns the first feature indicator under a random permutation of the features order. A rather surprising (and useful) fact is that a min-hash collision is an unbiased estimator of the Jaccard similarity:

$$P(h_i(x) = h_i(z)) = \frac{|x \cap z|}{|x \cup z|} = s(x,z)$$

i.e. the probability to select as the minimum feature indicator a non-null feature that belongs to both x and z is exactly the fraction of features that x and z have in common over the total number of non-null features of x and z. In order to decrease the (high) variance of this estimate one can take N independent min-hash functions and compute the number n of times that $h_i(x) = h_i(z)$. The estimated value n/N is the average of N different 0-1 random variables, which evaluates to one when $h_i(x) = h_i(z)$ and zero in all other cases. The average of these unbiased estimators of $s(x,z)$ is also an unbiased estimator, with an expected error bounded by $O(1/\sqrt{N})$ [8], or, equivalently, for any constant $\gamma > 0$ we can compute a constant $N = O(1/\gamma^2)$ such that the expected error of the estimate is at most γ. For example, with 400 hash functions the estimate of $s(x,z)$ would have an expected error $\leq .05$.

We collect the results of the entire set of min-hash functions in an *instance sketch* as the tuple $\langle h_1(x), \ldots, h_N(x) \rangle$. In order to obtain an efficient neighbor search procedure, we build an inverse index that returns all instances with the same min-hash value in $O(1)$. More precisely, given the $i-$th hash function and a value $\bar{h} = h_i(x)$, we collect the set of instances $Z_i(\bar{h}) = \{z \in P : h_i(z) = \bar{h}\}$. The approximate neighbourhood Z of an instance x is then induced from the multi-set $Z = \{Z_i\}_{i=1}^N$. Note that when γ (or equivalently N) is fixed, the complexity to build a single signature is constant and therefore the complexity for building the index is linear in the size of the dataset. To improve the quality of the returned neighbors we consider only the most frequent elements in Z and sort them according to their `NSPDK` similarity to x. The $k-$neighborhood $N_k(x)$ is finally the set of the k-closest elements. If the size of Z is small and independent of the dataset size $|P|$, these steps can be performed in constant time.

[8] The relation can be obtained by standard Chernoff bounds for sums of 0-1 random variables.

S.3 `BlockClust` Parameters Optimization

In order to assess the best parameter settings for each tool used in the pipeline, attribute discretization and selection, we applied `BlockClust` on a specific data set with different parameter settings and different attribute combinations. We call each attribute combination as a *configuration*. For each configuration and parameter setting we measure the performance of the clustering and chose the best configuration and parameter settings for the usage of the `BlockClust`. In order to measure performance for the known ncRNA families we had to look at the annotations, hence it is supervised learning.

S.3.1 Mapping. We removed adapters and linkers from all raw reads using `fastx-clipper`[9] and applied `segemehl` Hoffmann *et al.* (2009) to align the clipped reads to the human genome (we reported only best scoring hits and required a minimum mapping accuracy of 85%). `segemehl` can efficiently deal mismatches and indels, it is independent of the underlying sequencing platforms and handles reads of different lengths. To correct for multiple mappings, we normalized the read counts n of each tag by the number of mappings k in the reference genome. Thus, the *tag expressions* n/k are assigned to each tag.

We relied on supervised learning to set up `BlockClust` and to find optimal attribute combinations and the best parameter values for the external tools run by our pipeline (`blockbuster` and `NSPDK`). Among others, this comprises the following major steps: partitioning of labeled input data; attribute generation, encoding, discretization and selection; parameter optimization; clustering.

S.3.2 Random partitioning. We randomly partitioned each benchmark dataset into three sub sets: train ($\sim 35\%$), validation ($\sim 35\%$), and test set ($\sim 30\%$). The training and validation set is used to benchmark our approach, e.g. during attribute selection and parameter optimization. The independent test set is used to obtain a final performance estimate on the fully trained model. Note, that these random splits were done on the level of reads to ensure unbiased learning. They are independent of any subsequent `blockbuster` or `BlockClust` call, no blocks or block groups have been assigned yet. Thus, instead of randomly distributing reads among train, validation and test set, we relied on the concept of "*read stretches*" for an unbiased partitioning of the data. We define a "*read stretches*" as a series of sorted reads separated by a maximum distance d. With the exception of a few ribosomal RNAs, most of the classic short ncRNAs are not longer than 500 nt. Thus, we set d to 500 and split the data on the level of read stretches, ensuring that most of the subsequently computed read profiles cover full ncRNA genes.

[9] `http://hannonlab.cshl.edu/fastx_toolkit`

Table S1. Overview on the samples used to benchmark `BlockClust`. Last three columns correspond to the number of blockgroups found by `blockbuster`, number of block groups filtered by length and expression level annotation and number of block groups retain at the end after intersecting with annotation.

GEO accession	Organism	Tissue/Cell line	Seq. machine	#Reads	#Tags	#BGs	#Filtered	#Known
GSE16368/GSM450239	Human	H1 cell line	Illumina GAII	16830686	618590	2404	755	629
GSE31069/GSM769509	Human	MCF-7 cytoplasmic	Illumina GAII	15493265	571470	2503	687	508
GSE31069/GSM769510	Human	MCF-7 total cell	Illumina GAII	14670735	519579	2168	586	474
GSE31069/GSM769511	Human	MCF-7 cytoplasmic	Illumina GAII	9237490	380461	1957	603	466
GSE31069/GSM769512	Human	MCF-7 total cell	Illumina GAII	8689337	320205	1770	552	458
GSE26545/GSM652847	Human	Cortex of brain	Illumina GAII	8241330	416757	1253	414	378
GSE26545/GSM652851	Human	Gyrus of the brain	Illumina GAII	6486498	490336	1232	464	410
GSE18012/GSM450597	Human	Gyrus of the brain (2 days)	Illumina GAII	6754470	231368	752	215	190
GSE18012/GSM450598	Human	Gyrus of the brain (34 days)	Illumina GAII	7299034	343234	1019	321	296
GSE18012/GSM450603	Human	Gyrus of the brain (98 years)	Illumina GAII	5763414	184097	760	238	224
GSE18012/GSM450605	Human	Gyrus of the brain (14 years)	Illumina GAII	8538940	729571	1554	524	482
GSE31037/GSM768988	Human	Skin	Illumina GAIIx	15579483	616913	2678	880	729
GSE31037/GSM769007	Human	Skin	Illumina GAIIx	21217688	360220	2534	863	750
GSE26545/GSM652849	Chimp	Cortex of brain	Illumina GAII	7776308	387720	1254	458	290
GSE26545/GSM652853	Chimp	Gyrus of the brain	Illumina GAII	7240683	512620	1272	413	247
GSE36639/GSM897819	Mouse	NIH 3T12 cells	Illumina GA	1843676	99306	625	247	223
GSE36639/GSM897820	Mouse	NIH 3T12 cells	Illumina GA	5694227	182587	1246	429	350
GSE36639/GSM897821	Mouse	NIH 3T12 cells	Illumina GA	8526798	213149	1520	461	370
GSE36639/GSM897822	Mouse	NIH 3T12 cells	Illumina GA	5682600	200515	1263	437	349
GSE36639/GSM897823	Mouse	NIH 3T12 cells	Illumina GA	6521133	232861	1264	449	336
GSE38702/GSM947965	Mouse	Testis	Illumina HiSeq 2K	23783785	2592368	8530	3144	133
GSE38702/GSM947966	Mouse	Uterus	Illumina GA	15876066	526206	1083	472	339
GSE11624/GSM272651	Fly	S2 & KC cells	Illumina GA	746043	163952	790	186	177
GSE11624/GSM286601	Fly	male heads	Illumina GA	621971	120124	308	161	159
GSE11624/GSM286602	Fly	male body	Illumina GA	980097	287958	1008	383	367
GSE40015/GSM983642	Fly	Female body	Illumina GA	1943622	1067609	3760	475	313
GSE40015/GSM983641	Fly	Female body	Illumina GA	487729	378940	1236	312	280
GSE17153/GSM427301	Worm	One cell embryo	Illumina GA II	3742851	427651	722	73	61
GSE17153/GSM427346	Worm	Mixed embryos	Illumina GA II	2965597	200072	658	246	229
GSE25738/GSM632205	Plant	seedlings	Illumina GA	11772773	1686361	10672	3514	364
GSE25738/GSM632207	Plant	seedlings	Illumina GA	11955547	2019497	7901	2265	220
GSE36934/GSM906549	Plant	leaves	Illumina GA IIx	6946527	1562959	12209	3813	260

After partitioning the read stretches to train,test and validaiton sets, we compute block groups using `blockbuster` on each set individually. A block group generalizes the expression profile of a ncRNA.

S.3.3 Annotation. We assigned a specific ncRNA class label to each block group using ncRNA annotation from different sources. We considered all human ncRNAs from the Rfam v10.1 (Gardner *et al.*, 2011) and Ensembl release-72 (Flicek *et al.*, 2012) databases. In addition, we downloaded miRNAs from miRBase v19 (Griffiths-Jones *et al.*, 2006), tRNAs from gtRNAdb (Chan and Lowe, 2009). See Table 2 for details.

For a reliable annotation, we filtered for block groups consisting of $\geq$ 2 blocks, a minimum expression of 50, a length between 50 and 200 nt, and a reciprocal overlap of at least 70% for each block group and its associated ncRNA. Block groups overlapping more than one ncRNA are likely to exhibit blurred read profiles and have been discarded. In line, if multiple block groups are found at a single ncRNA, we kept the block group with the largest overlap and ignored all others. Furthermore we tested for overlap of block groups mRNAs or pseudogenes from Ensembl database release-72 (Flicek *et al.*, 2012). We discarded the block groups with some significant overlap with mRNAs, as we consider reliable ncRNAs only.

After annotating the block groups with known ncRNAs we combined all train data sets of the 4 libraries together to get a single train set. Analogously done for validation and test data sets.

S.4 Attribute selection.

In order to identify the characteristic attributes of block groups we analyzed different sets of attribute: 5 were specific to block groups, 5 modeled blocks, and 2 were intended to capture the relation between blocks, see Supplementary Table S3.

As characteristic attributes of block groups we analyzed entropies of tag starts, tag ends, tag lengths. We define *entropy of tag starts* as follows: let q_i denote the fraction of tags in a given block group starting at position i. The *entropy of tag starts* is then defined as $-\sum_i q_i \log_2 q_i$. Analogously, we defined the *entropy of tag ends* and *entropy of tag lengths*. In addition to these entropies, *median*

Table S2. Overview on the ncRNA classes and annotation sources used to develop and benchmark `BlockClust`. All numbers refer to version hg19 of the human genome.. Database versions are as follows: Ensembl v72, Rfam v11.0, miRBase v20.

ncRNA family	Database	No. of ncRNAs
tRNA	gtRNADB, Rfam, Ensembl	625+904+22
miRNA	miRBase, Rfam, Ensembl	1871+1232+3215
snoRNA C/D box	Rfam, Ensembl	511+748
snoRNA H/ACA box	Rfam, Ensembl	440+312
rRNA	Rfam, Ensembl	608+508
snRNA	Rfam, Ensembl	2023+1404
Y_RNA	Rfam, Ensembl	893+821

of tag expressions and *tag expressions in first quantile* reveal the distribution of expression levels of tags within the block group. For blocks also we computed the *entropy of tag lengths* and in addition *entropy of the tag expressions*. In block level the *number of multi mapped tags* is an important attribute to consider. For miRNAs and C/D box snoRNAs it is too low compared to tRNAs (see Supplementary Figure S1. This attribute is related to *tag expressions in first quantile* of block group. With increasing number of multi mapped tags, the tag expressions divides by number of times it mapped (see section S.3.1). Hence low expression in first quantile for tRNA, rRNA and snRNs. *Length of the block* it self and the *minimum tag length*, i.e., the shortest tag length within a block are also considered as block specific attributes. For each pair of two adjacent blocks, we computed their pairwise *block contiguity*. This single attribute represents the percentage overlap or percentage distance between two consecutive blocks, resp. We calculate total edge length between adjacent blocks, i.e. the number of bases spawned by both adjacent blocks including the gap between them. Then we calculate the distance or overlap between the those two blocks. To distinguish overlap and distance we use positive values for overlap and negative values for distance between blocks. We define *block contiguity* as the ratio of block overlap or distance to the total edge length. Let two adjacent blocks B_1 and B_2 with start positions s_{B_1}, s_{B_2} and end positions e_{B_1}, e_{B_2} respectively. The *block contiguity* for two adjacent blocks is defined as $(e_{B_1} - s_{B_2})/(e_{B_2} - s_{B_1})$.

For each block we computed median of tag expressions and take the difference of these medians for adjacent blocks as a attribute. Let a *block height* be the highest tag expression within a Supplementary Table S 3 gives an overview of selected attributes in learning phase.

Table S3. Attribute selection. Overview of the selected attributes for the graph encoding.

Category	Attribute
block group	entropy of tag starts
block group	entropy of tag ends
block group	entropy of tag lengths
block group	median of tag expressions
block group	tag expression levels in first quantile
block	number of multi mapped tags
block	entropy of tag lengths
block	entropy of tag expressions
block	minimum tag length
block	block length
pairs of blocks	contiguity
pairs of blocks	difference in median tag expressions

Table S4. Clustering performance of `BlockClust` on `Benchmark Data`. The AUC of block group similarities indicate that `BlockClust` is robust across these diverge data sets. Note that the training was done on human data sets and performs fairly well on fly, worm and plant.

