Approaches to analysis of chromosome conformation capture data

Joachim Wolff

Albert-Ludwigs-Universität
Freiburg im Breisgau

Technische Fakultät
Institut für Informatik
Lehrstuhl für Bioinformatik

Dissertation
zur Erlangung des akademischen Grades
Doctor rerum naturalium (Dr. rer. nat.) der
Technischen Fakultät der
Albert-Ludwigs-Universität Freiburg im Breisgau

Approaches to analysis of chromosome conformation capture data

Joachim Wolff

1. Reviewer Prof. Dr. Rolf Backofen
Lehrstuhl für Bioinformatik
Albert-Ludwigs-Universität Freiburg im Breisgau

2. Reviewer Prof. Dr. Ralf Gilsbach
Institute of Cardiovascular Physiology
Goethe-Universität Frankfurt am Main

Supervisor Prof. Dr. Rolf Backofen

Joachim Wolff
Approaches to analysis of chromosome conformation capture data
Dean: Prof. Dr. Roland Zengerle
Reviewers: Prof. Dr. Rolf Backofen and Prof. Dr. Ralf Gilsbach
Defense date: February 4, 2022
Supervisor: Prof. Dr. Rolf Backofen

Albert-Ludwigs-Universität
Freiburg im Breisgau
Lehrstuhl für Bioinformatik
Institut für Informatik
Technische Fakultät
Georges-Köhler-Allee 106
79110 Freiburg im Breisgau

Publication list

Joachim Wolff, Vivek Bhardwaj, Stephan Nothjunge, Gautier Richard, Gina Renschler, Ralf Gilsbach, Thomas Manke, Rolf Backofen, Fidel Ramírez, Björn Grüning. Galaxy HiCExplorer: a web server for reproducible Hi-C data analysis, quality control and visualization. *Nucleic Acids Research*, Volume 46, Issue W1, 2 July 2018, Pages W11-W16

Joachim Wolff, Leily Rabbani, Ralf Gilsbach, Gautier Richard, Thomas Manke, Rolf Backofen, Björn Grüning. Galaxy HiCExplorer 3: a web server for reproducible Hi-C, capture Hi-C and single-cell Hi-C data analysis, quality control and visualization. *Nucleic Acids Research*, Volume 48, Issue W1, 02 July 2020, Pages W177-W184

Joachim Wolff, Rolf Backofen, Björn Grüning. Robust and efficient single-cell Hi-C clustering with approximate k-nearest neighbor graphs. *Bioinformatics*. Accepted on 19 May 2021, published online on 22 May 2021. DOI: 10.1093/bioinformatics/btab394

Joachim Wolff, Nezar Abdennur, Rolf Backofen, Björn Grüning. scool: A new data storage format for single-cell Hi-C data. *Bioinformatics*, Volume 37, Issue 14, 15 July 2021, Pages 2053–2054

Lucille Lopez-Delisle, Leily Rabbani, **Joachim Wolff**, Vivek Bhardwaj, Rolf Backofen, Björn Grüning, Fidel Ramírez, Thomas Manke. pyGenomeTracks: reproducible plots for multivariate genomic data sets. *Bioinformatics*, Volume 37, Issue 3, 1 February 2021, Pages 422–423

Joachim Wolff, Rolf Backofen, Björn Grüning. Loop detection using Hi-C data with HiCExplorer (submitted)

Bérénice Batut, Saskia Hiltemann, Andrea Bagnacani, Dannon Baker, [...], **Joachim Wolff**, [...], Rolf Backofen, Anton Nekrutenko, Björn Grüning. Community-driven data analysis training for biology. Cell systems, 27 June 2018, 6(6):752-8

Martin Raden, Syed M Ali, Omer S Alkhnbashi, Anke Busch, Fabrizio Costa, Jason A Davis, Florian Eggenhofer, Rick Gelhausen, Jens Georg, Steffen Heyne, Michael Hiller, Kousik Kundu, Robert Kleinkauf, Steffen C Lott, Mostafa M Mohamed, Alexander Mattheis, Milad Miladi, Andreas S Richter, Sebastian Will, **Joachim Wolff**, Patrick R Wright, Rolf Backofen. Freiburg RNA tools: a central online resource for RNA-focused research and teaching. *Nucleic Acids Research*, Volume 46, Issue W1, 2 July 2018, Pages W25-W29

Björn Grüning, Ryan Dale, Andreas Sjödin, Brad A. Chapman, Jillian Rowe, Christopher H. Tomkins-Tinch, Renan Valieris, Johannes Köster & **The Bioconda Team**. Bioconda: sustainable and comprehensive software distribution for the life sciences. Nature Methods volume 15, 2 July 2018, pages 475-476

Oral presentations

Joachim Wolff: Developments of HiCExplorer. 5th early-stage researchers Next-Generation Sequencing Symposium. Heidelberg, Germany. 2017

Joachim Wolff: Galaxy HiCExplorer. Galaxy community kickoff meeting and Galaxy User Conference. Freiburg, Germany. 2018

Joachim Wolff: Galaxy HiCExplorer: HiCExplorer, deepTools3 and pyGenomeTracks. Galaxy community conference (GCC) / Bioinformatics Open Source Conference (BOSC). Portland, OR, USA. 2018

Joachim Wolff: HiCExplorer 3: A toolbox for Hi-C data analysis. Galaxy community conference (GCC). Freiburg, Germany. 2019

Presented posters

Fidel Ramírez, **Joachim Wolff**, Vivek Bhardwaj, Stephan Nothjunge, Gautier Richard, Gina Renschler, Ralf Gilsbach, Thomas Manke, Rolf Backofen, Björn Grüning: HiCExplorer. 2nd Cardiovascular Epigenetics Conference. Freiburg, Germany. 2018

Joachim Wolff, Leily Rabbani, Gautier Richard, Thomas Manke, Asifa Akhtar, Rolf Backofen, Fidel Ramírez, Björn A. Grüning. HiCExplorer 3: A Toolbox for Hi-C data analysis. François Jacob Conference: Evolution, Structure and Function of Chromosomes High Order Structure. Paris, France. 2019

Joachim Wolff, Leily Rabbani, Gautier Richard, Thomas Manke, Asifa Akhtar, Rolf Backofen, Fidel Ramírez, Björn A. Grüning. HiCExplorer 3: A Toolbox for Hi-C data analysis. Intelligent Systems for Molecular Biology (ISMB) / European Conference on Computational Biology (ECCB). Basel, Switzerland. 2019

Joachim Wolff, Leily Rabbani, Gautier Richard, Thomas Manke, Asifa Akhtar, Rolf Backofen, Fidel Ramírez, Björn A. Grüning. HiCExplorer 3: A Toolbox for Hi-C data analysis. Galaxy Community Conference (GCC). Freiburg, Germany. 2019

Joachim Wolff, Leily Rabbani, Gautier Richard, Thomas Manke, Asifa Akhtar, Rolf Backofen, Fidel Ramírez, Björn A. Grüning. HiCExplorer 3: A Toolbox for Hi-C data analysis. German Conference on Bioinformatics (GCB). Heidelberg, Germany. 2019

Lucille Lopez-Delisle, Leily Rabbani, **Joachim Wolff**, Vivek Bhardwaj, Rolf Backofen, Björn A. Grüning, Fidel Ramírez, Thomas Manke. pyGenomeTracks: Reproducible plots for multivariate genomic data sets. Bioinformatics Community Conference (BCC). Online conference. 2020

Workshops

Joachim Wolff, Leily Rabbani, Devon Ryan, Ralf Gilsbach. Hi-C data analysis. Galaxy Workshop Freiburg. Freiburg, Germany. September 2017, February 2018, September 2018, February 2019, September 2019, February 2020.

Devon Ryan, Ralf Gilsbach, **Joachim Wolff**. Methyl-C data analysis. Galaxy Workshop Freiburg. Freiburg, Germany. September 2017, February 2018, September 2018, February 2019, September 2019, February 2020.

Anika Erxleben, **Joachim Wolff**. ChIP-Seq data analysis. Galaxy Workshop Freiburg. February 2017.

Bérénice Batut, **Joachim Wolff**. Galaxy 101. Galaxy Community Conference (GCC) / Bioinformatics Open Source Conference (BOSC). Portland, OR, USA. 2018

Joachim Wolff, Leily Rabbani. Hi-C data analysis. Galaxy Community Conference (GCC). Freiburg, Germany. 2019

Zusammenfassung

Die dreidimensionale Struktur des Genoms findet zunehmend Beachtung in der Erforschung von regulatorischen Mechanismen in eukaryotischen Zellen. Methoden zur Messung der Expression wie RNA-Seq können die Genaktivität anzeigen, aber sind nicht dazu geeignet zu erklären, welche Faktoren die Genregulierung beeinflussen. Die Epigenetik, und im speziellen die Chromatinstruktur, bietet einen Erklärungsansatz für die biomedizinische Forschung an, welche Faktoren die Regulation beeinflussen. Hierbei sind die Interaktionen von Enhancer-Promotern ein Hauptkonzept zum Verständnis der Regulation von Genen. Es bedarf einer Bestätigung mittels biomedizinischen Laborverfahren, ob zwei DNA-Regionen wirklich miteinander interagieren. Ohne diese Bestätigung ist die Interaktion von Enhancer-Promotorn nur eine Interpretation der vorliegenden Daten. Die Feststellung der Chromosomen-Konformation (chromosome conformation capture (3C)) ist ein Verfahren, mit welcher räumlich nahe DNA-Regionen gemessen werden können. 3C-basierte Verfahren benötigen aufgrund der Zweidimensionalität gerade im Vergleich zu eindimensionalen Verfahren wie RNA-Seq oder ChIP-Seq einen quadratischen Faktor der zu erzeugenden DNA-Sequenzen, um eine gleiche Datenabdeckung zu erreichen. Hi-C ist hierbei das Verfahren zur Analyse des gesamten Genoms; es ist aber aufgrund der hohen Kosten nicht geeignet, die nötige Read-Coverage für spezifische Regionen im Regelfall bereitzustellen. Von Hi-C abgeleitete Verfahren wie capture Hi-C oder HiChIP sind aufgrund der Fokussierung auf vordefinierte Regionen wesentlich günstiger und stellen diese Spezifität bereit, sie benötigen aber wiederum eigene Analysemethoden. Hi-C erzeugt immer nur eine über mehrer Millionen Zellen kumulierte Datenmenge, zur Untersuchung von individuelle Zellen wurde die Erweiterung single-cell Hi-C entwickelt. Es erweitert die Analysemöglichkeiten hin zur Untersuchung von Unterschieden in der Chromatinstruktur von verschieden Zelltypen bzw. Stadien des Zellzykluses.

Die Analyse von Hochdurchsatz-Sequenzierungsdaten erfordert spezialisierte Methoden. Im Rahmen dieser Dissertation wurden für 3C sowie daraus abgeleitete Techniken verschiedene Analysemethoden und Vorgehen entwickelt. Ein Fokus lag hierbei auf Hi-C, capture Hi-C und single-cell Hi-C für welche Methoden zur besseren Analyse beigetragen wurden; unter anderem die Anpassung an molekularbiologische Neuerungen, dem Verbessern von Datenaustauschmöglichkeiten und der gezielten Komplexitätsreduktion in der Benutzung der Software. Die primäre Benutzergruppe der Analysesoftware - biomedizinische Forscher - haben oftmals keine grundlegenden Informatikkenntnisse. Die Bereitstellung der Software erfolgt über entwicklerseitige Verteilungskanäle wie den

Paketmanager ’Conda’; des Weiteren wird die Analysesoftware ’HiCExplorer’, ’scHiC-Explorer’ und ’pyGenomeTracks’ über einen Webserver im Rahmen von Software-as-a-Service (SaaS) angeboten. Darüber hinausgehend wird die Software auch per Docker-Container bereitgestellt. Dies löst das Problem der exakten Reproduzierbarkeit von Analysen und der Softwarearchivierung. Auch ermöglichen Container einen schnellen Einsatz von Software in Cloud-Umgebungen, wie es im Hintergrund für SaaS oftmals notwendig ist.

In dieser Dissertation wurde ein Algorithmus zur DNA-Schleifenerkennung in Hi-C Daten entwickelt, der auf kontinuierlichen negativen Binomialverteilungen basiert. Des Weiteren wurde eine Methode zur Detektion von differenziellen topologischen assoziierten Domänen (TADs), oder auch globale Vergleichsmethodiken wie der Vergleich des Verhältnisses von Kontakten kurzer und weiter genomischer Distanz, erstellt. Außerdem wurde eine Software zur Visualisierung von Hi-C, aber auch anderer genomischer Daten programmiert. Tools zur Analyse der Qualität der vorliegenden Hi-C, capture Hi-C und single-cell Hi-C Daten wurden erweitert, in bestehende Software integriert oder neu entwickelt.

Eine Erkennung von signifikanten respektive differenziellen DNA-Interaktionen, wie sie beispielsweise für Enhancer-Promoter DNA Interaktionen vorkommt, wurde ausgearbeitet. Der wesentliche Beitrag im Feld der single-cell Hi-C-Datenanalyse ist die Entwicklung eines Dateiformates zur effizienten Speicherung von single-cell Hi-C Daten. Es wurde eine Methode basierend auf approximativen k-nächste Nachbargraphen zur Dimensionsreduktion und Clustering von hochdimensionalen single-cell Hi-C Daten entworfen.

Abstract

The three-dimensional structure of the genome has a rising impact in the research to understand the regulatory mechanisms in eukaryotic cells. Expression-based methods like RNA-Seq can show if a gene is active or inactive; however, they cannot explain why the gene is regulated in this specific way. The chromatin structure, an epigenetic property, is the focus of biomedical researchers to explain the factors involved in the regulation. Enhancer and promoter interactions are one key concept to understand the regulation of genes. However, without wet-lab techniques providing evidence of the interaction of two specific DNA regions containing the enhancer and promoter regions, it is only an interpretation of the data. Chromosome conformation capture (3C) is a technique that can capture the spatial closeness of DNA regions; it is essential to mention that the interaction of these regions is only an interpretation of the spatial closeness. 3C and its derivatives like Hi-C are based on a two dimensional data structure and require, compared to one-dimensional techniques like RNA-Seq or ChIP-Seq, a squared factor of reads for a similar coverage. Hi-C is a genome-wide approach and is the method of choice for coarser analysis; however, it lacks a high read coverage due to the protocol's economic costs. Specialized but cheaper approaches like capture Hi-C or HiChIP fill this gap but require different analysis methods. Furthermore, Hi-C uses up to a million cells for one sample generation, resulting in an accumulated data profile. To overcome this, single-cell Hi-C exists and provides the foundation to analyze the differing chromatin structure of cell types respectively cell cycles.

The analysis of high-throughput sequencing data requires specialized algorithms and methods. In this dissertation, different analysis approaches to analyze chromosome conformation data (3C) have been developed. A particular focus was the 3C derivatives Hi-C, capture Hi-C, and single-cell Hi-C, where improved analysis methods, the adaption of new developments in the wet-lab protocols, the improved data exchange options, and a complexity reduction of the analysis pipeline were contributed. The target users of a Hi-C data analysis software are biomedical researchers without knowledge in computer science. The software has been distributed via package managers like 'Conda', and a web server, the Galaxy HiCExplorer, was provided to make the software HiCExplorer, scHiCExplorer, and pyGenomeTracks accessible via the software-as-a-service approach. The developed software is also provided as a Docker container, solving software reproducibility and archiving with all its dependencies, and enables fast usage in a cloud environment.

In this thesis, I developed a chromatin loop detection algorithm based on continuous negative binomial distributions for Hi-C data. Furthermore, algorithms for a differential analysis of TADs or global comparisons like short-to-long range contact ratios have been created. Visualization options have been programmed for both the Hi-C data itself, as well as to integrate Hi-C with other genomic data. Contributions have been made to extend or integrate quality control tools; unique quality control methods for capture Hi-C and single-cell Hi-C have been added. A method to detect and analyze the large scale of point-to-point interactions, i.e., enhancer-promoter interactions, in the context of capture Hi-C and HiChIP was designed, including features for significance and differential detection. The major contribution to single-cell Hi-C data was by creating a specialized file format to improve the interoperability of single-cell Hi-C experiments. A method to cluster high-dimensional single-cell Hi-C data using approximate k-nearest neighbor graphs was implemented.

Contents

List of Figures

Glossary

3C Chromosome conformation capture. 2, 3, 7, 8, 27, 29, 77

4C Chromosome conformation capture-on-chip or Circular chromosome conformation capture. 8, 27–29, 55, 56

5C Chromosome conformation capture carbon copy. 8, 27, 29

API Application Programming Interface. 33

ChIP-Seq Chromatin immunoprecipitation sequencing. 22, 27, 57, 72

CLI Command-line interface. 72

DNA Deoxyribonucleic acid. xiii, 1, 2, 7, 8, 13, 15–21, 23–32, 35, 37, 39, 47, 60, 76–78

FISH Fluorescence in situ hybridization. 28, 30, 76, 77

Hi-C High-throughput variant of 3C. xiii, 3, 4, 8–12, 21, 22, 27, 28, 30, 31, 35–43, 45, 46, 48, 49, 51, 55–61, 63–78

HiChIP Combined Hi-C and Chromatin immunoprecipitation sequencing. 8, 27, 58, 59

ICE iterative correction and eigenvector decomposition. 9

KR Knight and Ruiz matrix balancing. 9

PCR polymerase chain reaction. 27–29, 39

RNA Ribonucleic acid. xiii, 1, 7, 13, 15–17, 35, 77

RNA-Seq RNA sequencing. High-throughput sequencing of RNA. 7, 57, 72

SaaS Software-as-a-service. 72, 75

SSE Streaming SIMD Extensions. 10

TAD Topological associated domains. 3, 8, 22, 31, 32, 36, 40, 43–48, 55, 57, 73, 77

TSS Transcription start site. 23

1 Introduction

Regulatory mechanisms of gene expression are essential in eukaryotic organisms and are responsible for cell differentiation, the accurate response to external stimuli, or cell processes like mitosis. Misregulation of genes is considered as a significant factor for diseases like cancer or diabetes, involving a change or mutation in transcription factors, non-coding RNA (ncRNA), or the chromatin regulation and therefore its accessibility [1]. The most common approach to determine gene expression is by measuring the gene expression's product - the occurrence of messenger RNA (mRNA) with modern high-throughput sequencing techniques, e.g., RNA-Seq. Based on this technique, differential expression for a gene can be determined by comparing the expression level with wild-type samples. RNA-Seq can be a method to investigate the impact of medication in up- or down-regulating gene expression of a particular gene or a group of genes. The expression of genes is regulated by the binding of transcription factors (TF) to specific DNA sequences in the local environment of the gene. In general, this local environment is defined by the single dimension of the DNA sequence. However, in reality DNA exists in a three-dimensional space, and therefore not only genomic distance but also spatial distance should be taken into account. To be more precise, the regulation of the transcription process of DNA to RNA by the RNA polymerase is influenced by the interaction of the enhancer and promoter region upstream of the transcription start site (TSS). For the interaction of the enhancer and promoter region, several transcription factors are involved, and, caused by the distance of the enhancer and promoter region, the DNA must be bent in the three-dimensional space. The proximity enables the RNA polymerase to bind to the DNA and to transcribe the genetic information. In this context, the role of the chromatin structure in the transcription process and thereby the three-dimensional structure of DNA in the cell nucleus attracted the attention of researchers [2, 3, 4]. For example, the cause for a disease like the Cooks syndrome [5, 6] can be explained by a higher expression profile of KCNJ2; however, the cause for the higher expression remains unclear. An investigation of the DNA sequence shows a duplication of the KCNJ2 region close to the following gene, SOX9, which leads to a different three-dimensional structure of the chromatin. The first, wild-type KCNJ2, is controlled by its regulatory elements; however, the second, new, KCNJ2 is controlled by the copied regulation sequences of the downstream SOX9 gene [7].

To measure the three-dimensional chromatin contacts, Dekker *et al.* introduced in 2002 the 'chromosome conformation capture' (3C) technique [8]. 3C allowed for the first time to crosslink two DNA fibers of distant loci if they are close in the spatial space;

the technique's limitation was a restriction for two preselected DNA sequences ('one vs. one'). Simonis *et al.* [9], and Zhao *et al.* [10] developed independently from each other the very similar techniques 'Chromosome conformation capture-on-chip' and 'Circular chromosome conformation capture', or 4C, allowing to measure the interactions of one preselected region with the whole genome ('one vs. all'). It was followed by 'Chromosome conformation capture carbon copy' (5C) in 2006 from Dostie *et al.* [11] enabling the capture of multiple regions with each other ('many vs. many') and finally, in 2009, Lieberman-Aiden *et al.* introduced the genome-wide high-throughput sequencing derivative of 3C, Hi-C [12] ('all vs. all'). The advent of the 3C based sequencing methods unveiled multiple organizational structures of the DNA in the three-dimensional space: The euchromatin and heterochromatin associated open and closed compartments (A/B compartments) [12], topological associated domains (TADs) [13] and DNA loops [14]. It confirmed the well-known separation of the genome in chromosomes [12]. Based on Hi-C, different and specialized high-throughput approaches have been developed: capture Hi-C [15] to detect enhancer-promoter interactions, single-cell Hi-C [16] to examine the chromatin conformation during mitosis and the differences of cell types or HiChIP [17] to detect chimeric protein binding sites. The Hi-C protocol has been improved in an ongoing process, based on the initial work of Lieberman-Aiden [12]; Rao *et al.* [14] introduced in situ Hi-C in 2014, and the Arima Hi-C kit[1], using two restriction enzymes, was available in 2018.

The data analysis of the Hi-C protocols introduced above requires multiple consecutive stages. First, the raw sequencing data must be quality controlled for read errors, be trimmed if applicable, or returned to the sequencing facility if the data is faulty. Next, the raw reads must be mapped to a reference genome in a chimeric way caused by the Hi-C protocol's specificity. Given the raw data, a two-dimensional interaction matrix with an additional quality control concerning the chimeric reads is created. Based on this interaction matrix, which is the primary Hi-C data structure, data correction, A/B compartments, TADs, DNA loops, or differential analyses of A/B compartments, TADs, loops, or predefined matrix regions can be computed. The two-dimensional representation as an interaction matrix increases the data volume by a quadratic value compared to one-dimensional genomic data. To obtain the same read coverage per position in a Hi-C data structure compared to one-dimensional data like RNA-Seq, a higher read number is required: for example RNA-seq requires 20 - 40 million reads, whereas for Hi-C 400 - 1600 million reads are usually recommended. The higher read number leads to significantly higher costs to run the Hi-C experiments and makes the computation with Hi-C data resource intensive.

The software for quality control, trimming, and mapping are generic for most high-throughput sequencing data and can be used for Hi-C data with minor adjustments. The specific software stack for Hi-C data analysis starts with the creation of the interaction

[1] https://arimagenomics.com/

matrix. Much software in the Hi-C data analysis field is specialized to solve precisely one computational problem, for example, Gothic [18] focuses on bias removal and the identification of accurate contacts, cLoops [19] on loop calling, CHiCAGO [20] on capture Hi-C analysis, or particular clustering of single-cell Hi-C data (scHiCluster) [21]. Unexpectedly, much software does not provide the first steps of the analysis pipelines: the creation of the interaction matrix from raw data, the quality control of the chimeric reads, the correction with methods like KR [14, 22] or ICE [23] and lack of essential matrix transformations. The data formats of the interaction matrix differ, are not following any standard, are sometimes for "human recognition"[2] implemented as a dense matrix stored in a text file, lack proper documentation (e.g., HiCorrector [24]), or is presented as a screenshot of an excel table[3]. Also, the visualization of Hi-C data and its integration with other genomic data is often not available, making it unnecessarily difficult to create plots to gain insights into the data.

Another aspect is the run time and memory performance, which affects whether the software can be executed on the currently available computers. Many of the above-listed programs are designed for low-resolution Hi-C data, e.g., a 1 megabase pair (Mb) resolution matrix. 1 Mb means that 1 million base pairs are considered as a single data point; for example, with the mouse genome and its 2.7 gigabase pairs, a 1 Mb matrix would have (2700×2700) dimensions. However, analysis software like Homer [25] or HiCorrector expects a text file with a dense matrix; for 1 Mb, this might work, but a 10 kilobase pair (kb) resolution matrix would have (270000×270000) dimensions and limits the usability of the software.

The sustainability of research software is an important aspect. In general, software sustainability, preservation, and archiving are key aspects for long-term software-driven research as we have it today. It is highly problematic if the software is not maintained anymore as soon as the related original paper is published, making the software in many cases quickly unusable because of dependency updates, unfixed bugs, or hard-coded paths. Approaches to solve this are complex, especially if the full stack of dependencies is considered. Not only the obvious dependencies like other APIs in the direct application environment but also dependencies like the operating system or the specific hardware platform need to be considered. Trivial changes in the software development process already make some progress to solving these problems, for example publishing the source code and using an open-source license, listing the direct dependencies, and writing software in a reusable way. More complex solutions are containers (e.g. Docker) or virtual machines [26].

The sustainability of research software also contributes to the reproducibility of data analysis. Modern data analysis requires continuous software development. Consequently,

[2] https://github.com/jasminezhoulab/Hi-Corrector/blob/master/Manual_HiCorrector_1.2.pdf page 4, section 1

[3] Homer software: http://homer.ucsd.edu/homer/interactions/HiCmatrices.html

researchers need to publish the used mathematical methods, algorithms and the particular implementation, its source, and the version of the software itself and its dependencies. Data provenance is a key concept to achieve reproducibility of data analysis; otherwise, a reproduction of a published data analysis might lead to different results. The version numbers and specific implementation is essential to trace back an error in the analysis which an implementation bug might cause. To publish the source code of the software is not sufficient because software nowadays has always dependencies, and these are available for different CPU platforms or might use in the backend different implementations depending on what hardware is available or which compile flags have been used. For example, TensorFlow's machine learning library comes with a regular, non-optimized version, a version supporting SSE instruction set, and a version supporting GPUs with Nvidia's CUDA. All three implementations might have a different behavior on specific details, for example when rounding floating numbers or have different bugs in their different code paths. Unfortunately, this can all lead to differing computation results and might influence downstream the decision if, e.g., an expression of a gene is considered as differential or not. If the method of how the result is computed with all mentioned configurations is unknown, this might lead to difficulties in the review process or, even worse, to publishing wrong results. The analysis workflow is complicated, many tools are involved in the process, and several parts of the analysis pipeline might need to be run several times by the researchers to find the correct parameter setting for the specific data. To keep track of all configurations and results is daunting and a potential source for human-introduced errors. A solution for this is a software-driven approach which can, first, protocol the used software, the dependencies, and their versions; second, store every intermediate step and results with their settings in a digital 'lab-notebook'; third, be able to publish the first two points in a clear organized and online accessible way; and fourth, provide the option to re-run every step in the analysis with the software versions available at the time of the analysis and compare it to newer versions.

1.1 Thesis outline

This dissertation focuses on the development of approaches to analyze high-throughput sequencing chromosome conformation capture data. The Hi-C data analysis software, HiCExplorer, was initially designed by Fidel Ramírez of the Max-Planck-Institute for Epigenetics and Immunobiology Freiburg. It was maintained and further developed during this thesis. Moreover, the infrastructure for a user-friendly and reproducible usage of HiCExplorer was implemented by providing Conda packages and a Galaxy-based web server, available under `https://hicexplorer.usegalaxy.eu`. Modules for capture Hi-C data analysis have been added, and the single-cell Hi-C data analysis software scHiCExplorer was written throughout this thesis. Besides providing the analysis software suite, a method to cluster single-cell Hi-C data on high-resolution interaction matrices and a new single-cell Hi-C data format have been designed, implemented, and published. Also, the software pyGenomeTracks to create software-driven, workflow-enabled visualization of genomic data was designed and implemented.

1.2 Note on publications

The results shown in this dissertation have been published in various journals:

- **Joachim Wolff**, Vivek Bhardwaj, Stephan Nothjunge, Gautier Richard, Gina Renschler, Ralf Gilsbach, Thomas Manke, Rolf Backofen, Fidel Ramírez, Björn Grüning
  Galaxy HiCExplorer: a web server for reproducible Hi-C data analysis, quality control and visualization
  Nucleic Acids Research, Volume 46, Issue W1, 2 July 2018, Pages W11-W16 [27]

- **Joachim Wolff**, Leily Rabbani, Ralf Gilsbach, Gautier Richard, Thomas Manke, Rolf Backofen, Björn Grüning
  Galaxy HiCExplorer 3: a web server for reproducible Hi-C, capture Hi-C and single-cell Hi-C data analysis, quality control and visualization
  Nucleic Acids Research, Volume 48, Issue W1, 02 July 2020, Pages W177-W184 [28]

- **Joachim Wolff**, Rolf Backofen, Björn Grüning
  Robust and efficient single-cell Hi-C clustering with approximate k-nearest neighbor graphs
  Bioinformatics. Accepted on 19 May 2021. DOI: 10.1093/bioinformatics/btab394 [29]

- **Joachim Wolff**, Nezar Abdennur, Rolf Backofen, Björn Grüning
  scool: A new data storage format for single-cell Hi-C data.
  Bioinformatics, Volume 37, Issue 14, 15 July 2021, Pages 2053–2054 [30]

- Lucille Lopez-Delisle, Leily Rabbani, **Joachim Wolff**, Vivek Bhardwaj, Rolf Backofen, Björn Grüning, Fidel Ramírez, Thomas Manke
  pyGenomeTracks: reproducible plots for multivariate genomic data sets
  Bioinformatics. Volume 37, Issue 3, 1 February 2021, Pages 422–423 [31]

1.3 Note on software

The software written during this thesis is available via multiple channels. First, the source code repositories are hosted on GitHub: HiCExplorer[4], HiCMatrix[5], pyGenomeTracks[6], scHiCExplorer[7] and sparse-neighbors-search[8]. Second, the software is present with multiple versions on the Bioconda channel of Conda[9]; and as a container via BioContainers[10]. Third, the webserver Galaxy HiCExplorer is available under `https://hicexplorer.usegalaxy.eu`, and the HiCExplorer, scHiCExplorer, and pyGenomeTracks Galaxy wrappers are available as source code[11] and on the Galaxy ToolShed[12]. All HiCExplorer and scHiCExplorer tools on `https://hicexplorer.usegalaxy.eu` have been executed 14,412 times, and 1199 times for pyGenomeTracks (status of mid-April 2021); the Galaxy HiCExplorer tool suite was downloaded 332 times from the Galaxy ToolShed, scHiCExplorer tool suit four times, and pyGenomeTracks 413 times (status of early May 2021). HiCExplorer has been downloaded 59,526 times from Conda, scHiCExplorer 490 times, pyGenomeTracks 24,157 times, and sparse-neighbors-search 8074 times (status of early May 2021).

1.4 Impact on publications

The Hi-C data analysis software HiCExplorer was used in multiple, high-ranked publications, sorted by the journals impact factor (IF) of 2018 according to *bioxbio.com*[13]. The presented publications represent the status of April 2021:

- Chen et. al. Key role for CTCF in establishing chromatin structure in human embryos. Nature. 2019. [32] (IF 43.070)

- Samata et. al. Intergenerationally Maintained Histone H4 Lysine 16 Acetylation Is Instructive for Future Gene Activation. Cell. 2020. [33] (IF 36.216)

- Xie et. al. Biased gene retention during diploidization in Brassica linked to three-dimensional genome organization. Nature plants. 2019. [34] (IF 12.109)

[4] https://github.com/deeptools/HiCExplorer
[5] https://github.com/deeptools/HiCMatrix
[6] https://github.com/deeptools/pyGenomeTracks
[7] https://github.com/joachimwolff/scHiCExplorer
[8] https://github.com/joachimwolff/sparse-neighbors-search
[9] https://anaconda.org/bioconda/hicexplorer
[10] https://quay.io/repository/biocontainers/hicexplorer?tab=tags
[11] https://github.com/galaxyproject/tools-iuc
[12] https://toolshed.g2.bx.psu.edu/
[13] https://www.bioxbio.com

- Alavattam et. al. Attenuated chromatin compartmentalization in meiosis and its maturation in sperm development. Nature structural & molecular biology. 2019. [35] (IF 11.980)
- Li et. al. YY1 interacts with guanine quadruplexes to regulate DNA looping and gene expression. Nature Chemical Biology (2021). [36] (IF 12.154)
- Sun et. al. Heat stress-induced transposon activation correlates with 3D chromatin organization rearrangement in Arabidopsis. Nature communications. 2020. [37] (IF 11.878)
- Qin et. al. Alterations in promoter interaction landscape and transcriptional network underlying metabolic adaptation to diet. Nature communications. [38] (IF 11.878)
- Muller et. al. The impact of centromeres on spatial genome architecture. Trends in genetics. 2019. [39] (IF 10.627)
- Han et. al. Diploid genome architecture revealed by multi-omic data of hybrid mice. Genome Research. 2020. [40] (IF 9.944)

pyGenomeTracks has already been used widely by many peer-reviewed publications such as

- Boulias et. al. Identification of the m6Am methyltransferase PCIF1 reveals the location and functions of m6Am in the transcriptome. Molecular cell. 2019.(IF 15.584)
- Drexler et. al. Splicing kinetics and coordination revealed by direct nascent RNA sequencing through nanopores. Molecular Cell. 2020. (IF 15.584)
- Alavattam et. al. Attenuated chromatin compartmentalization in meiosis and its maturation in sperm development. Nature structural & molecular biology. 2019. (IF 11.980)
- Bhardwaj et. al. MAPCap allows high-resolution detection and differential expression analysis of transcription start sites. Nature communications. 2019. (IF 11.878)
- Nothjunge et. al. DNA methylation signatures follow preformed chromatin compartments in cardiac myocytes. Nature communications. 2017. (IF 11.878)
- Qin et. al. Alterations in promoter interaction landscape and transcriptional network underlying metabolic adaptation to diet. Nature Communications. 2020. (IF 11.878)
- Van Tran et. al. The human 18S rRNA m6A methyltransferase METTL5 is stabilized by TRMT112. Nucleic acids research. 2019. (IF 11.501)

- Navarro-Mendoza et. al. Early diverging fungus Mucor circinelloides lacks centromeric histone CENP-A and displays a mosaic of point and regional centromeres. Current Biology. 2019. (IF 9.601)

- Shuaib et. al. Nuclear AGO1 Regulates Gene Expression by Affecting Chromatin Architecture in Human Cells. Cell Systems. 2019. (IF 8.673[14]) [41]

- Antonova et. al. Heat-Shock Protein 90 Controls the Expression of Cell-Cycle Genes by Stabilizing Metazoan-Specific Host-Cell Factor HCFC1. Cell reports. 2019. (IF 8.109)

- Vara et. al. Three-dimensional genomic structure and cohesin occupancy correlate with transcriptional activity during spermatogenesis. Cell reports. 2019. (IF 8.109)

- Bhardwaj et. al. snakePipes: facilitating flexible, scalable and integrative epigenomic analysis. Bioinformatics. 2019. (IF 5.610)

- Eres et. al. Reorganization of 3D genome structure may contribute to gene regulatory evolution in primates. PLoS genetics. 2019. (IF 5.174)

- Arrigoni et. al. RELACS nuclei barcoding enables high-throughput ChIP-seq. Communications biology. 2018. [42]

[14] https://www.sciencedirect.com/journal/cell-systems

Background

2

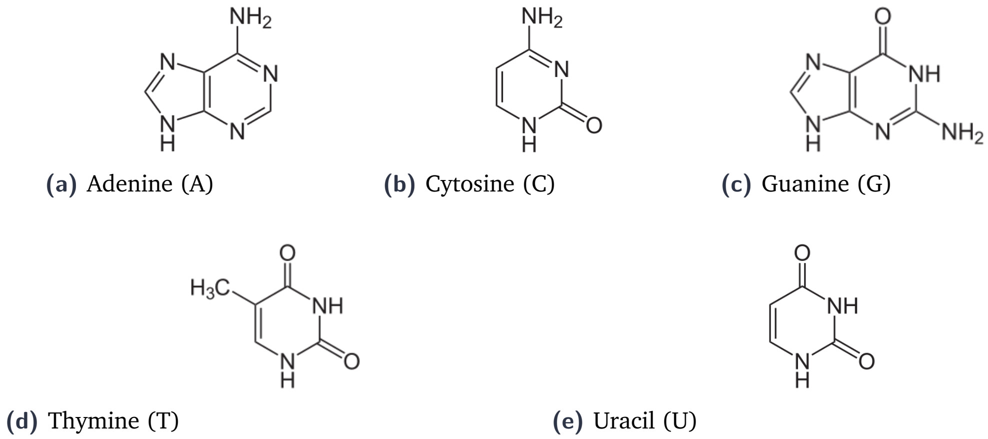
2.1 Biological background

2.1.1 DNA, RNA, and proteins

Deoxyribonucleic acid (DNA) was first discovered by Friedrich Miescher in 1871 [43], but the specific function of DNA was unknown. Nucleic acid was suspected to be the hereditary material for all living organisms since the early 20th century. It took Oswald Avery's experiments in 1944 to explicitly identify the deoxyribonucleic acid [44]. The DNA is located in the cells of eukaryotes, bacteria, and archaea; for eukaryotes, it is mainly located in the cell nucleus. The DNA is a polymer molecule that consists of four nucleotides, adenine (A), cytosine (C), guanine (G), and thymine (T) (Figure 2.1). The nucleotides have a pentose ring with an attached base (Figure 2.2 bottom, blue and green) and on 5' a phosphate group is attached (Figure 2.2 bottom, purple). The DNA is a polymer of the individual nucleotides, forming a nucleotide chain based on the sugar-phosphate bonds (Figure 2.2 top right). In 1953, James Watson and Francis Crick published, based on x-ray experiments of Maurice Wilkins and Rosalind Franklin [45, 46, 47, 48, 49], the structure of the DNA: a two-stranded double helix (Figure 2.2 left). The sugar-phosphate bonds are the backbone of the structure, and the nucleotides are oriented inwards, bonding to the other strand's nucleotides by hydrogen bonds (Figure 2.2 right). The hydrogen bonds form between adenine and thymine or cytosine and guanine. The order of A-T and C-G combinations are encoding for genes, and therefore DNA is also called the *blueprint of life*.

Ribonucleic acid (RNA) is, in contrast to DNA, single-stranded, has a ribose instead of a 2-desoxyribose as a pentose ring, and instead of thymine, it contains the nucleotide uracil (Figure 2.1e). In eukaryotes, the RNA is created inside the cell nucleus by transcribing the DNA with the RNA polymerase. It is an intermediate product of the gene expression process and, in its form as messenger RNA (mRNA), the base for protein synthesis. The role of RNA in the cell is more complex; non-coding RNAs (ncRNA) which do not encode for protein genes play essential roles. Examples are ribosomal RNA (rRNA), transfer RNA (tRNA), small nuclear RNA (snRNA), or long non-coding (lncRNA). The different subtypes have a role in gene regulation, polypeptide creation, or chromatin folding.

Proteins are the essential functional element of organisms and are made of amino acid chains based on the information given by a specific mRNA to the ribosome complex. The



(a) Adenine (A)

(b) Cytosine (C)

(c) Guanine (G)

(d) Thymine (T)

(e) Uracil (U)

Figure 2.1.: The nucleotides adenine (A), cytosine (C), guanine (G), thymine (T) and uracil (U) are the basis of DNA (A,C,G,T) and RNA (A,C,G,U).
For source and license information, please refer to the List of Figures.

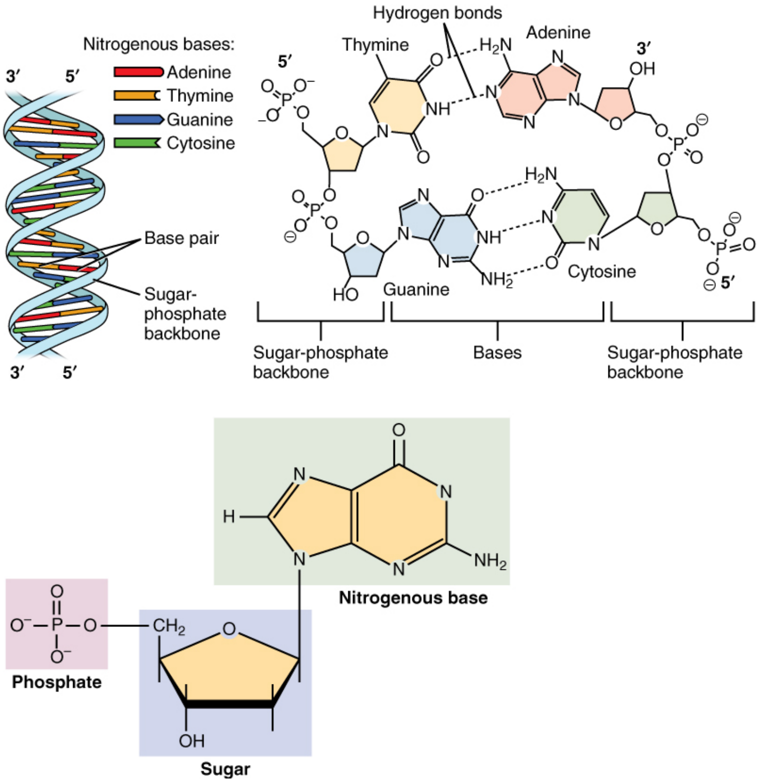


Figure 2.2.: The structure of DNA. Top left: the double helix structure forming the DNA molecule. Top right: A-T and G-C binding via hydrogen bonds, the sugar-phosphates of the nucleotides compose the backbone structure of the DNA. Bottom: A nucleotide has a central sugar group, with at 5' attached phosphate group. To the sugar on 2', a base binds, determining the individual nucleotide.
For source and license information, please refer to the List of Figures.

amino acid chain itself folds in the cellular solvent into a complex, three-dimensional structure and is in many cases combined with other proteins to protein complexes. The role of proteins in organisms is a vast collection of functions. Proteins bind to other molecules. For example, they can identify specific DNA sites and act as DNA or RNA polymerase, transcription factors, are antibodies of the immune system and can identify specific bacteria or viruses, form the structure of a cell, or are catalytic elements. Proteins have specific functional binding sites; this allows them to bind to their functional targets specifically. A good overview of the specifics and properties of proteins can be found in "Molecular biology of the cell" from Albert *et al.*, chapter 4 [50].

The central dogma of molecular biology describes the information flow in the cell. First, the DNA is accessed and replicated by the DNA polymerase (Figure 2.3 top). Replicating the DNA is the basis for cell replication, cell division, and differentiation. Second, the genetic information is first transcribed to RNA from DNA; the DNA is accessed by the RNA polymerase and transcribes the gene information to RNA (Figure 2.3 middle). The mRNA is used to create with the help of ribosomes the proteins by translation (Figure 2.3 bottom). Three consecutive nucleotides encode for one amino acid; a shift of just one position of the reading frame will create a different protein that is potentially non-functional with harmful consequences to the organism.

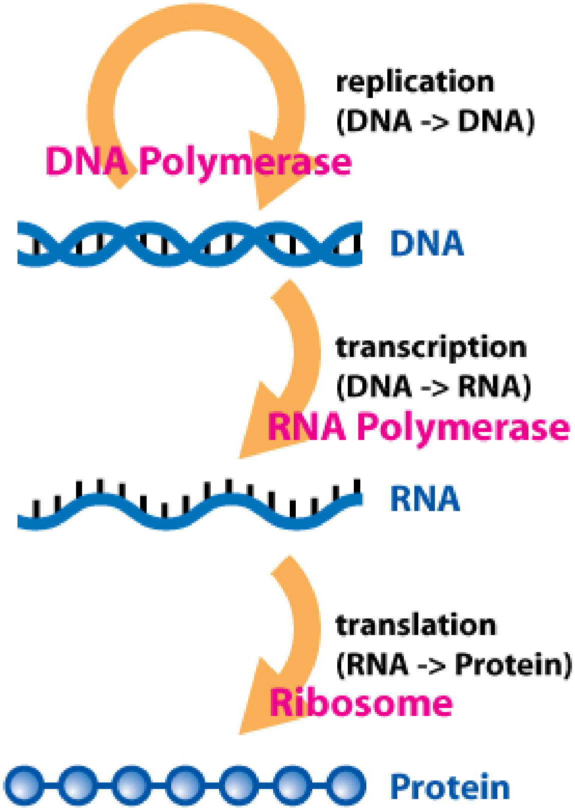


Figure 2.3.: The central dogma of molecular biology: DNA replicates itself with the DNA polymerase. The genetic information stored in the DNA is transcribed with the RNA polymerase to different types of RNA to express genes. The special type messenger RNA (mRNA) is afterward translated to proteins.
For source and license information, please refer to the List of Figures.

2.2 Organization of DNA within the cell nucleus

The DNA double helix is a fiber of, considering humans, approximately two meters of length [51] and it fits into each cell nucleus of a diameter of $\sim 10\ \mu m$ [52]. This is possible by a compact packaging of the DNA achieved by four packaging levels, called primary to the quaternary structure [53]. The raw DNA double helix has a diameter of 2 nm (Figure 2.5 1) and is wrapped around chromosomal proteins or more specific histones. The DNA is wrapped around eight histones, H2A, H2B, H3, H4 are present each two times. Figure 2.4b shows an electron microscope capture of this structure [54]. These histones are also termed core-histones and form together with the DNA a 11 nm fibre structure which is named *nucleosome* (Figure 2.5 2 and 3). Figure 2.4a shows a microscope capture of this structure [54]. The nucleosome and histone H1 form the chromatosome (Figure 2.5 4) and folds to a 30 nm fibre, the secondary structure (Figure 2.5 5). Figure 2.4c shows a microscope capture of this structure [54]. The 30 nm fibre is the base for chromatin loops, a 300 nm fibre (Figure 2.5 6) and each six loops form a rosetta, the tertiary structure. These are compressed and folded and form a 250 nm wide fibre named *chromatid*, the quaternary structure (Figure 2.5 7). Two connected so-called sister-chromatids form one *chromosome* (Figure 2.5 8).

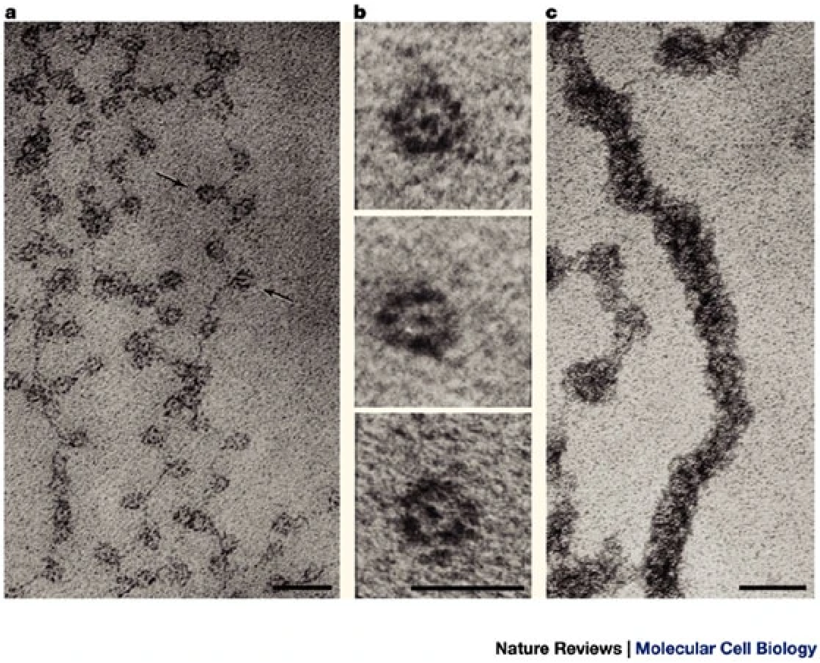


Figure 2.4.: Electron microscope capture of chromatin in the cell. **(a)** Chromatin strings 'beads on a string'. The beads are histone-DNA complexes (chromatosome); size marker of 30 nm. **(b)** The DNA wrapped around the histone complex; size marker of 10 nm. **(c)** The 30 nm compacted chromatin structure; size marker of 50 nm.
For source and license information, please refer to the List of Figures.

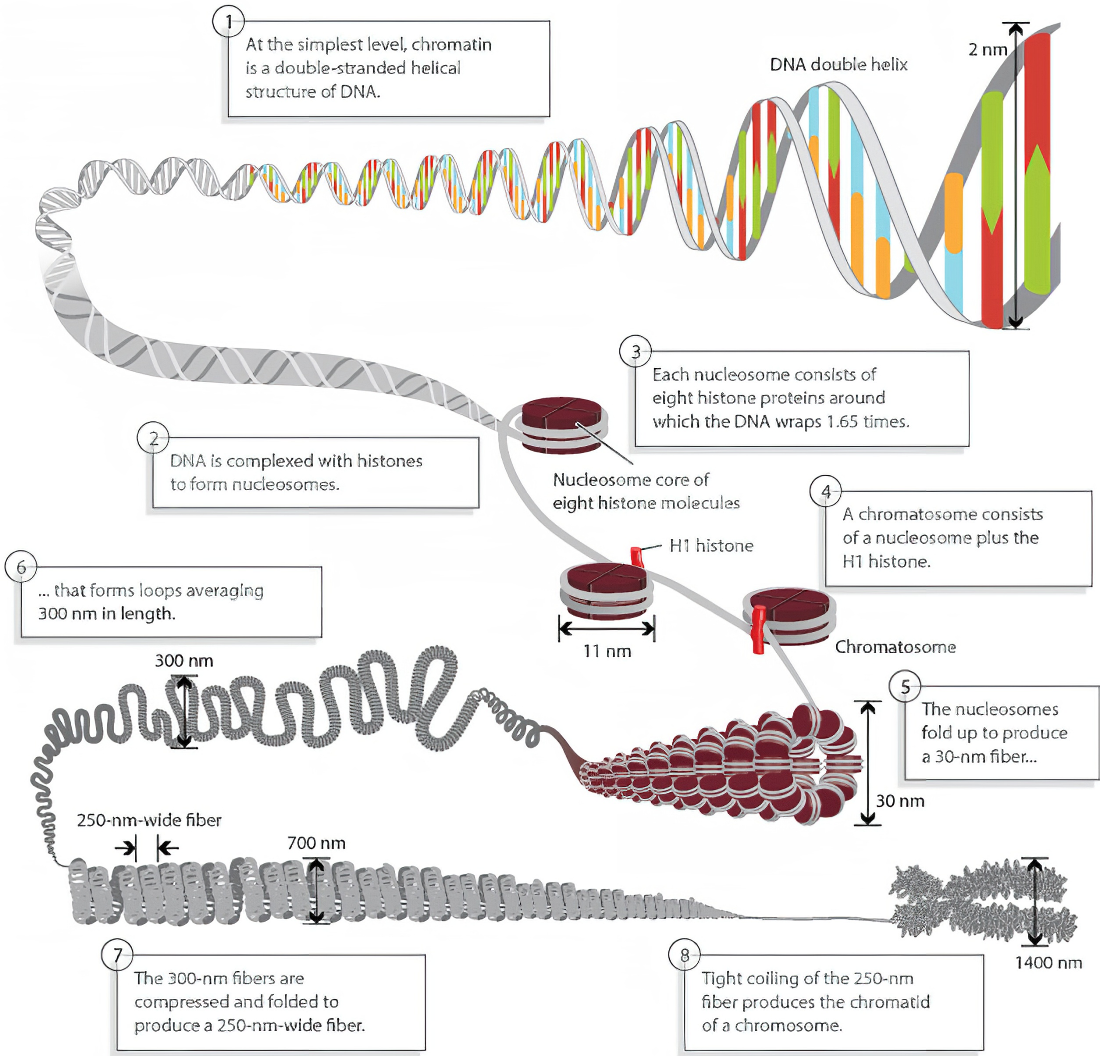

1
At the simplest level, chromatin is a double-stranded helical structure of DNA.
DNA double helix
2 nm
3
Each nucleosome consists of eight histone proteins around which the DNA wraps 1.65 times.
2
DNA is complexed with histones to form nucleosomes.
Nucleosome core of eight histone molecules
4
A chromatosome consists of a nucleosome plus the H1 histone.
H1 histone
6
... that forms loops averaging 300 nm in length.
11 nm
Chromatosome
300 nm
5
The nucleosomes fold up to produce a 30-nm fiber...
30 nm
250-nm-wide fiber
700 nm
1400 nm
7
The 300-nm fibers are compressed and folded to produce a 250-nm-wide fiber.
8
Tight coiling of the 250-nm fiber produces the chromatid of a chromosome.

Figure 2.5.: DNA organization and structure in the cell nucleus. The two meter long DNA molecule is compactly packed in four different levels enabling it to fit in the cell nucleus of $\sim 10\ \mu m$ diameter.
For source and license information, please refer to the List of Figures.

2.2.1 Epigenetics

The term epigenetics was introduced by Conrad Waddington in 1942 [55], using the ancient Greek επι(epi) for 'on top of / upon / over'[1] and 'genetics'. Waddington states to be inspired by the word 'epigenesis' going back to the Greek philosopher Aristotle; however, the origin and connection of Aristotle and 'epigenesis' are challenged by current research [56]. Nevertheless, Waddington named a branch of biology, but the understanding of epigenetics is different today, and the definition of the term changed over time [57, 58]. Waddington's understanding was 'interactions between genes and their products which bring the phenotype into being.'[57, 58, 59]. The National Human Genome Research Institute (NIH) of the United States defines *epigenetics* as a 'field

[1] https://www.wordreference.com/gren/%ce%b5%cf%80%ce%af

of science that studies heritable changes caused by the activation and deactivation of genes without any change in the underlying DNA sequence of the organism.'[2]. This definition includes modifications on the DNA by the methylation of cytosine (5mC); the post-translational modification of the histones of the chromatin with methylation, acetylation, phosphorylation, ubiquitylation, and sumoylation; or, as introduced in section 2.2, interaction, packing, and accessibility of the chromatin structure [60, 61]. Epigenetic processes explain how a zygote and all its daughter cells are based on the same DNA but develop into different cell types; the epigenetic modifications of a cell are inherited during mitosis. For example, diseases like some cancer types are linked to hypomethylation of CpG islands [60, 61, 62]; paternal inheritance of autism in three generations shows methylation changes in the sperm of the fathers [63], or schizophrenia in the offspring of women who were malnourished in the first trimester of the pregnancy [61]. Especially the last two examples provide insights into an inheritance of epigenetic patterns to the offspring and that not only the raw DNA is inherited.

DNA methylation

The methylation of DNA is a regulatory mechanism to silence genes that is inheritable and reversible. The DNA methyltransferase enzymes (DNMTs) are responsible for adding methylation to the 5th position of the cytosine pyridine, called 5-methylcytosine (5mC). The methylation is mostly present in CpG islands; however, the methylation upstream in the promoter's CpG islands is essential for an active gene expression. Methylation of this region leads to a repression of the specific downstream gene. It prevents the detection of the promoter region by the RNA polymerase and specific transcription factors. Low methylation correlates to open chromatin, see *mCHH* and PC1 (positive values) tracks in Figure 2.8 (Nothjunge *et al.* 2017 [64]). However, the methylation of the gene bodies has the opposite effect. It was shown that the methylation of gene bodies is correlated with the expression, but a demethylation causes a downregulation [65]. The methylation is a reversible process, and it was shown that first, the removal of DNMTs in human embryonic stem cells leads to a rapid cell death [66] and second, hypo- and hyper-methylation of the DNA is a characteristic of cancer cells [67].

[2] https://www.genome.gov/genetics-glossary/Epigenetics

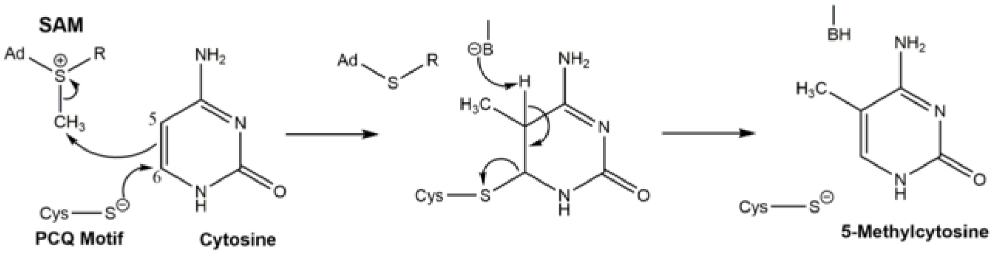


Figure 2.6.: Chemical reaction mechanism of the methylation of Cytosin with DNMTs to 5-methylcytosin (5mC).
For source and license information, please refer to the List of Figures.

Histone methylation

Chromatin is present in the cell in two forms: the open and accessible euchromatin, and the more closed and denser heterochromatin. The euchromatin is accessible to the RNA polymerase and transcription factors, and consequently, the genes can be expressed. The heterochromatin is, due to its denser and more packed nature, not accessible, and the regions located in the hetrochromatin are inactive. The modifications can be methylation, acetylation, phosphorylation, ubiquitylation, and sumoylation of a specific amino acid residue on the N terminus of the histone protein, e.g., methylation on the lysines or phosphorylation on serine or threonine[3] [60]; the different modifications of the histones are visualized in Figure 2.7. The modifications of the histones are correlated to the chromatin features; for example, H3K9me, H3K9me, or H3K27ac are correlated to accessible regions; H3K27me3, or H3K9me3 to denser chromatin [68, 69]; see Figure 2.8. The categorization of open and closed chromatin is a dynamic process and depends on the histones' modification. The modification pattern is inheritable; histones' modifications are first removed and later added again during cell division. The density of the chromatin changes during the cell cycle and can be studied with single-cell Hi-C. Moreover, the euchromatin/heterochromatin division can be calculated out of the Hi-C interaction matrix.

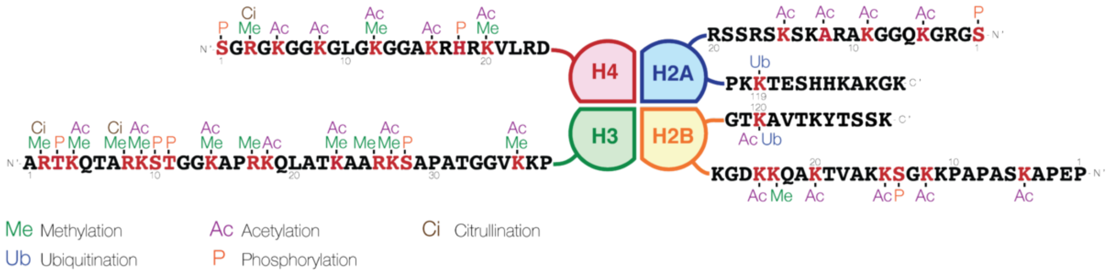


Figure 2.7.: Different modifications of histones H2A, H2B, H3 and H4 on the N terminus.
For source and license information, please refer to the List of Figures.

[3] https://www.abcam.com/epigenetics/histone-modifications

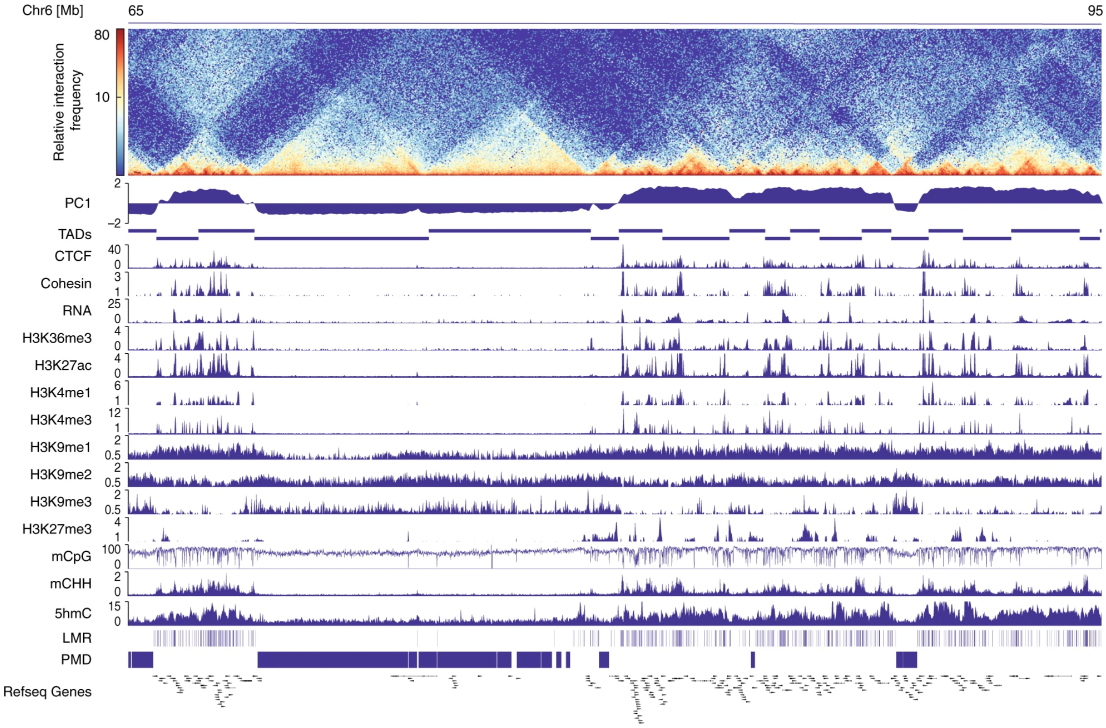


Figure 2.8.: **Top:** Hi-C interaction matrix showing chromatin structure features like topological associated domains (see section 3.1.2). **PC1** The computed principal component 1 of a Hi-C interaction matrix. Positive values correlate to open chromatin, negative values to closed chromatin. This is supported by ChIP-Seq data of modified histones i.e. open chromatin by **H3K36me3, H3K27ac, H3K4me1, H3K4me3** also by the expression of **RNA**; closed chromatin by **H3K9me3**. **mCpG** shows the ratio of methylation of the bases in the region, high methylated regions correlated with closed chromatin, the regions with low methylation with open chromatin; the non-CpG methylated regions (**mCHH**) and 5-hydroxymethylcytosine (**5hmC**) correlate with open chromatin too.
For source and license information, please refer to the List of Figures.

Embryogenesis

After a sperm fertilize the oocyte, the zygote is the first cell in the developing process of a new organism. It is a central scientific question how out of this one cell, many cells with very different characteristics develop. The term epigenetics has its origins in the research of the development of a phenotype, as discussed in the introduction of this chapter, subsection 2.2.1. Embryonic Stem cells (ESC or ES cells) have a reduced or loose chromatin structure compared to differentiated cells, heterochromatin-associated histones present in differentiated cells like H3K9me2 are less present; the euchromatin-associated histone H3K9ac decreases after the differentiation [70]. Chromatin structures like compartments or TADs are not present in the zygote and its totipotent state; it seems to reform with the change from totipotency to pluripotency [71]. It remains unclear if the loss of the structure is a feature necessary for totipotency or is a product of chromatin rearrangements. Genes associated with pluripotency are located in euchromatin, while differentiation-associated genes are in heterochromatic regions. In differentiated cells, this is inverted; the pluripotency genes are in heterochromatin, the differentiation is associated in euchromatin [70]. A deep understanding of epigenetic reprogramming processes during embryogenesis will help understand the factors involved in reprogramming and cell differentiation. With the knowledge, somatic cells can be reprogrammed to pluripotent cells, some factors involved in this process are already known. Especially in the context of a therapeutic usage, e.g., to replace or heal organs or damaged neural pathways, a medical use case is given. A current review of the developments in embryonic stem cells is Hutchins *et al.* [71].

2.3 Transcription regulation

The mechanisms of regulating the expression of genes are one of the major aspects of biomedical research. While all body cells have the same DNA, different genes are expressed in different types of cells. The regulation can be separated into two parts: First, the accessibility of the DNA for the RNA polymerase is controlled by methylations, as explained in subsection 2.2.1. Second, the initiation of the transcription process by the RNA polymerase is controlled by various mechanisms. In this context, a special focus is on the enhancer-promoter interactions. Promoters are short DNA segments upstream of the transcription start site (TSS) of a gene, while enhancers can be upstream, downstream, or in between introns [72]. It was observed that the chances of binding of the RNA polymerase to the TSS region and, therefore, a transcription of the gene is higher if the enhancer and promoter region interact with each other [73]. In this process, several factors are involved. A transcription activating protein (Figure 2.9 element 5) binds to the enhancer region (Figure 2.9 element 2). The binding causes a bending of the DNA and attracts mediator proteins, which help to make the contacts to the promoter region (Figure 2.9 element 6 and 3). The contact helps the RNA polymerase (Figure 2.9 element 7) to bind upstream of the gene to the transcription start site and to start the transcription process. Enhancers are not exclusively involved in transcription regulation; repressors and insulators need to be considered too. A crucial part of understanding the regulation process is to know which enhancers interact with which promoter. Enhancers are known to interact with promoters at greater genomic distances and to bypass promoters of closer loci [72]. Moreover, the influence of mutations in enhancer and promoter coding DNA regions on diseases or the evolution of species and their changed interaction pattern is of high interest [74]. Hi-C can be used to measure the contact between the enhancer and promoter regions; however, due to genome-wide interaction detection, specialized Hi-C derivatives such as capture Hi-C are preferred for this task.

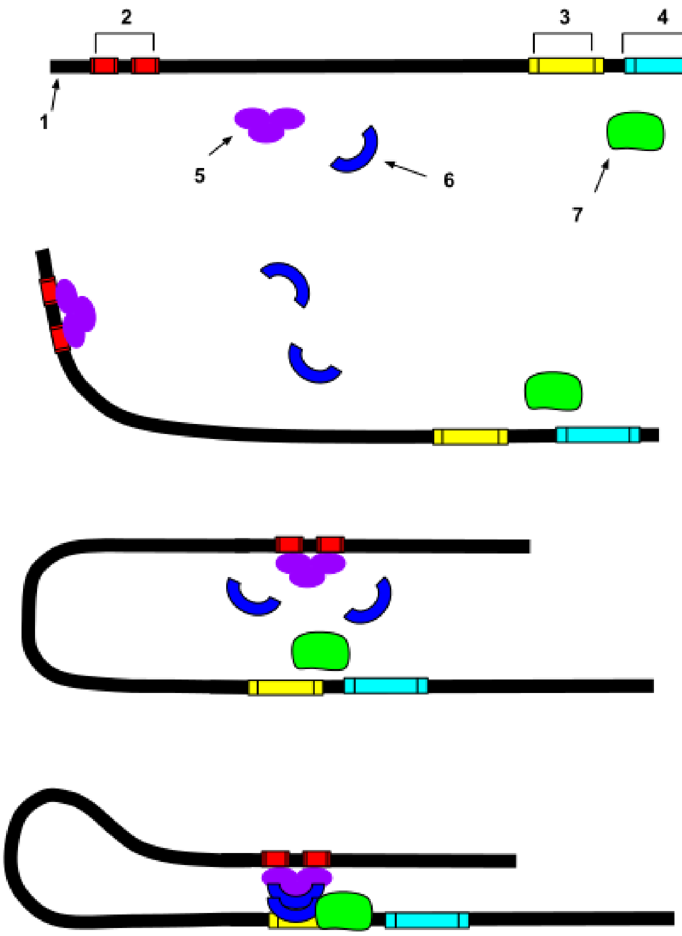
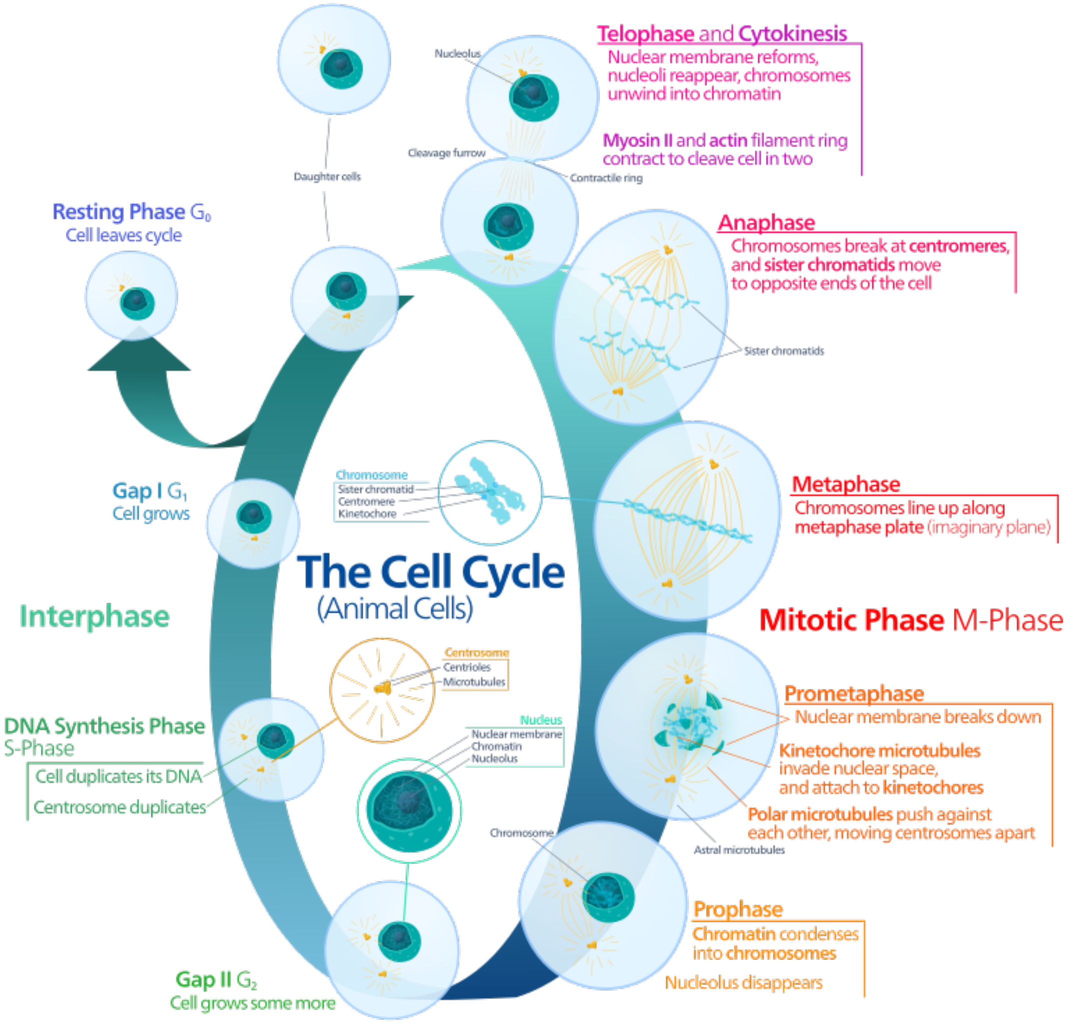


Figure 2.9.: Promoter - enhancer interaction for gene regulation. **First:** 1: DNA, 2: enhancer, 3: promoter, 4: gene, 5: transcription activator protein, 6: mediator protein, 7: RNA polymerase. **Second:** Transcription factor protein binds to the enhancer and starts to bend the DNA. **Third:** Enhancer region is in spatial space to the promoter region and recruits mediator proteins. **Fourth:** Spatial proximity of the transcription activator protein and the mediator protein to the promoter regions enables binding of RNA polymerase to bind downstream of the promoter sequence and to start the transcription process.
For source and license information, please refer to the List of Figures.

2.4 The cell cycle

A vital aspect of every organism is to grow and renew itself; thus, replication of cells with correct DNA inheritance is crucial. The term cell cycle describes a cell's process to replicate its DNA and divide into two daughter cells containing identical genetic information. The life cycle of a cell is separated roughly into two phases: The interphase with the gap phases G_1, G_2 and between them the synthesis or S-phase, and secondly, the stages of mitosis or M-phase including prophase, prometaphase, metaphase, anaphase and telophase (see Figure 2.10) [75]. Besides the two major phases, the resting phase or G_0 can describe cells that no longer divide like certain muscle or neuronal cells. The regulation of the cell cycle and therefore the transition of one phase to the next one is controlled by checkpoints; for example, the restriction (R) point of the G_1 phase must be passed to start the DNA replication of the S-phase; other checkpoints are in G2 and within the M-phase. The cell cycle starts in G_1 after the mitosis, needs to pass the R point, and replicates the DNA followed by G_2. To start mitosis, the checkpoint at the transition from G2 to mitosis needs to pass. The checkpoints have the role in guaranteeing that a cell cycle phase is successfully passed before the next phase starts. For example, the DNA must be identically replicated during the S-phase; an incomplete or erroneous replication can lead to life-threatening implications for the organism. The checkpoints are regulated by two classes of proteins, cyclins, and cyclin-dependent kinases. The stages of mitosis lead to cell division. In the interphase, the chromosomes have been duplicated, and after entering the first phase of the mitosis, the prophase, the chromatin starts to condense, and the well-known shape of the two chromatids (see Figure 2.12) is evolving. In the prometaphase, the nuclear membrane breaks, and the chromosomes start to relocate to the former center of the cell nucleus; in the metaphase, the chromosomes line up on one imaginary plane. Next, the chromosomes separate at the centromeres, and the chromatids move to the opposite ends of the cell; this phase is named anaphase. At this stage, it is guaranteed that the daughter cells inherit a complete chromosome set. Last, the telophase and cytokinesis: the nuclear membrane forms for each daughter cells, the chromosomes recondense to chromatin and end the mitotic phase of the cell cycle; the interphase starts again.



Telophase and Cytokinesis
Nuclear membrane reforms, nucleoli reappear, chromosomes unwind into chromatin
Myosin II and actin filament ring contract to cleave cell in two
Nucleolus
Cleavage furrow
Contractile ring
Daughter cells
Resting Phase G_0
Cell leaves cycle
Anaphase
Chromosomes break at centromeres, and sister chromatids move to opposite ends of the cell
Sister chromatids
Gap I G_1
Cell grows
Chromosome
Sister chromatid
Centromere
Kinetochore
Metaphase
Chromosomes line up along metaphase plate (imaginary plane)
The Cell Cycle
(Animal Cells)
Interphase
Mitotic Phase M-Phase
Centrosome
Centrioles
Microtubules
Prometaphase
Nuclear membrane breaks down
Kinetochore microtubules invade nuclear space, and attach to kinetochores
Polar microtubules push against each other, moving centrosomes apart
Astral microtubules
DNA Synthesis Phase
S-Phase
Cell duplicates its DNA
Centrosome duplicates
Nucleus
Nuclear membrane
Chromatin
Nucleolus
Chromosome
Prophase
Chromatin condenses into chromosomes
Nucleolus disappears
Gap II G_2
Cell grows some more

Figure 2.10.: The cell cycle in animals. The cell cycle is the process to duplicate a cell with the identical set of chromosomes. It can be divided into the interphase with an active metabolism and DNA replication (middle left G_1, S and G_2) and the mitotic (M) phase where the cell actively separates into two daughter cells (from bottom right prophase counter-clockwise to telophase and cytokinesis).
For source and license information, please refer to the List of Figures.

2.5 Chromosome conformation capture

Chromosome conformation capture (3C) [8] and its successors 4C [9, 10, 76, 77], 5C [11] and Hi-C [12] are standard technologies to study the 3D conformation of chromatin. Moreover, specialized deviates like capture Hi-C [15], single-cell Hi-C [16] or combinations with other sequencing technologies like ChIP-Seq named HiChIP [17] exists. The common point of all these technologies is to provide insights into the processes involved in chromatin folding, gene regulation and cell differentiation.

Chromosome conformation capture (3C) is a protocol developed by Job Dekker *et al.* in 2002 [8]. It can capture two specific genomic loci if they are in close spatial proximity. The capture of two DNA regions is called an *DNA interaction*. The model works by applying consecutive steps: crosslinking two close spatial regions with formaldehyde, digestion with a restriction enzyme, and an intramolecular ligation of the sticky ends is applied on the digested DNA fragments. In the next step, the crosslinking is reversed, and the ligated, chimeric DNA fragment can be detected with methods based on the polymerase chain reaction (PCR); see Figure 2.11 3C. Dekker *et al.* used the restriction enzyme EcoRI; however, today other restriction enzymes like HindIII, DpnII, or MboI are used. Use of the three restriction enzyme has given good results in Hi-C experiments; and they are routinely used in the wet-labs. The read resulting from the sequencing process is a chimeric one; the forward and reverse read must be mapped independently. The combined two independent locations define the interaction between two regions.

The 4C method was developed based on 3C method to capture one genomic loci's interaction with all other possible genomic loci. Simonis *et al.* [9], Zhao et. al. [10], Lomvardas et. al. [76] and Würtele et. al. [77] independently developed four similar 4C techniques; a detailed comparison is provided by Sati&Cavalli [78]. Simonis *et al.* is the most widley used approach and extends the 3C protocol by applying a second digestion step by a restriction enzyme, which has a higher cleavage efficiency and a different recognition motif. Following this step, a DNA circle formation of the sticky ends of the second restriction enzyme cut sites is enforced; see Figure 2.11 4C.

Chromosome conformation capture carbon copy (5C) was developed in 2007 by Dostie *et al.* [11] and extends the 3C protocol from Dekker *et al.* [8] to be able to perform a 'many-vs-many'-loci chromatin contacts detection. The extension adds '[...] highly multiplexed ligation-mediated amplification (LMA) to first copy and then amplify parts of the 3C library[...]' [11] to the regular 3C protocol before the PCR sequencing or microarray detection is applied; see Figure 2.11 5C.

The high-throughput variant of the 3C technologies is the Hi-C protocol by Lieberman-Aiden from 2009 [12]. It can capture in an unbiased approach all genome-wide interactions of all genomic loci. The protocol extends 3C by adding biotin after the digestion to mark the cut sites before they are ligated. After a sonication step to break the ligated

DNA into smaller pieces, the sequences labeled with biotin are pulled by a streptavidin bead. Streptavidin and biotin have a highly selective and strong bonding, and are often used to mark and select DNA sequences [79, 80]. Only chimeric reads are amplified with the PCR; see Figure 2.11 Hi-C. The Hi-C protocol was improved in 2014 by Rao *et al.* [14] by applying crucial steps in situ; therefore, the method is called in situ Hi-C. In 2018 the Arima Hi-C protocol (Arima Genomics) improved in situ Hi-C by adding a second restriction enzyme to the digestion phase. The usage of two different restriction enzymes creates a higher number of restriction site combinations and this is assumed to lead a higher number of valid reads [81].

Hi-C can show the contacts of all genomic loci; however, the interaction values are often too small to detect specific interactions. Due to the interaction matrix's squared nature, the growth factor in increasing the read coverage is quadratic. Moreover, if only specific interactions should be investigated in detail, it is a considerable overhead to sequence the entire genome. With capture Hi-C, it is possible to capture predefined genomic loci interactions with all other possible reference genome locations. In contrast to 4C, multiple locations with high-throughput sequencing can be analyzed at the same time.

Hi-C uses up to millions of cells (e.g. Rao *et al.* [14] with two to five million cells) to create the chimeric reads for one interaction matrix. This accumulation has the disadvantage of getting only a cumulative insight of the Hi-C contacts and does not differentiate chromatin dynamics and, therefore, interactions during a cell cycle. Also, differentiation between cell types and their specific interaction patterns is less meaningful if a accumulation over many cells is investigated. It was shown by microscope technologies like FISH that chromatin is highly dynamic (for example, [83, 84]); the cumulative measured data present in Hi-C and derived structures might be statistical artifacts. Nagano [16] introduced in 2013 the first single-cell Hi-C protocol by extending the Hi-C protocol as follows: After the ligation step, the individual cell nuclei are selected under the microscope and placed in individual tubes. On each cell, the regular Hi-C protocol with reversing the crosslinking and pulldown step is performed. After this procedure, the pulled fragments are digested a second time with a different restriction enzyme, and per cell, unique three basepairs long Illumina adapters are added (so-called *barcodes*). Based on the unique barcodes per cell, the reads per cell can be selected *in silico*. Single-cell Hi-C protocols are regularly improved and extended to increase the number of cells or the number of reads [85, 86, 87, 88, 89].

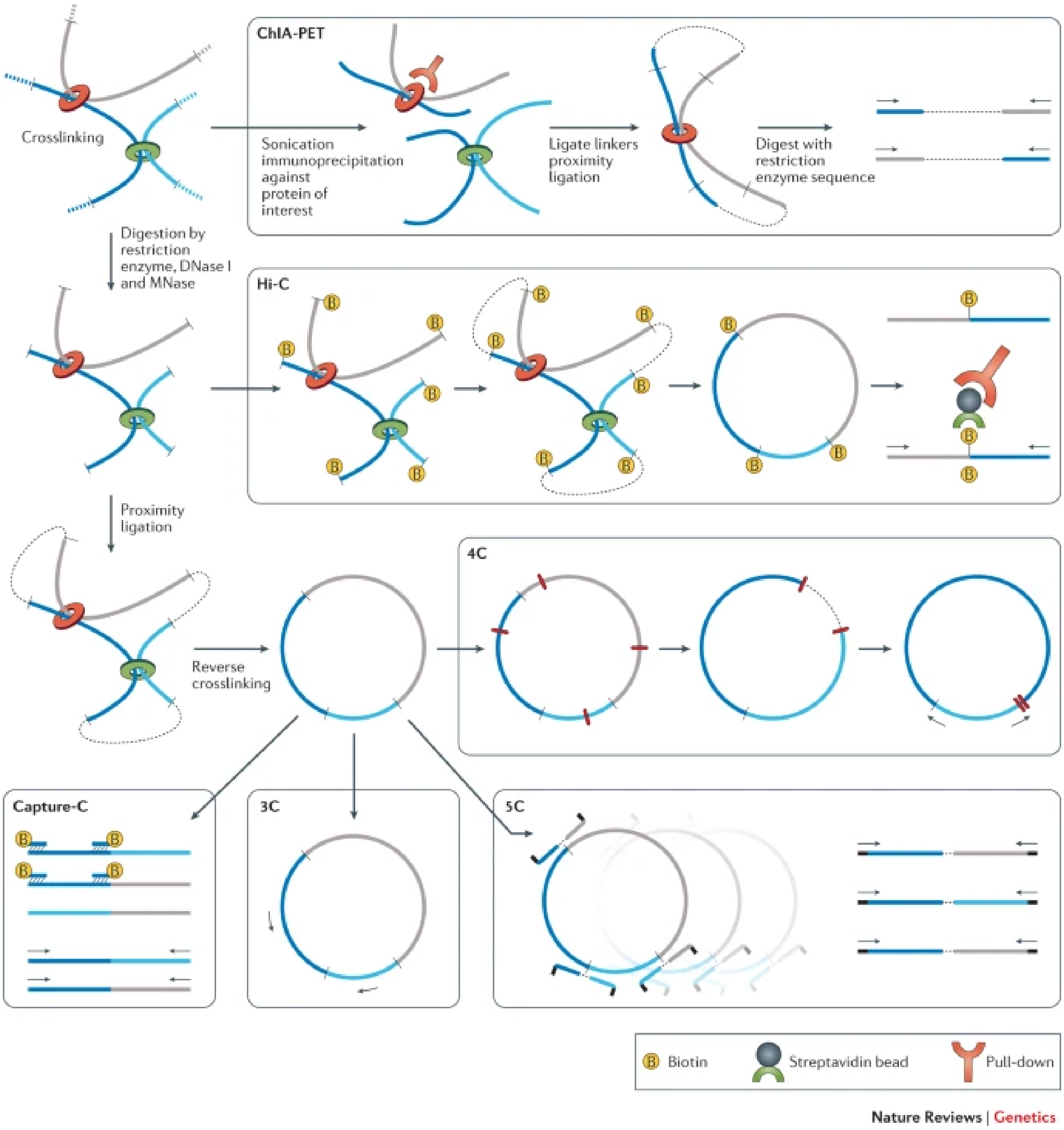

ChIA-PET
Crosslinking
Sonication immunoprecipitation against protein of interest
Ligate linkers proximity ligation
Digest with restriction enzyme sequence
Digestion by restriction enzyme, DNase I and MNase
Hi-C
B
Proximity ligation
4C
Reverse crosslinking
Capture-C
3C
5C
Biotin
Streptavidin bead
Pull-down
Nature Reviews | Genetics

Figure 2.11.: Overview of 3C methods and derivatives. **Upper left:** Crosslinking of spatial close proximities and digestion with a restriction enzyme. The restriction enzymes cut sites ('sticky ends') are ligated and the crosslinks are removed. Depending on the 3C derivative, additional steps are performed: **ChIA-PET** preselects protein of interest directly after the crosslinking (see [82] for details); **Hi-C** adds biotin before the ligation to mark the restriction cut sites to be able to select them for PCR sequencing; **4C** adds a second digestion step; **5C** copies and amplifies the crosslink removed chimeric sequence; and **Capture-C** sonics the chimeric DNA sequences and captures regions of interest by special designed oligonucleotide sequences.
For source and license information, please refer to the List of Figures.

2.6 Structural DNA elements

Chromosomes are the highest structural organization form of the compacted DNA within the cell nuclei. It is composed of the consistent DNA thread wrapped around histones and compacted (compare to Figure 2.5). The well-known image of clear dense chromosomes in the nuclei is only present at mitosis [90] (Figure 2.12), but the majority of the time, the cell cycle is in another phase. For a long time it remained unresolved if chromosomes have a territory (as suggested by Carl Rabl in 1885 [91]) or are wildly mixed within the nucleus. Cremer *et al.* [92] showed in 1982 that chromosomes have territories, this result was confirmed by FISH experiments from Parada *et al.* in 2002 [93] (Figure 2.13 from Bolzer *et al.* [94]). It was shown that the location of a chromosome remains at a similar locus following cell division [95]. Hi-C contacts, represented in a Hi-C interaction matrix, Figure 2.14d, show that each chromosome has a high intra-chromosomal interaction frequency, but that the number of interactions drops significantly for inter-chromosomal interactions. This supports the observations by Cremer and Parada that chromosomes have a territory.

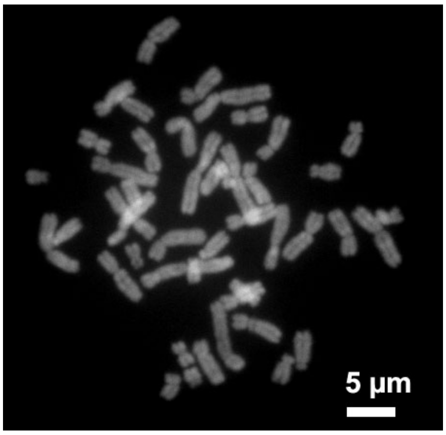


Figure 2.12.: Chromosomes in the cell nuclei in a compact form during mitosis. For source and license information, please refer to the List of Figures.

The chromatin is categorized in euchromatin (open) and heterochromatin (closed) [96], these two categories are also called A (open) and B (closed) compartments or A/B compartments [12], because A/B compartments are correlated with known features of euchromatin and heterochromatin. The differentiation is based on the biological properties of these chromatin regions. The heterochromatin is dense-packed chromatin mostly located at the centromere and telomere, repressing DNA transcription, while euchromatin is more open, active for gene transcription, and contains gene-rich regions. Both regions are defined by specific histone modifications and enrichments of proteins: euchromatin has enriched H3K4me, while heterochromatin is enriched for H3K9me and HP1α [96, 97]. Heterochromatic regions can influence neighboring euchromatic regions and repress gene transcription [98]. Lieberman-Aiden *et al.* [12] showed that

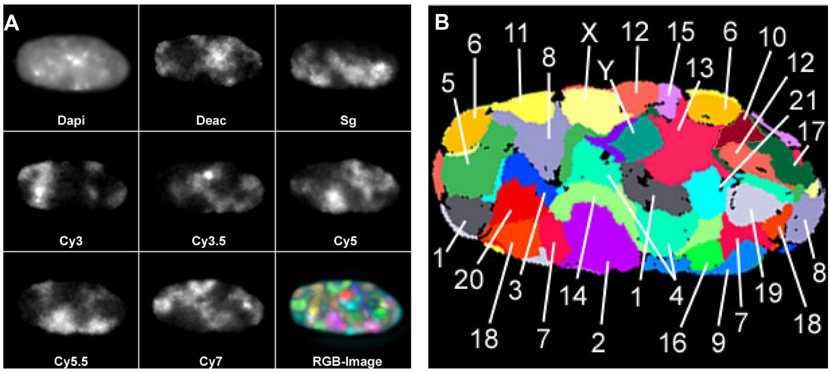


Figure 2.13.: Chromosomes in the cell nuclei. **A**: Wide-field microscopy using eight channels: 1. DNA counterstain, and the seven following use different fluorescent markers. RGB image is a compilation of the seven fluorescent images. **B**: False color image with the chromosome labels.
For source and license information, please refer to the List of Figures.

A/B compartments can be computed using the Hi-C interaction matrix, see section 3.1.2 and Figure 2.14c.

Topological associated domains (TADs) are regions at the main diagonal of the Hi-C matrix with a triangular shape, covering high interacting regions of the chromatin within regions with a median size of 880 kb [13], see Figure 2.14b. TADs are intra-chromosomal regions where the boundaries have an enrichment of the 11-zinc finger protein CTCF and cohesin. Their functional role in the gene regulation process is not fully understood; studies like [7] showed the impact of TAD boundaries and its changes to gene expression and therefore the influence of TAD structures to cause diseases; however, other results challenge these findings [99]. Moreover, it is unclear if a gene and regulating factors are within one TAD if the spatial proximity implied by the high contact number results in physical proximity; and if this physical proximity is the cause for activation of the transcription [100].

DNA loops are single point enriched regions in respective to their background and represent enhancer-promoter interactions, gene loops, architectural loops or polycomb-mediated loops [101] (Figure 2.14a). DNA loops are bound by CTCF and cohesin-associated proteins [102], and their size is usually limited by a CTCF binding motif on the DNA (CCGCGNGGNGGCA) [103] and its next inverted CTCF binding motif [14]. The role of DNA loops for the DNA structure and its functionality within the nucleus is an open question. The singular point interactions are associated with enhancer-promoter interactions. However, the strong correlation with CTCF and cohesin, which are correlated to DNA repair responses, raises the question if observed loops are also dynamic appearing locations indicating a DNA repair process. Both enhancer-promoter and DNA repair responses need to be investigated with single-cell Hi-C data because Hi-C is an accumulation over millions of cells. However, the resolution of single-cell Hi-C needs to be improved first.

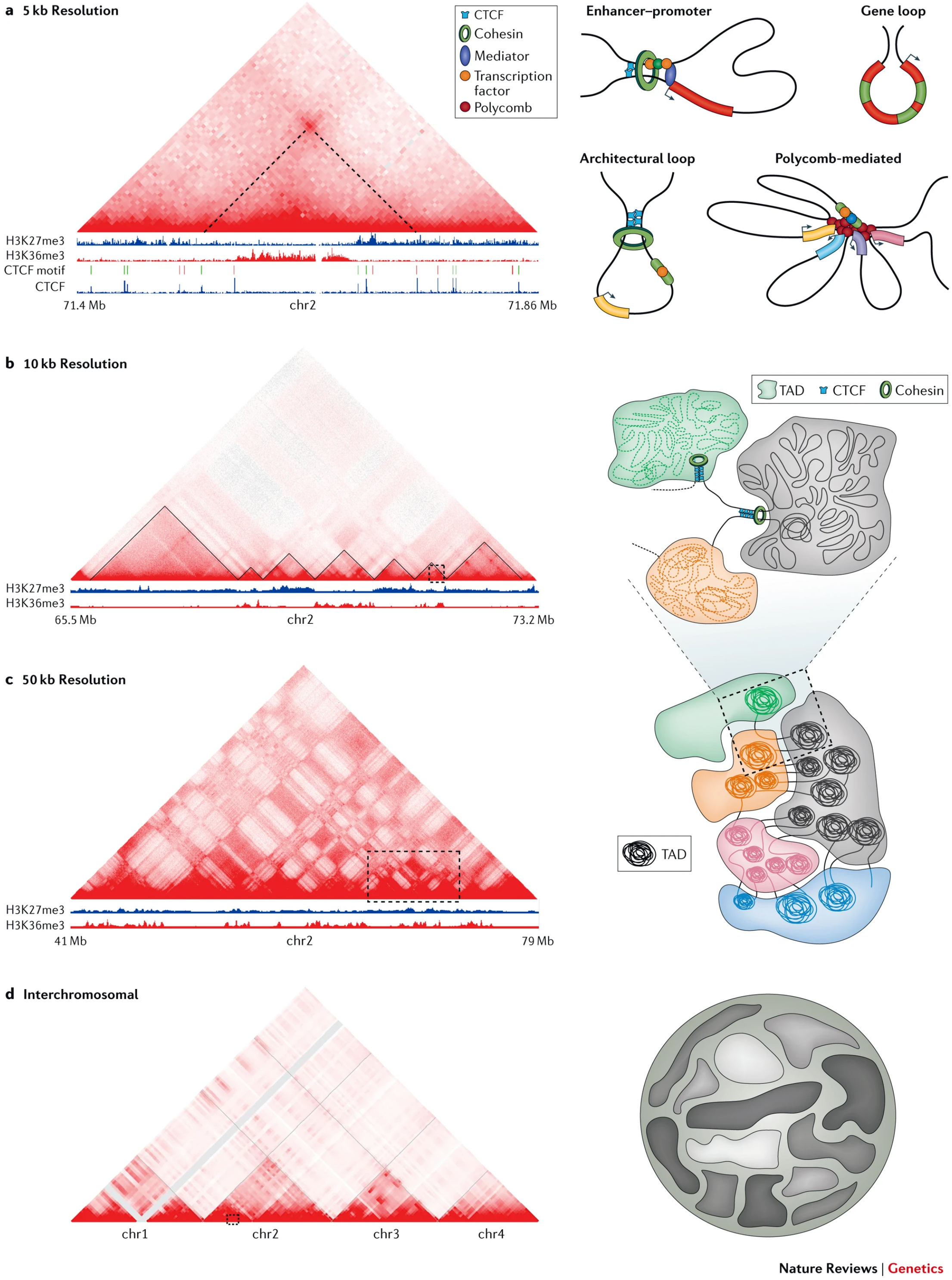


Figure 2.14.: Structural DNA elements. **a)** A loop is a single enriched region in relation to its local neighborhood. DNA loops are associated with enhancer-promoter interactions, gene loops, architectural loops or polycomb-mediated loops. **b)** Topological associated domains (TADs) are triangular shaped regions with a high interaction frequency. **c)** A/B compartments are correlated with euchromatin and heterochomatin. **d)** Chromosomes have their own territories within the cell nucleus and correspond to the enriched intra-chromosomal contacts and very few inter-chromosomal contacts.
For source and license information, please refer to the List of Figures.

2.7 Related software

Scientific research is based on the ability to recreate and reproduce experiments to prove the researchers' claims' correctness or falsify them. In today's data and software-driven research, a problematic aspect is that in many cases, the used data analysis software is not easily reusable because it may contain hard-coded paths, imports of undefined dependencies, or uses outdated and possibly unavailable dependencies. Moreover, even if a software is reusable, the identical versions of the software and its dependencies must be used for an exact reproduction of published research results, but these are often unknown. A newer version usually contains bug fixes, corrected methods, or an API change and is no longer compatible. Another aspect is the desired user group of bioinformatics software: biomedical and pharmaceutical researchers who are not familiar with installation routines, the bash, solving dependencies, or even any kind of programming. Biomedical researchers should focus on the data analysis itself and not on how to get the software running.

2.7.1 Conda

The Conda package manager is a package manager to distribute and install software of any programming language. Compared to classical package managers from the Linux eco-system like apt, yum, or Pacman, Conda runs with user rights, supports environments to separate installations, and enables the usage of multiple versions of the same software. The ability to install software from all programming languages without the need to compile on the end-users system, to install the software in any version that was ever available and to install the software within the home directory, and therefore not influencing the system-wide software, makes Conda a very useful package manager for research applications. Conda enables researchers to publish software with well-defined versions of the dependencies, which provides two benefits. First, the software runs on a user system as the software developers intended it. Second, even years later, a specific version of a software package, together with the required dependencies, can be installed within minutes. With these properties, Conda contributes to a higher reproducibility of data-based scientific research. In this thesis, the presented software is available in the bioconda channel [104] of Conda, a widely used repository of Conda packages, focusing on tools with an application in bioinformatics.

2.7.2 Docker

Docker[4] is a container technology and has its origins in cloud computing [105]. The idea is to have a pre-configured software and data environment available with one command and no further configuration, except the one-time installation of the Docker software on the host system, is necessary. In contrast to virtual machines, it shares the host operating system, making it lightweight and faster available. The benefits of using containers in the distribution of scientific software are the possibility of pre-configuring the software in the container, adding the data used for analysis, preserving and archiving the software, and all its dependencies [106]. While the deployment via a package manager is suitable for many users, it cannot guarantee identical versions of all dependencies or only with quite some labor. However, once the software is installed in a container, it stays as it is, even if it is reused multiple years later, and provides, therefore, an important platform for sustainable and reproducible software-driven research.

2.7.3 Galaxy

The Galaxy project [107] is an open-source project focusing on making scientific software accessible through a web browser-based interface. The goal hereby is to avoid user-sided software installation, provide a trackable history of software runs, enable automated workflow processing, and give the administrator of the Galaxy instance the power to decide how many resources one tool can access. Galaxy is scalable; it can run self-administered on a notebook, on a single small server in a lab, or run on a compute cluster in combination with cluster schedulers like HTCondor. The trackable run histories become beneficial for researchers in multiple manners: First, it serves as a work-notebook protocolling each step of analysis, saves the used parameters and tool versions. The protocolling is extremely helpful if many tools and parameters are used, as it can become quite overwhelming and daunting to kept track of these by hand. Second, these histories can be made public and shared, which helps reviewers of manuscripts or, in general, researchers who want to know all the details of the analysis. Many of today's published articles lack precisely this; it is often the case that only the results are shown, but not mentioned which software or in which versions and parameter settings they have been used. Moreover, intermediate files are seldomly published. With publishing each step of the data analysis, from the import of the raw data, all the used intermediate analysis software, its parameters, and intermediate results, to the final results, scientific findings become more trustworthy and less vulnerable to manipulations.

[4] https://www.docker.com/

3 Chromosome conformation capture analysis

3.1 Hi-C data analysis

Data processing and analysis of high-throughput sequencing data require complex and highly individualized software-based approaches. Standardized tools can be used only to a certain amount; usually, the investigation of the base pair calling probability, the trimming of adapter sequences, or the mapping to a reference genome can be accomplished. The specialized software *HiCExplorer* was developed in this thesis to analyze Hi-C data. The following section summarizes the workflow for Hi-C data analysis and the tools provided by HiCExplorer. It is based on a large extent on the publications: Wolff *et al.* 'Galaxy HiCExplorer: a web server for reproducible Hi-C data analysis, quality control and visualization', 2018 [27]; Wolff *et al.* 'Galaxy HiCExplorer 3: a web server for reproducible Hi-C, capture Hi-C and single-cell Hi-C data analysis, quality control and visualization', 2020 [28]; and Lopez-Delisle *et al.* 'pyGenomeTracks: reproducible plots for multivariate genomic datasets', 2021 [31].

3.1.1 Pre-processing

High-throughput sequencing data is generated by sequencing DNA or RNA with sequencing machines, which can sequence up to 1.8 tera base pairs in three days providing an error rate of $< 1\%$ [108]. The sequenced reads are stored with a unique id, quality information, and other metadata in text files; the most common file format is the *FASTQ* format [109]. The first quality control step of the reads is to investigate the base-calling quality with a tool like *FastQC*[1]. The quality score for each base pair is created by the sequencing machine and uses the logarithmically transformed probability error named Phred-Score [110, 111]:

$$Q = -10 * log_{10}P$$

The quality of the individual bases should be higher than a Phred-Score of 30, indicating a correct detection of the base with 99.9%. The occurrence of a lower read quality usually occurs at the start or end of a sequence. The lower quality might indicate the non-removal of adapter sequences; the end sequences have a higher noise ratio

[1] http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

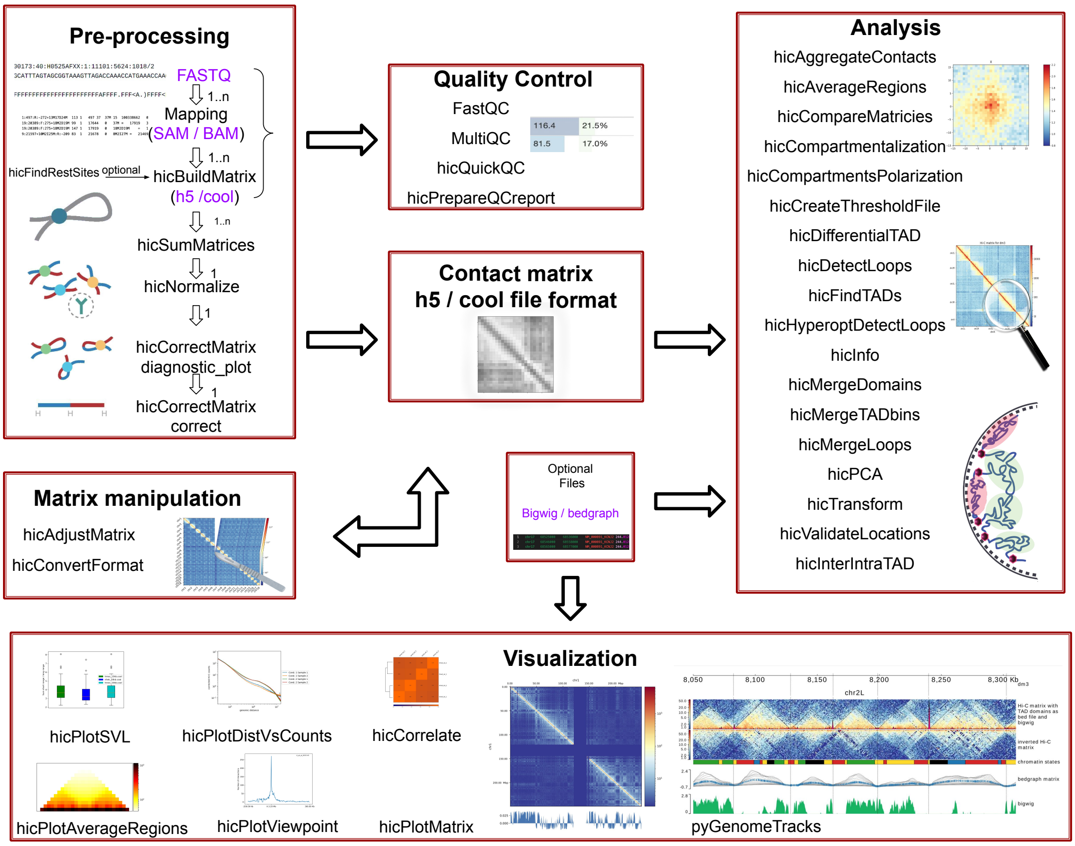


Figure 3.1.: Hi-C data analysis workflow with HiCExplorer. **Pre-processing:** Required input is in FASTQ file format, undergo a quality control with FastQC. Mapping of the raw files with a mapping software of free choice, the tool *hicBuildMatrix* creates the interaction matrix, the central data structure. The data can be normalized and ligation effects be corrected. **Analysis:** HiCExplorer supports various analysis methods e.g. to detect A/B compartments, TADs or loops. Moreover, methods to compare matrices or to validate detected regions with orthogonal data is supported. **Visualization:** The interaction matrix can be plotted with *hicPlotMatrix*, predefined viewpoints (*hicPlotViewpoint*), correlations (*hicCorrelate*) or interaction ratios (*hicPlotSVL*). Hi-C can be visualized integrated with other types of data stored in *bigwig* or *bedgraph* file. *Matrix manipulations:* To remove problematic regions of a matrix the tool *hicAdjustMatrix* can be used; or if the file format of the matrix needs to be converted to a format of an alternative tool for Hi-C data analysis, the tool *hicConvertFormat* provides this functionality.
For source and license information, please refer to the List of Figures.

depending on the sequencing machine. In both cases, it is recommended to remove these base pairs from the sequences.

The sequenced reads for Hi-C are chimeric; therefore, the sequencing uses the pair-end mode, sequencing the start (forward) and end (reverse) of a read. The forward reads represent the genomic location A of a chimeric read and the reverse the genomic location B. The order of the reads in the forward and reverse read file is the crucial relation of the reads, associating each other. The reads' chimeric nature is the fundamental concept of Hi-C and must be kept in mind in all following pre-processing steps.

After the first quality control of the reads, they need to be mapped to a reference genome. Mapping is the computation of a read's genomic location in a given reference genome, and Hi-C data have to be mapped as single-end due to its chimeric nature. Mapping software maps the forward strand and the reverse strand independently to the reference genome, and if the two genomic loci are close to each other, the location is accepted for the read. This principle helps to increase the accuracy of the mapping, but due to the chimeric nature of the Hi-C reads, the forward and reverse read are of a large distance and would be interpreted as faulty reads in pair-end mode. Software to map sequences is for example HISAT2 [112], Bowtie2 [113] or BWA [114].

The mapped forward and reverse reads are used to create a Hi-C data analysis's central data structure: the contact matrix M. This matrix, also called interaction matrix, has on the x- and y-axis the genomic locations, i.e., the value at position (i, j) counts the occurrence of chimeric reads where the forward read was mapped to position i in the reference genome, and the reverse read to position j.

The central data structure is a contact matrix M where the entries $m_{i,j}$ are the observed contacts between the two loci i and j. The contact matrix is a squared and symmetric matrix, $|i| == |j|$.

$$M = \begin{bmatrix} m_{0,0} & \dots & m_{0,j} \\ \vdots & \dots & \vdots \\ m_{i,0} & \dots & m_{i,j} \end{bmatrix} \tag{3.1}$$

The interactions of genomic loci are represented in the interaction matrix per genomic region. A so-called *binning* is applied. The more subsequent genomic regions are accumulated in one pixel of the matrix, the lower the so-called *matrix resolution* is. A matrix with a resolution of 1 megabase pairs (Mb) covers a genomic range of one million base pairs per bin, while a matrix with a resolution of 10 kilobase pairs (kb) covers only 10,000 base pairs per bin. In order to store the genomic position and range information of a particular pixel a list of length i is used:

$$intervals = [(chr, start, end)_0, ..., (chr, start, end)_i] \tag{3.2}$$

Therefore, each data point $m_{i,j}$ of the interaction matrix contains the interaction information of the regions $intervals_i$ and $intervals_j$. Certain factors restrict the resolution of an interaction matrix. The first factor which heavily effects the resolution is the read coverage. The fewer reads are available, the lower the interaction values per bin will be. The matrix will be sparser and the explanatory power is less convincing. Second, the Hi-C protocol also has a major influence on the resolution of the matrix. A per base-pair level resolution is technically impossible because the reads are digested by restriction enzymes that cut DNA at specific patterns. These so-called *restriction enzyme cut sites* represent the highest possible resolution, the *restriction cut site resolution*. Compared to the fixed binned variant, the restriction cut site resolution does have a variable bin size. The regions pooled into one bin are determined by the cut pattern of the restriction enzyme. The occasion of the cut size varies per restriction enzyme but is usually in the range of a few hundred kilobases. The third and last factor influencing the interaction matrix resolution are the available compute resources. The human reference genome has around 3 billion base pairs, assuming a base-pair level resolution, the matrix would have a dimension of $(3\ billion \times 3\ billion)$, using a 10 kb resolution, reduces this to $(300,000 \times 300,000)$. Some operations have high memory requirements, resulting in consumption of several hundred gigabytes of memory for a typical input dataset. While this is today theoretically available, it is still expensive and difficult to access for many researchers.

Interaction matrices are computed with HiCExplorer's *hicBuildMatrix* and the restriction cut sites with the tool *hicFindRestSites*. The computed interaction matrices are stored in either HiCExplorers native file format h5 from Ramírez [115] or in the *cooler* file format [116]. The cooler file format support was added to increase interaction matrices' exchangeability and interoperability between different Hi-C data analysis software. However, many software supports only their self-developed formats, which is in many cases a simple text-based file format. In the better cases, this represents the data's sparsity, but dense text matrix formats like Homer are also available. While binary formats with the ability to compress data and enable random access to regions are highly beneficial, HiCExplorer supports the import and export to several file formats to increase the exchangeable and, therefore, the reusability and sustainability of research data. It supports importing Juicer's .hic format [117], homer, hic-pro [118], cool, and h5; and exports cool, h5, ginteractions [119], hic-pro and Homer file formats.

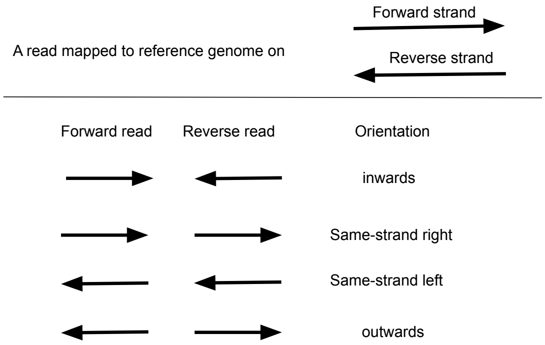


Figure 3.2.: Mapped reads orientation. Hi-C are paired-end reads and contain therefore a forward and reverse read. A forward read mapped to the forward strand and the reverse read to the reverse strand, the orientation of the pair-end read is *inward*. A forward and reverse read mapped to the forward strand, the orientation of the pair-end read is *same-strand right*, if both reads are mapped to the reverse strand the orientation is *same-strand left*. The outward orientation is given if the forward read is mapped to the reverse strand and the reverse read to the forward strand. For source and license information, please refer to the List of Figures.

HiCExplorer's *hicBuildMatrix* creates the interaction matrix based on the mapped reads and serves as a second quality control. A correct Hi-C read is chimeric, and the two interacting DNA regions' genomic loci are from different fragments, the read orientation is inward oriented, a restriction cut site is in between and the two mapped locations have to have a minimum distance. The Hi-C protocol may create certain errors: A fragment from one location can be ligated not with another fragment, but with its own other end. In this case, three categories are differentiated: circularised DNA, dangling ends and internal fragments. All three have inward orientation (see Figure 3.2 for orientation information), the circularised DNA contains a restriction cut site within the read (self-ligation, reads mapped within 800 base pairs), the dangling end has one restriction cut site at the end, and the internal fragment contains no restriction cut site (Figure 3.3c, d, e). Another error are reads containing multiple fragments and/or continuous fragments (Figure 3.3b), or are simple PCR duplicates (Figure 3.3f) [120]. Additional to this, self-circles are reads mapped within 25 kb, and have an outward orientation. HiCExplorer provides a quality report stating the findings of the raw Hi-C data containing information about the amount of reads in total, valid Hi-C reads, how many and why reads have been filtered out, information about the distribution of the orientation and the amount of intra-chromosomal short (< 20 kb) and long range ($>= 20$ kb) contacts (Figure 3.4); multiple HiCExplorer quality reports can be investigated and pooled together in one report with MultiQC [121][2], see Figure 3.31.

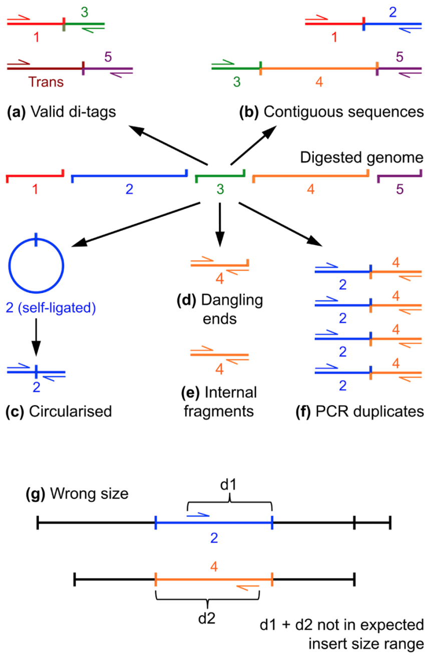
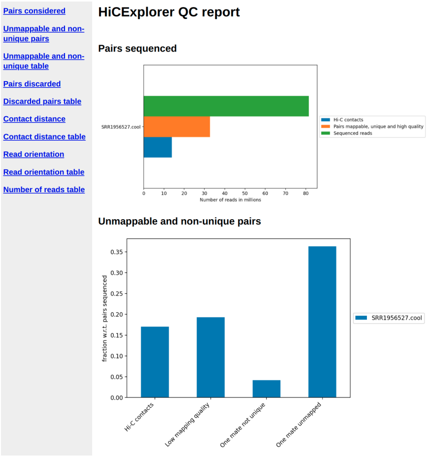


Figure 3.3.: Potential read errors to filter out at the creation time of the interaction matrix. For source and license information, please refer to the List of Figures.

[2] https://github.com/ewels/MultiQC/blob/master/multiqc/modules/hicexplorer/hicexplorer.py



Pairs considered

Unmappable and non-unique pairs

Unmappable and non-unique table

Pairs discarded

Discarded pairs table

Contact distance

Contact distance table

Read orientation

Read orientation table

Number of reads table

HiCExplorer QC report

Pairs sequenced

SRR1956527.cool
Hi-C contacts
Pairs mappable, unique and high quality
Sequenced reads
0
10
20
30
40
50
60
70
80
Number of reads in millions

Unmappable and non-unique pairs

Figure 3.4.: Quality control report of HiCExplorer on mouse mm9 data from [122]. For source and license information, please refer to the List of Figures.

The raw Hi-C matrix can be subsequently normalized, especially if data from two replicates or conditions have to be compared downstream in the analysis process. HiCExplorer offers normalization to an equal read coverage, to the norm range of 0 to 1, or by a user-given multiplicative value. Imakaev [23] introduced 'iterative correction and eigenvector decomposition' (ICE) and Rao *et al.* [14] used the matrix balancing algorithm from Knight and Ruiz [22] to correct Hi-C data. One key assumption is made to correct Hi-C interaction matrices: The number of ligation products is proportional to the contact probabilities of genomic loci; all genomic loci should have in sum the same amount of interactions with all other loci. Recent publications challenge this assumption [100], questioning the interpretation of the contacts generated by crosslinking and ligation as it is applied in Hi-C.

3.1.2 Analysis

The analysis of Hi-C data is multi-variant and provides deep insights into the chromatin structure. The introduced chromatin structures of section 2.6 can be computed out of a Hi-C interaction matrix. The principal component analysis detects euchromatin and heterochromatin; topologically associated domain boundaries are computed by detecting the lower amount of interactions between TADs; the loops are recognized as single point enriched interactions. Moreover, various other methods exist: The average contact structure of specified regions to analyze the global TAD structure, an aggregation of user-predefined regions to detect enrichment of contacts at multiple locations, or a correlation of Hi-C interaction matrices. In the following section, the analysis methods are discussed in detail.

A/B compartments

A/B compartments or the euchromatin/heterochromatin regions of the chromatin can be computed based on Hi-C data. Lieberman-Aiden [12] described the following algorithm to compute it, the computation is calculated on intra-chromosomal data and independent per chromosome.

First, the Hi-C interaction matrix is converted to an observed/expected matrix. The observed values are given by the contact matrix M with its entries $m_{i,j}$. How the expected value is computed can differ, depending on the genome, the read coverage or if proximity ligation effects should be corrected. Lieberman-Aiden [12] computes the expected value using the maximal possible amount of contacts of a distance per chromosome

$$N_d = \{m_{i,j} \mid |i-j| = d\} \tag{3.3}$$

$$exp_{i,j} = \frac{\sum m_{i,j}}{|N_d|} \; \forall i,j : |i-j| = d \tag{3.4}$$

This means that we have the same expected value for each genomic distance d: $exp_{i,j} = exp_d \mid \forall i,j : |i-j| = d$.

Alternatively, N_d can be computed considering only the non-zero contacts:

$$N_d = \{m_{i,j} \mid |i-j| = d \wedge m_{i,j} \neq 0\} \tag{3.5}$$

Homer software computes the expected value 'assuming each region has an equal chance of interacting with every other region in the genome and that regions are expected to interact depending on their linear distance along the chromosome.'[3]:

$$exp'_{i,j} = exp_{i,j} * \frac{\sum_{k=0}^{n} m_{i,k} * \sum_{k=0}^{n} m_{k,j}}{\sum_{k=0}^{n} \sum_{l=0}^{n} m_{k,l}} \tag{3.6}$$

The observed/expected matrix is named $M*$. Each entry is defined as:

$$m^*_{i,j} = \frac{m_{i,j}}{exp_{i,j}} \tag{3.7}$$

Second, a principal component analysis is applied on the observed/expected matrix M^* by computing the covariance matrix Cov and eigenvector decomposition:

$$\bar{m}^*_i = \frac{\sum_{k=0}^{n} m^*_{i,k}}{n} \tag{3.8}$$

[3] http://homer.ucsd.edu/homer/interactions/HiCmatrices.html

$$cov_{i,j} = \frac{\sum_{k=1}^{n}(m^*_{i,k} - \bar{m}^*_i)(m^*_{k,j} - \bar{m}^*_j)}{n-1} \tag{3.9}$$

$$Cov * v = \lambda * v \tag{3.10}$$

where v is a $n * 1$ eigenvector and λ its associated eigenvalue. The eigenvectors of the largest and second-largest eigenvalues, v_1 and v_2 respectively, are used as principal component 1 (PC1) and principal component 2 (PC2). Lieberman-Aiden defined in [12] the positive values of a principal component associated with open or A compartment, and negative values of a principal component as associated with closed or B compartment. Eigenvectors are unique up to the sign for the same eigenvalue λ:

$$Cov * (-v) = -(Cov * v) = -(\lambda * v) = \lambda * (-v) \tag{3.11}$$

The algorithm used to solve the eigenvector decomposition determines, therefore, whether the sign needs to be flipped to fulfill Lieberman-Aidens definition. Moreover, the implementation of computing A/B compartments deals with several software-based issues. The computation is based on the libraries *NumPy*[4] and *SciPy*[5]. These libraries can use differing Linear Algebra libraries[6] in the back-end for the computation, like OpenBLAS[7] or Intel's MKL[8]. To the user of HiCExplorer, it is relatively intransparent which implementation is used. Furthermore, it should be mentioned that the algorithms to compute the eigenvector decomposition are approximative algorithms. In Figure 3.5 it can be seen that the sign of the eigenvectors varies depending on the software version of HiCExplorer and the *NumPy* and *SciPy* dependencies. To hold the definition of Lieberman-Aiden, a method to flip the sign of the eigenvector is provided. The following consideration is taken into account. Open (A) chromatin is associated with gene expression, and therefore the occurrence of known genes in the open compartment is higher, see the PC1 and Refseq Genes track in Figure 2.8. For this reason, the gene occurrences per bin of the Hi-C matrix are counted, and the result is correlated using the Pearson correlation to the eigenvector. Depending on a positive or negative correlation, the eigenvector values can change their sign.

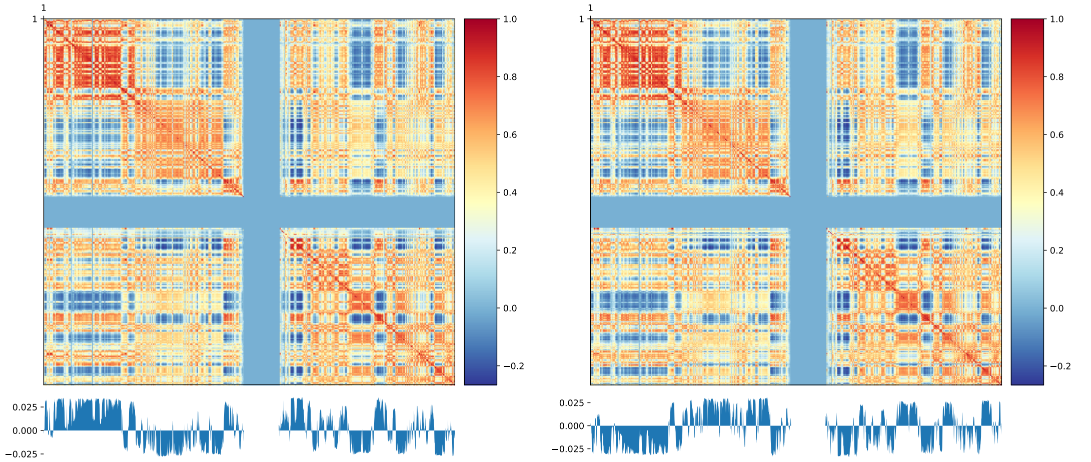
[4] https://numpy.org/

[5] https://www.scipy.org/

[6] https://numpy.org/doc/stable/user/building.html

[7] https://www.openblas.net/

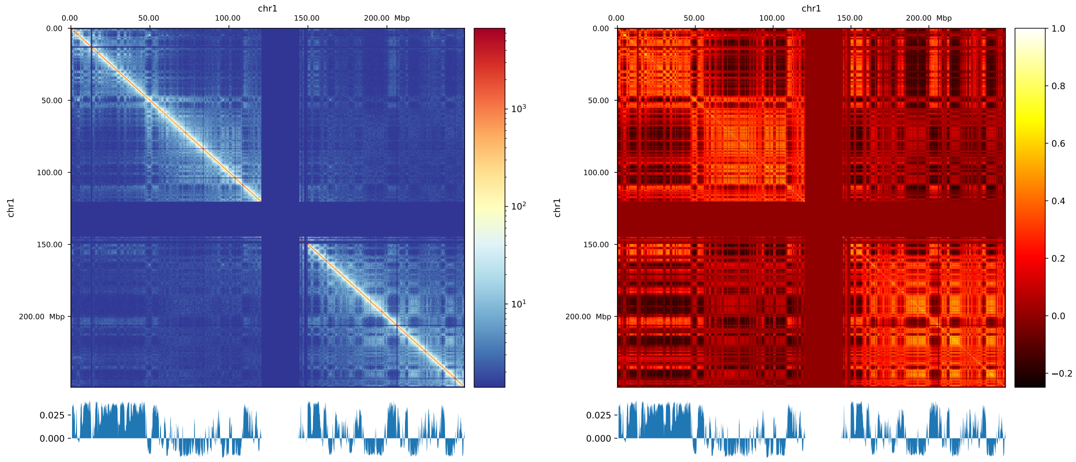
[8] https://software.intel.com/content/www/us/en/develop/tools/oneapi/components/onemkl.htmlgs.94c9j2



(a) HiCExplorer 2.1.4, scipy version 1.0, numpy version 1.13. PC1

(b) HiCExplorer 3.3.1, scipy version 1.3, numpy version 1.17. PC1

Figure 3.5.: Different PCA tracks resulting from different software versions.
For source and license information, please refer to the List of Figures.



(a) Hi-C matrix and PC 1

(b) Pearson correlation matrix and PC 1

Figure 3.6.: **(a)** A/B compartments on GM12878 chromosome 1, 100 kb from Rao *et al.* [14]. Positive values are associated to open, negative values to closed chromatin. The void region is the centromeric region of the chromosome and contains no data. Regular Hi-C matrix and the first principal component 1. **(b)** Hi-C matrix transformed to Pearson correlation matrix and principal component 1. The checkerboard pattern of the Pearson correlation matrix strongly correlate with the positive / negative values of the first principal component.
For source and license information, please refer to the List of Figures.

Topological associated domains

The topological associated domains have a very clear identification pattern, they form regions of high interactions with a clear drop of interactions to genomic loci outside of a TAD (Figure 2.14). The approach used by HiCExplorer is introduced by Ramírez *et al.* [115] and identifies the amount of reads per position via a z-score based approach. First, the mean and standard deviation for each distribution given by the genomic distance $d = |i - j|$ is computed:

$$\mu_d = \frac{1}{n}(\sum_1^n m_{i,j}), \text{if } |i-j| = d \tag{3.12}$$

$$\sigma_d = \sqrt{\frac{1}{n}(\sum_1^n m_{i,j} - \mu_d), \text{if } |i-j| = d} \tag{3.13}$$

Second, the z-score matrix Z with the same dimensions as the interaction matrix M is calculated:

$$z_{i,j} = \frac{m_{i,j} - \mu_d}{\sigma_d} \tag{3.14}$$

For each bin l where $l = z_{i,j} \mid |i-j| = 0$ (i.e. bins on the main diagonal), the window size w is used to extract a submatrix Z_l from the z-score matrix Z:

$$Z_l = \begin{bmatrix} z_{i-w,j} & \dots & z_{i-w,j+w} \\ \vdots & \dots & \vdots \\ z_{i,j} & \dots & z_{i,j+w} \end{bmatrix} \tag{3.15}$$

The TAD separation score for a bin l is computed by the mean of the submatrix Z_l. TAD boundaries are given by local minima of the TAD separation score. To improve the statistical power, Ramírez *et al.* selected multiple window sizes to compute multiple TAD separation scores and used the mean value of the multiple separation scores to detect the minima, see Figure 3.7.

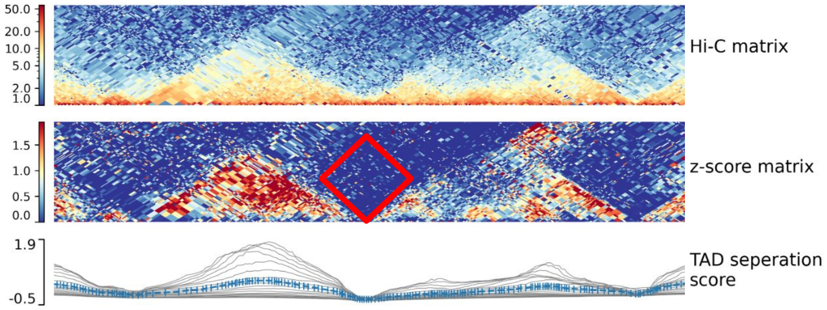


Figure 3.7.: The TAD separation score is computed by transforming the Hi-C contact matrix per genomic distance to a z-score matrix. For each bin l on the main diagonal, a submatrix Z_l is extracted from the z-score matrix (red diamond). The TAD separation score for this bin is computed by the mean value of the extracted z-score submatrix. Multiple window sizes for the submatrices are used to increase the statistical power; therefore, multiple lines are plotted in the TAD separation score track. The blue line is the mean value of all scores. At TAD boundaries, the TAD separation score reaches a minimum because the number of interactions is lower than expected (blue colors) according to the z-score. For source and license information, please refer to the List of Figures.

The approach is from a mathematical perspective problematic, as the z-scores are computed per genomic distance, and for each genomic locus on the main diagonal, the z-scores from the different distances are averaged. First, the assumption of a z-score distribution per genomic distance implies a normal distribution, which is incorrect. The density plots of the interaction value occurrence per genomic distance from multiple resolution matrices indicate a non-normal distribution, compare to Figure 3.8. Second, by averaging z-score values from multiple distributions, data points from a different origin are implicitly compared. However, other approaches work similarly; for comparison and benchmarking methods, refer to [123].

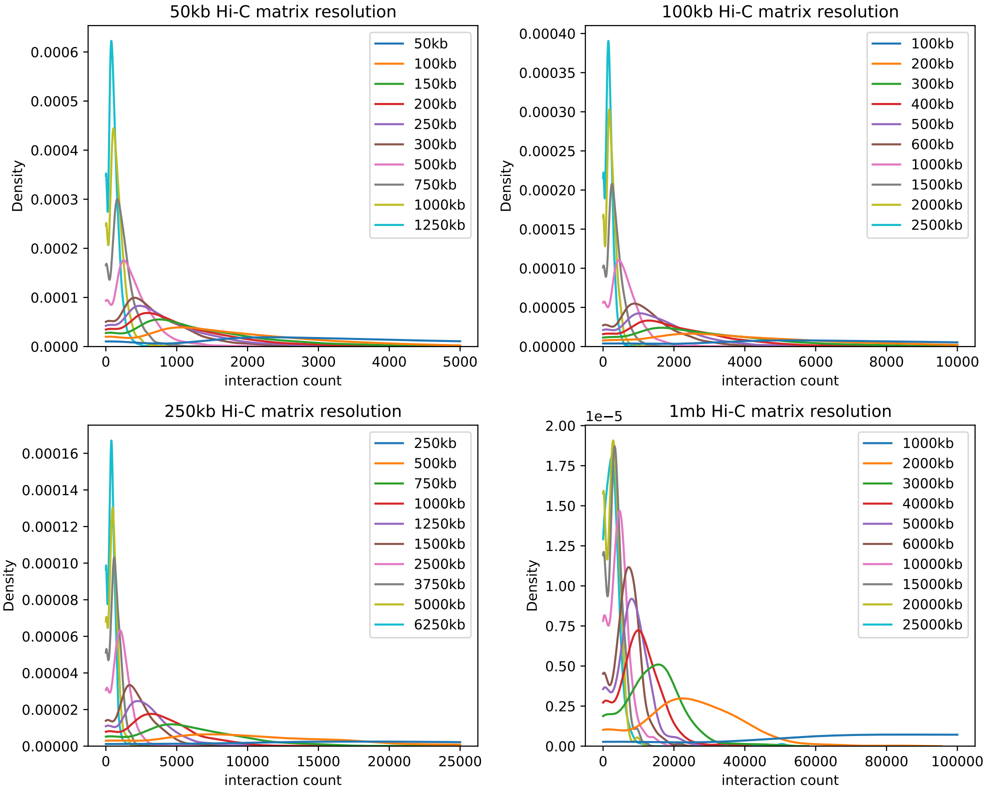


Figure 3.8.: Distribution of interaction values for different genomic distances on differing Hi-C interaction matrix resolutions. Data: Rao *et al.* [14], GM12878, chromosome 1. For source and license information, please refer to the List of Figures.

To overcome the TAD algorithm's mathematical issues, machine learning based approaches were implemented with the assistance of a student (Albert Lidel[9]) as part of a master's project. TAD boundary detection is basically a classification problem, in which the presence or absence of a boundary in an interaction submatrix is determined. Boundaries are easy to detect at first sight: triangular regions with high interaction counts are located on either side of a boundary. Between the TADs, a significantly lower number of interactions is present. However, as can be seen in Figure 3.10, the decision is not always that simple. Larger structures and less clearly defined structures exist, and TAD substructures given by nested TADs or a hierarchical order of TADs are

[9] http://www.bioinf.uni-freiburg.de/Lehre/Theses/P_Albert_Lidel_Report_Project.pdf

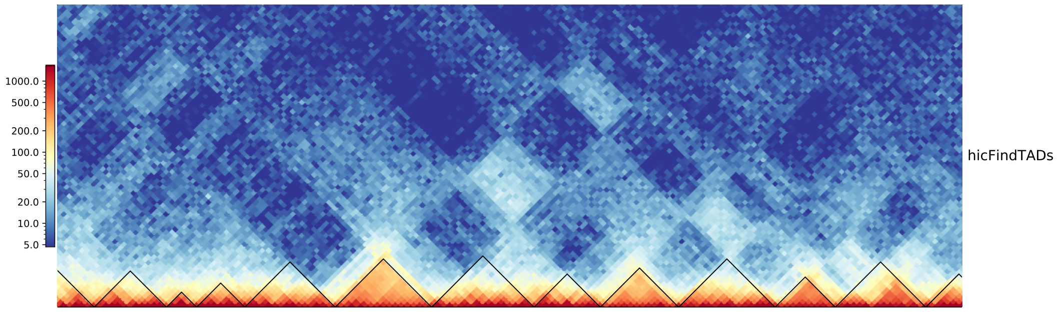


Figure 3.9.: Detected TADs by *hicFindTADs* on GM12878 chromosome 1 25 - 40 Mb, 100 kb resolution from Rao *et al.* [14].
For source and license information, please refer to the List of Figures.

also present. For this reason, the class of ensemble learning [124] algorithms seems the best choice. Ensemble learning works by using multiple classifiers and a majority vote for the final classification decision. The ensemble learning techniques Random Forest, boosting with AdaBoost [125, 126], bagging with Decision Trees, or Support Vector Classifiers have been tested for their ability to reproduce the detected TAD boundaries of *hicFindTADs* as a proof-of-concept. However, none of them showed a good performance. Promising results have been achieved with an Easy Ensemble Classifier [127], a combination of bagging and boosting in the form of a Bagging Classifier [128], using an internal AdaBoost Classifier. This method also corrects the imbalanced data by random undersampling. The correction is necessary because the proportion of the extracted submatrices containing TAD boundaries is low. To train the classifiers, detected TAD boundaries by existing TAD classification algorithms had been used. As input, the interaction matrix was split around the main diagonal into areas of 2 megabase pairs, and each submatrix was classified to contain a TAD boundary or not. Each boundary was correlated with CTCF, such that only boundaries with a CTCF peak are accepted to correct for false detection. The classifier which was trained on the Rao *et al.* GM12878 Hi-C matrix [14] achieved a lower correlation of the TAD boundaries with CTCF, 52% vs 62%, compared to Ramírez. However, the machine learning-based approach detects more TAD boundaries, 1199 compared to 659, and has, therefore, a higher absolute number of CTCF correlated boundaries. The replication value of the results of other TAD callers *hicFindTADs* (91.7%), *ClusterTAD* [129] (86.7%), and *rGMAP* [130] (95.8%) is

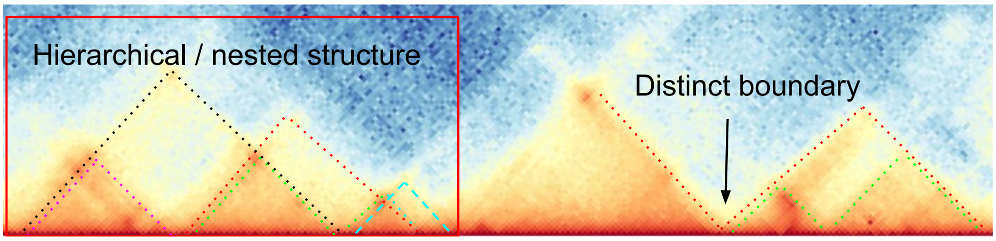


Figure 3.10.: TAD boundaries can be very clear in their shape (right structure) but also be nested and or embedded in a hierarchy.
For source and license information, please refer to the List of Figures.

high. This shows the power of the machine learning-based approach. The properties of an algorithm can be replicated only by their results. The major issue in validating TAD boundaries is that no ground truth for TAD boundaries is available. The locations of CTCF are correlated to TAD boundaries [131]; however, CTCF is present at more locations than TAD boundaries. For example, it is also associated with loop anchor points [14] or is involved in DNA repair processes [132]. The presence of CTCF therefore does not necessarily indicate a TAD boundary, nor must a boundary have CTCF present. Additional to this issue are the structures of TADs. While the triangular shape is relatively simple to detect, super- and substructures are also present, making a precise border detection difficult. In this context, results from population Hi-C are in contradiction to single-cell Hi-C results. Population-based Hi-C shows static structures between cell types and even between different species, while single-cell Hi-C shows a highly dynamic behavior of TADs between cells of the same cell type [133]. Taking these arguments into consideration, it is not very easy to validate the results of a TAD calling algorithm. The detected structures must be carefully interpreted to discern their biological meaning. Nevertheless, the detection of TADs is an essential task in a Hi-C analysis, especially because not all their functions and origins have been understood [100].

Another issue in detecting TADs is the sensitivity to the read coverage and resolution of the matrix. Detecting TADs on multiple resolutions detects different boundaries and, by this, possible nested TAD structures. The supervised bachelor thesis by Sarah Domogella[10] considers TADs of different resolutions to identify a hierarchy of the TADs. TADs of multiple resolutions are clustered based on their location, and a parent-child relationship is given if the TAD of the higher resolution is embedded to the area of the TAD of the lower resolution.

TADs are part of the euchromatin and are genomic loci with a high correlation of active gene expressions. This is caused by the close spatial proximity of the DNA, and therefore, regulatory elements like enhancers and promoters can be in contact. In this context, it is of scientific interest if the pattern of TADs changes between two samples, e.g., between a wildtype and a knock-out sample. The read coverage of the two samples must be normalized to an equal amount; it might be the case that inter-chromosomal contacts must be removed if they are present on an unusual amount. They might be biasing the normalization and, therefore, the differential testing. To differentially test the TAD regions, the TADs must be first computed on one sample, the *target matrix* (Figure 3.11 top). Based on it, the left inter-TAD region (Figure 3.11 bottom beige area), the TAD region itself (or intra-TAD) (Figure 3.11 bottom red area) and the right inter-TAD region (Figure 3.11 bottom blue area) is cut out on both samples and each is tested individually with the Wilcoxon-rank sum test [134] under the null hypothesis that they are equal. The user is able to specify if only the intra-TAD or a combination of left inter- and

[10] http://www.bioinf.uni-freiburg.de/Lehre/Theses/BA_Sarah_Domogalla.pdf

intra-TAD, right inter- and intra-TAD, or all three must be rejected to consider the TAD as differentially expressed.

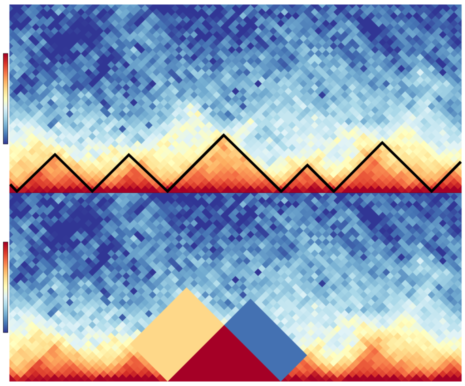


Figure 3.11.: Differential TAD test scheme. **Top:** A segment of a Hi-C interaction matrix rotated by 45 degrees. The black line indicates the detected TADs by *hicFindTADs*. **Bottom:** The same Hi-C interaction matrix segment. The red triangle is the detected TAD, named the intra-TAD region. Left of it, the left inter-TAD region. Visualized by a beige rectangle. Right of the TAD, the right inter-TAD region, visualized by the blue rectangle. Per TAD, all three regions can be used to test a wildtype sample for a differential interaction expression in comparison to a treatment sample. For source and license information, please refer to the List of Figures.

Loops

The detection of chromatin loops is a feature to detect possible enhancer-promoter interactions, gene loops, architectural modeling loops, and polycomb-mediated loops [14, 101], see also Figure 2.14a. The loop detection algorithm developed in this thesis is based on the algorithm HiCCUPS by Rao *et al.* [14]. Chromatin loop detection algorithms usually work in two phases. First, chromatin loops are regions in the interaction matrix with significantly higher interactions than the surrounding regions. These areas need to be detected. Second, several of these identified regions need to be pooled to define one enriched interaction. This general approach is for example used by HiCCUPS, Fit-Hi-C [135] or Peakachu [136]. For the first step, HiCCUPS uses the Poisson distribution. Althought this fits the data, see Figure 3.12, the approach by HiCCUPS is problematic. The standard in Hi-C is a KR or ICE corrected matrix, in which the discrete values of the raw Hi-C data are converted to continuous values. As the Poisson distribution is a discrete probability distribution, HiCCUPS reverts the correction factors to retrieve the original raw, and therefore discrete, values. Transforming the data to match the assumptions of a particular distribution is a poor approach; instead, a more appropriate distribution should be selected. Moreover, the Poisson distribution defines the mean to be the same as the variance, which can lead to overdispersion. An overdispersion test [137] on the raw Hi-C data shows overdispersion for 80% of the genomic distances, see Figure 3.13.

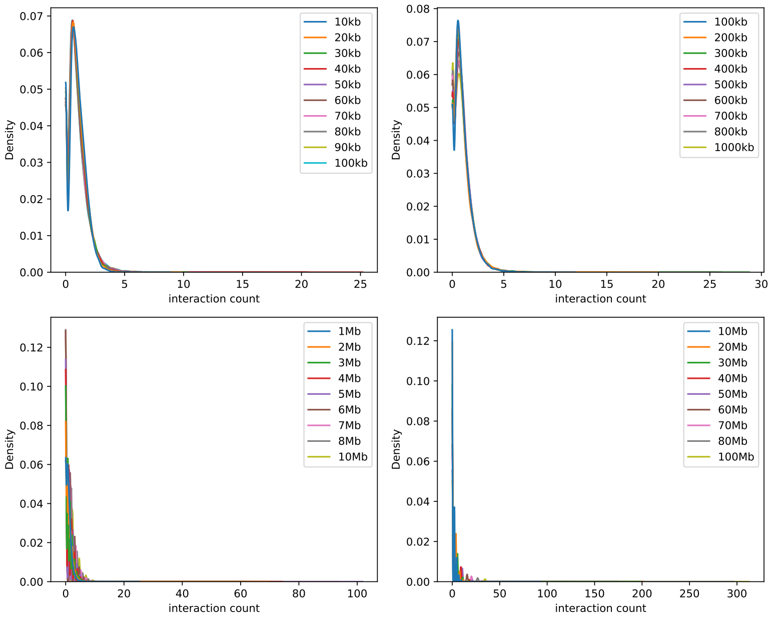


Figure 3.12.: Various genomic distance distributions for loop detection on Rao *et al.* [14] GM12878 10 kb Hi-C interaction matrix, chromosome 1. **Top left:** 10 - 100 kb, interval size 10 kb. **Top right:** 100 - 1000 kb, interval size 100 kb. **Bottom left:** 1 - 10 Mb, interval size 1 Mb. **Bottom right:** 10 - 100 Mb, interval size 10 Mb. For source and license information, please refer to the List of Figures.

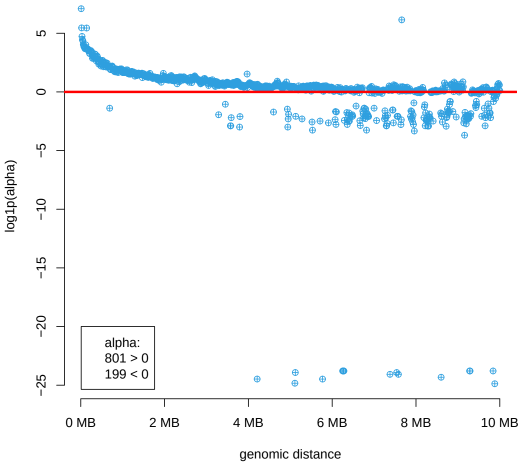


Figure 3.13.: Overdispersion test from Cameron & Trivedi 1990 [137]. Tested on the raw data of chromosome 1 of GM12878 cells, 10 kb resolution. The majority of the distances (80.1%) has an overdispersion. For source and license information, please refer to the List of Figures.

The overdispersion of the Poisson distribution is solved by the negative binomial distribution, which defines the mean and variance using two separate parameters. While a negative binomial distribution is also discrete, gamma functions can exchange the factorial operations in the binomial part of the negative binomial distribution to make the distribution continuous. The use of a *continuous negative binomial* distribution is adapted from edgeR, a software package for differential expression analysis [138, 139].

The probability mass function of the negative binomial distribution $\forall k \in \mathbb{N}$ and $\forall r \in \mathbb{N}$:

$$f(k, r, p) = \binom{k + r - 1}{k} p^k (1 - p)^r \tag{3.16}$$

The restriction to natural numbers comes from the binomial coefficient:

$$\binom{k + r - 1}{k} = \frac{(k + r - 1)!}{(k!) * (k + r - 1 - k)!} = \frac{(k + r - 1)!}{(k!) * (r - 1)!} \tag{3.17}$$

However, the gamma function is defined for any $n \in \mathbb{N}$:

$$\Gamma(n) = (n - 1)! \tag{3.18}$$

Moreover, the gamma function is defined for any $n \in \mathbb{R}_{>0}$

$$\Gamma(n) = \int_0^\infty x^{n-1} * e^{-x} dx \tag{3.19}$$

With Equation 3.19 the binomial coefficient can be reformulated as:

$$\binom{k + r - 1}{k} = \frac{\Gamma(k + r - 1 + 1)}{\Gamma(k + 1) * \Gamma(k + r - 1 - k + 1)} = \frac{\Gamma(k + r)}{\Gamma(k + 1) * \Gamma(r)} \tag{3.20}$$

Which leads to the probability mass function for a *continuous negative binomial* distribution with $\forall k \in \mathbb{R}_{>0}$ and $\forall r \in \mathbb{R}_{>0}$:

$$f(k, r, p) = \frac{\Gamma(k + r)}{\Gamma(k + 1) * \Gamma(r)} p^k (1 - p)^r \tag{3.21}$$

A *continuous negative binomial* distribution is fitted to detect higher than expected interactions per genomic distance d.

The p-value of an interaction i at the genomic distance d is given by the cumulative density function (CDF):

$$pvalue\ of\ i = \begin{cases} 1 - CDF_d(i) & \text{if } i > 0. \\ 1 & \text{if i = 0.} \end{cases} \tag{3.22}$$

Each pixel gets a p-value assigned and independently per distribution, a p-value threshold is used to detect the outliers. These outliers are interpreted as loop candidates. However, a loop is not a single enrichment point but is usually represented in the two-dimensional Hi-C matrix by accumulating outliers. The selected candidate needs to be considered relative to its local background, and also in one loop neighborhood, only one interaction can be the loop candidate. The interaction in a neighborhood with the highest observed/expected score is the loop candidate; the candidates' p-values are not considered. As a final step to identify a loop, the loop region is tested against the neighborhood with the doughnut approach similar to the HiCCUPS algorithm [14], see Figure 3.14.

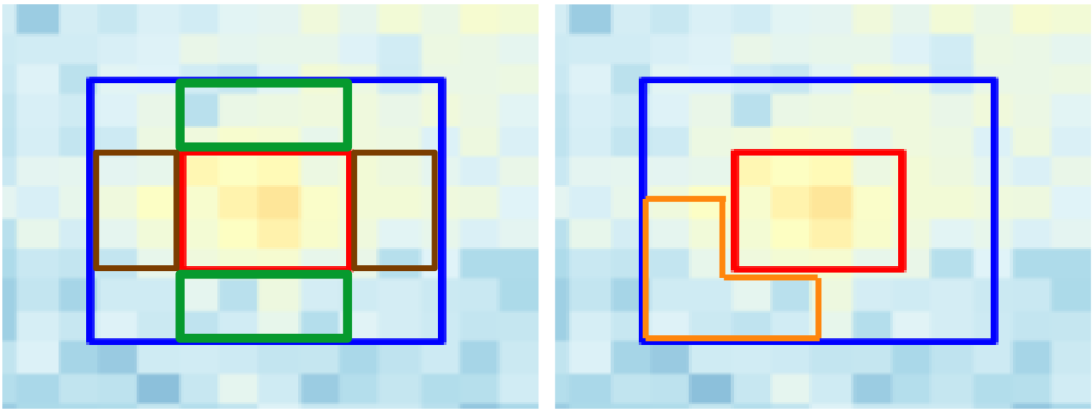


Figure 3.14.: Doughnut approach similar to Rao *et al.* [14] to test the peak region (red area) individually against the horizontal (green area), the vertical (brown area), and lower left corner neigborhood (orange area) with the Wilcoxon rank-sum test under H0 the distributions are equal. A loop is only accepted if all tests are rejected.
For source and license information, please refer to the List of Figures.

A comparison of different loop detection algorithms shows a great variation in the detected locations. The first approach to measure a loop detection algorithms' quality is to compute the overlap with existing algorithms. The algorithm presented here has been compared with HiCCUPS, cooltools[11], chromosight [140], Homer [25], Fit-Hi-C 2 [135], and Peakachu [136]. The comparison is computed on the GM12878 cell line Hi-C data from Rao *et al.* [14]. HiCExplorer and HiCCUPS share around 40% of the detected loops. Cooltools reimplementation of the HiCCUPS' algorithm also shares a similar level with HiCExplorer, see Figure 3.15a. The overlap of detected loops between HiCExplorer and HiCCUPS and the other algorithms is similar, but at a low level. The only exception is chromosight, Figure 3.15b, but this high overlap is caused purely by the large number of loop locations (six times higher) found by chromosight. A second approach to measure the quality of an algorithm is comparing the ratio of detected elements to a ground truth. Nevertheless, ground truth for loops is impossible to create because they are dynamic structures representing, e.g., enhancer-promoter contacts that vanish as soon as a gene is

[11] https://github.com/open2c/cooltools

no longer expressed. For this reason, a different method is used. Rao *et. al.* [14] showed a high agreement of loop anchor points locations with the position of CTCF. Moreover, CTCF and cohesin are the protein complexes involved in loop extrusion [14, 101, 141]. It should be noted that not every loop is bound by CTCF and cohesin, and not every location of CTCF or cohesin is related to the formation of a loop. Nonetheless, lacking a better method, the quality of a loop detection algorithm is measured by the ratio of CTCF or cohesin present at detected loop locations and all detected loop locations. Using this measurement, the detected loop locations of HiCExplorer occur at 64% of CTCF locations and for 25% at locations of the cohesin subcomplex RAD21. HiCCUPS has a slightly lower (61% and 22%) occurrence, but a marginally better overlap to locations of active enhancers-promoter marks by H3K27ac [142] (86% vs. 92%), and the cohesin subcomplex SMC1 (91% vs. 96%), see Table 3.2. The other algorithms, except the reimplementation of HiCCUPS by cooltools, have significantly less overlap with their detected loop locations for all investigated proteins.

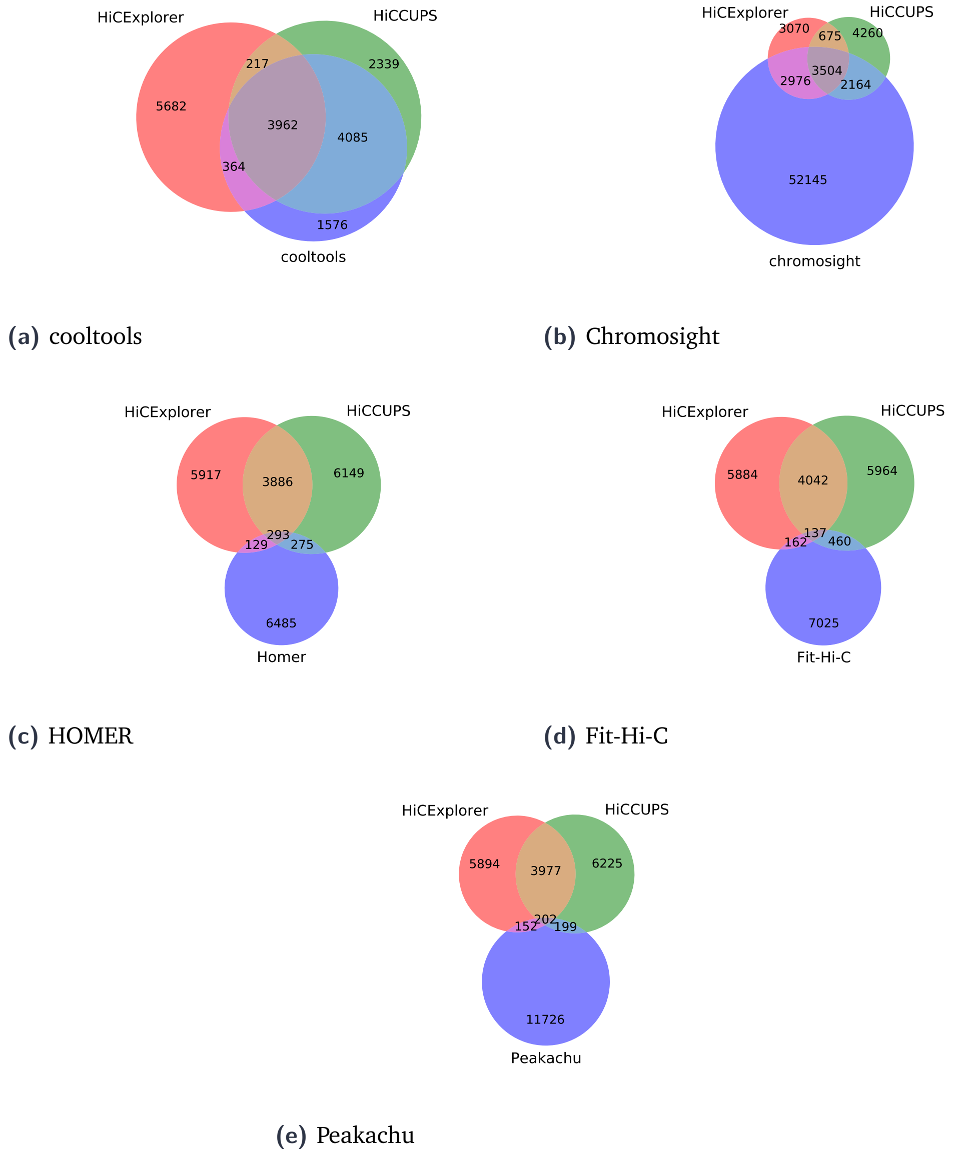
Algorithm	Detected loops
HiCExplorer	10225
HiCCUPS	10603
cooltools	9987
chromosight	60789
Homer	7182
Fit-Hi-C 2	7784
Peakachu	12279

Table 3.1.: The number of detected loops on GM12878 cell line on 10 kb resolution and 8 Mb genomic distance restriction.

Data	CTCF ChIA-PET	H3K27ac HiChIP	RAD21 ChIA-PET	SMC1 HiChIP
HiCExplorer	6540 (0.64)	8835 (0.86)	2577 (0.25)	9346 (0.91)
HiCCUPS	6564 (0.61)	9831 (0.92)	2385 (0.22)	10179 (0.96)
cooltools	5467 (0.54)	8857 (0.88)	1781 (0.17)	9396 (0.94)
chromosight	7205 (0.11)	41599 (0.68)	1785 (0.02)	47056 (0.77)
Homer	1349 (0.18)	5368 (0.74)	286 (0.03)	6470 (0.90)
FitHi-C 2	163 (0.02)	2279 (0.29)	109 (0.01)	2656 (0.34)
Peakachu	686 (0.05)	4873 (0.39)	78 (0.006)	6150 (0.50)

Table 3.2.: Intersection of detected loops of the GM12878 cell line on 10 kb resolution and 8 Mb genomic distance restriction with various HiChIP and ChiA-PET locations: CTCF ChIA-PET (GSM1872886); H3K27ac HiChIP (GSE101498), SMC1 HiChIP (GSE80820), and RAD21 ChIA-PET (GSM1436265) data.

The detection can be computed on multiple resolutions of the same data and be merged with HiCExplorer's *hicMergeLoops* to increase the number of detected loops. The tool *hicValidateLocations* is provided to validate the detected loop locations with protein peak data.



(a) cooltools

(b) Chromosight

(c) HOMER

(d) Fit-Hi-C

(e) Peakachu

Figure 3.15.: Intersection of detected loops of HiCExplorer, HiCCUPS and either cooltools, chromosight, HOMER, Fit-Hi-C or Peakachu. HiCExplorer, HiCCUPS and cooltools have the highest relative intersection, while chromosight has the highest absolute number of shared loops with HiCExplorer, due to the fact it detects six times more interactions than the other methods. Homer, Fit-Hi-C and Peakachu have only a minor intersection.
For source and license information, please refer to the List of Figures.

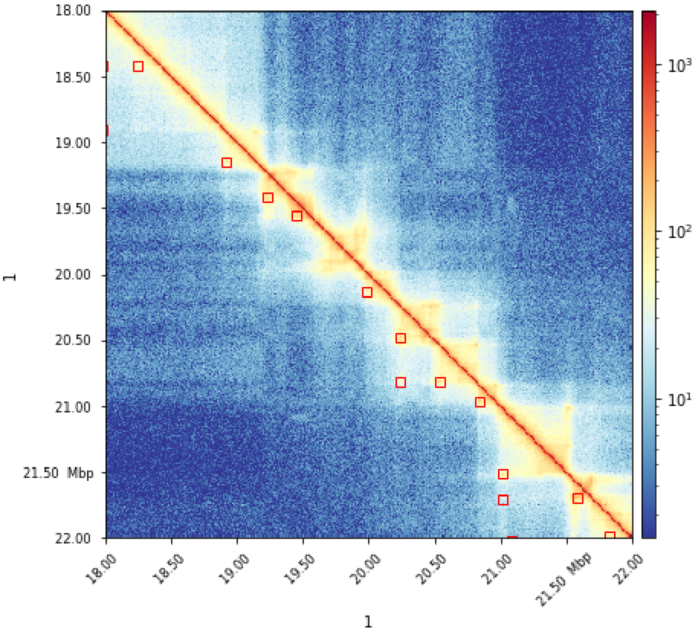
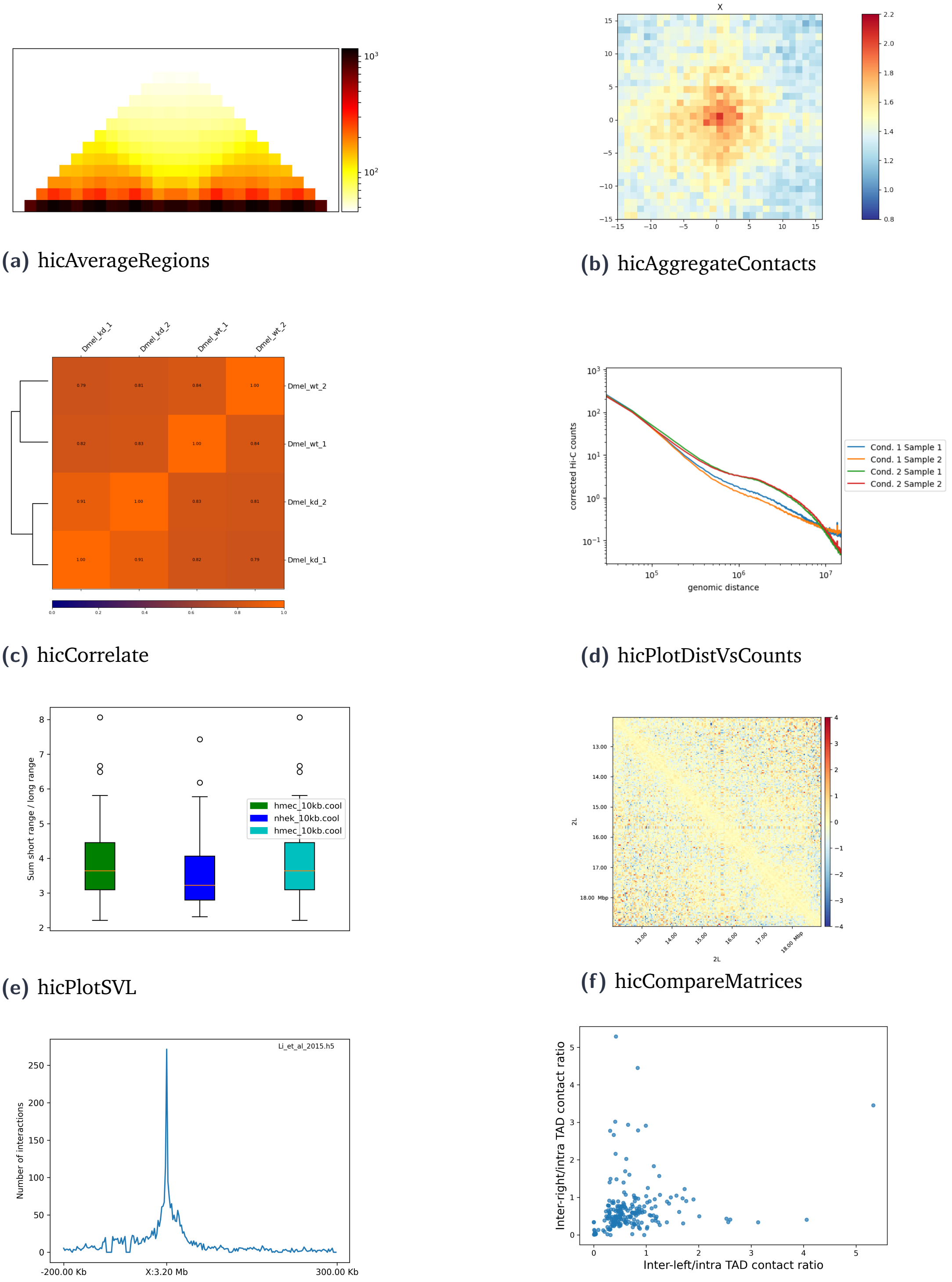


Figure 3.16.: Detected loops on chr1 18-22 Mb on Rao *et al.* [14] GM12878 data. The loops are highlighted by the red squares. Plotted with *hicPlotMatrix* and *–loop* parameter. For source and license information, please refer to the List of Figures.

Other analysis methods

Besides the analysis methods to detect chromatin's three major structures in Hi-C data, HiCExplorer supports various additional methods. *hicAverageRegions* and *hicPlotAverageRegions* visualize the averaged contacts of a list of reference points in a given range. This gives insights into the global TAD structure of a Hi-C sample and can indicate differences between wildtype and treatment samples (Figure 3.17a). *hicAggregateContacts* aggregates contacts of regions of interest, for example, of protein binding sites and enables a global view of overall regions if a higher number of contacts in these regions is observed (Figure 3.17b). However, *hicAverageRegions* and *hicAggregateContacts* operate on a global view, and individual changes or differential behaviors of single regions might vanish in the global perspective. *hicCorrelate* correlates two or more Hi-C interaction matrices of the same genomes and sizes with the Pearson or Spearman correlation. This allows the computation of matrices' global relations, which provides a useful quality control method; biological and technical replicas have high correlations. Moreover, the correlation results of *hicCorrelate* are used as input for a hierarchical clustering method to reveal the relationship of different samples. This gives insights on how similar or distant cell types are (Figure 3.17c). The tool *hicPlotDistVsCounts* displays the relation of interaction numbers per genomic distance. Similar samples like replicates should behave the same, and structural changes are visible by differing interaction counts. The tool can operate globally or on a local one if a specific region is investigated (Figure 3.17d). The relationship of the number of short-range contacts to longe range contacts can be visualized with *hicPlotSVL*. Per chromosome, the ratio is computed, and for one sample, a boxplot is used to visualize the ratios (Figure 3.17e). *hicCompareMatrices* analyzes two interaction matrix per pixel, and provides a global and local comparison simultaneously. Either the difference, the ratio, or log2ratio per pixel can be used as an analysis method (Figure 3.17f). *hicViewpoint* extracts the reads at a given reference point within a user-defined range and creates virtual 4C data. This option can replace 4C or cHi-C experiments if the read coverage and Hi-C resolution are sufficient (Figure 3.17g). Last, *hicInterIntraTAD* creates a scatter plot between the ratio of left inter-TADs vs. intra-TAD contacts and the right inter-TAD vs. intra-TAD contacts. This plot helps to investigate the global contact pattern of a matrix (Figure 3.17h).



(a) hicAverageRegions

(b) hicAggregateContacts

(c) hicCorrelate

(d) hicPlotDistVsCounts

(e) hicPlotSVL

(f) hicCompareMatrices

(g) hicPlotViewpoint

(h) hicInterIntraTAD

Figure 3.17.: Different Hi-C data analysis methods supported by HiCExplorer. **(a)** hicAverageRegions: Mean value computation for reference points to detect global changes in the chromatin structure close to the main diagonal. For each reference point a submatrix is extracted, and all submatrices are used to compute a mean signal of the regions. The reference points have one dimensional coordinates. **(b)** hicAggregateContacts: Aggregation of reference points to detect global events of e.g. enhancer-promoter interactions. The reference points have two dimensional coordinates. **(c)** hicCorrelate: Pearson or Spearman correlation of multiple Hi-C matrices with an additional hierarchical clustering based on the correlation values. **(d)** hicPlotDistVsCounts: Global genomic distances vs. the number of interactions at these distances. This is used to compare multiple Hi-C matrices by their global interaction pattern. **(e)** hicPlotSVL: Box plots of the ratio of short/long distance contacts per chromosomes for multiple Hi-C matrices. **(f)** hicCompareMatrices: Direct comparison of the values of two Hi-C matrices with difference or log2 ratio. **(g)** hicPlotViewpoint: Extraction of a virtual 4C with a given reference point and a certain up- and downstream region. This shows all interactions of the reference points with the chromatin regions up- and downstream of it. **(h)** hicInterIntraTAD: A scatter plot between the ratio of left inter-TAD contacts vs. intra-TAD contacts on the x-axis and the ration of right inter-TAD contacts vs. intra-TAD contacts on the y-axis.

For source and license information, please refer to the List of Figures.

3.1.3 Visualization

The visualization of research results is critical in understanding the data better and presenting results. HiCExplorer provides several visualization tools that are partially already covered in the 'Analysis' section. The two main visualization tools of HiCExplorer are the integrated *hicPlotMatrix* to plot a Hi-C interaction matrix or only subareas. It can highlight detected loops and TADs, and adds additional data tracks in the bigwig or bedgraph format. These are usually A/B compartment tracks, RNA-seq or ChIP-Seq data files. The second tool *'hicPlotTADs'* which has been outsourced from HiCExplorer, and is independently developed from HiCExplorer as *'pyGenomeTracks'* [31]. With *pyGenomeTracks*, several data sources like Hi-C, ChIP-Seq, RNA-Seq, or gene annotations can be visualized for a specific region to present the results in a compact and correlating way. See Figure 3.18 as an example plot of *pyGenomeTracks*. The benefit in a combined plot of different data types is to get an intuitive understanding of the data. For example, in Figure 3.18 the CP190 data (pink) is overlayed to the Hi-C track and shows the peaks are present mainly at the boundaries of the TADs. Also the chromatin states show a correlation to the TADs. Closed chromatin is present for regions without a clear TAD pattern (8,250 - 8,300 kb) and open chromatin where TADs are present. With a visualization approach as it is provided by *pyGenomeTracks*, researchers can get a fast insight, apply afterwards more complex analysis methods and use the visualizations for a good presentation of their data.

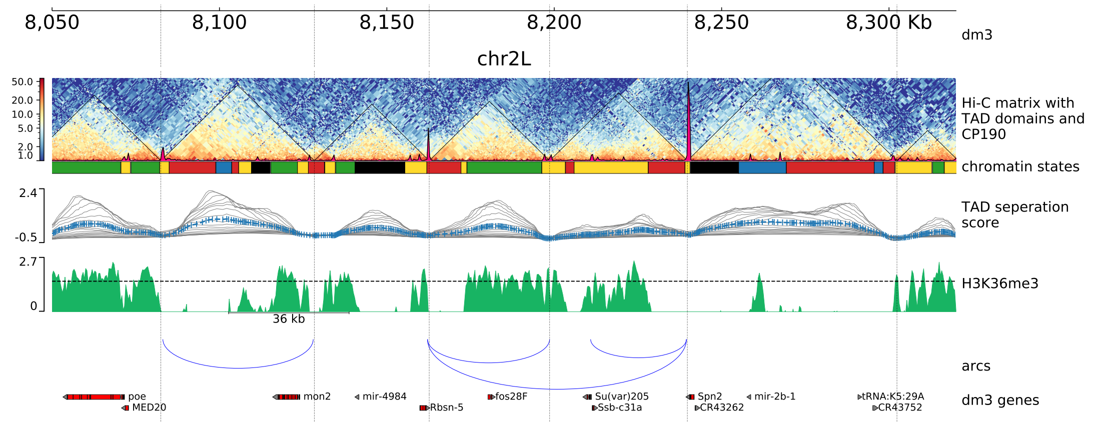


Figure 3.18.: An example plot created by pyGenomeTracks. It shows the genomic locus of chromosome 2L of Kc167 cell line, 8.05 - 8.31 Mb. It contains the Hi-C data with detected TADs (black lines) and the coverage of profile of CP190 (pink). The peaks of CP190 at the TAD boundaries indicate a connection of CP190 with the formation of TADs in drosophila. This is followed by a chromatin state track and the TAD separation scores. The TAD separation score is computed for different window sizes, resulting in multiple scores (grey lines). The blue line is the mean value of the TAD separation scores, and at its local minimum, a TAD boundary is detected. For details on the TAD separation score, see section 3.1.2. The green data track shows an example of H3K36me3 histone marks, and its correlating to the TADs in the open chromatin states. The blue arcs are artificial example arcs of potential CP190 peak interactions. The last track is a gene track showing two rows of genes from drosophila melanogaster reference genome dm3. For source and license information, please refer to the List of Figures.

3.2 Capture Hi-C data analysis

The following chapter describes the analysis workflow for capture Hi-C data analysis. It is implemented in the capture Hi-C modules of HiCExplorer and described in Wolff *et al.* 'Galaxy HiCExplorer 3: a web server for reproducible Hi-C, capture Hi-C and single-cell Hi-C data analysis, quality control and visualization', 2020 [28].

Capture Hi-C (cHi-C) and HiChIP provide location-specific interactions. They are limited to predefined target regions or target proteins and avoid the genome-wide ('all vs. all') interactions of Hi-C. Current Hi-C approaches reflect all genome-wide interactions, and therefore, the economic costs are high. For example, a read coverage of 20 - 40 million reads, common in one-dimensional approaches like RNA-Seq or ChIP-Seq, requires 400 - 1600 million reads for the two-dimensional Hi-C interaction matrix. However, the investigation of specific regions requires only the genome-wide interactions of these specific regions, not the genome-wide interactions of all locations. The reduced number of reads by a restriction to a minor number of sites of interest allows, especially considering the economical cost, a deeper sequencing and a higher read coverage for these selected regions.