GEO accession	miRNA		tRNA		CD-box		HACA-box		rRNA		snRNA		YRNA		Average	
	#	AUC	#	AUC	#	AUC	#	AUC	#	AUC	#	AUC	#	AUC	#	AUC
GSE16368/GSM450239	226	0.899	208	0.843	95	0.719	14	0.803	38	0.836	14	0.679	31	0.754	629	0.835
GSE31069/GSM769509	170	0.926	218	0.774	29	0.827	7	0.866	52	0.776	18	0.596	13	0.592	508	0.819
GSE31069/GSM769510	164	0.899	190	0.816	67	0.772	12	0.813	24	0.884	5	0.501	11	0.639	474	0.835
GSE31069/GSM769511	134	0.925	222	0.778	33	0.795	0	0	47	0.766	19	0.545	10	0.559	466	0.806
GSE31069/GSM769512	148	0.907	186	0.822	77	0.779	7	0.754	25	0.797	5	0.652	8	0.841	458	0.839
GSE26545/GSM652847	166	0.888	127	0.702	43	0.675	3	0.719	2	0.991	2	0.698	35	0.667	378	0.779
GSE26545/GSM652851	164	0.905	154	0.702	39	0.639	4	0.785	3	0.862	12	0.800	34	0.679	410	0.780
GSE18012/GSM450597	146	0.850	22	0.628	14	0.590	1	1.000	1	1.000	1	1.000	5	0.910	190	0.809
GSE18012/GSM450598	178	0.916	78	0.776	19	0.641	2	0.918	2	0.881	1	1.000	16	0.729	296	0.851
GSE18012/GSM450603	157	0.899	51	0.744	7	0.767	1	1.000	2	0.898	2	0.990	4	0.976	224	0.862
GSE18012/GSM450605	189	0.911	150	0.714	46	0.630	9	0.727	3	0.690	40	0.839	44	0.705	482	0.793
GSE31037/GSM768988	182	0.932	243	0.748	117	0.830	42	0.911	89	0.768	41	0.659	10	0.609	729	0.813
GSE31037/GSM769007	207	0.905	245	0.774	128	0.829	40	0.892	76	0.769	33	0.645	16	0.629	750	0.817
Human	2231	0.905	2094	0.770	714	0.759	142	0.859	364	0.789	193	0.690	237	0.693	5994	0.817
GSE26545/GSM652849	149	0.951	130	0.723	6	0.859	0	0	0	0	1	1.000	4	0.734	290	0.844
GSE26545/GSM652853	145	0.931	92	0.695	5	0.773	0	0	0	0	2	0.989	3	0.784	247	0.839
Chimp	294	0.941	222	0.711	11	0.820	0	0	0	0	3	0.993	7	0.755	537	0.842
GSE36639/GSM897819	133	0.951	90	0.699	0	0	0	0	0	0	0	0	0	0	223	0.849
GSE36639/GSM897820	153	0.964	144	0.697	0	0	0	0	53	0.966	0	0	0	0	350	0.854
GSE36639/GSM897821	162	0.962	149	0.716	0	0	0	0	59	0.948	0	0	0	0	370	0.861
GSE36639/GSM897822	146	0.973	150	0.702	0	0	0	0	53	0.926	0	0	0	0	349	0.849
GSE36639/GSM897823	131	0.940	142	0.700	0	0	0	0	63	0.879	0	0	0	0	336	0.827
GSE38702/GSM947965	56	0.865	77	0.752	0	0	0	0	0	0	0	0	0	0	133	0.800
GSE38702/GSM947966	156	0.932	133	0.762	0	0	0	0	50	0.945	0	0	0	0	339	0.868
Mouse	937	0.949	885	0.716	0	0	0	0	278	0.931	0	0	0	0	2100	0.848
GSE11624/GSM272651	48	0.983	124	0.903	0	0	0	0	0	0	5	0.649	0	0	177	0.917
GSE11624/GSM286601	69	0.970	90	0.787	0	0	0	0	0	0	0	0	0	0	159	0.866
GSE11624/GSM286602	65	0.992	193	0.754	0	0	0	0	98	0.986	11	0.742	0	0	367	0.858
GSE40015/GSM983641	64	0.967	231	0.937	4	0.909	0	0	0	0	14	0.639	0	0	313	0.929
GSE40015/GSM983642	59	0.991	213	0.960	2	0.996	0	0	0	0	6	0.862	0	0	280	0.965
Fly	305	0.980	851	0.880	6	0.938	0	0	98	0.986	36	0.709	0	0	1296	0.907
GSE17153/GSM427301	11	0.930	15	0.701	0	0	0	0	15	0.982	5	0.632	0	0	61	0.801
GSE17153/GSM427346	32	0.982	142	0.776	0	0	0	0	15	0.986	0	0	0	0	229	0.768
Worm	43	0.969	157	0.769	0	0	0	0	30	0.984	5	0.632	0	0	290	0.775
GSE25738/GSM632205	20	0.892	341	0.875	0	0	0	0	0	0	1	1.000	0	0	364	0.876
GSE25738/GSM632207	17	0.886	201	0.910	0	0	0	0	0	0	2	0.844	0	0	220	0.907
GSE36934/GSM906549	22	0.885	237	0.884	0	0	0	0	0	0	0	0	0	0	260	0.885
Plant	59	0.888	779	0.887	0	0	0	0	0	0	3	0.896	0	0	844	0.887
All	3869	0.925	4988	0.795	731	0.762	142	0.859	770	0.873	240	0.698	244	0.694	11061	0.839

Table S5. Classification performance of `BlockClust` on `Benchmark Data`. `BlockClust` was applied on a total of 32 independent data sets from 6 different species and several tissues and cell lines. Despite of some poor recall values for CD-box snoRNAs and tRNAs, `BlockClust` performed well on these diverse data sets.

GEO accession	miRNA			tRNA			snoRNA C/D-box		
fold	#	PPV	Recall	#	PPV	Recall	#	PPV	Recall
GSE16368/GSM450239	226	0.887	0.832	208	0.814	0.822	95	0.592	0.337
GSE31069/GSM769509	170	0.888	0.882	218	0.821	0.821	29	0.526	0.345
GSE31069/GSM769510	164	0.885	0.890	190	0.950	0.795	67	0.743	0.388
GSE31069/GSM769511	134	0.883	0.903	222	0.829	0.806	33	0.647	0.333
GSE31069/GSM769512	148	0.878	0.872	186	0.903	0.747	77	0.795	0.403
GSE26545/GSM652847	166	0.875	0.928	127	0.831	0.504	43	0.700	0.326
GSE26545/GSM652851	164	0.885	0.848	154	0.770	0.500	39	0.636	0.359
GSE18012/GSM450597	146	0.946	0.959	22	0.800	0.545	14	0.600	0.214
GSE18012/GSM450598	178	0.955	0.944	78	0.786	0.564	19	0.375	0.158
GSE18012/GSM450603	157	0.980	0.943	51	0.806	0.490	7	0.286	0.286
GSE18012/GSM450605	189	0.898	0.884	150	0.638	0.587	46	0.421	0.174
GSE31037/GSM768988	182	0.945	0.940	243	0.633	0.732	117	0.886	0.265
GSE31037/GSM769007	207	0.954	0.894	245	0.651	0.792	128	0.732	0.234
GSE26545/GSM652849	149	0.969	0.846	130	0.985	0.508	6	0.545	1.000
GSE26545/GSM652853	145	0.977	0.862	92	0.881	0.402	5	0.167	0.400
GSE36639/GSM897819	133	0.961	0.940	90	1.000	0.433	0	0.000	0.000
GSE36639/GSM897820	153	0.985	0.837	144	0.971	0.701	0	0.000	0.000
GSE36639/GSM897821	162	0.986	0.876	149	0.882	0.705	0	0.000	0.000
GSE36639/GSM897822	146	0.992	0.877	150	0.940	0.627	0	0.000	0.000
GSE36639/GSM897823	131	0.975	0.893	142	0.844	0.570	0	0.000	0.000
GSE38702/GSM947965	56	1.000	0.804	77	0.965	0.714		0.000	0.000
GSE38702/GSM947966	156	1.000	0.808	133	0.808	0.444	0	0.000	0.000
GSE11624/GSM272651	48	0.977	0.875	124	0.968	0.726	0	0.000	0.000
GSE11624/GSM286601	69	1.000	0.768	90	1.000	0.611	0	0.000	0.000
GSE11624/GSM286602	65	0.977	0.661	193	0.787	0.782	0	0.000	0.000
GSE40015/GSM983641	64	1.000	0.781	231	0.950	0.831	4	0.000	0.000
GSE40015/GSM983642	59	1.000	0.898	213	0.972	0.831	2	0.000	0.000
GSE17153/GSM427301	11	0.714	0.909	15	0.136	0.200	0	0.000	0.000
GSE17153/GSM427346	32	0.806	0.906	142	0.828	0.747	0	0.000	0.000
GSE25738/GSM632205	20	0.941	0.800	341	1.000	0.595	0	0.000	0.000
GSE25738/GSM632207	17	1.000	0.647	201	0.985	0.647	0	0.000	0.000
GSE36934/GSM906549	22	0.944	0.773	237	1.000	0.641	0	0.000	0.000

Table S6. Clustering performance of `BlockClust` on 10 random test splits of `Development Data` measured by the average per instance AUC ROC.

fold	miRNA		tRNA		C/D-box		H/ACA-box		rRNA		snRNA		YRNA		Average	
	#	AUC	#	AUC	#	AUC	#	AUC	#	AUC	#	AUC	#	AUC	#	AUC
1	156	0.891	166	0.728	89	0.729	7	0.849	12	0.770	7	0.653	7	0.711	444	0.789
2	157	0.893	180	0.734	64	0.720	5	0.800	28	0.832	9	0.668	9	0.722	452	0.802
3	171	0.884	158	0.772	81	0.750	5	0.795	26	0.942	6	0.666	10	0.646	457	0.822
4	160	0.908	173	0.757	87	0.744	3	0.970	28	0.877	6	0.650	6	0.719	463	0.812
5	181	0.890	174	0.703	78	0.761	4	0.942	16	0.845	9	0.580	6	0.799	468	0.798
6	169	0.906	174	0.728	74	0.695	1	1.000	21	0.840	5	0.650	9	0.624	453	0.795
7	169	0.905	175	0.712	88	0.754	3	0.808	17	0.893	6	0.641	7	0.645	466	0.797
8	166	0.897	174	0.771	68	0.698	8	0.767	24	0.914	10	0.591	3	0.682	453	0.816
9	176	0.910	183	0.735	75	0.731	5	0.808	12	0.899	8	0.621	9	0.589	468	0.802
10	172	0.881	177	0.768	76	0.711	1	1.000	13	0.864	7	0.688	9	0.758	455	0.812

Table S7. Clustering performance of `BlockClust` on 10 random test splits of `Development Data` measured on `MCL` clustering precision.

fold	miRNA		tRNA		C/D-box		rRNA		Average	
	#clusters	Precision	#clusters	Precision	#clusters	Precision	#clusters	Precision	#clusters	Precision
1	7	0.833	20	0.827	10	0.614	1	1.000	38	0.777
2	8	0.830	22	0.794	6	0.662	2	1.000	38	0.792
3	10	0.824	18	0.861	8	0.898	2	1.000	38	0.852
4	7	0.881	20	0.896	13	0.680	2	1.000	42	0.832
5	12	0.846	19	0.819	10	0.597	1	0.750	42	0.772
6	8	0.910	24	0.834	9	0.608	1	1.000	42	0.804
7	7	0.822	18	0.831	8	0.739	1	1.000	36	0.813
8	11	0.851	11	0.875	3	0.628	2	0.961	27	0.845
9	20	0.841	8	0.833	3	0.711	2	0.833	34	0.823
10	13	0.912	12	0.808	4	0.761	1	1.000	20	0.853

Table S8. Classification performance of `BlockClust` on 10 random test splits of the `Development Data`. For miRNAs and tRNAs we achieved a mean precision about 0.9 and for C/D-box snoRNAs about 0.87. The recall for these three classes varies. The highest mean recall obtained for miRNAs about 0.89 and for tRNAs it is 0.8. C/D-box snoRNAs show lowest mean recall about 0.47.

	miRNA			tRNA			snoRNA C/D-box		
fold	#	PPV	Recall	#	PPV	Recall	#	PPV	Recall
1	156	0.887	0.853	166	0.873	0.873	89	0.868	0.371
2	157	0.937	0.847	180	0.869	0.778	64	0.931	0.422
3	171	0.874	0.930	158	0.881	0.892	81	0.911	0.506
4	160	0.898	0.881	173	0.944	0.775	87	0.796	0.449
5	181	0.883	0.917	174	0.879	0.839	78	0.836	0.654
6	169	0.852	0.917	174	0.848	0.897	74	0.879	0.392
7	169	0.936	0.776	175	0.888	0.771	88	0.971	0.375
8	166	0.896	0.9334	174	0.937	0.598	68	0.833	0.515
9	176	0.917	0.875	183	0.943	0.721	75	0.819	0.480
10	172	0.924	0.924	177	0.925	0.830	76	0.852	0.606

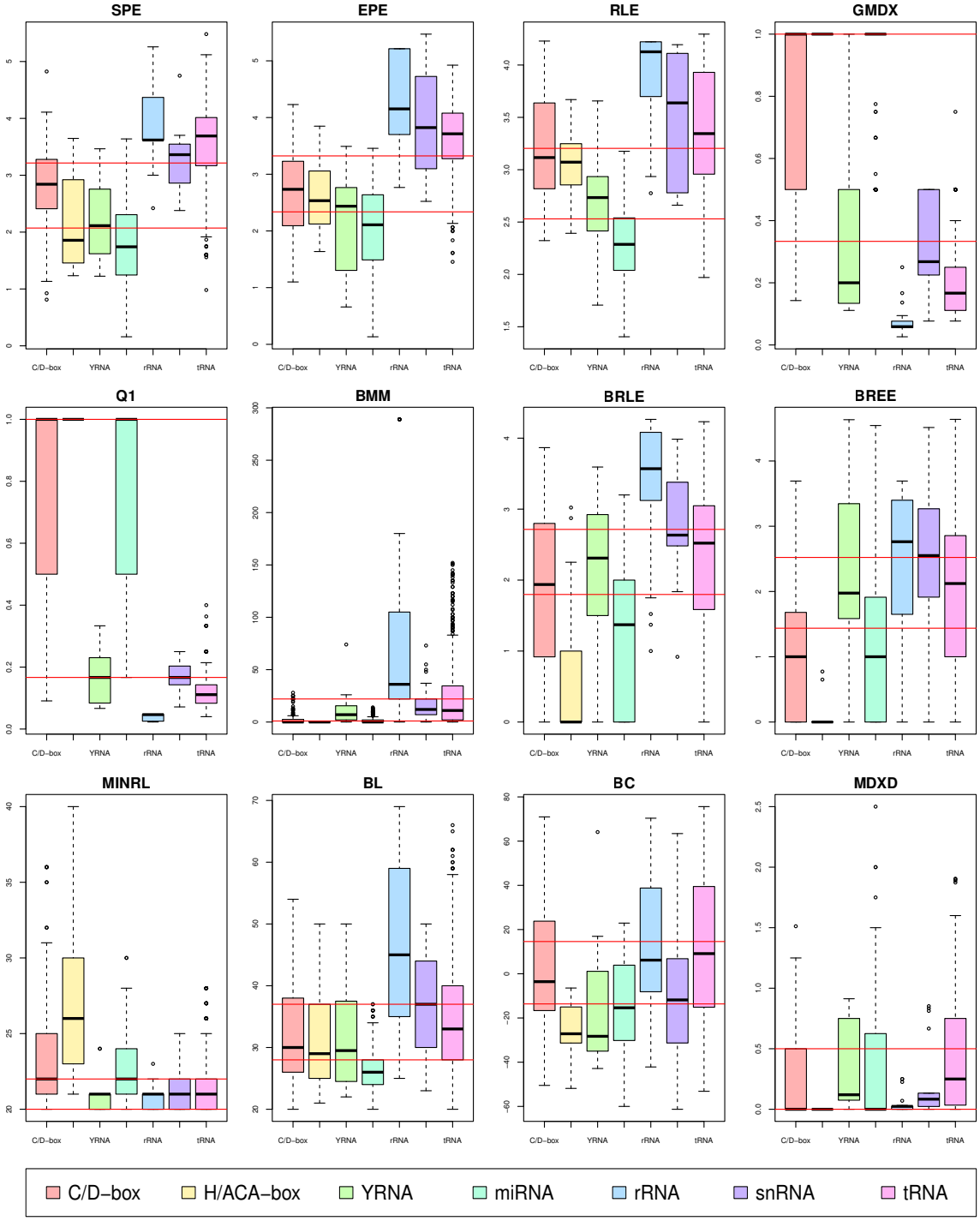


Fig. S1: Attribute value boxplots. The selected discretization levels are depicted with red lines.