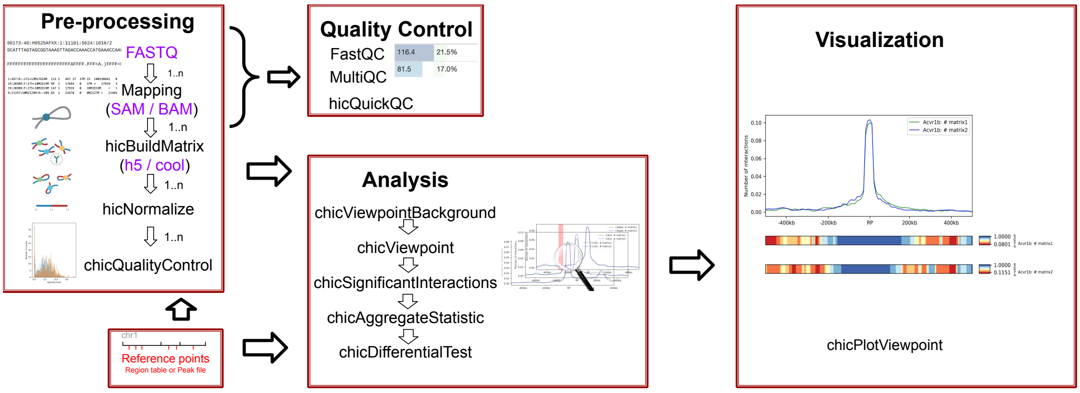


Figure 3.19.: Capture Hi-C workflow. **Pre-processing:** Required input are the raw FASTQ files, which undergo a quality control with FastQC. Mapping of the raw files with a mapping software of free choice, the tool *hicBuildMatrix* creates the interaction matrix, the central data structure. Afterwards the data can be normalized, and the first capture Hi-C specific quality control to detect too sparse viewpoints is recommended, as additional input a list of predefined reference points is required. **Analysis:** The analysis of capture Hi-C data requires the computation of a background model to decide if a given interaction is significant for its relative genomic distance, afterwards the viewpoints need to be extracted from the interaction matrix. *chicSignificantInteractions* computes significant interactions per matrix and can prepare these in combination with the succeeding *chicAggregateStatistics* for the differential analysis. Alternatively *chicAggregateStatistics* accepts a predefined list with regions of interest for the differential analysis with *chicDifferentialTest*. **Visualization:** The visualization plots one viewpoint of one or many samples in one plot, see Figure 3.23. The computed p-values of the significance detection or the detected differential areas can be highlighted.
For source and license information, please refer to the List of Figures.

3.2.1 Pre-processing

Preselected regions are usually promoter regions of the genes which are to be investigated. With capture Hi-C, the genome-wide interactions with these preselected regions can be retrieved. While the interactions can be stored in a regular Hi-C matrix, the pre-processing, analysis, and visualization differ for capture Hi-C. The preselected regions are called the *reference points* (the peak in the middle labeled as RP, Figure 3.20), the up- and downstream region including the specific reference point is the *viewpoint*. A viewpoint provides all genome-wide interactions of a reference point with all other locations. However, the number of interactions is very low for longer distances. For this reason, a viewpoint is usually restricted up- and downstream to a certain distance. For example the distances in Figure 3.20 is restricted to +/- 200 kb from the reference point. The up- and downstream regions are defined in relative distances to their specific reference point and not in absolute genomic positions. This approach is used to create background distributions for each relative distance using all viewpoints of a capture Hi-C dataset. The significance of an interaction at its relative distance to the reference point can be computed using the background distributions.

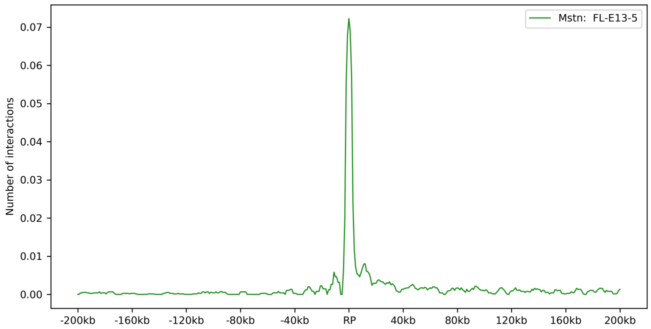


Figure 3.20.: A capture Hi-C viewpoint. This example shows the promoter region of the mouse gene Mstn, located at chromosome 1 at position 53.1 Mb (mm9 reference genome). The up- and downstream interactions of the reference point (RP) are recorded. The up- and downstream regions are defined in relative distances to the reference point. The units of the interactions in the plot are the relative interactions computed by $interaction_i / \sum_{j=0}^{n} interaction_j$. Data from Andrey *et al.* [143]. For source and license information, please refer to the List of Figures.

The pre-processing workflow of capture Hi-C and HiChIP data is similar to Hi-C. First, the raw FASTQ reads have to be quality controlled by FastQC, and if necessary, existing adapters need to be removed. Second, the chimeric reads are mapped to a reference genome, and the mapped data is used to create an interaction matrix. The quality of the chimeric reads can be controlled with the quality report created by *hicBuildMatrix*, or multiple reports be summarized with *MultiQC*. To quality control the pre-defined reference points' interactions, the tool *chicQualityControl* is used, see Figure 3.21. Each viewpoint is considered independently; viewpoints that contain no or too few interactions are removed. However, the threshold is user-defined with a sparsity level. These removed viewpoints can be caused by no interactions of the reference point with the DNA in the viewpoint, that this region contains no restriction cut site or that the digestion and the ligation failed in this region, see Figure 3.22

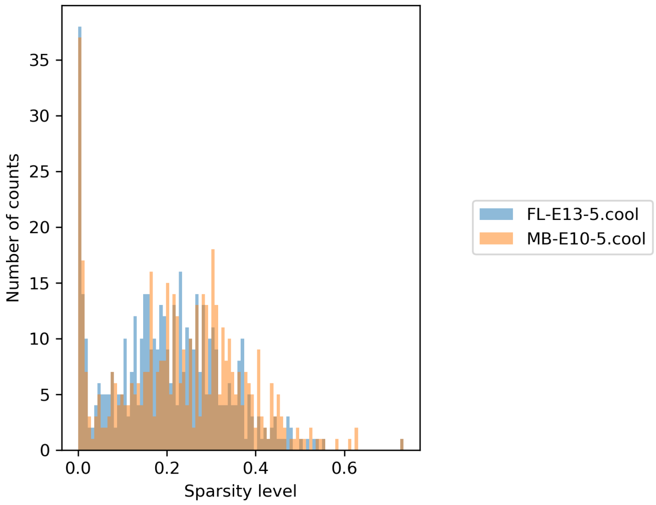


Figure 3.21.: Capture Hi-C quality control. The quality control measures the sparsity i.e. the number of detected positions with at least one interaction vs. all possible positions. If a region has no interactions, the sparsity will be 0. In the above sample more than 35 viewpoints have around 0 interactions and are therefore removed from the data. The user can define a sparsity threshold.
For source and license information, please refer to the List of Figures.

3.2.2 Analysis

The analysis of capture Hi-C is implemented in HiCExplorer by creating a viewpoint for all regions of interest. While a Hi-C interaction matrix is two-dimensional, the interactions of one specific location with all other locations are equal to a row in the interaction matrix and, therefore, a one-dimensional vector. The specific location, or reference point (RP), defines the neutral point in such a one-dimensional data structure. All interactions are indexed according to this position in relative distances. Upstream locations have a negative relative distance; downstream locations have a positive relative distance. For the analysis of the data, a background distribution is required. However, not just a single background distribution is used, but one distribution for each relative distance. This is implemented by extracting the data for each relative distance of all viewpoints, and an empirical continuous negative binomial distribution, as defined in section 3.1.2, is fitted. This approach is chosen to identify interactions of a reference point with a region that has more interactions than expected, considering all other interactions of all other reference points at this relative distance. The underlying idea is

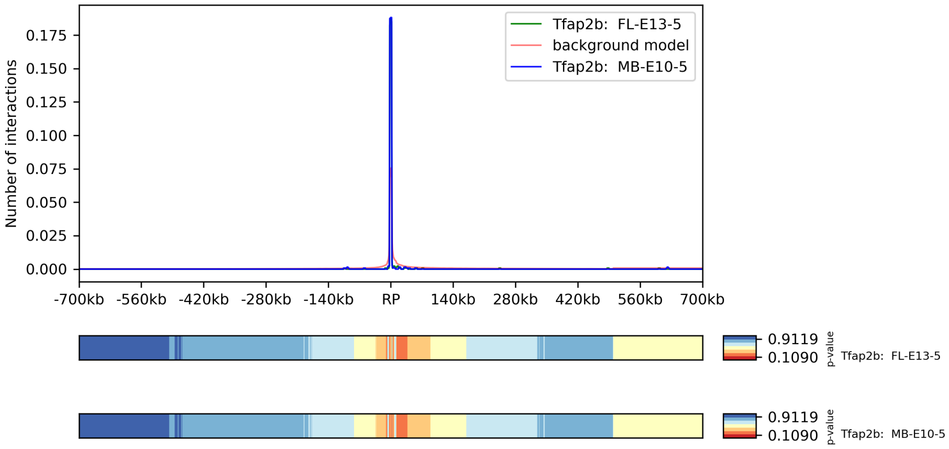


Figure 3.22.: A viewpoint with almost no interactions of the reference point (RP) with other genomic regions. Reasons for an occurrence of these viewpoints are a mistake in the data creation, e.g. a failed capture step or a failure in the ligation. It is also possible that the given reference point simply does not have any interactions within its viewpoint.
For source and license information, please refer to the List of Figures.

that the relative distance of interacting enhancers and promoters varies, and therefore an enriched interaction value at a specific distance can be interpreted as enhancer-promoter interactions. Moreover, if an high interaction count at a specific relative distance occurs regularly, it is unlikely to be an enhancer-promoter interaction. In addition to the empirical continuous negative binomial backgrounds, a simplistic approach using the average interaction value per relative distance is also offered.

Using *chicViewpoint* the viewpoint interaction data is extracted from the interaction matrix and a p-value is computed per interaction. The p-value of an interaction i at the relative distance rd is given by the cumulative density function of the empirical fitted continuous negative binomial distribution for the relative distance rd:

$$pvalue\ of\ i = \begin{cases} 1 - CDF_{rd}(i) & \text{if } i > 0. \\ 1 & \text{if i = 0.} \end{cases} \tag{3.23}$$

The extracted data is stored in a HDF5 container for fast access and user-friendly handling. In a third step, significant interactions using the p-values or the average background value combined with x-fold thresholds to detect significant interactions are computed; per relative distance, an individual threshold needs to be set. For the differential analysis, significant interactions can be used; however, a user can define a file with predefined locations. For example, these can be selected enhancer locations. The differential analysis uses a chi-squared test to test for differential interactions: the sum of interactions in a viewpoint and the interactions at the selected relative distance for two samples to be tested against each other.

3.2.3 Visualization

The viewpoints can be plotted with *chicPlotViewpoint*, for better visual comparability, the relative interactions are plotted. The significant interactions and differentially detected interactions can be highlighted; the p-values per interaction are visualized as a heatmap bar.

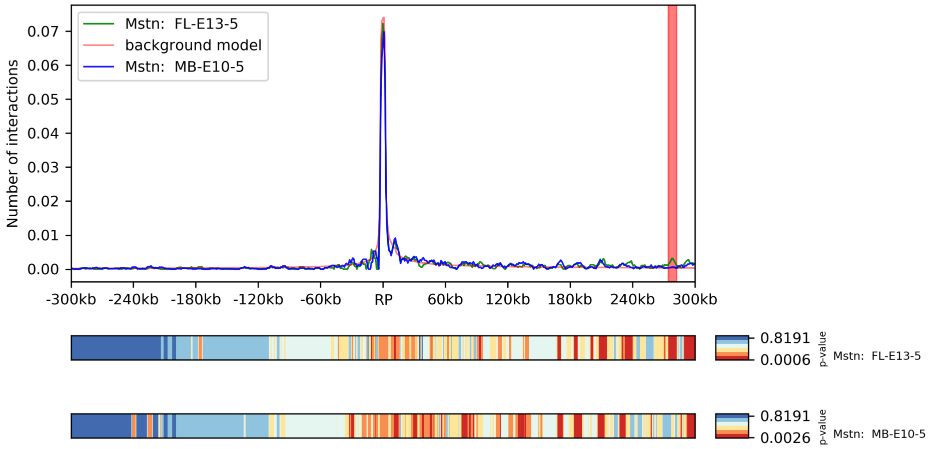


Figure 3.23.: An example of a differential interaction between two samples of the gene Mstn under two conditions, FL (Forelimb) and MB (Midbrain), and development stages (E13-5 and E10-5) in one location, highlighted in red. The heatmap under the plot are the computed p-values given the background per relative genomic distance. Data from Andrey *et al.* [143].
For source and license information, please refer to the List of Figures.

3.3 Single-cell Hi-C data analysis

Single-cell Hi-C data extends the Hi-C approach by providing the interaction data per individual cell and is, therefore, able to gain insight into the different chromatin structures of different cell types or the structure of different cell cycle phases. In contrast to Hi-C, the interaction matrix is generated individually per cell, while Hi-C provides a cumulative interaction matrix over potentially a million cells. The workflow to analyze single-cell Hi-C data is similar to Hi-C, but the need to deal with many different cells results in an increased complexity and requires the development of additional methods e.g. for cell clustering. The data analysis workflow, methods, algorithms, and file formats described here are presented in detail in the publications Wolff *et al.* 'Galaxy HiCExplorer 3: a web server for reproducible Hi-C, capture Hi-C and single-cell Hi-C data analysis, quality control and visualization', 2020 [28]; Wolff *et al.* 'Scool: a new data storage format for single-cell Hi-C data' [30], 2021; and Wolff *et al.* 'Robust and efficient single-cell Hi-C clustering with approximate k-nearest neighbor graphs', 2021 [29].

3.3.1 Pre-processing

The pre-processing of single-cell Hi-C data differs in comparison to Hi-C or capture Hi-C. The raw FASTQ data contains the reads of multiple cells and demultiplexing to retrieve the reads per cell needs to be applied. The encoding to associate a read with a cell differs from method to method [16, 85, 86, 87, 88, 89, 144]. A common approach is to use *barcodes* for this encoding. Barcodes are artificial nucleotides, these are attached to the start of a read, and the combination of these nucleotides is unique per cell. However, the way of encoding and storing the required information is performed differently. Demultiplexing needs to be applied for the majority of the single-cell Hi-C protocols independent of scHiCExplorer. The reason lies in the variety of possible encoding for barcodes and how the barcode information is stored. It is not possible to implement a general solution; scHiCExplorer provides only the demultiplexing of data from Nagano 2017 [85] because this study provided only raw data, while others provided interaction matrices. Depending on how the data is provided, it might be necessary to remove the barcodes and/or adapters from the reads after demultiplexing, and general quality control is required. Following the demultiplexing, each cell's FASTQ data needs to be mapped to the correct reference genome, and for each cell, the interaction matrix using HiCExplorer's *hicBuildMatrix* in the *cooler* file format is created. The amount of cells which can be processed is in the range from currently a few hundred (e.g., Flyamer *et al.* [86]) to multiple thousand to ten-thousands (e.g., Nagano *et al.* [85], or Ramani *et al.* [144]) cells, demanding high automatization and high computational resources. Both are offered via the Galaxy HiCExplorer. As an additional step to Hi-C, it

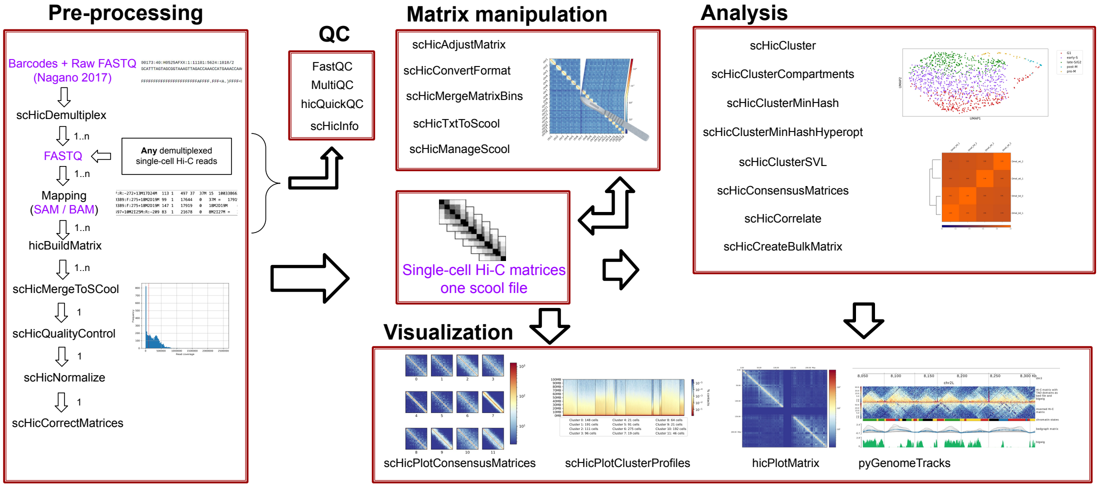


Figure 3.24.: **Pre-processing:** The pre-processing of single-cell Hi-C data requires demultiplexing of the reads. All reads are in one FASTQ file and are associated to one cell with a specific barcode. Next, for each cell the reads need to be mapped and an individual interaction matrix is created with *hicBuildMatrix*. With the tool *scHicMergeToScool* the *n* individual matrices are merged to one *scool* file [30], followed by quality control, normalization and ligation bias correction. **QC:** *FastQC* to control the quality of the raw reads, *MultiQC* to control in one overview the QC reports of *hicBuildMatrix* for all cells, *hicQuickQC* to get a first impression of the Hi-C read quality; *scHicInfo* to gain insights in the data stored in *scool* file. **Analysis:** To analyse single-cell Hi-C data, multiple cluster approaches with different dimensions techniques are provided: *scHicCluster*, *scHicClusterSVL*, *scHicClusterCompartments* and *scHicClusterMinHash*. *scHicCorrelate* computes a Pearson correlation per matrix; *scHicConsensusMatrix* computes the consensus matrix of given clusters and *scHicCreateBulkMatrix* to create a single matrix out of all matrices. **Visualization:** The single-cell data can be visualized as consensus matrices (*scHicPlotConsensusMatrices*), as a cluster profile plot (*scHicPlotClusterProfiles*). Individual interaction matrices of cells can be additionally visualized with *hicPlotMatrix* or *pyGenomeTracks*. **Matrix manipulations:** Methods to change the bin size or to remove certain areas of the matrices are given, also tools to import and export competing file formats.
For source and license information, please refer to the List of Figures.

is recommended to store the individual interaction matrices in a specialized file format. This has two advantages: firstly, improved structure and organization, and avoidance of human-introduced errors due to handling several thousand files; and secondly, reduction of storage space. The *scool* file format [30] is an extension of the *cooler* file format by Abdennur [116] and enables sharing of overlapping data structures of the individual cool files to save storage space. The file format structure is described in Figure 3.25. Moreover, the handling of the matrices can be taken over by software. This is less error-prone and computations are faster. Major data structures like the *bins* storing the genomic positions need to be loaded only once, and a native, analysis software supported parallelization is more optimal than user-created multiple calls via the command-line interface. Publications working with single-cell Hi-C data like Nagano *et al.* [16], Stevens *et al.* [87] or Ramani *et al.* [144] published their interaction matrices as text-based files, Gassler *et al.* [145] used individual cool file; competitive software like Zhou *et al.* [21] used text-based files too.

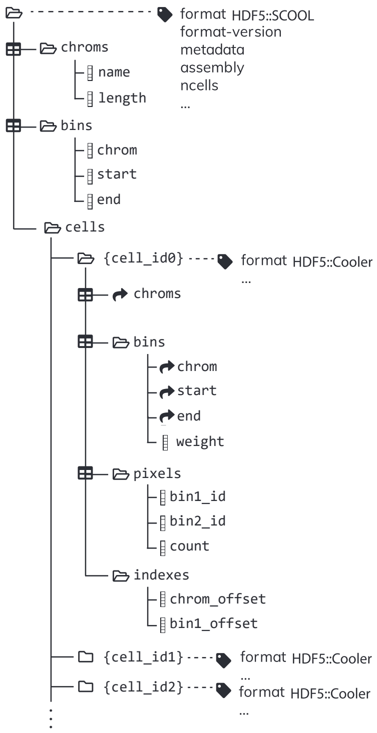


Figure 3.25.: Layout of the single-cell Hi-C data format *scool*. All cells share the chromosomes and their specific information (*chroms* with *name* and *length*); also the binning information is equal in all cells (*bins* with the chromosome name *chrom* and the *start* and *end* position). The group *cells* contains the interaction information per cell which cannot be shared. To achieve the compatibility of an internal cell to the *cooler* format, they follow the same structure (*chroms*, *bins*, *pixels* and *indices*), but shared data (*chroms*, *bins*) is linked to the mutual information in the root of the file format. The group *bins* contain the additional column *weight* to store the matrix correction information individually per file.
For source and license information, please refer to the List of Figures.

Apart from the reads' quality control and their chimeric properties, the single-cell Hi-C files need a third quality control step. The read coverage is narrow, especially in comparison to regular Hi-C. While Hi-C has a read coverage of a few hundred million or even more, single-cell Hi-C interaction matrices have significant less reads. Lando *et al.* [146] list in their review the following read coverages: Flyamer *et al.* [86] with an average of 480,000 contacts per cell, Nagano *et al.* [85] and Stevens *et al.* [87] have on average 70,000 and 80,000 contacts per cell, however, Ramani *et al.* [144] only a bit more than 700 contacts per cell. While this might change in the future, it is currently the case that some matrices have either a too low read coverage or are, in the important area around the main diagonal, too sparse. The sparsity and the low read coverage are problematic for the data analysis of the individual interaction matrices. The data might be so sparse that for several cells, the similarity for the chromatin structure between different cells is meaningless. For these reasons matrices which are too sparse are not considered for the analysis and are removed from the data.

3.3.2 Analysis

The study of single-cell Hi-C data focuses on structural differences and similarities between different cell cycle phases or cell types. scHiCExplorer offers multiple approaches to explore and analyze the single-cell data. First, the interaction matrices of m cells with $(n \times n)$ dimensions are compiled to one $(m \times (n \times n))$ or $(m \times n^2)$ matrix. On raw data, commonly used clustering algorithms like k-means or spectral clustering can be applied with *scHicCluster*. Moreover, the dimensions can be reduced, for example, with a principal component analysis or a k-nearest neighbor graph using the euclidean distance. The number of dimensions is a general issue in single-cell Hi-C. Using, for example, data mapped to the mouse reference genome mm9 creates with a one megabase pair resolution matrix (2700×2700) dimensions, i.e., the compiled interaction matrix has $(m \times 7.2 \; million)$ dimensions; using a 10 kb resolution, the compiled interaction matrix has $(m \times 72.9 \; billion)$ dimensions. The explanatory power of distances in high dimensional space is minor (the so-called 'curse of dimensionality' [147, 148, 149]), and the usage of euclidean distance or similar measures, and therefore the usage of clustering directly on the raw data, is limited.

A major focus in this dissertation concerning the clustering of single-cell Hi-C data was the development of an approach that can correctly distinguish between the subsets in the data. Moreover, the algorithm needs to be able to run on high-resolution single-cell Hi-C data. The clustering of high-resolution single-cell Hi-C data with dimensions in the billions, makes a dimension reduction necessary. To achieve this, an approach based on MinHash [150] was developed [29]. The tested approaches based on principal component analysis or k-nearest neighbor graphs with the euclidean distance did not provide good results. As shown in Wolff *et al.* [29], the principal component analysis on the raw 10-kb data required memory in the petabyte range, and the Euclidean distance is not an optimal distance measure for Hi-C data. The Euclidean distance considers all interactions equally; however, two equidistant measures, e.g., between 0 and 100 contacts, and, between 100 and 200 contacts, should be interpreted differently. Two matrices share similar properties if they have contacts in the same regions, in contrast to a matrix without any contacts in this region. For this reason, a binary measure like the Jaccard index is more appropriate. The Jaccard index is defined as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{3.24}$$

where A, B are the non-zero feature ids of two Hi-C interaction matrices. The first step of a dimension reduction can be computed by a k-nearest neighbors graph. However, creating a k-nearest neighbors graph has a quadratic runtime. This runtime can be reduced to a linear runtime if the Jaccard index is replaced with an approximation: MinHash in combination with an inverse index. MinHash is defined by computing the argmin value over all non-zero feature ids $a \in A$ of a hash function f:

$$h(A) = \ argmin_{a \in A} f(a) \tag{3.25}$$

Hash values for i different MinHash functions are computed for each interaction matrix. The computed values are stored in two ways. First, each interaction matrix is represented by a vector of i hash values, called a signature:

$$signature \ = < h_0(A), ..., h_i(A) > \tag{3.26}$$

Second, the computed hash values and the associated interaction matrix ids are stored in an inverse index per MinHash function. All MinHash values are first computed in $O(m \times i) \in O(m)$ and the hash values are stored in the signatures and the inverse index. To compute the similarity of the matrices and to create the approximate k-nearest neighbor graph, each signature of an interaction matrix is checked for collisions with the inverse index in $O(m \times i \times 1) \in O(m)$. A collision occurs if at least two different matrices have the same hash value for the same MinHash function. In this case, all matrix ids with this hash value are returned by the inverse index. All returned interaction matrix ids are collected, counted and sorted by occurrence. The interaction matrix with the highest occurrence is therefore the most similar one. The precomputing of the hash values and the collision check results in a linear runtime of $O(2 \times m \times i \times 1) \in O(m)$.

To validate the clustering algorithm and to optimize the parameters, pre-labeled single-cell Hi-C interaction matrices with the resolution of 1 Mb and 10 kb from Nagano *et al.* [85] have been used. The results on the 1 Mb interaction matrices show that clustering on the MinHash based approximate k-nearest neighbor graph with its $m \times m$ dimensions does not create a good clustering result [29]. Out of five different cell phases (G1, early-S, late-S/G2, post-M, and pre-M), post-M and pre-M are not identified, and the three others are heavily mixed (Appendix of [29], Table 6). An additional principal component analysis on the approximate k-nearest neighbor's graph followed by a UMAP embedding is necessary. The usage of only one method, PCA or UMAP, also results in a non-distinguishable clustering (Appendix of [29] Table 1, 4 and 5). The comparison of the clustering results of the MinHash based approach and the competitive algorithm *scHiCluster* from Zhou *et al.* [21], demonstrated a high quality clustering results on low-resolution 1 Mb data (Appendix of [29] Table 1 and 18). In particular the two tiny clusters of post-M and pre-M cells could be detected, while Zhou's *scHiCluster* missed them. Furthermore, the MinHash-based approach could compute a clustering on the same single-cell Hi-C data from Nagano using a 10 kb resolution. Around 40 GB of memory and a compute time of just over 6 minutes have been used; while Zhou's *scHiCluster* required over four days to load the data from one chromosome and required almost one terabyte of memory without creating any result. However, the results using the proposed algorithm to cluster the same data on a 10 kb resolution are not good. This can be explained by the lack of an appropriate read coverage for this high resolution and, therefore, the presence of highly sparse structures. The high sparsity leads to less collisions and the implicated similarity is less meaningful.

Several methods are additionally offered to reduce the number of dimensions: *scHicClusterSVL* computes per cell per chromosome the ratio of short to long-range interactions and returns per cell a vector with one ratio per chromosome. *scHicClusterCompartments* computes the A/B compartments for each cell and returns for a cell a vector of size n. The results in [28] show that their ability to create distinguishable clusters is limited in comparison to the MinHash based approach. A major challenge is the availability of compute resources, especially on 10 kb data. Except for the *scHicClusterMinHash* and *scHicClusterSVL*, no approach can compute a result without exceeding a memory usage of one terabyte. Last, the tool *scHicCorrelateMatrices* uses the Pearson or Spearman correlation per interaction matrix to investigate the similarities. However, it is a one value per comparison computation and might be a bit too coarse. Furthermore, the visualization is of limited value when comparing a large amount of cells. The tool is useful to gain insights if the clustering-based, e.g., the MinHash approach, has worked well or a parameter adjustment is necessary (Figure 3.26). The tool *scHicCreateBulkMatrix* can be used to create a single interaction matrix from all the interaction matrices.

3.3.3 Visualization

The need to visualize several thousand interaction matrices, without losing too much information, but retaining important details, requires different visualization methods to Hi-C. The first approach is a cluster profile plot, similar to [85]. The individual matrices are plotted by their associated cluster and then sort internally by their short to long-distance ratio. Per matrix, the ratio of all interactions per genomic distance vs. all interactions is displayed as a heatmap. A good cluster result as shown in Figure 3.27 has a similar profile for the whole cluster. The second visualization method provided in scHiCExplorer is to compute for each cluster a consensus matrix, similar to [85]. It can be computed for the whole interaction matrix or only for a subset like a chromosome, see Figure 3.28. A consensus matrix per cluster is computed by the sum of all contact matrices of the cells associated to a cluster. All consensus matrices are additional normalized to the same value range. A good visualization shows a clear pattern around the main diagonal; however, single cells that are wrongly classified vanish in the consensus data. The third option, Figure 3.29, visualizes the embedded single-cell Hi-C matrices as a scatter plot. The embedded interaction matrices are labeled by their associated cell cycle phase as classified by Nagano 2017 [85]. The scatter plots highlight a difficult subject of this visualization method: Figure 3.29 (a) shows the first

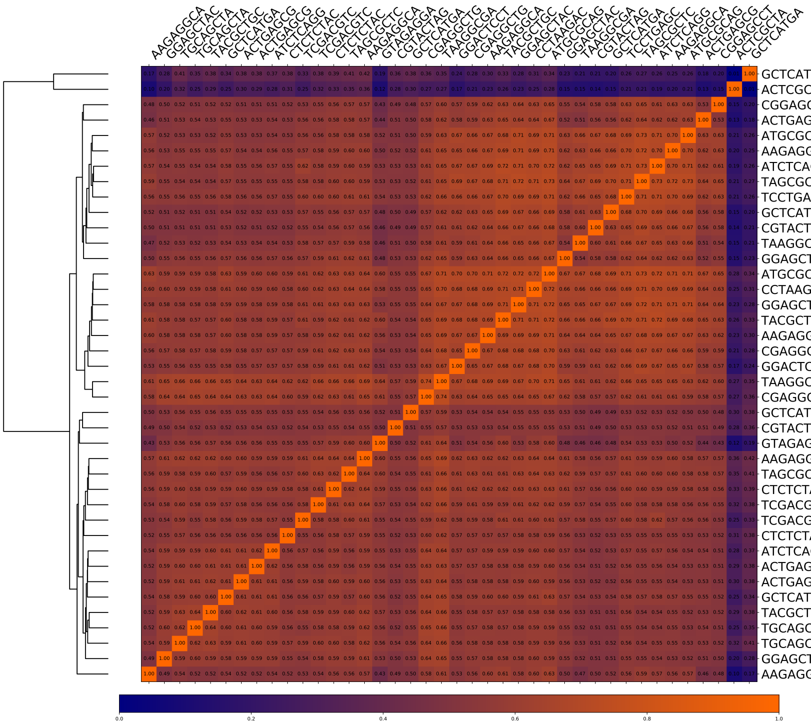
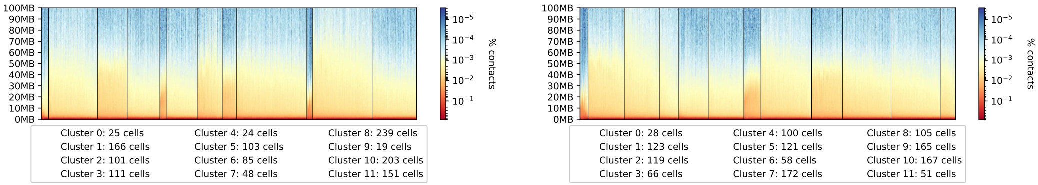


Figure 3.26.: Pearson correlation of pre-classified cell phases pre-M and post-M by Nagano 2017 [85]. Cells labeled by their barcodes.
For source and license information, please refer to the List of Figures.

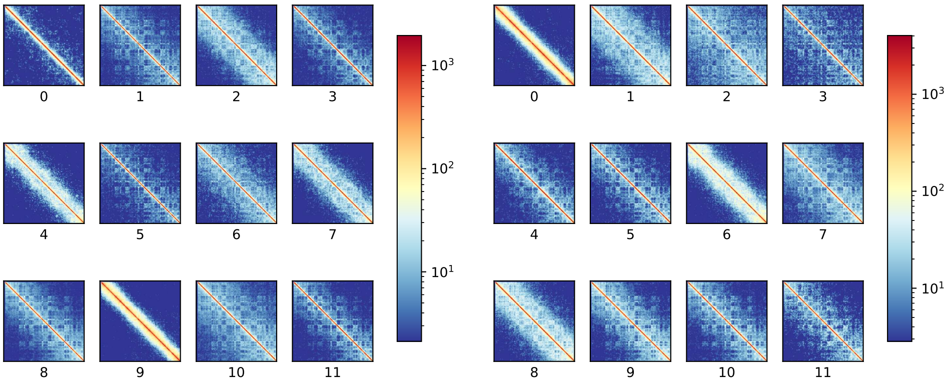
two dimensions of a UMAP embedding to five dimensions. This embedding creates the best cluster results but contains many overlaps of the clusters in the visualization. On the other hand, Figure 3.29 (b) shows the result of a UMAP embedding to two dimensions. The clusters are distinguished, but the clustering algorithms' differentiation power is worse than with five dimensions (see Appendix of [29], Table 1 and 2). Last, every individual single-cell matrix can be visualized with HiCExplorer's *hicPlotMatrix* or be used as an input for *pyGenomeTracks*.



(a) k-nn MinHash

(b) Zhou's scHiCluster

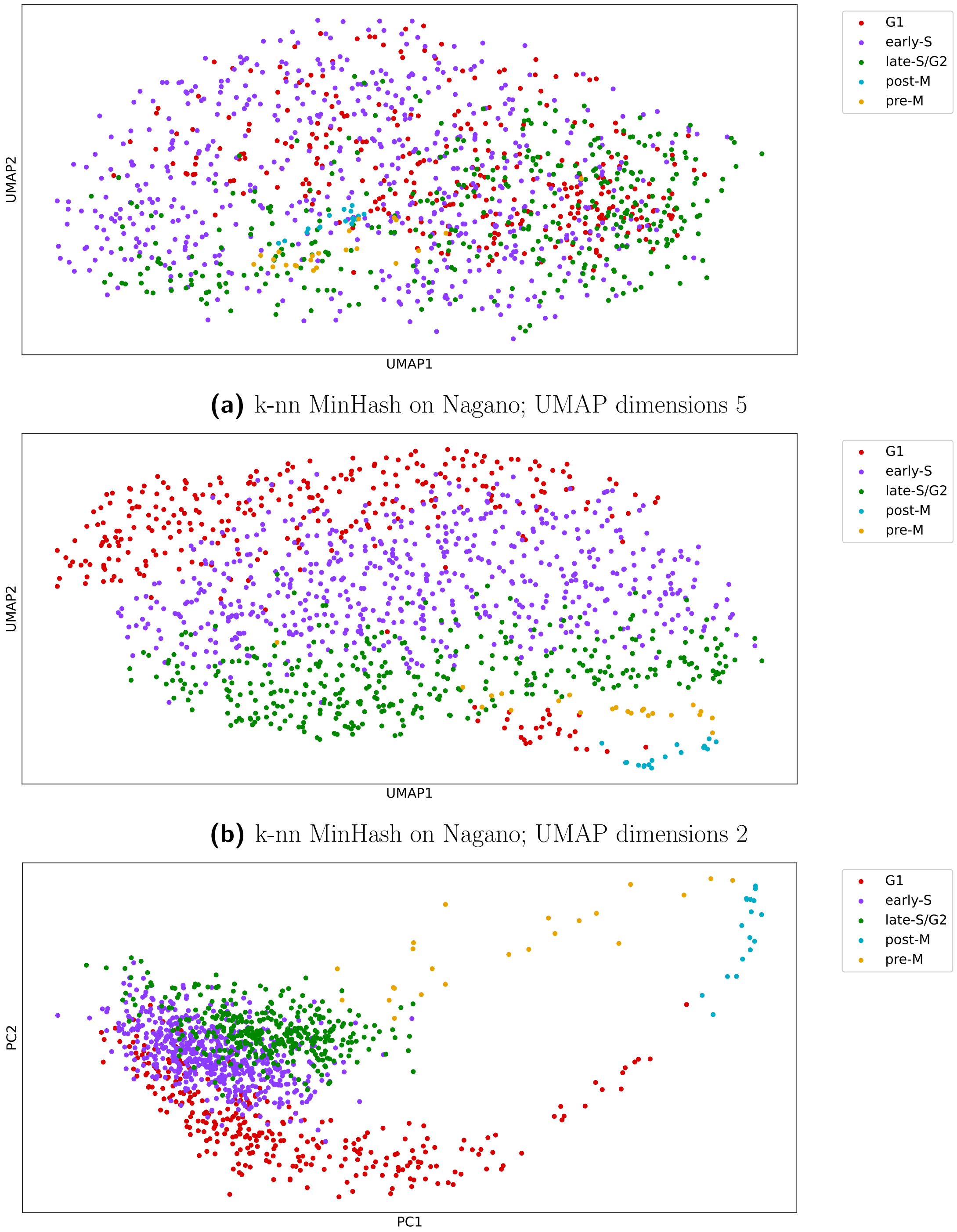
Figure 3.27.: Cluster profile plot for single-cell Hi-C data on Nagano 2017 [85], cell cycle phase data. Cell cycle labels given by Nagano 2017. On the x-axis, the clusters are arranged by their cluster id, and each cell of a cluster is displayed with the ratio of the contacts per genomic distance vs. all contacts of a matrix. The genomic distance is the unit for the y-axis. A good clustering is given if the pattern of the cells of a cluster is very similar. **(a)** Clustering based on *scHicClusterMinHash*. The small clusters of the pre-M and post-M cells (cluster 0 and 9) are clearly distinguished from others. **(b)** The results have been computed by the competitive algorithm *scHiCluster* by Zhou *et al.* [21]. The cells of pre-M and post-M cell cycle phase cannot be distinguished are mixed in cluster 0, additional pre-M cells are part of cluster 4. (See Appendix of [29] Table 1 and Table 18).
For source and license information, please refer to the List of Figures.



(a) k-nn MinHash

(b) Zhou's scHiCluster

Figure 3.28.: Cluster consensus plot for single-cell Hi-C data on Nagano 2017 [85], cell cycle phase data. The cluster labels are provided by Nagano 2017. A consensus matrix is created by summing all contact matrices of the cells of one cluster to one matrix. All consensus matrices are normalized to the same value range. **(a)** Computed with *scHicClusterMinHash*. A clear distinction of different chromatin folding properties per cluster can be observed. The dynamic change in the structure from very dense post-M phase (cluster 9) to slowly opening in the G1 phase (clusters 4, 7, 2), to an intermediate cluster of G1 and early-S cells (cluster 6 and 10) to open chromatin in the early-S-phase (clusters 8 and 1). During the late-S/G2 phase (clusters 11, 3 and 5) the structure becomes denser again to a very dense structure in pre-M phase (cluster 0). **(b)** The result of the competitive algorithm *scHiCluster* from Zhou *et al.* [21]. The chromatin properties are similar to *scHicClusterMinHash* results. However, the pre-M and post-M phases are mixed in cluster 0.
For source and license information, please refer to the List of Figures.



(a) k-nn MinHash on Nagano; UMAP dimensions 5

(b) k-nn MinHash on Nagano; UMAP dimensions 2

(c) Zhou's scHiCluster on Nagano

Figure 3.29.: Scatter plot of different single-cell Hi-C interaction matrix embeddings. Cell labels provided by Nagano *et al.* [85]. **(a)** The embedding to five UMAP dimensions creates better cluster results, but is not optimal for a two dimensional visualization. **(b)** The two dimensional embedding visualizes the data better, but has a worse differentiation power for the clusters. **(c)** Scatter plot of the embedding of the single-cell Hi-C interaction matrices by the algorithm *scHiCluster* from Zhou *et al.* [21]. A very good embedding is provided for G1, early-S and late-S/G2. Also the dynamic of the cell cycle with an arc of G1, post-M and pre-M cells is indicated. However, this embedding makes it difficult for cluster algorithms to distinguish between post-M and pre-M cells.
For source and license information, please refer to the List of Figures.

3.3.4 Matrix manipulations

scHiCExplorer offers multiple functions to operate on the matrices. *scHicAdjustMatrices* can export a subset of the first n matrices or remove individual chromosomes from the data; *scHicMergeMatrixBins* can decrease the resolution of the matrices. *scHicManageScool* can update a single-cell cooler format matrix as used in the previous version of scHiCExplorer (< 4) to the current version, export matrices by their name given in a list either as individual cool files or as a new scool matrix. *scHicConvertFormat* exports scool matrices to the text-based matrix formats as required by Zhou's scHiCluster [21] or as sparse matrix text files. *scHicTxtToScool* can import the text files-based matrices as used by Ramani 2017 [144] and writes them to a scool file.

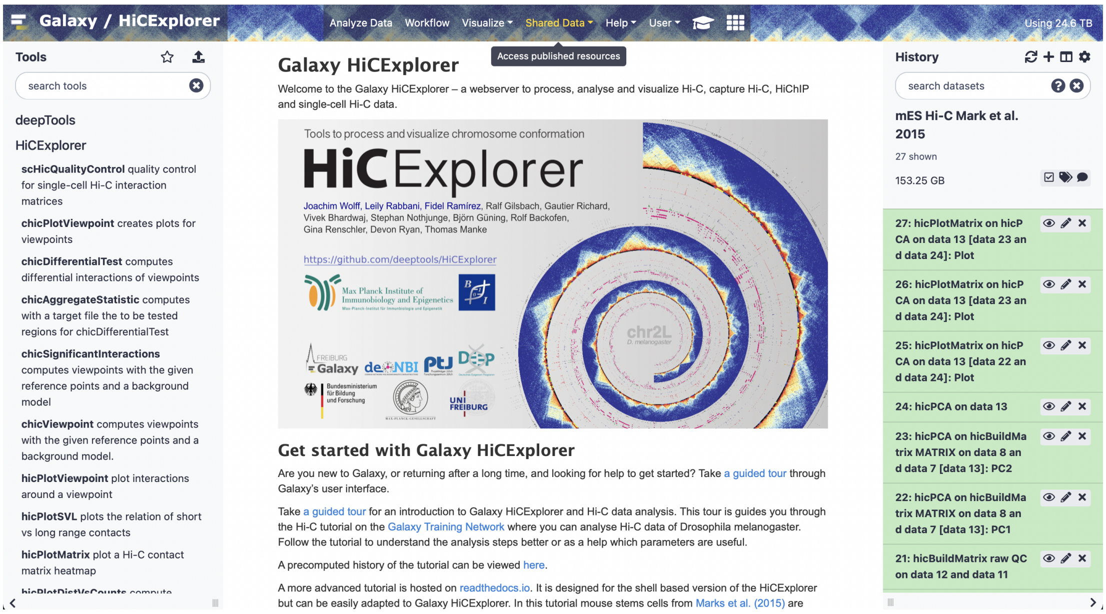
3.4 Webserver

The following section is based on two publications: Wolff *et al.* 'Galaxy HiCExplorer: a web server for reproducible Hi-C data analysis, quality control and visualization', 2018 [27]; and Wolff *et al.* 'Galaxy HiCExplorer 3: a web server for reproducible Hi-C, capture Hi-C and single-cell Hi-C data analysis, quality control and visualization', 2020 [28].

The analysis methods, visualizations, and workflows provided by HiCExplorer, scHiCExplorer, and pyGenomeTracks are based on command-line interface tools available for Linux and macOS-based systems. For computer scientists and users with a bioinformatics background, a standard command-line interface (CLI) is offered. However, many users who want to analyze chromatin conformation capture data have a broad background in microbiology but do not have the necessary skills to use CLI tools. At the same time, these users are the target users for any high-throughput analysis software. To solve this issue, the Galaxy HiCExplorer webserver was developed during this thesis.

Galaxy, introduced in subsection 2.7.3, is a software to provide command-line interface tools in a web-based environment. The Galaxy HiCExplorer integrates HiCExplorer, scHiCExplorer, pyGenomeTracks and additional tools required in data analysis. For example, the software FastQC for quality control of the raw reads, MultiQC, analyzes multiple quality reports of *hicBuildMatrix* in one document, the interactive Hi-C interaction matrix visualization tool HiGlass [151] or additional high-throughput analysis software like deepTools [152] to integrate Hi-C data with ChIP-Seq or RNA-Seq data. The Galaxy HiCExplorer has the advantage that it provides on https://hicexplorer.usegalaxy.eu all tools, documentation and extensive compute resources for a Hi-C data analysis. With this, the HiCExplorer is offered as a SaaS for biomedical researchers. Moreover, Galaxy allows storing data analyses in histories to have a permanent lab-notebook, and all intermediate steps are transparent and replicable. These histories can furthermore be published to provide a better insight into how analyses and publications are computed.

The Galaxy HiCExplorer provides preconfigured workflows to automate intermediate analysis steps; for example, to detect TADs from raw data, the data needs to be mapped, an interaction matrix is created, the data be corrected, and the TADs to be called. These manual steps can be compiled into one manual step with one of the workflows provided. The workflows are based on a graphical user interface similar to bash-based approaches like Snakemake [153] or CWL [154]. As part of the Galaxy HiCExplorer, the Hi-C data analysis tutorial 'Hi-C analysis of Drosophila melanogaster cells using HiCExplorer'[12] is provided via the Galaxy training network [155].


Galaxy / HiCExplorer
Analyze Data
Workflow
Visualize
Shared Data
Help
User
Using 24.6 TB
Access published resources
Tools
search tools
History
search datasets
Tools to process and visualize chromosome conformation
HiCExplorer
Joachim Wolff, Leily Rabbani, Fidel Ramírez, Ralf Gilsbach, Gautier Richard, Vivek Bhardwaj, Stephan Nothjunge, Björn Güning, Rolf Backofen, Gina Renschler, Devon Ryan, Thomas Manke
https://github.com/deeptools/HiCExplorer
Max Planck Institute of Immunobiology and Epigenetics
chr2L
D. melanogaster

deepTools

HiCExplorer

scHicQualityControl quality control for single-cell Hi-C interaction matrices

chicPlotViewpoint creates plots for viewpoints

chicDifferentialTest computes differential interactions of viewpoints

chicAggregateStatistic computes with a target file the to be tested regions for chicDifferentialTest

chicSignificantInteractions computes viewpoints with the given reference points and a background model

chicViewpoint computes viewpoints with the given reference points and a background model.

hicPlotViewpoint plot interactions around a viewpoint

hicPlotSVL plots the relation of short vs long range contacts

hicPlotMatrix plot a Hi-C contact matrix heatmap

hicPlotDistVsCounts compute

Galaxy HiCExplorer

Welcome to the Galaxy HiCExplorer – a webserver to process, analyse and visualize Hi-C, capture Hi-C, HiChIP and single-cell Hi-C data.

Get started with Galaxy HiCExplorer

Are you new to Galaxy, or returning after a long time, and looking for help to get started? Take a guided tour through Galaxy's user interface.

Take a guided tour for an introduction to Galaxy HiCExplorer and Hi-C data analysis. This tour is guides you through the Hi-C tutorial on the Galaxy Training Network where you can analyse Hi-C data of Drosophila melanogaster. Follow the tutorial to understand the analysis steps better or as a help which parameters are useful.

A precomputed history of the tutorial can be viewed here.

A more advanced tutorial is hosted on readthedocs.io. It is designed for the shell based version of the HiCExplorer but can be easily adapted to Galaxy HiCExplorer. In this tutorial mouse stems cells from Marks et al. (2015) are

mES Hi-C Mark et al. 2015

27 shown

153.25 GB

27: hicPlotMatrix on hicPCA on data 13 [data 23 and data 24]: Plot

26: hicPlotMatrix on hicPCA on data 13 [data 23 and data 24]: Plot

25: hicPlotMatrix on hicPCA on data 13 [data 22 and data 24]: Plot

24: hicPCA on data 13

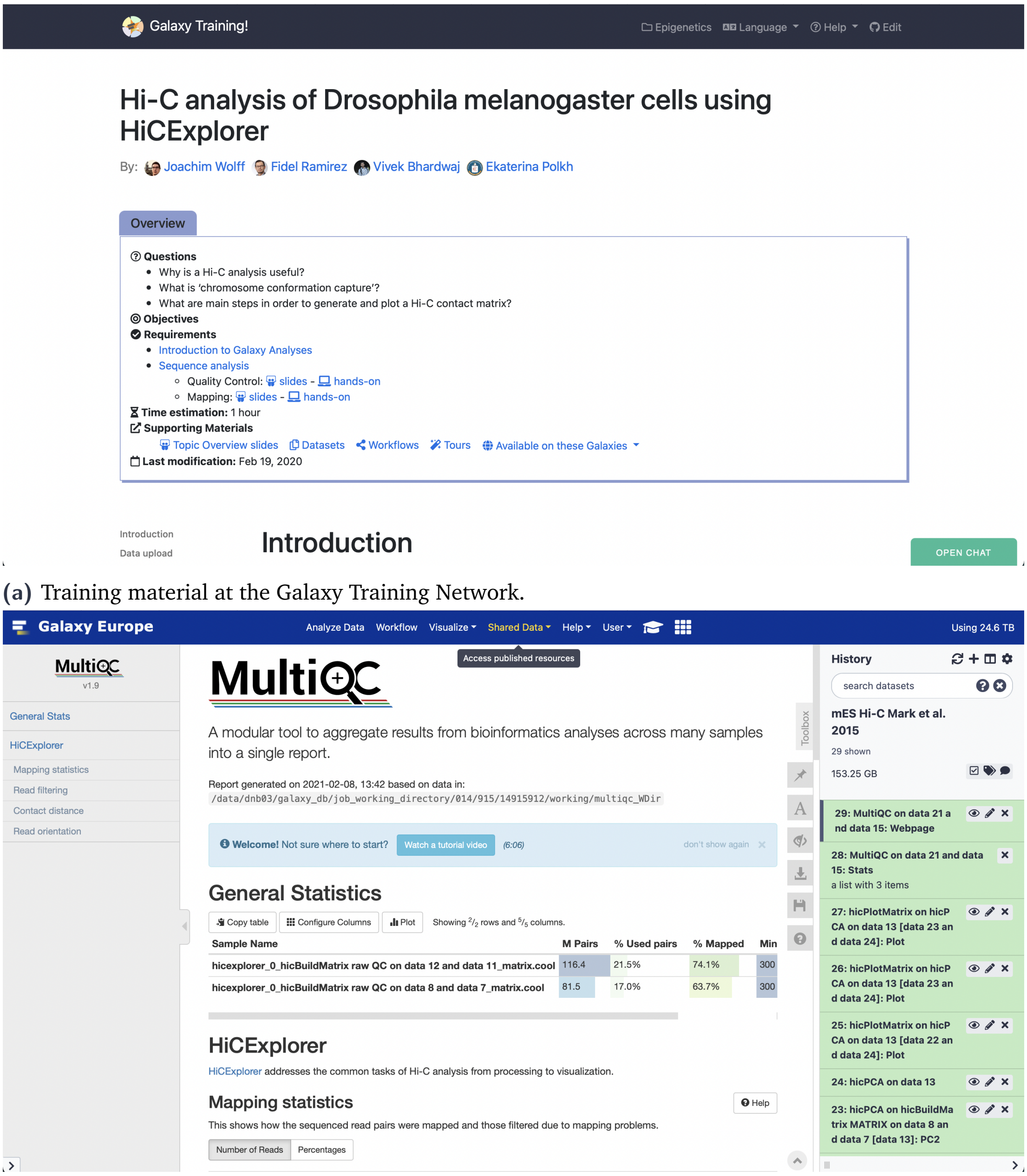
23: hicPCA on hicBuildMatrix MATRIX on data 8 and data 7 [data 13]: PC2

22: hicPCA on hicBuildMatrix MATRIX on data 8 and data 7 [data 13]: PC1

21: hicBuildMatrix raw QC on data 12 and data 11

Figure 3.30.: Galaxy HiCExplorer on https://hicexplorer.usegalaxy.eu. On the left the tool panel with HiCExplorer, scHiCExplorer, pyGenomeTracks and other necessary tools like FastQC, Trimgalore! or deepTools. In the center documentation and help, if a tool is selected it shows the tool parameters. On the right side the history of executed jobs with input and output data.
For source and license information, please refer to the List of Figures.

[12] https://training.galaxyproject.org/training-material/topics/epigenetics/tutorials/hicexplorer/tutorial.html



Hi-C analysis of Drosophila melanogaster cells using HiCExplorer

By: Joachim Wolff, Fidel Ramirez, Vivek Bhardwaj, Ekaterina Polkh

Overview

Questions
- Why is a Hi-C analysis useful?
- What is 'chromosome conformation capture'?
- What are main steps in order to generate and plot a Hi-C contact matrix?

Objectives

Requirements
- Introduction to Galaxy Analyses
- Sequence analysis
  - Quality Control: slides - hands-on
  - Mapping: slides - hands-on

Time estimation: 1 hour

Supporting Materials

Topic Overview slides, Datasets, Workflows, Tours, Available on these Galaxies

Last modification: Feb 19, 2020

(a) Training material at the Galaxy Training Network.

MultiQC

A modular tool to aggregate results from bioinformatics analyses across many samples into a single report.

Report generated on 2021-02-08, 13:42 based on data in:
`/data/dnb03/galaxy_db/job_working_directory/014/915/14915912/working/multiqc_WDir`

Welcome! Not sure where to start? Watch a tutorial video (6:06) don't show again

General Statistics

Showing $^2/_2$ rows and $^5/_5$ columns.

Sample Name	M Pairs	% Used pairs	% Mapped	Min
hicexplorer_0_hicBuildMatrix raw QC on data 12 and data 11_matrix.cool	116.4	21.5%	74.1%	300
hicexplorer_0_hicBuildMatrix raw QC on data 8 and data 7_matrix.cool	81.5	17.0%	63.7%	300

HiCExplorer

HiCExplorer addresses the common tasks of Hi-C analysis from processing to visualization.

Mapping statistics

This shows how the sequenced read pairs were mapped and those filtered due to mapping problems.

History

search datasets

mES Hi-C Mark et al. 2015

29 shown

153.25 GB

- 29: MultiQC on data 21 and data 15: Webpage
- 28: MultiQC on data 21 and data 15: Stats
  a list with 3 items
- 27: hicPlotMatrix on hicPCA on data 13 [data 23 and data 24]: Plot
- 26: hicPlotMatrix on hicPCA on data 13 [data 23 and data 24]: Plot
- 25: hicPlotMatrix on hicPCA on data 13 [data 22 and data 24]: Plot
- 24: hicPCA on data 13
- 23: hicPCA on hicBuildMatrix MATRIX on data 8 and data 7 [data 13]: PC2

(b) MultiQC for HiCExplorer QC reports.

Figure 3.31.: Training and MultiQC for HiCExplorer. The training material covers the basic analysis steps for a Hi-C data analysis. MultiQC provides an overview of multiple QC reports from FastQC or HiCExplorer's *hicBuildMatrix*.
For source and license information, please refer to the List of Figures.

4 Discussion

The primary aim at the beginning of this thesis, to unify many pre-processing, analysis, and visualization steps of a Hi-C data analysis in one software, was achieved. The integration of the *cooler* file format as proposed by Abdennur *et al.* [116] provides improved interoperability and reproducibility in the Hi-C field. More and more Hi-C data analysis projects adopt the *cooler* file format, for example cooltools[1] or chromosight [140]. However, for modern data analysis it is crucial that the same software setting can be used to reproduce the analysis results of a study. While many other Hi-C data analysis software need to be installed manually, and the dependencies have to be resolved by the user, HiCExplorer is available as a Conda package in all its versions. It takes one command to install an older HiCExplorer version with all of the old dependencies. A Docker container is also provided achieving the goal of software preservation. The more reliable installation routines improve the reproducibility of analysis results. Moreover, by providing the web server *https://hicexplorer.usegalaxy.eu*, the presented environment is the only one providing Hi-C data analysis as SaaS. SaaS opens the computationally resource-demanding analysis of Hi-C data to a broader range of researchers by providing large high-performance computing (HPC) and cloud computing resources in the background. The integration to the Galaxy environment brings additional benefits: First, all processing steps, the input data, the parameters, and the exact version of the used software are logged. This makes a data analysis less daunting and reduces human-introduced errors. The history can be shared with users using the same Galaxy platform or publicly to transparently publish the research results and the path from the raw data to the results. Second, Galaxy provides workflows to process tools and their output in a consecutive way, reducing the number of manual analysis steps. The requirement to use multiple analysis software was reduced; however, if it is required, import and export to external formats are supported.

During this thesis, a comprehensive single-cell Hi-C data analysis software, scHiCExplorer, has also been developed. scHiCExplorer is the only software available that covers all aspects of pre-processing, analysis, and visualization. Most competitors focus on solving only one problem; however, creating the data from scratch is the user's task. Many publications published their single-cell Hi-C data either in a raw FASTQ format that required long pre-processing times or text file-based matrices. The manual handling of potentially a few thousand to ten-thousand files makes human errors inevitable. The availability only as raw data and the need to create the interaction matrices from

[1] https://github.com/open2c/cooltools

scratch can cause differences in a third-party replication of the published results. The developed single-cell *cooler* format *scool* solves this issue. The presented approach to cluster single-cell Hi-C data based on a dimension reduction with MinHash can create better cluster results than the competitors and detect small clusters. The MinHash based solution can cluster high-resolution single-cell Hi-C data with a relatively low memory footprint, while the competitors cannot compute results with less than one terabyte of memory. Considering that read coverage and cell number will only continue to rise, the improved methods to store and process single-cell Hi-C data will become even more critical in the future.

The data generated by high-throughput sequencing approaches is never error-free and therefore a good quality control of the data is crucial. The software created and discussed in this thesis provides this. The raw reads can be checked by FastQC, which is integrated into the Galaxy HiCExplorer webserver, also tools for trimming the reads are integrated. A second quality control for specific Hi-C properties is achieved at the build time of the interaction matrix. The created quality reports can be pooled with MultiQC to one report. The capture Hi-C modules provide a check for too sparse viewpoints, and the single-cell Hi-C analysis software scHiCExplorer provides a check for too sparse matrices.

A reliable and reproducible visualization is the key to understanding research results better. Many biomedical research publications create figures in a non-reproducible and problematic manner by combining plots from different analysis types via image manipulation software like Inkscape or Adobe Illustrator; others use simply screenshots of genome browsers. These widespread methods counteract the scientific community's efforts to publish transparent results that anyone can reproduce. pyGenomeTracks offers a unique approach by providing an initialization file where the reference to the specific raw data is stored, together with the parameters used to create the diagram. It allows to combine data stored in the most common file formats like *bigwig*, *bed*, *bedgraph* or *cool* independent of the high-throughput approach that generated the data.

The chromatin structure is receiving increased attention in high-throughput data-based experiments. The chromatin structure's explanatory power for processes like transcription regulation is high; only with it, the contact between DNA sequences and, therefore, with transcriptional regulating elements can be shown. With decreasing costs and time efforts caused by improved wet-lab protocols, more and more labs consider adding a Hi-C technique based analysis to their experiments. However, Hi-C has a few limitations. Hi-C assumes that a contact between DNA sites implies proximity within the nuclei, but this needs further verification with orthogonal methods like 3D DNA FISH. Contact of two regions could also be the fixation step's accidental product, and any contact can be biased by the fixation, digestion, or ligation step of Hi-C.

In this context the verification of the detect results of Hi-C experiments is important. Rao *et al.* [14] used 3D FISH to validate four detected loops of Hi-C data; Sanborn *et al.*

[156] used CRISPR mediated genome editing to change the CTCF binding motif at 13 loop locations and these loops disappeared. Additional to the motif editing, Sanborn *et al.* modified the CTCF binding domains, resulting in a disruption of many loops, which proves the observations made with Hi-C and the correlations to CTCF. Foissac *et al.* [157] confirmed the Hi-C findings of A/B compartments and TADs with ATAC-Seq [158].

Moreover, Hi-C can only detect the contact between two DNA sites; however, it seems unlikely that this contact can occur exclusively between two sites without involvement of additional sites in the nucleus. These concerns are solved by a variety of alternative techniques to determine the chromatin structure. First, 3D DNA FISH provides insights into the actual distance between DNA contacts based on optical fluorescence methods. However, it cannot be used in a high-throughput approach; only a limited number of interactions can be verified. GAM [159] uses cryosectioning and laser microdissection to create nuclear slices and provides a ligation free approach. Contacts are determined by counting the co-segregation frequency of two regions in a slice. An additional model, however, is used to correct for random and accurate contacts of close distances (100nm). GAM allows the recording of chromatin contacts of three or more regions. SPRITE [160] crosslinks the DNA like 3C based methods but has no ligation step. The crosslinked DNA fragments are split across well plates; each adds a different barcode to the fragments. In a five-fold iteration, the fragments are re-pooled, split again to the wells and get additional barcodes attached. The reads with the same barcode combination are assumed to be the ones initially crosslinked. Like Hi-C, SPRITE can be used to detect loops and TADs and can, due to the ligation-free approach, detect multiple contacts. For example, long-range contacts are better detectable as well as super-enhancer regions. The integration of ligation free approaches to the chromatin conformation capture analysis software stack needs to be implemented in the future. Techniques like SPRITE promise to clarify Hi-C's particular concerns and extend the biological insights by providing multi-contact sides.

A different area of epigenetics are regulating RNAs [161]. RNA types like dsRNA, siRNA, or miRNA have an essential role in regulating mRNA translation to proteins by interacting with and binding to mRNAs, respectively deacetylation of the poly-A tail. However, the role of siRNA is of interest in the context of the chromatin structure. The role of active siRNA molecules, the *RNA-induced silencing complex (RISC)* is important to inhibit the translation of RNA by the active formation of heterochromatin. The argonaute protein locates specific chromosomal regions with the help of the siRNA, and growing transcripts are recognized. This leads to the increased likeliness of methylation of H3K9, and the condensation of the chromatin area. The role of these RNA interferences and their implication for the regulation of chromatin structure is a growing research area to understand the regulating mechanisms in a cell. Hi-C can help solve this, but it is required to combine it with methods that can detect the interferences of the RNA. A technique is for example the high-throughput screening of RNAi [162].

Last, the chromatin structure simulation might help to investigate the folding principles and the involved elements better. Considering Hi-C, it is obvious to try to predict Hi-C interaction matrices as a first step into the area of artificially created chromosome folding data. Zhang *et al.* [163] used a random forest approach using structure-associated proteins as features and the Hi-C interaction matrix as a target. Schwessinger *et al.* [164] use the DNA base pairs to predict the 3D structure using a deep neural network. During this theses, two master projects were supervised to replicate and improve the ideas from Zhang *et al.* (Bajorat[2], Krauth[3]); and one master thesis investigating an approach based on neural networks was completed (Krauth)[4]. The approach based on conditional generative adversarial networks creates simulated high-resolution Hi-C data (see Figure 4.1) and is worth investigating as the foundation for a simulation of chromosome conformation capture under differing conditions. The in silico procedure has the potential to help to understand the biological processes faster and better, but all in silico findings need to be validated in vitro.

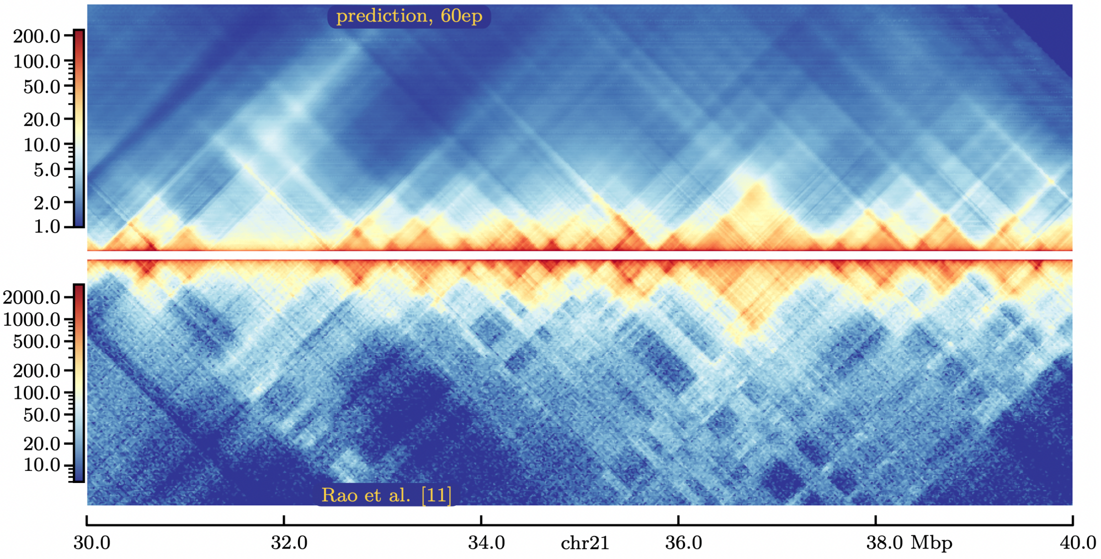


Figure 4.1.: Hi-C prediction with a cGAN approach. **Top:** The prediction of Gm12878 cell line based on the learned K562 cell line. Chromosome 21 30 - 40 Mb. **Bottom:** Gm12878 Hi-C matrix from Rao *et al.* [14].
For source and license information, please refer to the List of Figures.

[2] http://www.bioinf.uni-freiburg.de/Lehre/Theses/TP_Andre_Bajorat.pdf

[3] http://www.bioinf.uni-freiburg.de/Lehre/Theses/P_Ralf_Krauth_Report_Project.pdf

[4] https://github.com/MasterprojectRK/reportMasterthesisRK/blob/master/thesis_main.pdf

5 Galaxy HiCExplorer: a web server for reproducible Hi-C data analysis, quality control and visualization

Galaxy HiCExplorer: a web server for reproducible Hi-C data analysis, quality control and visualization.
Joachim Wolff, Vivek Bhardwaj, Stephan Nothjunge, Gautier Richard, Gina Renschler, Ralf Gilsbach, Thomas Manke, Rolf Backofen, Fidel Ramírez, Björn Grüning.
Nucleic Acids Research, Volume 46, Issue W1, 2 July 2018, Pages W11-W16
https://doi.org/10.1093/nar/gky504

Personal contribution

I contributed by conceiving and writing the manuscript and the presented Galaxy HiCExplorer webserver. Moreover, I extended the software HiCExplorer by designing and implementing new tools: *hicPCA, hicTransform, hicPlotViewpoint*; contributed by substantially improving the tool *hicBuildMatrix* and by adding support for the new *cooler* file format. I contributed by writing the Galaxy Training Network tutorial, improving the general documentation of HiCExplorer and by creating documentation for the webserver. Furthermore, I developed workflows for HiCExplorer on Galaxy. In recognition of these significant contributions, I am listed as the first author for this publication.

Contribution of Vivek Bhardwaj

Contributed by implementing features of HiCExplorer; general contributions to the source code of HiCExplorer and discussions about Hi-C approaches.

Contribution of Stephan Nothjunge

Contributed by using and testing the HiCExplorer with Hi-C data and by reviewing the manuscript.

Contribution of Gautier Richard

Contributed by general contributions to the source code and discussions about Hi-C approaches. Design of the used graphics.

Contribution of Gina Renschler

Contributed by using and testing the HiCExplorer with Hi-C data; wrote substantial amounts of the documentation. Participating in writing and reviewing the manuscript.

Contribution of Ralf Gilsbach

Contributed by using and testing the HiCExplorer with Hi-C data and by reviewing the manuscript.

Contribution of Thomas Manke

Participating in writing and reviewing the manuscript.

Contribution of Rolf Backofen

General PhD supervision of Joachim Wolff and advice during the PhD process.

Contribution of Fidel Ramírez

Contributed by starting the HiCExplorer project and implementing the basic functionality of it: Building of the matrix, correction, TAD detection, and visualization. Helped to write and review the manuscript.

Contribution of Björn Grüning

Contributed by writing first versions of the Galaxy wrappers for the webserver. General PhD supervision of Joachim Wolff and advice during the PhD process. Review of the manuscript.

Joachim Wolff

Joachim Wolff

The following co-authors confirm the above stated contribution:

Name	Date	Signature
Dr. Vivek Bhardwaj	16.04.2021	VivekBhardwaj
Dr. Stephan Nothjunge	12.04.21	S.Noth
Dr. Gautier Richard	13/04/2021	G Richard
Dr. Gina Renschler	12.04.21	
Prof. Dr. Ralf Gilsbach	14.4.21	Prof. Dr. Ralf Gilsbach Institut für Kardiovaskuläre Physiologie FB Medizin | Goethe Universität Theodor-Stern-Kai 7 60590 Frankfurt am Main
Dr. Thomas Manke	12.04.2021	
Prof. Dr. Rolf Backofen	22.04.2021	Digital unterschrieben von Prof. Dr. Rolf Backofen Datum: 2021.04.22 21:03:33 +02'00'
Dr. Fidel Ramírez	16.04.2021	Fidel R.
Dr. Björn Grüning	22.04.2021	B. Grüning

Published online 13 June 2018 *Nucleic Acids Research, 2018, Vol. 46, Web Server issue* ***W11–W16***
doi: 10.1093/nar/gky504

Galaxy HiCExplorer: a web server for reproducible Hi-C data analysis, quality control and visualization

Joachim Wolff[1], Vivek Bhardwaj[2,6], Stephan Nothjunge[5,8], Gautier Richard[2,7], Gina Renschler[2,6], Ralf Gilsbach[5], Thomas Manke[2], Rolf Backofen[1,3,4], Fidel Ramírez[2,*] and Björn A. Grüning[1,3,*]

[1]Bioinformatics Group, Department of Computer Science, University of Freiburg, Georges-Köhler-Allee 106, 79110 Freiburg, Germany, [2]Max Planck Institute of Immunobiology and Epigenetics, Stübeweg 51, 79108 Freiburg im Breisgau, [3]Center for Biological Systems Analysis (ZBSA), University of Freiburg, Habsburgerstr. 49, 79104 Freiburg, Germany, [4]BIOSS Centre for Biological Signaling Studies, University of Freiburg, Schänzlestr. 18, 79104 Freiburg, Germany, [5]Institute of Experimental and Clinical Pharmacology and Toxicology, Faculty of Medicine, University of Freiburg, Albertstr. 25, 79104 Freiburg, Germany, [6]Faculty of Biology, University of Freiburg, Schänzlestr. 1, 79104 Freiburg, Germany, [7]IGEPP, INRA, Agrocampus Ouest, Univ Rennes, 35600 Le Rheu, France and [8]Hermann Staudinger Graduate School, University of Freiburg, Hebelstrasse 27, 79104 Freiburg, Germany

Received February 23, 2018; Revised April 27, 2018; Editorial Decision May 10, 2018; Accepted May 22, 2018

ABSTRACT

Galaxy HiCExplorer is a web server that facilitates the study of the 3D conformation of chromatin by allowing Hi-C data processing, analysis and visualization. With the Galaxy HiCExplorer web server, users with little bioinformatic background can perform every step of the analysis in one workflow: mapping of the raw sequence data, creation of Hi-C contact matrices, quality assessment, correction of contact matrices and identification of topological associated domains (TADs) and A/B compartments. Users can create publication ready plots of the contact matrix, A/B compartments, and TADs on a selected genomic locus, along with additional information like gene tracks or ChIP-seq signals. Galaxy HiCExplorer is freely usable at: https://hicexplorer.usegalaxy.eu and is available as a Docker container: https://github.com/deeptools/docker-galaxy-hicexplorer.

INTRODUCTION

Chromosome conformation capture techniques are now widely used to analyse the 3D conformation of chromatin inside the nucleus across a rising number of species, tissues and experimental conditions. In particular, the Hi-C protocol (1) has helped to uncover folding principles of chromatin, demonstrating that the genome is partitioned into active and inactive compartments (called A and B) (1) and that these compartments are further subdivided into topological associated domains (TADs) (2,3). Furthermore, Hi-C has allowed identification of chromatin loops (4,5), as well as enhancer–promoter interactions (6,7) and their influence on gene expression (8,9).

However, Hi-C data processing requires tabulating hundreds of millions to billions of paired-end reads into large matrices. This poses bioinformatic challenges for efficient processing of the data and subsequent analyses. Here, we introduce Galaxy HiCExplorer, a package that aims to make Hi-C data processing, analysis and visualization available to non-bioinformaticians. Our goal is to provide a software environment able to automate the whole workflow of Hi-C data analyses from raw read mapping, filtering and correction, to the computation of topological associated domains and A/B compartments, and finally to the visualization of contact matrices, along with various other genomic features and omics data. Moreover, Galaxy HiCExplorer is easy to install, maintainable, stable and well documented. The availability of a docker container in conjunction with Bioconda (http://dx.doi.org/10.1101/207092), eliminates the need for complex software and dependency installations. Finally, HiCExplorer is transparently developed by a community of collaborators based on best practices (10) for version control, code revisions, manual and automated testing and comprehensive documentation.

COMPREHENSIVE SERVER FOR HI-C ANALYSES

Galaxy HiCExplorer is freely available at https://hicexplorer.usegalaxy.eu as well as a Docker container: https://github.com/deeptools/docker-galaxy-hicexplorer. Galaxy HiCExplorer was designed to provide an easily accessible data-analysis environment such that

[*]To whom correspondence should be addressed. Tel: +49 761 2037460; Fax: +49 761 2037462; Email: gruening@informatik.uni-freiburg.de
Correspondence may also be addressed to Fidel Ramírez. Email: ramirez@ie-freiburg.mpg.de

biomedical researchers can focus on critical research aspects instead of dealing with terminal-based applications that are not user-friendly. It smoothly integrates the HiCExplorer analysis toolset (8) into the Galaxy scientific analysis platform to provide web-based, easy-to-use and thoroughly tested workflows that provide pipelines for the most common Hi-C data processing steps.

In contrast to other available Hi-C analysis software like HiCUP (14), HOMER (15) and TADbit (16) among others (see (17,18) for a comprehensive list of tools), Galaxy HiCExplorer provides a fully comprehensive analysis pipeline available to much broader community of researchers and is not restricted to a subset of important features. HiC-Pro (19) is one of the few packages that offers a complete pipeline; however, its visualization tools are limited and it is only available as a command line tool. Similarly, Juicer (20) offers a command line tool processing pipeline while Juicebox (21) only provides visualizations. Moreover, the integration of HiCExplorer into Galaxy offers the possibility to process and integrate other data types like ChIP-Seq or RNA-Seq into the analysis using the same interface. None of the aforementioned tools offer web server access except HiFive (22).

A strong advantage of HiCExplorer is that it can take multiple matrix data formats developed by different research groups as input. Thus, it is well integrated in the landscape of Hi-C data analysis algorithms, as Hi-C matrices can be produced by other tools and visualized with HiCExplorer. Conversely, matrices can be created with HiCExplorer and then exported to be used by other software. Currently, the Galaxy HiCExplorer supports two major formats: The HiCExplorer specific h5 format and to promote standardization of Hi-C contact matrices the cooler format (23) developed within the 4D nucleome project (24).

GALAXY HiCExplorer TOOLS AND WORKFLOWS

Galaxy HiCExplorer provides a plethora of tools for processing, normalization, analysis, and visualization of Hi-C data (Figure 1A). Apart from HiCExplorer, the https://hicexplorer.usegalaxy.eu website and the Docker container also include the genome alignment tools BWA-MEM (25) and Bowtie2 (26), as well as additional tools for text manipulation, data import and quality control. The inclusion of deepTools (27) further facilitates the integration of ChIP-seq, RNA-seq, MNase-seq as well as other kind of datasets with Hi-C data.

The analysis of Hi-C data can be divided into three steps: pre-processing (including quality control), analysis and visualization.

Pre-processing and quality control

hicBuildMatrix. A contact matrix is the main data structure of Hi-C data analysis which is generated from the individual alignment of valid Hi-C paired-end reads. This tool filters out potentially erroneous reads, such as unmappable reads, self-ligated reads, dangling-ends, PCR duplicates or incomplete digestions (4,14) and tabulates the results based on user defined bins (either based on restriction sites or on fixed size bins). Because building the Hi-C matrix is one of the most time consuming steps in the Hi-C workflow, we developed *hicBuildMatrix* to be multi-processing to significantly reduce running time. A comprehensive quality report is generated as an HTML file. This report includes a number of useful quality measures including: number of valid Hi-C read pairs and the number of filtered reads per category (unmappable and non-unique pairs, duplicates, dangling ends, self-circles, etc.), number of intra-chromosomal, short-range (<20 kb) and long-range contacts, and read pair orientation. Reports from multiple samples can be integrated using MultiQC (28) or using the HiCExplorer tool *hicQC*. Inspection of the *hicBuildMatrix* quality reports helps to identify potential biases or errors in the Hi-C library preparation. For example, a high number of dangling ends is indicative of a problem with the re-ligation step or inefficient removal of dangling ends. The quality report can also be useful to identify differences (long-range versus short-range contacts enrichment for instance) between samples obtained in different conditions.

hicMergeMatrixBins. After a Hi-C contact matrix has been created, lower resolution matrices can be obtained by merging neighboring bins. This is mostly useful for visualization at different zoom levels or to create matrices of lower resolution (larger bin size) in the event of a Hi-C matrix being too poor due to low sequencing depth.

hicCorrelate. This tool computes the correlation between several Hi-C matrices (Figure 1B). *hicCorrelate* can produce a scatter plot or a heatmap using either Pearson or Spearman correlations. The computation of the correlation can be restricted to a range of genomic distances to avoid biasing the correlation results with background contacts. These correlations are useful as a quality control step to compare replicates and to test for differences between various treatments.

hicPlotDistVsCounts. This tool plots the average number of Hi-C contacts at different genomic distances (Figure 1C). It allows the estimation of long-range and short-range contacts from multiple samples at once, and is a useful tool for both quality control and comparison of, for example, treated versus untreated samples that alter chromosome conformation.

hicSumMatrices. After different replicates or similarly obtained Hi-C matrices have been compared using *hicCorrelate*, they can be added up into one single contact matrix with this tool.

hicCorrectMatrix. Allows the removal of biases from the Hi-C matrix using a very fast version of the iterative correction algorithm from Imakaev *et al.* (29). Before the contact matrix is corrected, the right thresholds to prune values need to be selected. The *diagnostic plot* helps users in determining these thresholds.

Analysis

hicFindTADs. This utility can identify TADs from a given corrected contact matrix by first computing a TAD-

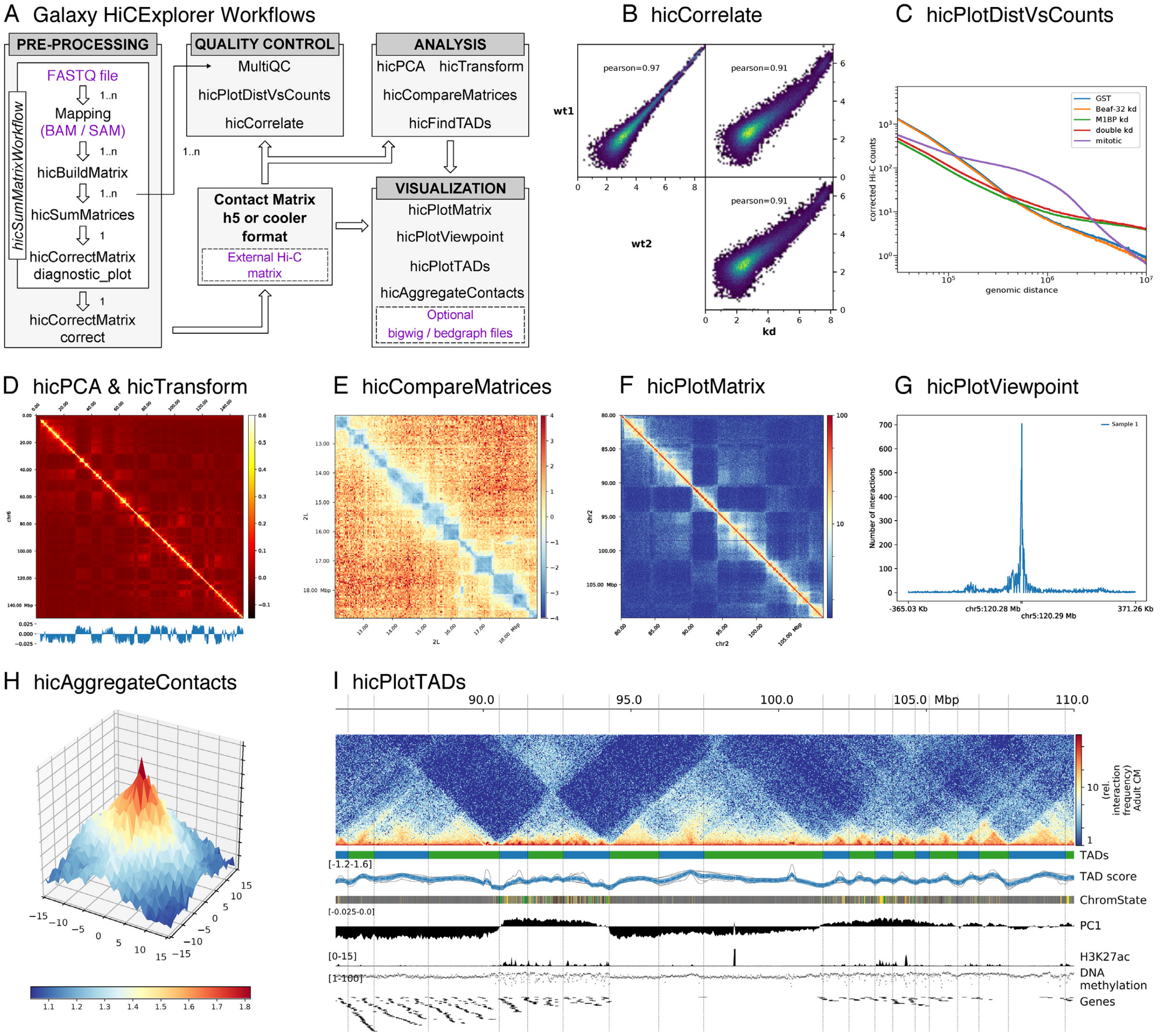


Figure 1. (**A**) Galaxy HiCExplorer workflows and tools. Entry points for external data are highlighted in purple. **Quality control tools:** (**B**) Output of *hicCorrelate* comparing two wild types and one knockdown samples. (**C**) Output of *hicPlotDistVsCounts* that shows changes of the number of contacts for different conditions. **Analysis tools:** (**D**) *hicPlotMatrix* of the Pearson correlation matrix derived from a contact matrix for chromosome 6 in mouse computed with hicTransform. The optional data track at the bottom shows the first eigenvector for A/B compartment obtained using *hicPCA*. (**E**) The pixel difference between a Hi-C corrected matrix for wild type condition and a knock down was computed using *hicCompareMatrices* and a 7Mb region is visualized using *hicPlotMatrix*. **Visualization tools:** (**F**) Contact matrix plot of a 80 to 105 Mb region of chromosome 2 in log scale. (**G**) Example output of *hicPlotViewpoint* showing the corrected number of Hi-C contacts for a single bin in chromosome 5 (output similar to 4C-seq) (11). (**H**) A Hi-C matrix was converted into an observed vs. expected matrix using *hicTransform* and this matrix, together with the location of high-affinity sites from (12) were used to run *hicAggregateContacts*. (**I**) 85 Mb to 110 Mb region from human chromosome 2 visualized using *hicPlotTADs*. TADs were computed by *hicFindTADs*. The additional tracks added correspond to: TAD- separation score (as reported by *hicFindTADs*), chromatin state , principal component 1 (A/B compartment) computed using *hicPCA*, ChIP-seq coverage for the H3K27ac mark, DNA methylation, and a gene track. Hi-C data for B, C, E and H from *Drosophila melanogaster* S2 cells from (8). Hi-C data for D, F and I from mouse cardiac myocytes (13). Additional tracks in I from (13).

separation score and then identifying local minima indicative of TAD boundaries (8). In contrast to other TAD identification methods, this tool also returns the TAD-separation score, which can be visualized in a genome browser or using *hicPlotTADs*. The TAD-separation score contains useful information to identify strong and weak boundaries and the density of contacts within TAD and can be visualized along with the Hi-C matrix (see *hicPlotTADs* tool).

hicPCA. A/B compartments (1) refer to open and closed chromatin that is spatially separated in the cell nucleus (30,31). We compute this using eigenvector decomposition as described by Lieberman-Aiden (1) and using the first and second eigenvector. The positive/negative values correspond to open/closed chromatin. A visualization of A/B compartments is shown in Figure 1D.

hicTransform. The three matrices used to compute the A/B compartments (observed/expected, Pearson correlation and covariance matrices) are useful during visualization to achieve a better understanding of the Hi-C data. To enable this, *hicTransform* can compute these three matrices independently of *hicPCA*, and the matrices can then be plotted using the visualization tools.

hicCompareMatrices. *hicCompareMatrices* allows the computation of difference, ratio or log2ratio between two matrices. This is useful to compare replicates or samples from different conditions. It can, for example, help to characterize TAD structure modifications when followed by *hicPlotMatrix* (Figure 1E).

Visualization

hicPlotMatrix. This tool is used to plot contact matrices for a collection of individual chromosomes. It has multiple options to select the matrix colors and the values range. Additionally, *bigwig* tracks can be attached to plot additional features such as A/B compartments or ChIP-seq data. It is possible to plot a multitude of domains; the entire interaction matrix, individual chromosomes, multiple chromosomes, and various regions of interest (see Figure 1D–F).

hicPlotViewpoint. The viewpoint plot supports a visualization of the number of interactions around a specific reference point or region in the genome, and makes the long-range interactions visible as shown in Figure 1G. The output is comparable to what is obtained using the 4C-seq protocol.

hicAggreateContacts. Facilitates the analysis of long range-contacts by visualizing the average contacts over multiple smaller matrices around a given set of regions (Figure 1H).

hicPlotTADs. To visualize the computed TADs this tool flips the main diagonal of the Hi-C contact matrix by 45° and marks the TADs with triangles. It is possible to plot multiple matrices and add additional data like genes, chromatin states, long-range interactions and any other feature that can be represented as a bigwig or bedgraph file like methylation data, ChIP-seq, or RNA-seq to visually correlate them with TADs and their boundaries. There are multiple options to select the Hi-C matrix layout and colormap, different ways to visualize genes and regions files and also multiple configurations to plot coverage tracks like color, line width, line type, as dots, filled etc. (Figure 1I).

Workflows

Galaxy HiCExplorer provides pre-defined workflows to reduce intermediate steps and to guide a researcher through the different stages. The Galaxy framework offers the possibility to connect tools into workflows called Galaxy workflows. The provided workflows are subdivided into categories depending on the start of the analysis: First, raw FASTQ files are mapped to generate a contact matrix and its corrected equivalent. Different workflows are provided to cover the case of running many analyses in parallel or whether replicates should be merged to one contact matrix. Second, said contact matrix (or other) is used to compute TADs, A/B compartments and/or to plot them using the provided workflows. All workflows are linked on the homepage of the Galaxy HiCExplorer.

All Galaxy Workflows share a common notion that they should guide the researcher through the analysis, i.e. most parameters in the workflows do not need to be changed. The reference genome needs to be set for the mappers, and a desired bin size as well as the used restriction sites needs to be selected in order to build the contact matrix. Every workflow containing a plotting step needs the region to plot as input.

IMPLEMENTATION

Galaxy HiCExplorer is implemented as a Docker container based on the web-based Galaxy scientific workflow platform (32). HiCExplorer itself is implemented in Python, supporting version 2.7, 3.5 and 3.6, and available as a Bioconda package (http://dx.doi.org/10.1101/207092) and as BioContainer (33). This guarantees a fixation of versions and therefore reproducibility of analysis. Galaxy wrappers for HiCExplorer are available at the Galaxy tool shed.

USING HiCExplorer

Installation and usage

The Galaxy HiCExplorer web server can be used by visiting http://hicexplorer.usegalaxy.eu, or by installing it on a personal computer or locally (e.g. an institute intranet). For this, pre-configured Docker containers and conda packages are available.

Galaxy HiCExplorer:

Docker :

docker run -p 8080:80 quay.io/bgruening/galaxy-hicexplorer

hicexplorer.usegalaxy.eu : On https://hicexplorer.usegalaxy.eu all HiCExplorer tools and workflows are installed. Use this option if you require high computational resources (e.g. large memory requirements).

HiCExplorer:

The HiCExplorer as a command line tool is available via *conda* or *BioContainers*.

Conda : *conda install hicexplorer -c bioconda*

BioContainer :

docker run quay.io/biocontainers/hicexplorer:latest

Training

Training and a documentation are crucial to enable as many scientists as possible to use and understand the Galaxy HiCExplorer. To introduce scientists who are new to Galaxy a guided tour through the Galaxy interface is provided as well as a tour to learn Hi-C data analysis. The tour content is available on the Galaxy Training Network (http://dx.doi.org/10.1101/225680) as well and includes example data hosted on Zenodo. All intermediate files are available in the shared data library of the Galaxy HiCExplorer.

For advanced users a detailed step-by-step tutorial for the analysis of Hi-C data from mouse embryonic stem-cells, as well as a comprehensive API documentation, is hosted at https://hicexplorer.readthedocs.org. The how-to describes how to set up the mapping of the reads. It suggests parameter settings for the creation of Hi-C contact matrices and describes the process of merging and threshold determination to remove poor bins prior to correction. The determination of TADs using the separation score is described in detail, including examples on visualization.

DISCUSSION

Galaxy HiCExplorer gives researchers the opportunity to run their Hi-C data analysis in a user-friendly, web browser based environment. The highly configurable framework provided by Galaxy makes this web server extendable to the various needs of researchers. Especially in conjunction with software for other high-throughput analysis protocols like RNA-seq or ChIP-seq, Galaxy HiCExplorer serves as a powerful basis for flexible explorative biomedical research in a high-throughput sequencing data analysis environment.

By combining all the necessary stages of pre-processing and visualization into a single tool, analysis not only becomes easier, but faster, highly reproducible, and more readily exchangeable. Biomedical researchers can focus their efforts on their data analysis without having to concern themselves with the particulars of managing various different software setups and configurations or learning to use command-line tools in an UNIX environment.

ACKNOWLEDGEMENTS

We thank the bioinformatics group at the University of Freiburg and the bioinformatics unit at the Max Planck Institute of Immunobiology and Epigenetics Freiburg.

FUNDING

German Research Foundation for the Collaborative Research Centre 992 Medical Epigenetics [SFB 992/1 2012 and SFB 992/2 2016 awarded to T.M. and R.B. and to Lutz Hein (in support of S.N. and R.G.) and for the DFG project GI 747/2-1 to R.G]; Federal Ministry of Education and Research through the German Epigenome Programme DEEP [01KU1216G awarded to T.M.]; German Federal Ministry of Education and Research [031 A538A de.NBI-RBC awarded to R.B.]; German Federal Ministry of Education and Research [031 L0101C de.NBI-epi awarded to B.G.]. Funding for open access charge: German Federal Ministry of Education and Research.
Conflict of interest statement. None declared.

REFERENCES

1. Lieberman-Aiden,E., Van Berkum,N.L., Williams,L., Imakaev,M., Ragoczy,T., Telling,A., Amit,I., Lajoie,B.R., Sabo,P.J., Dorschner,M.O. *et al.* (2009) Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science*, **326**, 289–293.
2. Dixon,J.R., Selvaraj,S., Yue,F., Kim,A., Li,Y., Shen,Y., Hu,M., Liu,J.S. and Ren,B. (2012) Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, **485**, 376–380.
3. Nora,E.P., Lajoie,B.R., Schulz,E.G., Giorgetti,L., Okamoto,I., Servant,N., Piolot,T., Van Berkum,N.L., Meisig,J., Sedat,J. *et al.* (2012) Spatial partitioning of the regulatory landscape of the X-inactivation centre. *Nature*, **485**, 381–385.
4. Rao,S.S.P., Huntley,M.H., Durand,N.C. and Stamenova,E.K. (2014) A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell*, **159**, 1665–1680.
5. Sanborn,A.L., Rao,S.S.P., Huang,S.-C., Durand,N.C., Huntley,M.H., Jewett,A.I., Bochkov,I.D., Chinnappan,D., Cutkosky,A., Li,J. *et al.* (2015) Chromatin extrusion explains key features of loop and domain formation in wild-type and engineered genomes. *Proc. Natl. Acad. Sci. U.S.A.*, **112**, E6456–E6465.
6. Bonev,B., Mendelson Cohen,N., Szabo,Q., Fritsch,L., Papadopoulos,G.L., Lubling,Y., Xu,X., Lv,X., Hugnot,J.P., Tanay,A. *et al.* (2017) Multiscale 3D genome rewiring during mouse neural development. *Cell*, **171**, 557–572.
7. Ron,G., Globerson,Y., Moran,D. and Kaplan,T. (2017) Promoter-enhancer interactions identified from Hi-C data using probabilistic models and hierarchical topological domains. *Nat. Commun.*, **8**, 2237.
8. Ramírez,F., Bhardwaj,V., Arrigoni,L., Lam,K.C., Grüning,B.A., Villaveces,J., Habermann,B., Akhtar,A. and Manke,T. (2018) High-resolution TADs reveal DNA sequences underlying genome organization in flies. *Nat. Commun.*, **9**, 189.
9. Babaei,S., Mahfouz,A., Hulsman,M., Lelieveldt,B.P., de Ridder,J. and Reinders,M. (2015) Hi-C chromatin interaction networks predict Co-expression in the mouse cortex. *PLoS Comput. Biol.*, **11**, e1004221.
10. Jiménez,R.C., Kuzak,M., Alhamdoosh,M., Barker,M., Batut,B., Borg,M., Capella-Gutierrez,S., Chue Hong,N., Cook,M., Corpas,M. *et al.* (2017) Four simple recommendations to encourage best practices in research software. *F1000Research*, **6**, 876.
11. Andrey,G., Schöpflin,R., Jerković,I., Heinrich,V., Ibrahim,D.M., Paliou,C., Hochradel,M., Timmermann,B., Haas,S., Vingron,M. *et al.* (2017) Characterization of hundreds of regulatory landscapes in developing limbs reveals two regimes of chromatin folding. *Genome Res.*, **27**, 223–233.
12. Ramírez,F., Lingg,T., Toscano,S., Lam,K.C., Georgiev,P., Chung,H.R., Lajoie,B.R., de Wit,E., Zhan,Y., de Laat,W. *et al.* (2015) High-affinity sites form an interaction network to facilitate spreading of the MSL complex across the X chromosome in Drosophila. *Mol. Cell*, **60**, 146–162.
13. Nothjunge,S., Nührenberg,T.G., Grüning,B.A., Doppler,S.A., Preissl,S., Schwaderer,M., Rommel,C., Krane,M., Hein,L. and Gilsbach,R. (2017) DNA methylation signatures follow preformed chromatin compartments in cardiac myocytes. *Nat. Commun.*, **8**, 1667.
14. Wingett,S., Ewels,P., Furlan-Magaril,M., Nagano,T., Schoenfelder,S., Fraser,P. and Andrews,S. (2015) HiCUP: pipeline for mapping and processing Hi-C data. *F1000Research*, **1310**, 1–12.
15. Heinz,S., Benner,C., Spann,N., Bertolino,E., Lin,Y.C., Laslo,P., Cheng,J.X., Murre,C., Singh,H. and Glass,C.K. (2010) Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. *Mol. Cell*, **38**, 576–589.
16. Serra,F., Baù,D., Goodstadt,M., Castillo,D., Filion,G. and Marti-Renom,M.A. (2017) Automatic analysis and 3D-modelling of Hi-C data using TADbit reveals structural features of the fly chromatin colors. *PLoS Comput. Biol.*, **13**, e1005665.
17. Schmid,M.W., Grob,S. and Grossniklaus,U. (2015) HiCdat: A fast and easy-to-use Hi-C data analysis tool. *BMC Bioinformatics*, **16**, 277.
18. Forcato,M., Nicoletti,C., Pal,K., Livi,C.M., Ferrari,F. and Bicciato,S. (2017) Comparison of computational methods for Hi-C data analysis. *Nat. Methods*, **14**, 679–685.
19. Servant,N., Varoquaux,N., Lajoie,B.R., Viara,E., Chen,C.J., Vert,J.P., Heard,E., Dekker,J. and Barillot,E. (2015) HiC-Pro: An optimized and flexible pipeline for Hi-C data processing. *Genome Biology*, **16**, 259.

20. Durand,N.C., Shamim,M.S., Machol,I., Rao,S.S., Huntley,M.H., Lander,E.S. and Aiden,E.L. (2016) Juicer provides a One-Click system for analyzing Loop-Resolution Hi-C experiments. *Cell Syst.*, **3**, 95–98.
21. Durand,N.C., Robinson,J.T., Shamim,M.S., Machol,I., Mesirov,J.P., Lander,E.S. and Aiden,E.L. (2016) Juicebox provides a visualization system for Hi-C contact maps with unlimited zoom. *Cell Syst.*, **3**, 99–101.
22. Sauria,M.E., Phillips-Cremins,J.E., Corces,V.G. and Taylor,J. (2015) HiFive: A tool suite for easy and efficient HiC and 5C data analysis. *Genome Biol.*, **16**, 237.
23. Abdennur,N., Goloborodko,A., Imakaev,M. and Mirny,L. (2017) *mirnylab/cooler v0.7.6*. zenodo.org.
24. Dekker,J., Belmont,A.S., Guttman,M., Leshyk,V.O., Lis,J.T., Lomvardas,S., Mirny,L.A., O'Shea,C.C., Park,P.J., Ren,B. *et al.* (2017) The 4D nucleome project. *Nature*, **549**, 219–226.
25. Li,H. (2013) Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arxiv.org*, [arXiv:1303.3997].
26. Langmead,B. and Salzberg,S.L. (2012) Fast gapped-read alignment with Bowtie 2. *Nat. Methods*, **9**, 357–359.
27. Ramírez,F., Ryan,D.P., Grüning,B., Bhardwaj,V., Kilpert,F., Richter,A.S., Heyne,S., Dündar,F. and Manke,T. (2016) deepTools2: a next generation web server for deep-sequencing data analysis. *Nucleic Acids Res.*, **44**, W160–W165.
28. Ewels,P., Magnusson,M., Lundin,S. and Käller,M. (2016) MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics*, **32**, 3047–3048.
29. Imakaev,M., Fudenberg,G., McCord,R.P., Naumova,N., Goloborodko,A., Lajoie,B.R., Dekker,J. and Mirny,L.A. (2012) Iterative correction of Hi-C data reveals hallmarks of chromosome organization. *Nat. Methods*, **9**, 999–1003.
30. Stevens,T.J., Lando,D., Basu,S., Atkinson,L.P., Cao,Y., Lee,S.F., Leeb,M., Wohlfahrt,K.J., Boucher,W., O'Shaughnessy-Kirwan,A. *et al.* (2017) 3D structures of individual mammalian genomes studied by single-cell Hi-C. *Nature*, **544**, 59–64.
31. Dixon,J.R., Jung,I., Selvaraj,S., Shen,Y., Antosiewicz-Bourget,J.E., Lee,A.Y., Ye,Z., Kim,A., Rajagopal,N., Xie,W. *et al.* (2015) Chromatin architecture reorganization during stem cell differentiation. *Nature*, **518**, 331–336.
32. Afgan,E., Baker,D., van den Beek,M., Blankenberg,D., Bouvier,D., Čech,M., Chilton,J., Clements,D., Coraor,N., Eberhard,C. *et al.* (2016) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Res.*, **44**, W3–W10.
33. da Veiga Leprevost,F., Grüning,B.A., Alves Aflitos,S., Röst,H.L., Uszkoreit,J., Barsnes,H., Vaudel,M., Moreno,P., Gatto,L., Weber,J. *et al.* (2017) BioContainers: an open-source and community-driven framework for software standardization. *Bioinformatics*, **33**, 2580–2582.

6

Galaxy HiCExplorer 3: a web server for reproducible Hi-C, capture Hi-C and single-cell Hi-C data analysis, quality control and visualization

Galaxy HiCExplorer 3: a web server for reproducible Hi-C, capture Hi-C and single-cell Hi-C data analysis, quality control and visualization
Joachim Wolff, Leily Rabbani, Ralf Gilsbach, Gautier Richard, Thomas Manke, Rolf Backofen, Björn Grüning
Nucleic Acids Research, Volume 48, Issue W1, 02 July 2020, Pages W177-W184
https://doi.org/10.1093/nar/gkaa220

Personal contribution

I contributed by conceiving and writing the manuscript and the presented Galaxy HiCExplorer 3 webserver. The following tools for Hi-C data analysis were designed and implemented by me: *hicQuickQC, hicConvertFormat, hicNormalize, hicDetectLoops, hicPlotSVL*. The module for capture Hi-C data was conceived, designed and implemented by me, including the tools: *chicQualityControl, chicViewpointBackground, chicViewpoint, chicSignificantInteractions, chicAggregateStatistic, chicDifferentialTest, chicPlotViewpoint*. The single-cell Hi-C software *scHiCExplorer* with the tools *scHicDemultiplex, scHicMergeToSCool, scHicQualityControl, scHicCreateBulkMatrix, scHicCluster, scHicClusterMinHash, scHicClusterSVL, scHicClusterCompartments, scHicConsensusMatrices, scHicPlotClusterProfiles, scHicPlotConsensusMatrices* was also conceived, designed and implemented by me. Furthermore, I contributed by updating the Galaxy Training Network tutorial, improving the general documentation for HiCExplorer and scHiCExplorer. In recognition of these significant contributions, I am listed as the first author for this publication.

Contribution of Leily Rabbani

Contributed by implementing features of HiCExplorer: hicCompartmentalization, KR correction algorithm, general contributions to the source code and discussions about Hi-C approaches; support of J.W. to write the manuscript.

Contribution of Ralf Gilsbach

Contributed with wet-lab data and requirements for the capture Hi-C modules and by reviewing the manuscript.

Contribution of Gautier Richard

Contributed by general contributions to the source code and discussions about Hi-C approaches. Design of the used graphics.

Contribution of Thomas Manke

Participating in writing and reviewing the manuscript.

Contribution of Rolf Backofen

General PhD supervision of Joachim Wolff and advice during the PhD process.

Contribution of Björn Grüning

General PhD supervision of Joachim Wolff and advice during the PhD process. Lab internal review of the manuscript.

Joachim Wolff

Joachim Wolff

The following co-authors confirm the above stated contribution:

Name	Date	Signature
Dr. Leily Rabbani	13.04.2021	L. Rabbani
Prof. Dr. Ralf Gilsbach	14.4.21	Prof. Dr. Ralf Gilsbach Institut für Kardiovaskuläre Physiologie FB Medizin | Goethe Universität Theodor-Stern-Kai 7 60590 Frankfurt am Main
Dr. Gautier Richard	13/04/2021	G Richard.
Dr. Thomas Manke	12.04.2021	
Prof. Dr. Rolf Backofen	22.04.2021	Digital unterschrieben von Prof. Dr. Rolf Backofen Datum: 2021.04.22 21:01:59 +02'00'
Dr. Björn Grüning	22.04.2021	B. Grüning

Published online 17 April 2020

Nucleic Acids Research, 2020, Vol. 48, Web Server issue ***W177–W184***
doi: 10.1093/nar/gkaa220

Galaxy HiCExplorer 3: a web server for reproducible Hi-C, capture Hi-C and single-cell Hi-C data analysis, quality control and visualization

Joachim Wolff [1,*], Leily Rabbani[2], Ralf Gilsbach [3,4,5], Gautier Richard [6], Thomas Manke [2], Rolf Backofen [1,7] and Björn A. Grüning [1]

[1]Bioinformatics Group, Department of Computer Science, University of Freiburg, Georges-Köhler-Allee 106, 79110 Freiburg, Germany, [2]Max Planck Institute of Immunobiology and Epigenetics, Stübeweg 51, 79108 Freiburg im Breisgau, Germany, [3]Institute for Cardiovascular Physiology, Goethe University, Frankfurt am Main, Germany, [4]German Center of Cardiovascular Research (DZHK), Partner site RheinMain, Frankfurt am Main, Germany, [5]Institute of Experimental and Clinical Pharmacology and Toxicology, Faculty of Medicine, University of Freiburg, Germany, [6]INRAE, Agrocampus Ouest, Université de Rennes, IGEPP, F-35650 Le Rheu, France and [7]Signalling Research Centres BIOSS and CIBSS, University of Freiburg, Schänzlestr. 18, 79104 Freiburg, Germany

Received February 04, 2020; Revised March 11, 2020; Editorial Decision March 20, 2020; Accepted March 24, 2020

ABSTRACT

The Galaxy HiCExplorer provides a web service at https://hicexplorer.usegalaxy.eu. It enables the integrative analysis of chromosome conformation by providing tools and computational resources to pre-process, analyse and visualize Hi-C, Capture Hi-C (cHi-C) and single-cell Hi-C (scHi-C) data. Since the last publication, Galaxy HiCExplorer has been expanded considerably with new tools to facilitate the analysis of cHi-C and to provide an in-depth analysis of Hi-C data. Moreover, it supports the analysis of scHi-C data by offering a broad range of tools. With the help of the standard graphical user interface of Galaxy, presented workflows, extensive documentation and tutorials, novices as well as Hi-C experts are supported in their Hi-C data analysis with Galaxy HiCExplorer.

INTRODUCTION

Chromosome conformation capture (3C) (1) and its successors 4C (2,3), 5C (4) and Hi-C (5) have developed into the standard technologies used in studying the 3D conformation of chromatin. They can provide insights into the processes involved in chromatin folding and gene regulation. Hi-C technology is a well established method to study genome wide interaction of data and can detect large-scale chromosome structures, such as active and inactive (A/B) compartments (5,6), topological associated domains (TADs) (7,8), chromatin loop structures (9) or ratios of short to long range interaction counts. Although Hi-C is a powerful approach for studying the 3D structure of chromatin globally, it is limited in its ability to investigate location specific interactions, such as promoter-enhancer interactions, due to the need for high coverage and sequencing costs. Moreover, Hi-C is unable to capture protein-DNA interactions in the chromatin conformation context. To overcome these shortcomings, capture Hi-C (cHi-C) techniques have been developed. These assays are generating data, which are enriched for the predefined targets, such as promoter regions (Promoter cHi-C) (10), proteins or protein modifications (HiChIP) (11); HiChIP is able to capture chimeric protein-DNA interactions, including transcription factors or histone modifications. The location specific enrichment provides a significantly better signal-to-noise ratio and can therefore be used for a more location sensitive analysis. Capture Hi-C data cannot be analysed with established Hi-C algorithms and need their own tools. With the rise of single-cell sequencing technologies, the single-cell Hi-C (scHi-C) approach has been developed to allow for a deeper insight into the chromatin conformation dynamics between cell types, for instance during the cell cycle (12). For a review on the abilities and current developments of Hi-C and related techniques, the reviews of McCord *et al.* (13), Kempfer and Pombo (14) or Bonev and Cavalli (15) are recommended. The scHi-C analyses are much more resource intensive than Hi-C analyses and need specialized algorithms for dimension reduction. Galaxy HiCExplorer meets these requirements by providing efficient and easy to use tools for the analysis of Hi-C, cHi-C and scHi-C through a comprehensive and unified web server accessible at https://hicexplorer.usegalaxy.eu. It provides computational capabilities for even the most demanding analyses. Additionally, Galaxy HiCExplorer is easy to deploy locally

[*]To whom correspondence should be addressed. Tel: +49 761 2037460; Email: wolffj@informatik.uni-freiburg.de

thanks to the installer for a local Galaxy instance. Moreover, a command line version is provided by conda and is available via the bioconda channel (16).

RELATED WORKS

Galaxy HiCExplorer is designed as an easy-to-use online service which is accessible through a web browser. Thus, no installation is required. By embedding it into Galaxy (17) and the https://usegalaxy.eu environment, it facilitates reproducible, shareable research as well as easily accessible data analysis. With Galaxy HiCExplorer, researchers can focus on their data analysis without facing any computational limitation or software dependency issue. To offer more flexibility, it is also possible to install Galaxy HiCExplorer on a local Galaxy instance. Hi-C data processing and downstream analysis are supported by many tool suites, such as Juicer (18), HiCUP (19), HOMER (20), HiC-Pro (21), HiFive (22) and the recently published HiCeekR (23). Juicer, HiC-Pro and HiCeekR offer several tools but are limited to a local installation. HiFive offers a Galaxy integration, but lacks the support of external data formats like *cool* file format (24). HiCUP and HOMER support only certain parts of Hi-C data analysis. Among the above tools, HiC-Pro is the only one with the ability to analyse cHi-C and HiChIP data. scHiCNorm (25) and scHiCluster(26) provide support for single-cell Hi-C data normalization and clustering, but suffer from the lack of a tool suite to guide researchers through the workflow of processing single-cell Hi-C data from the raw FASTQ files to the clustering of cells, including methods for building interaction matrices, quality control, dimension reduction and visualization. scHiC-Norm and scHiCluster use text files to store the scHi-C interaction matrices, which are particularly space consuming, not easily shareable and prone to error accumulation. Galaxy HiCExplorer addresses all these shortcomings by providing a tool suite to support the analysis of Hi-C, captured Hi-C (e.g. Promoter cHi-C, HiChIP) and single-cell Hi-C data from the raw input data to publication ready results, as shown in Figure 2. Most importantly, none of the mentioned tools provide large computational resources to support Hi-C, cHi-C and single-cell Hi-C data analysis.

GALAXY HICEXPLORER

Galaxy HiCExplorer offers a large collection of tools to pre-process, analyse and visualize Hi-C, cHi-C and scHi-C data. In addition to its assay-specific modules, users can benefit from the external pre-processing software for quality control of raw data and mappers such as BWA-MEM or Bowtie2 which are provided on the https://hicexplorer.usegalaxy.eu web server as well as the computational resources available. Moreover, for interactive Hi-C matrix exploration we have recently integrated HiGlass (27) into Galaxy. In the following, we briefly describe the new modules which have been added since our original publications on HiCExplorer 1 (28) and 2 (29).

HiCExplorer

HiCExplorer provides a variety of tools for a complete Hi-C data analysis, starting with tools to control the quality of data to create, adjust, normalize and correct interaction matrices. Furthermore, it provides tools for downstream analysis of Hi-C data such as identification of A/B compartments, TADs, loops or the computation of short versus long range contact ratios per chromosome. Finally, HiCExplorer has many options available for data visualisation such as plotting the interaction matrices, visualization of the detected TADs with pyGenomeTracks or creating aggregated contacts images. The workflow of Hi-C data analysis with Galaxy HiCExplorer is shown in Figure 1A. MultiQC, as shown in Figure 1A, supports HiCExplorer. If the structure of the quality report is changed, an update for MultiQC is necessary and the non-updated MultiQC might not work with the most recent quality report version.

Pre-processing.

hicQuickQC. The creation of Hi-C interaction matrices, as well as the investigation of the quality of the data afterwards, may require a long processing time and is also resource intensive. To get a swift insight into the quality of Hi-C data, hicQuickQC has been introduced. It computes a quick summary of the Hi-C data quality using only a small subset of reads. The computation time to create the quality report with hicQuickQC for the first 1 million reads takes <3 min. The quality report is equal to the quality report of hicBuildMatrix and the only difference is that it is based only on the first 1 million reads instead of the full dataset.

hicFindRestSites. Hi-C interaction matrices with fixed size bins are not always the best representation of the data. In fact, with a sufficient sequencing depth, bins of a restriction fragment size are a better alternative. To generate such matrices, this tool generates a list of restriction sites for user-defined enzymes. This list can be used as an input to hicBuildMatrix to create restriction site resolution Hi-C matrices.

hicConvertFormat. Support for external interaction matrix data formats is missing in most Hi-C data analysis software. This makes it difficult to compare matrices which have been built with different software and to directly use them for further analysis. Instead, the matrices need to be built from scratch, which is time consuming and potentially error prone. This tool supports loading matrices of *cool*, HiCExplorers *h5*, Juicers *hic*, *Homer* and *HiCPro* format and can convert them to *cool*, *h5*, *Homer* and *ginteractions* (30) format.

hicNormalize. Normalization is a crucial step to be able to compare the interaction matrices obtained with a different sequencing depth. For this purpose, hicNormalize supports three normalization methods: (a) to the depth of the matrix with the least read coverage, (b) to the value range of 0 to 1 and (c) to a user defined scaling factor. For details on the normalization methods consult our Supplementary materials.

hicCorrectMatrix. Correcting the Hi-C interaction matrices is a necessary step to remove technical biases. In addition to the iterative correction (ICE) algorithm from

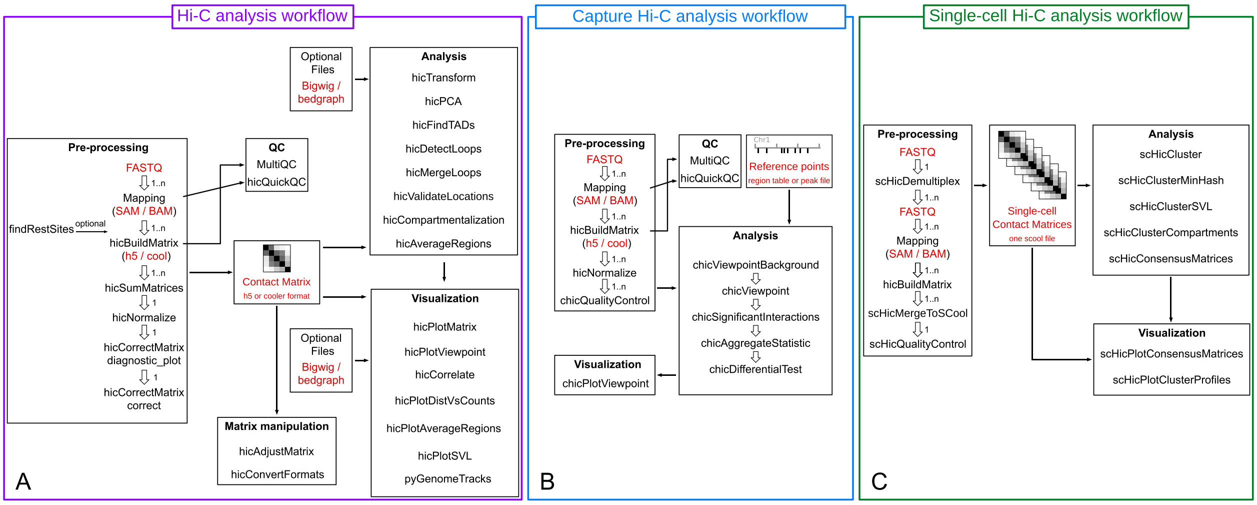


Figure 1. Analysis workflow for Hi-C (**A**), cHi-C (**B**) and scHi-C (**C**). All the workflows use the *hicBuildMatrix* to create the individual contact matrices. Hi-C and cHi-C supports HiCExplorer's h5 and cool interaction matrix file format; however, scHi-C pipeline creates one cool file per cell. These files can then be merged into a single multi-cool (scool) matrix with *scHicMergeToSCool*.

Imakaev (31), HiCExplorer also offers the Knight-Ruiz correction (32), first used for Hi-C matrices by (9). The method is more memory efficient, is faster than the ICE algorithm and better suited for the analysis of high-resolution and deep read coverage interaction matrices.

Analysis.

hicDetectLoops. Chromatin loops are long range chromatin interactions and present in Hi-C matrices as enriched regions in comparison to their local neighborhood. Depending on the read coverage and the resolution of the Hi-C interaction matrix, it is for instance possible to detect enhancer–promoter interactions. Due to its sensitivity to the read coverage it is recommended to run the loop detection on different resolutions and to merge them afterwards, using *hicMergeLoops*, into one loop file. By merging, overlapping loops are pooled into one loop. In addition, the tool *hicValidateLocations* can be used to confirm that the detected loops are correlated with detected locations of a protein of interest. For example, CTCF is known as a loop binding factor in mammals (7,9) and should therefore be present at many loop locations. Finally, the detected loops can be visualised with *hicPlotMatrix*, see Figure 2A. For details regarding the algorithm and benchmarks, consider (33).

hicCompartmentalization. This tool supports the analysis of interactions at the level of (active and inactive) compartments. These two large chromosomal domains can be defined through a principal component analysis (5) and are provided in Galaxy HiCExplorer by the existing *hicPCA* module. To visualize the difference in the interaction frequencies within and between the different compartments, a polarization plot can be generated using a method which was first introduced by (6). See Figure 2D.

hicAverageRegions. The comparison of specific regions between different samples can pose a challenge. One typical use case could be the comparison between multiple detected TADs on a wild type and a treatment sample. This tool extracts Hi-C submatrices corresponding to the upstream and downstream regions of reference anchors (e.g. a subset of TAD boundaries, promoter regions or any predefined positions of interest). It computes the average contacts of these submatrices and uses them to detect the potential differences of contact patterns located around these anchors, see Supplementary materials. The average of collected submatrices can be visualized with *hicPlotAverageRegions*, as shown in Figure 2C.

hicPlotSVL. Comparing the ratio of short range interaction to long range interaction between Hi-C matrices obtained in various experimental conditions can guide the understanding of chromatin topology and its folding principles. To this end, this tool computes the ratio per chromosome and plots it per sample as a boxplot, as shown in Figure 2B. For the mathematical details, please consult our Supplementary material.

pyGenomeTracks. The visualization tool *hicPlotTADs* which, was mentioned in the previous publication (29), came to the attention of many of our users. However, there was always some confusion as to whether or not it is for Hi-C data only which was never the case. To solve this, hicPlotTADs was renamed to pyGenomeTracks and is independently developed.

Capture Hi-C

The cHi-C modules of HiCExplorer are designed for analysing Promoter cHi-C and HiChIP. HiCExplorer will also accept data from other Capture Hi-C methods, including ChiA-PET (34). if dedicated preprocessing steps were performed to obtain compatible mapping data. Furthermore, it can be used to generate virtual 4C plots from Hi-C data. As for Hi-C data, cHi-C interaction matrices are

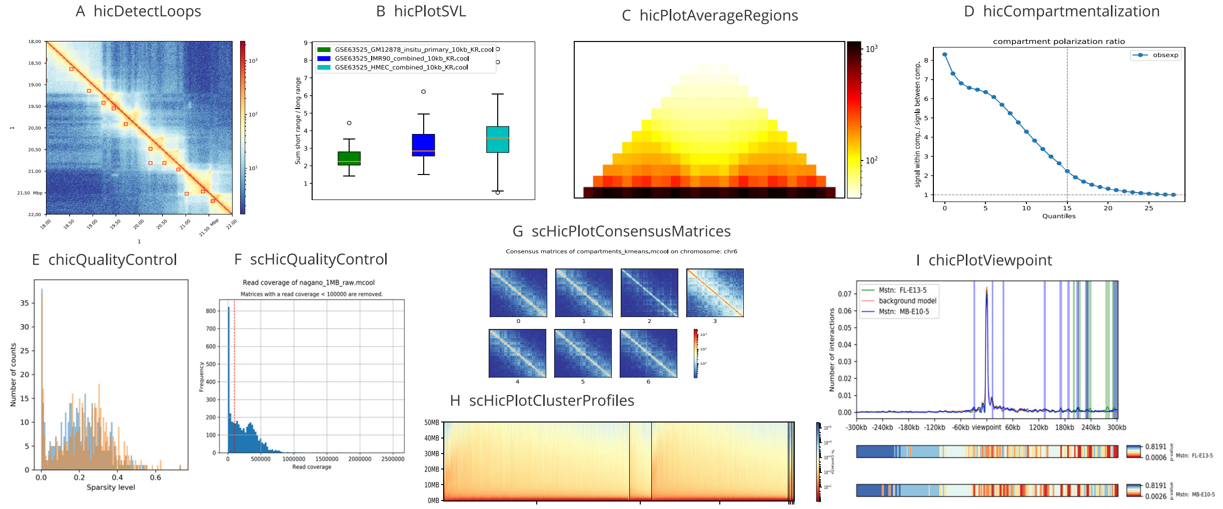


Figure 2. (**A**) Detected loops on *GM12878 primary* data from (9), computed by *hicDetectLoops* and visualised by *hicPlotMatrix*. (**B**) Short to long range contact interaction ratios created by *hicPlotSVL* on *GM12878 primary*, *IMR90* and *HMEC* data from (9). (**C**) Average regions of detected TADs from *hicFindTADs* on *GM12878 primary*, chromosome 1; data from (9). (**D**) The level of compartments separation on *GM12878 primary* data from (9), computed by *hicCompartmentalization*. (**E**) Quality control plot for *FL-E13-5* and *MB-E10-5* showing the sparsity distribution, data from (42). (**F**) Quality control plot for single-cell Hi-C data by (36). It shows the read coverage per cell, cells with <100 000 reads are discarded. (**G**) Consensus matrix plot for single-cell Hi-C data on 1 Mb resolution. Cells are dimension reduced by computing A/B compartments per cell and clustered with k-means. The consensus matrix of a cluster is the average of all interaction matrices of the cluster members. Data from (36). (**H**) Single-cell Hi-C cluster profile, created after dimension reduction by *scHicClusterMinHash* and spectral clustering on 1 Mb single-cell Hi-C data from (36). (**I**) Viewpoint of the gene *MSTN* on *FL-E13-5* and *MB-E10-5* with mean background and p-values per relative distance via continuous negative binomial distributions, data from (42).

built with hicBuildMatrix. The regions of interest in these protocols, such as the location of the promoters in cHi-C or the binding sites of the target protein for HiChIP, are referred to as *reference points*. In the case of HiChIP, reference points are either annotated with peak calling tools, such as MACS2 (35) using either the HiChIP mapping file or ChIP-seq data, or regions (e.g. promoters) are manually selected. The region defined up- and downstream of a reference point is referred to as a *viewpoint*. Figure 2I illustrates all the up- and downstream distances within a viewpoint by their *relative distance* to a specific reference point. A background model is created which takes interactions per relative distance from all viewpoints into account. It is in the downstream analysis used to detect higher interactions as expected for a relative distance. These interactions are potentially different between a treatment and a control sample and therefore can be used for a differential test. The cHi-C workflow of Galaxy HiCExplorer is shown in Figure 1B. Please consult our Supplementary material concerning details of the presented cHi-C methods.

Pre-processing.

chicQualityControl. This module is designed to investigate the quality of every single viewpoint, taking the sparsity of the interaction counts into account. A viewpoint will be removed if the sparsity of the data at this viewpoint is lower than a given threshold. To help users in setting an appropriate threshold, the tool generates several quality plots from which one is presented in Figure 2E.

chicViewpointBackground. The background model per relative distance is computed by taking all interaction counts of a relative distance over all viewpoints and samples into account. Based on this model, interactions with higher counts than an expected count will be identified during the downstream analysis.

Analysis.

chicViewpoint. This tool extracts the interaction counts of each viewpoint from the interaction matrix, associates additional information and writes the viewpoint data to a file. Based on the background model, a *P*-value for each interaction count is computed. The *P*-value is an indicator if a specific count at a relative distance is in an expected range or higher.

chicSignificantInteractions. Using the *P*-values of a viewpoint, this tool decides via a threshold if an interaction at a relative distance is significant.

chicAggregateStatistic. The differential testing investigates if solitary interactions of two viewpoints have a different interaction frequency. These solitary interactions are either provided by a predefined target file or detected with *chicSignificantInteractions*. This tool aggregates the provided interactions from two viewpoints and prepares them as input for *chicDifferentialTest*.

chicDifferentialTest. The differential testing examines one solitary interaction between two viewpoints, under consideration of the interaction frequency at the reference points. As a differential test either chi^2-test or Fisher's test can be used under the null hypothesis that the interaction frequency is equal.

Visualization.

chicPlotViewpoint. To visualize one or several viewpoints, *chicPlotViewpoint* has been introduced with the possibility of adding a mean background signal and highlighting the significant or differential interactions. Moreover, the computed p-values can be added as an additional heatmap as seen as in Figure 2I.

Single-cell Hi-C

Single-cell Hi-C explores how chromatin is being folded and which elements contribute to its regulation on a single-cell scale. While analyzing Hi-C data is computationally expensive, this can increase drastically for scHi-C data. The reason for this is the increase in the number of Hi-C interaction matrices that need to be analysed from one to several thousand, with a corresponding increase in runtime and memory. The read coverage of scHi-C data is currently not high (36) and 1 megabase (Mb) resolution matrices are used to avoid generating highly sparse matrices. However, as sequencing costs decline, resolutions of 10 kb may be achievable and the demand for dimension reduction techniques, such as those presented here, will be indispensable. With scHiCExplorer, a software suite is provided to process single-cell Hi-C data offering tools for demultiplexing, matrix handling, correction, dimension reduction, clustering and visualisation. Figure 1C shows the workflow of single-cell Hi-C data analysis with Galaxy HiCExplorer. scHiCExplorer can be used for general processing of single-cell Hi-C data as long as the forward and reverse strand for each cell are provided as a BAM/SAM file. All pre-processing steps like adapter and/or barcode trimming, demultiplexing and mapping needs to be applied by third-party tools.

Pre-processing.

scHicDemultiplex. Raw FASTQ data from a single-cell experiment usually contains reads from multiple cells which are encoded with different barcodes. This tool supports demultiplexing of an interleaved FASTQ file into one FASTQ file per cell. The demultiplexing is implemented to support the method which has been introduced in Nagano (36) for barcoding. Due to the lack of a standard method on how to encode barcodes, presently, demultiplexing is limited to FASTQ files with the same barcoding method as in (36). Other demultiplexing tools are part of the general Galaxy tool suite.

scHicMergeToSCool. Every single-cell interaction matrix can be created with *hicBuildMatrix*. *scHicMergeToSCool* can merge individual matrices into a joint matrix in multi-cool format (24), which will be used in all subsequent downstream analysis and visualization tools. While using the API of cooler, the data is not stored with multiple resolutions as it is defined by (24). The cool file is used as a container format for the individual cool files of the Hi-C matrices. For this reason, the format is referred to as *scool*.

scHicQualityControl. Since scHi-C data is a very sparse, not all matrices have sufficient read coverage to be considered for the downstream analysis. Thus, the quality control module removes interaction matrices of cells with total read counts below a user-specified threshold (see Figure 2E) or very sparse interaction matrices.

scHicCreateBulkMatrix. This tool supports to pool all matrices stored in the *scool* file to one single Hi-C interaction matrix and enables the analysis like in regular Hi-C.

Several modules of HiCExplorer are also required in single-cell Hi-C data analysis. To provide an equal functionality at the single cell level and to support the scool file format, scHiCExplorer reuses these modules from HiCExplorer. These are *scHicNormalize*, *scHicCorrectMatrices*, *scHicAdjustMatrix*, *scHicMergeMatrixBins* and *scHicInfo*. scHiCExplorer adds the functionality of handling the multiple matrices stored in the scool file and distributes the computations over several threads.

Dimension and clustering reduction. Clustering cells is a common approach to study the difference between them and to learn about their relations from single cell data. scHiCExplorer provides the *k-means* and *spectral* clustering methods. K-means was used on scHi-C data by (36) or (26), but the choice of a clustering algorithm is always dependent on the data. For this reason, scHiCExplorer provides additional the spectral clustering and will continue adding standard clustering algorithms in the future. However, reducing the dimensions of the underlying matrices is necessary to be able to cluster cells in a reasonable amount of time and to decrease the memory footprint; as shown in Supplementary Table S1. The usage of dimension reduction is also often necessary to achieve good results (37–39). The results in the Supplementary Figures S1–S4 confirm this. The need to reduce the dimensions becomes obvious when matrices of higher resolutions are used. The combined raw data matrix for a scHi-C dataset has a dimensionality of *cells*features*, where $features = bins*bins$ for one matrix. As an example, mapping of the Nagano 2017 (36) data on the mouse mm9 genome and using it to make a 1 Mb resolution matrix, will already return a matrix of 2500*7.3 million dimensions; this number will increase to 2500*7.3 billion dimensions if the resolution of the matrix increases to 10 kb.

scHicCluster. A principal component analysis (reducing to *samples*bins*) or a *k*-nearest neighbors matrix (reducing to *samples*samples*) can be chosen as the desired method to reduce the dimensions of data. However, a clustering of the raw data without applying any dimension reduction is also supported.

scHicClusterMinHash. Clustering and dimension reduction techniques of *scHicCluster* usually work with low resolutions like 1 Mb but require a large amount of memory (>1 TB) on matrices of higher resolutions such as 10 kb. MinHash (40) is an approximate nearest neighbors method which computes the *k*-nearest neighbors matrix via local sensitive hash functions and reduces the number of dimensions to *samples*samples*. MinHash's approximate computation of the *k*-nearest neighbors makes it possible to process 10 kb resolution scHi-C data. Our implementation runs for just over one hour and needs 53GB of memory, for more details consider (41).

scHicClusterSVL. This dimension reduction method computes the ratio of short range and long range contacts per chromosome and reduces the dimensions of the matrix to *samples*chromosomes*.

scHicClusterCompartments. This method computes the A/B compartments of each cell and clusters cells based on their compartments. It reduces the matrix dimensions to *samples*bins*.

Visualization. Due to the high dimensionality of matrices per cell (*bins*bins*), a satisfactory visual representation of single-cell Hi-C data clustering is difficult to achieve. Traditional methods represent the data in a two dimensional space; however, decreasing dimensionality from a few million (e.g. a 1 Mb resolution matrix) or billion (e.g. a 10 kb resolution matrix) to two dimensions will create a non-meaningful representation. scHiCExplorer offers two alternative representations of cells' clusters: Per cluster (a) a consensus matrix of all cells is plotted or (b) each cell of a cluster is visualized with its decreasing contact frequency by increasing the distance from the main diagonal.

scHicConsensusMatrices. Using the results of the clustering, this tool merges all matrices of one cluster into a single interaction matrix and normalizes the resulting consensus matrices to the same read coverage. This matrix can be visualized as the consensus matrix of a cluster by *scHicPlotConsensusMatrices* and reveals the clustering power in separation of the cells based on their chromatin density. See Figure 2G.

scHicPlotClusterProfiles. A cluster profile shows the decrease of contact frequencies per cell from the main diagonal to 50 Mb distance from it. A good clustering is achieved if the decreasing of contact frequency is similar for all cells of a cluster and if the profiles of various clusters differ. Figure 2H shows the different cells grouped by clusters on the x-axis and the decreasing contact frequency by an increasing distance from the main diagonal on the y-axis.

IMPLEMENTATION

Galaxy HiCExplorer is implemented as a collection of Galaxy tool wrappers and is available on the Galaxy ToolShed. The Galaxy integration is provided for HiCExplorer as well as scHiCExplorer. HiCExplorer and scHiCExplorer are both implemented in Python 3.6 and are available on Bioconda (16). The Knight-Ruiz correction and the MinHash approximate *k*-nearest neighbors for the dimension reduction are implemented in C++ and are also available on Bioconda.

USING HICEXPLORER

Installation and usage

Galaxy HiCExplorer can be used as a web server and is accessible via https://hicexplorer.usegalaxy.eu. All presented tools are publicly available and may be used without any required registration. Unregistered users are provided with 11 GB storage space, while registered users are granted 250GB. Registered users have the opportunity to apply for more storage. Users are strongly encouraged to use https://hicexplorer.usegalaxy.eu web server if high compute resources are required. Galaxy HiCExplorer is GDPR compliant; deleted datasets will be permanently removed within 14 days and the data of unregistered users is deleted after an inactivity of 90 days.

TRAINING

To support researchers in their analysis of Hi-C, cHi-C or scHi-C data, tutorials and a detailed documentation are available on https://hicexplorer.readthedocs.io and https://schicexplorer.readthedocs.io. As presented in (29), the guided tours for novice users of Galaxy as well as the Galaxy HiCExplorer specific tutorial are available on the Galaxy Training Network (43). The cHi-C tutorial uses Promoter cHi-C example data to guide users through the complete analysis workflow starting from building a cHi-C contact matrix, creating a background model, detecting significant and differential interactions to a plotting of the viewpoints. The tutorial of the single-cell data explains the barcoding, the mapping, creation and merging of scHi-C matrices. Moreover it shows different clustering techniques including the dimension reduction and the visual representation of the clustering.

DISCUSSION

The presented web server on https://hicexplorer.usegalaxy.eu gives researchers the opportunity to focus on their data analysis in a user friendly, reproducible and computationally powerful environment. With the deep integration of HiCExplorer into the Galaxy environment, users are now able to combine their Hi-C, cHi-C (Promoter cHi-C, HiChIP) or scHi-C data with their data from other high-throughput assays like ChIP-Seq or RNA-Seq and run multi-omics analyses, all within their web browser. Galaxy HiCExplorer is suited for both experts and newcomers to the Hi-C field, thanks to the provided tutorials that give all users a clear introduction on how to use HiCExplorer for their data analyses. Moreover, the tools recently added to HiCExplorer offer the possibility to resolve the dynamic chromatin topology inherent to different cell types provided by scHi-C. The automated management of a large number of cells in the scHi-C pipeline will help researchers to decipher the principles of chromatin folding in the context of cell cycle and cell type specificity. Moreover, the new tools of Galaxy HiCExplorer are able to analyse precise interactions between regulatory regions and their target genes assisted by cHi-C techniques. This expansion of Galaxy HiCExplorer allows for a better understanding of how 3D structure of a genome may affect an organism's phenotype.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We thank Simon Bray and Anup Kumar of the Bioinformatics group at the Albert-Ludwigs-Universität of Freiburg im Breisgau, Germany and Simon Rapple for proof reading the manuscript.

FUNDING

German Science Foundation [CRC992 'Medical Epigenetics' to L.R., T.M., R.B., B.G.; GI 747/2-1, GI 747/3-1 to R.G.]; German Federal Ministry of Education and Research [031 A538A de.NBI-RBC awarded to R.B.]; German Federal Ministry of Education and Research [031 L0101C de.NBI-epi awarded to B.G.]; R.B. was supported by the German Research Foundation (DFG) under Germany's Excellence Strategy [CIBSS-EXC-2189-Project ID 390939984]. Funding for open access charge: German Federal Ministry of Education and Research.

Conflict of interest statement. None declared.

REFERENCES

1. Dekker,J., Rippe,K., Dekker,M. and Kleckner,N. (2002) Capturing chromosome conformation. *Science*, **295**, 1306–1311.
2. Simonis,M., Klous,P., Splinter,E., Moshkin,Y., Willemsen,R., De Wit,E., Van Steensel,B. and De Laat,W. (2006) Nuclear organization of active and inactive chromatin domains uncovered by chromosome conformation capture–on-chip (4C). *Nat. Genet.*, **38**, 1348.
3. Zhao,Z., Tavoosidana,G., Sjölinder,M., Göndör,A., Mariano,P., Wang,S., Kanduri,C., Lezcano,M., Sandhu,K. S., Singh,U. *et al.* (2006) Circular chromosome conformation capture (4C) uncovers extensive networks of epigenetically regulated intra-and interchromosomal interactions. *Nat. Genet.*, **38**, 1341.
4. Dostie,J., Richmond,T.A., Arnaout,R.A., Selzer,R.R., Lee,W.L., Honan,T.A., Rubio,E.D., Krumm,A., Lamb,J., Nusbaum,C. *et al.* (2006) Chromosome Conformation Capture Carbon Copy (5C): a massively parallel solution for mapping interactions between genomic elements. *Genome Res.*, **16**, 1299–1309.
5. Lieberman-Aiden,E., Van Berkum,N.L., Williams,L., Imakaev,M., Ragoczy,T., Telling,A., Amit,I., Lajoie,B.R., Sabo,P.J., Dorschner,M.O. *et al.* (2009) Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science*, **326**, 289–293.
6. Schwarzer,W., Abdennur,N., Goloborodko,A., Pekowska,A., Fudenberg,G., Loe-Mie,Y., Fonseca,N.A., Huber,W., Haering,C.H., Mirny,L. *et al.* (2017) Two independent modes of chromatin organization revealed by cohesin removal. *Nature*, **551**, 51.
7. Dixon,J.R., Selvaraj,S., Yue,F., Kim,A., Li,Y., Shen,Y., Hu,M., Liu,J.S. and Ren,B. (2012) Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, **485**, 376–380.
8. Nora,E.P., Lajoie,B.R., Schulz,E.G., Giorgetti,L., Okamoto,I., Servant,N., Piolot,T., Van Berkum,N.L., Meisig,J., Sedat,J. *et al.* (2012) Spatial partitioning of the regulatory landscape of the X-inactivation centre. *Nature*, **485**, 381–385.
9. Rao,S. S.P., Huntley,M.H., Durand,N.C. and Stamenova,E.K. (2014) A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell*, **159**, 1665–1680.
10. Dryden,N.H., Broome,L.R., Dudbridge,F., Johnson,N., Orr,N., Schoenfelder,S., Nagano,T., Andrews,S., Wingett,S., Kozarewa,I. *et al.* (2014) Unbiased analysis of potential targets of breast cancer susceptibility loci by Capture Hi-C. *Genome Res.*, **24**, 1854–1868.
11. Mumbach,M.R., Rubin,A.J., Flynn,R.A., Dai,C., Khavari,P.A., Greenleaf,W.J. and Chang,H.Y. (2016) HiChIP: efficient and sensitive analysis of protein-directed genome architecture. *Nat. Methods*, **13**, 919.
12. Nagano,T., Lubling,Y., Stevens,T.J., Schoenfelder,S., Yaffe,E., Dean,W., Laue,E.D., Tanay,A. and Fraser,P. (2013) Single-cell Hi-C reveals cell-to-cell variability in chromosome structure. *Nature*, **502**, 59.
13. McCord,R.P., Kaplan,N. and Giorgetti,L. (2020) Chromosome conformation capture and beyond: toward an integrative view of chromosome structure and function. *Mol. Cell*, **77**, 688–708.
14. Kempfer,R. and Pombo,A. (2019) Methods for mapping 3D chromosome architecture. *Nat. Rev. Genet.* **21**, 207–226.
15. Bonev,B. and Cavalli,G. (2016) Organization and function of the 3D genome. *Nat. Rev. Genet.*, **17**, 661.
16. Grüning,B., Dale,R., Sjödin,A., Chapman,B.A., Rowe,J., Tomkins-Tinch,C.H., Valieris,R. and Köster,J. (2018) Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nat. Methods*, **15**, 475.
17. Afgan,E., Baker,D., van den Beek,M., Blankenberg,D., Bouvier,D., Čech,M., Chilton,J., Clements,D., Coraor,N., Eberhard,C. *et al.* (2016) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Res.*, **44**, W3–W10.
18. Durand,N.C., Shamim,M.S., Machol,I., Rao,S.S., Huntley,M.H., Lander,E.S. and Aiden,E.L. (2016) Juicer provides a one-click system for analyzing Loop-Resolution Hi-C experiments. *Cell Syst.*, **3**, 95–98.
19. Wingett,S., Ewels,P., Furlan-Magaril,M., Nagano,T., Schoenfelder,S., Fraser,P. and Andrews,S. (2015) HiCUP: pipeline for mapping and processing Hi-C data. *F1000Research*, **4**, 1310.
20. Heinz,S., Benner,C., Spann,N., Bertolino,E., Lin,Y.C., Laslo,P., Cheng,J.X., Murre,C., Singh,H. and Glass,C.K. (2010) Simple combinations of Lineage-Determining transcription factors prime cis-Regulatory elements required for macrophage and B cell identities. *Mol. Cell*, **38**, 576–589.
21. Servant,N., Varoquaux,N., Lajoie,B.R., Viara,E., Chen,C.J., Vert,J.P., Heard,E., Dekker,J. and Barillot,E. (2015) HiC-Pro: an optimized and flexible pipeline for Hi-C data processing. *Genome Biol.*, **16**, 259.
22. Sauria,M.E., Phillips-Cremins,J.E., Corces,V.G. and Taylor,J. (2015) HiFive: a tool suite for easy and efficient HiC and 5C data analysis. *Genome Biol.*, **16**, 237.
23. Di Filippo,L., Righelli,D., Gagliardi,M., Matarazzo,M.R. and Angelini,C. (2019) HiCeekR: a novel Shiny app for Hi-C data analysis. *Front. Genet.*, **10**, 1079.
24. Abdennur,N. and Mirny,L.A. (2019) Cooler: scalable storage for Hi-C data and other genomically labeled arrays. *Bioinformatics*, **36**, 311–316.
25. Liu,T. and Wang,Z. (2018) scHiCNorm: a software package to eliminate systematic biases in single-cell Hi-C data. *Bioinformatics*, **34**, 1046–1047.
26. Zhou,J., Ma,J., Chen,Y., Cheng,C., Bao,B., Peng,J., Sejnowski,T.J., Dixon,J.R. and Ecker,J.R. (2019) Robust single-cell Hi-C clustering by convolution-and random-walk–based imputation. *Proc. Natl. Acad. Sci. U.S.A.*, **116**, 14011–14018.
27. Kerpedjiev,P., Abdennur,N., Lekschas,F., McCallum,C., Dinkla,K., Strobelt,H., Luber,J.M., Ouellette,S.B., Azhir,A., Kumar,N. *et al.* (2018) HiGlass: web-based visual exploration and analysis of genome interaction maps. *Genome Biol.*, **19**, 125.
28. Ramírez,F., Bhardwaj,V., Arrigoni,L., Lam,K.C., Grüning,B.A., Villaveces,J., Habermann,B., Akhtar,A. and Manke,T. (2018) High-resolution TADs reveal DNA sequences underlying genome organization in flies. *Nat. Commun.*, **9**, 189.
29. Wolff,J., Bhardwaj,V., Nothjunge,S., Richard,G., Renschler,G., Gilsbach,R., Manke,T., Backofen,R., Ramírez,F. and Grüning,B.A. (2018) Galaxy HiCExplorer: a web server for reproducible Hi-C data analysis, quality control and visualization. *Nucleic Acids Res.*, **46**, W11–W16.
30. Lun,A.T., Perry,M. and Ing-Simmons,E. (2016) Infrastructure for genomic interactions: bioconductor classes for Hi-C, ChIA-PET and related experiments. *F1000Research*, **5**, 950.
31. Imakaev,M., Fudenberg,G., McCord,R.P., Naumova,N., Goloborodko,A., Lajoie,B.R., Dekker,J. and Mirny,L.A. (2012) Iterative correction of Hi-C data reveals hallmarks of chromosome organization. *Nat. Methods*, **9**, 999–1003.
32. Knight,P.A. and Ruiz,D. (2013) A fast algorithm for matrix balancing. *IMA J. Numer. Anal.*, **33**, 1029–1047.
33. Wolff,J., Backofen,R. and Gruening,B. (2020) Loop detection using Hi-C data with HiCExplorer. bioRxiv doi: https://doi.org/10.1101/2020.03.05.979096, 06 March 2020, preprint: not peer reviewed.
34. Fullwood,M.J., Liu,M.H., Pan,Y.F., Liu,J., Xu,H., Mohamed,Y.B., Orlov,Y.L., Velkov,S., Ho,A., Mei,P.H. *et al.* (2009) An oestrogen-receptor-α-bound human chromatin interactome. *Nature*, **462**, 58–64.

35. Zhang,Y., Liu,T., Meyer,C.A., Eeckhoute,J., Johnson,D.S., Bernstein,B.E., Nusbaum,C., Myers,R.M., Brown,M., Li,W. *et al.* (2008) Model-based analysis of ChIP-Seq (MACS). *Genome Biol.*, **9**, R137.
36. Nagano,T., Lubling,Y., Várnai,C., Dudley,C., Leung,W., Baran,Y., Cohen,N.M., Wingett,S., Fraser,P. and Tanay,A. (2017) Cell-cycle dynamics of chromosomal organization at single-cell resolution. *Nature*, **547**, 61.
37. Lee,G., Rodriguez,C. and Madabhushi,A. (2007) An empirical comparison of dimensionality reduction methods for classifying gene and protein expression datasets. In *International Symposium on Bioinformatics Research and Applications*. Springer pp. 170–181.
38. Deegalla,S. and Boström,H. (2007) Classification of microarrays with knn: comparison of dimensionality reduction methods. In *International Conference on Intelligent Data Engineering and Automated Learning*. Springer pp. 800–809.
39. DeTomaso,D., Jones,M.G., Subramaniam,M., Ashuach,T., Chun,J.Y. and Yosef,N. (2019) Functional interpretation of single cell similarity maps. *Nat. Commun.*, **10**, 4376.
40. Broder,A.Z. (1997) On the resemblance and containment of documents. In: *Proceedings Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*. IEEE pp. 21–29.
41. Wolff,J., Backofen,R. and Gruening,B. (2020) Approximate k-nearest neighbors graph for single-cell Hi-C dimensional reduction with MinHash. bioRxiv doi: https://doi.org/10.1101/2020.03.05.978569, 05 March 2020, preprint: not peer reviewed.
42. Andrey,G., Schöpflin,R., Jerković,I., Heinrich,V., Ibrahim,D.M., Paliou,C., Hochradel,M., Timmermann,B., Haas,S., Vingron,M. *et al.* (2017) Characterization of hundreds of regulatory landscapes in developing limbs reveals two regimes of chromatin folding. *Genome Res.*, **27**, 223–233.
43. Batut,B., Hiltemann,S., Bagnacani,A., Baker,D., Bhardwaj,V., Blank,C., Bretaudeau,A., Brillet-Guéguen,L., Čech,M., Chilton,J. *et al.* (2018) Community-driven data analysis training for biology. *Cell Syst.*, **6**, 752–758.

7 Robust and efficient single-cell Hi-C clustering with approximate k-nearest neighbor graphs

Robust and efficient single-cell Hi-C clustering with approximate k-nearest neighbor graphs
Joachim Wolff, Rolf Backofen, Björn A Grüning
Bioinformatics. Accepted on 19 May 2021 DOI: 10.1093/bioinformatics/btab394

Personal contribution
I contributed by writing the manuscript, designing the method for clustering of single-cell Hi-C data with the dimension reduction technique MinHash, and implemented the approach. In recognition of these significant contributions, I am listed as the first author for this publication.

Contribution of Rolf Backofen
General PhD supervision of Joachim Wolff and advice during the PhD process.

Contribution of Björn A Grüning
General PhD supervision of Joachim Wolff and advice during the PhD process.

Joachim Wolff

Joachim Wolff

The following co-authors confirm the above stated contribution:

Name	Date	Signature
Prof. Dr. Rolf Backofen	15.01.2021	
Dr. Björn Grüning	03.09.2021	B. Grüning

Bioinformatics, 2021, 1–8
doi: 10.1093/bioinformatics/btab394
Advance Access Publication Date: 22 May 2021
Original Paper

Genome analysis

Robust and efficient single-cell Hi-C clustering with approximate k-nearest neighbor graphs

Joachim Wolff [1,*], Rolf Backofen [1,2] and Björn Grüning [1]

[1]Bioinformatics Group, Department of Computer Science, University of Freiburg, 79110 Freiburg, Germany and [2]Signalling Research Centre CIBSS, University of Freiburg, 79104 Freiburg, Germany

*To whom correspondence should be addressed.

Associate Editor: Pier Luigi Martelli

Received on April 14, 2020; revised on May 3, 2021; editorial decision on May 14, 2021; accepted on May 19, 2021

Abstract

Motivation: Hi-C technology provides insights into the 3D organization of the chromatin, and the single-cell Hi-C method enables researchers to gain knowledge about the chromatin state in individual cell levels. Single-cell Hi-C interaction matrices are high dimensional and very sparse. To cluster thousands of single-cell Hi-C interaction matrices, they are flattened and compiled into one matrix. Depending on the resolution, this matrix can have a few million or even billions of features; therefore, computations can be memory intensive. We present a single-cell Hi-C clustering approach using an approximate nearest neighbors method based on locality-sensitive hashing to reduce the dimensions and the computational resources.

Results: The presented method can process a 10 kb single-cell Hi-C dataset with 2600 cells and needs 40 GB of memory, while competitive approaches are not computable even with 1 TB of memory. It can be shown that the differentiation of the cells by their chromatin folding properties and, therefore, the quality of the clustering of single-cell Hi-C data is advantageous compared to competitive algorithms.

Availability and implementation: The presented clustering algorithm is part of the scHiCExplorer, is available on Github https://github.com/joachimwolff/scHiCExplorer, and as a conda package via the bioconda channel. The approximate nearest neighbors implementation is available via https://github.com/joachimwolff/sparse-neighbors-search and as a conda package via the bioconda channel.

Contact: wolffj@informatik.uni-freiburg.de

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

1 Introduction

The chromosome conformation capture technique 3C (Dekker *et al.*, 2002) and its successors 4C (Simonis *et al.*, 2006; Zhao *et al.*, 2006), 5C (Dostie *et al.*, 2006) and Hi-C (Lieberman-Aiden *et al.*, 2009) have given insights into the organization of the 3D structure of the DNA and its impact on gene regulation over the last few years. Direct chromatin interactions can provide evidence, for example, for enhancer-promoter interactions and their contribution to the regulation process. Several reviews have been published in recent years, giving a broad overview of different Hi-C techniques and their abilities: Kempfer and Pombo (2020), McCord *et al.* (2020) and Bonev and Cavalli (2016). Single-cell Hi-C (Flyamer *et al.*, 2017; Gassler *et al.*, 2017; Nagano *et al.*, 2013; 2017; Ramani *et al.*, 2017; Stevens *et al.*, 2017) extends Hi-C to individual cells and provides insights into the processes of cell differentiation and division with respect to the dynamics of chromosome conformation. While Hi-C data analysis demands high computational resources, single-cell Hi-C increases this demand further due to the need to not only process one interaction matrix but potentially several thousands of them. Cell clustering, based on the interaction matrices to differentiate by the chromatin folding properties, is one of the most important parts of single-cell Hi-C data analysis to gain information about similarity and, therefore, the linkage between different cells. Hi-C interaction matrices are two-dimensional, representing the contacts between each pair of genomic positions. The interaction matrices do not represent a per base-pair interaction between loci but a binned one; i.e. multiple continuous base-pairs are counted as one interaction. This is referred to as a *resolution*, the fewer base-pairs per bin, the higher the resolution. The presented approach flattens the interaction matrices of a cell to a single dimension. It creates a new matrix where each row represents one cell to use classical clustering algorithms, such as k-means or spectral clustering. The downside of this approach is a high feature number; for example, with 1 megabase (Mb) resolution matrices and the mice mm9 reference genome (https://www.ncbi.nlm.nih.gov/assembly/GCF_000001635.26), 7.6 million features are present while using 10 kilobases (kb) matrices the matrix has 76 billion features.

Dimension reduction is a well-known approach to improve the clustering quality (Deegalla and Boström, 2007; DeTomaso *et al.*, 2019; Lee *et al.*, 2007). Computing a k-nearest neighbors graph, represented as a matrix, is one of them. A k-nearest neighbors graph connects nodes with k other nodes, and the edge weights represent the similarity between two nodes. In this work, each cell is considered a node, and the edge weight is the similarity between the two cells. With a k-nearest neighbors graph, the number of features is reduced to the number of cells. The exact k-nearest neighbor's graph algorithm has a run time of $O((n \times f)^2)$, with n the number of cells and f the number of features. As long as f is reasonably small, the computation time will mainly depend on the number of cells n, but as the number of features rises to the millions, the compute time becomes more dependent on the features rather than the number of cells. Moreover, the higher the features, the less meaningful similarity between two cells is. Both phenomenons are known in the context of the curse of dimensionality (Aggarwal *et al.*, 2001; Bellman, 2015; Beyer *et al.*, 1999; Chen, 2009; Hammer, 1962; Hinneburg *et al.*, 2000; Houle *et al.*, 2010). For many k-nearest neighbor graphs, distance metrics such as the Euclidean distance or similar metrics are used to compute the relation of two instances. In Hi-C, using the Euclidean distance or similar metrics is, in our opinion, problematic. Consider the following: one cell has 0 interactions at a specific location, a second cell has 100 and a third cell 200. Using the Euclidean distance, the first and third cells are equidistant from the second cell. However, in our opinion, the results must be interpreted so that the second and third cells have recorded interactions and are therefore closer to each other than a cell without any interactions. To generalize this argument, Hi-C matrices with similar structures like A/B compartments, TADs or loops should, in our opinion, considered as more similar to each other, independent of the interaction intensity. Metrics like the Euclidean distance cannot guarantee this property; however, due to Hi-C matrices' very sparse nature, the Jaccard index can provide this. Similar observations concerning the sparsity of the data and the problematical usage of the Euclidean distance have been made in single-cell RNA-seq.

In this article, we propose, therefore, an algorithm to overcome these limitations. A k-nearest neighbor graph is computed to reduce the high number of features with respect to the number of cells. Instead of the problematic Euclidean distance, a measurement with a binary interpretation of the contacts, the Jaccard index, is used. Concerning the expected increasing read coverage and cell number, the quadratic run time to construct the k-nearest neighbor graph is replaced by a linear run time solution. The linearity is achieved by exchanging the Jaccard index by its approximation, MinHash (Broder, 1997), a locality-sensitive hash function technique.

2 Materials and methods

The interaction matrices of cells need to be compiled into one interaction matrix to cluster single-cell Hi-C data. Each individual single-cell matrix's dimensions depend on the used reference genome and the resolution of the Hi-C data. To compile the individual single-cell matrices with $(n \times n)$ dimensions to one matrix without losing any information, each interaction matrix is flattened to $1 \times (n \times n)$ dimensions:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{bmatrix} \tag{1}$$

Subsequently all m flattened interaction matrices are compiled to one interaction matrix with $(m \times (n \times n))$ or $(m \times n^2)$:

$$\begin{matrix} [1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9] \\ & & & & \vdots & & & & \\ [1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9] \end{matrix} \Rightarrow \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ & & & & \vdots & & & & \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{bmatrix} \tag{2}$$

Figure 1(A) provides an abstract graphical description.

This new compiled single-cell Hi-C matrix can be used to apply well-known clustering algorithms like k-means or spectral clustering directly. However, research on the curse of dimensionality shows that the more features are available, the less meaningful a similarity is (Aggarwal *et al.*, 2001; Beyer *et al.*, 1999; Hinneburg *et al.*, 2000). Our approach reduces the number of features before a clustering algorithm is applied. For this, we compute a k-nearest neighbors graph using the approximation of the Jaccard index, MinHash, as a similarity measure. Subsequently, a principal component analysis (PCA) and a UMAP embedding (McInnes *et al.*, 2020) are used to reduce the dimensions of the k-nearest neighbor's graph to low dimensional space.

2.1 Jaccard index

The Jaccard index of two cells is given by their sets A, B of non-zero feature ids. A non-zero feature id is the feature index position of a feature which cell has at its index at least one recorded Hi-C interaction.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

Based on the Jaccard index, the similarity between two cells in terms of how many features they share can be used to compute a k-nearest neighbors graph where the edge weight is the similarity. However, the computation of a k-nearest neighbors graph is in $O(n^2)$. Its approximation replaces the Jaccard index with MinHash (Broder, 1997) to compute in linear time.

2.2 MinHash

Cells which share features are more likely to be similar to each other compared to cells with less common features. MinHash uses this fact; for each cell, only a set of features' id A of non-zero features (non-zero Hi-C interactions) are considered (similar to Heyne *et al.*, 2012), and the hash value per MinHash function h is computed as the argmin over all non-zero features $a \in A$ of a hash function f. A set of MinHash functions H and hash functions F are used; $h \in H$ and $f \in F$. The similarity between two cells is computed by counting the number of collisions overall MinHash functions.

$$h(A) = argmin_{a \in A} f(a) \tag{4}$$

Broder shows that MinHash is an unbiased estimator of the Jaccard index:

$$P(h(A) = h(B)) = (|A \cap B|)/(|A \cup B|) = J(A, B) \tag{5}$$

2.3 Clustering

Multiple options are available to process the Hi-C contacts to compute the k-nearest neighbor's graph with MinHash. The first option uses inter- and intra-chromosomal contacts; the second option only intra-chromosomal contacts. The first option has the benefit of considering potential important long-range contacts; however, distinguishing them from noise is only possible with a high read coverage. It might be, therefore, beneficial for the cluster results to consider only intra-chromosomal contacts. The parameters used to compute the k-nearest neighbor's graph are the number of employed hash functions and, therefore, how many collisions occur. The number k of neighbors to be computed and if the additional Euclidean distance based on the pre-selection of candidates should be considered. The number of features of the k-nearest neighbor graph is still considered as high dimensional. A principal component analysis followed by a UMAP embedding is applied before the clustering to reduce the number of dimensions further. For the clustering algorithms, we use the algorithms offered by scikit-learn (Pedregosa *et al.*, 2011) and limit ourselves to the clustering algorithms that support a user-specified fixed number of clusters. These are K-means, spectral clustering, birch and agglomerative clustering.

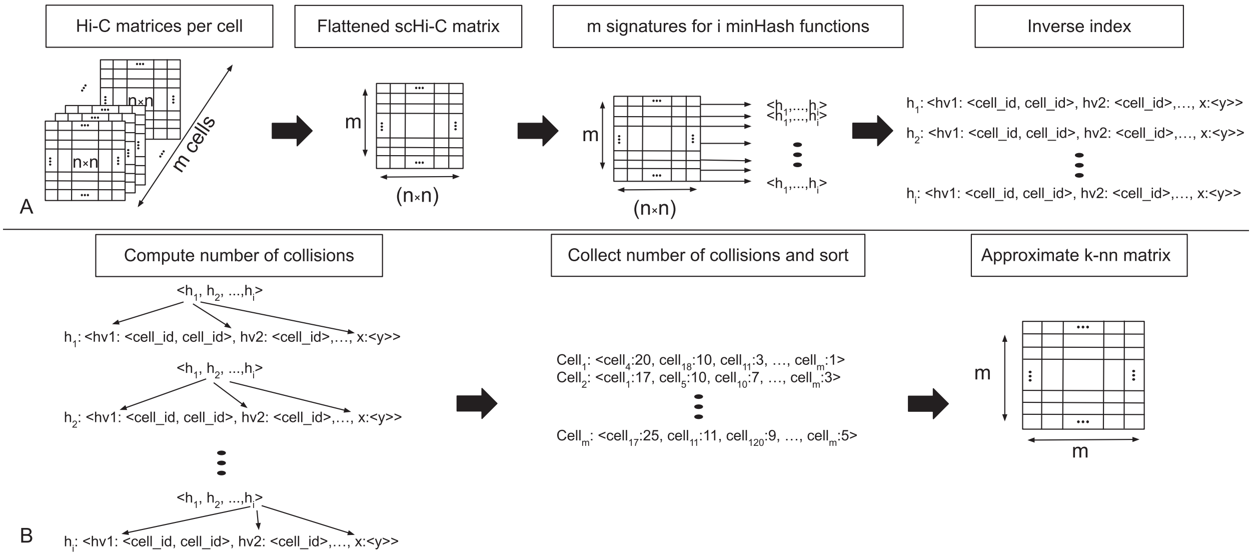

Hi-C matrices per cell
Flattened scHi-C matrix
m signatures for i minHash functions
Inverse index
A
n×n
n×n
m cells
m
(n×n)
m
(n×n)
<h1,...,hi>
<h1,...,hi>
<h1,...,hi>
h1: <hv1: <cell_id, cell_id>, hv2: <cell_id>,..., x:<y>>
h2: <hv1: <cell_id, cell_id>, hv2: <cell_id>,..., x:<y>>
hi: <hv1: <cell_id, cell_id>, hv2: <cell_id>,..., x:<y>>
Compute number of collisions
Collect number of collisions and sort
Approximate k-nn matrix
<h1, h2, ...,hi>
h1: <hv1: <cell_id, cell_id>, hv2: <cell_id>,..., x:<y>>
<h1, h2, ...,hi>
h2: <hv1: <cell_id, cell_id>, hv2: <cell_id>,..., x:<y>>
<h1, h2, ...,hi>
B
hi: <hv1: <cell_id, cell_id>, hv2: <cell_id>,..., x:<y>>
Cell1: <cell4:20, cell18:10, cell11:3, ..., cellm:1>
Cell2: <cell1:17, cell5:10, cell10:7, ..., cellm:3>
Cellm: <cell17:25, cell11:11, cell120:9, ..., cellm:5>
m
m

Fig. 1. (**A**) Pre-processing and fitting: All $n \times n$ Hi-C matrices of the m cells are flattened to one single-cell Hi-C (scHi-C) matrix with $m \times (n \times n)$ dimensions. For each row a signature is computed and inserted into the inverse index. (**B**) K-nearest neighbors computation: Per signature, the hash function h_i is checked if the hash value at signature index i is present in the inverse index. If such a collision is detected, the associated cell ids are stored. After all hash functions are checked, the number of occurrences for the cell_ids is counted and sorted. This order gives the nearest neighbor's relationship

2.4 Implementation

2.4.1 Inverse index

Fast computation of a k-nearest neighbors graph requires a linear query time and a significant reduction of the number of features to overcome the curse of dimensionality. A regular index stores the computed hash values of a hash function per cell, leading to $O(n \times n \times h) \in O(n^2)$ to create the k-nearest neighbor graph. In order to reduce the construction to linear time, an *inverse index* is used. Per hash function, the hash values with the corresponding cell id are stored. To construct a k-nearest neighbors graph, for each cell, the hash functions have to be checked for collisions which is per hash function in $O(1)$ and for all cells $O(n \times h) \in O(n)$.

2.4.2 Fitting

The MinHash values of all hash functions together are called the *signature* of the cell; these signatures are inserted into the inverse index to achieve a fast query time. The run time of the fitting depends on the number of cells n, the number of hash functions h and the number of non-zero features per cell f and is given as $O(n \times h \times f) \in O(n)$. For an example of the fitting and the inverse index structure, refer to Figure 2.

2.4.3 Collision based approximate nearest neighbors graph

The number of *hash collisions* between two cells gives an estimate of their similarity. The signature of a cell is used to search for *hash collisions* in the inverse index to compute the estimate. A *hash collision* between two cells is defined as the same hash value for the same hash function. The more collisions two cells have, the more similar they are. The query time of this approach depends only on the number of used hash functions and, if not stored in memory from the fitting phase, the computation of signatures. The effect of sorting all occurrences of collisions and the query time of the used data structures of the inverse index on the run time should also be considered, although it is negligible from the user's point of view.

Signatures: <2, 5, 1>; <4, 7, 2>; <4,7,1>; <5, 5, 1>;

Hash function 1: <2: (1); 4:(2, 3); 5: (4)>

Hash function 2: <5: (1, 4); 7: (2, 3)>

Hash function 3: <1: (1, 3, 4); 2: (2)>

Fig. 2. An example signature and inverse index: The signature is created for four cells and three hash functions. The inverse index stores the computed hash value and the id of the cell for each hash function. For example, for the second cell $< 4, 7, 2 >$ the first hash function *Hash function* 1 stores the computed hash value 4 and associates the id of the cell: *Hash function* $1 : < 2 : (1); 4 : (2, 3); 5 : (4) >$. The same hash function and hash value occur for cell number three again; this is a *collision* of hash function 1 for cell 2 and cell 3

2.4.4 Technical implementation

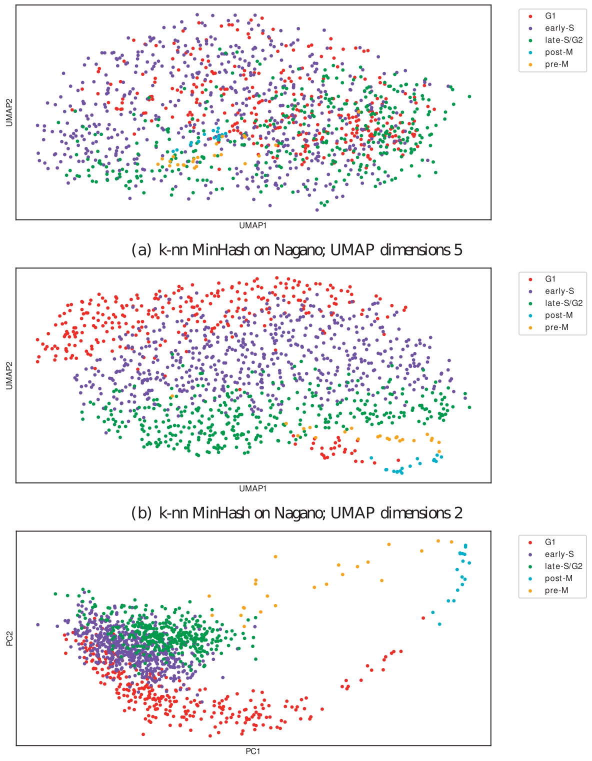
For this implementation, we use the hash function '32 bit mix function' designed by Thomas Wang (https://gist.github.com/badboy/6267743#32-bit-mix-functions) published in 1997/2007. The hash function is always the same; however, for each hash function $f \in F$ the seed differs. The index values for a 10 kb resolution matrix exceed the data range of 32-bit by 5 bit. The index values are modified via modulo operation to fit the 32-bit range to avoid a 64-bit hashing. The sparsity of the data is advantageous and results in more than 98% unique indices after the modulo operation. To compute the approximate nearest neighbors with MinHash, a highly optimized library, 'sparse-neighbors-search', was implemented in C++ with SSE and OpenMP support. To ensure user accessibility, the C++ library is embedded in a Python 3.6, 3.7 and 3.8 interface. The MinHash approximation of a k-nearest neighbors graph is part of the scHiCExplorer (Wolff *et al.*, 2020a); a software to process, analyze and visualize single-cell Hi-C data.

3 Results

The algorithm is tested with differing properties and settings to evaluate the clustering abilities of the proposed algorithm. The clustering is tested on the matrices at different levels of processing. Compared here is the ability to detect the different cell cycle phases (Nagano *et al.*, 2017) respectively the cell types (Ramani *et al.*, 2017) based on the low dimensional embedding of the Hi-C cells. First, the MinHash approach and its differentiation ability is discussed. Second, the best settings for the algorithm are investigated, and third, the proposed solution is compared to the competing algorithm *scHiCluster* from Zhou *et al.* (2019); also a clustering based on a principal component analysis on the raw matrices, and a k-nearest neighbor graph computed with scikit-learns implementation are considered.

3.1 Embedding and differentiability of MinHash

The Jaccard index-based approach with its approximation via MinHash, combined with a consecutively PCA and UMAP embedding for a further dimension reduction, provides good differentiability of the test data. The 1 MB cell cycle data from Nagano *et al.* (2017) shown in Figure 3a is reduced to five UMAP components and visualized are the first two dimensions. The visualization with the first two UMAP dimensions is not indicating a good clustering result (Fig. 3a), but an embedding with the same parameters, but reducing to two UMAP dimensions instead of five, improves this (Fig. 3b). However, the clustering results of this approach are not as good as for the five UMAP dimensions (Supplementary Tables S1 versus S2). Early-S (purple), late-S/G2 (green) and G1 (red) cell cycles are differentiated, and post-M (cyan) and pre-M (yellow) are projected to a similar location; an overlap of the different cell cycle phases is given. Good clustering results are confirmed by validating the detect clusters by Nagano *et al.* (2017) provided cell cycle labels (Supplementary Table S1). A batch effect is slightly visible (Supplementary Fig. S1a) but is not dominating. The 1 MB cell type data from Ramani *et al.* (2017) are displayed in Figure 4a and b. The four cell lines are provided from two batches, and a strong batch effect is visible (Supplementary Fig. S2a). The embedding of the ML1 batch with HeLa and HAP1 cells show a clear differentiation of the two cells (Fig. 4a), and the ML3 batch with K562 and GM12878 provides a good differentiation too (Fig. 4b). However, the ML3 embedding has some minor issues: K562 cells are projected to the top to the area of GM12878 cells. It requires further investigation if this is an error by the embedding approach or if, as the spatial separation indicates, further subtypes are present within the dataset. Ramani *et al.* (2017) provides only the cell type labels, but it is not unlikely that the cell type data itself contains cells with a different cell cycle phase.



(a) k-nn MinHash on Nagano; UMAP dimensions 5

(b) k-nn MinHash on Nagano; UMAP dimensions 2

(c) Zhou's scHiCluster on Nagano

Fig. 3. Embedding into a two dimensional space based on cell cycle data from Nagano *et al.* (2017). Computed on 1275 cell cycle phase cells with their cell cycle phase label. (**a** and **b**) are computed with the proposed algorithm. (a) is with 5 UMAP dimensions and plotted with the first two, (b) uses the same parameters but with two UMAP dimensions. The second approach is better for a visualization, however, Supplementary Tables S1 and S2 clearly indicate the clustering result with the first approach is better. (**c**) shows the first two principal components of Zhou's scHiCluster

3.2 Jaccard versus Euclidean distance

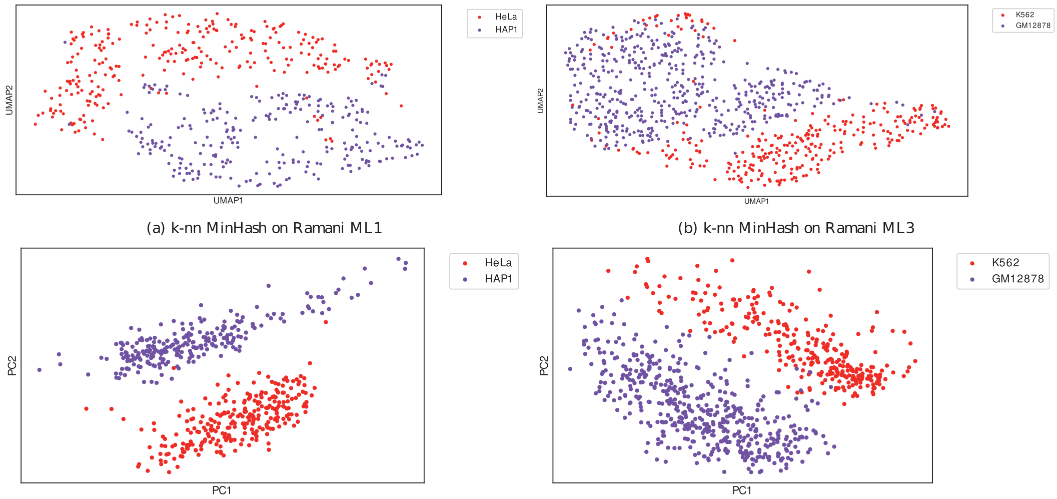
The proposed algorithm's primary aim is to reduce the high dimensional space of the single-cell Hi-C data from millions and billions of dimensions to a lower-dimensional space to improve the clustering abilities. This involves several dimension reduction steps: The reduction of the single-cell Hi-C interaction data via a k-nearest neighbors graph to (*cell* × *cell*) dimensions. The two measures to compute the k-nearest neighbor graph, namely the approximate Jaccard index and Euclidean distance, have a different impact on the embedding results. On the 1275 cells from Nagano *et al.* (2017) with a 1 Mb resolution and the five pre-classified cell cycle phases (G1, early-S, late-S/G2, post-M and pre-M), the approximate Jaccard index can create a distinguishable clustering, while the Euclidean based approach falls behind in terms of accuracy. For example, for an accuracy level of at least 70% of uniquely classified cells of a cell phase per cluster: the Jaccard index-based approach detects 73% of G1, 61% of early-S, 87% of late-S/G2, 94% of post-M and 91% of pre-M; while for the Euclidean distance only 37% of G1, 35% of early-S and 32% of late-S/G2 and both post-M and pre-M are not detected (Supplementary Tables S1 and S3). The Euclidean distance's performance can be explained by its behavior in high dimensions (Aggarwal *et al.*, 2001; Beyer *et al.*, 1999; Hinneburg *et al.*, 2000). Moreover, the Euclidean distance does not differ between no-contacts and contacts, whereas the Jaccard index, on the other hand, exactly makes this distinction and is, therefore, more suitable.