Supplementary material for

[P2] FOXG1 Regulates PRKAR2B Transcriptionally and Posttranscriptionally via miR200 in the Adult Hippocampus

Supplementary Methods

Cell Culture and transfection

Mouse neuroblastoma cell line, Neuro-2a (N2a) were cultured in Dulbecco's modified Eagle's medium (DMEM, ThermoScientific, Schwerte, Germany) supplemented with 10% fetal bovine serum (FBS, ThermoScientific), 1% non essential amino acids (NEAA, ThermoScientific), 1% L-glutamine, and 1% penicillin, streptomycin, and neomycin (PSN, ThermoScientific). Cells were maintained at 37°C, 95% relative humidity and 5% CO_2. Cells were seeded either on coverslips, for PLA, or in 6 well plates and were transfected with Lipofectamine LTX adding a total amount of 2.5 ug plasmid according to manufactureres instructions (ThermoScientific). 10 cm dishes were transfected using calcium phosphate transfection method. For KD experiments, cells were selected with 2.4 ng/ml puromycin for 24 h.

Plasmids

pLenti-III-Empty-2A-GFP and pLenti-III-FOXG1-HA-2A-GFP (abmGood, Canada), pLenti-III-FOXG1-Au1-2A-GFP, pLenti-III-FOXG1-D2-Au1-2A-GFP (cloned by Dr. Gensch), pLKO-non-target-GFP, pLKO-shFoxg1-GFP and pLKO-shDdx5-GFP (Sigma). pmiRGlo-empty (Promega). pCX-miR200b/a/429/200c/141 and pCX-eGFP-miR200-sponge were generous gifted by Dr. Harold Cremer. pCX-miR200b/a/429 and pmiRGlo-miR200b/a/429 were subcloned from pCX-miR200b/a/429/200c/141 plasmid. pmiRGlo-Prkar2b-3'UTR-short, pmiRGlo-Prkar2b-3'UTR-inverted, pmiRGlo-Prkar2b-3'UTR-T7, pmiRGlo-5'-MCS-empty and pmirGlo-5'-MCS-Prkar2b-5'-region were cloned by GenScript USA Inc. pCMV-FOXG1-Au1 was subcloned from pLenti-III-FOXG1-Au1-2A-GFP .

RNA isolation, reverse transcription, and quantitative real-time PCR (qRT-PCR)

RNA was isolated from harvested cells and frozen tissue using miRNeasy kit (Qiagen) according to the manufacturer's instructions including on-column DNA digestion. 1 µg of total RNA was reverse transcribed either with RevertAid MMuLV (Fermentas, ThermoScientific) or with miScript® II RT kit (Qiagen) according to manufacturer's protocol. mRNA samples were subjected to DNase I treatment just before cDNA synthesis with amplification grade DNase I (Sigma Aldrich) for 30 min at RT. qRT-PCR analysis for mRNA and pri-miRNA were performed on CFX-Connect Real-Time PCR detection system (Bio-Rad) using Go Taq qPCR Master Mix (Promega, Mannheim, Germany) or using Qiagen miScript SYBR® Green PCR Kit with Qiagen miScript® primer assay (miR200a/b/429 and U6) or Qiagen miScript® Precursor assay (pre-miR200a/b/429) according to manufacturer's protocol. Primers were used at a concentration of 250 nM each. *Gapdh* or *U6* were used as reference genes. For mRNA and pri-miR200, PCR program was 3 minutes at 95°C, 40 cycles of 15 sec at 95°C and 30 sec at an annealing temperature (58°C-63°C), followed by 1 min at 95°C, 1 min at 55°C and melting curve cycle. For pre-miR200 and mature miR200, PCR program was 15 min at 95°C, 40 cycles of 15 sec at 94°C, 30 sec at 55°C and 30 sec at 70°C, followed by 1 min at 95°C, 1 min at 55°C and melting curve cycle. Primers used had an efficiency level between 85% and 110%. Primer sequences are listed in Supplementary Table S1. qRT-PCR results were analysed using the ΔΔCt method [1].

Bioinformatics analysis of RNAseq and FOX transcription factor binding motifs

For 6 week old *Foxg1*$^{cre/+}$ mice hippocampal RNAseq and miRNA-200b/a/429 overexpressing N2a cells RNAseq, n = 3 and n = 2 were used respectively. Bioinformatics analysis was performed using the Freiburger Galaxy Server [2]. At

first, the sequenced reads in FASTQ files were inspected using FastQC [3]. With no remarkable quality flaws from the FastQC reports, low quality bases from the 3' end were trimmed using TrimGalore [4]. For quality trimming, Phred score cut-off of 28 was used. Reads were aligned to mouse genome build mm10 using TopHat2 [5]. For Foxg1$^{cre/+}$ samples, we set options --mate-inner-dist to 0, --mate-std-dev to 80 and --library-type to fr-firststrand; whereas default settings were used for the mir200-OE in N2a samples. For mapping both datasets --GTF option with a gene annotation model from ensemble release 79 [6] in gene transfer format were used. Later, htseq-count [7] was used to count the number of aligned reads per gene. For both datasets we set --mode to union and for *Foxg1*$^{cre/+}$ datasets set --stranded to reverse. In the end, DESeq2 [8] was used for differential gene expression analysis. Adjusted *p* value of 0.05 or less as the significant threshold was chosen for differentially expressed genes.

Small RNA-Seq was performed on an Illumina HiSeq 2000 system. Small RNA libraries were prepared from 1 µg total RNA using the Illumina TruSeq Small RNA Sample Preparation kit. For processing of sequencing data a customized in-house software pipeline was used. Quality check and demultiplexing were performed using the CASAVA 1.8.2 software (Illumina). We trimmed the 3' adapters and filtered out the reads with the minimum length of 15 nucleotides using cutadapt. We first map the reads to the reference genome created from microRNA sequences. Remaining unmapped reads were then mapped to mouse genome. We used rna-STAR for all the mapping. We allowed no mismatches for the reads <25b, one mismatch for reads between 26b to 33b. We mapped all the reads in the non-splice-junction-aware mode. For comparison of miRNA expression between samples, a differential expression analysis was performed using R, DESeq2 and RUVseq package. miRNAs were considered to be differentially expressed with an adjusted p-value below 0.05.

To identify putative binding sites of forkhead box binding sites, we downloaded all mouse FOX transcription factor binding profiles from JASPAR database [9]. We subsequently used the tool FIMO [10] with default settings to search for the FOX motifs on 1000 bp upstream of the mir200b/a/429 gene cluster.

GO term and KEGG pathway analyses were performed with DAVID [11, 12]. The 34 overlapping genes from miR200-overexpression and *Foxg1*$^{cre/+}$ RNA-Seq were used for GO term analyses for biological processes and cellular compartments using the ''official gene names'' and "mus musculus" as species. To identify miR200 targets among the 34 targets, the following miRNA target prediction tools were used: Targetscan v6.2 [13], miRanda [14], miRDB [15], MicroCosm v5 [16].

Luciferase Assay

N2a cells were transfected with pmiRGlo-miR200b/a/429 for DROSHA activity assay or with pmiRGlo-Prkar2b-3'UTR-short, pmiRGlo-Prkar2b-3'UTR-Invert or pmiRGlo-Prkar2b-3'UTR-T_7 for miR200 target validation experiment. For FOXG1 activity on *Prkar2b* promoter we used the plasmids pmirGlo-5'-MCS-empty and pmirGlo-5'-MCS-Prkar2b-5'-region. Cells were harvested 48 h after transfection with 1X Passive Lysis Buffer (Promega). The luciferase assay was performed with the Dual Luciferase System Kit (Promega) according to manufacturer's instructions. Shortly, 5 µl cell lysate was first incubated with LARII as substrate for firefly luciferase followed by Stop&Glo to inhibit firefly luciferase activity and as a substrate for the renilla luciferase. Luminescence intensity was measured with a 2 s delay for 10 s, with the Glomax96 luminometer. Firefly luciferase activity was normalized to renilla luciferase activity to calculate relative luciferase activity of each condition.

SILAC and mass spectrometry

For HA-co-IP, one 6-well plate with 250000 cells per well were transfected either with pLenti3-Foxg1-HA-T2A-eGFP (abmGood, Canada) or pLenti3-Foxg1-Au1-T2A-eGFP. Cells were lysed in co-IP buffer (100 mM NaCl, 20 mM Tris, 1 mM EDTA, 0.5% NP40-alternative, pH7.4) supplemented with protease inhibitor (cOmplete Protease Inhibitor Cocktail, Roche-Diagnostics, Mannheim, Germany). Protein amounts of both conditions were estimated by Bradford reagent (BioRad, Munich, Germany). 1.4 mg of each condition was precleared for 1~h with sepharose beads (Protein A Sepharose Cl-4B, GE Healthcare), before HA-co-IP was performed with 70 µl of HA-coupled sepharose beads (#3956, Cell-Signaling, Frankfurt a. M., Germany) over night. Antigen-coupled beads were washed 3 times in co-IP buffer. After the last washing, HA-IP and MOCK-IP were pooled and resuspended in 60 µl 1x Laemmli buffer.

Samples for mass spectroscopy were prepared with 1 mM DTT for 5 min at 95 ^{o}C and alkylated using 5.5 mM iodacetamide for 30 min at 25^{o}C. Protein mixtures were separated by SDS-PAGE (4-12% Bis-Tris mini gradient gel) and gel lanes were cut into 10 equal slices. Gel fractions were in-gel digested using trypsin (Promega, Mannheim, Germany) [17]. Digests were performed overnight at 37°C in 0.05 M NH4HCO3 (pH 8). About 0.1 µg of protease was used for each gel band. Peptides were extracted from the gel slices with ethanol and resulting peptide mixtures were processed on STAGE tips as described [18].

Samples analyzed by MS were measured on LTQ Orbitrap XL mass spectrometer (ThermoFisherScientific, Bremen, Germany) coupled to an Agilent 1200 nanoflow-HPLC (Agilent Technologies GmbH, Waldbronn, Germany). HPLC-column tips (fused silica) with 75 µm inner diameter were self-packed with Reprosil-Pur 120 ODS-3 to a length of 20 cm. No pre-column was used. Peptides were injected at a flow of 500 nl/ min in 92% buffer A (0.5% acetic acid in HPLC gradient grade water) and 2% buffer

B (0.5% acetic acid in 80% acetonitrile, 20% water). Separation was achieved by a linear gradient from 10% to 30% of buffer B at a flow rate of 250 nl/min. The mass spectrometer was operated in the data-dependent mode and switched automatically between MS (max. of 1 x10 ions) and MS/MS. Each MS scan was followed by a maximum of five MS/MS scans in the linear ion trap using normalized collision energy of 35% and a target value of 5,000. Parent ions with a charge states of z = 1 and unassigned charge states were excluded from fragmentation. The mass range for MS was m/z = 370 to 2,000. The resolution was set to 60,000. MS parameters were as follows: spray voltage 2.3 kV; no sheath and auxiliary gas flow; ion transfer tube temperature 125°C. Software Xcalibur (Thermo Scientific) and Mascot Daemon version 2.4.0 (Matrix Science, London, UK) were used for data acquisition and processing.

The MS raw data files were uploaded into the MaxQuant software version 1.4.1.2 [19], which performs peak and SILAC-pair detection, generates peak lists of mass error corrected peptides and data base searches. A full-length mouse database containing common contaminants, such as keratins and enzymes used for in-gel digestion, was employed, carbamidomethylcysteine was set as fixed modification and methionine oxidation and protein amino-terminal acetylation were set as variable modifications. Double SILAC was chosen as quantification mode. Three miss cleavages were allowed, enzyme specificity was trypsin/P+DP, and the MS/MS tolerance was set to 0.5 Da. The average mass precision of identified peptides was in general less than 1 ppm after recalibration. Peptide lists were further used by MaxQuant to identify and relatively quantify proteins using the following parameters: peptide, and protein false discovery rates (FDR) were set to 0.01, maximum peptide posterior error probability (PEP) was set to 0.1, minimum peptide length was set to 6, minimum number peptides for identification and quantitation of proteins was set to

two, of which one must be unique, and identified proteins have been re-quantified. The “match-between-run” option (2 min) was used.

Immunoprecipitation

Tissue or N2a cells were lysed in co-IP buffer (100 mM NaCl, 20 mM Tris, 1 mM EDTA, 0.5% NP40-alternative, pH 7.4) supplemented with protease inhibitor (Roche) and lysed by incubation for 30 min on ice, triturating every 10 min 20 times. After centrifugation (10 min, 13000 rpm) the supernatant was collected. Protein concentrations were determined with Bradford reagent (Bio-Rad). 5% input was saved and equal amounts of protein were used for MOCK and all co-IPs. Protein G Dynabeads (10004D, ThermoScientific) were coupled for 1 h at room temperature and 1 h at 4°C with Co-IP antibodies or control IgG antibody (rabbit IgG kch-504-250, Diagenode, Seraing, Belgium). Cell lysates were blocked with Protein G Dynabeads for 1 h at 4°C, subsequently transferred to antibody-coupled bead and incubated while rotating over night at 4°C. Antigen-coupled beads were washed 3 times with co-IP buffer before they were resuspended in 30 µl 1x laemmli buffer. 5% input and the complete Co-IP sample were used for immunoblotting.