3.3 Embedding via UMAP with and without prior PCA

The principal component analysis reduces the matrix dimensions from (*cells* × *cells*) to a user-defined number of components (PC) (*cells* × $|PC|$). The problem of not using a principal component analysis is present for the pre-M and post-M cells: The post-M cells are mixed with pre-M cells (cluster 10), and the pre-M cells vanish in cluster 4, which is dominated by late-S/G2 cells (Supplementary Table S4). Third, using UMAP in combination with the metric 'Canberra' (Lance and Williams, 1966) reduces the number of dimensions to a user-defined number of UMAP components (UMAP_COMP) with $|PC| > |UMAP_{COMP}|$: $(cells \times |UMAP_{COMP}|)$. This creates better clustering results in comparison to the dataset that was only using principal component analysis (Supplementary Tables S1 versus S4). Performing no principal component analysis followed by UMAP has a worse detection rate and does not recognize any pre-M and post-M cells (Supplementary Table S5). The situation is identical if the clustering is directly applied to the approximate k-nearest neighbor's graph without an additional PCA and UMAP embedding (Supplementary Table S6).

3.4 Other parameters properties

The ideal parameter setting to compute the approximate k-nearest neighbor graph is investigated; it is beneficial to initially use only intra-chromosomal contacts (Supplementary Table S7), as well as more hash functions to contribute to a better differentiation (Supplementary Tables S8 and S9). In this context, the density of a matrix is also essential. For example, the density distribution of the cells in a 30 Mb context around the main diagonal of a 1 kb matrix (from Gassler *et al.*, 2017) with a density of 0.000002 is too sparse to create a substantial amount of hash collisions, independent of the number of hash functions used (Supplementary Figs S5–S13). It is beneficial to compute a full k-nearest neighbor graph and not, e.g. a 100-nearest neighbor or a 1000-nearest neighbors graph (Supplementary Tables S10 and S11). Last, the method to cluster the data is investigated; spectral clustering is compared to the other tested approaches, the algorithm with the best precision (Supplementary Tables S1 and 12–S17).



(a) k-nn MinHash on Ramani ML1

(b) k-nn MinHash on Ramani ML3

(c) Zhou's scHiCluster on Ramani ML1

(d) Zhou's scHiCluster on Ramani ML3

Fig. 4. Embedding into a two dimensional space for cell type data from Ramani *et al.* (2017). Separated by the two batches ML1 (**a** and **c**) and ML3 (**b** and **d**) and labeled by their cell types

3.5 Comparison with competing approaches

The differentiation ability of the proposed algorithm is, compared to Zhou's scHiCluster, on a more advanced level. Considering a unique level of 70% of a cell phase per cluster, Zhou's scHiCluster detects 53% of G1 (versus 73%), 50% of early-S (versus 61%), 54% late-S/G2 (versus 87%) and is not able to detect any of the pre-M and post-M cells. Considering a uniqueness level of 80%, Zhou's scHiCluster detects more G1 cells (53% versus 51%) but less early-S (50% versus 60%), late-S/G2 (54% versus 87%), post-M (0% versus 94%) and pre-M (0% versus 91%); consider Supplementary Tables S1 and S18. For both embedding approaches, a distorted relation of the number of cells from each cell phase could be problematic. Three phases are present 1235 out of 1275 times (G1 300, early-S 573, late-S/G2 362), while post-M is present 17 and pre-M 23 times.

Besides the clusters with a high amount of a unique cell phase, the clustering result shows that mixed clusters do not have a random structure but represent the cell cycle's dynamic process. Cluster 5 of the proposed algorithm contains 11% of early-S and 88% late-S cells; Cluster 2, 6, 8 and 10 a mix of G1 and early-S cells. The two major phases in each cluster are consecutive in the cell cycle, and a strict separation with no overlaps of phases would be an unexpected result.

A batch effect is slightly visible (Supplementary Fig. S1a–c) but does not dominate the differentiation of the embedding. Furthermore, the detection rates of the clustering directly applied on a k-nearest neighbor graph computed by scikit-learn's implementation (Supplementary Table S19), on a principal component analysis reduced dataset (Supplementary Table S20) or on the raw data (Supplementary Table S21) are significantly worse and cannot compete with the proposed algorithm. The embedding on 10 kb resolution is different. While Zhou's scHiCluster cannot perform the computation within a reasonable time nor operate within generous memory requirements (Supplementary Tables S22 and S23), the proposed algorithm has significant issues distinguishing the cell phases. Two cell cycle phases (early-S and late-S/G2) are partially differentiated; however, they have significant overlaps with each other, especially for the G1 phase, and not all are embedded in a particular region. Post-M cells are embedded into one region, but the pre-M cells are distributed over the embedding, with no exact region, and therefore, no clustering can be achieved for this cell cycle phase (Supplementary Tables S24–S26). Investigating the batch relation shows no correlation between the batch and the embedded region (Supplementary Fig. S3). The bad detection rate can be explained by a too sparse dataset with a density of 0–0.0006 (Supplementary Fig. S6 (right)). Even a high number of hash functions does not help to create a meaningful similarity between the cells (Supplementary Tables S24–S26 with 20 000; 40 000 and 50 000 hash functions).

Considering the different cell type data from Ramani *et al.* (2017), both the proposed algorithm and Zhou's scHiCluster show a separation by the two batches, ML1 and ML3 (Supplementary Fig. S2a and b). For this reason, the cells of the two batches are separately computed. While per batch, only two cell types are present (ML1: HeLa and HAP1; ML3: K562 and GM12878), the results indicate subtypes in the data. Both Zhou's scHiCluster and the proposed algorithm benefit from using more clusters. For ML1, the proposed algorithm outperforms Zhou's scHiCluster if two clusters are used: Considering a uniqueness of at least 70%, the proposed algorithm detects 91% of HeLa cells and 94% of HAP1, while Zhou's scHiCluster detects 72% for both cell types. A uniqueness level of 80% or 90% keeps the results at an equivalent level for the proposed algorithm but let it drop to 0% for Zhou's approach. However, the situation is different if three clusters are used: at a level of 90%, the proposed algorithm detects 95% of HeLa and 92% of HAP1 while Zhou's approach detects 96% and 100% (Supplementary Tables S27 and S28). The situation is similar for ML3: Using two clusters, the proposed algorithm detects slightly more cells for GM12878 (73% versus 72%), but both detect 0% of the K562 at a uniqueness level of 70%. Using five clusters shows an advantage of Zhou's scHiCluster, where it detects 94% for K562 and 98%for GM12878 at a uniqueness level of 80%; the proposed algorithm detects 78% for K562 and 94% for GM12878 (Supplementary Tables S29 and S30). Working on 10 kb data from Ramani *et al.* (2017), Zhou's scHiCluster cannot compute it within a reasonable time and memory constraints; however, the results of the proposed algorithm are mixed. A batch effect of ML1 and ML3 is visible (Supplementary Fig. S4), but a clear differentiation of the cell types not (Supplementary Tables S31 and S32). A differentiation of more extensive parts of the GM12878 cells for a uniqueness level of 70% is possible with 74%, but upon a closer investigation of the clusters, it is evident that a high mixture of the cells is given. This is especially true for ML1, and the cell types HeLa and HAP1, where no clear differentiation is possible. We assume the density of 0–0.00004 for most of the cells (Supplementary Fig. S6 (right)) is too sparse to create a good nearest neighbors computation.

3.6 Contact decay profiles

Contact decay profiles show for each cell in a given cluster the summed number of contacts per genomic distance. Each row is the genomic distance between the Hi-C contacts' two locations, and the columns are the cells. It is the nature of Hi-C contacts to decay with increasing distances between the two locations. Moreover, the decay of contacts should have a similar pattern for each detected cluster since the clusters' cells are sorted by the short to long-distance contact ratio. The plot gives a global indication of the detected clusters' correctness but incorrectly detected individual cells vanish. Figure 5 shows a contact decay plot based on the cluster results as shown in Supplementary Tables S1 (the proposed algorithm) and S18 (Zhou's scHiCluster approach) on the cell cycle data from Nagano. Both results are very similar from a global perspective. Clear contact decay patterns within the clusters and differences to the other clusters are visible, indicating the dimension reduction, embedding and clustering are general functional. It should be noted that the proposed algorithm can detect the post-M (cluster 9) and pre-M (cluster 0), while Zhou's scHiCluster mixes both (cluster 0) or mixes it with late-S/G2 cells (cluster 4). In contrast to these results are the contact decay profiles, where clustering was performed on: the raw interaction matrices (Supplementary Fig. S14a), a Euclidean distance-based k-nn (Supplementary Fig. S14g, h), the proposed algorithm using the Euclidean distance (Supplementary Fig. S14b), without an intermediate principal component analysis (Supplementary Fig. S14d) or without UMAP (Supplementary Fig. S14e). All the results in Supplementary Figure S14 have significant differences in the contact decay within the clusters, clearly indicating an overlap of cells that are not consecutive in the cell cycle.

3.7 Consensus matrices

A consensus matrix is a bulk mean Hi-C matrix of all cells of a cluster. In the best case, all cells of a cluster have a similar chromatin pattern and provide an insight into the chromatin folding properties. The more noisy a consensus matrix is, the more likely the cells from different cell cycle phases or cell types are merged into the same cluster. Figure 6 shows the consensus matrices for chromosome 6 based on the clusters presented in Supplementary Tables S1 and S18. For both the proposed algorithm and Zhou's scHiCluster, different Hi-C contact matrix patterns and, therefore, different chromatin folding properties are well developed. Given a uniqueness of $> 80\%$ as shown in Supplementary Tables S1 and S18, the cell cycle stage G1 is represented by clusters 2, 4 and 7 for the proposed algorithm and clusters 6 and 8 for Zhou's scHiCluster. The patterns for the clusters are similar, and the same is true for the early-S clusters from us (1 and 8) and Zhou (7 and 9), late-S/G2 (3, 5 and 11 respectively 4 and 5); however, post-M and pre-M are identified by the proposed algorithm (cluster 0 and 9), where Zhou's scHiCluster instead mixes post-M and pre-M cells in cluster 0. A closer look at the consensus matrices for the other investigated approaches confirms the findings of the previous sections that the usage of raw data, the euclidean distance, a 100-nearest neighbors graph, no PCA, no UMAP, inter- and intra-chromosomal contacts, or the usage of the scikit-learn k-nn do not lead to a good differentiation of the cell cycles (Supplementary Figs S15 and S16).

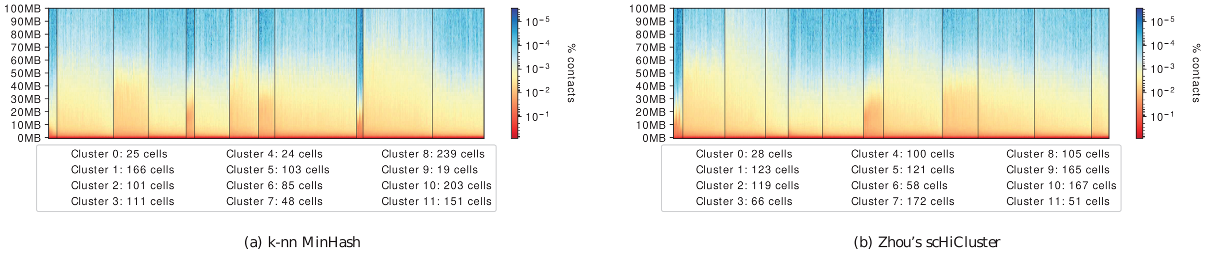


Fig. 5. Contact decay profile of the clusters; computed by scHicClusterMinHash with spectral clustering (**a**), Zhou's scHiCluster (**b**). Computation on 1 Mb resolution, with 1275 cell cycle phase cells from Nagano *et al.* (2017). Numbers indicate the cluster id and how many cells they contain; clusters are to be read from left to right. The number of clusters is 12 to have comparability to the cluster results of Nagano *et al.* (2017)

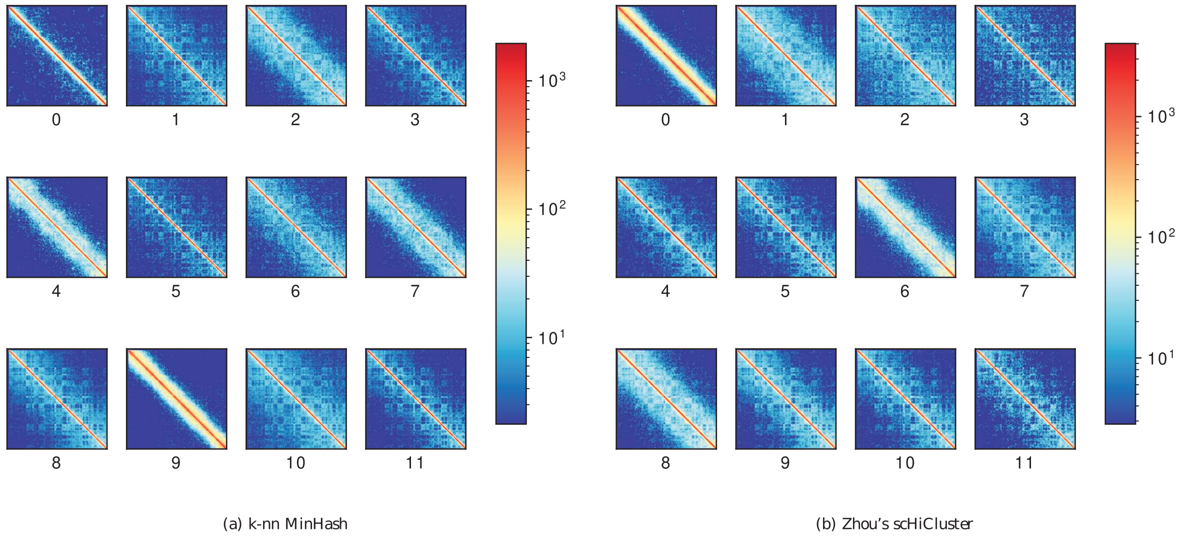


Fig. 6. Consensus clusters computed by scHicClusterMinHash with spectral clustering (**a**) and Zhou's scHiCluster (**b**). Computation on 1 Mb resolution, with 1275 cell cycle phase cells from Nagano *et al.* (2017), with 12 clusters. Numbers under the matrices indicate the cluster id. The number of clusters is 12 to have comparability to the cluster results of Nagano *et al.* (2017)

3.8 Runtime and memory

The approximate Jaccard index computation achieves faster or similar run times compared to the sklearn implementation of a k-nearest neighbors search, which is based on ball-trees, under consideration of the one megabase resolution single-cell Hi-C dataset as shown in Supplementary Tables S33 and S34. The runtimes can be influenced by the clustering algorithm used. This is especially the case for the clustering on raw data where k-means runs for around 40 min; for all others, a difference is present but is minor. However, we cannot explain the outstanding runtime of k-means on the XEON machine; a run on the AMD Ryzen based computer shows a runtime of 12 min, but a similar runtime for all other algorithms. The classical and naive way to reduce dimensions is a principal component analysis (PCA), but the method uses a high amount of memory (170 GB) even on the low-resolution matrix. For the 10 kb resolution matrix, the PCA method throws an error that it is *'unable to allocate 1.28 PiB for an array with shape (2633, 69647960281) and data type float64'*. All approximate k-nearest neighbor graph approaches use a similar amount of memory caused by the memory consumption at the read-in stage of the individual single-cell Hi-C matrices. The provided Euclidean mode of the proposed algorithm has a little increased run time compared to sklearns implementation. The runtimes of the proposed algorithm computed on a state-of-the-art computer with an SSD compute the clustering on the low-resolution matrix in around a minute and uses less than 8 GB of memory. The clustering on the high-resolution matrix is computed in 3:30 min and uses 40 GB of memory (compare Supplementary Tables S22 and S23). To have reduced memory usage, the mode *–saveMemory* is offered. The proposed algorithm uses one core to load in a batch processing way data; the user can define the share of the to be processed matrices. The 10 kb resolution matrices' processing with a share of 1% of the data took 13 min, but the memory usage is reduced to 12.5 GB (Supplementary Table S23). The more hash functions are used, the longer the run times are. The runtimes on a 1 Mb resolution using only cells with available labels is faster compared to Zhou's scHiCluster even for 20 000 hash functions (Supplementary Table S35).

Zhou's scHiCluster has a runtime of 14 min with the CPU implementation and 7 min on the GPU on a low resolution (1 Mb) matrix (Supplementary Table S34). The benefits of the proposed algorithm in terms of runtime and memory usage are significant under the consideration of a high-resolution single-cell Hi-C dataset. As shown in Supplementary Table S22, all methods besides the proposed algorithm cannot be computed due to their high memory usage of more than one terabyte. Considering Zhou's scHiCluster, we canceled the computation after 97 h runtime; the computation of the first loaded chromosome (chromosome 10) was not finished but had a peak memory usage of 970 GB. The data for Zhou's scHiCluster was stored in a RAM disk to exclude potential network file system issues. Only the proposed algorithm can compute a result while using a moderate 40 GB of memory; these resources are available for most researchers.

4 Discussion

It was shown that an approximate k-nearest neighbors graph can be used to reduce the number of dimensions required to cluster single-cell Hi-C data, with higher accuracy, faster run times and enabling users to analyze high-resolution data with a vastly reduced memory burden. The approximation of the Jaccard index proves to be a suitable similarity measure to create a base for clustering, while the Euclidean distance, considering the curse of dimensionality and the unique properties of Hi-C data, is shown to be not such an appropriate measure. The cluster results based on the approximate k-nearest neighbors with MinHash, the additional PCA on the computed k-nearest neighbor's graph, the UMAP embedding and a spectral clustering show a better differentiation of the chromatin folding properties compared to competitive methods. The presented approach to reduce the number of features, especially when dealing with millions to billions of dimensions, is crucial to achieving adequate run time and memory usages. Access to computers with more than 1 TB of memory is currently difficult, but access to computers or cluster nodes with 40 GB of memory is available to most researchers. The presented approximate nearest neighbors graph enables a broader range of researchers to work with single-cell Hi-C data and adds with the approximate Jaccard index a method to create a k-nearest neighbors graph. Moreover, the proposed algorithm is embedded into the scHiCExplorer, a software suite for single-cell Hi-C data analyses, and supports the native single-cell Hi-C format *scool* (Wolff *et al.*, 2020b). Thanks to the availability of the approximate k-nearest neighbor search as an independent software package, it can be easily integrated into other research issues dealing with similar matrix properties, as is the case in single-cell RNA-seq.

Acknowledgements

The authors thank Simon Bray, Anup Kumar, Mehmet Tekman and Leily Rabbani for proofreading the manuscript. They thank Mehmet Tekman for support and discussions on single-cell data and methods. They thank Milad Miladi and Fabrizio Costa for the supervision of a first implementation of the sparse-neighbors-search library during the master thesis of J.W. in the winter semester 2015/2016. Special thanks to Milad Miladi for discussions and helpful input and feedback on applying the MinHash approach to single-cell Hi-C data.

Funding

German Federal Ministry of Education and Research [031 A538A de. NBI-RBC awarded to R.B.]; German Federal Ministry of Education and Research [031 L0101C de. NBI-epi awarded to B.G.]. R.B. was supported by the German Research Foundation (DFG) under Germany's Excellence Strategy (CIBSS—EXC-2189 – Project ID 390939984).

Conflict of Interest: none declared.

References

Aggarwal,C.C. *et al.* (2001) On the surprising behavior of distance metrics in high dimensional space. In: *International Conference on Database Theory*. Springer, Berlin, Heidelberg. pp. 420–434.

Bellman,R.E. (2015) *Adaptive Control Processes: A Guided Tour*. Princeton University Press. Princeton, NJ, USA.

Beyer,K. *et al.* (1999) When is nearest neighbors meaningful. In *Proceedings of ICDT*, Vol. 99.

Bonev,B. and Cavalli,G. (2016) Organization and function of the 3d genome. *Nat. Rev. Genet.*, **17**, 661–678.

Broder,A.Z. (1997) On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*. IEEE, pp. 21–29.

Chen,L. (2009) *Curse of Dimensionality*. Springer US, Boston, MA, pp. 545–546.

Deegalla,S. and Boström,H. (2007) Classification of microarrays with knn: comparison of dimensionality reduction methods. In: *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, Berlin, Heidelberg, pp. 800–809.

Dekker,J. *et al.* (2002) Capturing chromosome conformation. *Science*, **295**, 1306–1311.

DeTomaso,D. *et al.* (2019) Functional interpretation of single cell similarity maps. *Nat. Commun.*, **10**, 1–11.

Dostie,J. *et al.* (2006) Chromosome conformation capture carbon copy (5c): a massively parallel solution for mapping interactions between genomic elements. *Genome Res.*, **16**, 1299–1309.

Flyamer,I.M. *et al.* (2017) Single-nucleus hi-C reveals unique chromatin reorganization at oocyte-to-zygote transition. *Nature*, **544**, 110–114.

Gassler,J. *et al.* (2017) A mechanism of cohesin-dependent loop extrusion organizes zygotic genome architecture. *EMBO J.*, **36**, 3600–3618.

Hammer,P. (1962) Adaptive control processes: a guided tour (R. Bellman).

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

Hinneburg,A. *et al.* (2000) What is the nearest neighbor in high dimensional spaces? In: *26th Internat. Conference on Very Large Databases*, Cairo, Egypt, pp. 506–515.

Houle,M.E. *et al.* (2010) Can shared-neighbor distances defeat the curse of dimensionality? In: *International Conference on Scientific and Statistical Database Management*. Springer, Berlin, Heidelberg, pp. 482–500.

Kempfer,R. and Pombo,A. (2020) Methods for mapping 3d chromosome architecture. *Nat. Rev. Genet.*, **21**, 207–226

Lance,G.N. and Williams,W.T. (1966) Computer programs for hierarchical polythetic classification ("Similarity Analyses"). *Comput. J.*, **9**, 60–64.

Lee,G. *et al.* (2007) An empirical comparison of dimensionality reduction methods for classifying gene and protein expression datasets. In: *International Symposium on Bioinformatics Research and Applications*. Springer, Berlin, Heidelberg, pp. 170–181.

Lieberman-Aiden,E. *et al.* (2009) Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science*, **326**, 289–293.

McCord,R.P. *et al.* (2020) Chromosome conformation capture and beyond: toward an integrative view of chromosome structure and function. *Mol. Cell*, 77, 688–708.

McInnes,L. *et al.* (2020) Umap: uniform manifold approximation and projection for dimension reduction. ArXiv e-prints 1802.03426, 2018.

Nagano,T. *et al.* (2013) Single-cell hi-c reveals cell-to-cell variability in chromosome structure. *Nature*, **502**, 59–64.

Nagano,T. *et al.* (2017) Cell-cycle dynamics of chromosomal organization at single-cell resolution. *Nature*, **547**, 61–67.

Pedregosa,F. *et al.* (2011) Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830.

Ramani,V. *et al.* (2017) Massively multiplex single-cell hi-c. *Nat. Methods*, **14**, 263–266.

Simonis,M. *et al.* (2006) Nuclear organization of active and inactive chromatin domains uncovered by chromosome conformation capture–on-chip (4c). *Nat. Genet.*, **38**, 1348–1354.

Stevens,T.J. *et al.* (2017) 3d structures of individual mammalian genomes studied by single-cell hi-c. *Nature*, **544**, 59–64.

Wolff,J. *et al.* (2020a) Galaxy HiCExplorer 3: a web server for reproducible Hi-C, capture Hi-C and single-cell Hi-C data analysis, quality control and visualization. *Nucleic Acids Res.*, **48**, W177–W184.

Wolff,J. *et al.* (2020b) Scool: a new data storage format for single-cell Hi-C data. *Bioinformatics*, btaa924.

Zhao,Z. *et al.* (2006) Circular chromosome conformation capture (4c) uncovers extensive networks of epigenetically regulated intra- and interchromosomal interactions. *Nat. Genet.*, **38**, 1341–1347.

Zhou,J. *et al.* (2019) Robust single-cell hi-c clustering by convolution-and random--walk–based imputation. *Proc. Natl. Acad. Sci. USA*, **116**, 14011–14018.

8

scool: A new data storage format for single-cell Hi-C data

scool: A new data storage format for single-cell Hi-C data
Joachim Wolff, Nezar Abdennur, Rolf Backofen, Björn A Grüning
Bioinformatics, Volume 37, Issue 14, 15 July 2021, Pages 2053–2054
https://doi.org/10.1093/bioinformatics/btaa924

Personal contribution
I contributed by writing the manuscript, designing the single-cell cooler (scool) file format, implementing the format as part of the cooler API and extending the documentation accordingly. In recognition of these significant contributions, I am listed as the first author for this publication.

Contribution of Nezar Abdennur
Review and discussion of the single-cell cooler file format, its implementation and general support for writing the manuscript.

Contribution of Rolf Backofen
General PhD supervision of Joachim Wolff and advice during the PhD process.

Contribution of Björn A Grüning
General PhD supervision of Joachim Wolff and advice during the PhD process.

Joachim Wolff

Joachim Wolff

The following co-authors confirm the above stated contribution:

Name	Date	Signature
Dr. Nezar Abdennur	12 April, 2021	
Prof. Dr. Rolf Backofen	22.04.2021	Digital unterschrieben von Prof. Dr. Rolf Backofen Datum: 2021.04.22 21:02:42 +02'00'
Dr. Björn Grüning	22.04.2021	B. Grüning

Bioinformatics, 2020, 1–2
doi: 10.1093/bioinformatics/btaa924
Advance Access Publication Date: 2 November 2020
Applications Note

Genome analysis

Scool: a new data storage format for single-cell Hi-C data

Joachim Wolff [1,*], Nezar Abdennur [2], Rolf Backofen[1,3] and Björn Grüning[1]

[1]Bioinformatics Group, Department of Computer Science, University of Freiburg, Freiburg 79110, Germany, [2]Institute for Medical Engineering and Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA and and [3]Signalling Research Centres BIOSS and CIBSS, University of Freiburg, Freiburg 79104, Germany

*To whom correspondence should be addressed.

Associate Editor: Peter Robinson

Received on July 22, 2020; revised on October 7, 2020; editorial decision on October 14, 2020; accepted on October 20, 2020

Abstract

Motivation: Single-cell Hi-C research currently lacks an efficient, easy to use and shareable data storage format. Recent studies have used a variety of sub-optimal solutions: publishing raw data only, text-based interaction matrices, or reusing established Hi-C storage formats for single interaction matrices. These approaches are storage and pre-processing intensive, require long labour time and are often error-prone.

Results: The single-cell cooler file format (*scool*) provides an efficient, user-friendly and storage-saving approach for single-cell Hi-C data. It is a flavour of the established cooler format and guarantees stable API support.

Availability and implementation: The single-cell cooler format is part of the cooler file format as of API version 0.8.9. It is available via pip, conda and github: https://github.com/mirnylab/cooler.

Contact: wolffj@informatik.uni-freiburg.de

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

1 Introduction

The storage, processing and analysis of single-cell Hi-C data face several challenges. First, the pre-processing overhead for single-cell Hi-C is both storage-intensive and time-consuming. For example, reproducing the results of the Nagano *et al.* (2017) single-cell Hi-C study requires downloading, demultiplexing and mapping more than 1.1 TB of compressed raw FASTQ data and creating thousands of interaction matrices. Second, manually handling so many files is unwieldy and prone to error. For example, some studies (Nagano *et al.*, 2013; Ramani *et al.*, 2017; Steven *et al.*, 2017) have published their pre-processed data as text-based files. Depending on the resolution, these files potentially store millions to billions of features in an uncompressed text file without fast random or partial access. By contrast, studies like Gassler *et al.* (2017) published pre-processed *cool* files (Abdennur and Mirny, 2019) for each cell and at multiple resolutions. However, due to redundancy in data storage and the complexitiy of handling a proliferation of files, this one-matrix-per-file approach has limited scalability and makes reproducible analysis challenging.

Here, we present the single-cell cooler file format, a ‘flavour’ of the cooler file format (Abdennur and Mirny, 2019), that stores multiple single-cell sparse Hi-C interaction matrices at a common resolution in a single HDF5 (Koziol and Robinson, 2018) file, allowing portable, space-efficient and fast access to single-cell interaction data. It uses the recommended extension.*scool*.

2 Materials and methods

We adopt the basic structure of the cooler format to create a collection of single-cell interaction matrices having common dimensions (see Fig. 1 A and B). Internally, all single-cell interaction matrices are stored under a group/cells and each matrix is identified by a unique cell ID and has the structure of a standard cooler *data collection* (Fig. 1A), allowing it to be read independently and transparently with the regular cooler API (see Listing 2). However, to eliminate redundancy, data structures that are shared between all cells are implemented as HDF5 hard-links pointing to the data that is shared between the cells, which is stored in the root group (Fig. 1B). These include the index-associated genomic coordinates of the Hi-C contacts:/bins/chrom,/bins/start,/bins/end, and the general information about the stored chromosomes:/chroms. These shared data structures provide significant space reduction when consolidating contact maps from a multitude of cells into a single file as opposed to use a large collection of separate cooler files. As a matrix format, a scool file stores binned contact data conforming to a specific genomic segmentation. While binning naturally leads to a loss of information and comparing datasets can be difficult when bin sizes are not compatible, single-cell cooler files can be binned at any resolution and even lossless contact maps can be produced using 1-bp resolution, if desired.

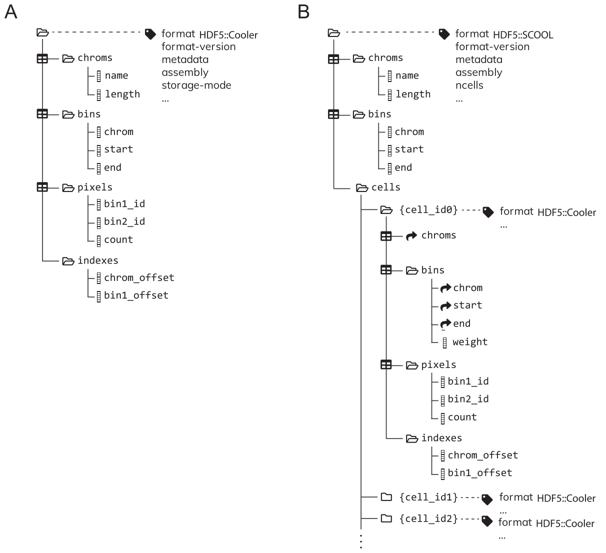


Fig. 1. (**A**) The structure of the cooler file format from Abdennur and Mirny (2019). (**B**) The structure of the single-cell cooler file format as a flavour of the cooler format. Hard linked groups and arrays are denoted with the curved arrow icon

2.1 Metadata

The single-cell cooler format stores specific metadata HDF5 attributes at the root level of the file: the format string HDF5::SCOOL, the format-version, whether the bin-type is fixed or variable, the bin-size, the genome assembly, the number of stored cells ncells and the optional field metadata for quality information or other user metadata.

2.2 Creation

To create a single-cell cooler file, the API can be used by calling the function *cooler.create_scool* and providing a file name, a dictionary of *bins* with the unique cell name as key (or a global common bin table, see Supplementary Material) and a dictionary mapping unique cell names to pixel information (Listing 1).

```
import cooler
bins_dict = {'cell1': bins1, 'cell2': bins2}
pixel_dict = {'cell1': pixels1, 'cell2': pixels2}
cooler.create_scool(cool_uri=file_name, bins=bins_dict,
    cell_name_pixels_dict=pixel_dict)
```

Listing 1 Python API example to create a scool file

2.3 Access

The interaction matrices in a single-cell cooler file can be listed with *cooler.fileops.list_coolers*. The interaction matrix of one cell can be retrieved using the resource syntax:

```
if cooler.fileops.is_scool_file(file_path):
    matrices_list = cooler.fileops.list_scool_cells(
        file_path)
    for cell in matrices_list:
        clr = cooler. Cooler(file_path + '::' + cell)
```

Listing 2 Python API example to read cells of a scool file

3 Results

The single-cell Hi-C data provided by Nagano *et al.* (2017) as raw FASTQ files has a compressed size of more than 1 TB. After demultiplexing, mapping and matrix creation several terabytes are consumed. At 10 kb resolution, 3882 individual cool files have a size of 3 GB, which is reduced to 1.9 GB using scool. At 1 MB, the cool files require 350 MB and the scool 267 MB. Gassler *et al.* (2017) provide 144 individual cool files at different resolutions. The storage reduction provided by scool is 2300–116 MB at 1 kb; 348–65 MB at 10 kb; 120–28 MB at 40 kb; and 63–26 MB at 100 kb. Compression ratios (see Supplementary Table S2) depend on the density and the resolution of the data. Generally, there is a greater overhead of storing a full bin table for each cell the fewer reads relative to the number of possible interactions and the higher the resolution. For example, the density for the 10 kb single-cell Hi-C data from Gassler *et al.* (2017) is up to 0.0004, while for Nagano *et al.* (2017), it is up to 0.0012. Accordingly, the scool/cool compression ratio for Gassler *et al.* (2017) (0.193) is better than that for Nagano *et al.* (2017) (0.633). See the Supplementary Material for more read coverages, densities and compression rates with respect to text and cooler files.

4 Conclusion

The single-cell cooler format makes it possible to store thousands of state-of-the-art single-cell Hi-C matrices in a single file with minimal redundancy. By storing all matrices in a space-efficient way, the reproducibility of single-cell Hi-C analyses is better achievable and the data are more accessible to a broader range of researchers. A portable container format prevents the complexity of managing thousands of files or needing to download and process large amounts of raw data from scratch. The embedding into the cooler API guarantees a fast and reliable access to the individual single-cell matrices and facilitates the use of parallel computing to improve analysis performance. The scool format is ideal for single-cell Hi-C data analysis software and is supported by scHiCExplorer (Wolff *et al.*, 2020).

Funding

We acknowledge funding from the German Science Foundation [CRC992 'Medical Epigenetics' to R.B. and B.G.]; and German Federal Ministry of Education and Research [031 A538A de. NBI-RBC awarded to R.B. and 031 L0101C de. NBI-epi awarded to B.G.]. R.B. was supported by the German Research Foundation (DFG) under Germany's Excellence Strategy [CIBSS-EXC-2189-Project ID 390939984].

Conflict of Interest: none declared.

References

Abdennur,N. and Mirny,L.A. (2019) Cooler: scalable storage for Hi-C data and other genomically labeled arrays. *Bioinformatics*, **36**, 311–316.

Gassler,J. *et al.* (2017) A mechanism of cohesin-dependent loop extrusion organizes zygotic genome architecture. *EMBO J.*, **36**, 3600–3618.

Koziol,Q. and Robinson,D. (2018) *HDF5*. [Computer Software] https://dx.doi.org/10.11578/dc.20180330.1. (22 October 2020, date last accessed).

Nagano,T. *et al.* (2013) Single-cell Hi-C reveals cell-to-cell variability in chromosome structure. *Nature*, **502**, 59–64.

Nagano,T. *et al.* (2017) Cell-cycle dynamics of chromosomal organization at single-cell resolution. *Nature*, **547**, 61–67.

Ramani,V. *et al.* (2017) Massively multiplex single-cell Hi-C. *Nat. Methods*, **14**, 263–266.

Steven,T.J. *et al.* (2017) 3D structures of individual mammalian genomes studied by single-cell Hi-C. *Nature*, **544**, 59–64.

Wolff,J. *et al.* (2020) Galaxy HiCExplorer 3: a web server for reproducible Hi-C, capture Hi-C and single-cell Hi-C data analysis, quality control and visualization. *Nucleic Acids Res.*, **48**, W177–W184.

9 pyGenomeTracks: Reproducible plots for multivariate genomic data sets

pyGenomeTracks: Reproducible plots for multivariate genomic data sets
Lucille Lopez-Delisle, Leily Rabbani, **Joachim Wolff**, Vivek Bhardwaj, Rolf Backofen, Björn Grüning, Fidel Ramírez, Thomas Manke
Bioinformatics, Volume 37, Issue 3, 1 February 2021, Pages 422–423
https://doi.org/10.1093/bioinformatics/btaa692

Personal contribution
I contributed by conceiving the manuscript and writing parts of it. I implemented the Hi-C support for pyGenomeTracks, and generally maintained the pyGenomeTracks project by organizing the repository, restructuring source code and maintaining the documentation.

Contribution of Lucille Lopez-Delisle
Contributed by implementing features of the pyGenomeTracks software, general maintaining the project, participating in writing the manuscript.

Contribution of Leily Rabbani
Contributed by implementing features of the pyGenomeTracks software and participating in writing the manuscript.

Contribution of Vivek Bhardwaj
Contributed by implementing features of the pyGenomeTracks software.

Contribution of Rolf Backofen
General PhD supervision of Joachim Wolff and advice during the PhD process.

Contribution of Björn Grüning
General PhD supervision of Joachim Wolff and advice during the PhD process.

Contribution of Fidel Ramírez
Contributed by designing the original pyGenomeTracks implementation (hicPlotTADs) and providing feedback about the manuscript.

Contribution of Thomas Manke
Contributed by providing feedback for the manuscript.

Joachim Wolff

Joachim Wolff

The following co-authors confirm the above stated contribution:

Name	Date	Signature
Dr. Lucille Lopez-Delisle	12/04/21	Delisle
Dr. Leily Rabbani	13.04.2021	L.Rabbani
Dr. Vivek Bhardwaj	16.04.2021	VivekBhardwaj
Prof. Dr. Rolf Backofen	22.04.2021	R. Backofen Digital unterschrieben von Prof. Dr. Rolf Backofen Datum: 2021.04.22 21:04:16 +02'00'
Dr. Björn Grüning	22.04.2021	B. Grüning
Dr. Fidel Ramírez	16.04.2021	Fidel R.
Dr. Thomas Manke	12.04.2021	Manke

Bioinformatics, 37(3), 2021, 422–423
doi: 10.1093/bioinformatics/btaa692
Advance Access Publication Date: 3 August 2020
Applications Note

Genome analysis

pyGenomeTracks: reproducible plots for multivariate genomic datasets

Lucille Lopez-Delisle[1], Leily Rabbani[2], Joachim Wolff[3], Vivek Bhardwaj[2], Rolf Backofen[3,4], Björn Grüning[3], Fidel Ramírez[2,*] and Thomas Manke[2]

[1]UPDUB, ISREC Department, School of Life Sciences (SV), EPFL, 1015 Lausanne, Switzerland, [2]Bioinformatics Group, Max Planck Institute of Immunobiology and Epigenetics, 79108 Freiburg, Germany, [3]Bioinformatics Group, Department of Computer Science, University of Freiburg, 79110 Freiburg, Germany and [4]Signalling Research Centres BIOSS and CIBSS, University of Freiburg, 79104 Freiburg, Germany

*To whom correspondence should be addressed.
Associate Editor: Robinson Peter

Received on April 30, 2020; revised on July 20, 2020; editorial decision on July 21, 2020; accepted on July 27, 2020

Abstract

Motivation: Generating publication ready plots to display multiple genomic tracks can pose a serious challenge. Making desirable and accurate figures requires considerable effort. This is usually done by hand or using a vector graphic software.

Results: pyGenomeTracks (PGT) is a modular plotting tool that easily combines multiple tracks. It enables a reproducible and standardized generation of highly customizable and publication ready images.

Availability and implementation: PGT is available through a graphical interface on https://usegalaxy.eu and through the command line. It is provided on conda via the bioconda channel, on pip and it is openly developed on github: https://github.com/deeptools/pyGenomeTracks.

Contact: fidel.ramirez@boehringer-ingelheim.com

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

1 Introduction

The analysis and visualization of multivariate genomic data faces several challenges. On one hand, there is a wide range of processing steps needed to analyze and to summarize large-scale data at a genome-wide level. Considerable effort has led to efficient tools as well as the adoption of scalable pipelines and frameworks, which provide a high degree of standardization and reproducibility (Bhardwaj *et al.*, 2019; Grüning *et al.*, 2018). On the other hand, advanced tools have been developed to support the visualization of genome-wide information and global patterns (Gehlenborg *et al.*, 2010). However, to turn genome-wide insights into testable interventions and validation experiments, researchers will usually return to locus-specific exploration. This is possible with a wide range of interactive genome browsers (Robinson *et al.*, 2011), and advanced browsers for three-dimensional data (Kerpedjiev *et al.*, 2018). Unfortunately, this exploration process is hard to standardize and yields heavily post-processed 'snapshots' to communicate the results. With pyGenomeTracks (PGT), we present a new and open software, which helps to standardize the generation of high-quality images in a programmatic approach. PGT supports the integrated visualization for a large variety of data sources, such as gene annotations, gene expression, chromatin signals and chromatin interactions.

2 Materials and methods

PGT provides an opportunity to map several genomic data tracks from a variety of resources onto one or a given list of genomic coordinates and generates an image per given coordinate including all of the input tracks. It offers support for a wide range of standard data formats in bioinformatics such as bigwig, bedgraph, epilogos, bed, gtf, narrow peaks, cool and HiCExplorer's native h5 format.

The only preprocessing step to generate a multitracks plot is to prepare a configuration file which contains all necessary parameters to plot the desired tracks of multiple input files. PGT provides a simple script (*make_tracks_file*) to generate a configuration file from a collection of input files. A usage example of it is shown in Supplementary Section S1.

This configuration file defines best practice, but it can also be fully customized by the user. In a configuration file, each track is defined as a block of parameters starting with its name *[track name]* and continues with the parameters for that track such as the file location, its title, height, color and so on, as has been shown in the Supplementary Section S1.

For the plot generation, users need to define the precise genomic coordinates either by providing a single coordinate or by providing

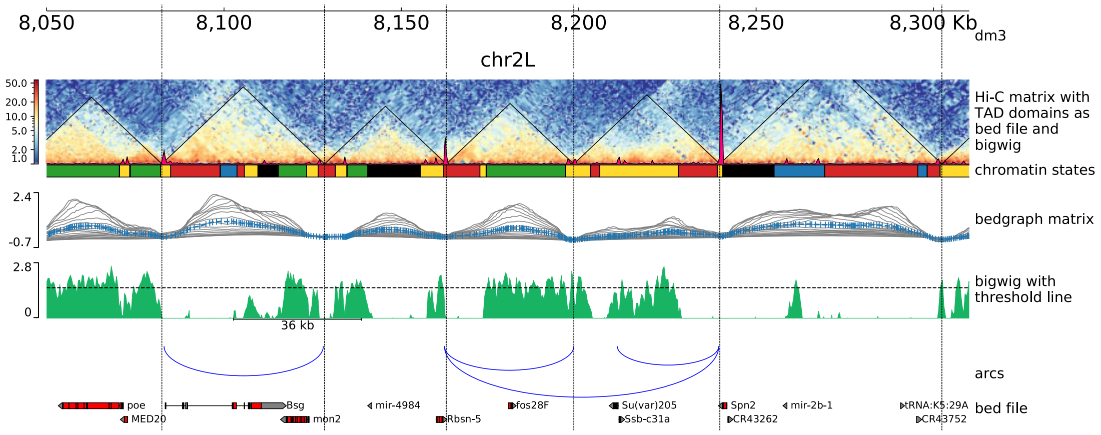


Fig. 1. An example plot generated by PGT on *Drosophila melanogaster* (dm3) data, Kc167 cell line. The first track from the top shows the genomic locus (chromosome 2L 8.05–8.31 Mb). The second track illustrates a Hi-C matrix track (Li *et al.*, 2015) overlaid by its detected TADs, via HiCExplorer and a coverage profile of CP190 ChIP. The matrix was in HiCExplorer h5 format, TADs are given as a bed file which is a direct output of HiCExplorer's hicFindTADs and the ChIP-Seq profile is provided as a bigwig file. The succeeding track shows the chromatin states, provided as a bed file, where the colors used are as defined in the ninth field of the bed file. The next track visualizes the TAD separation scores, the data are presented in a bedgraph matrix file format from HiCExplorer hicFindTADs. The green track shows a filled-out curve representation of the data from H3K36me3 histone mark, provided as a bigwig file along with an additional horizontal threshold line as well as a scale bar indicating the distance between two different peaks of interest. The blue arcs show artificially created links that could be contacts between different CP190 peaks. Finally, the last track is a gene track of dm3, available in bed format. The configuration file is available in Supplementary Section S3

a bed file with multiple genomic regions. PGT supports several output formats such as *eps, pdf, pgf, png, ps, raw, rgba, svg* and *svgz*, which offers a broad degree of flexibility. The tool can easily generate the requested figure by running a single command line as has been presented below.

```
$ pyGenomeTracks --tracks tracks.ini --region \
chr2L:8050000-8300000 --outFileName image.pdf
```

Moreover, for users who prefer a graphical interface, PGT is available as a tool on the European Galaxy server https://usegalaxy.eu, and can be installed on any local Galaxy instance (Afgan *et al.*, 2016) via ToolShed (see Supplementary Fig. S1).

To illustrate the functionality of PGT, Figure 1 provides an example of a multitrack visualization from an integrated multiomics screen (Ramírez *et al.*, 2018) generated with PGT version 3.5. Please refer to the Supplementary Data for additional examples and a detailed documentation is available on https://pygenometracks.readthedocs.io.

3 Conclusion

With PGT, it is possible to integrate multiple data sources from a wide variety of genomics assays and to generate publication ready plots. The presence of a configuration file (.ini file) provides flexibility to easily change or reorder the data tracks. To ensure maximal reproducibility, PGT also uses conda, which allows specific versions of all dependent tools to be flexibly chosen. This approach enables other researchers to readily reproduce the images and validate them swiftly. The supported output file formats, such as eps, svg or png, offer a high degree of freedom to generate plots in standardized formats which are required by a variety of major journals. PGT can be used as a command line or Galaxy-based tool. The latter is available on https://usegalaxy.eu with all configuration options, or it can be installed on any local Galaxy instance. It provides an easy way for users to run their analysis on Galaxy in a transparent and reproducible way. PGT presents a well-structured approach for generating genomics data plots and can also be used in automated workflow processing.

Funding

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 192904750 – SFB 992. German Federal Ministry of Education and Research [031 A538A de. NBI-RBC awarded to R.B.; 031 L0101C de. NBI-epi awarded to B.G.]. R.B. was supported by the German Research Foundation (DFG) under Germany's Excellence Strategy [CIBSS-EXC-2189-Project ID 390939984]. We acknowledge funding from the Ecole Polytechnique Fédérale (EPFL, Lausanne). LD was paid by the European Research Council (ERC) grant RegulHox (No 588029).

Conflict of Interest: none declared.

References

Afgan,E. *et al.* (2016) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Res.*, **44**, W3–W10.

Bhardwaj,V. *et al.* (2019) snakepipes: facilitating flexible, scalable and integrative epigenomic analysis. *Bioinformatics*, **35**, 4757–4759.

Gehlenborg,N. *et al.* (2010) Visualization of omics data for systems biology. *Nat. Methods*, **7**, S56–S68.

Grüning,B. *et al.*; The Bioconda Team. (2018) Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nat. Methods*, **15**, 475–476.

Kerpedjiev,P. *et al.* (2018) HiGlass: web-based visual exploration and analysis of genome interaction maps. *Genome Biol.*, **19**, 1–12.

Li,L. *et al.* (2015) Widespread rearrangement of 3D chromatin organization underlies polycomb-mediated stress-induced silencing. *Mol. Cell*, **58**, 216–231.

Ramírez,F. *et al.* (2018) High-resolution tads reveal DNA sequences underlying genome organization in flies. *Nat. Commun.*, **9**, 1–15.

Robinson,J.T. *et al.* (2011) Integrative genomics viewer. *Nat. Biotechnol.*, **29**, 24–26.

Appendix A

A.1 Galaxy HiCExplorer 3: a web server for reproducible Hi-C, capture Hi-C and single-cell Hi-C data analysis, quality control and visualization

Supplementary material: Galaxy HiCExplorer 3: a web server for reproducible Hi-C, capture Hi-C and single-cell Hi-C data analysis, quality control and visualisation

1 HiCExplorer methods

1.1 hicNormalize

In the HiCExplorer software the term *normalize* is used in the context of adjusting the interaction values to the same value range. To achieve this, three methods are offered.
In the following the interaction matrix is defined as:

$$ICM = \begin{bmatrix} ic_{00} & \cdots & ic_{0n} \\ \vdots & \dots & \vdots \\ ic_{n0} & \cdots & ic_{nn} \end{bmatrix} \tag{1}$$

1.1.1 Smallest

This mode normalizes the read coverage of all given matrices ICM_l to the sum of the lowest read coverage present:

$$read\ coverage_l = \sum ICM_l \tag{2}$$

$$min\_index = argmin(read\ coverage) \tag{3}$$

$$adjust\_factor = \frac{\sum ICM_l}{\sum ICM_{min\_index}} \tag{4}$$

$$ic\_l_{j,k} = \frac{ic\_l_{j,k}}{adjust\_factor} \tag{5}$$

1.1.2 Norm range

The *norm range* mode normalizes each interaction matrix ICM independently to a 0 to 1 value range.

$$max\_value = \max(ICM) \tag{6}$$

$$min\_value = \min(ICM) \tag{7}$$

$$min\_max\_difference = max\_value - min\_value \tag{8}$$

$$ic_{i,j} = \frac{ic_{i,j} - min\_value}{min\_max\_difference} \tag{9}$$

1.1.3 Multiplicative mode

The *multiplicative mode* gives the option to multiply each interaction with a user defined value.

$$ic_{i,j} = ic_{i,j} * value \tag{10}$$

1.2 hicAverageRegions

hicAverageRegions takes as input a bed file with regions of interest. The user can define if the start, end or the center (end - start) should be considered as the reference point. Based on a reference point $icm_{i,j}$ and the user given range r, a sub-matrix per reference point is extracted:

$$ICM\_sub = \begin{bmatrix} ic_{i-r,j-r} & \cdots & ic_{i-r,j+r} \\ \vdots & icm_{i,j} & \vdots \\ ic_{i+r,j-r} & \cdots & ic_{i+r,j+r} \end{bmatrix} \tag{11}$$

All sub-matrices ICM_sub are added to one matrix and is divided by the number of sub-matrices. This resulting matrix is called the average region matrix.

1.3 hicPlotSVL

For each chromosome of each interaction matrix the short vs. long range distance ratio is computed as:

$$short\_range = \sum_{i=0}^{i<minRange} \sum_{j=0}^{j<minRange} ic_{i,j} \tag{12}$$

$$long\_range = \sum_{i=minRange}^{i<maxRange} \sum_{j=minRange}^{j<maxRange} ic_{i,j} \tag{13}$$

$$svl = \frac{short\_range}{long\_range} \tag{14}$$

All short vs long range values are ordered by the chromosomes and between different samples a Wilcoxon rank-sum is computed. The rank-sum test determines if two samples have a different ratio (small p-value) or not.

1.4 hicCompartmentalization

This tool helps in studying the polarization of the compartments by ordering the values of $PC1$ in an ascending manner and re-ordering their corresponding bins on the observed/expected matrix, we call it the 'polarization matrix'. With this method, all the bins with negative $PC1$ values (representative of inactive compartment (B)) should be shifted to the top/left corner of polarization matrix and all those with positive values (representative of active compartment (A)) should be moved at bottom/right cornet of the matrix. If there will be a clear compartmentalization on the given genome, it is expected that the sum of the contacts in these two corners be larger than the sum of the contacts on the two other corners of the matrix which contain the in-between compartments contacts.

The ascending ordering happens after dividing the values into a given number of quantiles, therefore the polarization matrix dimension is $quantiles * quantiles$. To make the polarization plot by counting the contacts of the polarization matrix, we apply the following method on each bin of the polarization matrix:

$$\begin{gathered} within\_comps = \sum matrix[0:b, 0:b] + \sum matrix[q-b:q, q-b:q]; \\ between\_comps = \sum matrix[0:b, q-b:q] + \sum matrix[q-b:q, 0:b]; \\ within\_to\_between = \tfrac{within\_comps}{between\_comps} \end{gathered} \tag{15}$$

Where b is the $bin + 1$ and q is the given number of quantiles.

2 Capture Hi-C

2.1 Background model

The user given reference point with the up- and downstream given distance is defined as the viewpoint. A relative distance is defined as the distance up- or downstream to a reference point.

To build the background model, all viewpoints from all samples are considered. Per relative distance rd over all viewpoints v one continuous negative binomial distribution is fitted:

$$X_{rd} \sim cNB_{rd}(r_{rd}, p_{rd}) \tag{16}$$

The continuous negative binomial distribution is created by exchanging the binomial coefficient of the probability mass function by gamma functions. Continuous negative binomial functions are used by edgeR [1, 2]; moreover, it was discussed on the website stackexchange[1] how to generalize negative binomial functions. This continuous negative binomial function is also used in the loop detection[2].

$$f(k, r, p) = \frac{\Gamma(k + r)}{\Gamma(k + 1) * \Gamma(r)} p^k (1 - p)^r \tag{17}$$

The p-value of an interaction i at the relative distance rd is given as:

$$pvalue\ of\ i = P(x \geq i) = 1 - \sum_{k=0}^{i-1} f_{rd}(k, r_{rd}, p_{rd}) \tag{18}$$

Additionally, the mean background per relative distance rd over all viewpoints v is computed.

2.2 Significant interaction detection

The detection of significant interactions is accomplished in three steps:

1. Loose p-value: all interactions which have this p-value or less are accepted as a candidate

2. x-Fold: all interactions with a interaction value $value * x - fold > mean\_background_{rd}$ are accepted as a candidate

3. For all interactions: if their neighbor interaction is a candidate too, consider their interaction as one and add them together. Add neighboring elements together as long they fulfil condition 1 or 3. Recompute all p-values for the new interaction and accept as significant if their p-value from cNB_{rd} is $p - value \leq threshold.$

2.3 Differential test

All interactions of interest are tested with Fisher's exact test or the chi^2 contingency test. Values for the test are always the interaction value of the reference point and the interaction value of the interaction of interest. These values are used to test against a second sample (e.g. wild type).

3 scHiCExplorer methods

scHiCExplorer uses traditional clustering algorithms which require a two dimensional matrix as an input but the nature of a Hi-C matrix is that it is already present in two dimensions; leading to three dimensions.
Let all single-cell Hi-C matrices ICM be given as $n \times n$ and each pixel as $icm_{k,l}$. Each Hi-C interaction matrix is flattened to one dimension and is stacked together with all the other flattened matrices to one two dimensional matrix $scICM$ where each row i resents therefore a cell, each feature j an interaction. An interaction at $scICM_{i,j}$ is equal to the single-cell Hi-C matrix of cell i and the interaction at position j is $j = (k * n) + l$. This results in the matrix $scICM$ with $i \times (n * n)$.

3.1 Dimension reduction

The raw clustering approaches of *scHicCluster* do not use any dimension reduction technique and operate directly on the matrix $scICM$. This can be problematic because the number of dimensions can go to the millions or even billions, depending on the resolution on the Hi-C matrices. Moreover, the clustering results are bad. Please consider Supplement Figure 1a, 1b and the cluster profile Supplement Figure 4a, 4b.

3.1.1 PCA

To compute the principle components, first the covariance matrix on $scICM$ is generated and then the eigenvectors are calculated on this matrix. Only the first i componets are considered, resulting in a dimension reduced matrix $scPCA$ of $i \times i$.

[1] https://stats.stackexchange.com/questions/310676/continuous-generalization-of-the-negative-binomial-distribution/311927
[2] https://www.biorxiv.org/content/early/2020/03/06/2020.03.05.979096

3.1.2 K-nearest neighbors

The k-nearest neighbors graph approach computes on $scICM$ for each cell i the i-nearest neighbors based on the euclidean distance. With this approach the dimensions can be reduced to $i \times i$ and each pixel i, j represents the euclidean distance between the two cells. However, the user can define a different value for the k-nn and is therefore able to reduce the compute time.

3.1.3 Approximate nearest neighbors: MinHash

The MinHash approach computes approximate nearest neighbors via an approximation of the Jaccard similarity. Moreover, it offers the option to precompute the Jaccard similarity and based on the subset of nearest neighbors the exact nearest neighbors via the euclidean distance can be computed. Please consider Wolff 2020: Approximate k-nearest neighbors graph for single-cell Hi-C dimensional reduction with MinHash[3] for more details.

3.1.4 Short vs long range ratio

For each single-cell Hi-C matrix ICM with $n \times n$ the short vs long range ration per chromosome is computed as described in Section 1.3. Let the number of all present single-cell matrices be i. All ratios per chromosome of all single-cell Hi-C matrices are stacked together resulting in a dimension reduced matrix $scSVL$ with $i \times |chromosomes|$ dimensions.

3.1.5 A/B compartments

For each single-cell Hi-C matrix ICM with $n \times n$ the A/B compartments are computed per chromosome and the first principal component is taken as the vector describing the matrix. Let the number of all present single-cell matrices be i. All first principal components of all single-cell Hi-C matrices are stacked together resulting in a dimension reduced matrix $scABC$ with $i \times n$ dimensions.

3.2 Clustering

As clustering methods k-means and spectral clustering are offered. Please consider the following Figures 1, 2, 3 and 4 for a comparison of the dimension reduction techniques and the quality of the clustering.

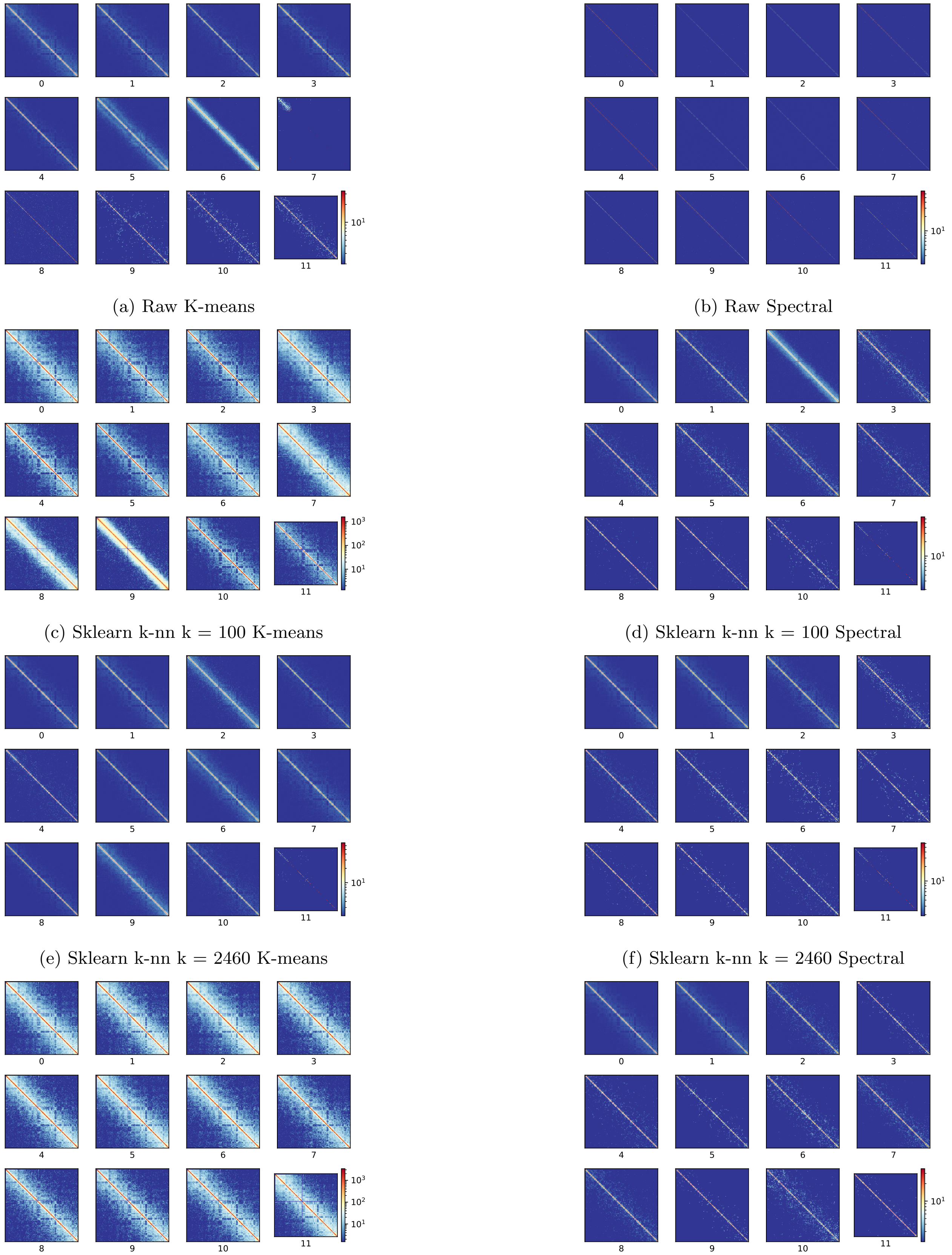
[3] http://dx.doi.org/10.1101/2020.03.05.978569

References

[1] Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 2010.

[2] Davis J McCarthy, Yunshun Chen, and Gordon K Smyth. Differential expression analysis of multifactor rna-seq experiments with respect to biological variation. *Nucleic acids research*, 40(10):4288–4297, 2012.

[3] Takashi Nagano, Yaniv Lubling, Csilla Várnai, Carmel Dudley, Wing Leung, Yael Baran, Netta Mendelson Cohen, Steven Wingett, Peter Fraser, and Amos Tanay. Cell-cycle dynamics of chromosomal organization at single-cell resolution. *Nature*, 547(7661):61, 2017.



(a) Raw K-means

(b) Raw Spectral

(c) Sklearn k-nn k = 100 K-means

(d) Sklearn k-nn k = 100 Spectral

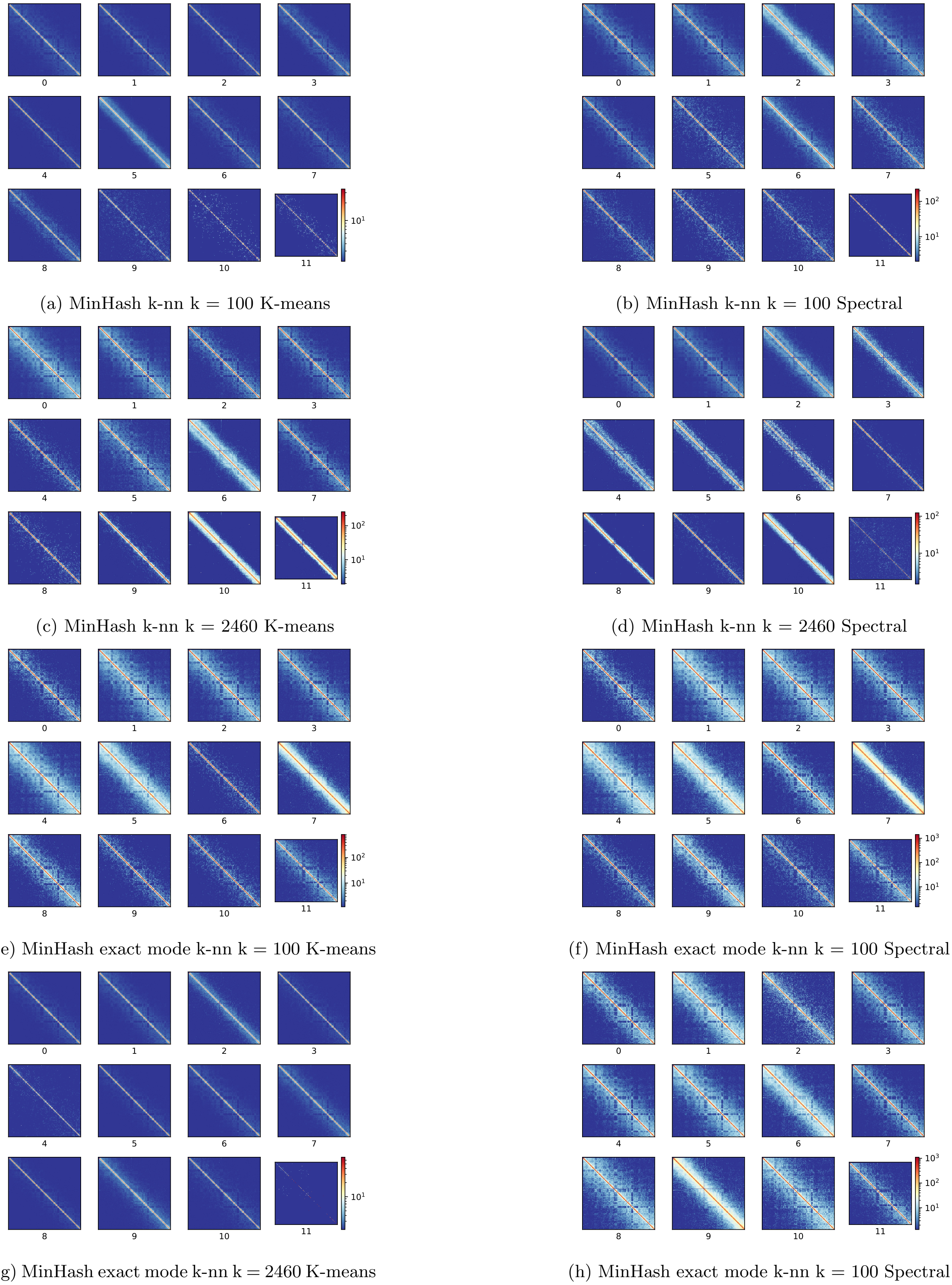
(e) Sklearn k-nn k = 2460 K-means

(f) Sklearn k-nn k = 2460 Spectral

(g) Principal component analysis K-Means

(h) Principal component analysis Spectral

Figure 1: Consensus matrices of the different clusters on 2460 cells from [3] Diploid cells, chromosome 1. K-Means and spectral clustering was used on the different dimension reduced scHi-C matrices. Results from scHicCluster on raw data (1a, 1b) and on dimension reduced data with *k-nearest neighbors* (1c, 1d, 1e, 1f) and *PCA* (1g, 1h).



(a) MinHash k-nn k = 100 K-means

(b) MinHash k-nn k = 100 Spectral

(c) MinHash k-nn k = 2460 K-means

(d) MinHash k-nn k = 2460 Spectral

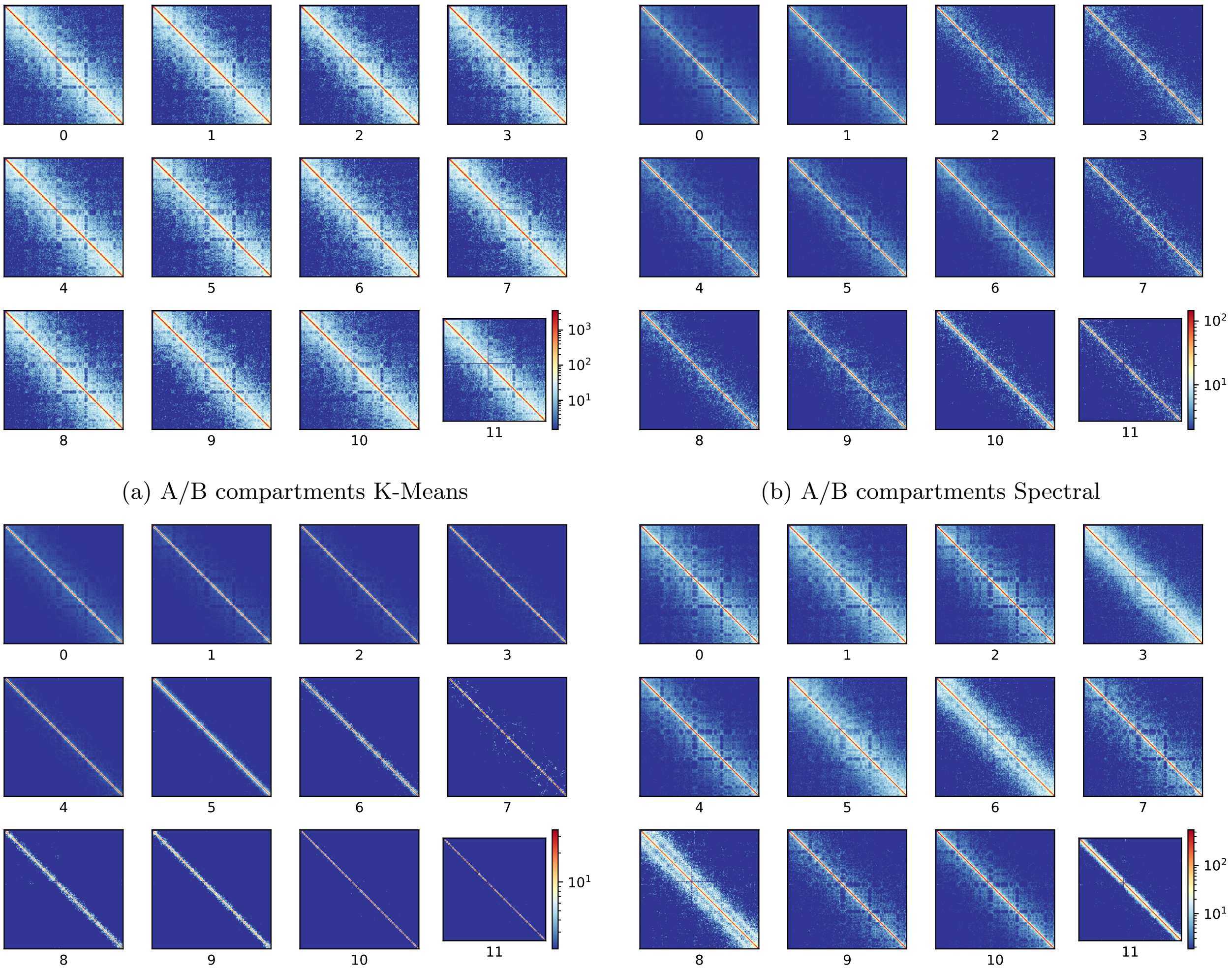
(e) MinHash exact mode k-nn k = 100 K-means

(f) MinHash exact mode k-nn k = 100 Spectral

(g) MinHash exact mode k-nn k = 2460 K-means

(h) MinHash exact mode k-nn k = 100 Spectral

Figure 2: Consensus matrices of the different clusters on 2460 cells from [3] Diploid cells, chromosome 1. K-Means and spectral clustering were applied on results from scHicClusterMinHash.



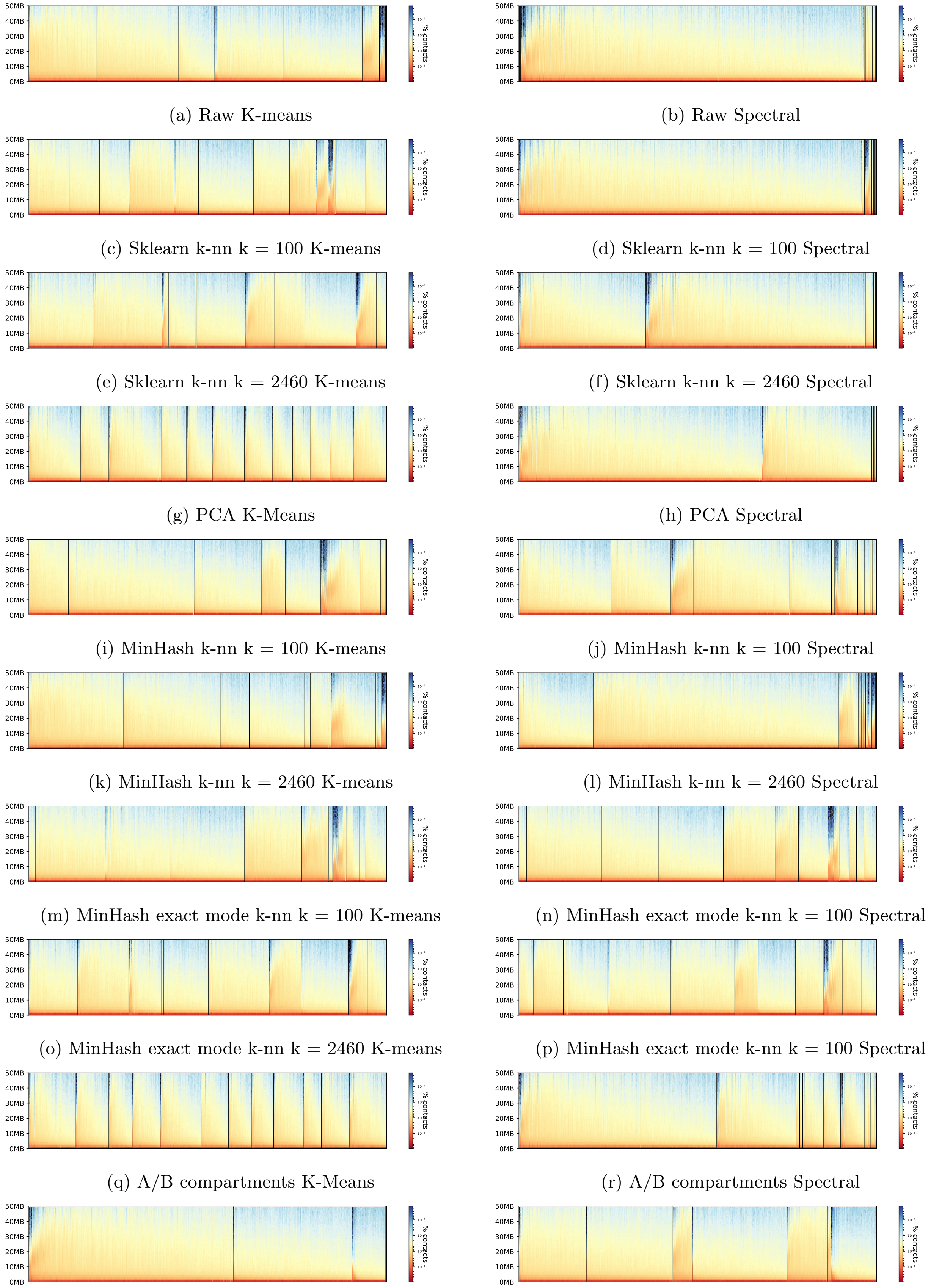
(a) A/B compartments K-Means

(b) A/B compartments Spectral

(c) SVL K-Means

(d) SVL Spectral

Figure 3: Consensus matrices of the different clusters on 2460 cells from [3] Diploid cells, chromosome 1. K-Means and spectral clustering were applied on results from scHicClusterCompartments (3a, 3b) and scHicClusterSVL (3c, 3d).



(a) Raw K-means

(b) Raw Spectral

(c) Sklearn k-nn k = 100 K-means

(d) Sklearn k-nn k = 100 Spectral

(e) Sklearn k-nn k = 2460 K-means

(f) Sklearn k-nn k = 2460 Spectral

(g) PCA K-Means

(h) PCA Spectral

(i) MinHash k-nn k = 100 K-means

(j) MinHash k-nn k = 100 Spectral

(k) MinHash k-nn k = 2460 K-means

(l) MinHash k-nn k = 2460 Spectral

(m) MinHash exact mode k-nn k = 100 K-means

(n) MinHash exact mode k-nn k = 100 Spectral

(o) MinHash exact mode k-nn k = 2460 K-means

(p) MinHash exact mode k-nn k = 100 Spectral

(q) A/B compartments K-Means

(r) A/B compartments Spectral

(s) SVL K-Means

(t) SVL Spectral

Figure 4: Cluster profile of the different clusters on 2460 cells from [3] Diploid cells. K-Means and spectral clustering were applied on the different dimension reduced scHi-C matrices. Results from scHicCluster on raw (4a, 4b) with knn (4c, 4d, 4e, 4f) mode, PCA (4g, 4h); scHicClusterMinHash (4i, 4j, 4k, 4l, 4m, 4n, 4o, 4p); scHicClusterCompartments (4q, 4r) and scHicClusterSVL (4s, 4t).

Method	Runtime	Memory
Raw and K-Means	1:52 h	33 GB
Raw and Spectral	3:02 min	6.7 GB
PCA and K-Means	7:27 min	220 GB
PCA and Spectral	7:39 min	220 GB
sklearn k-nn k = 100 and K-means	2:11 min	6.7 GB
sklearn k-nn k = 100 and Spectral	3:40 min	6.7 GB
sklearn k-nn k = 2460 and K-means	18:41 min	6.7 GB
sklearn k-nn k = 2460 and Spectral	3:44 min	6.7 GB
MinHash k = 100 and K-means	2:18 min	6.7 GB
MinHash k = 100 and Spectral	3:27 min	6.7 GB
MinHash k = 2460 and K-means	6:59 min	6.7 GB
MinHash k = 2460 and Spectral	3:20 min	6.7 GB
MinHash exact mode k = 100 and K-means	4:25 min	6.7 GB
MinHash exact mode k = 100 and Spectral	2:56 min	6.7 GB
MinHash exact mode k = 2460 and K-means	1:12 h	6.7 GB
MinHash exact mode k = 2460 and Spectral	1:03 h	6.7 GB
A/B compartments K-means	40:52 min	6.7 GB
A/B compartments Spectral	1:10 h h	6.7 GB
SVL K-means	1:51 min h	6.7 GB
SVL compartments Spectral	1:52 min h	6.7 GB

(a) Data: 1 MB resolution, with 2460 cells.

Method	Runtime	Memory
Raw and K-Means	-	> 1 TB
Raw and Spectral	-	> 1 TB
PCA and K-Means	-	> 1 TB
PCA and Spectral	-	> 1 TB
sklearn k-NN and K-Means	-	> 1 TB
sklearn k-NN and Spectral	-	> 1 TB
MinHash k = 2508 and K-Means	1:13 h	53 GB
MinHash k = 2508 and Spectral	1:04 h	53 GB
MinHash exact mode k = 2508 and K-Means	3:19 h	53 GB
MinHash exact mode k = 2508 and Spectral	3:11 h	53 GB
MinHash exact mode k = 50 and K-Mans	1:08 h	53 GB
MinHash exact mode k = 50 and Spectral	1:03 h	53 GB
MinHash exact mode k = 200 and K-Mans	1:17 h	53 GB
MinHash exact mode k = 200 and Spectral	1:09 h	53 GB
A/B compartments K-means	> 14 days	- GB
A/B compartments Spectral	> 14 days	- GB
SVL K-means	1:11 h	6.7 GB
SVL compartments Spectral	1:14 h	6.7 GB

(b) Data: 10 kb resolution, with 2460 cells. The raw matrix, PCA and sklearn k-nn methods requested more than the available 1 TB of memory and could not be computed; A/B compartments were computing for 14 days and the computation has been canceled by us.

Table 1: Data from [3] Diploid cells, with 12 clusters. For clustering K-means and spectral clustering were used, MinHash with 800 hash functions. All results computed on 2x XEON E5-2630 v4 @ 2.20GHz 2x 10 cores / 2x 20 threads, 1 TB memory.

A.2 Appendix for Robust and efficient single-cell Hi-C clustering with approximate k-nearest neighbor graphs

Supplementary material: Robust and efficient single-cell Hi-C clustering with approximate k-nearest neighbor graphs

Joachim Wolff [1*], Rolf Backofen [1,2], Björn Grüning [1]

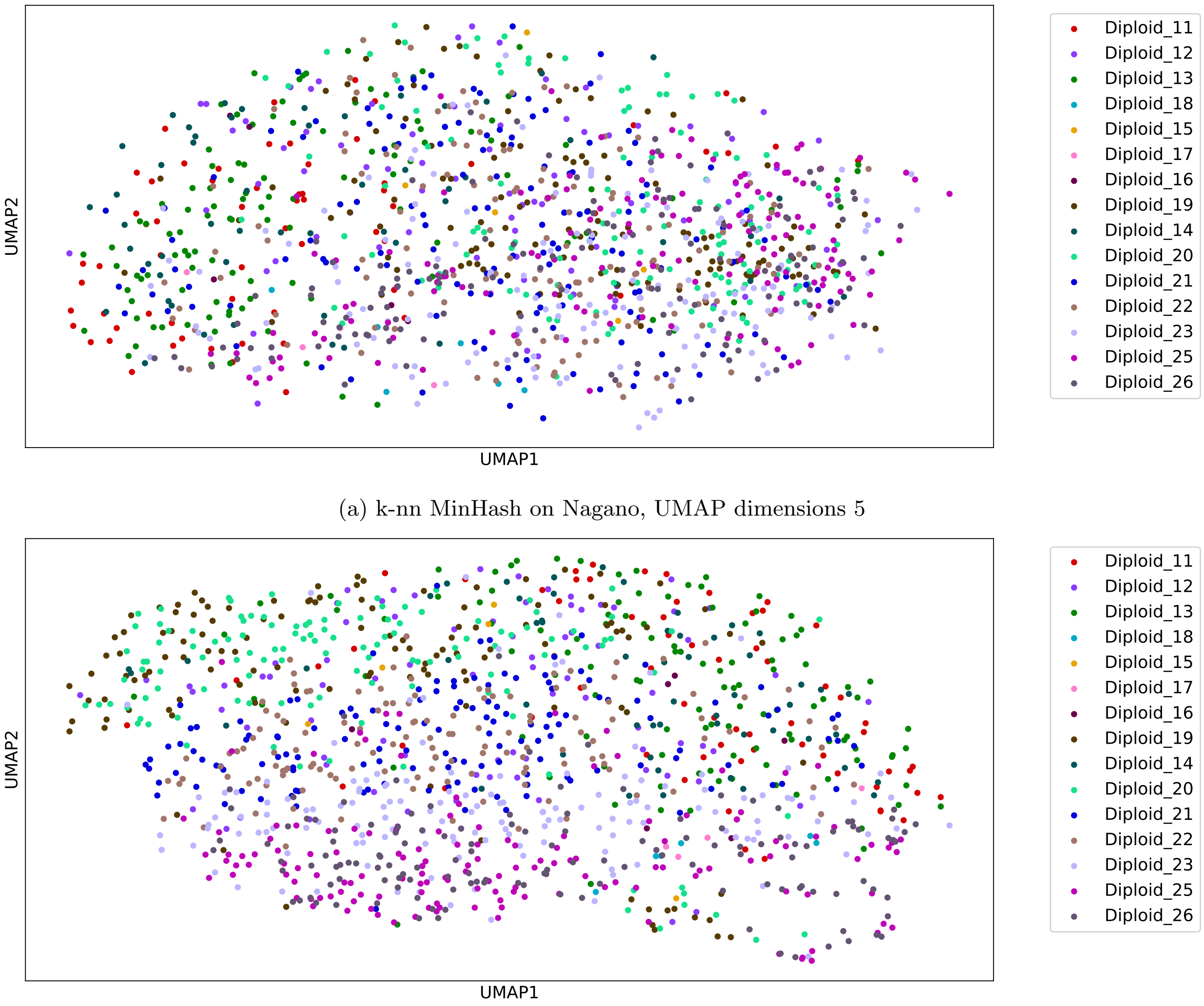
[1] Bioinformatics Group, Department of Computer Science, University of Freiburg, Georges-Köhler-Allee 106, 79110 Freiburg, Germany

[2]Signalling Research Centre CIBSS, University of Freiburg, Schänzlestr. 18, 79104 Freiburg, Germany

*To whom correspondence should be addressed.