Immunoblotting

Protein samples for WB were prepared as described for the co-IP samples. Protein or co-IP samples were loaded either on 8% or 10% SDS-polyacrylamide gels and run at 120V for 1.5 h. Proteins were transferred to PVDF membranes (Trans-blot Turbo Transfer Pack) using the Trans-blot Turbo Transfer System (Bio-Rad) following the manufacturer’s instructions. Membranes were blocked with 5% BSA in TBS-T (blocking buffer) for 1 h and incubated overnight with primary antibodies (diluted in blocking buffer). Membranes were washed, incubated with secondary antibodies for

1 h and detected using Femto substrates (Thermo Scientific) and LAS ImageQuant System (GE Healthcare, Little Chalfont, UK).

Cell fractionation

For protein and co-IP, cytoplasm, nucleoplasm and chromatin were fractionated according to the protocol reported in [20].

Proximity ligation assay (PLA)

N2a cells were fixed in 4% PFA. Cells were permeabilized for 15 min with 0.1% Triton-X100, before incubation with the Duolink blocking solution for 1 h. Cells were incubated with primary antibodies diluted in Duolink antibody diluent solution over night at 4°C. After washing the cells, they were first incubated with PLA-RED Probes for 1h at 37°C, then with ligation solution for 30 min at 37°C and finally amplification solution was added for 100 min at 37°C. For the following immunocytochemistry, cells were blocked again with Duolink blocking solution/0.1% Triton-X100 for 1 h before they were incubated with the primary Lamin B1 antibody in the blocking solution over night at 4°C. Following washing with PBS, cells were incubated with the donkey-anti-rabbit-488 (1:500, 711-545-152, Dianova) for 1 h at room temperature. Before mounting coverslips with fluorescent mounting medium (#S3023, DAKO, Jena, Germany), nuclei were stained with DAPI.

RNA immunoprecipitation (RIP)

N2a cells were cultured in 10 cm dishes (one 10 cm dish was used per RIP) and after 48 h of transfection, cells were collected and lysed using 750 µl RIPA buffer (150 mM NaCl, 1% NP-40, 0.5% Sodium deoxycholate, 0.1% SDS, 50 mM Tris-HCl (pH 7.4), 1 mM EDTA). 250 µl lysate were used for FOXG1-Au1 or DDX5 RIP and 250 µl lysate

for IgG RIP. The remaining lysate was saved as input. 7 µl of anti-Au1, anti-DDX5 or an appropriate IgG antibody was incubated with Protein G Dynabeads in RIPA for 2 hr at RT. The cell lysates were incubated with the antibody coupled beads overnight at 4°C. After incubation, the beads were washed four times in high salt RIPA buffer (1 M NaCl, 1% NP-40, 0.5% Sodium deoxycholate, 0.1% SDS, 50 mM Tris-HCl (pH 7.4), 1 mM EDTA), followed by a final wash in 1 ml PBS. 100 µl of beads were collected for protein analysis by immunoblot and 900 µl of beads were used for RNA extraction using Qiagen miRNeasy kit according to the manufacturer's protocol.

miRNA analysis by Northern hybridization

The separation of RNA samples enriched for small RNAs via denaturing polyacrylamide gel electrophoresis and their analysis by Northern hybridization was performed as described [21] with the following variations. 4-5 µg of small RNA per lane were separated on polyacrylamide (PAA)-urea minigels (15% PAA, 0.5 g/ml urea, 1x Tris-Borate-EDTA (TBE) buffer), electroblotted and cross-linked onto positivated Porablot NY plus nylon membrane (Macherey-Nagel GmbH & Co. KG). The RNA sizes were estimated using the microRNA Marker (NEB). For the hybridization of the U6 snRNA, membranes were prehybridized for 60 min at 62°C, for the detection of miRNAs and the marker at 45°C with hybridization buffer (50% deionized formamide, 7% SDS, 250 mM sodium chloride, 120 mM sodium phosphate, pH 7.2) under continuous rotation. Probes (for the detection of miR429, mir200a, miR200b and U6 snRNA) were generated by *in vitro* transcription using *mir*Vana™ miRNA probe construction kit (Thermo Fischer Scientific), while the microRNA marker was detected by hybridization with probe (5´-AAATCTCAACCAGCCACTGCT-3´-Biotin) supplied by NEB. The probes against

microRNA marker were 5'-end-labeled using 50 µCi [γ32P] ATP (3.000 Ci/mmol, Hartmann Analytic) and 20 U of T4 polynucleotide kinase (Thermo Fisher Scientific) for 30 min at 37°C. Membranes were hybridized at 62°C (U6 snRNA) or 45°C (all other probes) over night and washed at 57°C (U6 snRNA) or 40°C (other probes) with washing solutions I (2x SSC and 1% SDS), II (1x SSC and 0.5% SDS) and III (0.1x SSC and 0.1% SDS) for 10 min each. The signals were detected with a storage phosphor screen (Kodak) and a GE Typhoon FLA 9500 imaging system.

Mouse hippocampus dissection, culture of neurons and viral transduction

NMRI (Charles River) hippocampi of P0 embryos were dissected and collected in 5 ml Hanks' Balanced Salt (HBSS, Fisher Scientific, Schwerte, Germany) and dissociated in 0.25% Trypsin/EDTA (Fisher Scientific) at 37 °C for 10 min. Dissociation was stopped by adding NB-complete medium and 10% fetal bovine serum (FBS, Fisher Scientific). Cells were collected by centrifugation and cultured in NB-complete medium (Neurobasal medium (Fisher Scientific) supplemented with B27 (Fisher Scientific), L-glutamine (0.5 mM, Fisher Scientific), penicillin-streptomycin-neomycin (PSN, Fisher Scientific), apo-transferrin (5 µg/ml, Sigma, München, Germany), superoxid-dismutase (0.8 µg/ml, Sigma) and glutathione (1 µg/ml, Sigma)). Cells were always seeded on poly-ornithine (0.1 mg/ml, Sigma) and laminin (1 µg/ml, Sigma) wells of 24 well plates.

Lentiviral particles using pLKO1-shDdx5-puro, pLKO1-shFoxg1-puro-GFP or pLKO1-non-target-puro (Sigma) were prepared according to the protocol described previously [22, 23]. On day *in vitro* (DIV) 2 cells were transfected with lentiviral particles. At DIV5, transduced cells were selected with 0.3 µg/ml puromycin and cell proliferation was inhibited by addition of 2 µM AraC, while performing a half medium

change. Medium was changed again at DIV9 including 2 µM AraC. Cells were harvested at DIV11 in Qiazol reagent and used for RNA extraction.

BrdU (Bromodeoxyuridine) incorporation and immunofluorescence

For proliferation assays, N2A cells were transfected with the plasmids indicated in the figures and a 1 h BrdU pulse (Roche BrdU Kit) was given before fixation. Cells were fixed with 4% PFA for 20 min at room temperature. For the BrdU antigen retrieval, fixed cells were treated with 1N HCl during 30 min followed by two washes with Boratbuffer (150 mM H3BO3, pH 8,4) of 10 min each for neutralization. Cells were then permeabilised and blocked in 10% horse serum / 0.1% Triton-X100/PBS for 1 hour and incubation with anti-BrdU antibody (1:200, sheep, ab1893, abcam) was perfomed over night at 4°C in blocking solution. Cells were washed 3 times in PBS and then incubated with fluorophore-coupled secondary antibodies in blocking solution at room temperature. After 3 washes with PBS, cells were incubated for 1 min in DAPI solution and washed 3 more times in PBS. Coverslips were mounted on glass slides with fluorescent mounting medium (#S3023, DAKO, Jena, Germany). Images were obtained using an Axioplan M2 fluorescent microscope (Zeiss) and processed with FIJI (ImageJ, v. 2.0.0-rc-43/1.51d)[24][23][22][21][20][18][17] and , Inkscape (v. 0.91).

References

1. **Livak KJ, Schmittgen TD (2001) Analysis of Relative Gene Expression Data Using Real-Time Quantitative PCR and the 2−ΔΔCT Method. Methods 25:402–408. https://doi.org/10.1006/meth.2001.1262**

2. **Grüning BA, Fallmann J, Yusuf D, et al (2017) The RNA workbench: best practices for RNA and high-throughput sequencing bioinformatics in Galaxy. Nucleic Acids Res 45:W560–W566. https://doi.org/10.1093/nar/gkx409**

3. **Andrews S FastQC A Quality Control tool for High Throughput Sequence Data**

4. **Krueger F A wrapper tool around Cutadapt and FastQC to consistently apply quality and adapter trimming to FastQ files, with some extra functionality for MspI-digested RRBS-type (Reduced Representation Bisufite-Seq) libraries.**

5. **Kim D, Pertea G, Trapnell C, et al (2013) TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. Genome Biol 14:R36. https://doi.org/10.1186/gb-2013-14-4-r36**

6. **Yates A, Akanni W, Amode MR, et al (2015) Ensembl 2016. Nucleic Acids Res 44:D710. https://doi.org/10.1093/nar/gkv1157**

7. **Anders S, Pyl PT, Huber W (2014) HTSeq—a Python framework to work with high-throughput sequencing data. Bioinformatics 31:166. https://doi.org/10.1093/bioinformatics/btu638**

8. **Love MI, Huber W, Anders S (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol 15:550. https://doi.org/10.1186/s13059-014-0550-8**

9. **Mathelier A, Fornes O, Arenillas DJ, et al (2016) JASPAR 2016: a major expansion and update of the open-access database of transcription factor binding profiles. Nucleic Acids Res 44:D110–D115. https://doi.org/10.1093/nar/gkv1176**

10. **Grant CE, Bailey TL, Noble WS (2011) FIMO: scanning for occurrences of a given motif. Bioinformatics 27:1017–1018. https://doi.org/10.1093/bioinformatics/btr064**

11. **Huang DW, Sherman BT, Lempicki RA (2009) Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. Nucleic Acids Res 37:1–13. https://doi.org/10.1093/nar/gkn923**

12. **Huang DW, Sherman BT, Lempicki RA (2009) Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. Nat Protoc 4:44–57. https://doi.org/10.1038/nprot.2008.211**

13. **Lewis BP, Shih I -hun., Jones-Rhoades MW, et al (2003) Prediction of Mammalian MicroRNA Targets. Cell 115:787–798. https://doi.org/10.1016/S0092-8674(03)01018-3**

14. **Betel D, Wilson M, Gabow A, et al (2008) The microRNA.org resource: targets and expression. Nucleic Acids Res 36:D149–D153. https://doi.org/10.1093/nar/gkm995**

15. **Wong N, Wang X (2015) miRDB: an online resource for microRNA target prediction and functional annotations. Nucleic Acids Res 43:D146–D152. https://doi.org/10.1093/nar/gku1104**

16. **Griffiths-Jones S, Saini HK, van Dongen S, Enright AJ (2008) miRBase: tools for microRNA genomics. Nucleic Acids Res 36:D154–D158. https://doi.org/10.1093/nar/gkm952**

17. **Shevchenko A, Tomas H, Havli J, et al (2007) In-gel digestion for mass spectrometric characterization of proteins and proteomes. Nat Protoc 1:2856–2860. https://doi.org/10.1038/nprot.2006.468**

18. **Rappsilber J, Mann M, Ishihama Y (2007) Protocol for micro-purification, enrichment, pre-fractionation and storage of peptides for proteomics using StageTips. Nat Protoc 2:1896–1906. https://doi.org/10.1038/nprot.2007.261**

19. **Cox J, Mann M (2008) MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. Nat Biotechnol 26:1367–1372. https://doi.org/10.1038/nbt.1511**

20. **Vance KW, Sansom SN, Lee S, et al (2014) The long non□coding RNA Paupar regulates the expression of both local and distal genes. EMBO J e201386225. https://doi.org/10.1002/embj.201386225**

21. **Behler J, Sharma K, Wilde A, et al (2018) The host-encoded RNase E endonuclease as the maturation enzyme of a CRISPR-Cas subtype III-B system. Nature Microbiol in press**

22. **Hellbach N, Weise SC, Vezzali R, et al (2014) Neural deletion of Tgfbr2 impairs angiogenesis through an altered secretome. Hum Mol Genet 23:6177–6190. https://doi.org/10.1093/hmg/ddu338**

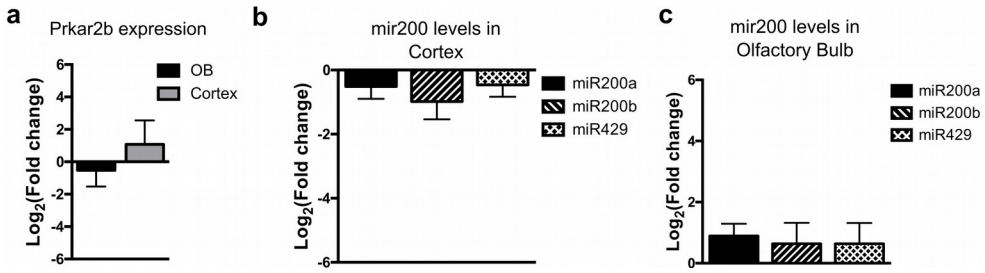
23. **Vezzali R, Weise SC, Hellbach N, et al (2016) The FOXG1/FOXO/SMAD network balances proliferation and differentiation of cortical progenitors and activates Kcnh3 expression in mature neurons. Oncotarget 5:**

24. **Schindelin J, Arganda-Carreras I, Frise E, et al (2012) Fiji: an open-source platform for biological-image analysis. Nat Methods 9:676–682. https://doi.org/10.1038/nmeth.2019**

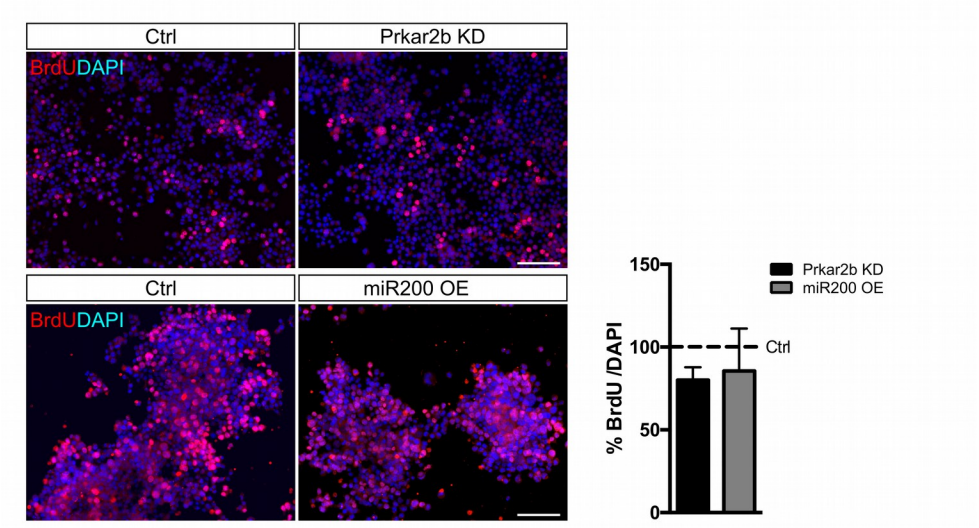
Supplementary Table S1: qRT-PCR Primers

Ddx5_forq	TCCAGAGGGCTAGATGTGGAA
Ddx5_revq	TGCCTGTTTTGGTACTGCGA
Enc1_forq	TACTGCATTCGTCAGCACCT
Enc1_revq	ATGACATCTCATCTCGCCGT
Foxg1_forq	AATGACTTCGCAGACCAGCA
Foxg1_revq	CCGGACAGTCCTGTCGTAAA
Gapdh_forq	CGGCCGCATCTTCTTGTG
Gapdh_revq	TGACCAGGCGCCCAATAC
Homer3_forq	CAGTCGAGCCAACACTGTCT
Homer3_revq	AGATTTCTCTCGAGCCAGCC
Itpka_forq	ATCTGCTGAGCGACAGTGAG
Itpka_revq	GCAGATTGACCATGGTACGG
Nrp2_forq	GACGATCGGGAGAGATTTCCA
Nrp2_revq	AATCCTCACCTGCAAAAGCTG
Olfml2b_forq	GCTGACAGCCGAGATTTGAAG
Olfml2b_revq	GTGGTAACCGAATGCAGCTT
pri-miR200a_forq	CGGACAGTGCTGGATTTCTT
pri-miR200a_revq	CAGGAGGACAAGTGTGTATCATC
pri-miR200b_forq	TGATCTCTAATACTGCCTGGTAATG
pri-miR200b_revq	CCATAGCCCTACCTTGGATAAG
pri-miR429_forq	CCTTCCCTCTACAGGTATCAAATC
pri-miR429_revq	GACGGCATTACCAGACAGTATTA
Prkar2b_forq	GCAAGAGGCTTGCAAAGACA
Prkar2b_revq	CGTGTTCCCCTTCTTTGACC
Serpinf1_forq	ACGGCTTGGACTCTGATCTC
Serpinf1_revq	TCAAGTTCTGGGTCACGGTC
Sqle_forq	TCACCATGGCCGATTCATCA
Sqle_revq	CCTTGTATTGCACGCCGATT
Stc2_forq	TTTCTGCACAACGCTGGAAA
Stc2_revq	CCAAATTTATGACGCAGGGCA
Tmem108_forq	GCATGCAGAAGTAAGCGTGT
Tmem108_revq	CCTGCTGGTCCTCCAGTTTA
Ybx3_forq	TACAGACGCGGCTACTATGG
Ybx3_revq	GGGCTCAAATCCTTCACTGC
Zdhhc15_forq	GAAGCAGATGCTTGTGGACAT
Zdhhc15_revq	TTGATTAGATGGCACCGGTCA
Zeb1_forq	CTGCTCCCTGTGCAGTTACA
Zeb1_revq	CTTGAACTTGCGGTTTCCCC

Zfp462_forq	GGCCACGGATTGTCAGTCTCC
Zfp462_revq	GGGGCTCCGAGTAAGAGGAGT



Supplementary Figure 1



Supplementary Figure 2

Supplementary material for

[P3] ChiRA: an integrated framework for chimeric read analysis from RNA-RNA interactome and RNA structurome data

ChiRA: an integrated framework for Chimeric Read Analysis from RNA-RNA interactome and RNA structurome data

Pavankumar Videm[1], Anup Kumar[1], Oleg Zharkov[1], Björn Andreas Grüning[1], and Rolf Backofen[1,2]

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

S1 Glossary of terms

This section gives a brief into the terminology used in the manuscript.

- *Read arm*: An aligned portion of the read.
- *Singleton read*: A read with only one arm. A single arm may multi-map to different positions on the reference.
- *Chimeric read*: A read with at least two non-overlapping arms. Each of these arms called as a *chimeric arm*.
- *Read segment*: A region on the read assembled by merging highly overlapping arms.
- *Expressed locus*: A region on the reference built from overlapping alignments. This can be achieved by a simple overlapping method or `blockbuster`'s Gaussian approximation of read coverage.
- *Common Read Loci*: A set of expressed loci that share most of their multi-mapped reads.

S2 Data and pre-processing

We applied `ChiRA` on CLASH, CLEAR-CLIP, PARIS and SPLASH datasets. Altogether 46 samples were analyzed using `ChiRA`. The details of the samples are in Table S1.

Protocol	Samples (GEO or SRA ids)	#	Organism	Characteristics
CLASH	GSM1219487, GSM1219488, GSM1219489, GSM1219490, GSM1219491, GSM1219492	6	human	Flp-In T-REx 293
CLEAR-CLIP	GSM1881516, GSM1881517, ..., GSM1881541	26	mouse	cortex CLEAR-CLIP
PARIS	GSM1917755, GSM1917756, GSM1917757, GSM1917758, GSM1917759, GSM1917760	6	human & mouse	HEK293T (3), mouse ES(3)
SPLASH	SRR3404939, SRR3404940, SRR3404941, SRR3404942, SRR3404943, SRR3404926, SRR3404927, SRR3404928	8	human	Lymphoblastoid Cells PolyA (4), H1 hES PolyA (2), H1 hES treated with retinoic acid Poly A (2)

Table S1: Summary of the samples analyzed using `ChiRA`

The whole analysis of the above mentioned samples was carried out on RNA workbench [1]. We used `fastq-dump` Galaxy tool [2] to fetch the fastq files from the sequence read archive (SRA) database. Then we removed adapters and low quality ends using `cutadapt` [3]. We followed the instructions from the papers and gene expression omnibus sample pages to reproduce the pre-processing step. For CLASH, we used 5′ adapter - ACACGACGCTCTTCCGATCT and 3′ adapter - TGGAATTCTCGGGTGCCAAGG. For the preprocessing of the CLEAR-CLIP data, we used 5′ adapter - NNNNAGGGAGGACGATGCGG and 3′ adapter - GTGTCAGTCACTTCCAGCGG. For PARIS datasets, 3′ adapter - AGATCGGAAGAGCGGTTCAG was used. The SPLASH samples in the SRA were already processed, hence we used them as they are. We filtered out the reads shorter than 16 nucleotides (*–minimum-length 16*) and trimmed bases from read ends with a Phred quality score less than 20 (*–quality-cutoff 20*). For all CLEAR-CLIP samples a 5 bases long 5′ degenerate linker, for CLASH GSM1219491 sample a 8nt long barcode, and for GSM1219492 sample a 9nt long barcode were stripped while deduplicating using `ChiRA-collapse` tool (*–umi_len N*).

S3 Calculation of Transcripts per Million

A the end of the EM algorithm, for each CRL c, we re-calculate the absolute abundances of as ${\rho'}_c = \sum_s Pr[z_{s,c} = 1|\hat{\rho}, Y]$. of abundance of c to the length of c in kilobases. This measure is generally known as reads per kilobase (RPK). As each CRL is a set of expressed loci, a CRL cannot have a single length. Hence we take the median of the lengths of loci l belong to the CRL c as the length of the CRL.

$$RPK_c = 10^3 \frac{{\rho'}_c}{\text{median}_{l \in c}\{length(l)\}}$$

Then transcripts per million calculated as relative RPK of a CRL compared to the total RPK of the sample in millions.

$$TPM_c = 10^6 \frac{RPK_c}{\sum_{c'} RPK_{c'}}$$

S4 Data availability

We carried the whole `ChiRA` based analysis on the RNA workbench. Hence all the data can easily be accessible and each parameter of all the tool runs can be traced back. Table S2 contains the links to Galaxy histories and their corresponding workflows used. The description of the parameters that were changed from the defaults are described in Section S2. All the workflows and histories can be imported and adapted.

Data	**History**	**Workflow**
CLASH	https://rna.usegalaxy.eu/u/videmp/h/clash-analysis	https://rna.usegalaxy.eu/u/videmp/w/chira-clash-clearclip
CLEAR-CLIP	https://rna.usegalaxy.eu/u/videmp/h/clear-clip-analysis	https://rna.usegalaxy.eu/u/videmp/w/chira-clash-clearclip
PARIS	https://rna.usegalaxy.eu/u/videmp/h/paris-analysis	https://rna.usegalaxy.eu/u/videmp/w/paris-analysis
SPLASH	https://rna.usegalaxy.eu/u/videmp/h/splash-analysis	https://rna.usegalaxy.eu/u/videmp/w/splash-analysis
Benchmark using `BWA-MEM`	https://rna.usegalaxy.eu/u/videmp/h/benchmark-chira-bwa	https://rna.usegalaxy.eu/u/videmp/w/benchmark-chira-bwa
Benchmark using `CLAN`	https://rna.usegalaxy.eu/u/videmp/h/benchmark-chira-clan	https://rna.usegalaxy.eu/u/videmp/w/benchmark-chira-clan

Table S2: Links to the analysis histories and the wokflows

S5 CRL validation

The complete sample-wise CRLs information is given in Table S3. The column description is as follows. PSI: column shoes the average percentage sequence identity among the loci within the CRLs of each sample. #qualified CRLs: Number of CRLs per sample used in validating CRLs. A qualified CRL must contain at least 2 distinct genomic loci that associate with different genes. The genes that are associated with the loci within a the associated genes must exist in the annotation. %explained by databases: Average percentage of genes per CRL that are in agreement with the Ensembl protein family or kegg pathways information. #reads involved: Number of unique reads that are involved in CRLs. This is the number of reads that are possibly rescued by creating CRLs. Note that these are the deduplicated reads. Compared to these deduplicated reads, the number of initial sequenced reads that resulted in these can sometimes be in orders of magnitude.

Protocol	Sample	APSI	#qualified CRLs	%explained by databases	#reads involved
CLASH	GSM1219487	94.29	894	74.15	68724
	GSM1219488	93.20	599	79.44	53078
	GSM1219489	92.62	675	75.53	81176
	GSM1219490	92.33	1763	80.36	227305
	GSM1219491	90.82	1075	81.06	295311
	GSM1219492	90.75	979	80.06	272882
CLEAR-CLIP	GSM1881516	92.43	269	62.61	23813
	GSM1881517	88.99	1026	33.13	42469
	GSM1881518	91.77	330	55.94	26294
	GSM1881519	91.07	318	51.32	24299
	GSM1881520	91.24	211	58.48	19220
	GSM1881521	94.35	255	70.90	17943
	GSM1881522	91.07	318	51.32	24299
	GSM1881523	95.32	262	55.25	15475
	GSM1881524	94.83	113	57.10	5546
	GSM1881525	90.72	838	31.02	34595
	GSM1881526	90.39	289	43.11	18574
	GSM1881527	89.50	668	37.87	28075
	GSM1881528	90.90	316	51.49	23608
	GSM1881529	90.08	479	51.47	40234
	GSM1881530	88.67	847	39.70	28228
	GSM1881531	95.60	113	58.81	3366
	GSM1881532	94.88	216	49.24	5280
	GSM1881533	95.15	149	59.31	5972
	GSM1881534	94.45	142	51.52	7072
	GSM1881535	93.98	135	53.92	3827
	GSM1881536	95.58	186	49.02	5130
	GSM1881537	94.75	165	45.85	6789
	GSM1881538	94.72	156	56.31	6449
	GSM1881539	94.92	90	59.93	7252
	GSM1881540	94.44	112	64.49	6778
	GSM1881541	95.16	111	54.77	7955
PARIS	GSM1917755	95.27	523	77.33	194356
	GSM1917756	95.66	584	73.98	177179
	GSM1917757	96.14	1505	50.44	394708
	GSM1917758	97.64	650	79.75	89945
	GSM1917759	96.94	709	73.09	141218
	GSM1917760	97.23	632	79.01	129261
SPLASH	SRR3404943	92.65	866	65.63	129261
	SRR3404926	93.32	324	63.48	108724
	SRR3404927	93.14	792	65.21	182786
	SRR3404928	93.71	343	60.00	83129
	SRR3404939	92.37	618	54.12	211750
	SRR3404940	92.89	592	68.33	176528
	SRR3404941	94.20	50	62.35	11714
	SRR3404942	93.12	201	57.25	125647

Table S3: Sample-wise summary of the numbers related to Figures 4 and 5

References

[1] J. Fallmann, P. Videm, A. Bagnacani, B. Batut, M. A. Doyle, T. Klingstrom, F. Eggenhofer, P. F. Stadler, R. Backofen, and B. Grüning, "The rna workbench 2.0: next generation rna data analysis," *Nucleic acids research*, vol. 47, no. W1, pp. W511–W515, 2019.

[2] R. Leinonen, H. Sugawara, and M. S. and, "The sequence read archive," *Nucleic Acids Research*, vol. 39, pp. D19–D21, nov 2010.

[3] M. Martin, "Cutadapt removes adapter sequences from high-throughput sequencing reads," *EMBnet. journal*, vol. 17, no. 1, pp. 10–12, 2011.

Bibliography

[1] Francis Crick. Central dogma of molecular biology. *Nature*, 227(5258):561–563, 1970. (Cited on pages 3 and 9.)

[2] Walter Gilbert. Origin of life: The RNA world. *nature*, 319(6055):618–618, 1986. (Cited on page 3.)

[3] ENCODE Project Consortium et al. The ENCODE (ENCyclopedia of DNA elements) project. *Science*, 306(5696):636–640, 2004. (Cited on pages 3 and 11.)

[4] Manel Esteller. Non-coding RNAs in human disease. *Nature reviews genetics*, 12(12):861–874, 2011. (Cited on page 3.)

[5] 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 3 and 19.)

[6] John S Mattick and Igor V Makunin. Non-coding RNA. *Human molecular genetics*, 15(suppl_1):R17–R29, 2006. (Cited on page 11.)

[7] Paulo P Amaral and John S Mattick. Noncoding RNA in development. *Mammalian genome*, 19(7-8):454–492, 2008. (Cited on page 11.)

[8] Anthony K Henras, Christophe Dez, and Yves Henry. RNA structure and function in C/D and H/ACA s (no) RNPs. *Current opinion in structural biology*, 14(3):335–343, 2004. (Cited on page 12.)

[9] Stefanie A Mortimer, Mary Anne Kidwell, and Jennifer A Doudna. Insights into RNA structure and function from genome-wide studies. *Nature Reviews Genetics*, 15(7):469–479, 2014. (Cited on page 12.)

[10] Michael Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic acids research*, 31(13):3406–3415, 2003. (Cited on page 12.)

[11] Ivo L Hofacker. Vienna RNA secondary structure server. *Nucleic acids research*, 31(13):3429–3431, 2003. (Cited on page 12.)

[12] Jessica S Reuter and David H Mathews. RNAstructure: software for RNA secondary structure prediction and analysis. *BMC bioinformatics*, 11(1):1–9, 2010. (Cited on page 12.)