1 Batch effects

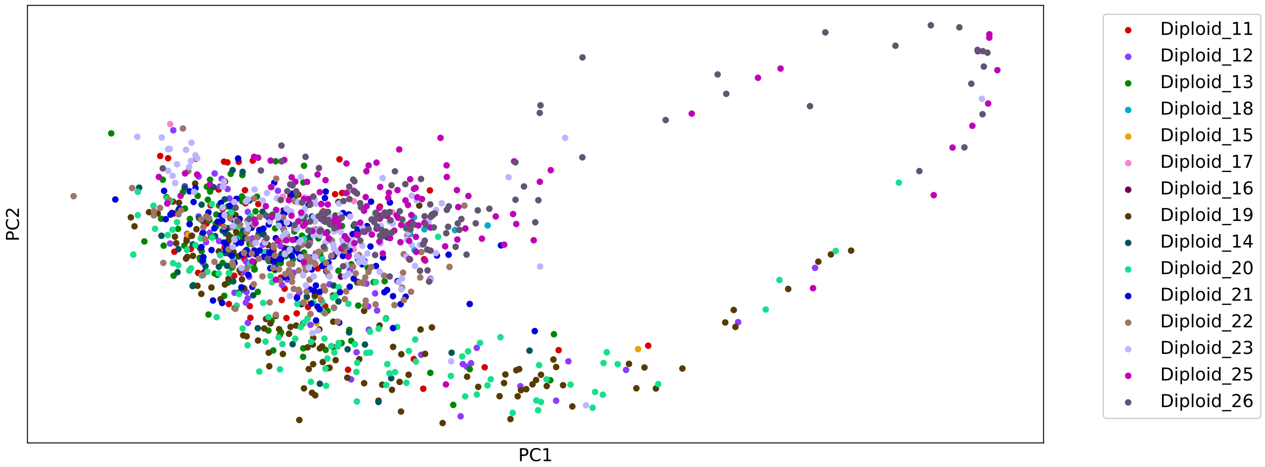
1.1 Nagano 1 Mb



(a) k-nn MinHash on Nagano, UMAP dimensions 5

(b) k-nn MinHash on Nagano, UMAP dimensions 2

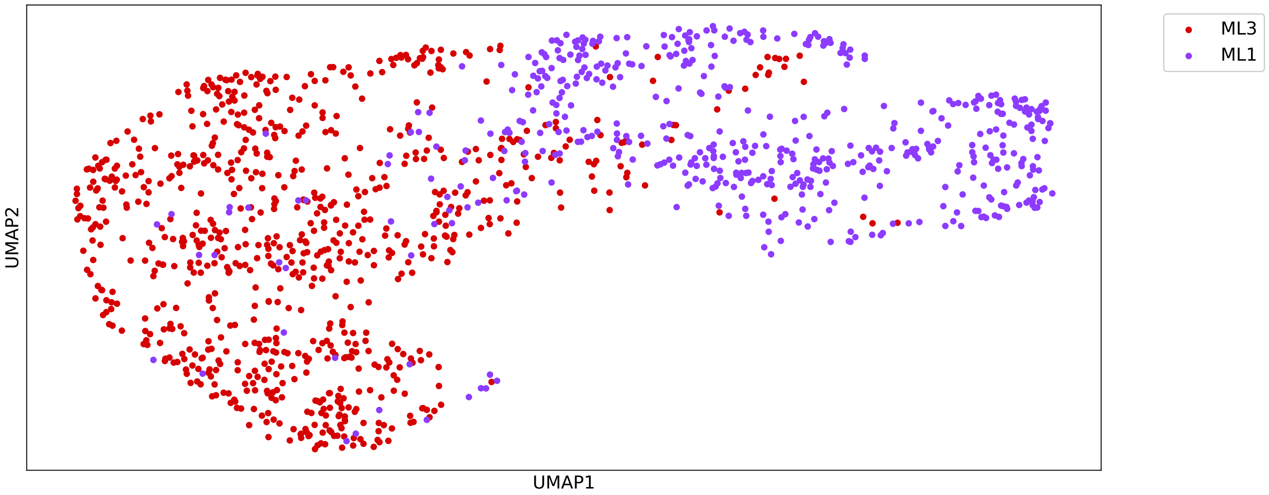
Figure S 1: Batch effects on Nagano 1 MB



(c) Zhou’s scHiCluster on Nagano

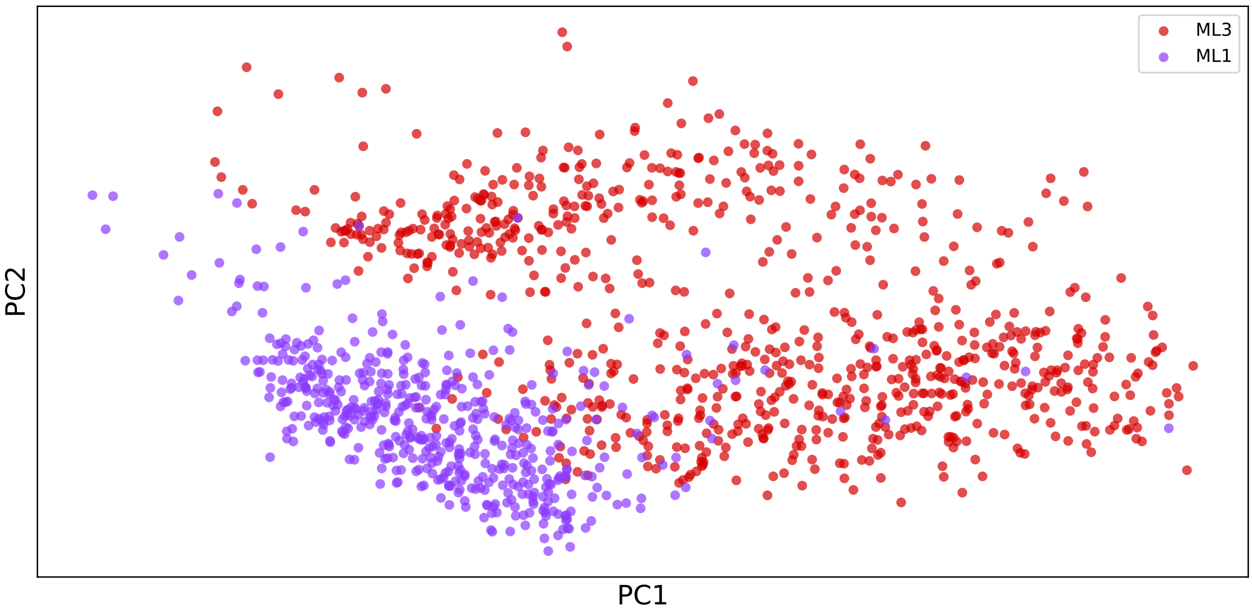
Figure S 1: Batch effects on Nagano 1 MB

1.2 Ramani 1 Mb



(a) k-nn MinHash on Ramani ML1 and ML3

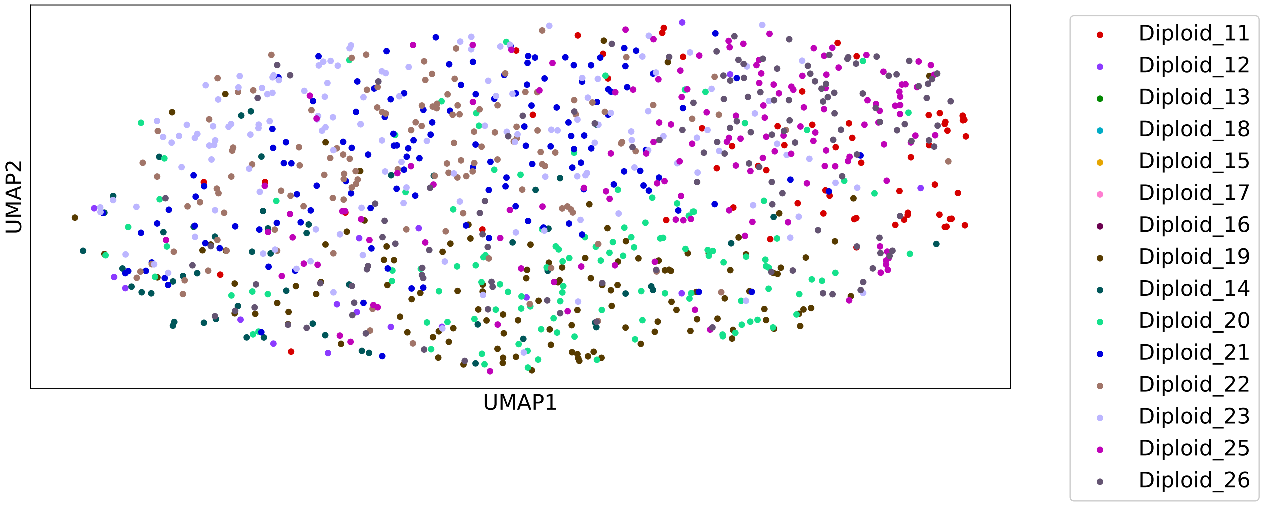
Figure S 2: Batch effects



(b) Zhou's scHiCluster ML1 ML3

Figure S 2: Batch effects

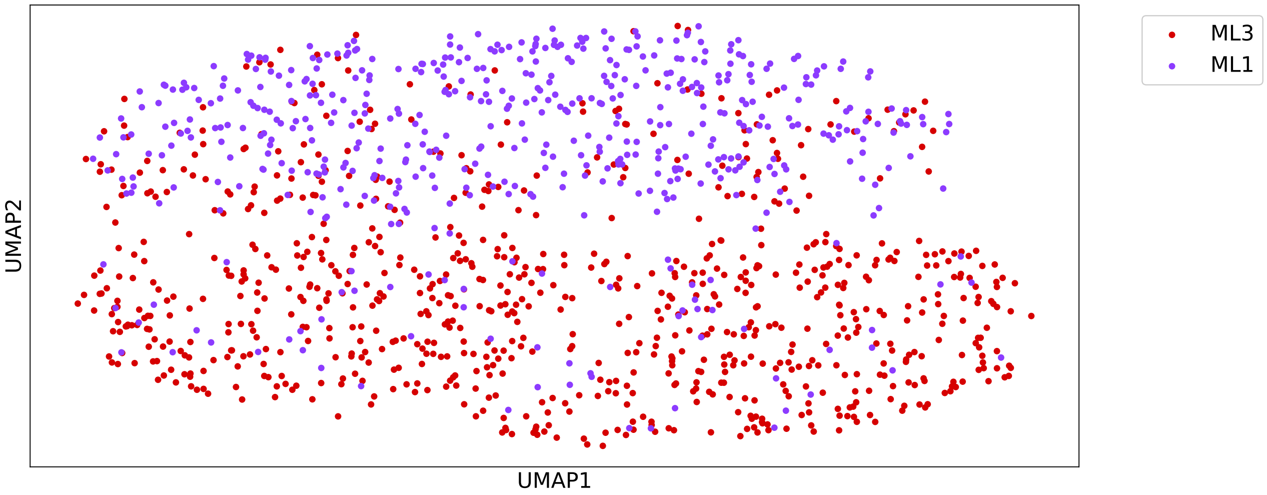
1.3 Nagano 10 kb



(a) k-nn MinHash on Nagano 10 kb resolution

Figure S 3: Batch effects

1.4 Ramani 10 kb



(a) k-nn MinHash on Ramani 10 kb resolution

Figure S 4: Batch effects

2 Density distributions for single-cell interaction matrices

The density of a cell is measured by the number of binary contacts a cell has vs. the number of contacts it could have, i.e., the number of non-zero values of a matrix vs. all values. Single-cell Hi-C matrices have the disadvantage, especially in comparison to regular Hi-C, that their read coverage with around 100,000 reads per cell is low. The majority of the contacts are recorded in close proximity, i.e., around the main diagonal. For these reasons, the density measure is restricted to (possible) interaction pairs within a distance of 30 Mb.

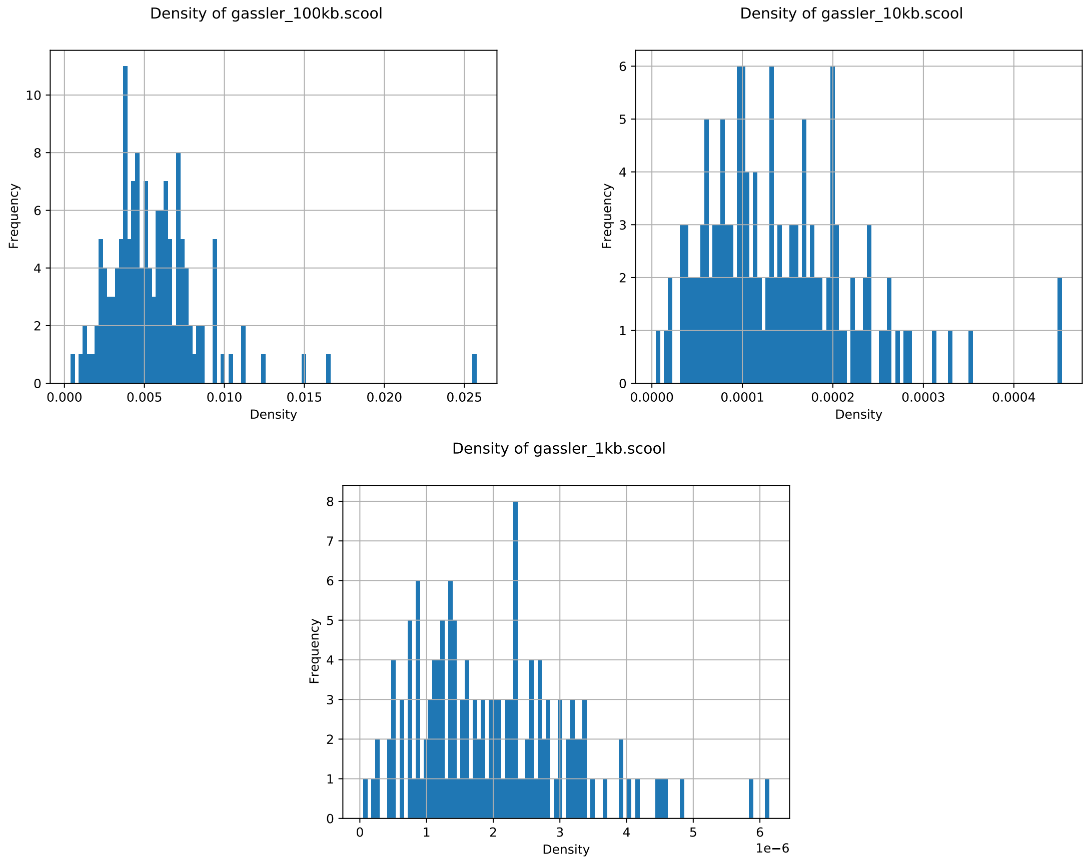


Figure S 5: Density distributions for 100kb, 10kb and 1kb for 144 cells from Gassler *et al.* (2017)

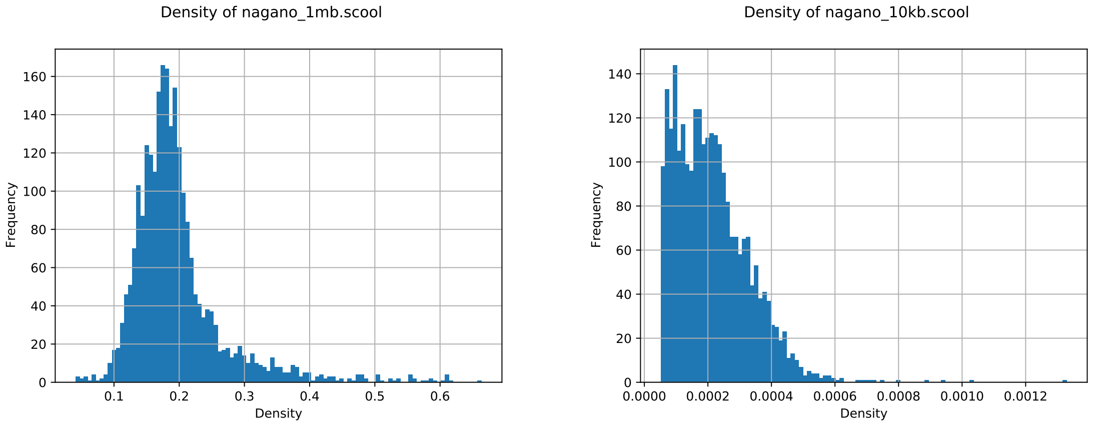


Figure S 6: Density distributions for 1Mb and 10kb for 2472 cells from Nagano *et al.* (2017).

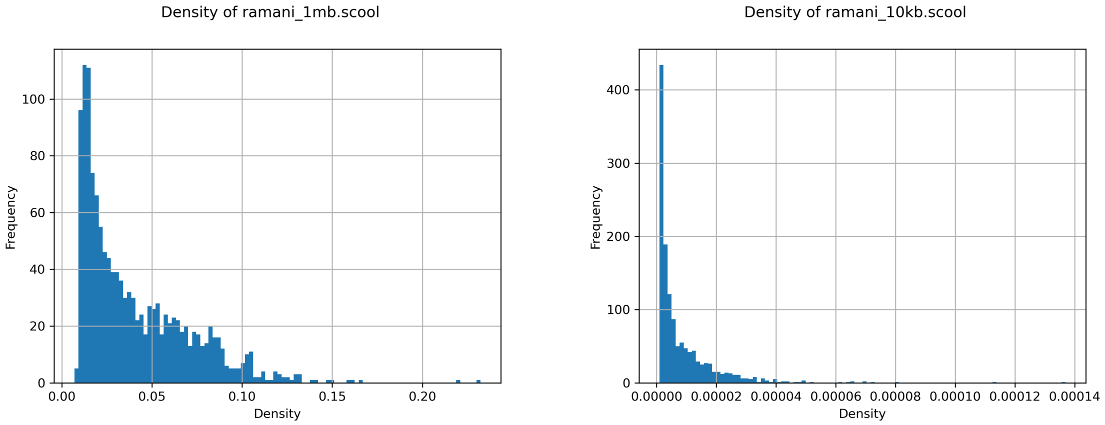


Figure S 7: Density distributions for 1Mb and 10kb for 1329 cells from Ramani *et al.* (2017).

3 MinHash collision statistics

3.1 Collisions per cell

The here shown collision statistics are for different interaction matrix resolutions from Nagano *et al.* (2017) and Gassler *et al.* (2017).

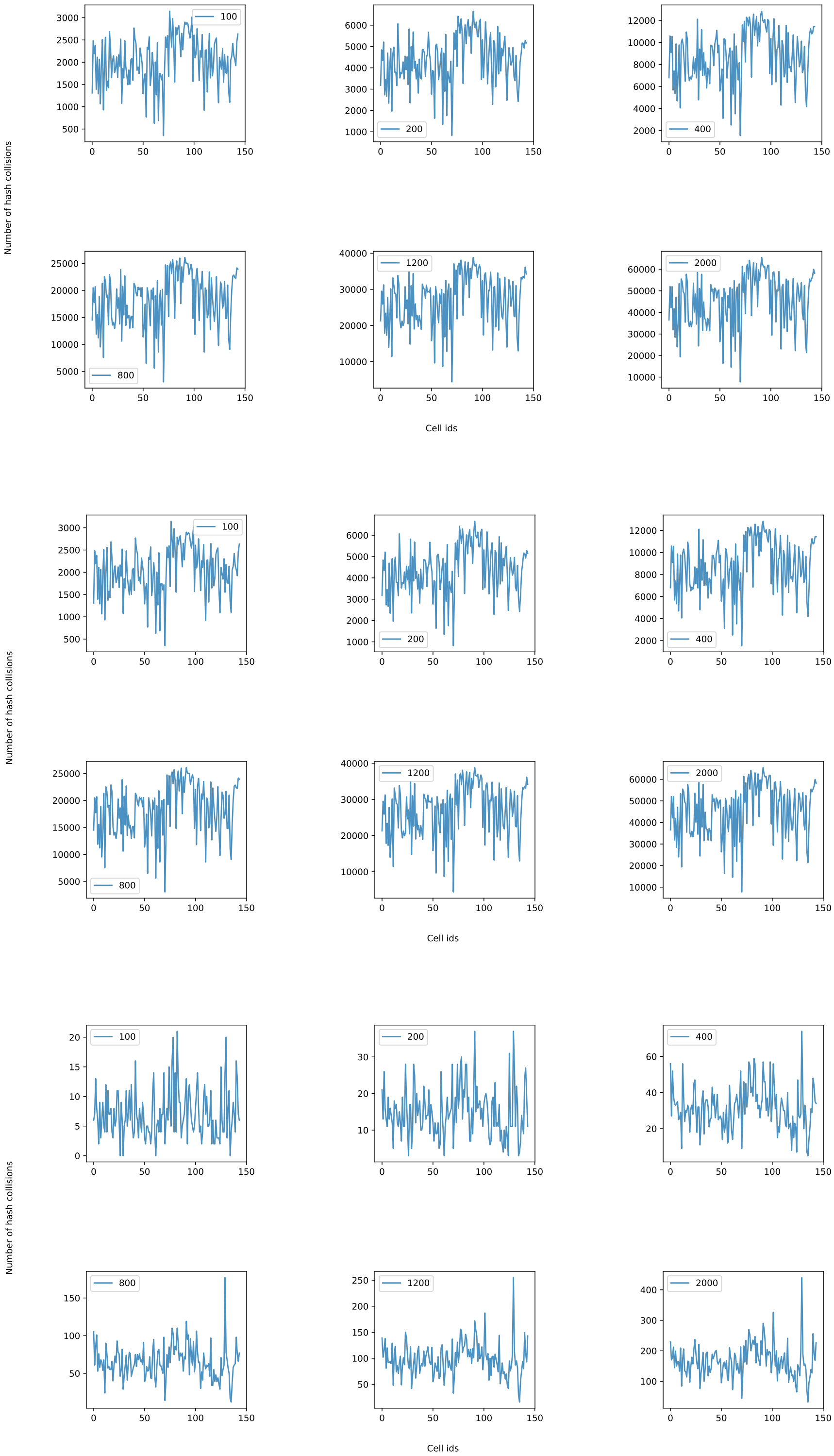


Figure S 8: Number of hash collisions per cell for 100kb (top), 10kb (middle) and 1kb (bottom) for 144 cells from Gassler *et al.* (2017). The number of hash collisions is shown for 100, 200, 400, 800, 1200 and 2000 hash functions. One collision occurs if two cells have the same hash value for a hash function.

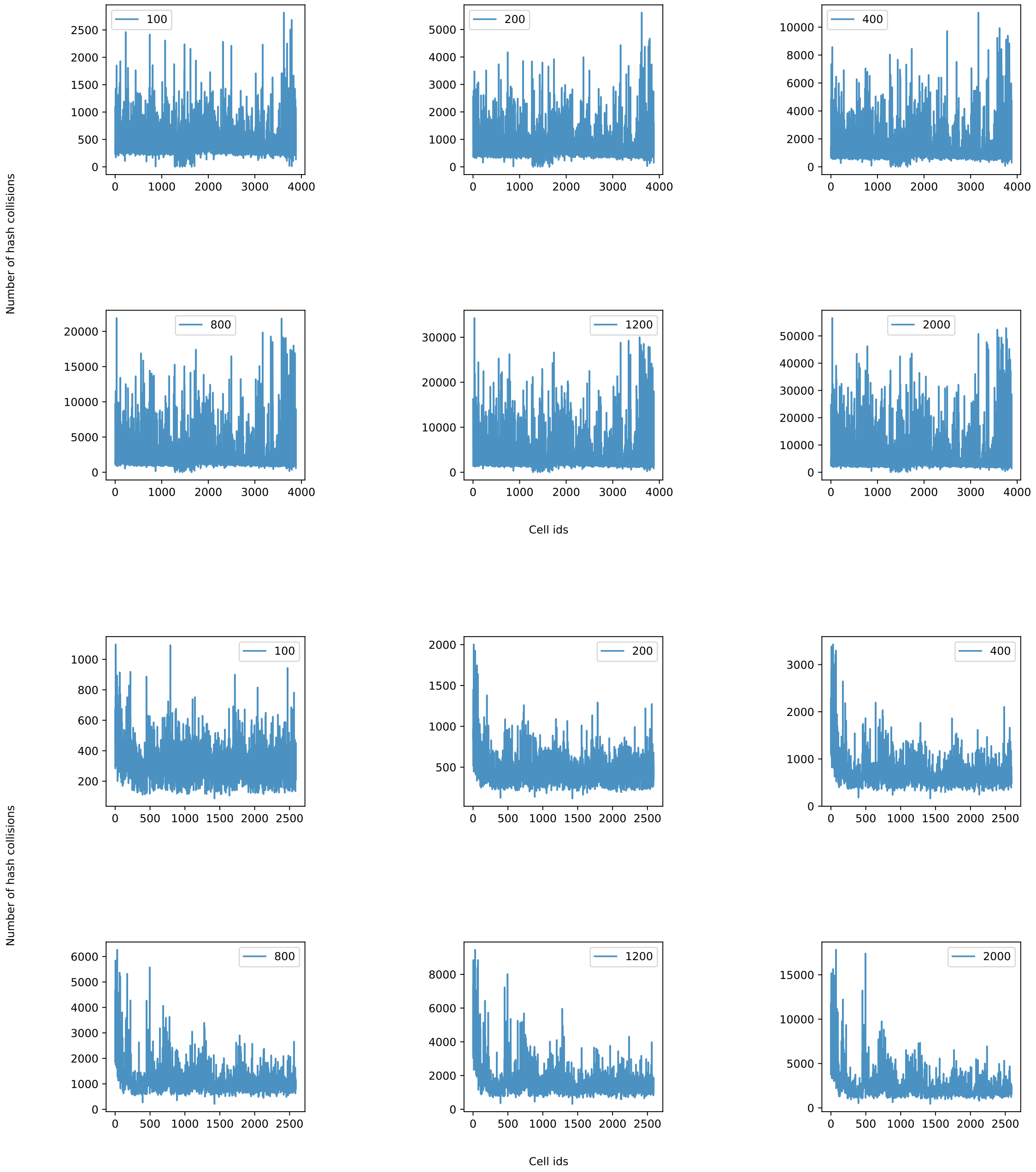
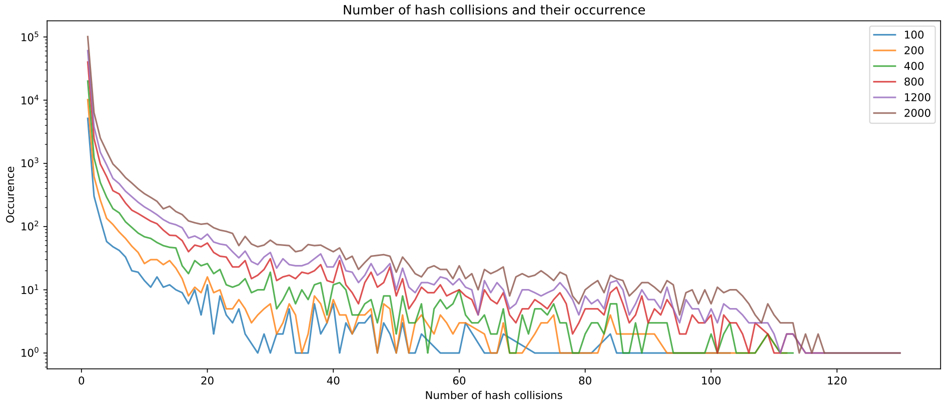
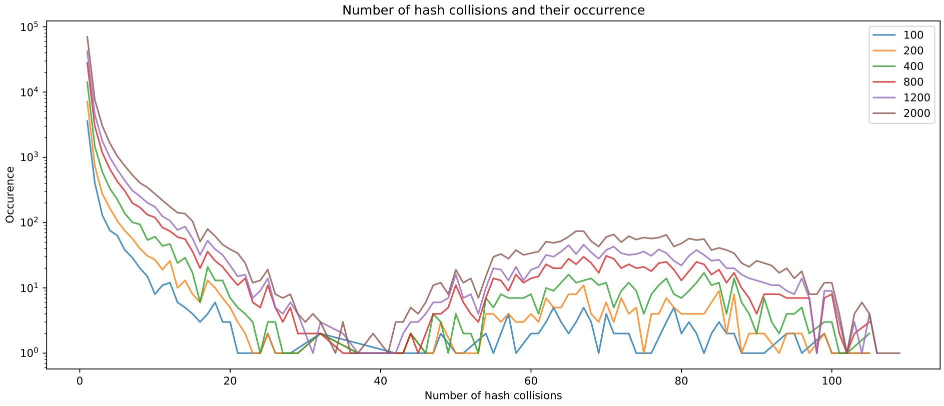


Figure S 9: Number of hash collisions per cell for 1Mb (top) and 10kb (bottom) cells from Nagano *et al.* (2017). The number of hash collisions is shown for 100, 200, 400, 800, 1200 and 2000 hash functions. One collision occurs if two cells have for one hash function the same hash value.

3.2 Collision occurrences

The collision occurrences statistics maps the number of collisions for a hash value of a hash function (x-axis) with the occurrences of the number of collisions' overall hash functions and hash values.





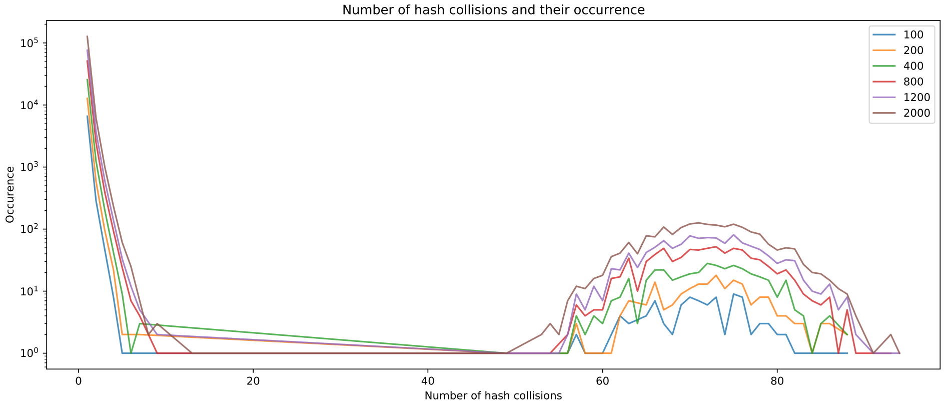
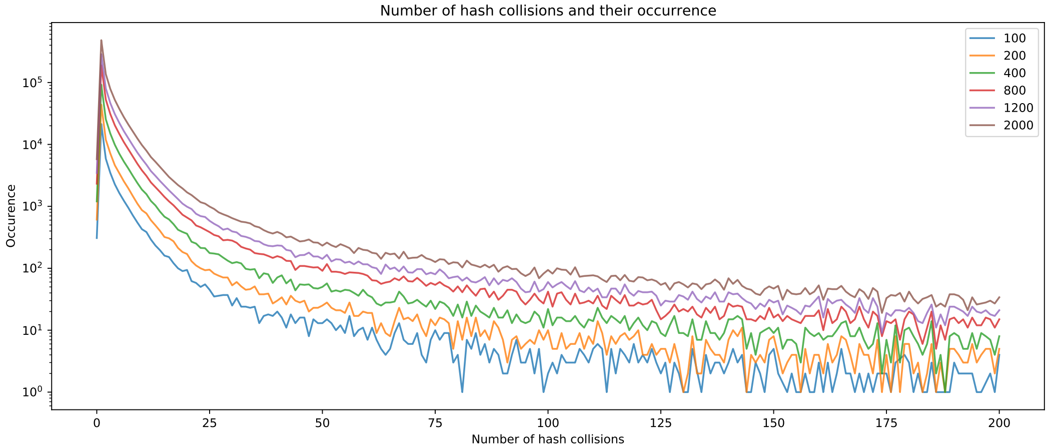


Figure S 10: Size of hash collisions (x-axis) and their collision occurrences (y-axis) for 100kb (top), 10kb (middle) and 1kb (bottom) for 144 cells from Gassler *et al.* (2017). The number of hash collisions is shown for 100, 200, 400, 800, 1200 and 2000 hash functions.



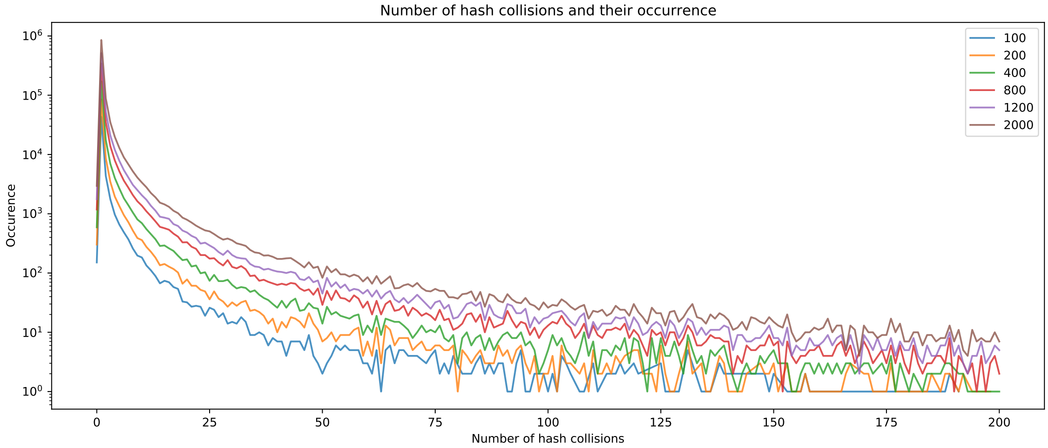


Figure S 11: Size of hash collisions (x-axis) and their collision occurrences (y-axis)for 1Mb (top) and 10kb (bottom) for 2472 cells from Nagano *et al.* (2017). The number of hash collisions is shown for 100, 200, 400, 800, 1200 and 2000 hash functions.

3.3 Number of hash values per hash function

The here shown collision statistics are for different interaction matrix resolutions from Nagano *et al.* (2017) and Gassler *et al.* (2017).

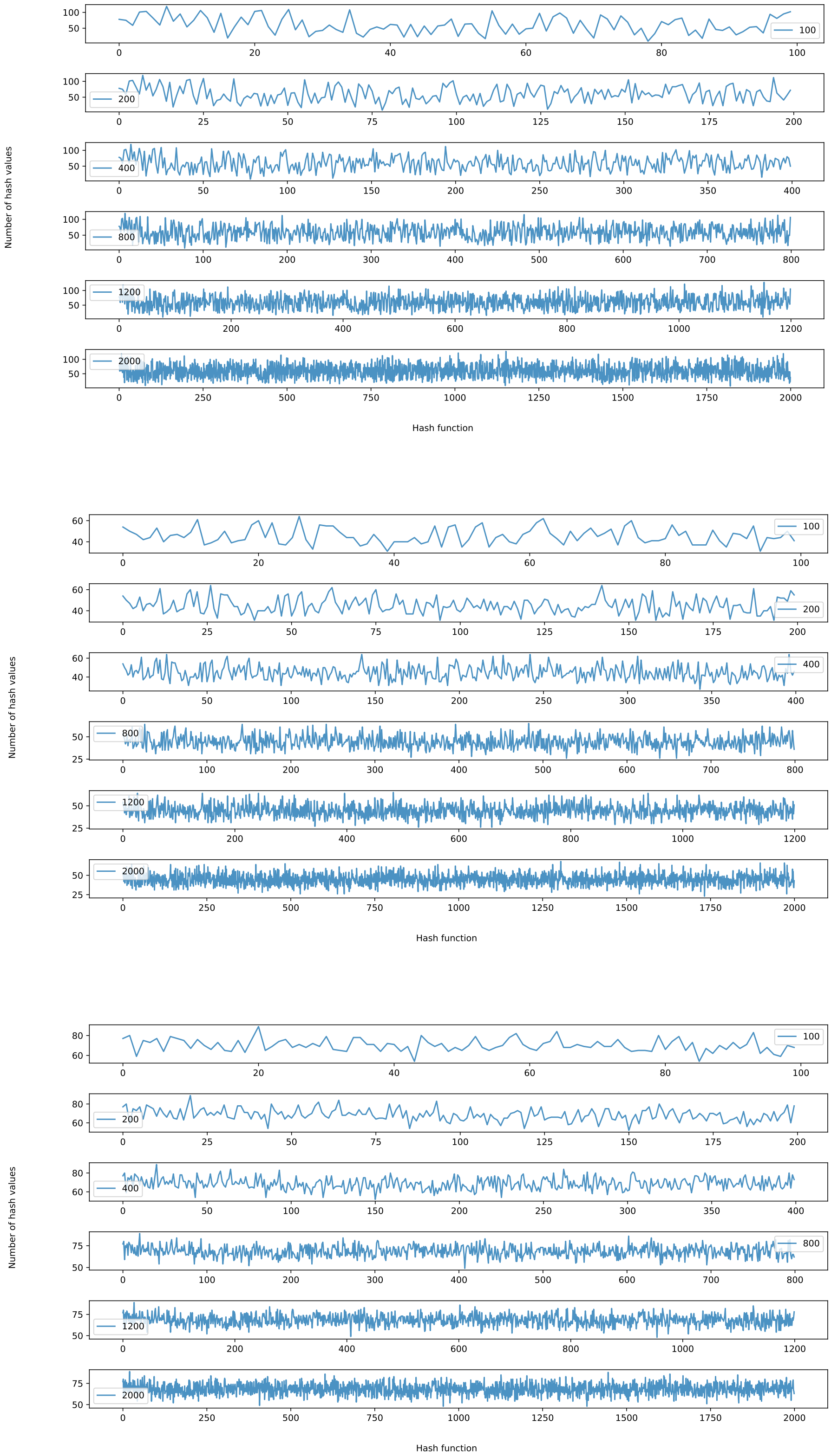


Figure S 12: Number of hash values per hash function for 100kb (top), 10kb (middle) and 1kb (bottom) for 144 cells from Gassler *et al.* (2017). The number of hash values per hash function is shown for 100, 200, 400, 800, 1200 and 2000 hash functions.

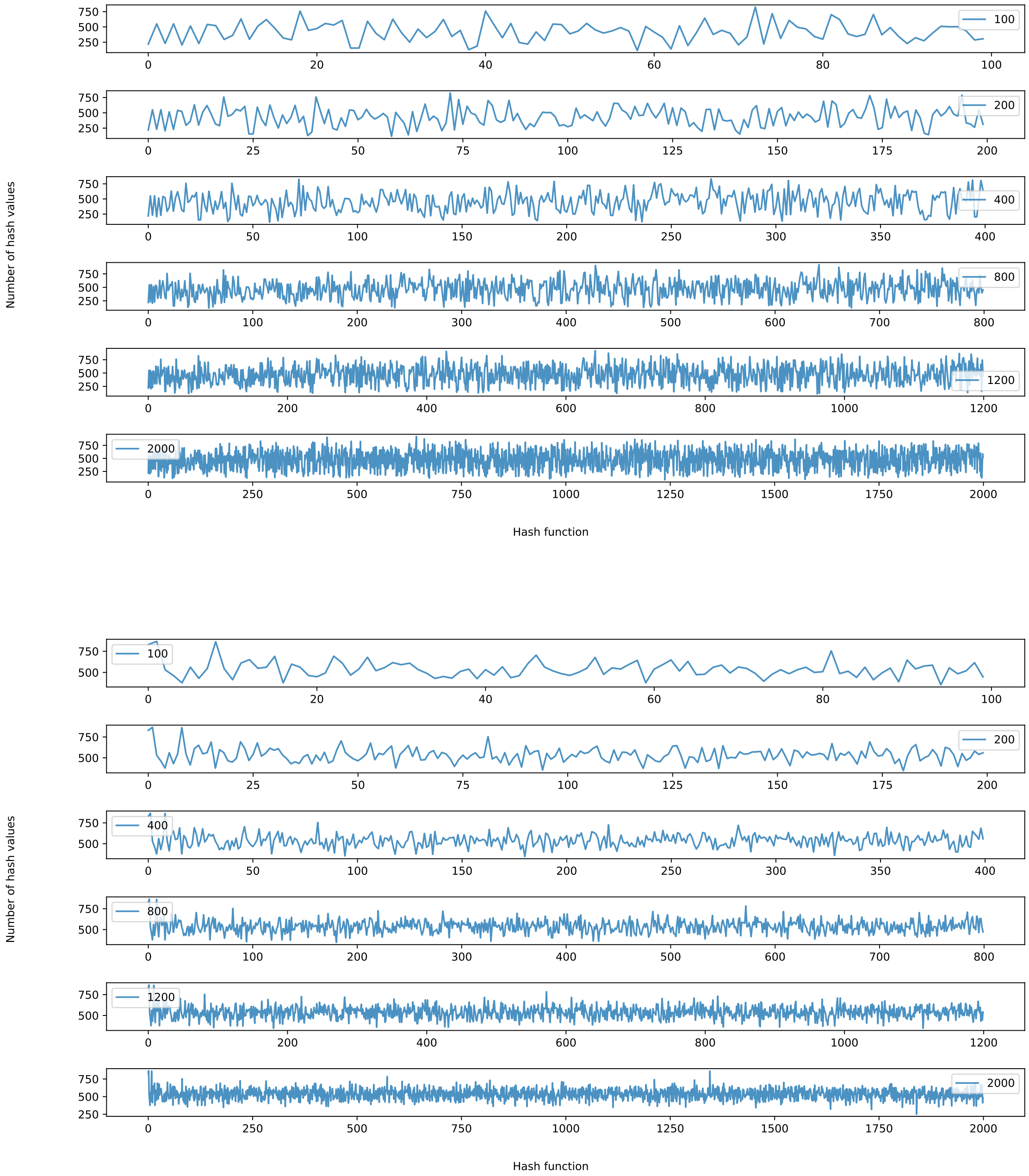


Figure S 13: Number of hash values per hash function for 1Mb (top) and 10kb (bottom) for 2472 cells from Nagano *et al.* (2017). The number of hash collisions is shown for 100, 200, 400, 800, 1200 and 2000 hash functions. One collision occurs if two cells have for one hash function the same hash value.

4 Cluster results

4.1 Cluster overlaps

Percentage values define the number of cells of a cluster which are associated with a specific cell cycle phase, i.e., in Table 1 cluster 1 has 166 cells, and 2 are associated with cell cycle stage G1; therefore, 2 / 166 or 1.2% of cluster 1 is from cell cycle stage G1. Correct identified: This measures how many percent of a cluster are unique identified with a cell phase or cell type. For example, 155 cells out of 300 G1 cells are unique, with a level of at least 80 % in their clusters. These are Cluster 2 with 84.1%, Cluster 4 with 95.8%, and 7 with each 97.9%.

4.1.1 Nagano 1MB

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (25 cells)	0 cells / 0.00 %	0 cells / 0.00 %	4 cells / 16.00 %	0 cells / 0.00 %	21 cells / 84.00 %
Cluster 1 (166 cells)	2 cells / 1.20 %	147 cells / 88.55 %	17 cells / 10.24 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (101 cells)	85 cells / 84.16 %	16 cells / 15.84 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 3 (111 cells)	0 cells / 0.00 %	10 cells / 9.01 %	100 cells / 90.09 %	0 cells / 0.00 %	1 cell / 0.90 %
Cluster 4 (24 cells)	23 cells / 95.83 %	0 cells / 0.00 %	0 cells / 0.00 %	1 cell / 4.17 %	0 cells / 0.00 %
Cluster 5 (103 cells)	0 cells / 0.00 %	12 cells / 11.65 %	91 cells / 88.35 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 6 (85 cells)	66 cells / 77.65 %	19 cells / 22.35 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 7 (48 cells)	47 cells / 97.92 %	1 cell / 2.08 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 8 (239 cells)	15 cells / 6.28 %	202 cells / 84.52 %	22 cells / 9.21 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 9 (19 cells)	2 cells / 10.53 %	0 cells / 0.00 %	0 cells / 0.00 %	16 cells / 84.21 %	1 cell / 5.26 %
Cluster 10 (203 cells)	60 cells / 29.56 %	141 cells / 69.46 %	2 cells / 0.99 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 11 (151 cells)	0 cells / 0.00 %	25 cells / 16.56 %	126 cells / 83.44 %	0 cells / 0.00 %	0 cells / 0.00 %
Correct identified > 70%	221 / 300 (73.67 %)	349 / 573 (60.91 %)	317 / 362 (87.57 %)	16 / 17 (94.12 %)	21 / 23 (91.30 %)
Correct identified > 80%	155 / 300 (51.67 %)	349 / 573 (60.91 %)	317 / 362 (87.57 %)	16 / 17 (94.12 %)	21 / 23 (91.30 %)
Correct identified > 90%	70 / 300 (23.33 %)	0 / 573 (0.00 %)	100 / 362 (27.62 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 1: Overlaps of detect clusters with known cell cycle stages from Nagano *et al.* (2017). Clustering with approximate k-nearest neighbors, 28000 hash functions, 55 principal components and UMAP k-neighbors 58, UMAP components 5 and UMAP min distance 0.2886.

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (95 cells)	0 cells / 0.00 %	1 cell / 1.05 %	94 cells / 98.95 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 1 (158 cells)	33 cells / 20.89 %	124 cells / 78.48 %	1 cell / 0.63 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (88 cells)	72 cells / 81.82 %	16 cells / 18.18 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 3 (125 cells)	5 cells / 4.00 %	31 cells / 24.80 %	82 cells / 65.60 %	0 cells / 0.00 %	7 cells / 5.60 %
Cluster 4 (104 cells)	13 cells / 12.50 %	89 cells / 85.58 %	2 cells / 1.92 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 5 (117 cells)	70 cells / 59.83 %	47 cells / 40.17 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 6 (94 cells)	0 cells / 0.00 %	39 cells / 41.49 %	55 cells / 58.51 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 7 (91 cells)	84 cells / 92.31 %	7 cells / 7.69 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 8 (53 cells)	21 cells / 39.62 %	0 cells / 0.00 %	0 cells / 0.00 %	17 cells / 32.08 %	15 cells / 28.30 %
Cluster 9 (99 cells)	1 cell / 1.01 %	94 cells / 94.95 %	4 cells / 4.04 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 10 (158 cells)	1 cell / 0.63 %	110 cells / 69.62 %	46 cells / 29.11 %	0 cells / 0.00 %	1 cell / 0.63 %
Cluster 11 (93 cells)	0 cells / 0.00 %	15 cells / 16.13 %	78 cells / 83.87 %	0 cells / 0.00 %	0 cells / 0.00 %
Correct identified > 70%	156 / 300 (52.00 %)	307 / 573 (53.58 %)	172 / 362 (47.51 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 80%	156 / 300 (52.00 %)	183 / 573 (31.94 %)	172 / 362 (47.51 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	84 / 300 (28.00 %)	94 / 573 (16.40 %)	94 / 362 (25.97 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 2: Overlaps of detect clusters with known cell cycle stages from Nagano *et al.* (2017). Clustering with approximate k-nearest neighbors, 28000 hash functions, 55 principal components and UMAP k-neighbors 58, UMAP components 2 and UMAP min distance 0.2886.

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (153 cells)	113 cells / 73.86 %	27 cells / 17.65 %	13 cells / 8.50 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 1 (151 cells)	39 cells / 25.83 %	100 cells / 66.23 %	12 cells / 7.95 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (70 cells)	0 cells / 0.00 %	67 cells / 95.71 %	3 cells / 4.29 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 3 (114 cells)	4 cells / 3.51 %	17 cells / 14.91 %	84 cells / 73.68 %	3 cells / 2.63 %	6 cells / 5.26 %
Cluster 4 (96 cells)	4 cells / 4.17 %	24 cells / 25.00 %	66 cells / 68.75 %	1 cell / 1.04 %	1 cell / 1.04 %
Cluster 5 (93 cells)	2 cells / 2.15 %	23 cells / 24.73 %	57 cells / 61.29 %	3 cells / 3.23 %	8 cells / 8.60 %
Cluster 6 (76 cells)	24 cells / 31.58 %	49 cells / 64.47 %	3 cells / 3.95 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 7 (131 cells)	85 cells / 64.89 %	43 cells / 32.82 %	3 cells / 2.29 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 8 (152 cells)	8 cells / 5.26 %	138 cells / 90.79 %	6 cells / 3.95 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 9 (72 cells)	21 cells / 29.17 %	44 cells / 61.11 %	7 cells / 9.72 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 10 (42 cells)	0 cells / 0.00 %	5 cells / 11.90 %	33 cells / 78.57 %	1 cell / 2.38 %	3 cells / 7.14 %
Cluster 11 (125 cells)	0 cells / 0.00 %	36 cells / 28.80 %	75 cells / 60.00 %	9 cells / 7.20 %	5 cells / 4.00 %
Correct identified > 70%	113 / 300 (37.67 %)	205 / 573 (35.78 %)	117 / 362 (32.32 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 80%	0 / 300 (0.00 %)	205 / 573 (35.78 %)	0 / 362 (0.00 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	0 / 300 (0.00 %)	205 / 573 (35.78 %)	0 / 362 (0.00 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 3: Approximate nearest neighbors with MinHash to preselect a candidate set. On the candidate set the nearest neighbors for a cell are computed with the Euclidean distance.

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (36 cells)	0 cells / 0.00 %	12 cells / 33.33 %	24 cells / 66.67 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 1 (199 cells)	0 cells / 0.00 %	164 cells / 82.41 %	35 cells / 17.59 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (130 cells)	0 cells / 0.00 %	117 cells / 90.00 %	13 cells / 10.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 3 (44 cells)	23 cells / 52.27 %	21 cells / 47.73 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 4 (62 cells)	0 cells / 0.00 %	3 cells / 4.84 %	41 cells / 66.13 %	0 cells / 0.00 %	18 cells / 29.03 %
Cluster 5 (258 cells)	0 cells / 0.00 %	75 cells / 29.07 %	183 cells / 70.93 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 6 (129 cells)	99 cells / 76.74 %	30 cells / 23.26 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 7 (48 cells)	46 cells / 95.83 %	1 cell / 2.08 %	0 cells / 0.00 %	1 cell / 2.08 %	0 cells / 0.00 %
Cluster 8 (193 cells)	55 cells / 28.50 %	137 cells / 70.98 %	1 cell / 0.52 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 9 (70 cells)	0 cells / 0.00 %	5 cells / 7.14 %	65 cells / 92.86 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 10 (21 cells)	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %	16 cells / 76.19 %	5 cells / 23.81 %
Cluster 11 (85 cells)	77 cells / 90.59 %	8 cells / 9.41 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Correct identified > 70%	222 / 300 (74.00 %)	418 / 573 (72.95 %)	248 / 362 (68.51 %)	16 / 17 (94.12 %)	0 / 23 (0.00 %)
Correct identified > 80%	123 / 300 (41.00 %)	281 / 573 (49.04 %)	65 / 362 (17.96 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	123 / 300 (41.00 %)	117 / 573 (20.42 %)	65 / 362 (17.96 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 4: After the approximate nearest neighbors graph a principal component analysis but no UMAP embedding is computed before the data is clustered.

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (30 cells)	14 cells / 46.67 %	16 cells / 53.33 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 1 (177 cells)	0 cells / 0.00 %	75 cells / 42.37 %	102 cells / 57.63 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (105 cells)	0 cells / 0.00 %	19 cells / 18.10 %	86 cells / 81.90 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 3 (191 cells)	47 cells / 24.61 %	143 cells / 74.87 %	1 cell / 0.52 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 4 (162 cells)	0 cells / 0.00 %	24 cells / 14.81 %	120 cells / 74.07 %	0 cells / 0.00 %	18 cells / 11.11 %
Cluster 5 (94 cells)	85 cells / 90.43 %	9 cells / 9.57 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 6 (68 cells)	51 cells / 75.00 %	17 cells / 25.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 7 (81 cells)	0 cells / 0.00 %	61 cells / 75.31 %	20 cells / 24.69 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 8 (112 cells)	92 cells / 82.14 %	20 cells / 17.86 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 9 (92 cells)	1 cell / 1.09 %	85 cells / 92.39 %	6 cells / 6.52 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 10 (24 cells)	10 cells / 41.67 %	0 cells / 0.00 %	0 cells / 0.00 %	14 cells / 58.33 %	0 cells / 0.00 %
Cluster 11 (139 cells)	0 cells / 0.00 %	104 cells / 74.82 %	27 cells / 19.42 %	3 cells / 2.16 %	5 cells / 3.60 %
Correct identified > 70%	228 / 300 (76.00 %)	393 / 573 (68.59 %)	206 / 362 (56.91 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 80%	177 / 300 (59.00 %)	85 / 573 (14.83 %)	86 / 362 (23.76 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	85 / 300 (28.33 %)	85 / 573 (14.83 %)	0 / 362 (0.00 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 5: After the approximate nearest neighbors graph no principal component analysis but an UMAP embedding is computed before the data is clustered.

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (104 cells)	86 cells / 82.69 %	18 cells / 17.31 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 1 (187 cells)	1 cell / 0.53 %	155 cells / 82.89 %	31 cells / 16.58 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (67 cells)	0 cells / 0.00 %	4 cells / 5.97 %	46 cells / 68.66 %	0 cells / 0.00 %	17 cells / 25.37 %
Cluster 3 (143 cells)	1 cell / 0.70 %	124 cells / 86.71 %	18 cells / 12.59 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 4 (248 cells)	0 cells / 0.00 %	75 cells / 30.24 %	173 cells / 69.76 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 5 (85 cells)	60 cells / 70.59 %	25 cells / 29.41 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 6 (48 cells)	46 cells / 95.83 %	1 cell / 2.08 %	0 cells / 0.00 %	1 cell / 2.08 %	0 cells / 0.00 %
Cluster 7 (41 cells)	23 cells / 56.10 %	18 cells / 43.90 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 8 (42 cells)	37 cells / 88.10 %	5 cells / 11.90 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 9 (24 cells)	2 cells / 8.33 %	0 cells / 0.00 %	0 cells / 0.00 %	16 cells / 66.67 %	6 cells / 25.00 %
Cluster 10 (120 cells)	0 cells / 0.00 %	27 cells / 22.50 %	93 cells / 77.50 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 11 (166 cells)	44 cells / 26.51 %	121 cells / 72.89 %	1 cell / 0.60 %	0 cells / 0.00 %	0 cells / 0.00 %
Correct identified > 70%	229 / 300 (76.33 %)	400 / 573 (69.81 %)	93 / 362 (25.69 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 80%	169 / 300 (56.33 %)	279 / 573 (48.69 %)	0 / 362 (0.00 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	46 / 300 (15.33 %)	0 / 573 (0.00 %)	0 / 362 (0.00 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 6: After the approximate nearest neighbors graph no principal component analysis and no UMAP embedding is computed before the data is clustered.

4.2 Intra- and inter-chromosomal contacts

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (177 cells)	22 cells / 12.43 %	100 cells / 56.50 %	55 cells / 31.07 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 1 (205 cells)	14 cells / 6.83 %	127 cells / 61.95 %	64 cells / 31.22 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (169 cells)	109 cells / 64.50 %	60 cells / 35.50 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 3 (83 cells)	0 cells / 0.00 %	17 cells / 20.48 %	64 cells / 77.11 %	0 cells / 0.00 %	2 cells / 2.41 %
Cluster 4 (141 cells)	0 cells / 0.00 %	9 cells / 6.38 %	132 cells / 93.62 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 5 (16 cells)	1 cell / 6.25 %	0 cells / 0.00 %	0 cells / 0.00 %	15 cells / 93.75 %	0 cells / 0.00 %
Cluster 6 (220 cells)	4 cells / 1.82 %	174 cells / 79.09 %	42 cells / 19.09 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 7 (30 cells)	29 cells / 96.67 %	0 cells / 0.00 %	0 cells / 0.00 %	1 cell / 3.33 %	0 cells / 0.00 %
Cluster 8 (55 cells)	51 cells / 92.73 %	1 cell / 1.82 %	1 cell / 1.82 %	0 cells / 0.00 %	2 cells / 3.64 %
Cluster 9 (102 cells)	42 cells / 41.18 %	60 cells / 58.82 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 10 (22 cells)	0 cells / 0.00 %	0 cells / 0.00 %	2 cells / 9.09 %	1 cell / 4.55 %	19 cells / 86.36 %
Cluster 11 (55 cells)	28 cells / 50.91 %	25 cells / 45.45 %	2 cells / 3.64 %	0 cells / 0.00 %	0 cells / 0.00 %
Correct identified > 70%	80 / 300 (26.67 %)	174 / 573 (30.37 %)	196 / 362 (54.14 %)	15 / 17 (88.24 %)	19 / 23 (82.61 %)
Correct identified > 80%	80 / 300 (26.67 %)	0 / 573 (0.00 %)	132 / 362 (36.46 %)	15 / 17 (88.24 %)	19 / 23 (82.61 %)
Correct identified > 90%	80 / 300 (26.67 %)	0 / 573 (0.00 %)	132 / 362 (36.46 %)	15 / 17 (88.24 %)	0 / 23 (0.00 %)

Table 7: Computing the approximate nearest neighbors graph with all Hi-C contacts: intra- and inter-chromosomal contacts.

4.3 Number of hash functions

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (144 cells)	4 cells / 2.78 %	70 cells / 48.61 %	69 cells / 47.92 %	1 cell / 0.69 %	0 cells / 0.00 %
Cluster 1 (89 cells)	48 cells / 53.93 %	39 cells / 43.82 %	1 cell / 1.12 %	1 cell / 1.12 %	0 cells / 0.00 %
Cluster 2 (164 cells)	7 cells / 4.27 %	119 cells / 72.56 %	37 cells / 22.56 %	0 cells / 0.00 %	1 cell / 0.61 %
Cluster 3 (183 cells)	44 cells / 24.04 %	99 cells / 54.10 %	39 cells / 21.31 %	0 cells / 0.00 %	1 cell / 0.55 %
Cluster 4 (10 cells)	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %	7 cells / 70.00 %	3 cells / 30.00 %
Cluster 5 (67 cells)	56 cells / 83.58 %	5 cells / 7.46 %	3 cells / 4.48 %	0 cells / 0.00 %	3 cells / 4.48 %
Cluster 6 (96 cells)	71 cells / 73.96 %	22 cells / 22.92 %	2 cells / 2.08 %	0 cells / 0.00 %	1 cell / 1.04 %
Cluster 7 (10 cells)	2 cells / 20.00 %	0 cells / 0.00 %	0 cells / 0.00 %	8 cells / 80.00 %	0 cells / 0.00 %
Cluster 8 (151 cells)	43 cells / 28.48 %	100 cells / 66.23 %	8 cells / 5.30 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 9 (13 cells)	1 cell / 7.69 %	0 cells / 0.00 %	2 cells / 15.38 %	0 cells / 0.00 %	10 cells / 76.92 %
Cluster 10 (182 cells)	15 cells / 8.24 %	77 cells / 42.31 %	88 cells / 48.35 %	0 cells / 0.00 %	2 cells / 1.10 %
Cluster 11 (166 cells)	9 cells / 5.42 %	42 cells / 25.30 %	113 cells / 68.07 %	0 cells / 0.00 %	2 cells / 1.20 %
Correct identified > 70%	127 / 300 (42.33 %)	119 / 573 (20.77 %)	0 / 362 (0.00 %)	15 / 17 (88.24 %)	10 / 23 (43.48 %)
Correct identified > 80%	56 / 300 (18.67 %)	0 / 573 (0.00 %)	0 / 362 (0.00 %)	8 / 17 (47.06 %)	0 / 23 (0.00 %)
Correct identified > 90%	0 / 300 (0.00 %)	0 / 573 (0.00 %)	0 / 362 (0.00 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 8: 2000 hash functions.

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (125 cells)	3 cells / 2.40 %	101 cells / 80.80 %	21 cells / 16.80 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 1 (153 cells)	58 cells / 37.91 %	93 cells / 60.78 %	2 cells / 1.31 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (163 cells)	1 cell / 0.61 %	14 cells / 8.59 %	147 cells / 90.18 %	0 cells / 0.00 %	1 cell / 0.61 %
Cluster 3 (189 cells)	3 cells / 1.59 %	98 cells / 51.85 %	88 cells / 46.56 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 4 (66 cells)	65 cells / 98.48 %	0 cells / 0.00 %	1 cell / 1.52 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 5 (76 cells)	0 cells / 0.00 %	6 cells / 7.89 %	69 cells / 90.79 %	0 cells / 0.00 %	1 cell / 1.32 %
Cluster 6 (22 cells)	3 cells / 13.64 %	0 cells / 0.00 %	0 cells / 0.00 %	16 cells / 72.73 %	3 cells / 13.64 %
Cluster 7 (17 cells)	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %	17 cells / 100.00 %
Cluster 8 (106 cells)	61 cells / 57.55 %	44 cells / 41.51 %	1 cell / 0.94 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 9 (23 cells)	22 cells / 95.65 %	0 cells / 0.00 %	0 cells / 0.00 %	1 cell / 4.35 %	0 cells / 0.00 %
Cluster 10 (147 cells)	78 cells / 53.06 %	68 cells / 46.26 %	1 cell / 0.68 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 11 (188 cells)	6 cells / 3.19 %	149 cells / 79.26 %	32 cells / 17.02 %	0 cells / 0.00 %	1 cell / 0.53 %
Correct identified > 70%	87 / 300 (29.00 %)	250 / 573 (43.63 %)	216 / 362 (59.67 %)	16 / 17 (94.12 %)	17 / 23 (73.91 %)
Correct identified > 80%	87 / 300 (29.00 %)	101 / 573 (17.63 %)	216 / 362 (59.67 %)	0 / 17 (0.00 %)	17 / 23 (73.91 %)
Correct identified > 90%	87 / 300 (29.00 %)	0 / 573 (0.00 %)	216 / 362 (59.67 %)	0 / 17 (0.00 %)	17 / 23 (73.91 %)

Table 9: 10,000 hash functions.

4.4 K-neighbors

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (103 cells)	10 cells / 9.71 %	82 cells / 79.61 %	11 cells / 10.68 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 1 (177 cells)	16 cells / 9.04 %	115 cells / 64.97 %	46 cells / 25.99 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (183 cells)	17 cells / 9.29 %	97 cells / 53.01 %	69 cells / 37.70 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 3 (193 cells)	40 cells / 20.73 %	126 cells / 65.28 %	27 cells / 13.99 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 4 (83 cells)	0 cells / 0.00 %	5 cells / 6.02 %	72 cells / 86.75 %	0 cells / 0.00 %	6 cells / 7.23 %
Cluster 5 (113 cells)	98 cells / 86.73 %	10 cells / 8.85 %	3 cells / 2.65 %	0 cells / 0.00 %	2 cells / 1.77 %
Cluster 6 (116 cells)	0 cells / 0.00 %	18 cells / 15.52 %	96 cells / 82.76 %	0 cells / 0.00 %	2 cells / 1.72 %
Cluster 7 (46 cells)	10 cells / 21.74 %	3 cells / 6.52 %	3 cells / 6.52 %	17 cells / 36.96 %	13 cells / 28.26 %
Cluster 8 (83 cells)	50 cells / 60.24 %	33 cells / 39.76 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 9 (18 cells)	0 cells / 0.00 %	8 cells / 44.44 %	10 cells / 55.56 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 10 (109 cells)	12 cells / 11.01 %	72 cells / 66.06 %	25 cells / 22.94 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 11 (51 cells)	47 cells / 92.16 %	4 cells / 7.84 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Correct identified > 70%	145 / 300 (48.33 %)	82 / 573 (14.31 %)	168 / 362 (46.41 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 80%	145 / 300 (48.33 %)	0 / 573 (0.00 %)	168 / 362 (46.41 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	47 / 300 (15.67 %)	0 / 573 (0.00 %)	0 / 362 (0.00 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 10: Computations with a 100-nearest neighbors graph.

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (119 cells)	0 cells / 0.00 %	80 cells / 67.23 %	38 cells / 31.93 %	0 cells / 0.00 %	1 cell / 0.84 %
Cluster 1 (227 cells)	111 cells / 48.90 %	116 cells / 51.10 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (210 cells)	0 cells / 0.00 %	146 cells / 69.52 %	61 cells / 29.05 %	0 cells / 0.00 %	3 cells / 1.43 %
Cluster 3 (148 cells)	137 cells / 92.57 %	11 cells / 7.43 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 4 (75 cells)	0 cells / 0.00 %	13 cells / 17.33 %	62 cells / 82.67 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 5 (100 cells)	0 cells / 0.00 %	27 cells / 27.00 %	73 cells / 73.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 6 (53 cells)	0 cells / 0.00 %	8 cells / 15.09 %	34 cells / 64.15 %	0 cells / 0.00 %	11 cells / 20.75 %
Cluster 7 (81 cells)	0 cells / 0.00 %	12 cells / 14.81 %	67 cells / 82.72 %	0 cells / 0.00 %	2 cells / 2.47 %
Cluster 8 (48 cells)	0 cells / 0.00 %	23 cells / 47.92 %	21 cells / 43.75 %	0 cells / 0.00 %	4 cells / 8.33 %
Cluster 9 (139 cells)	35 cells / 25.18 %	98 cells / 70.50 %	6 cells / 4.32 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 10 (50 cells)	11 cells / 22.00 %	39 cells / 78.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 11 (25 cells)	6 cells / 24.00 %	0 cells / 0.00 %	0 cells / 0.00 %	17 cells / 68.00 %	2 cells / 8.00 %
Correct identified > 70%	137 / 300 (45.67 %)	137 / 573 (23.91 %)	202 / 362 (55.80 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 80%	137 / 300 (45.67 %)	0 / 573 (0.00 %)	129 / 362 (35.64 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	137 / 300 (45.67 %)	0 / 573 (0.00 %)	0 / 362 (0.00 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 11: Computations with a 500-nearest neighbors graph.

4.5 Cluster algorithms

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (126 cells)	1 cell / 0.79 %	112 cells / 88.89 %	13 cells / 10.32 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 1 (114 cells)	97 cells / 85.09 %	17 cells / 14.91 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (131 cells)	0 cells / 0.00 %	9 cells / 6.87 %	122 cells / 93.13 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 3 (104 cells)	0 cells / 0.00 %	70 cells / 67.31 %	34 cells / 32.69 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 4 (107 cells)	23 cells / 21.50 %	83 cells / 77.57 %	1 cell / 0.93 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 5 (59 cells)	59 cells / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 6 (51 cells)	13 cells / 25.49 %	0 cells / 0.00 %	0 cells / 0.00 %	17 cells / 33.33 %	21 cells / 41.18 %
Cluster 7 (118 cells)	44 cells / 37.29 %	73 cells / 61.86 %	1 cell / 0.85 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 8 (142 cells)	1 cell / 0.70 %	123 cells / 86.62 %	18 cells / 12.68 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 9 (101 cells)	0 cells / 0.00 %	16 cells / 15.84 %	83 cells / 82.18 %	0 cells / 0.00 %	2 cells / 1.98 %
Cluster 10 (112 cells)	62 cells / 55.36 %	50 cells / 44.64 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 11 (110 cells)	0 cells / 0.00 %	20 cells / 18.18 %	90 cells / 81.82 %	0 cells / 0.00 %	0 cells / 0.00 %
Correct identified > 70%	156 / 300 (52.00 %)	318 / 573 (55.50 %)	295 / 362 (81.49 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 80%	156 / 300 (52.00 %)	235 / 573 (41.01 %)	295 / 362 (81.49 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	59 / 300 (19.67 %)	0 / 573 (0.00 %)	122 / 362 (33.70 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 12: k-means

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (113 cells)	110 cells / 97.35 %	3 cells / 2.65 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 1 (208 cells)	67 cells / 32.21 %	139 cells / 66.83 %	2 cells / 0.96 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (198 cells)	0 cells / 0.00 %	21 cells / 10.61 %	176 cells / 88.89 %	0 cells / 0.00 %	1 cell / 0.51 %
Cluster 3 (136 cells)	1 cell / 0.74 %	93 cells / 68.38 %	42 cells / 30.88 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 4 (47 cells)	22 cells / 46.81 %	0 cells / 0.00 %	0 cells / 0.00 %	17 cells / 36.17 %	8 cells / 17.02 %
Cluster 5 (126 cells)	0 cells / 0.00 %	110 cells / 87.30 %	16 cells / 12.70 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 6 (118 cells)	0 cells / 0.00 %	17 cells / 14.41 %	101 cells / 85.59 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 7 (72 cells)	61 cells / 84.72 %	11 cells / 15.28 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 8 (66 cells)	5 cells / 7.58 %	61 cells / 92.42 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 9 (71 cells)	28 cells / 39.44 %	43 cells / 60.56 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 10 (91 cells)	6 cells / 6.59 %	69 cells / 75.82 %	16 cells / 17.58 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 11 (29 cells)	0 cells / 0.00 %	6 cells / 20.69 %	9 cells / 31.03 %	0 cells / 0.00 %	14 cells / 48.28 %
Correct identified > 70%	171 / 300 (57.00 %)	240 / 573 (41.88 %)	277 / 362 (76.52 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 80%	171 / 300 (57.00 %)	171 / 573 (29.84 %)	277 / 362 (76.52 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	110 / 300 (36.67 %)	61 / 573 (10.65 %)	0 / 362 (0.00 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 13: agglomerative ward

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (102 cells)	0 cells / 0.00 %	72 cells / 70.59 %	30 cells / 29.41 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 1 (90 cells)	0 cells / 0.00 %	12 cells / 13.33 %	78 cells / 86.67 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (221 cells)	0 cells / 0.00 %	25 cells / 11.31 %	193 cells / 87.33 %	0 cells / 0.00 %	3 cells / 1.36 %
Cluster 3 (59 cells)	22 cells / 37.29 %	0 cells / 0.00 %	0 cells / 0.00 %	17 cells / 28.81 %	20 cells / 33.90 %
Cluster 4 (160 cells)	0 cells / 0.00 %	137 cells / 85.62 %	23 cells / 14.37 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 5 (158 cells)	49 cells / 31.01 %	108 cells / 68.35 %	1 cell / 0.63 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 6 (159 cells)	121 cells / 76.10 %	38 cells / 23.90 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 7 (120 cells)	69 cells / 57.50 %	51 cells / 42.50 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 8 (36 cells)	36 cells / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 9 (37 cells)	0 cells / 0.00 %	23 cells / 62.16 %	14 cells / 37.84 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 10 (85 cells)	1 cell / 1.18 %	61 cells / 71.76 %	23 cells / 27.06 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 11 (48 cells)	2 cells / 4.17 %	46 cells / 95.83 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Correct identified > 70%	157 / 300 (52.33 %)	316 / 573 (55.15 %)	271 / 362 (74.86 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 80%	36 / 300 (12.00 %)	183 / 573 (31.94 %)	271 / 362 (74.86 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	36 / 300 (12.00 %)	46 / 573 (8.03 %)	0 / 362 (0.00 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 14: agglomerative complete

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (95 cells)	0 cells / 0.00 %	62 cells / 65.26 %	33 cells / 34.74 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 1 (163 cells)	0 cells / 0.00 %	143 cells / 87.73 %	20 cells / 12.27 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (98 cells)	3 cells / 3.06 %	36 cells / 36.73 %	58 cells / 59.18 %	0 cells / 0.00 %	1 cell / 1.02 %
Cluster 3 (63 cells)	63 cells / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 4 (163 cells)	51 cells / 31.29 %	111 cells / 68.10 %	1 cell / 0.61 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 5 (52 cells)	15 cells / 28.85 %	37 cells / 71.15 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 6 (120 cells)	101 cells / 84.17 %	19 cells / 15.83 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 7 (87 cells)	0 cells / 0.00 %	9 cells / 10.34 %	78 cells / 89.66 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 8 (114 cells)	63 cells / 55.26 %	51 cells / 44.74 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 9 (98 cells)	1 cell / 1.02 %	91 cells / 92.86 %	6 cells / 6.12 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 10 (180 cells)	0 cells / 0.00 %	14 cells / 7.78 %	166 cells / 92.22 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 11 (42 cells)	3 cells / 7.14 %	0 cells / 0.00 %	0 cells / 0.00 %	17 cells / 40.48 %	22 cells / 52.38 %
Correct identified > 70%	164 / 300 (54.67 %)	271 / 573 (47.29 %)	244 / 362 (67.40 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 80%	164 / 300 (54.67 %)	234 / 573 (40.84 %)	244 / 362 (67.40 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	63 / 300 (21.00 %)	91 / 573 (15.88 %)	166 / 362 (45.86 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 15: agglomerative average

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (1263 cells)	297 cells / 23.52 %	567 cells / 44.89 %	359 cells / 28.42 %	17 cells / 1.35 %	23 cells / 1.82 %
Cluster 1 (2 cells)	0 cells / 0.00 %	1 cell / 50.00 %	1 cell / 50.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (1 cells)	0 cells / 0.00 %	0 cells / 0.00 %	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 3 (1 cells)	0 cells / 0.00 %	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 4 (1 cells)	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 5 (1 cells)	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 6 (1 cells)	0 cells / 0.00 %	0 cells / 0.00 %	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 7 (1 cells)	0 cells / 0.00 %	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 8 (1 cells)	0 cells / 0.00 %	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 9 (1 cells)	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 10 (1 cells)	0 cells / 0.00 %	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 11 (1 cells)	0 cells / 0.00 %	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Correct identified > 70%	3 / 300 (1.00 %)	5 / 573 (0.87 %)	2 / 362 (0.55 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 80%	3 / 300 (1.00 %)	5 / 573 (0.87 %)	2 / 362 (0.55 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	3 / 300 (1.00 %)	5 / 573 (0.87 %)	2 / 362 (0.55 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 16: agglomerative single

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (195 cells)	3 cells / 1.54 %	35 cells / 17.95 %	119 cells / 61.03 %	17 cells / 8.72 %	21 cells / 10.77 %
Cluster 1 (93 cells)	0 cells / 0.00 %	70 cells / 75.27 %	23 cells / 24.73 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (206 cells)	0 cells / 0.00 %	19 cells / 9.22 %	185 cells / 89.81 %	0 cells / 0.00 %	2 cells / 0.97 %
Cluster 3 (76 cells)	5 cells / 6.58 %	66 cells / 86.84 %	5 cells / 6.58 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 4 (87 cells)	18 cells / 20.69 %	68 cells / 78.16 %	1 cell / 1.15 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 5 (52 cells)	15 cells / 28.85 %	37 cells / 71.15 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 6 (114 cells)	1 cell / 0.88 %	100 cells / 87.72 %	13 cells / 11.40 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 7 (100 cells)	74 cells / 74.00 %	26 cells / 26.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 8 (121 cells)	2 cells / 1.65 %	103 cells / 85.12 %	16 cells / 13.22 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 9 (77 cells)	74 cells / 96.10 %	3 cells / 3.90 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 10 (99 cells)	53 cells / 53.54 %	46 cells / 46.46 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 11 (55 cells)	55 cells / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Correct identified > 70%	203 / 300 (67.67 %)	444 / 573 (77.49 %)	185 / 362 (51.10 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 80%	129 / 300 (43.00 %)	269 / 573 (46.95 %)	185 / 362 (51.10 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	129 / 300 (43.00 %)	0 / 573 (0.00 %)	0 / 362 (0.00 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 17: birch

4.6 Competing approaches

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (28 cells)	1 cell / 3.57 %	0 cells / 0.00 %	0 cells / 0.00 %	17 cells / 60.71 %	10 cells / 35.71 %
Cluster 1 (123 cells)	85 cells / 69.11 %	38 cells / 30.89 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (119 cells)	30 cells / 25.21 %	86 cells / 72.27 %	3 cells / 2.52 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 3 (66 cells)	0 cells / 0.00 %	32 cells / 48.48 %	34 cells / 51.52 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 4 (100 cells)	0 cells / 0.00 %	4 cells / 4.00 %	83 cells / 83.00 %	0 cells / 0.00 %	13 cells / 13.00 %
Cluster 5 (121 cells)	0 cells / 0.00 %	6 cells / 4.96 %	115 cells / 95.04 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 6 (58 cells)	58 cells / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 7 (172 cells)	16 cells / 9.30 %	151 cells / 87.79 %	5 cells / 2.91 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 8 (105 cells)	101 cells / 96.19 %	4 cells / 3.81 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 9 (165 cells)	9 cells / 5.45 %	137 cells / 83.03 %	19 cells / 11.52 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 10 (167 cells)	0 cells / 0.00 %	80 cells / 47.90 %	87 cells / 52.10 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 11 (51 cells)	0 cells / 0.00 %	35 cells / 68.63 %	16 cells / 31.37 %	0 cells / 0.00 %	0 cells / 0.00 %
Correct identified > 70%	159 / 300 (53.00 %)	374 / 573 (65.27 %)	198 / 362 (54.70 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 80%	159 / 300 (53.00 %)	288 / 573 (50.26 %)	198 / 362 (54.70 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	159 / 300 (53.00 %)	0 / 573 (0.00 %)	115 / 362 (31.77 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 18: Overlaps of detect clusters with known cell cycle stages from Nagano *et al.* (2017). Clustering with Zhou's scHiCluster.

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (769 cells)	177 cells / 23.02 %	352 cells / 45.77 %	216 cells / 28.09 %	10 cells / 1.30 %	14 cells / 1.82 %
Cluster 1 (1 cells)	0 cells / 0.00 %	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (1 cells)	0 cells / 0.00 %	0 cells / 0.00 %	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 3 (1 cells)	0 cells / 0.00 %	0 cells / 0.00 %	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 4 (1 cells)	0 cells / 0.00 %	0 cells / 0.00 %	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 5 (1 cells)	0 cells / 0.00 %	0 cells / 0.00 %	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 6 (477 cells)	112 cells / 23.48 %	210 cells / 44.03 %	139 cells / 29.14 %	7 cells / 1.47 %	9 cells / 1.89 %
Cluster 7 (13 cells)	4 cells / 30.77 %	7 cells / 53.85 %	2 cells / 15.38 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 8 (1 cells)	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 9 (4 cells)	2 cells / 50.00 %	1 cell / 25.00 %	1 cell / 25.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 10 (4 cells)	4 cells / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 11 (2 cells)	0 cells / 0.00 %	2 cells / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Correct identified > 70%	5 / 300 (1.67 %)	3 / 573 (0.52 %)	4 / 362 (1.10 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 80%	5 / 300 (1.67 %)	3 / 573 (0.52 %)	4 / 362 (1.10 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	5 / 300 (1.67 %)	3 / 573 (0.52 %)	4 / 362 (1.10 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 19: Scikit-learn k-nearest neighbor with k=1275, with spectral clustering.

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (25 cells)	5 cells / 20.00 %	20 cells / 80.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 1 (21 cells)	1 cell / 4.76 %	18 cells / 85.71 %	2 cells / 9.52 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (1 cells)	0 cells / 0.00 %	0 cells / 0.00 %	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 3 (1 cells)	0 cells / 0.00 %	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 4 (9 cells)	3 cells / 33.33 %	6 cells / 66.67 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 5 (1 cells)	0 cells / 0.00 %	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 6 (1 cells)	0 cells / 0.00 %	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 7 (1202 cells)	287 cells / 23.88 %	518 cells / 43.09 %	357 cells / 29.70 %	17 cells / 1.41 %	23 cells / 1.91 %
Cluster 8 (1 cells)	0 cells / 0.00 %	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 9 (8 cells)	3 cells / 37.50 %	4 cells / 50.00 %	1 cell / 12.50 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 10 (4 cells)	1 cell / 25.00 %	3 cells / 75.00 %	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 11 (1 cells)	0 cells / 0.00 %	0 cells / 0.00 %	1 cell / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Correct identified > 70%	0 / 300 (0.00 %)	45 / 573 (7.85 %)	2 / 362 (0.55 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 80%	0 / 300 (0.00 %)	42 / 573 (7.33 %)	2 / 362 (0.55 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	0 / 300 (0.00 %)	4 / 573 (0.70 %)	2 / 362 (0.55 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 20: PCA on raw data with spectral clustering.

Cluster	G1 (300 cells)	early-S (573 cells)	late-S/G2 (362 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (216 cells)	79 cells / 36.57 %	101 cells / 46.76 %	31 cells / 14.35 %	1 cell / 0.46 %	4 cells / 1.85 %
Cluster 1 (58 cells)	0 cells / 0.00 %	33 cells / 56.90 %	25 cells / 43.10 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (108 cells)	24 cells / 22.22 %	58 cells / 53.70 %	26 cells / 24.07 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 3 (106 cells)	0 cells / 0.00 %	82 cells / 77.36 %	24 cells / 22.64 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 4 (226 cells)	82 cells / 36.28 %	110 cells / 48.67 %	30 cells / 13.27 %	2 cells / 0.88 %	2 cells / 0.88 %
Cluster 5 (99 cells)	2 cells / 2.02 %	13 cells / 13.13 %	77 cells / 77.78 %	0 cells / 0.00 %	7 cells / 7.07 %
Cluster 6 (197 cells)	112 cells / 56.85 %	64 cells / 32.49 %	16 cells / 8.12 %	0 cells / 0.00 %	5 cells / 2.54 %
Cluster 7 (80 cells)	0 cells / 0.00 %	6 cells / 7.50 %	70 cells / 87.50 %	0 cells / 0.00 %	4 cells / 5.00 %
Cluster 8 (92 cells)	1 cell / 1.09 %	84 cells / 91.30 %	7 cells / 7.61 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 9 (15 cells)	0 cells / 0.00 %	0 cells / 0.00 %	0 cells / 0.00 %	14 cells / 93.33 %	1 cell / 6.67 %
Cluster 10 (3 cells)	0 cells / 0.00 %	0 cells / 0.00 %	3 cells / 100.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 11 (75 cells)	0 cells / 0.00 %	22 cells / 29.33 %	53 cells / 70.67 %	0 cells / 0.00 %	0 cells / 0.00 %
Correct identified > 70%	0 / 300 (0.00 %)	166 / 573 (28.97 %)	203 / 362 (56.08 %)	14 / 17 (82.35 %)	0 / 23 (0.00 %)
Correct identified > 80%	0 / 300 (0.00 %)	84 / 573 (14.66 %)	73 / 362 (20.17 %)	14 / 17 (82.35 %)	0 / 23 (0.00 %)
Correct identified > 90%	0 / 300 (0.00 %)	84 / 573 (14.66 %)	3 / 362 (0.83 %)	14 / 17 (82.35 %)	0 / 23 (0.00 %)

Table 21: Clustering on raw interaction matrices with k-means clustering.

5 Runtimes on 10 kb resolution

Method	Runtime	Memory
Raw and K-Means	-	> 1 TB
Raw and Spectral	-	> 1 TB
PCA and K-Means	-	> 1 TB
PCA and Spectral	-	> 1 TB
scikit-learn k-nn k = 2633 and k-means	-	> 1 TB
scikit-learn k-nn k = 2633 and Spectral	-	> 1 TB
scHicClusterMinHash k = 2633 and k-means	06:17 min	40.1 GB
scHicClusterMinHash k = 2633 and Spectral	06:26 min	40.1 GB
scHicClusterMinHash eucl. k = 2633 and k-means	08:10 min	40.1 GB
scHicClusterMinHash eucl. k = 2633 and Spectral	08:08 min	40.1 GB
Zhou's scHiCluster CPU	- (*)	> 970 GB

Table 22: Runtimes and memory usage on 10 kb resolution, 2633 cells on a single-cell Hi-C matrix. Data from Nagano *et al.* (2017) Diploid cells, with 12 clusters. For clustering k-means and spectral clustering are used, scHicClusterMinHash with 800 hash functions, k=2472, applied PCA and 100 principal components for clustering. (*) Zhou's scHiCluster computed 97 hours the data for chromosome 10 and requested 970 GB of memory, the computation was canceled after this time. All results computed on 2x Intel XEON E5-2630 v4 @ 2.20GHz 2x 10 cores / 2x 20 threads, 1 TB memory.

Method	Runtime	Memory
Raw and K-Means	-	> 128 GB
Raw and Spectral	-	> 128 GB
PCA and K-Means	-	> 128 GB
PCA and Spectral	-	> 128 GB
scikit-learn k-nn k = 2632 and K-means	-	> 128 GB
scikit-learn k-nn k = 2632 and Spectral	-	> 128 GB
MinHash k = 2633 and K-means	03:39 min	40.1 GB
MinHash k = 2633 and Spectral	03:41 min	40.1 GB
MinHash k = 2633 and K-means (–saveMemory 1%)	12:53 min	12.5 GB
MinHash k = 2633 and K-means intra-chromosomal	08:26 min	35.8 GB
MinHash k = 2633 and Spectral intra-chromosomal	08:55 min	35.8 GB
MinHash euclidean k = 2633 and K-means	06:39 min	40.1 GB
MinHash euclidean k = 2633 and Spectral	06:39 min	40.1 GB
MinHash euclidean k = 2633 and K-means intra-chromosomal	11:47 min	35.8 GB
MinHash euclidean k = 2633 and Spectral intra-chromosomal	11:49 min	35.8 GB
Zhou's scHiCluster CPU	-	> 128 GB
Zhou's scHiCluster GPU	-	> 128 GB

Table 23: Runtimes and memory usage with 10 kb resolution on a single-cell Hi-C matrix with 2633 cells. Normalized to a read coverage of 100,000 reads and interaction values smaller 1 are *kept*. Data from Nagano *et al.* (2017) Diploid cells, with 12 clusters. For clustering K-means and spectral clustering are used, MinHash with 800 hash functions. All results computed on AMD Ryzen 3700X 8 cores / 16 threads, 128 GB memory; Nvidia GTX 1070 8 GB memory.

6 Nagano 2017 10 kb data

MinHash on 10 kb data from Nagano with 1088 cells. Parameters: differing number of hash functions, 44 principal components, spectral clustering. UMAP parameters: k-neighbors 36, components 9, min distance 0.05.

Cluster	G1 (249 cells)	early-S (448 cells)	late-S/G2 (341 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (71 cells)	17 cells / 23.94 %	12 cells / 16.90 %	39 cells / 54.93 %	0 cells / 0.00 %	3 cells / 4.23 %
Cluster 1 (67 cells)	2 cells / 2.99 %	60 cells / 89.55 %	5 cells / 7.46 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 2 (75 cells)	1 cell / 1.33 %	44 cells / 58.67 %	28 cells / 37.33 %	1 cell / 1.33 %	1 cell / 1.33 %
Cluster 3 (127 cells)	8 cells / 6.30 %	88 cells / 69.29 %	24 cells / 18.90 %	5 cells / 3.94 %	2 cells / 1.57 %
Cluster 4 (55 cells)	31 cells / 56.36 %	16 cells / 29.09 %	7 cells / 12.73 %	0 cells / 0.00 %	1 cell / 1.82 %
Cluster 5 (64 cells)	1 cell / 1.56 %	19 cells / 29.69 %	44 cells / 68.75 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 6 (111 cells)	22 cells / 19.82 %	53 cells / 47.75 %	27 cells / 24.32 %	6 cells / 5.41 %	3 cells / 2.70 %
Cluster 7 (117 cells)	68 cells / 58.12 %	24 cells / 20.51 %	19 cells / 16.24 %	4 cells / 3.42 %	2 cells / 1.71 %
Cluster 8 (54 cells)	17 cells / 31.48 %	21 cells / 38.89 %	13 cells / 24.07 %	0 cells / 0.00 %	3 cells / 5.56 %
Cluster 9 (50 cells)	29 cells / 58.00 %	17 cells / 34.00 %	2 cells / 4.00 %	0 cells / 0.00 %	2 cells / 4.00 %
Cluster 10 (74 cells)	11 cells / 14.86 %	51 cells / 68.92 %	12 cells / 16.22 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 11 (53 cells)	29 cells / 54.72 %	15 cells / 28.30 %	9 cells / 16.98 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 12 (65 cells)	5 cells / 7.69 %	8 cells / 12.31 %	49 cells / 75.38 %	0 cells / 0.00 %	3 cells / 4.62 %
Cluster 13 (95 cells)	8 cells / 8.42 %	20 cells / 21.05 %	63 cells / 66.32 %	1 cell / 1.05 %	3 cells / 3.16 %
Correct identified > 70%	0 / 249 (0.00 %)	60 / 448 (13.39 %)	49 / 341 (14.37 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 80%	0 / 249 (0.00 %)	60 / 448 (13.39 %)	0 / 341 (0.00 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	0 / 249 (0.00 %)	0 / 448 (0.00 %)	0 / 341 (0.00 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 24: 20000 hash functions.

Cluster	G1 (249 cells)	early-S (448 cells)	late-S/G2 (341 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (71 cells)	0 cells / 0.00 %	24 cells / 33.80 %	47 cells / 66.20 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 1 (80 cells)	6 cells / 7.50 %	57 cells / 71.25 %	15 cells / 18.75 %	1 cell / 1.25 %	1 cell / 1.25 %
Cluster 2 (114 cells)	45 cells / 39.47 %	50 cells / 43.86 %	14 cells / 12.28 %	0 cells / 0.00 %	5 cells / 4.39 %
Cluster 3 (93 cells)	11 cells / 11.83 %	20 cells / 21.51 %	56 cells / 60.22 %	0 cells / 0.00 %	6 cells / 6.45 %
Cluster 4 (104 cells)	77 cells / 74.04 %	16 cells / 15.38 %	9 cells / 8.65 %	0 cells / 0.00 %	2 cells / 1.92 %
Cluster 5 (32 cells)	8 cells / 25.00 %	4 cells / 12.50 %	4 cells / 12.50 %	16 cells / 50.00 %	0 cells / 0.00 %
Cluster 6 (82 cells)	0 cells / 0.00 %	34 cells / 41.46 %	48 cells / 58.54 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 7 (66 cells)	31 cells / 46.97 %	23 cells / 34.85 %	12 cells / 18.18 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 8 (63 cells)	8 cells / 12.70 %	37 cells / 58.73 %	17 cells / 26.98 %	0 cells / 0.00 %	1 cell / 1.59 %
Cluster 9 (58 cells)	12 cells / 20.69 %	26 cells / 44.83 %	16 cells / 27.59 %	0 cells / 0.00 %	4 cells / 6.90 %
Cluster 10 (69 cells)	9 cells / 13.04 %	5 cells / 7.25 %	52 cells / 75.36 %	0 cells / 0.00 %	3 cells / 4.35 %
Cluster 11 (100 cells)	4 cells / 4.00 %	80 cells / 80.00 %	16 cells / 16.00 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 12 (75 cells)	1 cell / 1.33 %	52 cells / 69.33 %	22 cells / 29.33 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 13 (71 cells)	37 cells / 52.11 %	20 cells / 28.17 %	13 cells / 18.31 %	0 cells / 0.00 %	1 cell / 1.41 %
Correct identified > 70%	77 / 249 (30.92 %)	137 / 448 (30.58 %)	52 / 341 (15.25 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 80%	0 / 249 (0.00 %)	80 / 448 (17.86 %)	0 / 341 (0.00 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	0 / 249 (0.00 %)	0 / 448 (0.00 %)	0 / 341 (0.00 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 25: 40000 hash functions.

Cluster	G1 (249 cells)	early-S (448 cells)	late-S/G2 (341 cells)	post-M (17 cells)	pre-M (23 cells)
Cluster 0 (102 cells)	55 cells / 53.92 %	25 cells / 24.51 %	12 cells / 11.76 %	6 cells / 5.88 %	4 cells / 3.92 %
Cluster 1 (151 cells)	84 cells / 55.63 %	42 cells / 27.81 %	20 cells / 13.25 %	0 cells / 0.00 %	5 cells / 3.31 %
Cluster 2 (143 cells)	17 cells / 11.89 %	27 cells / 18.88 %	93 cells / 65.03 %	0 cells / 0.00 %	6 cells / 4.20 %
Cluster 3 (95 cells)	4 cells / 4.21 %	87 cells / 91.58 %	4 cells / 4.21 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 4 (79 cells)	21 cells / 26.58 %	37 cells / 46.84 %	21 cells / 26.58 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 5 (33 cells)	6 cells / 18.18 %	8 cells / 24.24 %	6 cells / 18.18 %	11 cells / 33.33 %	2 cells / 6.06 %
Cluster 6 (111 cells)	2 cells / 1.80 %	50 cells / 45.05 %	59 cells / 53.15 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 7 (32 cells)	14 cells / 43.75 %	11 cells / 34.38 %	7 cells / 21.88 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 8 (77 cells)	9 cells / 11.69 %	49 cells / 63.64 %	17 cells / 22.08 %	0 cells / 0.00 %	2 cells / 2.60 %
Cluster 9 (45 cells)	25 cells / 55.56 %	14 cells / 31.11 %	6 cells / 13.33 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 10 (23 cells)	3 cells / 13.04 %	14 cells / 60.87 %	6 cells / 26.09 %	0 cells / 0.00 %	0 cells / 0.00 %
Cluster 11 (34 cells)	2 cells / 5.88 %	11 cells / 32.35 %	20 cells / 58.82 %	0 cells / 0.00 %	1 cell / 2.94 %
Cluster 12 (76 cells)	3 cells / 3.95 %	52 cells / 68.42 %	18 cells / 23.68 %	0 cells / 0.00 %	3 cells / 3.95 %
Cluster 13 (77 cells)	4 cells / 5.19 %	21 cells / 27.27 %	52 cells / 67.53 %	0 cells / 0.00 %	0 cells / 0.00 %
Correct identified > 70%	0 / 249 (0.00 %)	87 / 448 (19.42 %)	0 / 341 (0.00 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 80%	0 / 249 (0.00 %)	87 / 448 (19.42 %)	0 / 341 (0.00 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)
Correct identified > 90%	0 / 249 (0.00 %)	87 / 448 (19.42 %)	0 / 341 (0.00 %)	0 / 17 (0.00 %)	0 / 23 (0.00 %)

Table 26: 50000 hash functions.

7 Cluster results on Ramani 1MB

Cluster	HeLa (269 cells)	HAP1 (254 cells)
Cluster 0 (264 cells)	24 cells / 9.09 %	240 cells / 90.91 %
Cluster 1 (259 cells)	245 cells / 94.59 %	14 cells / 5.41 %
Correct identified > 70%	245 / 269 (91.08 %)	240 / 254 (94.49 %)
Correct identified > 80%	245 / 269 (91.08 %)	240 / 254 (94.49 %)
Correct identified > 90%	245 / 269 (91.08 %)	240 / 254 (94.49 %)

(a) ML1 with two clusters

Cluster	HeLa (269 cells)	HAP1 (254 cells)
Cluster 0 (244 cells)	8 cells / 3.28 %	236 cells / 96.72 %
Cluster 1 (265 cells)	258 cells / 97.36 %	7 cells / 2.64 %
Cluster 2 (14 cells)	3 cells / 21.43 %	11 cells / 78.57 %
Correct identified > 70%	258 / 269 (95.91 %)	247 / 254 (97.24 %)
Correct identified > 80%	258 / 269 (95.91 %)	236 / 254 (92.91 %)
Correct identified > 90%	258 / 269 (95.91 %)	236 / 254 (92.91 %)

(b) ML1 with three clusters

Table 27: Overlaps of detected clusters with known cell types from Ramani *et al.* (2017), ML1 batch. Approximate k-nn with MinHash, spectral clustering, 2000 hash functions, full-nearest neighbors graph, 7 principal components, intra-chromosomal contacts only, umap: n_neighbors 40, min_dist 0.25, n_components 2 for two clusters, n_components 6 for three clusters.

Cluster	HeLa (267 cells)	HAP1 (251 cells)
Cluster 0 (256 cells)	71 cells / 27.7%	185 cells / 72.3%
Cluster 1 (262 cells)	196 cells / 74.8%	66 cells / 25.2%
Correct identified > 70%	196 / 269 (72.86 %)	185 / 254 (72.83 %)
Correct identified > 80%	0 / 269 (0 %)	0 / 254 (0 %)
Correct identified > 90%	0 / 269 (0 %)	0 / 254 (0 %)

(a) ML1 with two clusters

Cluster	HeLa (267 cells)	HAP1 (251 cells)
Cluster 0 (231 cells)	8 cells / 3.4%	223 cells / 96.6%
Cluster 1 (258 cells)	258 cells / 100%	0 cells / 0%
Cluster 2 (29 cells)	1 cell / 3.4%	28 cells / 96.6%
Correct identified > 70%	258 / 267 (96.62 %)	251 / 251 (100 %)
Correct identified > 80%	258 / 267 (96.62 %	251 / 251 (100 %)
Correct identified > 90%	258 / 267 (96.62 %	251 / 251 (100 %)

(b) ML1 with three clusters

Table 28: Overlaps of detected clusters with known cell types from Ramani *et al.* (2017), ML1 batch. Results computed with Zhou's scHiCluster. Five cells had to be removed because they contained chromosomes with no interactions. Zhou's scHiCluster cannot handle this and crashes.

Cluster	K562 (304 cells)	GM12878 (502 cells)
Cluster 0 (372 cells)	241 cells / 64.78 %	131 cells / 35.22 %
Cluster 1 (434 cells)	63 cells / 14.52 %	371 cells / 85.48 %
Correct identified > 70%	0 / 304 (0.00 %)	371 / 502 (73.90 %)
Correct identified > 80%	0 / 304 (0.00 %)	371 / 502 (73.90 %)
Correct identified > 90%	0 / 304 (0.00 %)	0 / 502 (0.00 %)

(a) ML3 with two clusters

Cluster	K562 (304 cells)	GM12878 (502 cells)
Cluster 0 (177 cells)	23 cells / 12.99 %	154 cells / 87.01 %
Cluster 1 (223 cells)	23 cells / 10.31 %	200 cells / 89.69 %
Cluster 2 (208 cells)	184 cells / 88.46 %	24 cells / 11.54 %
Cluster 3 (58 cells)	54 cells / 93.10 %	4 cells / 6.90 %
Cluster 4 (140 cells)	20 cells / 14.29 %	120 cells / 85.71 %
Correct identified > 70%	238 / 304 (78.29 %)	474 / 502 (94.42 %)
Correct identified > 80%	238 / 304 (78.29 %)	474 / 502 (94.42 %)
Correct identified > 90%	54 / 304 (17.76 %)	0 / 502 (0.00 %)

(b) ML3 with five clusters

Table 29: Overlaps of detected clusters with known cell types from Ramani *et al.* (2017), ML3 batch. Approximate k-nn with MinHash, spectral clustering. More clusters can increase the accuracy of the detected cell types. Parameters: Spectral clustering. MinHash with 5000 hash functions, full-nearest neighbors graph, 13 principal components, inter and intra-chromosomal contacts, umap: n_neighbors 47, min_dist 0.33, n_components 2 for two clusters, n_components 9 for five clusters.

Cluster	K562 (301 cells)	GM12878 (501 cells)
Cluster 0 (384 cells)	248 cells / 64.6%	136 cells / 35.4%
Cluster 1 (418 cells)	53 cells / 12.6%	365 cells / 87.4%
Correct identified > 70%	0 / 301 (0 %)	365 / 501 (72.85 %)
Correct identified > 80%	0 / 301 (0 %)	365 / 501 (72.85 %)
Correct identified > 90%	0 / 301 (0 %)	0 / 501 (0.00 %)

(a) ML3 with two clusters

Cluster	K562 (301 cells)	GM12878 (501 cells)
Cluster 0 (168 cells)	0 cells / 0%	168 cells / 100%
Cluster 1 (134 cells)	16 cells / 11.9%	118 cells / 88.1%
Cluster 2 (201)	195 cells / 97%	6 cells / 3%
Cluster 3 (205)	0 cells / 0%	205 cells / 100%
Cluster 4 (94)	90 cells / 95.7%	4 cells / 4.3%
Correct identified > 70%	285 / 301 (94.68 %)	491 / 501 (98.00 %)
Correct identified > 80%	285 / 301 (94.68 %)	491 / 501 (98.00 %)
Correct identified > 90%	285 / 301 (94.68 %)	373 / 501 (74.45 %)

(b) ML3 with four clusters

Table 30: Overlaps of detected clusters with known cell types from Ramani *et al.* (2017), ML3 batch. Results computed with Zhou's scHiCluster. More clusters can increase the accuracy of the detected cell types. Four cells had to be removed because they contained chromosomes with no interactions. Zhou's scHiCluster cannot handle this and crashes.

8 Cluster results on Ramani 10 kb

Cluster	HeLa (269 cells)	HAP1 (254 cells)
Cluster 0 (278 cells)	121 cells / 43.53 %	157 cells / 56.47 %
Cluster 1 (245 cells)	148 cells / 60.41 %	97 cells / 39.59 %
Correct identified > 70%	0 / 269 (0.00 %)	0 / 254 (0.00 %)
Correct identified > 80%	0 / 269 (0.00 %)	0 / 254 (0.00 %)
Correct identified > 90%	0 / 269 (0.00 %)	0 / 254 (0.00 %)

Table 31: Ramani ML1 data with two clusters. 10 kb resolution.

Cluster	K562 (304 cells)	GM12878 (502 cells)
Cluster 0 (478 cells)	106 cells / 22.18 %	372 cells / 77.82 %
Cluster 1 (328 cells)	198 cells / 60.37 %	130 cells / 39.63 %
Correct identified > 70%	0 / 304 (0.00 %)	372 / 502 (74.10 %)
Correct identified > 80%	0 / 304 (0.00 %)	0 / 502 (0.00 %)
Correct identified > 90%	0 / 304 (0.00 %)	0 / 502 (0.00 %)

Table 32: Ramani ML3 data with two clusters, 10 kb resoltion.

9 Runtime and memory usage

The measurement of runtimes of algorithms with a high I/O and the requirement for a fast parallelization are very environment dependent. To give a broader overview, the here presented numbers are from a virtual machine with NFS storage and a state-of-the-art computer with a modern SSD. To show the impact of the number of hash functions, run times and memory usage are shown with a low number and a high number of hash function.

Method	Runtime	Memory
Raw and K-Means	39:15 min	7.2 GB
Raw and Spectral	01:29 min	4.5 GB
PCA and K-Means	05:37 min	170 GB
PCA and Spectral	05:35 min	170 GB
scikit-learn k-nn k = 2472 and k-means	01:19 min	4.5 GB
scikit-learn k-nn k = 2472 and Spectral	01:25 min	4.5 GB
scHicClusterMinHash k = 2472 and k-means	01:30 min	7.6 GB
scHicClusterMinHash k = 2472 and Spectral	01:35 min	7.6 GB
scHicClusterMinHash eucl. k = 2472 and k-means	02:04 min	7.6 GB
scHicClusterMinHash eucl. k = 2472 and Spectral	02:04 min	7.6 GB
Zhou's scHiCluster CPU	13:55 min	4.0 GB

Table 33: Runtimes and memory usage with 1 Mb 2472 cells on a single-cell Hi-C matrix. Data from Nagano *et al.* (2017) Diploid cells, with 12 clusters. For clustering k-means and spectral clustering are used, scHicClusterMinHash with 800 hash functions, k=2472, applied PCA and 100 principal components for clustering. All results computed on 2x Intel XEON E5-2630 v4 @ 2.20GHz 2x 10 cores / 2x 20 threads, 1 TB memory.

Method	Runtime	Memory
Raw and K-Means	12:22 min	7.2 GB
Raw and Spectral	1:37 min	4.0 GB
PCA and K-Means	-	> 128 GB
PCA and Spectral	-	> 128 GB
scikit-learn k-nn k = 2472 and K-means	1:24 min	4.0 GB
scikit-learn k-nn k = 2472 and Spectral	1:26 min	4.0 GB
MinHash k = 2472 and K-means	0:57 min	7.6 GB
MinHash k = 2472 and Spectral	0:59 min	7.6 GB
MinHash euclidean k = 2472 and K-means	1:55 min	7.6 GB
MinHash euclidean k = 2472 and Spectral	1:56 min	7.6 GB
Zhou's scHiCluster CPU	14:02 min	4.0 GB
Zhou's scHiCluster GPU	07:17 min	3.7 GB

Table 34: Runtimes and memory usage with 1 Mb on a single-cell Hi-C matrix with 2472 cells. Data from Nagano *et al.* (2017) Diploid cells, with 12 clusters. For clustering K-means and spectral clustering are used, MinHash with 800 hash functions and activated PCA. All results computed on AMD Ryzen 3700X 8 cores / 16 threads, 128 GB memory; Nvidia GTX 1070 8 GB memory.

9.1 High number of hash functions

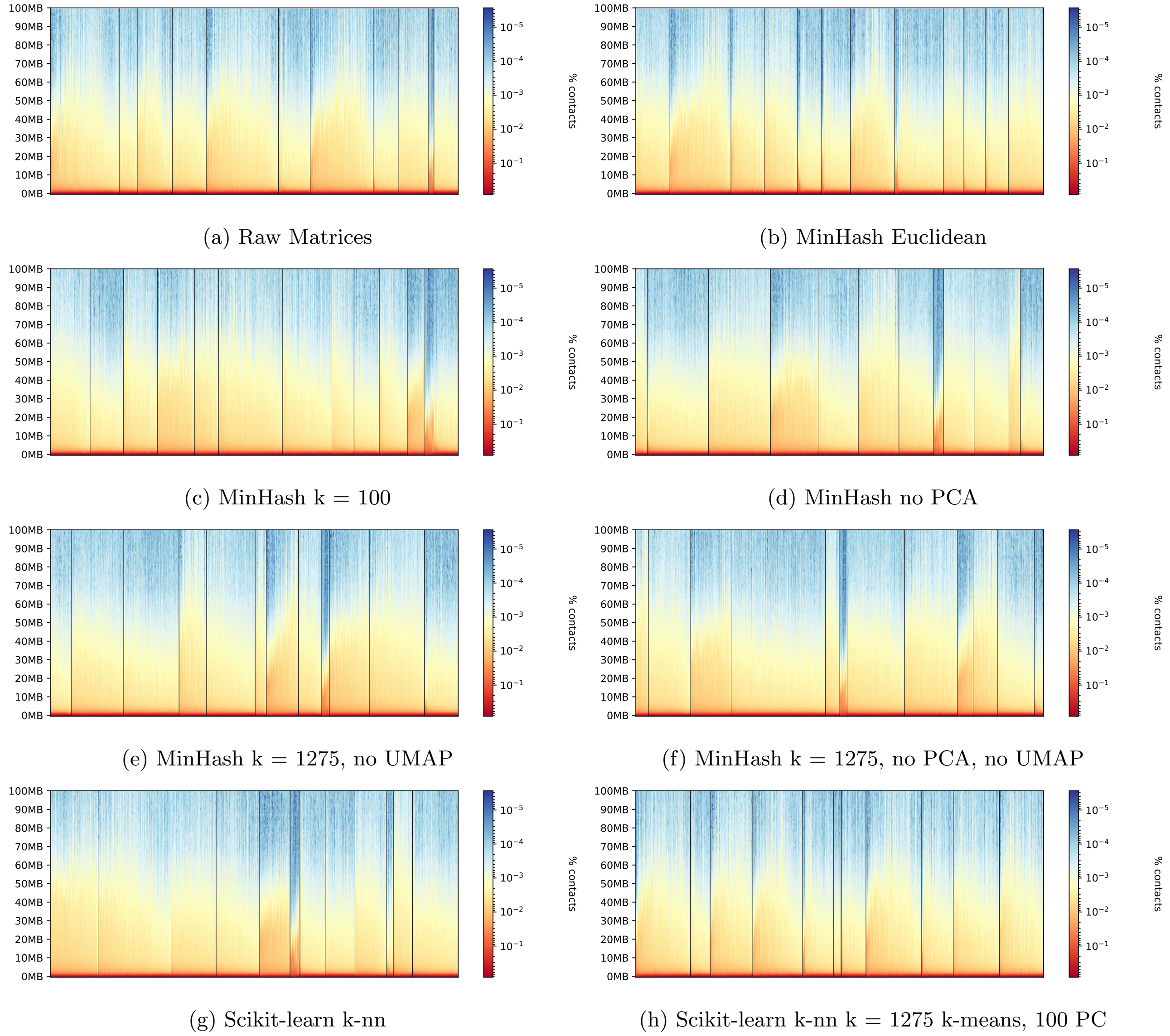
Method	Runtime	Memory
MinHash h = 800	0:42 min	1.6 GB
MinHash h = 2000	0:47 min	1.6 GB
MinHash h = 8000	1:10 min	1.7 GB
MinHash h = 15000	1:36 min	1.9 GB
MinHash h = 20000	2:00 min	2 GB
Zhou's scHiCluster CPU	6:50 min	2.4 GB
Zhou's scHiCluster GPU	3:40 min	2.7 GB

Table 35: Runtimes and memory usage with 1 Mb resolution on a single-cell Hi-C matrix with 1275 cells. Normalized to a read coverage of 100,000 reads and interaction values smaller 1 are *kept.* Data from Nagano *et al.* (2017) Diploid cells, with 12 clusters. For clustering spectral clustering is used, MinHash with a different number of hash functions *h*. All results computed on AMD Ryzen 3700X 8 cores / 16 threads, 128 GB memory; Nvidia GTX 1070 8 GB memory.

Method	Runtime	Memory
MinHash h = 800	04:04 min	13.9 GB
MinHash h = 4000	05:42 min	14.0 GB
MinHash h = 8000	07:33 min	14.1 GB
MinHash h = 15000	11:05 min	14.3 GB
MinHash h = 20000	13:30 min	14.5 GB
MinHash h = 40000	23:27 min	15.2 GB

Table 36: Runtimes and memory usage with 10 kb resolution on a single-cell Hi-C matrix with 1088 cells. Normalized to a read coverage of 100,000 reads and interaction values smaller 1 are *kept.* Data from Nagano *et al.* (2017) Diploid cells, with 14 clusters. For clustering spectral clustering is used, MinHash with a different number of hash functions *h*. All results computed on AMD Ryzen 3700X 8 cores / 16 threads, 128 GB memory; Nvidia GTX 1070 8 GB memory.

10 Cluster profiles Nagano data



(a) Raw Matrices

(b) MinHash Euclidean

(c) MinHash k = 100

(d) MinHash no PCA

(e) MinHash k = 1275, no UMAP

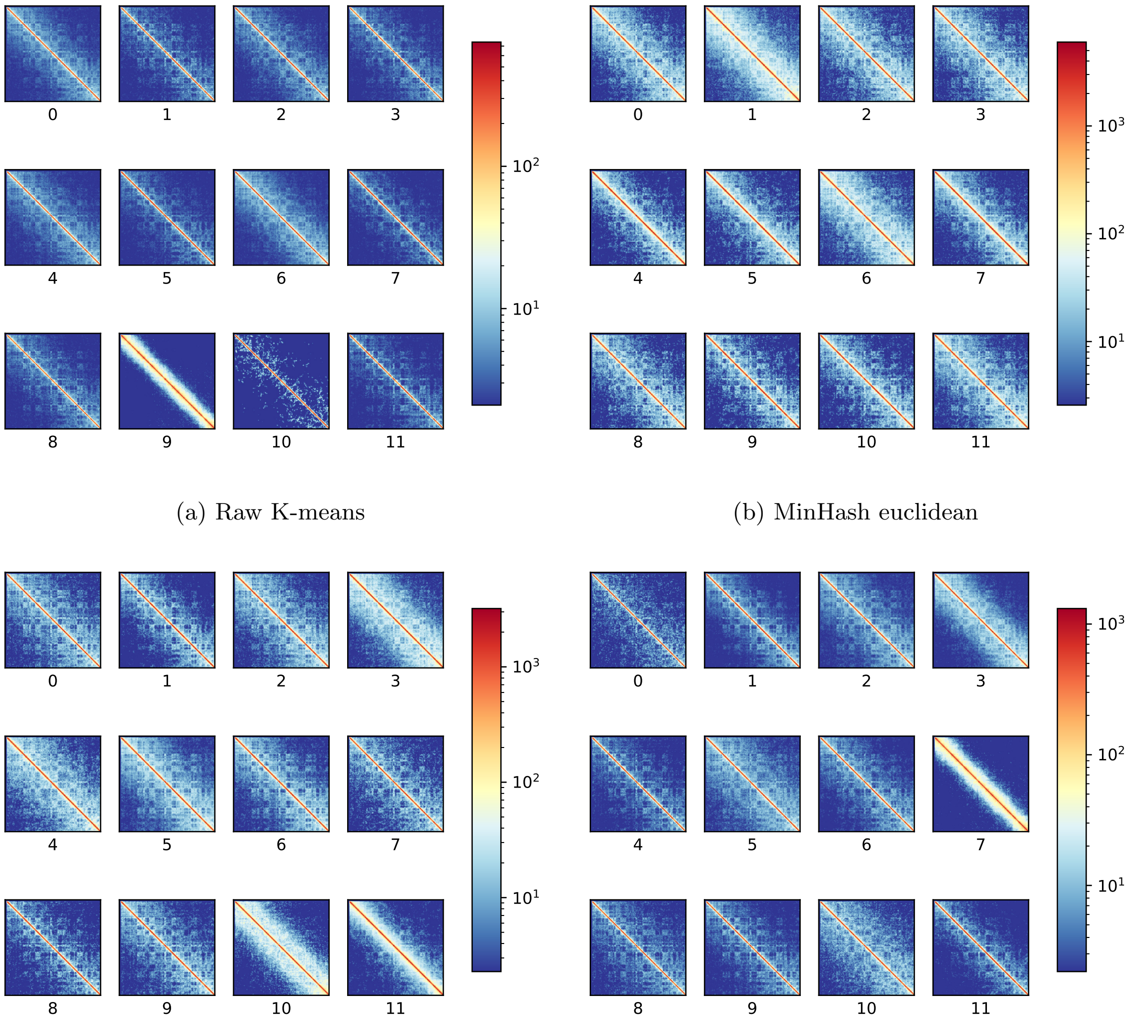
(f) MinHash k = 1275, no PCA, no UMAP

(g) Scikit-learn k-nn

(h) Scikit-learn k-nn k = 1275 k-means, 100 PC

Figure S 14: Cluster profile of the different clusters on 1275 cells from Nagano *et al.* (2017) Diploid cells. K-Means clustering was applied on all datasets, MinHash with intra-chromosomal data, PCA, UMAP and a full k-nn if not defined otherwise. Clustering on raw single-cell Hi-C interaction matrix (S 14a, S 14b k-nn with MinHash and the additional euclidean distance computation; S 14c MinHash with 100 nearest neighbors; S 14d MinHash without an intermediate PCA on the k-nn, S 14e MinHash with a PCA but no UMAP, S 14f MinHash only, without PCA and UMAP. S 14g shows the results if inter- and intrachromosomal data are used to create the k-nn with MinHash; S 14h shows the result of k-means applied on a k-nn with k=1275 using Scikit-learns' k-nn implementation.

11 Consensus matrices Nagano data



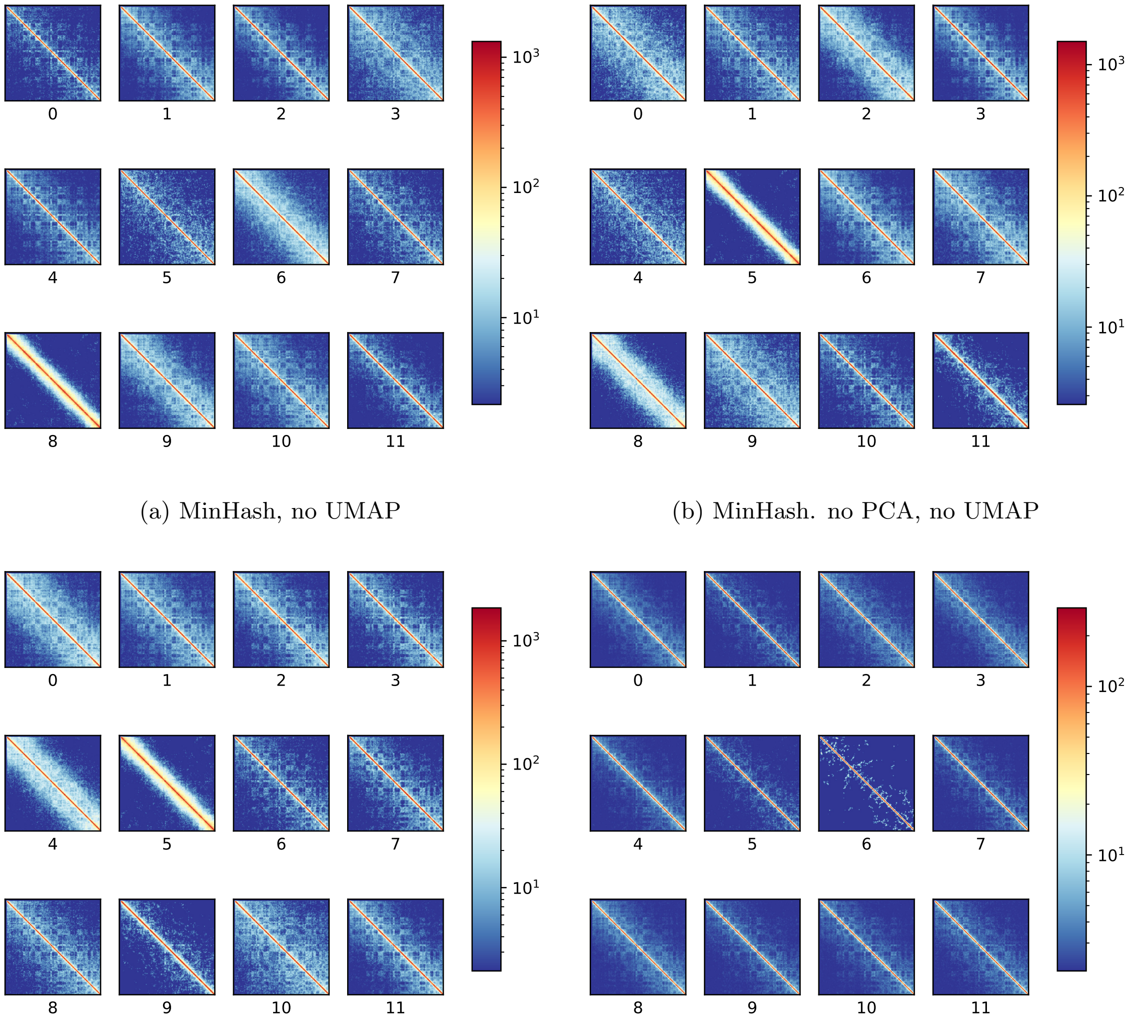
(a) Raw K-means

(b) MinHash euclidean

(c) MinHash k = 100

(d) MinHash, no PCA

Figure S 15: Consensus matrices of the different clusters on 1275 cells from Nagano *et al.* (2017) Diploid cells, chromosome 6. K-Means clustering was applied on all datasets, MinHash with intra-chromosomal data, PCA, UMAP and a full k-nn if not defined otherwise. Clustering on raw single-cell Hi-C interaction matrix (S 15a, S 15b k-nn with MinHash and the additional euclidean distance computation; S 15c MinHash with 100 nearest neighbors; S 15d MinHash without an intermediate PCA on the k-nn



(a) MinHash, no UMAP

(b) MinHash. no PCA, no UMAP

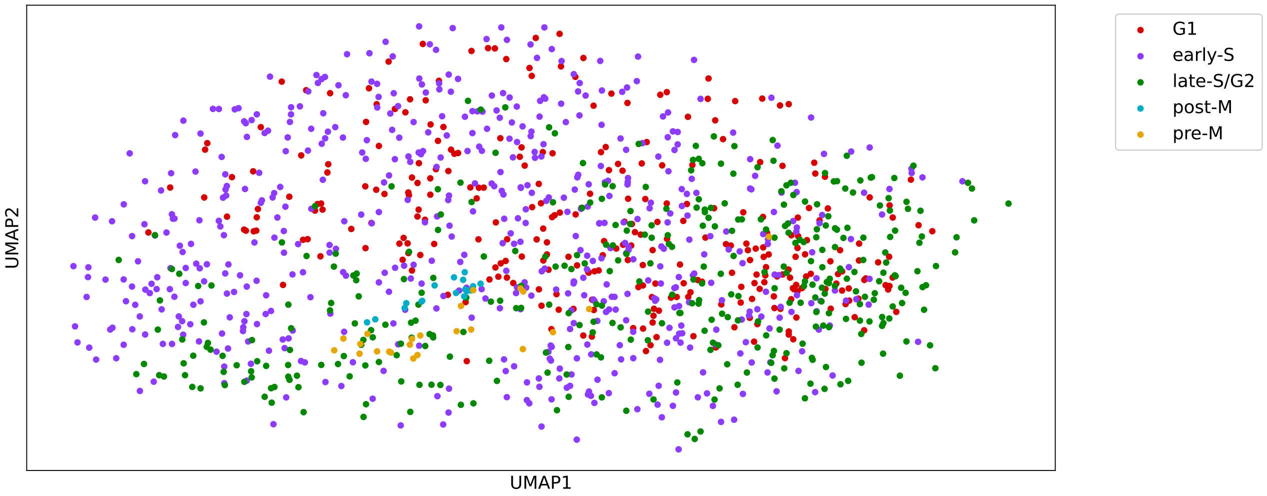
(c) inter- and intra-chromosomal contacts

(d) Scikit-learn k-nn

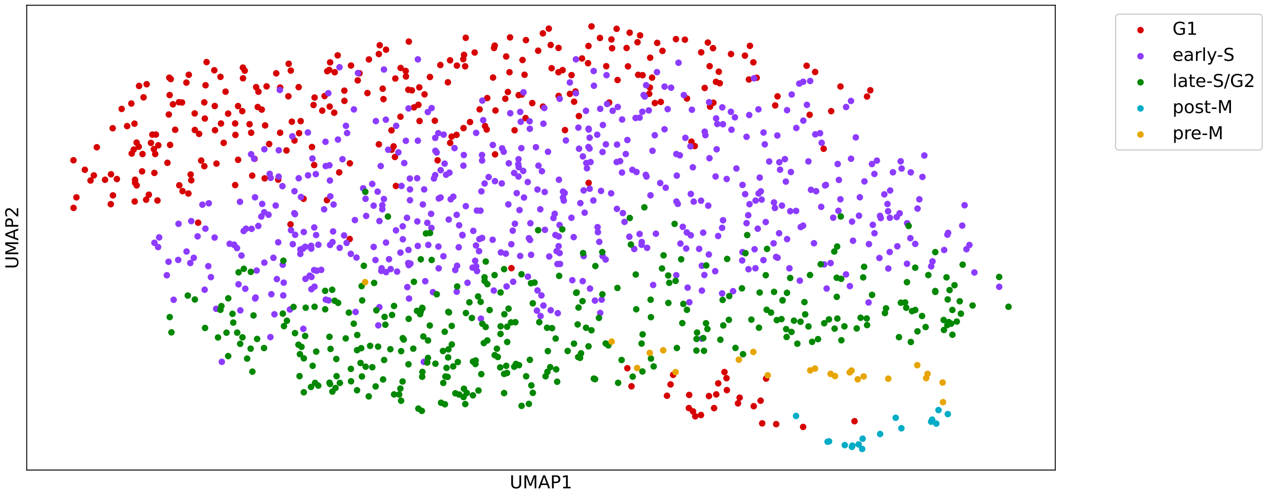
Figure S 16: Consensus matrices of the different clusters on 1275 cells from Nagano *et al.* (2017) Diploid cells, chromosome 6. K-Means clustering was applied on all datasets, MinHash with intra-chromosomal data, PCA, UMAP and a full k-nn if not defined otherwise. S 16a MinHash with a PCA but no UMAP, S 16b MinHash only, without PCA and UMAP. S 16c shows the results if inter- and intrachromosomal data are used to create the k-nn with MinHash; S 16d shows the result of k-means applied on a k-nn with k=1275 using scikit-learns k-nn implementation.

12 Scatter plots cell labels

12.1 Nagano 1 Mb

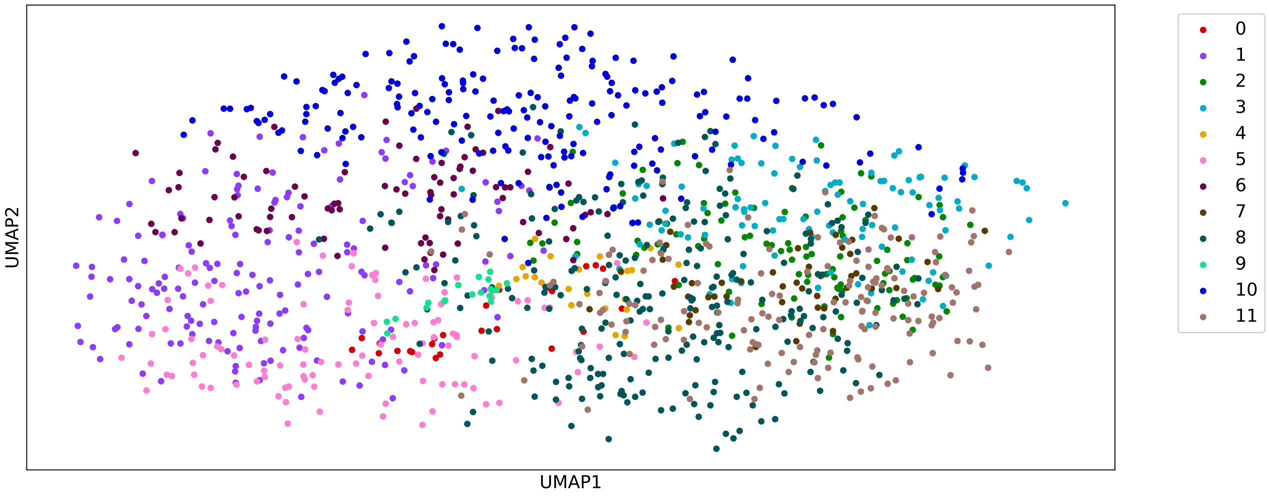


(a) k-nn MinHash on Nagano 1MB cell coloring UMAP dimensions 5

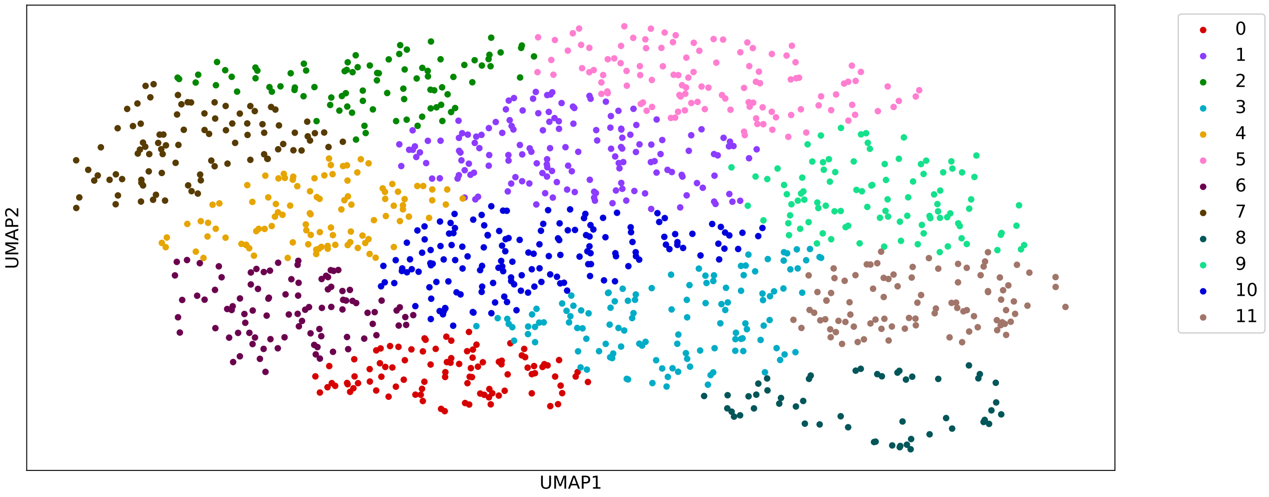


(b) k-nn MinHash on Nagano 1MB cell coloring UMAP dimensions 2

Figure S 17

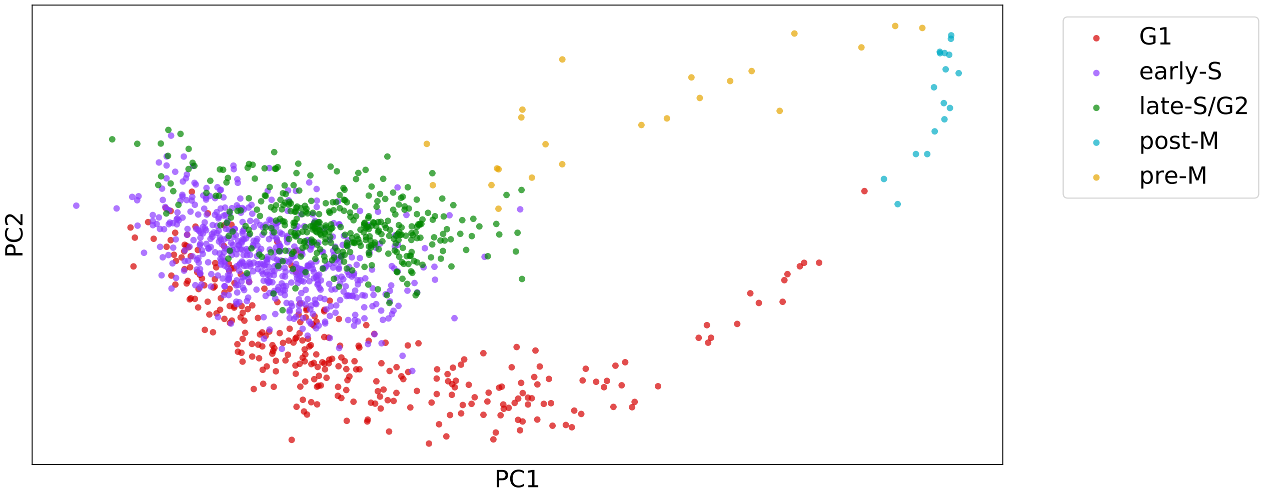


(c) k-nn MinHash on Nagano 1MB cell cluster result UMAP dimensions 5

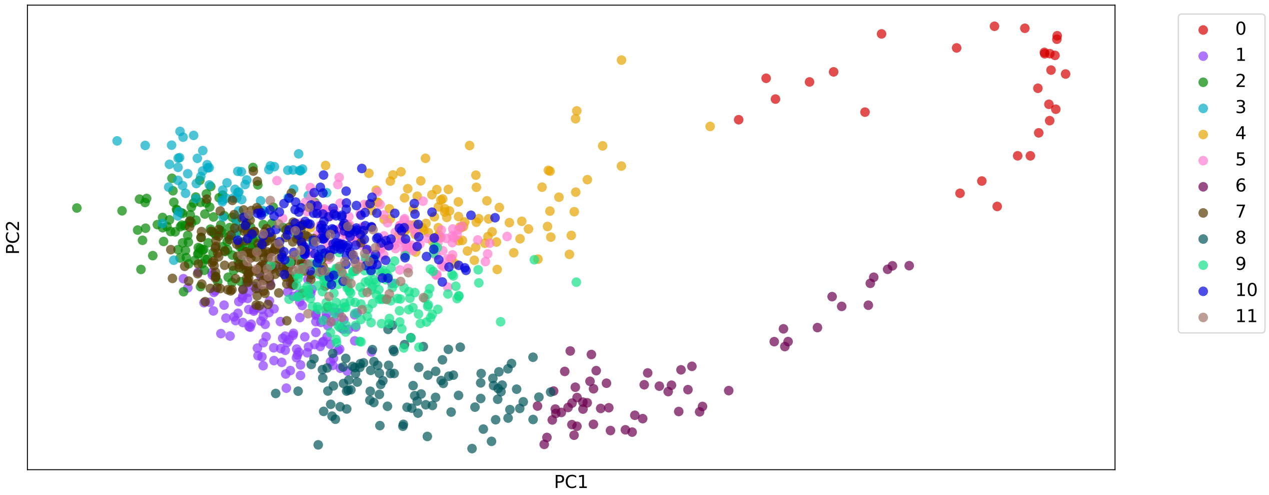


(d) k-nn MinHash on Nagano 1MB cell cluster result UMAP dimensions 2

Figure S 17



(a) Zhou's scHiCluster Nagano 1MB cell coloring

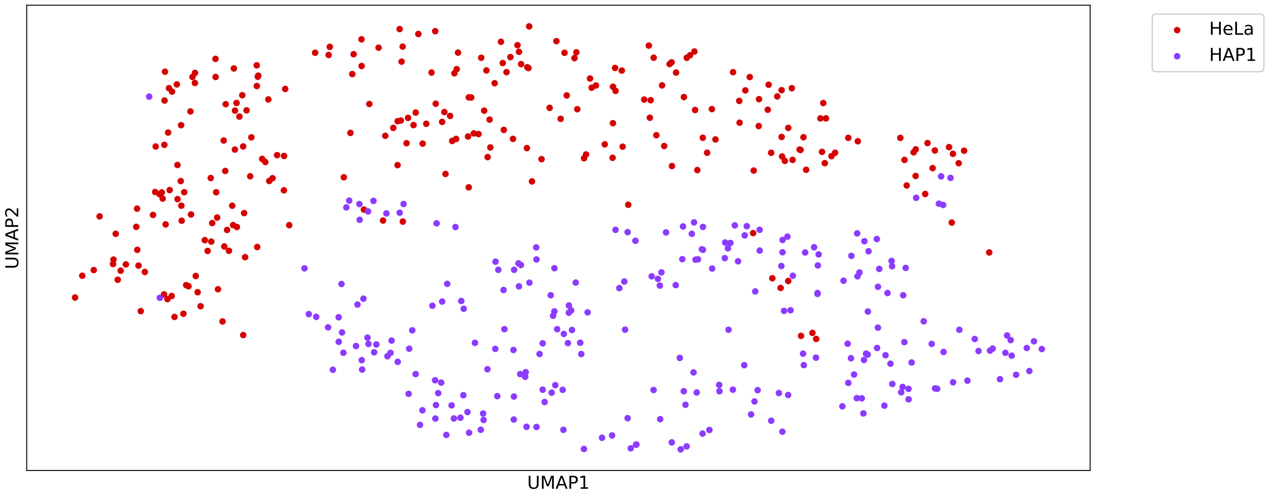


(b) Zhou's scHiCluster Nagano 1MB cell cluster result

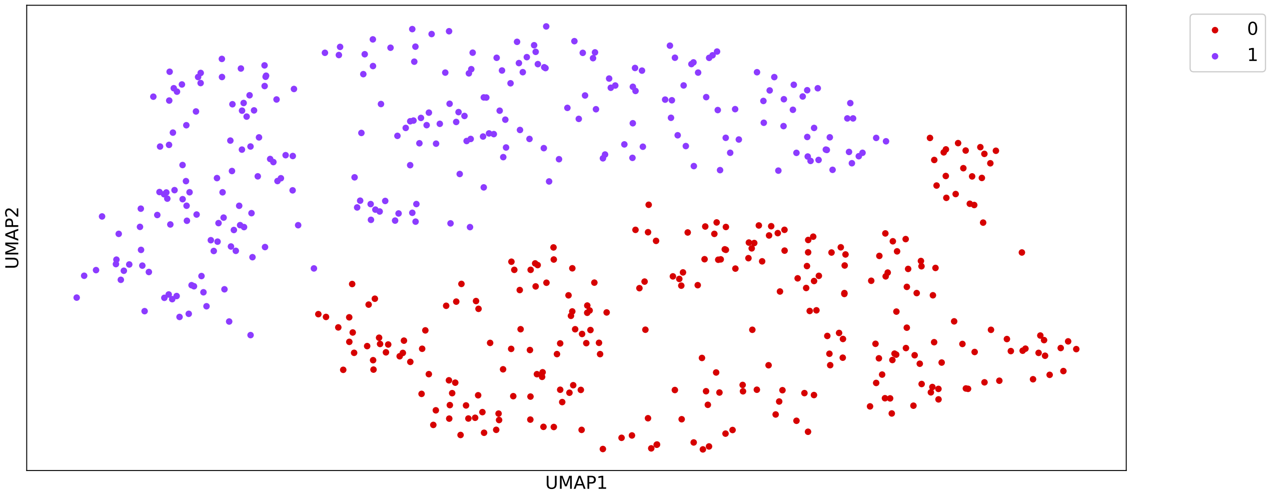
Figure S 18

12.2 Ramani 1 Mb

12.2.1 approximate k-nn with MinHash

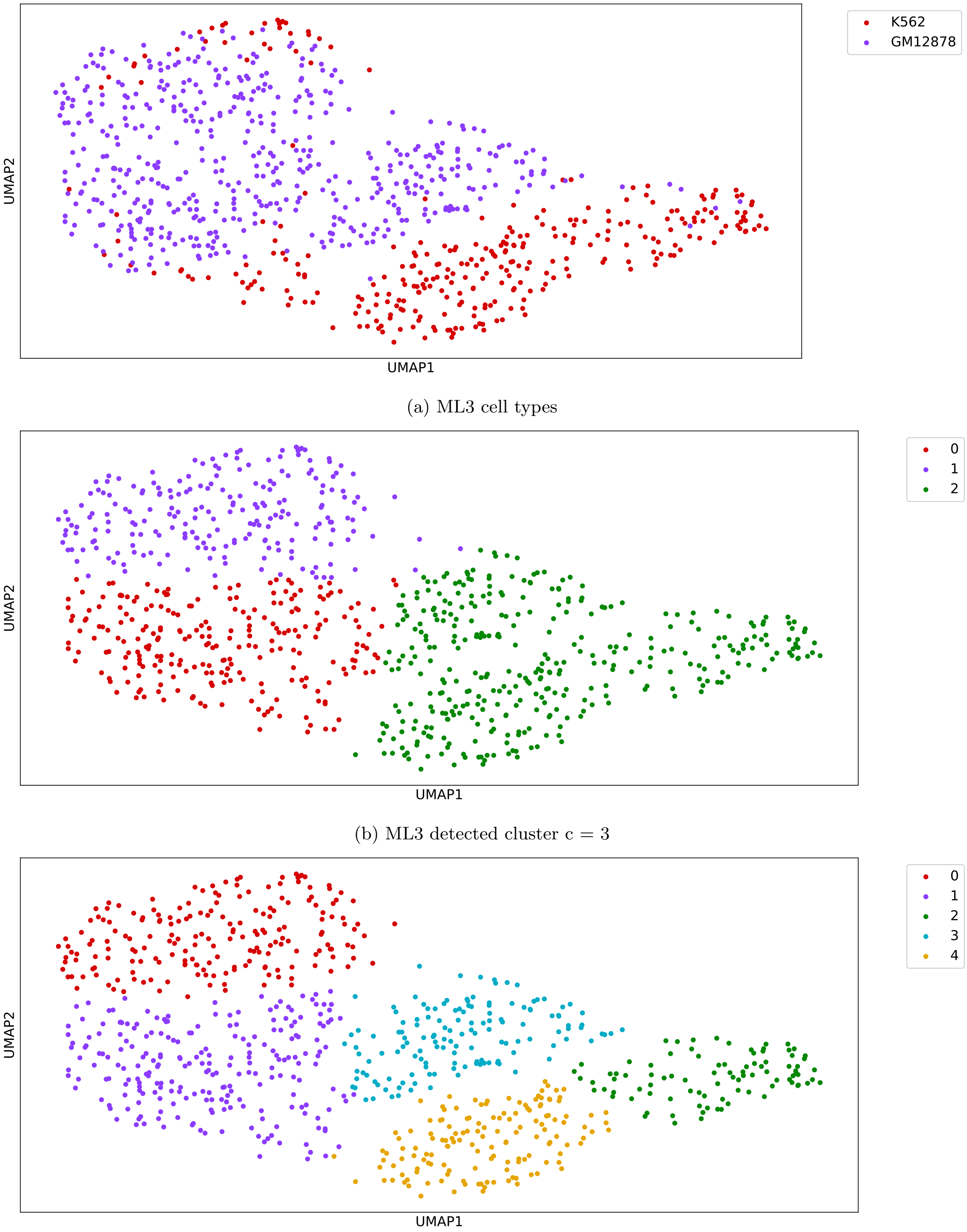


(a) ML1 cell types



(b) ML1 detected cluster c = 2

Figure S 19: Embedding of Ramani cell type data. 1 MB resolution, ML1 with approximate k-nn based on MinHash, 8 principal components, UMAP embedding and spectral clustering.



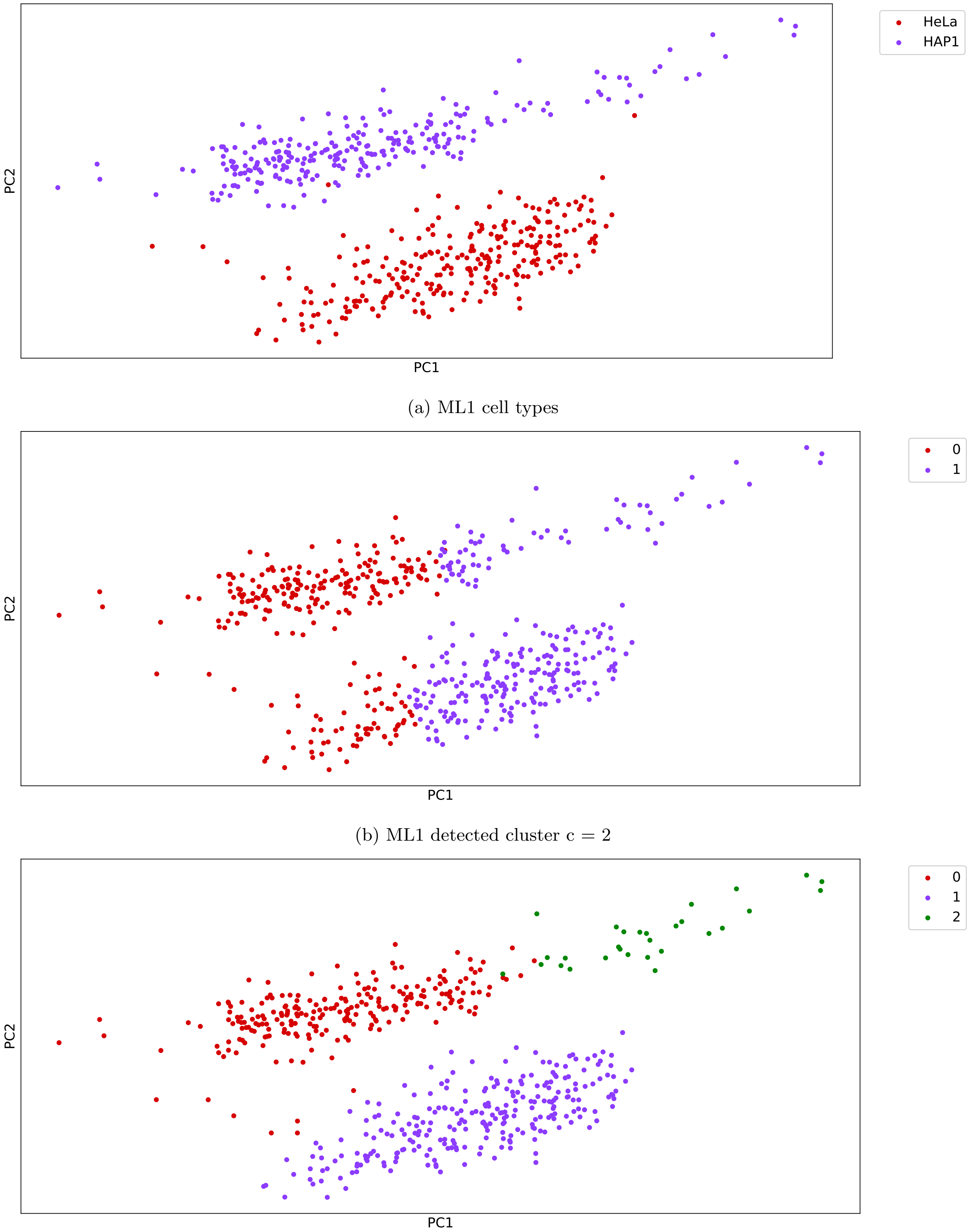
(a) ML3 cell types

(b) ML3 detected cluster c = 3

(c) ML3 detected cluster c = 5

Figure S 20: Embedding of Ramani cell type data. 1 MB resolution, ML3, with approximate k-nn based on MinHash, 8 principal components, UMAP embedding and spectral clustering. To detect more clusters than cell types can be beneficial.

12.2.2 Zhou's scHiCluster

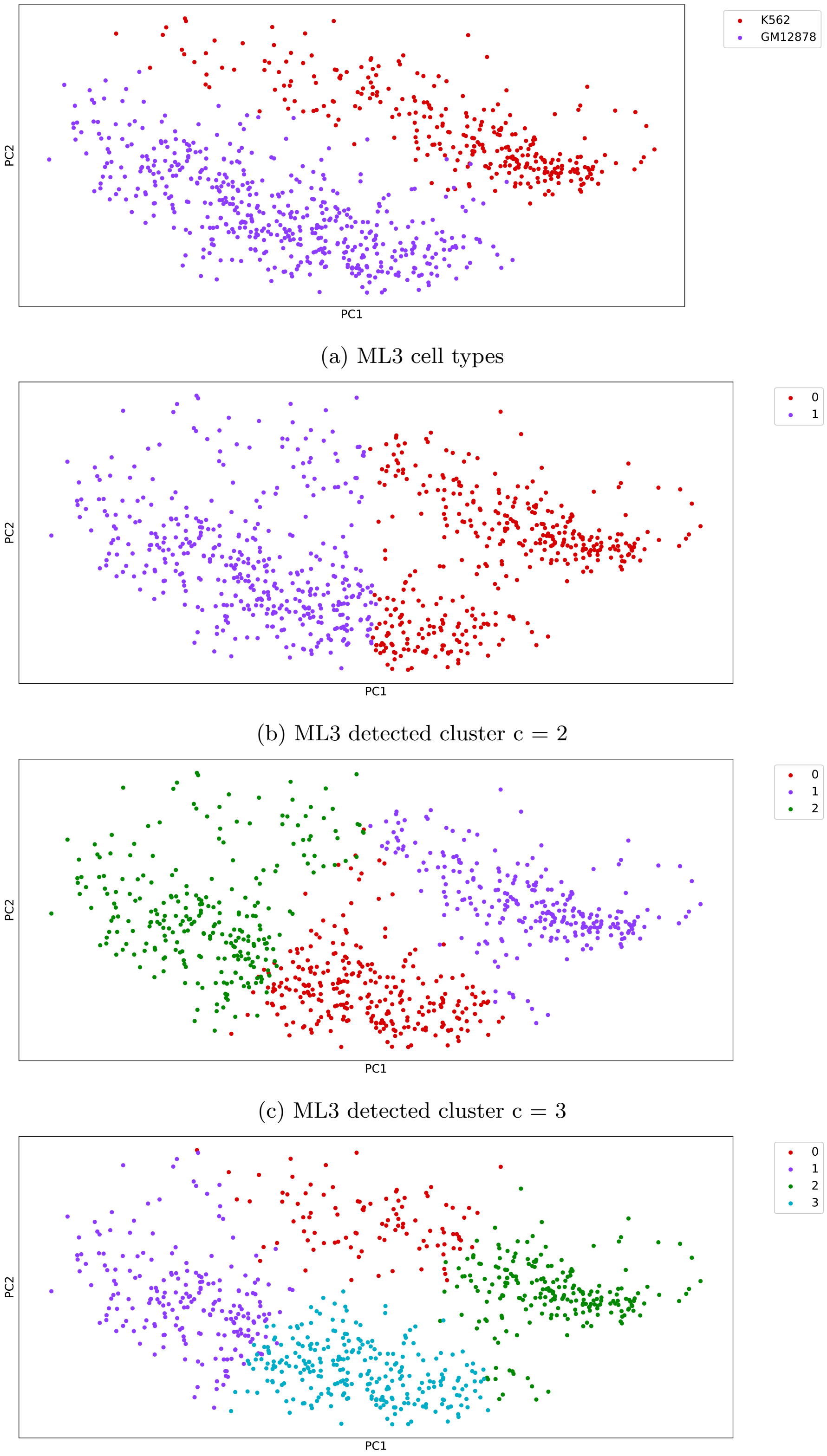


(a) ML1 cell types

(b) ML1 detected cluster c = 2

(c) ML1 detected cluster c = 3

Figure S 21: Embedding of Ramani cell type data. 1 MB resolution, ML1 with Zhou's scHiCluster. To detect more clusters than cell types can be beneficial.



(a) ML3 cell types

(b) ML3 detected cluster c = 2

(c) ML3 detected cluster c = 3

(d) ML3 detected cluster c = 4

Figure S 22: Embedding of Ramani cell type data. 1 MB resolution, ML3, with Zhou's scHiCluster. To detect more clusters than cell types can be beneficial.

13 Data collection and pre-processing

All used data is from Nagano *et al.* (2017): GEO94489; Gassler *et al.* (2017): GSE100569 and Ramani *et al.* (2017): GSE84920; and was pre-processed with scHiCExplorer (Wolff *et al.* (2020a)) version 7[*]. The raw data was quality controlled and read coverage normalized, it is available on Zenodo[†]. The single-cell Hi-C interaction matrices are stored in the *scool*[‡] file format (Wolff *et al.* (2020b)), available in the cooler (Abdennur and Mirny (2019)) package since version 0.8.9.

[*] https://github.com/joachimwolff/scHiCExplorer/tree/7

[†] https://doi.org/10.5281/zenodo.4308298

[‡] https://cooler.readthedocs.io/en/latest/schema.html#single-cell-single-resolution

References

Abdennur, N. and Mirny, L. A. (2019). Cooler: scalable storage for Hi-C data and other genomically labeled arrays. *Bioinformatics*, **36**(1), 311–316.

Gassler, J. *et al.* (2017). A mechanism of cohesin-dependent loop extrusion organizes zygotic genome architecture. *The EMBO journal*, **36**(24), 3600–3618.

Nagano, T. *et al.* (2017). Cell-cycle dynamics of chromosomal organization at single-cell resolution. *Nature*, **547**(7661), 61.

Ramani, V. *et al.* (2017). Massively multiplex single-cell hi-c. *Nature methods*, **14**(3), 263–266.

Wolff, J. *et al.* (2020a). Galaxy HiCExplorer 3: a web server for reproducible Hi-C, capture Hi-C and single-cell Hi-C data analysis, quality control and visualization. *Nucleic Acids Research*. gkaa220.

Wolff, J. *et al.* (2020b). Scool: a new data storage format for single-cell Hi-C data. *Bioinformatics*. btaa924.

A.3 Appendix for A new data storage format for single-cell Hi-C data

Supplementary material
scool: A new data storage format for single-cell Hi-C data

Joachim Wolff[1,*], Nezar Abdennur[2], Rolf Backofen[1,3], Björn Grüning[1]

[1]Bioinformatics Group, Department of Computer Science, University of Freiburg, Georges-Köhler-Allee 106, 79110 Freiburg, Germany

[2] Institute for Medical Engineering and Science, Massachusetts Institute of Technology, Cambridge, MA, 02139, USA

[3]Signalling Research Centres BIOSS and CIBSS, University of Freiburg, Schänzlestr. 18, 79104 Freiburg, Germany

*To whom correspondence should be addressed.

1 Create scool alternative with pandas dataframe

```
import cooler

pixel_dict = {'cell1' : pixels1, 'cell2' : pixels2}

cooler.create_scool(cool_uri=pFileName, bins=bins1, cell_name_pixels_dict=pixel_dict)
```

Listing 1: Python API example to create a scool file, *bins1* is a pandas dataframe

2 File sizes and compression rates

Files	txt	zip txt	cool	zip cool	scool	zip scool
Nagano 1 Mb	958.1 MB	273.3 MB	349.8 MB	193 MB	266.9 MB	181.5 MB
Nagano 10 kb	8.0 GB	2.2 GB	3.0 GB	2.5 GB	1.9 GB	1.8 GB
Gassler 100 kb	111.9 MB	31.2 MB	63.5 MB	56.8 MB	26.3 MB	23.0 MB
Gassler 40 kb	170.7 MB	46.6 MB	120.7 MB	111.9 MB	39.3 MB	35.9 MB
Gassler 10 kb	277.1 MB	72.3 MB	348.9 MB	341.1 MB	67.5 MB	62.8 MB
Gassler 1 kb	438.6 MB	115.8 MB	2.3 GB	1.8 GB	120.8 MB	112.5 MB

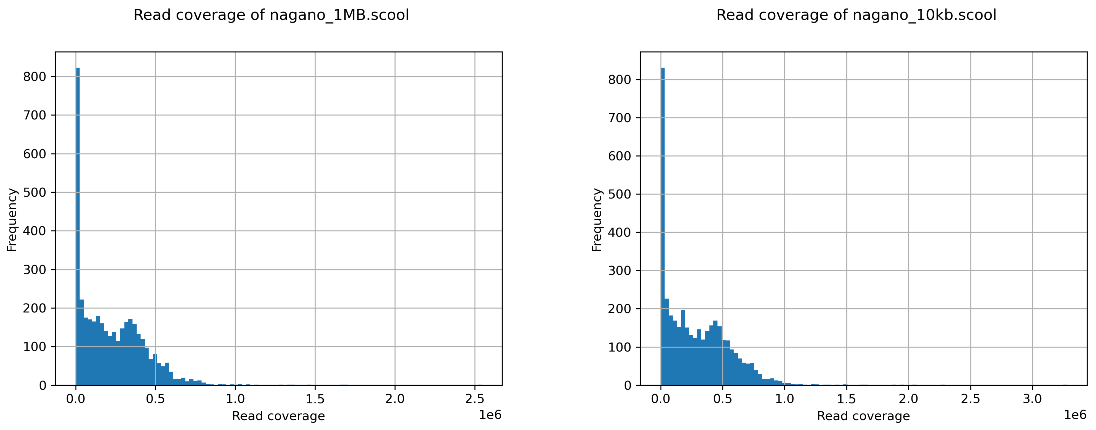
Table 1: File sizes of aggregate file sizes of all cells respectively of one file in case of scool or the compressed versions. As compression Ubuntu 20.04 GUI 'Compress...' with zip is used. The txt files contain only the sparse matrix information but no genomic position relation or metadata. Data from Nagano *et al.* (2017) and Gassler *et al.* (2017).

Files	zip txt / txt	scool / cool	zip scool / zip cool	scool / zip txt
Nagano 1 Mb	0.285	0.763	0.940	0.976
Nagano 10 kb	0.275	0.633	0.720	0.818
Gassler 100 kb	0.278	0.414	0.404	0.737
Gassler 40 kb	0.277	0.325	0.320	0.770
Gassler 10 kb	0.260	0.193	0.184	0.868
Gassler 1 kb	0.264	0.052	0.062	0.971

Table 2: File sizes from Table 1, the smaller the number, the higher the compression. The txt files contain only the sparse matrix information but no genomic position relation or metadata. Data from Nagano *et al.* (2017) and Gassler *et al.* (2017).

3 Read coverage distribution

3.1 Nagano 2017

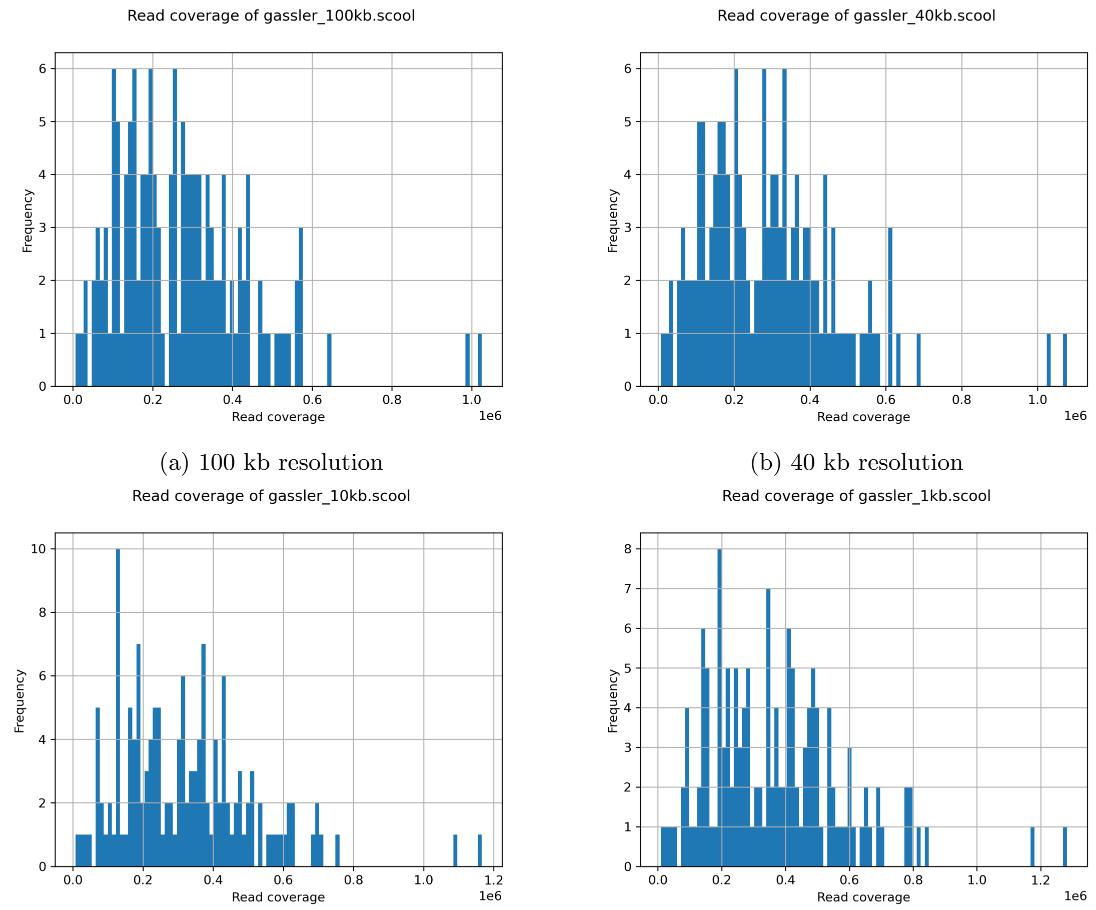


(a) 1 Mb resolution

(b) 10 kb resolution

Figure 1: Read coverage distribution over all 3882 cells from Nagano *et al.* (2017).

3.2 Gassler 2017



(a) 100 kb resolution

(b) 40 kb resolution

(c) 10 kb resolution

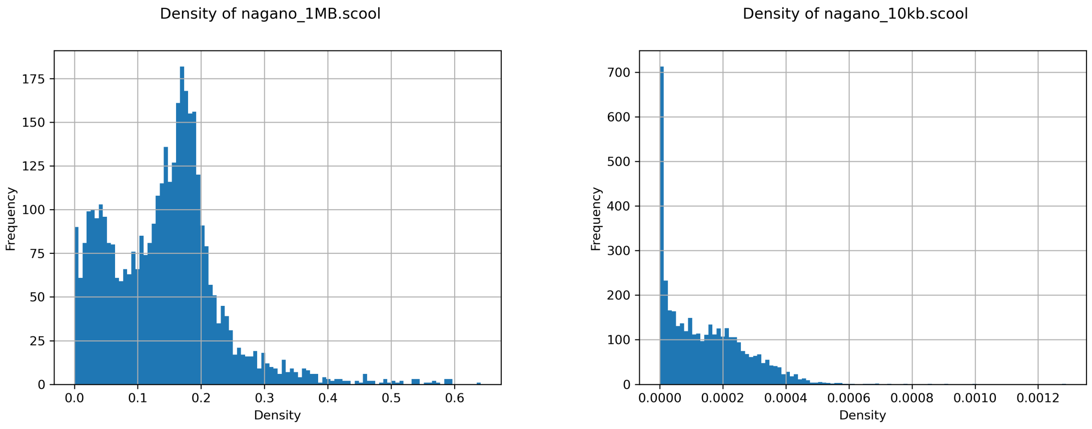
(d) 1 kb resolution

Figure 2: Read coverage distribution over all 144 cells from Gassler *et al.* (2017).

4 Read density distribution

All read densities are computed for contacts within a 30 Mb genomic distance. The very few long range contacts are excluded in this computation.

4.1 Nagano 2017

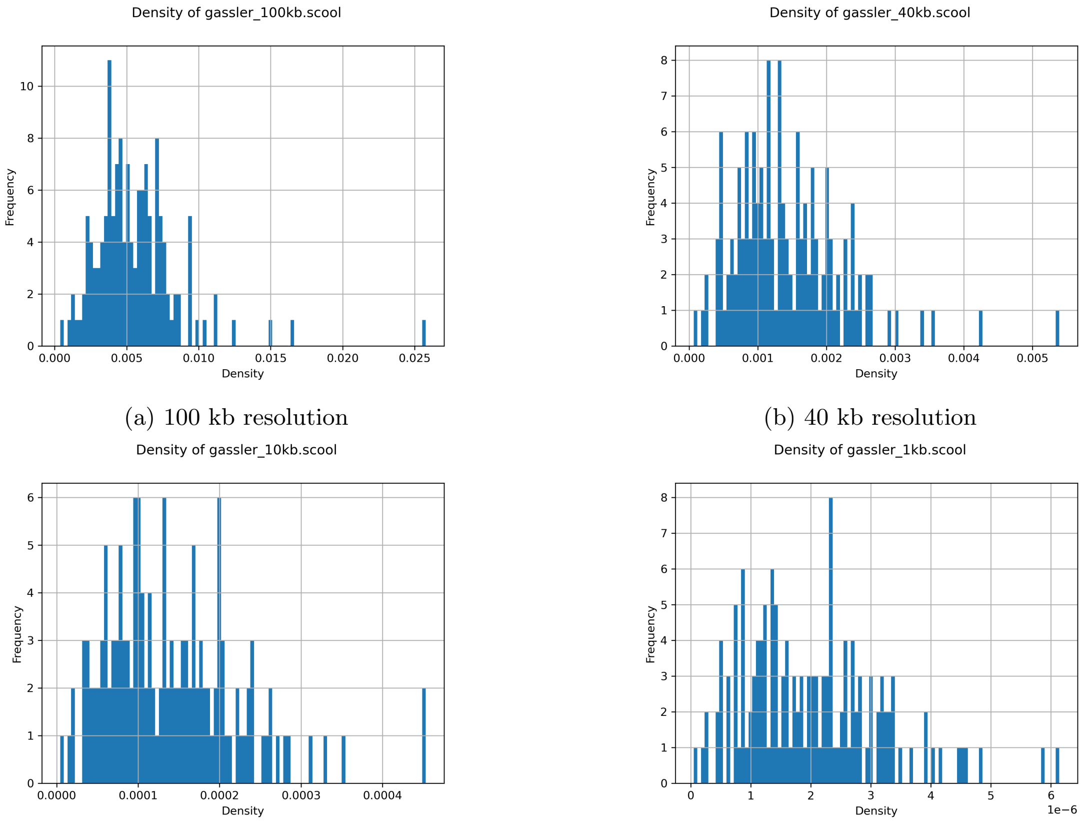


(a) 1 Mb resolution

(b) 10 kb resolution

Figure 3: Read density distribution over all 3882 cells from Nagano *et al.* (2017).

4.2 Gassler 2017



(a) 100 kb resolution

(b) 40 kb resolution

(c) 10 kb resolution

(d) 1 kb resolution

Figure 4: Read density distribution over all 144 cells from Gassler *et al.* (2017).

References

Gassler, J. *et al.* (2017). A mechanism of cohesin-dependent loop extrusion organizes zygotic genome architecture. *The EMBO journal*, **36**(24), 3600–3618.

Nagano, T. *et al.* (2017). Cell-cycle dynamics of chromosomal organization at single-cell resolution. *Nature*, **547**(7661), 61.

A.4 Appendix for pyGenomeTracks: Reproducible plots for multivariate genomic

Supplementary material: pyGenomeTracks: Reproducible plots for multivariate genomic data sets

Lucille Lopez-Delisle[1], Leily Rabbani[2], Joachim Wolff[3], Vivek Bhardwaj[2], Rolf Backofen[3,4], Björn Grüning[3], Fidel Ramírez[2,*], and Thomas Manke[2]

[1]EPFL SV ISREC UPDUB, 1015 Lausanne, Switzerland
[2]Max Planck Institute of Immunobiology and Epigenetics, Stübeweg 51, 79108 Freiburg im Breisgau, Germany
[3] Bioinformatics Group, Department of Computer Science, University of Freiburg, Georges-Köhler-Allee 106, 79110 Freiburg, Germany
[4]Signalling Research Centres BIOSS and CIBSS, University of Freiburg, Schänzlestr. 18, 79104 Freiburg, Germany

[*]To whom correspondence should be addressed.

1 *make_tracks_file* or how to generate a configuration file

The script *make_tracks_file* can generate a configuration file from input file(s).

The full documentation is available on https://pygenometracks.readthedocs.io.

```
$ make_tracks_file --trackFiles file1.bed file2.bw -o tracks.ini
```

Will output with version 3.5

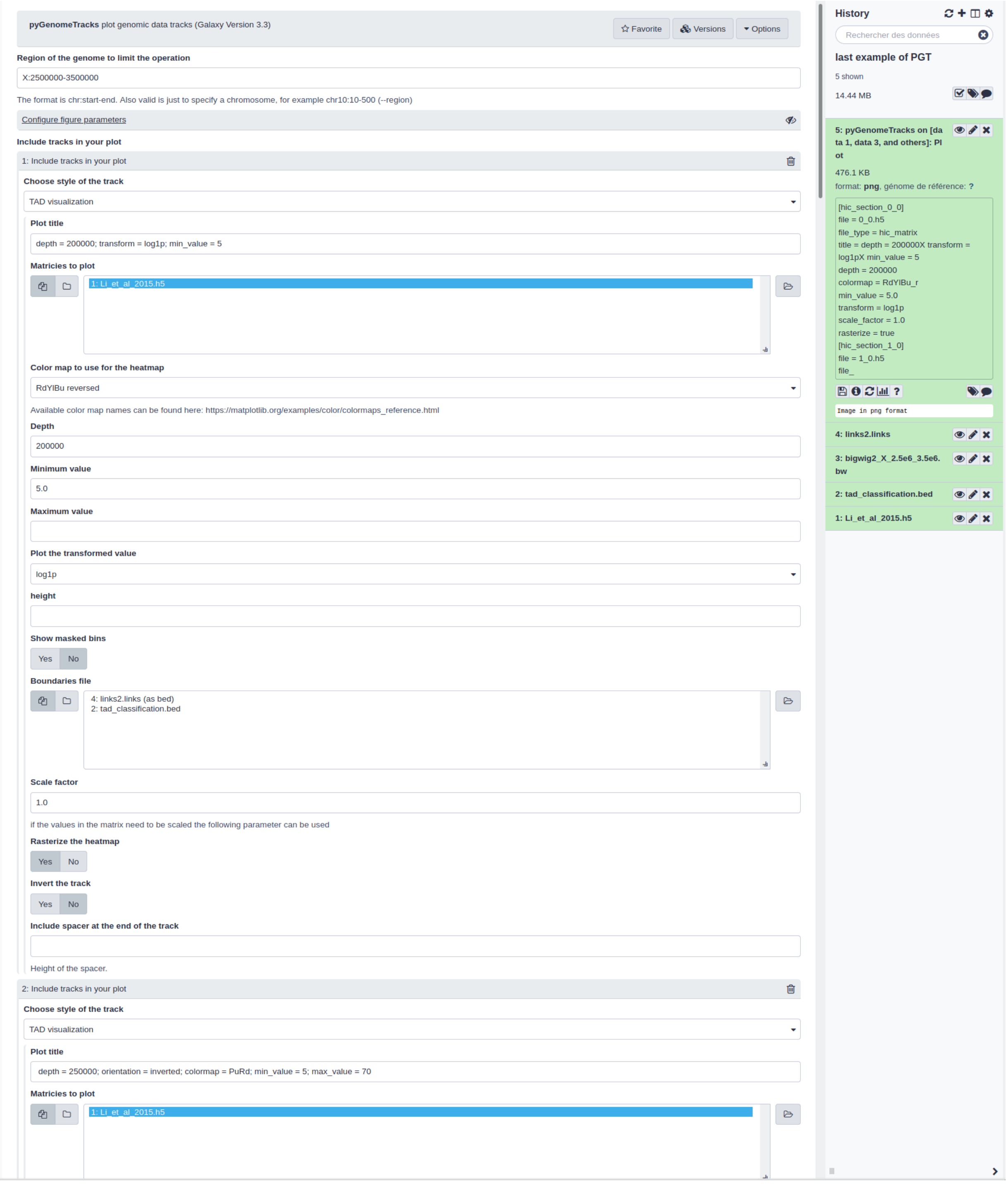
```
[x-axis]
#optional
#fontsize = 20
# default is bottom meaning below the axis line
# where = top

[spacer]
# height of space in cm (optional)
height = 0.5


[file1]
file = file1.bed

# title of track (plotted on the right side)
title = file1
# height of track in cm (ignored if the track is overlay on top the previous track)
height = 2
# if you want to plot the track upside-down:
# orientation = inverted
# if you want to plot the track on top of the previous track. Options are 'yes' or 'share-y'.
# For the 'share-y' option the y axis values is shared between this plot and the overlay plot.
# Otherwise, each plot use its own scale
#overlay_previous = yes

# If the bed file contains the exon
# structure (bed 12) then this is plotted. Otherwise
# a region **with direction** is plotted.
# If the bed file contains a column for color (column 9), then this color can be used by
# setting:
#color = bed_rgb
```

```
# if color is a valid colormap name (like RbBlGn), then the score (column 5) is mapped
# to the colormap.
# In this case, the the min_value and max_value for the score can be provided, otherwise
# the maximum score and minimum score found are used.
#color = RdYlBu
#min_value=0
#max_value=100
# If the color is simply a color name, then this color is used and the score is not considered.
color = darkblue
# whether printing the labels
labels = false
# optional:
# by default the labels are not printed if you have more than 60 features.
# to change it, just increase the value:
#max_labels = 60
# optional: font size can be given to override the default size
fontsize = 10
# optional: line_width
#line_width = 0.5
# the display parameter defines how the bed file is plotted.
# Default is 'stacked' where regions are plotted on different lines so
# we can see all regions and all labels.
# The other options are ['collapsed', 'interleaved', 'triangles']
# These options assume that the regions do not overlap.
# `collapsed`: The bed regions are plotted one after the other in one line.
# `interleaved`: The bed regions are plotted in two lines, first up, then down, then up etc.
# optional, default is black. To remove the border, simply set 'border_color' to none
# Not used in tssarrow style
#border_color = black
# style to plot the genes when the display is not triangles
#style = UCSC
#style = flybase
#style = tssarrow
# maximum number of gene rows to be plotted. This
# field is useful to limit large number of close genes
# to be printed over many rows. When several images want
# to be combined this must be set to get equal size
# otherwise, on each image the height of each gene changes
#gene_rows = 10
# by default the ymax is the number of
# rows occupied by the genes in the region plotted. However,
# by setting this option, the global maximum is used instead.
# This is useful to combine images that are all consistent and
# have the same number of rows.
#global_max_row = true
# If you want to plot all labels inside the plotting region:
#all_labels_inside = true
# If you want to display the name of the gene which goes over the plotted
# region in the right margin put:
#labels_in_margin = true
# if you use UCSC style, you can set the relative distance between 2 arrows on introns
# default is 2
#arrow_interval = 2
# if you use tssarrow style, you can choose the length of the arrow in bp
# (default is 4% of the plotted region)
#arrow_length = 5000
# if you use flybase or tssarrow style, you can choose the color of non-coding intervals:
#color_utr = grey
# as well as the proportion between their height and the one of coding
# (by default they are the same height):
#height_utr = 1
# By default, for oriented intervals in flybase style,
# or bed files with less than 12 columns, the arrowhead is added
# outside of the interval.
# If you want that the tip of the arrow correspond to
# the extremity of the interval use:
# arrowhead_included = true
# optional. If not given is guessed from the file ending.
file_type = bed

[file2]
file = file2.bw

# title of track (plotted on the right side)
title = file2
# height of track in cm (ignored if the track is overlay on top the previous track)
height = 2
# if you want to plot the track upside-down:
# orientation = inverted
# if you want to plot the track on top of the previous track. Options are 'yes' or 'share-y'.
# For the 'share-y' option the y axis values is shared between this plot and the overlay plot.
# Otherwise, each plot use its own scale
#overlay_previous = yes

color = #666666
# To use a different color for negative values
#negative_color = red
# To use transparency, you can use alpha
# default is 1
# alpha = 0.5
# the default for min_value and max_value is 'auto' which means that the scale will go
# roughly from the minimum value found in the region plotted to the maximum value found.
min_value = 0
#max_value = auto
# The number of bins takes the region to be plotted and divides it
# into the number of bins specified
# Then, at each bin the bigwig mean value is computed and plotted.
# A lower number of bins produces a coarser tracks
number_of_bins = 700
# to convert missing data (NaNs) into zeros. Otherwise, missing data is not plotted.
nans_to_zeros = true
# The possible summary methods are given by pyBigWig:
# mean/average/stdev/dev/max/min/cov/coverage/sum
# default is mean
summary_method = mean
# for type, the options are: line, points, fill. Default is fill
# to add the preferred line width or point size use:
# type = line:lw where lw (linewidth) is float
# similarly points:ms sets the point size (markersize (ms) to the given float
# type = line:0.5
# type = points:0.5
# set show_data_range to false to hide the text on the left showing the data range
show_data_range = true
# to compute operations on the fly on the file
# or between 2 bigwig files
# operation will be evaluated, it should contains file or
# file and second_file,
# we advice to use nans_to_zeros = true to avoid unexpected nan values
#operation = 0.89 * file
#operation = - file
#operation = file - second_file
#operation = log2((1 + file) / (1 + second_file))
#operation = max(file, second_file)
#second_file = path for the second file
# To log transform your data you can also use transform and log_pseudocount:
# For the transform values:
```

```
# 'log1p': transformed_values = log(1 + initial_values)
# 'log': transformed_values = log(log_pseudocount + initial_values)
# 'log2': transformed_values = log2(log_pseudocount + initial_values)
# 'log10': transformed_values = log10(log_pseudocount + initial_values)
# '-log': transformed_values = - log(log_pseudocount + initial_values)
# For example:
#tranform = log
#log_pseudocount = 2
# When a transformation is applied, by default the y axis
# gives the transformed values, if you prefer to see
# the original values:
#y_axis_values = original
# If you want to have a grid on the y-axis
#grid = true
file_type = bigwig
```

2 Galaxy wrapper

A history where you can see an example of pyGenomeTracks inputs and outputs is available at https://usegalaxy.eu/u/ldelisle/h/last-example-of-pgt.



pyGenomeTracks plot genomic data tracks (Galaxy Version 3.3)

Region of the genome to limit the operation

X:2500000-3500000

The format is chr:start-end. Also valid is just to specify a chromosome, for example chr10:10-500 (--region)

Configure figure parameters

Include tracks in your plot

1: Include tracks in your plot

Choose style of the track

TAD visualization

Plot title

depth = 200000; transform = log1p; min_value = 5

Matricies to plot

1: Li_et_al_2015.h5

Color map to use for the heatmap

RdYlBu reversed

Available color map names can be found here: https://matplotlib.org/examples/color/colormaps_reference.html

Depth

200000

Minimum value

5.0

Maximum value

Plot the transformed value

log1p

height

Show masked bins

Yes [✓ No]

Boundaries file

4: links2.links (as bed)
2: tad_classification.bed

Scale factor

1.0

if the values in the matrix need to be scaled the following parameter can be used

Rasterize the heatmap

[✓ Yes] No

Invert the track

Yes [✓ No]

Include spacer at the end of the track

Height of the spacer.