[13] Kevin C Miranda, Tien Huynh, Yvonne Tay, Yen-Sin Ang, Wai-Leong Tam, Andrew M Thomson, Bing Lim, and Isidore Rigoutsos. A pattern-based method for the identification of MicroRNA binding sites and their corresponding heteroduplexes. *Cell*, 126(6):1203–1217, 2006. (Cited on page 13.)

[14] Inhan Lee, Subramanian S Ajay, Jong In Yook, Hyun Sil Kim, Su Hyung Hong, Nam Hee Kim, Saravana M Dhanasekaran, Arul M Chinnaiyan, and Brian D Athey. New class of microRNA targets containing simultaneous 5'-UTR and 3'-UTR interaction sites. *Genome research*, 19(7):1175–1183, 2009. (Cited on page 13.)

[15] Isidore Rigoutsos. New tricks for animal microRNAS: targeting of amino acid coding regions at conserved and nonconserved sites. *Cancer research*, 69(8):3245–3248, 2009. (Cited on page 13.)

[16] Sung Wook Chi, Gregory J Hannon, and Robert B Darnell. An alternative mode of microRNA target recognition. *Nature structural & molecular biology*, 19(3):321, 2012. (Cited on page 13.)

[17] 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 pages 13, 19 and 22.)

[18] Hilary C Martin, Shivangi Wani, Anita L Steptoe, Keerthana Krishnan, Katia Nones, Ehsan Nourbakhsh, Alexander Vlassov, Sean M Grimmond, and Nicole Cloonan. Imperfect centered miRNA binding sites are common and can mediate repression of target mRNAs. *Genome biology*, 15(3):1–22, 2014. (Cited on page 13.)

[19] Chanseok Shin, Jin-Wu Nam, Kyle Kai-How Farh, H Rosaria Chiang, Alena Shkumatava, and David P Bartel. Expanding the microRNA targeting code: functional sites with centered pairing. *Molecular cell*, 38(6):789–802, 2010. (Cited on page 13.)

[20] 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 13.)

[21] Ulrike Mückstein, Hakim Tafer, Jörg Hackermüller, Stephan H Bernhart, Peter F Stadler, and Ivo L Hofacker. Thermodynamics of RNA–RNA binding. *Bioinformatics*, 22(10):1177–1182, 2006. (Cited on page 13.)

[22] Rosalind C Lee, Rhonda L Feinbaum, and Victor Ambros. The C. elegans heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14. *cell*, 75(5):843–854, 1993. (Cited on page 14.)

[23] Brenda J Reinhart, Frank J Slack, Michael Basson, Amy E Pasquinelli, Jill C Bettinger, Ann E Rougvie, H Robert Horvitz, and Gary Ruvkun. The 21-nucleotide let-7 RNA regulates developmental timing in Caenorhabditis elegans. *nature*, 403(6772):901–906, 2000. (Cited on page 14.)

[24] Nelson C Lau, Lee P Lim, Earl G Weinstein, and David P Bartel. An abundant class of tiny RNAs with probable regulatory roles in Caenorhabditis elegans. *Science*, 294(5543):858–862, 2001. (Cited on page 14.)

[25] Matthew W Jones-Rhoades, David P Bartel, and Bonnie Bartel. MicroRNAs and their regulatory roles in plants. *Annu. Rev. Plant Biol.*, 57:19–53, 2006. (Cited on page 14.)

[26] Julius Brennecke, David R Hipfner, Alexander Stark, Robert B Russell, and Stephen M Cohen. bantam encodes a developmentally regulated microRNA that controls cell proliferation and regulates the proapoptotic gene hid in Drosophila. *Cell*, 113(1):25–36, 2003. (Cited on page 14.)

[27] Chang-Zheng Chen, Ling Li, Harvey F Lodish, and David P Bartel. MicroRNAs modulate hematopoietic lineage differentiation. *science*, 303(5654):83–86, 2004. (Cited on page 14.)

[28] Yong Peng and Carlo M Croce. The role of MicroRNAs in human cancer. *Signal transduction and targeted therapy*, 1(1):1–9, 2016. (Cited on page 14.)

[29] Temo Barwari, Abhishek Joshi, and Manuel Mayr. MicroRNAs in cardiovascular disease. *Journal of the American College of Cardiology*, 68(23):2577–2584, 2016. (Cited on page 14.)

[30] Karla F Meza-Sosa, David Valle-García, Gustavo Pedraza-Alva, and Leonor Pérez-Martínez. Role of microRNAs in central nervous system development and pathology. *Journal of neuroscience research*, 90(1):1–12, 2012. (Cited on page 14.)

[31] Sam Griffiths-Jones, Jerome HL Hui, Antonio Marco, and Matthew Ronshaugen. MicroRNA evolution by arm switching. *EMBO reports*, 12(2):172–177, 2011. (Cited on pages 14 and 64.)

[32] Jr-Shiuan Yang, Michael D Phillips, Doron Betel, Ping Mu, Andrea Ventura, Adam C Siepel, Kevin C Chen, and Eric C Lai. Widespread regulatory activity of vertebrate microRNA* species. *Rna*, 17(2):312–326, 2011. (Cited on page 14.)

[33] Sam Griffiths-Jones, Russell J Grocock, Stijn Van Dongen, Alex Bateman, and Anton J Enright. miRBase: microRNA sequences, targets and gene nomenclature. *Nucleic acids research*, 34(suppl_1):D140–D144, 2006. (Cited on page 14.)

[34] Julius Brennecke, Alexander Stark, Robert B Russell, and Stephen M Cohen. Principles of microRNA–target recognition. *PLoS biol*, 3(3):e85, 2005. (Cited on page 14.)

[35] Marc Robert Fabian, Nahum Sonenberg, and Witold Filipowicz. Regulation of mRNA translation and stability by microRNAs. *Annual review of biochemistry*, 79:351–379, 2010. (Cited on page 15.)

[36] Ana Eulalio, Eric Huntzinger, Tadashi Nishihara, Jan Rehwinkel, Maria Fauser, and Elisa Izaurralde. Deadenylation is a widespread effect of miRNA regulation. *Rna*, 15(1):21–32, 2009. (Cited on page 15.)

[37] Sergej Djuranovic, Ali Nahvi, and Rachel Green. miRNA-mediated gene silencing by translational repression followed by mRNA deadenylation and decay. *Science*, 336(6078):237–240, 2012. (Cited on page 15.)

[38] Yong Sun Lee, Yoshiyuki Shibata, Ankit Malhotra, and Anindya Dutta. A novel class of small RNAs: tRNA-derived RNA fragments (tRFs). *Genes & development*, 23(22):2639–2649, 2009. (Cited on page 16.)

[39] Thoru Pederson. Regulatory RNAs derived from transfer RNA? *Rna*, 16(10):1865–1869, 2010. (Cited on page 16.)

[40] Andrew Sobala and Gyorgy Hutvagner. Transfer RNA-derived fragments: origins, processing, and functions. *Wiley Interdisciplinary Reviews: RNA*, 2(6):853–862, 2011. (Cited on page 16.)

[41] Jennifer Gebetsberger and Norbert Polacek. Slicing tRNAs to boost functional ncRNA diversity. *RNA biology*, 10(12):1798–1806, 2013. (Cited on page 16.)

[42] Pankaj Kumar, Canan Kuscu, and Anindya Dutta. Biogenesis and function of transfer RNA-related fragments (tRFs). *Trends in biochemical sciences*, 41(8):679–689, 2016. (Cited on page 16.)

[43] Pankaj Kumar, Jordan Anaya, Suresh B Mudunuri, and Anindya Dutta. Meta-analysis of tRNA derived RNA fragments reveals that they are evolutionarily conserved and associate with AGO proteins to recognize specific RNA targets. *BMC biology*, 12(1):78, 2014. (Cited on page 16.)

[44] Canan Kuscu, Pankaj Kumar, Manjari Kiran, Zhangli Su, Asrar Malik, and Anindya Dutta. tRNA fragments (tRFs) guide Ago to regulate gene expression post-transcriptionally in a Dicer-independent manner. *Rna*, 24(8):1093–1105, 2018. (Cited on page 16.)

[45] Paul Anderson and Pavel Ivanov. tRNA fragments in human health and disease. *FEBS letters*, 588(23):4297–4304, 2014. (Cited on page 16.)

[46] Mark Helm. Post-transcriptional nucleotide modification and alternative folding of RNA. *Nucleic acids research*, 34(2):721–733, 2006. (Cited on pages 16 and 35.)

[47] Debrah M Thompson, Cheng Lu, Pamela J Green, and Roy Parker. tRNA cleavage is a conserved response to oxidative stress in eukaryotes. *Rna*, 14(10):2095–2103, 2008. (Cited on page 16.)

[48] Tamás Kiss. Small nucleolar RNAs: an abundant group of noncoding RNAs with diverse cellular functions. *Cell*, 109(2):145–148, 2002. (Cited on page 18.)

[49] Andrey G Balakin, Laurie Smith, and Maurille J Fournier. The RNA world of the nucleolus: two major families of small RNAs defined by different box elements with related functions. *cell*, 86(5):823–834, 1996. (Cited on page 18.)

[50] Ashesh A Saraiya and Ching C Wang. snoRNA, a novel precursor of microRNA in Giardia lamblia. *PLoS Pathog*, 4(11):e1000224, 2008. (Cited on page 18.)

[51] Christine Ender, Azra Krek, Marc R Friedländer, Michaela Beitzinger, Lasse Weinmann, Wei Chen, Sébastien Pfeffer, Nikolaus Rajewsky, and Gunter Meister. A human snoRNA with microRNA-like functions. *Molecular cell*, 32(4):519–528, 2008. (Cited on page 18.)

[52] Ryan J Taft, Evgeny A Glazov, Timo Lassmann, Yoshihide Hayashizaki, Piero Carninci, and John S Mattick. Small RNAs derived from snoRNAs. *Rna*, 15(7):1233–1240, 2009. (Cited on page 18.)

[53] Marina Falaleeva and Stefan Stamm. Processing of snoRNAs as a new source of regulatory non-coding RNAs: snoRNA fragments form a new class of functional RNAs. *Bioessays*, 35(1):46–54, 2013. (Cited on page 18.)

[54] A Gregory Matera, Rebecca M Terns, and Michael P Terns. Non-coding RNAs: lessons from the small nuclear and small nucleolar RNAs. *Nature reviews Molecular cell biology*, 8(3):209–220, 2007. (Cited on page 18.)

[55] Snehal Bhikhu Patel and Michel Bellini. The assembly of a spliceosomal small nuclear ribonucleoprotein particle. *Nucleic acids research*, 36(20):6482–6493, 2008. (Cited on page 18.)

[56] Tamás Kiss. Biogenesis of small nuclear RNPs. *Journal of cell science*, 117(25):5949–5951, 2004. (Cited on page 18.)

[57] Cindy L Will and Reinhard Lührmann. Spliceosome structure and function. *Cold Spring Harbor perspectives in biology*, 3(7):a003707, 2011. (Cited on page 18.)

[58] Lilian T C França, Emanuel Carrilho, and Tarso B L Kist. A review of DNA sequencing techniques. *Quarterly reviews of biophysics*, 35(2):169, 2002. (Cited on page 19.)

[59] Frederick Sanger, Steven Nicklen, and Alan R Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of the national academy of sciences*, 74(12):5463–5467, 1977. (Cited on page 19.)

[60] Allan M Maxam and Walter Gilbert. A new method for sequencing DNA. *Proceedings of the National Academy of Sciences*, 74(2):560–564, 1977. (Cited on page 19.)

[61] Frederick Sanger, Gilian M Air, Bart G Barrell, Nigel L Brown, Alan R Coulson, J_ C_ Fiddes, CA Hutchison, Patrick M Slocombe, and Mo Smith. Nucleotide sequence of bacteriophage φX174 DNA. *nature*, 265(5596):687–695, 1977. (Cited on page 19.)

[62] J Craig Venter, Mark D Adams, Eugene W Myers, Peter W Li, Richard J Mural, Granger G Sutton, Hamilton O Smith, Mark Yandell, Cheryl A Evans, Robert A Holt, et al. The sequence of the human genome. *science*, 291(5507):1304–1351, 2001. (Cited on page 19.)

[63] Michael L. Metzker. Sequencing technologies – the next generation. *Nat Rev Genet*, 11(1):31–46, Jan 2010. (Cited on page 19.)

[64] Ali Mortazavi, Brian A Williams, Kenneth McCue, Lorian Schaeffer, and Barbara Wold. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature methods*, 5(7):621–628, 2008. (Cited on pages 19 and 20.)

[65] David S Johnson, Ali Mortazavi, Richard M Myers, and Barbara Wold. Genome-wide mapping of in vivo protein-DNA interactions. *Science*, 316(5830):1497–1502, 2007. (Cited on page 19.)

[66] Ina Huppertz, Jan Attig, Andrea D'Ambrogio, Laura E Easton, Christopher R Sibley, Yoichiro Sugimoto, Mojca Tajnik, Julian König, and Jernej Ule. iCLIP: protein–RNA interactions at nucleotide resolution. *Methods*, 65(3):274–287, 2014. (Cited on page 19.)

[67] 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 page 19.)

[68] Grzegorz Kudla, Sander Granneman, Daniela Hahn, Jean D Beggs, and David Tollervey. Cross-linking, ligation, and sequencing of hybrids reveals RNA–RNA interactions in yeast. *Proceedings of the National Academy of Sciences*, 108(24):10010–10015, 2011. (Cited on pages 19 and 22.)

[69] 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 19.)

[70] Zhipeng Lu, Qiangfeng Cliff Zhang, Byron Lee, Ryan A Flynn, Martin A Smith, James T Robinson, Chen Davidovich, Anne R Gooding, Karen J Goodrich, John S Mattick, et al. RNA duplex map in living cells reveals higher-order transcriptome structure. *Cell*, 165(5):1267–1279, 2016. (Cited on pages 19 and 22.)

[71] Jong Ghut Ashley Aw, Yang Shen, Andreas Wilm, Miao Sun, Xin Ni Lim, Kum-Loong Boon, Sidika Tapsin, Yun-Shen Chan, Cheng-Peow Tan, Adelene YL Sim, et al. In vivo mapping of eukaryotic RNA interactomes reveals principles of higher-order organization and regulation. *Molecular cell*, 62(4):603–617, 2016. (Cited on pages 19 and 22.)

[72] David J Duggan, Michael Bittner, Yidong Chen, Paul Meltzer, and Jeffrey M Trent. Expression profiling using cDNA microarrays. *Nature genetics*, 21(1):10–14, 1999. (Cited on page 20.)

[73] Christine Debouck and Peter N Goodfellow. DNA microarrays in drug discovery and development. *Nature genetics*, 21(1):48–50, 1999. (Cited on page 20.)

[74] Jonathan R Pollack, Charles M Perou, Ash A Alizadeh, Michael B Eisen, Alexander Pergamenschikov, Cheryl F Williams, Stefanie S Jeffrey, David Botstein, and Patrick O Brown. Genome-wide analysis of DNA copy-number changes using cDNA microarrays. *Nature genetics*, 23(1):41–46, 1999. (Cited on page 20.)

[75] John C Marioni, Christopher E Mason, Shrikant M Mane, Matthew Stephens, and Yoav Gilad. RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome research*, 18(9):1509–1517, 2008. (Cited on page 20.)

[76] Z. Wang, M. Gerstein, and M. Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63, 2009. (Cited on page 20.)

[77] Ugrappa Nagalakshmi, Zhong Wang, Karl Waern, Chong Shou, Debasish Raha, Mark Gerstein, and Michael Snyder. The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science*, 320(5881):1344–1349, 2008. (Cited on page 20.)

[78] Ryan Lister, Ronan C O'Malley, Julian Tonti-Filippini, Brian D Gregory, Charles C Berry, A Harvey Millar, and Joseph R Ecker. Highly integrated single-base resolution maps of the epigenome in Arabidopsis. *Cell*, 133(3):523–536, 2008. (Cited on page 20.)

[79] Peng Cui, Qiang Lin, Feng Ding, Chengqi Xin, Wei Gong, Lingfang Zhang, Jianing Geng, Bing Zhang, Xiaomin Yu, Jin Yang, et al. A comparison between ribo-minus RNA-sequencing and polyA-selected RNA-sequencing. *Genomics*, 96(5):259–265, 2010. (Cited on page 20.)

[80] Ryan D Morin, Michael D O'Connor, Malachi Griffith, Florian Kuchenbauer, Allen Delaney, Anna-Liisa Prabhu, Yongjun Zhao, Helen McDonald, Thomas Zeng, Martin Hirst, et al. Application of massively parallel sequencing to microRNA profiling and discovery in human embryonic stem cells. *Genome research*, 18(4):610–621, 2008. (Cited on page 20.)

[81] Susanne Motameny, Stefanie Wolters, Peter Nürnberg, and Björn Schumacher. Next generation sequencing of miRNAs–strategies, resources and methods. *Genes*, 1(1):70–84, 2010. (Cited on page 20.)

[82] Sung Wook Chi, Julie B Zang, Aldo Mele, and Robert B Darnell. Argonaute HITS-CLIP decodes microRNA–mRNA interaction maps. *Nature*, 460(7254):479–486, 2009. (Cited on page 22.)

[83] 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 page 22.)

[84] Eesha Sharma, Tim Sterne-Weiler, Dave O'Hanlon, and Benjamin J Blencowe. Global mapping of human RNA-RNA interactions. *Molecular cell*, 62(4):618–626, 2016. (Cited on page 22.)

[85] Robert A Holt and Steven JM Jones. The new paradigm of flow cell sequencing. *Genome research*, 18(6):839–846, 2008. (Cited on page 22.)

[86] Peter JA Cock, Christopher J Fields, Naohisa Goto, Michael L Heuer, and Peter M Rice. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic acids research*, 38(6):1767–1771, 2010. (Cited on page 24.)

[87] Simon Andrews et al. FastQC: a quality control tool for high throughput sequence data, 2010. (Cited on pages 25 and 44.)

[88] Liguo Wang, Shengqin Wang, and Wei Li. RSeQC: quality control of RNA-seq experiments. *Bioinformatics*, 28(16):2184–2185, 2012. (Cited on page 25.)

[89] Cristian Del Fabbro, Simone Scalabrin, Michele Morgante, and Federico M Giorgi. An extensive evaluation of read trimming effects on Illumina NGS data analysis. *PloS one*, 8(12):e85024, 2013. (Cited on page 25.)

[90] Claire R Williams, Alyssa Baccarella, Jay Z Parrish, and Charles C Kim. Trimming of sequence reads alters RNA-Seq gene expression estimates. *BMC bioinformatics*, 17(1):103, 2016. (Cited on page 25.)

[91] Marcel Martin. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet. journal*, 17(1):10–12, 2011. (Cited on pages 25 and 48.)

[92] Anthony M Bolger, Marc Lohse, and Bjoern Usadel. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, 30(15):2114–2120, 2014. (Cited on page 25.)

[93] 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 26.)

[94] Steve Hoffmann, Christian Otto, Stefan Kurtz, Cynthia M Sharma, Philipp Khaitovich, Jörg Vogel, Peter F Stadler, and Jörg Hackermüller. Fast mapping of short sequences with mismatches, insertions and deletions using index structures. *PLoS Comput Biol*, 5(9):e1000502, 2009. (Cited on page 27.)

[95] Heng Li. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv preprint arXiv:1303.3997*, 2013. (Cited on pages 27 and 48.)

[96] 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 27.)

[97] Daehwan Kim, Geo Pertea, Cole Trapnell, Harold Pimentel, Ryan Kelley, and Steven L Salzberg. TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome biology*, 14(4):R36, 2013. (Cited on pages 27 and 44.)

[98] Cuncong Zhong and Shaojie Zhang. CLAN: the CrossLinked reads ANalyais tool. *bioRxiv*, page 233841, 2017. (Cited on pages 27 and 48.)

[99] Simon Anders, Paul Theodor Pyl, and Wolfgang Huber. HTSeq-a Python framework to work with high-throughput sequencing data. *Bioinformatics*, 31(2):166–169, 2015. (Cited on pages 28 and 44.)

[100] Yang Liao, Gordon K Smyth, and Wei Shi. featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics*, 30(7):923–930, 2014. (Cited on page 28.)

[101] Bo Li and Colin N Dewey. RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC bioinformatics*, 12(1):323, 2011. (Cited on page 28.)

[102] Adam Roberts and Lior Pachter. Streaming fragment assignment for real-time analysis of sequencing experiments. *Nature methods*, 10(1):71–73, 2013. (Cited on page 28.)

[103] Lior Pachter. Models for transcript quantification from RNA-Seq. *arXiv preprint arXiv:1104.3889*, 2011. (Cited on page 28.)

[104] D. Haussler. Convolution Kernels on Discrete Structures. Technical Report UCS-CRL-99-10, University of California at Santa Cruz, Santa Cruz, CA, USA, 1999. (Cited on page 29.)

[105] Pavankumar Videm, Dominic Rose, Fabrizio Costa, and Rolf Backofen. BlockClust: efficient clustering and classification of non-coding RNAs from short read RNA-seq profiles. *Bioinformatics*, 30(12):i274–i282, 2014. (Cited on page 35.)

[106] E Birney, JA Stamatoyannopoulos, A Dutta, R Guigó, TR Gingeras, EH Margulies, Z Weng, M Snyder, ET Dermitzakis, and RE Thurman. Baylor College of Medicine Human Genome Sequencing Center; Washington University Genome Sequencing Center; Broad Institute; Children's Hospital Oakland Research Institute.(2007). Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature*, 447(7146):799–816, 2007. (Cited on page 35.)

[107] Aarron T Willingham and Thomas R Gingeras. TUF love for "junk" DNA. *Cell*, 125(7):1215–1220, 2006. (Cited on page 35.)

[108] Elena Rivas and Sean R Eddy. Noncoding RNA gene detection using comparative sequence analysis. *BMC bioinformatics*, 2(1):1–19, 2001. (Cited on page 35.)

[109] Stefan Washietl, Ivo L Hofacker, and Peter F Stadler. Fast and reliable prediction of noncoding RNAs. *Proceedings of the National Academy of Sciences*, 102(7):2454–2459, 2005. (Cited on page 35.)

[110] Sven Findeiß, David Langenberger, Peter F Stadler, and Steve Hoffmann. Traces of post-transcriptional RNA modifications in deep sequencing data. *Biological chemistry*, 392(4):305–313, 2011. (Cited on page 35.)

[111] Bradley E Bernstein, John A Stamatoyannopoulos, Joseph F Costello, Bing Ren, Aleksandar Milosavljevic, Alexander Meissner, Manolis Kellis, Marco A Marra, Arthur L Beaudet, Joseph R Ecker, et al. The NIH roadmap epigenomics mapping consortium. *Nature biotechnology*, 28(10):1045–1048, 2010. (Cited on pages 35 and 36.)

[112] W James Kent, Charles W Sugnet, Terrence S Furey, Krishna M Roskin, Tom H Pringle, Alan M Zahler, and David Haussler. The human genome browser at UCSC. *Genome research*, 12(6):996–1006, 2002. (Cited on page 35.)

[113] Hai Yang Hu, Song Guo, Jiang Xi, Zheng Yan, Ning Fu, Xiaoyu Zhang, Corinna Menzel, Hongyu Liang, Hongyi Yang, Min Zhao, et al. MicroRNA expression and regulation in human, chimpanzee, and macaque brains. *PLoS Genet*, 7(10):e1002327, 2011. (Cited on page 36.)

[114] David Langenberger, CI Bermudez-Santana, Peter F Stadler, and Steve Hoffmann. Identification and classification of small RNAs in transcriptome sequence data. In *Biocomputing 2010*, pages 80–87. World Scientific, 2010. (Cited on page 37.)

[115] Mario Fasold, David Langenberger, Hans Binder, Peter F Stadler, and Steve Hoffmann. DARIO: a ncRNA detection and analysis tool for next-generation sequencing experiments. *Nucleic acids research*, 39(suppl_2):W112–W117, 2011. (Cited on page 37.)

[116] David Langenberger, Sachin Pundhir, Claus T Ekstrøm, Peter F Stadler, Steve Hoffmann, and Jan Gorodkin. deepBlockAlign: a tool for aligning RNA-seq profiles of read block patterns. *Bioinformatics*, 28(1):17–24, 2012. (Cited on page 37.)

[117] Fabrizio Costa and Kurt De Grave. Fast Neighborhood Subgraph Pairwise Distance Kernel. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 26th International Conference on Machine Learning (ICML 2010)*, pages 255–262, Haifa, Israel, June 2010. Omnipress. (Cited on page 37.)

[118] Stijn Van Dongen. Graph clustering via a discrete uncoupling process. *SIAM Journal on Matrix Analysis and Applications*, 30(1):121–141, 2008. (Cited on page 38.)

[119] Stefan C Weise, Ganeshkumar Arumugam, Alejandro Villarreal, Pavankumar Videm, Stefanie Heidrich, Nils Nebel, Verónica I Dumit, Farahnaz Sananbenesi, Viktoria Reimann, Madeline Craske, et al. FOXG1 regulates PRKAR2B transcriptionally and posttranscriptionally via miR200 in the adult hippocampus. *Molecular neurobiology*, 56(7):5188–5201, 2019. (Cited on page 43.)

[120] Pavankumar Videm, Anup Kumar, Oleg Zharkov, Björn Andreas Grüning, and Rolf Backofen. ChiRA: an integrated framework for chimeric read analysis from RNA-RNA interactome and RNA structurome data. *GigaScience*, 10(2), 01 2021. giaa158. (Cited on page 43.)

[121] Enis Afgan, Dannon Baker, Marius Van den Beek, Daniel Blankenberg, Dave Bouvier, Martin Čech, John Chilton, Dave Clements, Nate Coraor, Carl Eberhard, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic acids research*, 44(W1):W3–W10, 2016. (Cited on pages 44 and 56.)

[122] Felix Krueger. Trim Galore: a wrapper tool around Cutadapt and FastQC to consistently apply quality and adapter trimming to FastQ files, with some extra functionality for MspI-digested RRBS-type (Reduced Representation Bisufite-Seq) libraries. *URL http://www. bioinformatics. babraham. ac. uk/projects/trim_galore/.(Date of access: 28/04/2016)*, 2012. (Cited on page 44.)

[123] Andrew Yates, Wasiu Akanni, M Ridwan Amode, Daniel Barrell, Konstantinos Billis, Denise Carvalho-Silva, Carla Cummins, Peter Clapham, Stephen Fitzgerald, Laurent Gil, et al. Ensembl 2016. *Nucleic acids research*, 44(D1):D710–D716, 2016. (Cited on page 44.)

[124] Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome biology*, 15(12):550, 2014. (Cited on page 44.)

[125] Martine Manuel, Ben Martynoga, Tian Yu, John D West, John O Mason, and David J Price. The transcription factor Foxg1 regulates the competence of telencephalic cells to adopt subpallial fates in mice. *Development*, 137(3):487–497, 2010. (Cited on page 45.)

[126] Martine N Manuel, Ben Martynoga, Mike D Molinek, Jane C Quinn, Corinne Kroemmer, John O Mason, and David J Price. The transcription factor Foxg1 regulates telencephalic progenitor proliferation cell autonomously, in part by controlling Pax6 expression levels. *Neural development*, 6(1):1–12, 2011. (Cited on page 45.)

[127] Anthony Mathelier, Oriol Fornes, David J Arenillas, Chih-yu Chen, Grégoire Denay, Jessica Lee, Wenqiang Shi, Casper Shyr, Ge Tan, Rebecca Worsley-Hunt, et al. JASPAR 2016: a major expansion and update of the open-access database of transcription factor binding profiles. *Nucleic acids research*, 44(D1):D110–D115, 2016. (Cited on page 45.)

[128] Charles E Grant, Timothy L Bailey, and William Stafford Noble. FIMO: scanning for occurrences of a given motif. *Bioinformatics*, 27(7):1017–1018, 2011. (Cited on page 45.)

[129] Yong Zhang, Tao Liu, Clifford A Meyer, Jérôme Eeckhoute, David S Johnson, Bradley E Bernstein, Chad Nusbaum, Richard M Myers, Myles Brown, Wei Li, et al. Model-based analysis of ChIP-Seq (MACS). *Genome biology*, 9(9):1–9, 2008. (Cited on page 45.)

[130] Sven Heinz, Christopher Benner, Nathanael Spann, Eric Bertolino, Yin C Lin, Peter Laslo, Jason X Cheng, Cornelis Murre, Harinder Singh, and Christopher K Glass. Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. *Molecular cell*, 38(4):576–589, 2010. (Cited on page 45.)

[131] Changgeng Peng, Na Li, Yen-Kar Ng, Jingzhong Zhang, Florian Meier, Fabian J Theis, Matthias Merkenschlager, Wei Chen, Wolfgang Wurst, and Nilima Prakash. A unilateral negative feedback loop between miR-200 microRNAs and Sox2/E2F3 controls neural progenitor cell-cycle exit and differentiation. *Journal of Neuroscience*, 32(38):13292–13308, 2012. (Cited on page 45.)

[132] Ankita Pandey, Parul Singh, Abhishek Jauhari, Tanisha Singh, Farah Khan, Aditya B Pant, Devendra Parmar, and Sanjay Yadav. Critical role of the miR-200 family in regulating differentiation and proliferation of neurons. *Journal of neurochemistry*, 133(5):640–652, 2015. (Cited on page 45.)

[133] Vikram Agarwal, George W Bell, Jin-Wu Nam, and David P Bartel. Predicting effective microRNA target sites in mammalian mRNAs. *elife*, 4:e05005, 2015. (Cited on page 47.)

[134] Anton J Enright, Bino John, Ulrike Gaul, Thomas Tuschl, Chris Sander, and Debora S Marks. MicroRNA targets in Drosophila. *Genome biology*, 5(1):R1, 2003. (Cited on page 47.)

[135] Junjiro Horiuchi, Daisuke Yamazaki, Shintaro Naganos, Toshiro Aigaki, and Minoru Saitoe. Protein kinase A inhibits a consolidated form of memory in Drosophila. *Proceedings of the National Academy of Sciences*, 105(52):20976–20981, 2008. (Cited on page 47.)

[136] Eric R Kandel. The molecular biology of memory: cAMP, PKA, CRE, CREB-1, CREB-2, and CPEB. *Molecular brain*, 5(1):1–12, 2012. (Cited on page 47.)

[137] Xueyi Teng, Xiaomin Chen, Hua Xue, Yiheng Tang, Peng Zhang, Quan Kang, Yajing Hao, Runsheng Chen, Yi Zhao, and Shunmin He. NPInter v4. 0: an integrated database of ncRNA interactions. *Nucleic acids research*, 48(D1):D160–D165, 2020. (Cited on page 47.)

[138] Jing Gong, Di Shao, Kui Xu, Zhipeng Lu, Zhi John Lu, Yucheng T Yang, and Qiangfeng Cliff Zhang. RISE: a database of RNA interactome from sequencing experiments. *Nucleic acids research*, 46(D1):D194–D201, 2018. (Cited on page 47.)

[139] Alexander Junge, Jan C Refsgaard, Christian Garde, Xiaoyong Pan, Alberto Santos, Ferhat Alkan, Christian Anthon, Christian von Mering, Christopher T Workman, Lars Juhl Jensen, et al. RAIN: RNA–protein association and interaction networks. *Database*, 2017, 2017. (Cited on page 47.)

[140] Yuhao Chen and Xiaowei Wang. miRDB: an online database for prediction of functional microRNA targets. *Nucleic acids research*, 48(D1):D127–D131, 2020. (Cited on page 47.)

[141] Christelle Robert and Mick Watson. Errors in RNA-Seq quantification affect genes of relevance to human disease. *Genome biology*, 16(1):177, 2015. (Cited on page 49.)

[142] Rhoda J Kinsella, Andreas Kähäri, Syed Haider, Jorge Zamora, Glenn Proctor, Giulietta Spudich, Jeff Almeida-King, Daniel Staines, Paul Derwent, Arnaud Kerhornou, et al. Ensembl BioMarts: a hub for data retrieval across taxonomic space. *Database*, 2011, 2011. (Cited on page 53.)

[143] Anthony J Travis, Jonathan Moody, Aleksandra Helwak, David Tollervey, and Grzegorz Kudla. Hyb: a bioinformatics pipeline for the analysis of CLASH (crosslinking, ligation and sequencing of hybrids) data. *Methods*, 65(3):263–273, 2014. (Cited on page 53.)

[144] Jörg Fallmann, Pavankumar Videm, Andrea Bagnacani, Bérénice Batut, Maria A Doyle, Tomas Klingstrom, Florian Eggenhofer, Peter F Stadler, Rolf Backofen, and Björn Grüning. The RNA workbench 2.0: next generation RNA data analysis. *Nucleic acids research*, 47(W1):W511–W515, 2019. (Cited on pages 55 and 58.)

[145] Vincent J Henry, Anita E Bandrowski, Anne-Sophie Pepin, Bruno J Gonzalez, and Arnaud Desfeux. OMICtools: an informative directory for multi-omic data analysis. *Database*, 2014, 2014. (Cited on page 55.)

[146] Levin Clément, Dynomant Emeric, Mouchard Laurent, Landsman David, Hovig Eivind, Vlahovicek Kristian, et al. A data-supported history of bioinformatics tools. *arXiv preprint arXiv:1807.06808*, 2018. (Cited on page 55.)

[147] Sebastian J Schultheiss, Marc-Christian Münch, Gergana D Andreeva, and Gunnar Rätsch. Persistence and availability of web services in computational biology. *PLoS One*, 6(9):e24914, 2011. (Cited on page 55.)

[148] Fabian Kern, Tobias Fehlmann, and Andreas Keller. On the lifetime of bioinformatics web services. *Nucleic acids research*, 48(22):12523–12533, 2020. (Cited on page 56.)

[149] Jonathan D Wren, Constantin Georgescu, Cory B Giles, and Jason Hennessey. Use it or lose it: citations predict the continued online availability of published bioinformatics resources. *Nucleic acids research*, 45(7):3627–3633, 2017. (Cited on page 56.)

[150] Martin Raden, Syed M Ali, Omer S Alkhnbashi, Anke Busch, Fabrizio Costa, Jason A Davis, Florian Eggenhofer, Rick Gelhausen, Jens Georg, Steffen Heyne, et al. Freiburg RNA tools: a central online resource for RNA-focused research and teaching. *Nucleic acids research*, 46(W1):W25–W29, 2018. (Cited on page 56.)

[151] Andreas R Gruber, Ronny Lorenz, Stephan H Bernhart, Richard Neuböck, and Ivo L Hofacker. The vienna RNA websuite. *Nucleic acids research*, 36(suppl_2):W70–W74, 2008. (Cited on page 56.)

[152] Weizhong Li, Andrew Cowley, Mahmut Uludag, Tamer Gur, Hamish McWilliam, Silvano Squizzato, Young Mi Park, Nicola Buso, and Rodrigo Lopez. The EMBL-EBI bioinformatics web and programmatic tools framework. *Nucleic acids research*, 43(W1):W580–W584, 2015. (Cited on page 56.)

[153] Jeremy Leipzig. A review of bioinformatic pipeline frameworks. *Briefings in bioinformatics*, 18(3):530–536, 2017. (Cited on page 56.)

[154] Matthew C Chambers, Pratik D Jagtap, James E Johnson, Thomas McGowan, Praveen Kumar, Getiria Onsongo, Candace R Guerrero, Harald Barsnes, Marc Vaudel, Lennart Martens, et al. An accessible proteogenomics informatics resource for cancer researchers. *Cancer research*, 77(21):e43–e46, 2017. (Cited on page 57.)

[155] Björn A Grüning, Jörg Fallmann, Dilmurat Yusuf, Sebastian Will, Anika Erxleben, Florian Eggenhofer, Torsten Houwaart, Bérénice Batut, Pavankumar Videm, Andrea Bagnacani, et al. The RNA workbench: best practices for RNA and high-throughput sequencing bioinformatics in Galaxy. *Nucleic acids research*, 45(W1):W560–W566, 2017. (Cited on pages 57 and 58.)

[156] Simon A Bray, Xavier Lucas, Anup Kumar, and Björn A Grüning. The ChemicalToolbox: reproducible, user-friendly cheminformatics analysis on the Galaxy platform. *Journal of Cheminformatics*, 12(1):1–7, 2020. (Cited on page 57.)

[157] Sanchita Bhattacharya, Patrick Dunn, Cristel G Thomas, Barry Smith, Henry Schaefer, Jieming Chen, Zicheng Hu, Kelly A Zalocusky, Ravi D Shankar, Shai S Shen-Orr, et al. ImmPort, toward repurposing of open access immunological assay data for translational and clinical research. *Scientific data*, 5:180015, 2018. (Cited on page 57.)

[158] Jagat Singh Chauhan, Sandeep Kumar Dhanda, Deepak Singla, Subhash M Agarwal, Gajendra PS Raghava, Open Source Drug Discovery Consortium, et al. QSAR-based models for designing quinazoline/imidazothiazoles/pyrazolopyrimidines based inhibitors against wild and mutant EGFR. *PloS one*, 9(7):e101079, 2014. (Cited on page 57.)

[159] Tomasz Bednarz, Dadong Wang, Yulia Arzhaeva, Ryan Lagerstrom, Pascal Vallotton, Neil Burdett, Alex Khassapov, Piotr Szul, Shiping Chen, Changming Sun, et al. Cloud based toolbox for image analysis, processing and reconstruction tasks. In *Signal and image analysis for biomedical and life sciences*, pages 191–205. Springer, 2015. (Cited on page 57.)

[160] Anne Fouilloux. Getting your hands-on climate data (Galaxy Training Materials), 07 2020. [Online; accessed Mon Oct 05 2020]. (Cited on page 57.)

[161] Bérénice Batut, Saskia Hiltemann, Andrea Bagnacani, Dannon Baker, Vivek Bhardwaj, Clemens Blank, Anthony Bretaudeau, Loraine Brillet-Guéguen, Martin Čech, John Chilton, et al. Community-driven data analysis training for biology. *Cell systems*, 6(6):752–758, 2018. (Cited on pages 57 and 58.)

[162] Frédéric Rimet, Philippe Chaumeil, François Keck, Lenaïg Kermarrec, Valentin Vasselon, Maria Kahlert, Alain Franc, and Agnès Bouchez. R-Syst:: diatom: an open-access and curated barcode database for diatoms and freshwater monitoring. *Database*, 2016, 2016. (Cited on page 57.)

[163] Nancy Ide, James Pustejovsky, Christopher Cieri, Eric Nyberg, Denise DiPersio, Chunqi Shi, Keith Suderman, Marc Verhagen, Di Wang, and Jonathan Wright. The language application grid. In *International Workshop on Worldwide Language Service Infrastructure*, pages 51–70. Springer, 2015. (Cited on page 57.)

[164] Qiang Gu, Anup Kumar, Simon Bray, Allison Creason, Alireza Khanteymoori, Vahid Jalili, Björn Grüning, and Jeremy Goecks. Accessible, Reproducible, and Scalable Machine Learning for Biomedicine. *bioRxiv*, 2020. (Cited on page 57.)

[165] Monya Baker. 1,500 scientists lift the lid on reproducibility. 2016. (Cited on page 57.)

[166] Yang-Min Kim, Jean-Baptiste Poline, and Guillaume Dumas. Experimenting with reproducibility: a case study of robustness in bioinformatics. *GigaScience*, 7(7):giy077, 2018. (Cited on page 57.)

[167] Björn Grüning, Ryan Dale, Andreas Sjödin, Brad A Chapman, Jillian Rowe, Christopher H Tomkins-Tinch, Renan Valieris, and Johannes Köster. Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nature methods*, 15(7):475–476, 2018. (Cited on page 58.)

[168] Felipe da Veiga Leprevost, Björn A Grüning, Saulo Alves Aflitos, Hannes L Röst, Julian Uszkoreit, Harald Barsnes, Marc Vaudel, Pablo Moreno, Laurent Gatto, Jonas Weber, et al. BioContainers: an open-source and community-driven framework for software standardization. *Bioinformatics*, 33(16):2580–2582, 2017. (Cited on page 58.)

[169] Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239):2, 2014. (Cited on page 58.)

[170] Derek De Rie, Imad Abugessaisa, Tanvir Alam, Erik Arner, Peter Arner, Haitham Ashoor, Gaby Åström, Magda Babina, Nicolas Bertin, A Maxwell Burroughs, et al. An integrated expression atlas of miRNAs and their promoters in human and mouse. *Nature biotechnology*, 35(9):872–878, 2017. (Cited on page 63.)

[171] Kimberly A Dittmar, Jeffrey M Goodenbour, and Tao Pan. Tissue-specific differences in human transfer RNA expression. *PLoS Genet*, 2(12):e221, 2006. (Cited on page 63.)

[172] Hadi Jorjani, Stephanie Kehr, Dominik J Jedlinski, Rafal Gumienny, Jana Hertel, Peter F Stadler, Mihaela Zavolan, and Andreas R Gruber. An updated human snoRNAome. *Nucleic acids research*, 44(11):5068–5082, 2016. (Cited on page 63.)

[173] Sachin Pundhir and Jan Gorodkin. Differential and coherent processing patterns from small RNAs. *Scientific reports*, 5(1):1–16, 2015. (Cited on page 64.)

[174] Amadís Pagès, Ivan Dotu, Joan Pallarès-Albanell, Eulàlia Martí, Roderic Guigó, and Eduardo Eyras. The discovery potential of RNA processing profiles. *Nucleic acids research*, 46(3):e15–e15, 2018. (Cited on page 64.)

[175] S Steven Potter. Single-cell RNA sequencing for the study of development, physiology and disease. *Nature Reviews Nephrology*, 14(8):479–492, 2018. (Cited on page 64.)

[176] Michael Hagemann-Jensen, Ilgar Abdullayev, Rickard Sandberg, and Omid R Faridani. Small-seq for single-cell small-RNA sequencing. *Nature protocols*, 13(10):2407–2424, 2018. (Cited on page 64.)

[177] Zhengyun Xiao, Guo Cheng, Yang Jiao, Chen Pan, Ran Li, Danmei Jia, Jing Zhu, Chao Wu, Min Zheng, and Junling Jia. Holo-Seq: single-cell sequencing of holo-transcriptome. *Genome biology*, 19(1):1–22, 2018. (Cited on page 64.)

[178] Omid R Faridani, Ilgar Abdullayev, Michael Hagemann-Jensen, John P Schell, Fredrik Lanner, and Rickard Sandberg. Single-cell sequencing of the small-RNA transcriptome. *Nature biotechnology*, 34(12):1264–1266, 2016. (Cited on page 64.)

[179] Tomasz J Nowakowski, Neha Rani, Mahdi Golkaram, Hongjun R Zhou, Beatriz Alvarado, Kylie Huch, Jay A West, Anne Leyrat, Alex A Pollen, Arnold R Kriegstein, et al. Regulation of cell-type-specific transcriptomes by microRNA networks during human brain development. *Nature neuroscience*, 21(12):1784–1792, 2018. (Cited on page 64.)

[180] Nayi Wang, Ji Zheng, Zhuo Chen, Yang Liu, Burak Dura, Minsuk Kwak, Juliana Xavier-Ferrucio, Yi-Chien Lu, Miaomiao Zhang, Christine Roden, et al. Single-cell microRNA-mRNA co-sequencing reveals non-genetic heterogeneity and mechanisms of microRNA regulation. *Nature communications*, 10(1):1–12, 2019. (Cited on page 64.)