2: Include tracks in your plot

Choose style of the track

TAD visualization

Plot title

depth = 250000; orientation = inverted; colormap = PuRd; min_value = 5; max_value = 70

Matricies to plot

1: Li_et_al_2015.h5

History

last example of PGT

5 shown

14.44 MB

5: pyGenomeTracks on [data 1, data 3, and others]: Plot

476.1 KB

format: **png**, génome de référence: **?**

```
[hic_section_0_0]
file = 0_0.h5
file_type = hic_matrix
title = depth = 200000X transform =
log1pX min_value = 5
depth = 200000
colormap = RdYlBu_r
min_value = 5.0
transform = log1p
scale_factor = 1.0
rasterize = true
[hic_section_1_0]
file = 1_0.h5
file_
```

Image in png format

4: links2.links

3: bigwig2_X_2.5e6_3.5e6.bw

2: tad_classification.bed

1: Li_et_al_2015.h5

Figure 1: pyGenomeTracks wrapper for Galaxy.

3 Track file for the main figure

All used data is provided on zenodo: https://doi.org/10.5281/zenodo.3775381.

```
[x-axis]
fontsize = 20
title = dm3
```

```
[spacer]
height = 0.3

[HiC_Li_cubenas_et_al]
file = HiC_Cubenas.h5
height = 3
title = Hi-C matrix with TAD domains as bed file and bigwig
depth = 50000
transform = log1p
show_masked_bins = false
file_type = hic_matrix

[tad_classification]
file = tad__domains.bed
overlay_previous = share-y
color = none
height = 4
labels = false
fontsize = 10
file_type = domains

[CP190]
file = CP190.bw
overlay_previous = yes
show_data_range = false
height = 2
color = #FF007F
min_value = 0
number_of_bins = 700
nans_to_zeros = true
summary_method = mean
show_data_range = false
file_type = bigwig


[CP190_2]
file = CP190.bw
overlay_previous = yes
show_data_range = false
height = 2
color = #000000
min_value = 0
number_of_bins = 700
nans_to_zeros = true
summary_method = mean
type = line:0.75
show_data_range = false
file_type = bigwig


[chromatinStates_kc]
file = chromatinStates_kc.bed
title = chromatin states
height = 1
color = bed_rgb
display = collapsed
height = 0.5
labels = false
fontsize = 10
file_type = bed
show_data_range = false
```

```
[spacer]
height = 0.5

[tad_score]
file = tad__tad_score.bm
title = bedgraph matrix
color = none
height = 2
labels = false
fontsize = 10
type = lines
file_type = bedgraph_matrix

[spacer]
height = 0.5

[H3K36me3]
file = H3K36me3.bw
title = bigwig with threshold line
height = 2
color = #18B463
min_value = 0
number_of_bins = 700
nans_to_zeros = true
summary_method = mean
show_data_range = true
file_type = bigwig


[hlines]
file_type = hlines
y_values = 1.5
line_style = dashed
line_width = 1
overlay_previous = share-y
show_data_range = False

[scalebar]
file_type = scalebar
x_center = 8120730
size = 36000
where = bottom

[spacer]
height = 0.5

[vlines]
file = tad__domains.bed
type = vlines

[test arcs]
file = test.arcs
title = arcs
orientation = inverted
line_style = solid
height = 2

[genes]
file = dm3_genes_compact_no_cg.bed
height = 1
title = bed file
fontsize = 10
```

```
file_type = bed
gene_rows = 2
line_width = 0.5
color = red
```

4 Figure of the graphical abstract

The figure in the graphical abstract is more exhaustive than in the manuscript.

The first track from the top shows the genomic locus (chromosome 2L 8.05 Mb to 8.31 Mb). The second track illustrates a Hi-C matrix track (Li *et al.* (2015)) overlaid by its detected TADs, via HiCExplorer, and a coverage profile of CP190 ChIP. Although Hi-C tracks can be provided as cool Abdennur and Mirny (2019) or HiCExplorer's native h5 format (Ramírez *et al.* (2018)), here a matrix of h5 format has been used. TADs are given as a bed file which is a direct output of HiCExplorer's hicFindTADs, the ChIP-Seq profile is provided as a bigwig file (both Kent *et al.* (2010)). This track is followed by an inverted Hi-C matrix in h5 format (Cubeñas-Potts *et al.* (2017)). The interaction patterns in different conditions can be compared using this method. The succeeding track shows the chromatin states, provided as a bed file where the colors used are as defined in the 9th field of the bed file. The next track visualizes the TAD separation scores, the data is presented in a bedgraph matrix file format from HiCExplorer hicFindTADs. The green track shows a filled-out curve representation of the data from H3K36me3 histone mark, a mark which is correlated with the active chromatin state in *Drosophila melanogaster*, provided as a bigwig file. The following track shows another bigwig file as an orange line. The file contains the RNA polymerase II profile and the track has been plotted with an additional horizontal threshold line as well as a scale bar indicating the distance between two different peaks of interest. The blue arcs show artificially created links that could be contacts between different CP190 peaks. Finally the last track is a gene track of dm3. Although both gtf and bed formats (Karolchik *et al.* (2004)) are accepted by PGT, here a bed was used.

All used data is provided on zenodo: https://doi.org/10.5281/zenodo.3775381.

```
[x-axis]
fontsize = 20
title = dm3

[spacer]
height = 0.3

[HiC_Li_cubenas_et_al]
file = HiC_Cubenas.h5
height = 3
title = Hi-C matrix with TAD domains as bed file and bigwig
depth = 50000
transform = log1p
show_masked_bins = false
file_type = hic_matrix

[tad_classification]
file = tad__domains.bed
overlay_previous = yes
color = none
height = 4
labels = false
fontsize = 10
file_type = domains

[CP190]
file = CP190.bw
overlay_previous = yes
show_data_range = false
height = 2
color = #FF007F
min_value = 0
```

```
number_of_bins = 700
nans_to_zeros = true
summary_method = mean
show_data_range = false
file_type = bigwig


[CP190_2]
file = CP190.bw
overlay_previous = yes
show_data_range = false
height = 2
color = #000000
min_value = 0
number_of_bins = 700
nans_to_zeros = true
summary_method = mean
type = line:0.75
show_data_range = false
file_type = bigwig


[spacer]
height = 0.1

[HiC_cubenas_et_al]
file = HiC_Li_et_al.h5
title = inverted Hi-C matrix
depth = 50000
transform = log1p
show_masked_bins = false
file_type = hic_matrix
orientation = inverted
height = 3

[chromatinStates_kc]
file = chromatinStates_kc.bed
title = chromatin states
height = 1
color = bed_rgb
display = collapsed
height = 0.5
labels = false
fontsize = 10
file_type = bed
show_data_range = false


[spacer]
height = 0.5

[tad_score]
file = tad__tad_score.bm
title = bedgraph matrix
color = none
height = 2
labels = false
fontsize = 10
type = lines
file_type = bedgraph_matrix

[spacer]
height = 0.5
```

```
[H3K36me3]
file = H3K36me3.bw
title = bigwig
height = 2
color = #18B463
min_value = 0
number_of_bins = 700
nans_to_zeros = true
summary_method = mean
show_data_range = true
file_type = bigwig


[bigwig]
file = RNAPII.bw
title = bigwig with threshold and scalebar
type = line
color = orange
height = 3

[hlines]
file_type = hlines
y_values = 20
line_style = dashed
line_width = 1
overlay_previous = share-y

[scalebar]
file_type = scalebar
x_center = 8125730
size = 74770
where = bottom

[spacer]
height = 0.5

[vlines]
file = tad__domains.bed
type = vlines

[test arcs]
file = test.arcs
title = arcs
orientation = inverted
line_style = solid
height = 2

[genes]
file = dm3_genes_compact_no_cg.bed
height = 1
title = bed file
fontsize = 10
file_type = bed
gene_rows = 2
line_width = 0.5
color = red
```

5 Parameter supported for each track

Here is a table summarizing all the parameters supported for each track in version 3.5.

parameter	x_axis	epilogos	links	domains	bed	gtf	narrow_peak	bigwig	bedgraph	bedgraph_matrix	hlines	hic_matrix	scalebar
overlay_previous	X	X	X	X	X	X	X	X	X	X	X	X	X
where	X												X
fontsize	X				X	X							X
categories_file		X											
orientation		X	X	X	X	X	X	X	X	X	X	X	
links_type			X										
line_width			X	X	X	X	X				X		X
line_style			X								X		
color			X	X	X	X	X	X	X		X		X
alpha			X					X	X		X		X
max_value			X	X	X		X	X	X	X	X	X	
min_value			X	X	X			X	X	X	X	X	
ylim			X										
compact_arcs_level			X										
use_middle			X						X				
border_color				X	X	X							
prefered_name				X	X	X							
merge_transcripts				X	X	X							
labels					X	X							
style					X	X							
display					X	X							
max_labels					X	X							
global_max_row					X	X							
gene_rows					X	X							
arrow_interval					X	X							
arrowhead_included					X	X							
color_utr					X	X							
height_utr					X	X							
arrow_length					X	X							
all_labels_inside					X	X							
labels_in_margin					X	X							
show_data_range							X	X	X	X	X		
show_labels							X						
use_summit							X						
width_adjust							X						
type							X	X	X	X			
negative_color								X	X				
nans_to_zeros								X	X				
summary_method								X	X				
number_of_bins								X	X				
transform								X	X			X	
log_pseudocount								X	X				
y_axis_values								X	X				
second_file								X	X				
operation								X	X				
grid								X	X				
rasterize									X	X		X	
pos_score_in_bin										X			
plot_horizontal_lines										X			
colormap										X		X	
depth												X	
show_masked_bins												X	
scale_factor												X	
x_center													X
size													X

6 pyGenomeTracks examples

All used data is provided in our github repository:
https://github.com/deeptools/pyGenomeTracks/tree/master/examples and
https://github.com/deeptools/pyGenomeTracks/tree/master/pygenometracks/tests/test_data.

6.1 Basic examples

6.1.1 A bigwig track

```
[bigwig file test]
file = bigwig.bw
# height of the track in cm (optional value)
height = 4
title = bigwig
min_value = 0
max_value = 30
```

```
$ pyGenomeTracks --tracks bigwig_track.ini --region X:2,500,000-3,000,000 -o bigwig.png
```

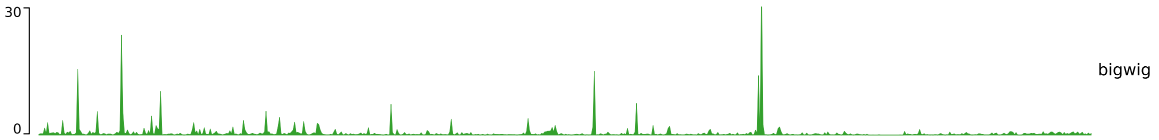


Figure 2: Bigwig track

6.1.2 Bigwig and genes

```
[bigwig file test]
file = bigwig.bw
# height of the track in cm (optional value)
height = 4
title = bigwig
min_value = 0
max_value = 30

[spacer]
# this simply adds an small space between the two tracks.

[genes]
file = genes.bed.gz
height = 7
title = genes
fontsize = 10
file_type = bed
gene_rows = 10

[x-axis]
fontsize=10
```

```
$ pyGenomeTracks --tracks bigwig_with_genes.ini --region X:2,800,000-3,100,000 -o bigwig_with_genes.eps
```

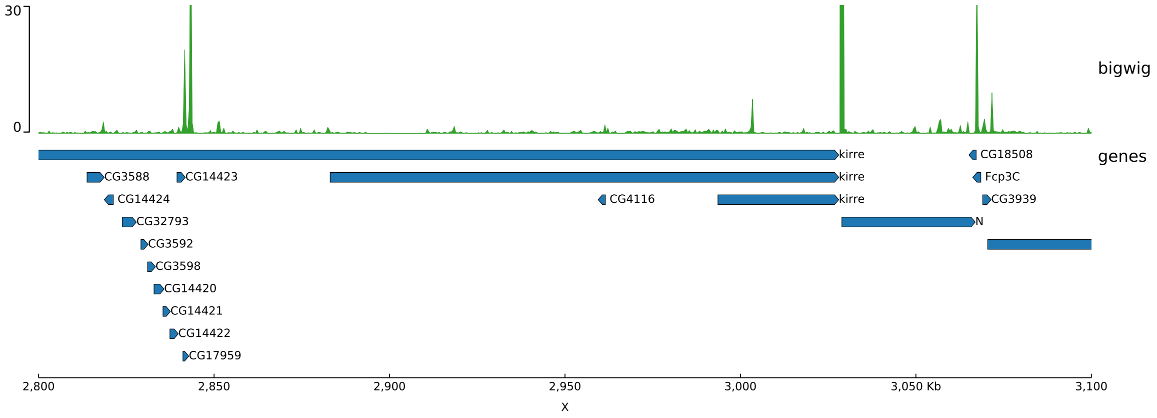


Figure 3: Bigwig and gene track

6.1.3 Bigwig, genes and vlines track

```
[bigwig file test]
file = bigwig.bw
# height of the track in cm (optional value)
height = 4
title = bigwig
min_value = 0
max_value = 30

[spacer]
# this simply adds an small space between the two tracks.

[genes]
file = genes.bed.gz
height = 7
title = genes
fontsize = 10
file_type = bed
gene_rows = 10

[x-axis]
fontsize=10

[vlines]
file = domains.bed
type = vlines
```

```
$ pyGenomeTracks --tracks bigwig_with_genes_and_vlines.ini --region X:2,800,000-3,100,000 -o
↪  bigwig_with_genes_and_vlines.eps
```

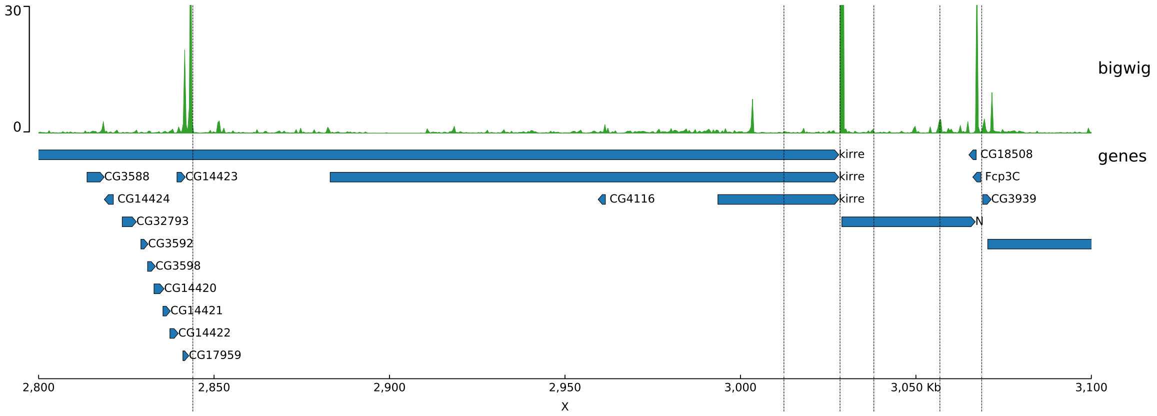


Figure 4: Bigwig, genes and vlines track

6.1.4 Bigwig overlay with transparency

```
[test bigwig]
file = bigwig2_X_2.5e6_3.5e6.bw
color = blue
height = 7
title = No alpha:
        (bigwig color=blue 2000 bins) overlaid with (bigwig color = (0.6, 0, 0) max over 300 bins) overlaid
↪  with (bigwig mean color = green 200 bins)
number_of_bins = 2000
min_value = 0
max_value = 30

[test bigwig max]
file = bigwig2_X_2.5e6_3.5e6.bw
color = (0.6, 0, 0)
summary_method = max
number_of_bins = 300
overlay_previous = share-y

[test bigwig mean]
file = bigwig2_X_2.5e6_3.5e6.bw
color = green
type = fill
number_of_bins = 200
overlay_previous = share-y

[spacer]

[test bigwig]
file = bigwig2_X_2.5e6_3.5e6.bw
color = blue
height = 7
title = alpha
        (bigwig color = blue 2000 bins) overlaid with (bigwig color = (0.6, 0, 0) alpha = 0.5 max over 300
↪  bins) overlaid with (bigwig mean color = green alpha = 0.5 200 bins)
number_of_bins = 2000
min_value = 0
max_value = 30

[test bigwig max]
file = bigwig2_X_2.5e6_3.5e6.bw
color = (0.6, 0, 0)
alpha = 0.5
summary_method = max
```

```
number_of_bins = 300
overlay_previous = share-y

[test bigwig mean]
file = bigwig2_X_2.5e6_3.5e6.bw
color = green
alpha = 0.5
type = fill
number_of_bins = 200
overlay_previous = share-y

[spacer]

[test bigwig]
file = bigwig2_X_2.5e6_3.5e6.bw
height = 7
title = alpha for lines/points:
        (bigwig color=(0.6, 0, 0) alpha = 0.5 max) overlaid with (bigwig mean color = green alpha = 0.5 line:2)
↪  overlaid with (bigwig min color = blue alpha = 0.5 points:2)
color = (0.6, 0, 0)
alpha = 0.5
summary_method = max
number_of_bins = 300
min_value = 0
max_value = 30

[test bigwig mean]
file = bigwig2_X_2.5e6_3.5e6.bw
color = green
type = line:2
alpha = 0.5
summary_method = mean
number_of_bins = 300
overlay_previous = share-y

[test bigwig min]
file = bigwig2_X_2.5e6_3.5e6.bw
color = blue
summary_method = min
number_of_bins = 1000
type = points:3
alpha = 0.5
overlay_previous = share-y

[x-axis]
```

```
$ pyGenomeTracks --tracks alpha.ini --region X:2700000-3100000 --trackLabelFraction 0.2 --dpi 130 -o
↪  master_alpha.png
```

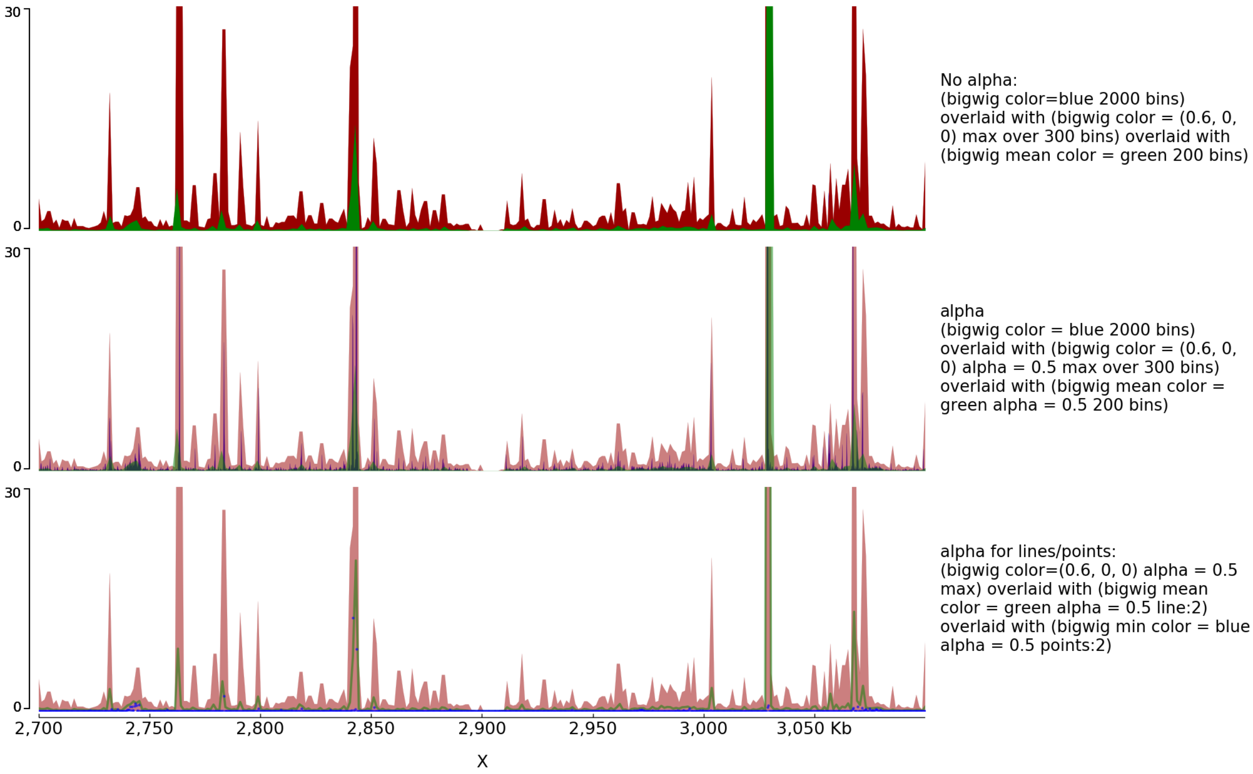


Figure 5: Bigwig overlay with transparency

6.2 Examples with bed and gtf

6.2.1 Bed and gtf format tracks

```
[x-axis]
where = top
title = where =top

[spacer]
height = 0.05

[genes 2]
file = dm3_genes.bed.gz
height = 7
title = genes (bed12) style = UCSC; fontsize = 10
style = UCSC
fontsize = 10

[genes 2bis]
file = dm3_genes.bed.gz
height = 7
title = genes (bed12) style = UCSC; arrow_interval=10; fontsize = 10
style = UCSC
arrow_interval = 10
fontsize = 10

[spacer]
height = 1

[test bed6]
file = dm3_genes.bed6.gz
height = 7
title = bed6 border_color = black; gene_rows=10; fontsize=7; color=Reds
```

```
        (when a color map is used for the color (e.g. coolwarm, Reds) the bed
        score column mapped to a color)
fontsize = 7
file_type = bed
color = Reds
border_color = black
gene_rows = 10

[spacer]
height = 1

[test bed4]
file = dm3_genes.bed4.gz
height = 10
title = bed4 fontsize = 10; line_width = 1.5; global_max_row = true
        (global_max_row sets the number of genes per row as the maximum found
        anywhere in the genome, hence the white space at the bottom)
fontsize = 10
file_type = bed
global_max_row = true
line_width = 1.5

[spacer]
height = 1

[test gtf]
file = dm3_subset_BDGP5.78.gtf.gz
height = 10
title = gtf from ensembl
fontsize = 12
file_type = bed

[spacer]
height = 1

[test bed]
file = dm3_subset_BDGP5.78_asbed_sorted.bed.gz
height = 10
title = gtf from ensembl in bed12
fontsize = 12
file_type = bed

[spacer]
height = 1

[test gtf collapsed]
file = dm3_subset_BDGP5.78.gtf.gz
height = 10
title = gtf from ensembl one entry per gene
merge_transcripts = true
prefered_name = gene_name
fontsize = 12
file_type = bed

[spacer]
height = 1

[x-axis]
fontsize = 30
title = fontsize = 30
```

```
$ pyGenomeTracks --tracks bed_and_gtf_tracks.ini --region X:3000000-3300000 --trackLabelFraction 0.2 --width 38
↪  --dpi 130 -o master_bed_and_gtf.eps
```

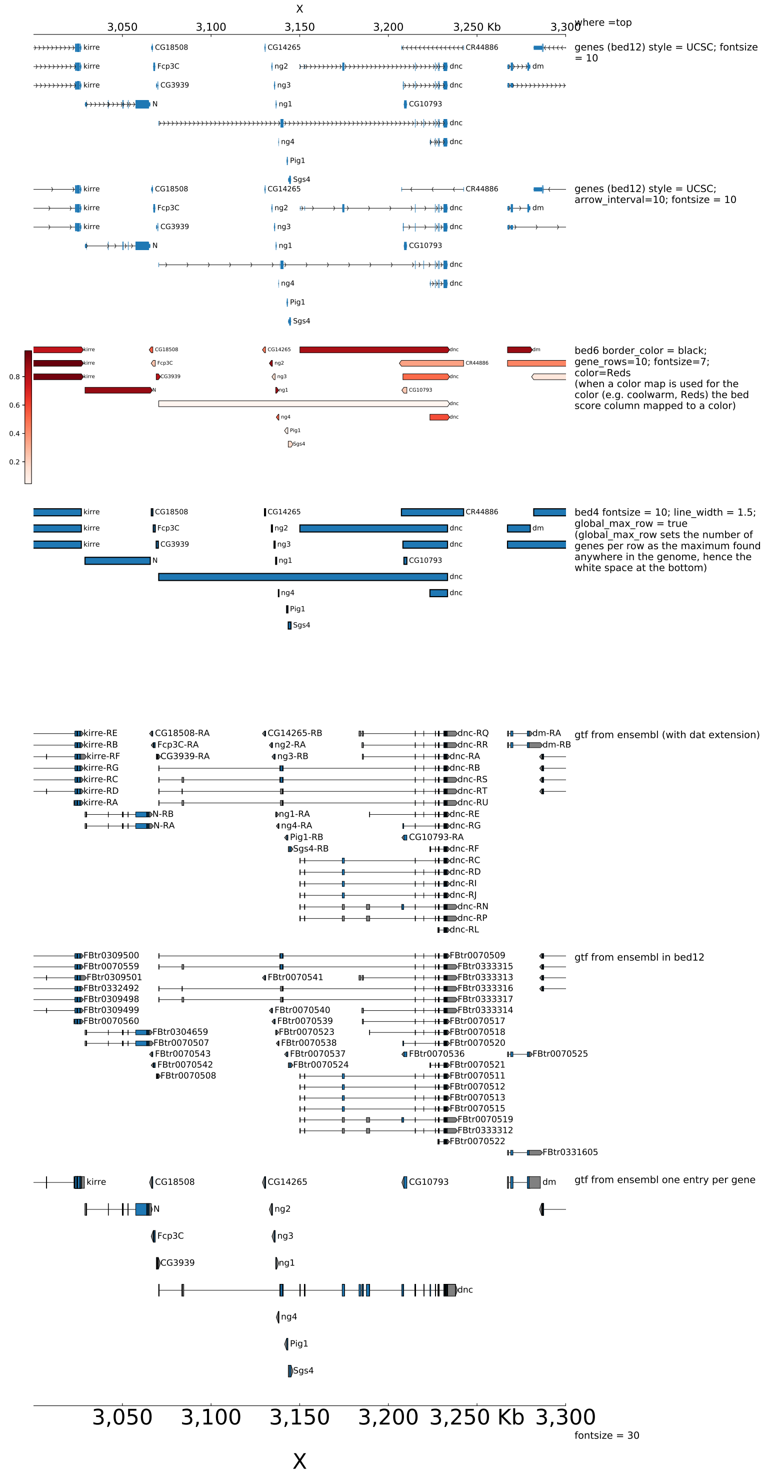


Figure 6: Bed and gtf format tracks

6.2.2 UTR settings

```
[x-axis]
where = top

[spacer]
height = 0.05

[genes 0]
file = dm3_genes.bed.gz
height = 7
title = genes (bed12) style = flybase; fontsize = 10
style = flybase
fontsize = 10

[spacer]
height = 1

[genes 1]
file = dm3_genes.bed.gz
height = 7
title = genes (bed12) style = flybase; fontsize = 10; color_utr = red
style = flybase
fontsize = 10
color_utr = red

[spacer]
height = 1

[genes 2]
file = dm3_genes.bed.gz
height = 7
title = genes (bed12) style = flybase; fontsize = 10; height_utr = 0.7
style = flybase
fontsize = 10
height_utr = 0.7
```

```
$ pyGenomeTracks --tracks bed_flybase_tracks.ini --region X:3000000-3300000 --trackLabelFraction 0.2 --width 38
↪  --dpi 130 -o master_bed_flybase.png
```

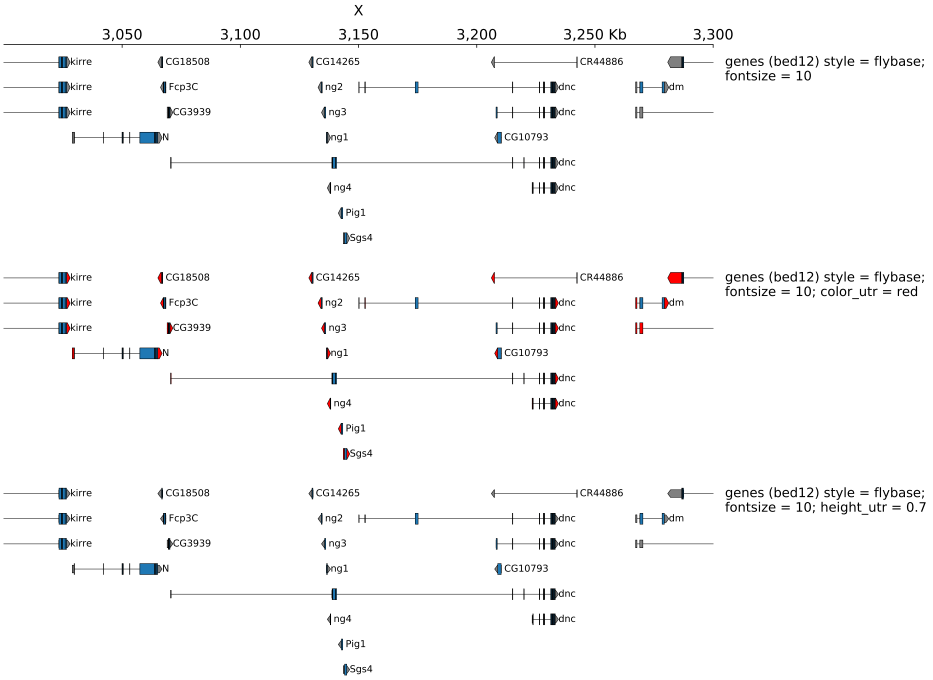


Figure 7: UTR

6.3 4C tracks

```
[x-axis]
where = top

[spacer]
height = 0.05

[test bedgraph]
file = GSM3182416_E12DHL_WT_Hoxd11vp.bedgraph.gz
color = blue
height = 5
title = bedgraph rasterize = true
rasterize = true
max_value = 10

[test bedgraph]
file = GSM3182416_E12DHL_WT_Hoxd11vp.bedgraph.gz
color = blue
height = 5
title = bedgraph
max_value = 10

[test bedgraph use middle]
file = GSM3182416_E12DHL_WT_Hoxd11vp.bedgraph.gz
color = blue
height = 5
title = bedgraph with use_middle = true
max_value = 10
use_middle = true

[genes]
file = HoxD_cluster_regulatory_regions_mm10.bed
height = 3
title = HoxD genes and regulatory regions
```

```
$ pyGenomeTracks --tracks bedgraph_useMid.ini --region chr2:74,000,000-74,800,000 --trackLabelFraction 0.2
↪  --width 38 --dpi 130  -o master_bedgraph_useMid_zoom.png
```

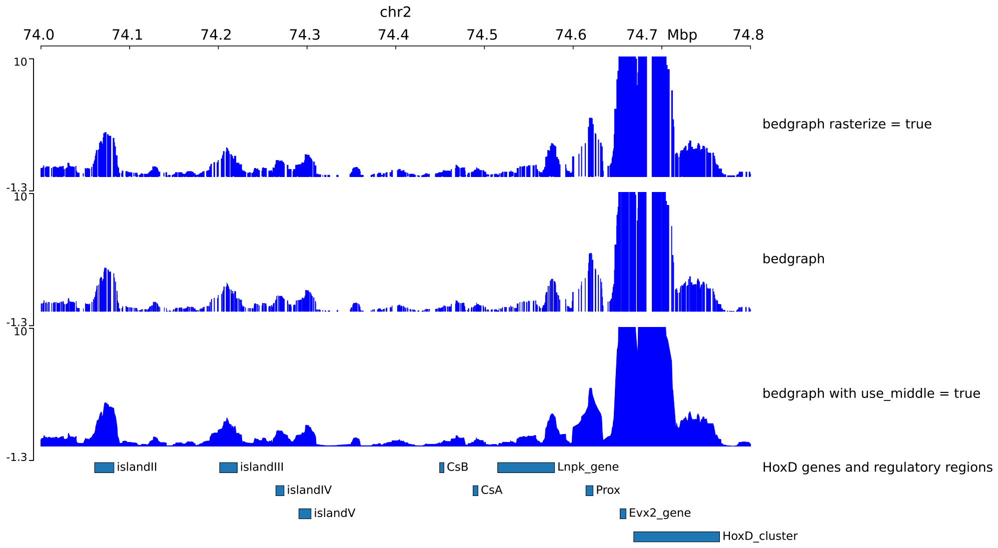


Figure 8: 4C track

6.4 Peaks

```
[narrow]
file = test2.narrowPeak
height = 4
max_value = 40
line_width = 0.1
title = max_value = 40;line_width = 0.1

[narrow 2]
file = test2.narrowPeak
height = 2
show_labels = false
show_data_range =  false
color = #00FF0080
use_summit = false
title = show_labels = false; show_data_range = false; use_summit = false; color = #00FF0080

[spacer]

[narrow 3]
file = test2.narrowPeak
height = 2
show_labels = false
color = #0000FF80
use_summit = false
width_adjust = 4
title = show_labels = false; use_summit = false; width_adjust = 4

[spacer]

[narrow 4]
file = test2.narrowPeak
height = 3
type = box
```

```
color = blue
line_width = 2
title = type = box; color = blue; line_width = 2

[spacer]

[narrow 5]
file = test2.narrowPeak
height = 3
type = box
color = blue
use_summit = false
title = type = box; color = blue; use_summit = false

[x-axis]
```

```
$ pyGenomeTracks --tracks narrow_peak2.ini --region X:2760000-2802000 --trackLabelFraction 0.2 --dpi 130 -o
↪  master_narrowPeak2.png
```

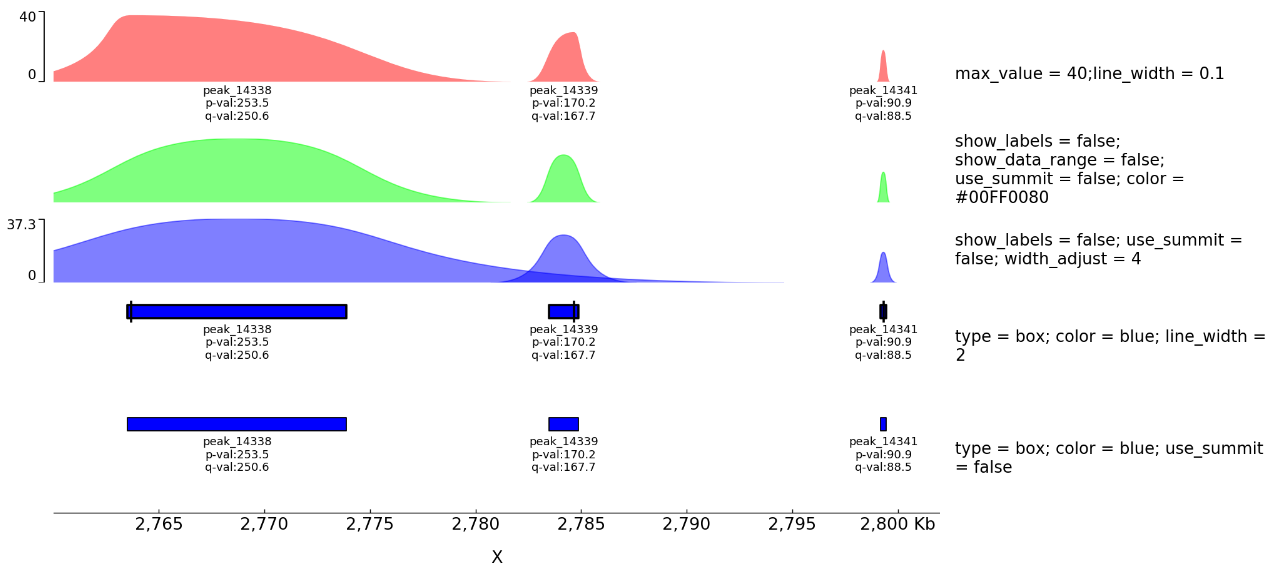


Figure 9: Peak track

6.5 Horizontal lines

```
[test hlines]
color = red
line_width = 2
line_style = dashed
y_values = 10, 200
min_value = 0
show_data_range = true
height = 5
title = hlines: color = red; line_width = 2; line_style = dashed; y_values = 10, 200
file_type = hlines

[spacer]

[test bigwig fill]
file = bigwig2_X_2.5e6_3.5e6.bw
```

```
color = gray
height = 2
type = fill
title = bigwig: gray fill overlayed with hlines at 10 and 200 blue dotted
max_value = 50

[test hlines ovelayed]
color = blue
line_style = dotted
y_values = 10, 200
overlay_previous = share-y
file_type = hlines

[spacer]

[x-axis]
```

```
$ pyGenomeTracks --tracks hlines.ini --region X:2700000-3100000 --trackLabelFraction 0.2 --dpi 130 -o
↪  master_hlines.png
```

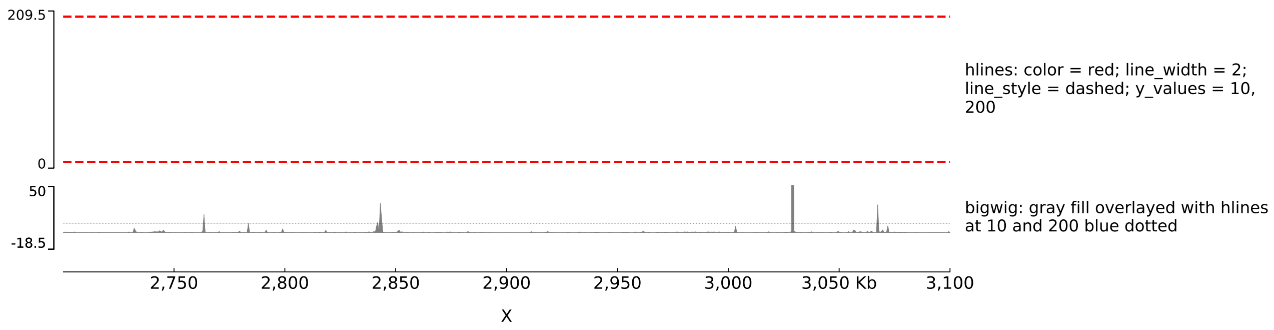


Figure 10: Horizontal lines track

6.6 Epilogos

```
[epilogos]
file = epilog.qcat.bgz
height = 5
title = height=5; categories_file=epilog_cats.json

[x-axis]
```

```
$ pyGenomeTracks  --tracks epilogos_track.ini --region X:3100000-3150000 -o epilogos_track.png
```

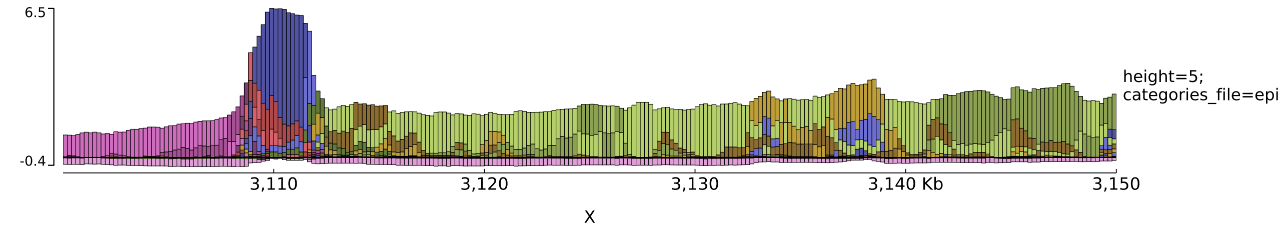


Figure 11: Epilogos track

6.6.1 Color setting

The color of the bars can be set by using a json file.

```
{
"categories":{
          "1":["Active TSS","#ff0000"],
          "2":["Flanking Active TSS","#ff4500"],
          "3":["Transcr at gene 5\" and 3\"","#32cd32"],
          "4":["Strong transcription","#008000"],
          "5":["Weak transcription","#006400"],
          "6":["Genic enhancers","#c2e105"],
          "7":["Enhancers","#ffff00"],
          "8":["ZNF genes & repeats","#66cdaa"],
          "9":["Heterochromatin","#8a91d0"],
          "10":["Bivalent/Poised TSS","#cd5c5c"],
          "11":["Flanking Bivalent TSS/Enh","#e9967a"],
          "12":["Bivalent Enhancer","#bdb76b"],
          "13":["Repressed PolyComb","#808080"],
          "14":["Weak Repressed PolyComb","#c0c0c0"],
          "15":["Quiescent/Low","#ffffff"]
        }
}
```

```
[epilogos]
file = epilog.qcat.bgz
height = 5
title = epilogos with custom colors
categories_file = epilog_cats.json

[epilogos inverted]
file = epilog.qcat.bgz
height = 5
title = epilogos inverted
orientation = inverted

[x-axis]
```

```
$ pyGenomeTracks  --tracks epilogos_track2.ini --region X:3100000-3150000 -o epilogos_track2.png
```

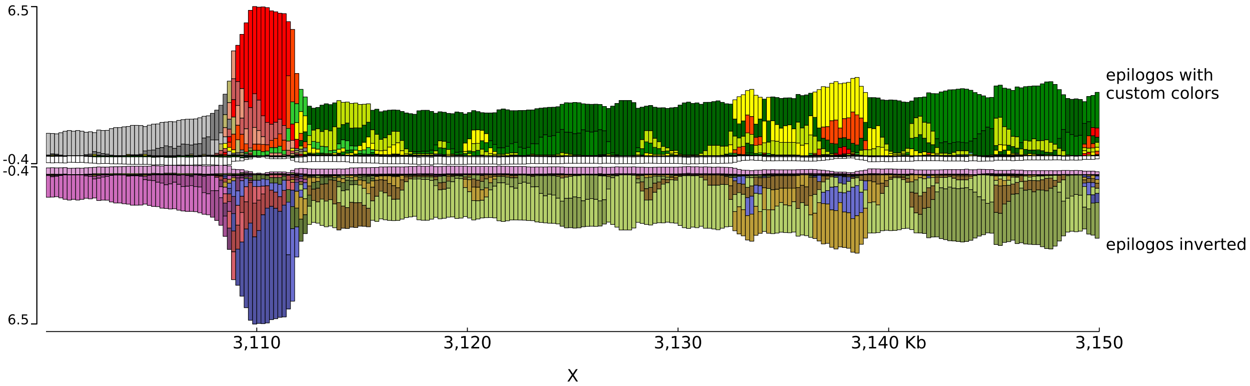


Figure 12: Epilogos track with color setting

6.7 Multiple combined tracks

```
[x-axis]
where = top
title = where=top

[spacer]
height = 0.05

[tads]
file = tad_classification.bed
title = TADs color = bed_rgb; border_color = black
file_type = domains
border_color = black
color = bed_rgb
height = 5

[tads 2]
file = tad_classification.bed
title = TADs orientation = inverted; color = #cccccc; border_color = red
file_type = domains
border_color = red
color = #cccccc
orientation = inverted
height = 3

[spacer]
height = 0.5

[tad state]
file = chromatinStates_kc.bed.gz
height = 1.2
title = bed display = interleaved; labels = false
display = interleaved
labels = false

[spacer]
height = 0.5

[tad state]
file = chromatinStates_kc.bed.gz
height = 0.5
title = bed display = collapsed; color = bed_rgb
labels = false
color = bed_rgb
display = collapsed

[spacer]
height = 0.5

[test bedgraph]
file = bedgraph_chrx_2e6_5e6.bg
color = blue
height = 1.5
title = bedgraph color = blue
max_value = 100

[test arcs]
file = test.arcs
title = links orientation = inverted
orientation = inverted
line_style = dashed
height = 2
```

```
[test bigwig]
file = bigwig2_X_2.5e6_3.5e6.bw
color = blue
height = 1.5
title = bigwig number_of_bins = 2000
number_of_bins = 2000

[spacer]

[test bigwig overlay]
file = bigwig2_X_2.5e6_3.5e6.bw
color = red
title = color:red; max_value = 50; number_of_bins = 100 (next track: overlay_previous = yes;
        max_value = 50; show_data_range = false; color = #0000FF80 (blue, with alpha 0.5))
min_value = 0
max_value = 50
height = 2
number_of_bins = 100

[test bigwig overlay]
file = bigwig_chrx_2e6_5e6.bw
color = #0000FF80
title =
min_value = 0
max_value = 50
show_data_range = false
overlay_previous = yes
number_of_bins = 100

[spacer]
height = 1

[tads 3]
file = tad_classification.bed
title = TADs color = #cccccc; border_color = red (next track:
        overlay_previous = share-y links_type = loops)
file_type = domains
border_color = red
color = #cccccc
height = 3

[test arcs overlay]
file = test.arcs
color = red
line_width = 10
links_type = loops
overlay_previous = share-y

[test arcs]
file = test.arcs
line_width = 3
color = RdYlGn
title = links line_width = 3 color RdYlGn
height = 3

[spacer]
height = 0.5
title = height = 0.5

[genes 2]
file = dm3_genes.bed.gz
height = 7
```

```
title = genes (bed12) style = flybase;fontsize = 10
style = flybase
fontsize = 10

[spacer]
height = 1

[test gene rows]
file = dm3_genes.bed.gz
height = 3
title = gene_rows = 3 (maximum 3 rows); style = UCSC
fontsize = 8
style = UCSC
gene_rows = 3

[spacer]
height = 1

[test bed6]
file = dm3_genes.bed6.gz
height = 7
title = bed6 border_color = black; gene_rows = 10; fontsize = 7; color = Reds
        (when a color map is used for the color (e.g. coolwarm, Reds) the bed
        score column mapped to a color)
fontsize = 7
file_type = bed
color = Reds
border_color = black
gene_rows = 10

[test bed6]
file = dm3_genes.bed6.gz
height = 10
title = bed6 fontsize = 10; line_width = 1.5; global_max_row = true
        (global_max_row sets the number of genes per row as the maximum found
        anywhere in the genome, hence the white space at the bottom)
fontsize = 10
file_type = bed
global_max_row = true
line_width = 1.5

[x-axis]
fontsize = 30
title = fontsize = 30

[vlines]
file = tad_classification.bed
type = vlines
```

```
$ pyGenomeTracks  --tracks browser_tracks.ini --region X:3000000-3500000 --trackLabelFraction 0.2 --width 38
↪  --dpi 130  -o master_plot.png
```

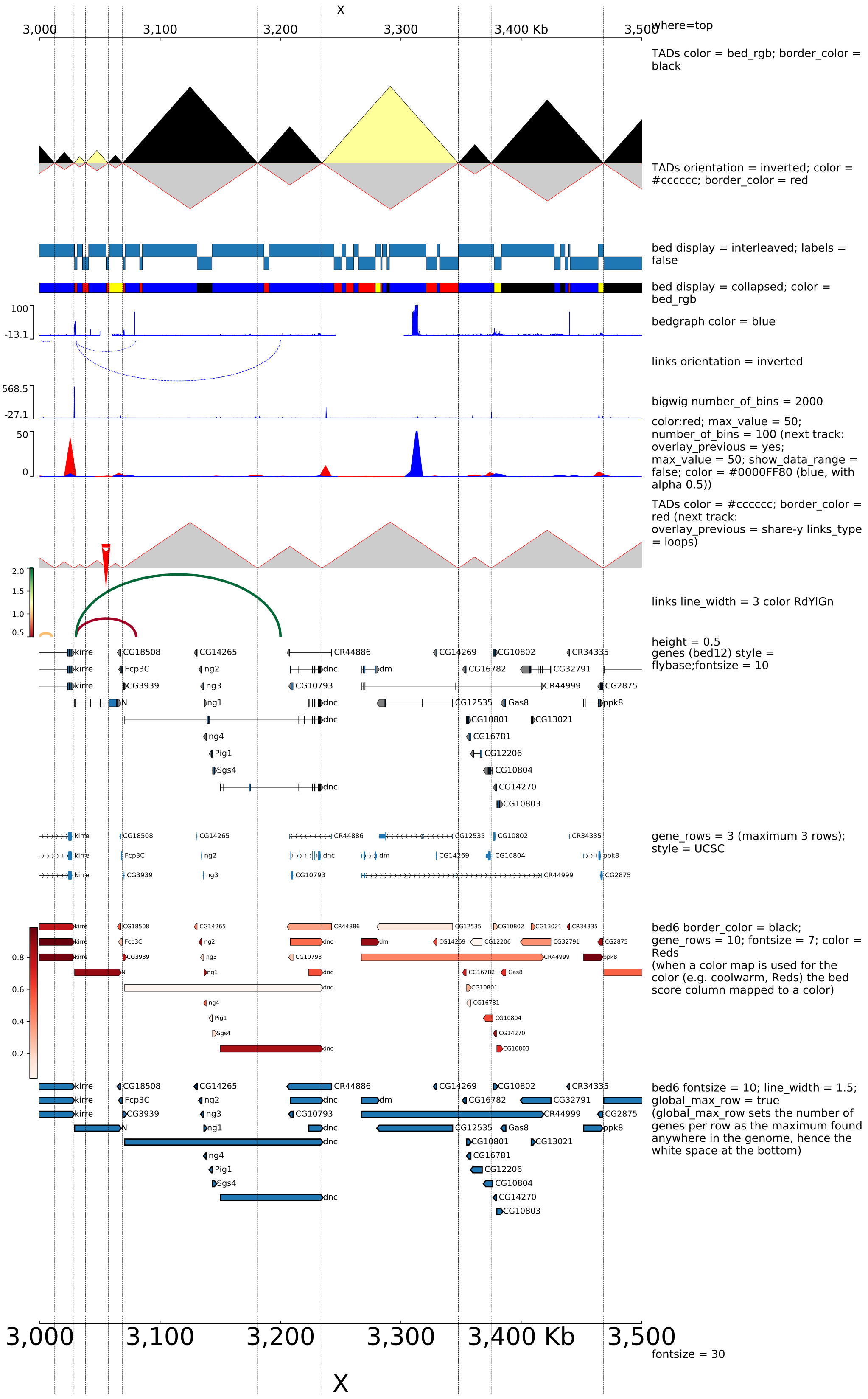


Figure 13: Multiple combined tracks

6.7.1 Multiple tracks with bigwigs

```
[test bigwig lines]
file = bigwig2_X_2.5e6_3.5e6.bw
color = gray
height = 2
type = line
title = orientation = inverted; show_data_range = false
orientation = inverted
show_data_range = false
max_value = 50

[test bigwig lines:0.2]
file = bigwig_chrx_2e6_5e6.bw
color = red
height = 2
type = line:0.2
title = type = line:0.2

[spacer]

[test bigwig points]
file = bigwig_chrx_2e6_5e6.bw
color = black
height = 2
min_value = 0
max_value = 100
type = points:0.5
title = type = point:0.5; min_value = 0; max_value = 100

[spacer]

[test bigwig nans to zeros]
file = bigwig_chrx_2e6_5e6.bw
color = red
height = 2
nans_to_zeros = true
title = nans_to_zeros = true

[spacer]

[test bigwig mean]
file = bigwig2_X_2.5e6_3.5e6.bw
color = gray
height = 5
title = gray:summary_method = mean; blue:summary_method = max;
        red:summary_method = min
type = line
summary_method = mean
max_value = 150
min_value = -5
show_data_range = false
number_of_bins = 300

[test bigwig max]
file = bigwig2_X_2.5e6_3.5e6.bw
#title = test
color = blue
type = line
summary_method = max
max_value = 150
min_value = -15
show_data_range = false
```

```
overlay_previous = share-y
number_of_bins = 300

[test bigwig min]
file = bigwig2_X_2.5e6_3.5e6.bw
color = red
type = line
summary_method = min
max_value = 150
min_value = -25
overlay_previous = share-y
number_of_bins = 300

[spacer]

[x-axis]
```

```
$ pyGenomeTracks  --tracks bigwig.ini --region X:2700000-3100000 --trackLabelFraction 0.2 --dpi 130 -o
↪  master_bigwig.png
```

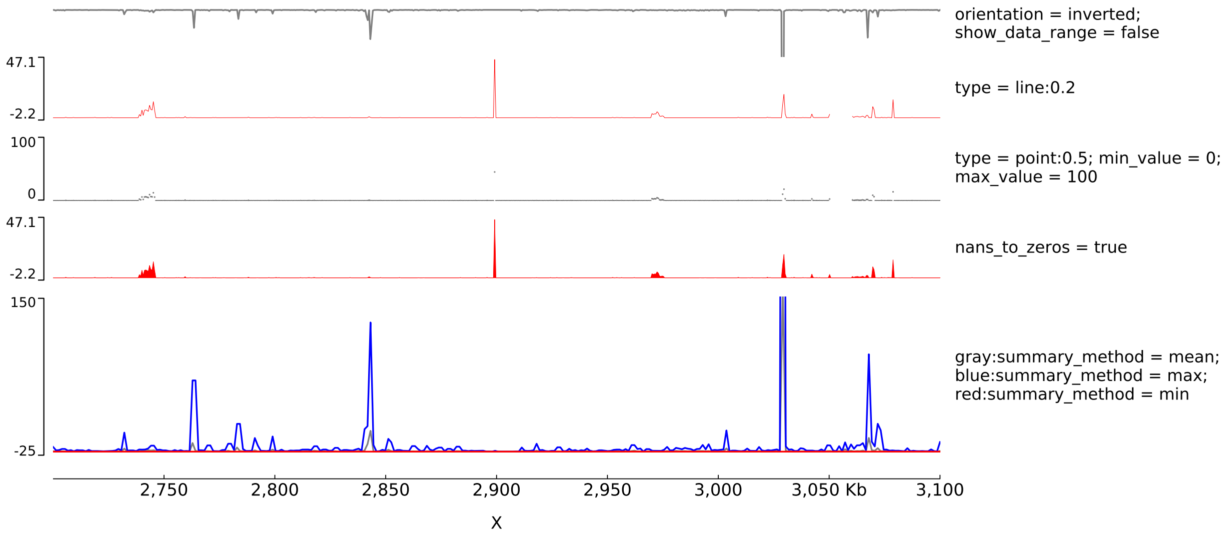


Figure 14: Multiple tracks with bigwigs

6.8 Hi-C tracks

```
[hic matrix]
file = Li_et_al_2015.h5
title = depth = 200000; transform = log1p; min_value = 5
depth = 200000
min_value = 5
transform = log1p
file_type = hic_matrix
show_masked_bins = false

[hic matrix]
file = Li_et_al_2015.h5
title = depth = 250000; orientation = inverted; colormap = PuRd; min_value = 5;
        max_value = 70
min_value = 5
```

```
max_value = 70
depth = 250000
colormap = PuRd
file_type = hic_matrix
show_masked_bins = false
orientation = inverted

[spacer]
height = 0.5

[hic matrix]
file = Li_et_al_2015.h5
title = depth = 300000; transform = log1p; colormap Blues (TADs:
        overlay_previous = share-y; line_width = 1.5)
colormap = Blues
min_value = 10
max_value = 150
depth = 300000
transform = log1p
file_type = hic_matrix

[tads]
file = tad_classification.bed
#title = TADs color = none; border_color = black
file_type = domains
border_color = black
color = none
height = 5
line_width = 1.5
overlay_previous = share-y
show_data_range = false

[spacer]
height = 0.5

[hic matrix]
file = Li_et_al_2015.h5
title = depth = 250000; transform = log1p; colormap = bone_r (links: overlay_previous = share-y;
        links_type = triangles; color = darkred; line_style = dashed, bigwig: color = red)
colormap = bone_r
min_value = 15
max_value = 200
depth = 250000
transform = log1p
file_type = hic_matrix
show_masked_bins = false

[test arcs]
file = links2.links
title =
links_type = triangles
line_style = dashed
overlay_previous = share-y
line_width = 0.8
color = darkred
show_data_range = false


[test bigwig]
file = bigwig2_X_2.5e6_3.5e6.bw
color = red
height = 4
title =
overlay_previous = yes
min_value = 0
max_value = 50
show_data_range = false

[spacer]
height = 0.5

[hic matrix]
file = Li_et_al_2015.h5
title = depth = 200000; show_masked_bins = true; colormap =
        ['blue', 'yellow', 'red']; max_value = 150
depth = 200000
colormap = ['blue', 'yellow', 'red']
max_value = 150
file_type = hic_matrix
show_masked_bins = true

[spacer]
height = 0.1

[x-axis]
```

```
$ pyGenomeTracks  --tracks browser_tracks_hic.ini --region X:2500000-3500000 --trackLabelFraction 0.23 --width
↪  38 --dpi 130 -o master_plot_hic.png
```

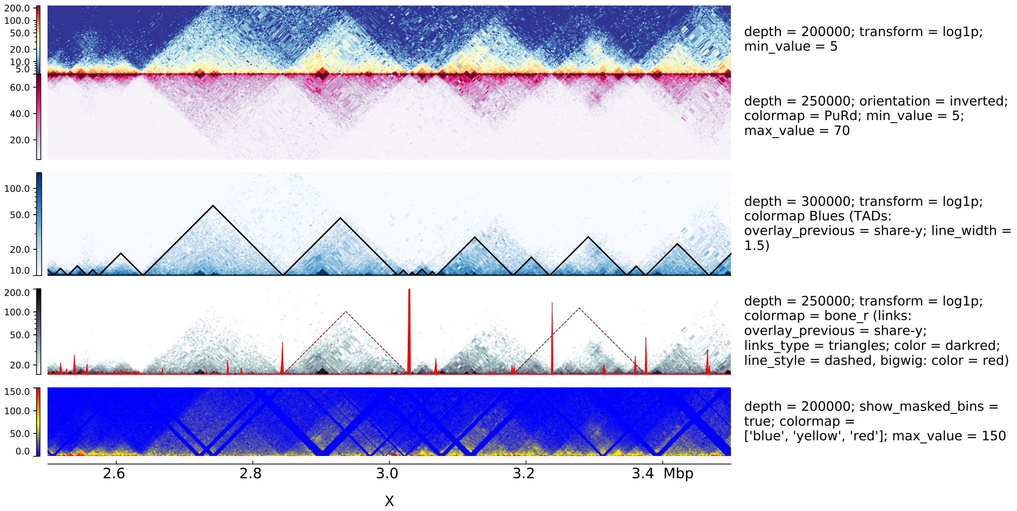


Figure 15: Hi-C tracks

References

Abdennur, N. and Mirny, L. A. (2019). Cooler: scalable storage for Hi-C data and other genomically labeled arrays. *Bioinformatics*, **36**(1), 311–316.

Cubeñas-Potts, C. *et al.* (2017). Different enhancer classes in drosophila bind distinct architectural proteins and mediate unique chromatin interactions and 3d architecture. *Nucleic acids research*, **45**(4), 1714–1730.

Karolchik, D. *et al.* (2004). The ucsc table browser data retrieval tool. *Nucleic acids research*, **32**(suppl_1), D493–D496.

Kent, W. J. *et al.* (2010). Bigwig and bigbed: enabling browsing of large distributed datasets. *Bioinformatics*, **26**(17), 2204–2207.

Li, L. *et al.* (2015). Widespread rearrangement of 3d chromatin organization underlies polycomb-mediated stress-induced silencing. *Molecular cell*, **58**(2), 216–231.

Ramírez, F. *et al.* (2018). High-resolution tads reveal dna sequences underlying genome organization in flies. *Nature communications*, **9**(1), 1–15.

Acknowledgments

I want to thank my collaboration partners from all over the world. Without their support, ideas, discussions, and help, this thesis would not have been possible to finish. Especially I want to thank current and former members of the Bioinformatics Unit of the Max-Planck Institute of Immunobiology and Epigenetics Freiburg: Leily Rabbani, Gautier Richard, Gina Renschler, Devon Rayn, Fidel Ramírez and Thomas Manke; current and former members of the Institute of Experimental and Clinical Pharmacology and Toxicology, University Freiburg and the later Institute of Cardiovascular Physiology, Goethe University Frankfurt: Ralf Gilsbach, Stephan Nothjunge, and Rebecca Bednarz; Lucille Lopez-Delisle of the UPDUB, ISREC Department, École Polytechnique Fédérale de Lausanne, Switzerland; and Nezar Abedennur of the Institute for Medical Engineering and Science, Massachusetts Institute of Technology, Cambridge, MA, USA.

I want to thank the members of the Bioinformatics Lab at the University Freiburg, especially Simon Bray, Anup Kumar, and Mehmet Tekman, for proofreading so many of my papers, ideas, discussions, and for a great time in quite a few pub visits. Also, I want to thank Milad Miladi, Anika Erxleben, Simon Bray, Leily Rabbani, and Björn Grüning for proofreading my thesis.

I want to thank Prof. Rolf Backofen and Björn Grüning for giving me the opportunity to make this PhD, and the general supervision of my thesis.

Bibliography

[1] Tong Ihn Lee and Richard A Young. “Transcriptional regulation and its misregulation in disease”. In: *Cell* 152.6 (2013), pp. 1237–1251 (cit. on p. 7).

[2] Shelley L Berger. “The complex language of chromatin regulation during transcription”. In: *Nature* 447.7143 (2007), pp. 407–412 (cit. on p. 7).

[3] Stephan Kadauke and Gerd A Blobel. “Chromatin loops in gene regulation”. In: *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms* 1789.1 (2009), pp. 17–25 (cit. on p. 7).

[4] Andrew J Bannister and Tony Kouzarides. “Regulation of chromatin by histone modifications”. In: *Cell research* 21.3 (2011), pp. 381–395 (cit. on p. 7).

[5] Roberta Goldshlag Cooks, Marjorie Hertz, Marissa Bat Miriam Katznelson, and Richard M Goodman. “A new nail dysplasia syndrome with onychonychia and absence and/or hypoplasia of distal phalanges”. In: *Clinical genetics* 27.1 (1985), pp. 85–91 (cit. on p. 7).

[6] NC Nevin, PS Thomas, DJ Eedy, and C Shepherd. “Anonychia and absence/hypoplasia of distal phalanges (Cooks syndrome): report of a second family.” In: *Journal of medical genetics* 32.8 (1995), pp. 638–641 (cit. on p. 7).

[7] Martin Franke, Daniel M Ibrahim, Guillaume Andrey, et al. “Formation of new chromatin domains determines pathogenicity of genomic duplications”. In: *Nature* 538.7624 (2016), pp. 265–269 (cit. on pp. 7, 31).

[8] Job Dekker, Karsten Rippe, Martijn Dekker, and Nancy Kleckner. “Capturing chromosome conformation”. In: *science* 295.5558 (2002), pp. 1306–1311 (cit. on pp. 7, 27).

[9] Marieke Simonis, Petra Klous, Erik Splinter, et al. “Nuclear organization of active and inactive chromatin domains uncovered by chromosome conformation captureonchip (4C)”. In: *Nature genetics* 38.11 (2006), p. 1348 (cit. on pp. 8, 27).

[10] Zhihu Zhao, Gholamreza Tavoosidana, Mikael Sjölinder, et al. “Circular chromosome conformation capture (4C) uncovers extensive networks of epigenetically regulated intra- and interchromosomal interactions”. In: *Nature genetics* 38.11 (2006), p. 1341 (cit. on pp. 8, 27).

[11] Josée Dostie, Todd A Richmond, Ramy A Arnaout, et al. “Chromosome Conformation Capture Carbon Copy (5C): a massively parallel solution for mapping interactions between genomic elements”. In: *Genome research* 16.10 (2006), pp. 1299–1309 (cit. on pp. 8, 27).

[12] Erez Lieberman-Aiden, Nynke L. Van Berkum, Louise Williams, et al. “Comprehensive mapping of long-range interactions reveals folding principles of the human genome”. In: *Science* 326.5950 (Oct. 2009), pp. 289–293. arXiv: `arXiv:1011.1669v3` (cit. on pp. 8, 27, 30, 41, 42).

[13] Jesse R Dixon, Siddarth Selvaraj, Feng Yue, et al. “Topological domains in mammalian genomes identified by analysis of chromatin interactions”. In: *Nature* 485.7398 (2012), pp. 376–380 (cit. on pp. 8, 31).

[14] Suhas SP Rao, Miriam H Huntley, Neva C Durand, et al. “A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping”. In: *Cell* 159.7 (2014), pp. 1665–1680 (cit. on pp. 8, 9, 28, 31, 40, 43, 45–49, 51, 52, 54, 76, 78).

[15] Borbala Mifsud, Filipe Tavares-Cadete, Alice N Young, et al. “Mapping long-range promoter contacts in human cells with high-resolution capture Hi-C”. In: *Nature genetics* 47.6 (2015), p. 598 (cit. on pp. 8, 27).

[16] Takashi Nagano, Yaniv Lubling, Tim J Stevens, et al. “Single-cell Hi-C reveals cell-to-cell variability in chromosome structure”. In: *Nature* 502.7469 (2013), pp. 59–64 (cit. on pp. 8, 27, 28, 63, 64).

[17] Maxwell R Mumbach, Adam J Rubin, Ryan A Flynn, et al. “HiChIP: efficient and sensitive analysis of protein-directed genome architecture”. In: *Nature methods* 13.11 (2016), pp. 919–922 (cit. on pp. 8, 27).

[18] Borbala Mifsud, Inigo Martincorena, Elodie Darbo, et al. “GOTHiC, a probabilistic model to resolve complex biases and to identify real interactions in Hi-C data”. In: *PloS one* 12.4 (2017), e0174744 (cit. on p. 9).

[19] Yaqiang Cao, Zhaoxiong Chen, Xingwei Chen, et al. “Accurate loop calling for 3D genomic data with cLoops”. In: *Bioinformatics* 36.3 (2020), pp. 666–675 (cit. on p. 9).

[20] Jonathan Cairns, Paula Freire-Pritchett, Steven W Wingett, et al. “CHiCAGO: robust detection of DNA looping interactions in Capture Hi-C data”. In: *Genome biology* 17.1 (2016), pp. 1–17 (cit. on p. 9).

[21] Jingtian Zhou, Jianzhu Ma, Yusi Chen, et al. “Robust single-cell Hi-C clustering by convolution-and random-walk–based imputation”. In: *Proceedings of the National Academy of Sciences* 116.28 (2019), pp. 14011–14018 (cit. on pp. 9, 65, 67, 70–72).

[22] Philip A Knight and Daniel Ruiz. “A fast algorithm for matrix balancing”. In: *IMA Journal of Numerical Analysis* 33.3 (2013), pp. 1029–1047 (cit. on pp. 9, 40).

[23] Maxim Imakaev, Geoffrey Fudenberg, Rachel Patton McCord, et al. “Iterative correction of Hi-C data reveals hallmarks of chromosome organization”. In: *Nature methods* 9.10 (2012), pp. 999–1003 (cit. on pp. 9, 40).

[24] Wenyuan Li, Ke Gong, Qingjiao Li, Frank Alber, and Xianghong Jasmine Zhou. “Hi-Corrector: a fast, scalable and memory-efficient package for normalizing large-scale Hi-C data”. In: *Bioinformatics* 31.6 (2015), pp. 960–962 (cit. on p. 9).

[25] Sven Heinz, Christopher Benner, Nathanael Spann, et al. “Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities”. In: *Molecular cell* 38.4 (2010), pp. 576–589 (cit. on pp. 9, 51).

[26] Simon Hettrick. “Research software sustainability: Report on a Knowledge Exchange workshop”. In: (2016) (cit. on p. 9).

[27] Joachim Wolff, Vivek Bhardwaj, Stephan Nothjunge, et al. “Galaxy HiCExplorer: a web server for reproducible Hi-C data analysis, quality control and visualization”. In: *Nucleic acids research* 46.W1 (2018), W11–W16 (cit. on pp. 11, 35, 72).

[28] Joachim Wolff, Leily Rabbani, Ralf Gilsbach, et al. “Galaxy HiCExplorer 3: a web server for reproducible Hi-C, capture Hi-C and single-cell Hi-C data analysis, quality control and visualization”. In: *Nucleic acids research* 48.W1 (2020), W177–W184 (cit. on pp. 11, 35, 58, 63, 68, 72).

[29] Joachim Wolff, Rolf Backofen, and Björn Grüning. "Robust and efficient single-cell Hi-C clustering with approximate k-nearest neighbor graphs". In: *Bioinformatics* (May 2021). btab394. eprint: `https://academic.oup.com/bioinformatics/advance-article-pdf/doi/10.1093/bioinformatics/btab394/38104756/btab394.pdf` (cit. on pp. 11, 63, 66, 67, 69, 70).

[30] Joachim Wolff, Nezar Abdennur, Rolf Backofen, and Björn Grüning. "Scool: a new data storage format for single-cell Hi-C data". In: *Bioinformatics* (2020) (cit. on pp. 11, 63, 64).

[31] Lucille Lopez-Delisle, Leily Rabbani, Joachim Wolff, et al. "pyGenomeTracks: reproducible plots for multivariate genomic datasets". In: *Bioinformatics* 37.3 (2021), p. 422 (cit. on pp. 11, 35, 57).

[32] Xuepeng Chen, Yuwen Ke, Keliang Wu, et al. "Key role for CTCF in establishing chromatin structure in human embryos". In: *Nature* 576.7786 (2019), pp. 306–310 (cit. on p. 12).

[33] Maria Samata, Anastasios Alexiadis, Gautier Richard, et al. "Intergenerationally Maintained Histone H4 Lysine 16 Acetylation Is Instructive for Future Gene Activation". In: *Cell* (2020) (cit. on p. 12).

[34] Ting Xie, Fu-Gui Zhang, Hong-Yu Zhang, et al. "Biased gene retention during diploidization in Brassica linked to three-dimensional genome organization". In: *Nature plants* 5.8 (2019), pp. 822–832 (cit. on p. 12).

[35] Kris G Alavattam, So Maezawa, Akihiko Sakashita, et al. "Attenuated chromatin compartmentalization in meiosis and its maturation in sperm development". In: *Nature structural & molecular biology* 26.3 (2019), pp. 175–184 (cit. on p. 13).

[36] Lin Li, Preston Williams, Wendan Ren, et al. "YY1 interacts with guanine quadruplexes to regulate DNA looping and gene expression". In: *Nature Chemical Biology* 17.2 (2021), pp. 161–168 (cit. on p. 13).

[37] Linhua Sun, Yuqing Jing, Xinyu Liu, et al. "Heat stress-induced transposon activation correlates with 3D chromatin organization rearrangement in Arabidopsis". In: *Nature communications* 11.1 (2020), pp. 1–13 (cit. on p. 13).

[38] Yufeng Qin, Sara A Grimm, John D Roberts, Kaliopi Chrysovergis, and Paul A Wade. "Alterations in promoter interaction landscape and transcriptional network underlying metabolic adaptation to diet". In: *Nature communications* 11.1 (2020), pp. 1–16 (cit. on p. 13).

[39] Héloïse Muller, José Gil Jr, and Ines Anna Drinnenberg. "The impact of centromeres on spatial genome architecture". In: *Trends in Genetics* 35.8 (2019), pp. 565–578 (cit. on p. 13).

[40] Zhijun Han, Kairong Cui, Katarzyna Placek, et al. "Diploid genome architecture revealed by multi-omic data of hybrid mice". In: *Genome Research* 30.8 (2020), pp. 1097–1106 (cit. on p. 13).

[41] Muhammad Shuaib, Krishna Mohan Parsi, Manjula Thimma, et al. "Nuclear AGO1 Regulates Gene Expression by Affecting Chromatin Architecture in Human Cells". In: *Cell Systems* 9.5 (2019), pp. 446–458 (cit. on p. 14).

[42] Laura Arrigoni, Hoor Al-Hasani, Fidel Ramírez, et al. "RELACS nuclei barcoding enables high-throughput ChIP-seq". In: *Communications biology* 1.1 (2018), pp. 1–12 (cit. on p. 14).

[43]Friedrich Miescher-Rüsch. *Ueber die chemische Zusammensetzung der Eiterzellen*. 1871 (cit. on p. 15).

[44]Oswald T Avery, Colin M MacLeod, and Maclyn McCarty. “Studies on the chemical nature of the substance inducing transformation of pneumococcal types: induction of transformation by a desoxyribonucleic acid fraction isolated from pneumococcus type III”. In: *The Journal of experimental medicine* 79.2 (1944), pp. 137–158 (cit. on p. 15).

[45]James D Watson and Francis HC Crick. “Genetical implications of the structure of deoxyribonucleic acid”. In: *Nature* 171.4361 (1953), pp. 964–967 (cit. on p. 15).

[46]JD Watson and FHC Chick. “Molecular structure of deoxypentose nucleic acids”. In: *Nature* 171 (1953) (cit. on p. 15).

[47]James D Watson, Francis HC Crick, et al. “A structure for deoxyribose nucleic acid”. In: *Nature* 171.4356 (1953), pp. 737–738 (cit. on p. 15).

[48]Rosalind E Franklin and Raymond George Gosling. “Evidence for 2-chain helix in crystalline structure of sodium deoxyribonucleate”. In: *Nature* 172.4369 (1953), pp. 156–157 (cit. on p. 15).

[49]Rosalind E Franklin and Raymond G Gosling. “Molecular configuration in sodium thymonucleate”. In: *Nature* 171.4356 (1953), pp. 740–741 (cit. on p. 15).

[50]Bruce Alberts, Alexander Johnson, Julian Lewis, et al. “Molecular biology of the cell”. In: (2018) (cit. on p. 17).

[51]Allison Piovesan, Maria Chiara Pelleri, Francesca Antonaros, et al. “On the length, weight and GC content of the human genome”. In: *BMC research notes* 12.1 (2019), pp. 1–7 (cit. on p. 18).

[52]Hui Bin Sun, Jin Shen, and Hiroki Yokota. “Size-dependent positioning of human chromosomes in interphase nuclei”. In: *Biophysical journal* 79.1 (2000), pp. 184–190 (cit. on p. 18).

[53]Katharina Munk. *Taschenlehrbuch Biologie: Genetik*. Georg Thieme Verlag, 2010 (cit. on p. 18).

[54]Donald E Olins and Ada L Olins. “Chromatin history: our view from the bridge”. In: *Nature reviews Molecular cell biology* 4.10 (2003), pp. 809–814 (cit. on p. 18).

[55]Conrad H Waddington. “The epigenotype”. In: *Endeavour* 1 (1942), pp. 18–20 (cit. on p. 19).

[56]Ina Goy. “Was Aristotle the ‘father’of the epigenesis doctrine?” In: *History and philosophy of the life sciences* 40.2 (2018), pp. 1–16 (cit. on p. 19).

[57]Eva Jablonka and Marion J Lamb. “The changing concept of epigenetics”. In: *Annals of the New York Academy of Sciences* 981.1 (2002), pp. 82–96 (cit. on p. 19).

[58]Cathérine Dupont, D Randall Armant, and Carol A Brenner. “Epigenetics: definition, mechanisms and clinical perspective”. In: *Seminars in reproductive medicine*. Vol. 27. 5. NIH Public Access. 2009, p. 351 (cit. on p. 19).

[59]CH Waddington. *The basic ideas of biology. In “Towards a Theoretical Biology”(P. Idegomena and CH Waddington, Eds.). Vol. 1*. 1968 (cit. on p. 19).

[60] Gerda Egger, Gangning Liang, Ana Aparicio, and Peter A Jones. "Epigenetics in human disease and prospects for epigenetic therapy". In: *Nature* 429.6990 (2004), pp. 457–463 (cit. on pp. 20, 21).

[61] Andrew P Feinberg. "The key role of epigenetics in human disease prevention and mitigation". In: *New England Journal of Medicine* 378.14 (2018), pp. 1323–1334 (cit. on p. 20).

[62] Huda Y Zoghbi and Arthur L Beaudet. "Epigenetics and human disease". In: *Cold Spring Harbor perspectives in biology* 8.2 (2016), a019497 (cit. on p. 20).

[63] Jason I Feinberg, Kelly M Bakulski, Andrew E Jaffe, et al. "Paternal sperm DNA methylation associated with early signs of autism risk in an autism-enriched cohort". In: *International journal of epidemiology* 44.4 (2015), pp. 1199–1210 (cit. on p. 20).

[64] Stephan Nothjunge, Thomas G Nührenberg, Björn A Grüning, et al. "DNA methylation signatures follow preformed chromatin compartments in cardiac myocytes". In: *Nature communications* 8.1 (2017), pp. 1–9 (cit. on p. 20).

[65] Xiaojing Yang, Han Han, Daniel D De Carvalho, et al. "Gene body methylation can alter gene expression and is a therapeutic target in cancer". In: *Cancer cell* 26.4 (2014), pp. 577–590 (cit. on p. 20).

[66] Jing Liao, Rahul Karnik, Hongcang Gu, et al. "Targeted disruption of DNMT1, DNMT3A and DNMT3B in human embryonic stem cells". In: *Nature genetics* 47.5 (2015), pp. 469–478 (cit. on p. 20).

[67] Melanie Ehrlich. "DNA hypomethylation in cancer cells". In: *Epigenomics* 1.2 (2009), pp. 239–259 (cit. on p. 20).

[68] Artem Barski, Suresh Cuddapah, Kairong Cui, et al. "High-resolution profiling of histone methylations in the human genome". In: *Cell* 129.4 (2007), pp. 823–837 (cit. on p. 21).

[69] Tiantian Zhang, Zhuqiang Zhang, Qiang Dong, Jun Xiong, and Bing Zhu. "Histone H3K27 acetylation is dispensable for enhancer activity in mouse embryonic stem cells". In: *Genome biology* 21.1 (2020), pp. 1–7 (cit. on p. 21).

[70] Yonggang Zhou, Johnny Kim, Xuejun Yuan, and Thomas Braun. "Epigenetic modifications of stem cells: a paradigm for the control of cardiac progenitor cells". In: *Circulation research* 109.9 (2011), pp. 1067–1081 (cit. on pp. 22, 23).

[71] Andrew P Hutchins, Li Sun, Gang Ma, and Xiuling Fu. "Chromatin and epigenetic rearrangements in embryonic stem cell fate transitions". In: *Frontiers in Cell and Developmental Biology* 9 (2021), p. 174 (cit. on pp. 22, 23).

[72] Stefan Schoenfelder and Peter Fraser. "Long-range enhancer–promoter contacts in gene expression control". In: *Nature Reviews Genetics* (2019), p. 1 (cit. on p. 23).

[73] Wulan Deng, Jongjoo Lee, Hongxin Wang, et al. "Controlling long-range genomic interactions at a native locus by targeted tethering of a looping factor". In: *Cell* 149.6 (2012), pp. 1233–1244 (cit. on p. 23).

[74] Len A Pennacchio, Wendy Bickmore, Ann Dean, Marcelo A Nobrega, and Gill Bejerano. "Enhancers: five essential questions". In: *Nature Reviews Genetics* 14.4 (2013), pp. 288–295 (cit. on p. 23).

[75] KA Schafer. “The cell cycle: a review”. In: *Veterinary pathology* 35.6 (1998), pp. 461–478 (cit. on p. 24).

[76] Stavros Lomvardas, Gilad Barnea, David J Pisapia, et al. “Interchromosomal interactions and olfactory receptor choice”. In: *Cell* 126.2 (2006), pp. 403–413 (cit. on p. 27).

[77] Hugo Würtele and Pierre Chartrand. “Genome-wide scanning of HoxB1-associated loci in mouse ES cells using an open-ended Chromosome Conformation Capture methodology”. In: *Chromosome Research* 14.5 (2006), pp. 477–495 (cit. on p. 27).

[78] Satish Sati and Giacomo Cavalli. “Chromosome conformation capture technologies and their impact in understanding genome function”. In: *Chromosoma* 126.1 (2017), pp. 33–44 (cit. on p. 27).

[79] Christopher M Dundas, Daniel Demonte, and Sheldon Park. “Streptavidin–biotin technology: improvements and innovations in chemical and biological applications”. In: *Applied microbiology and biotechnology* 97.21 (2013), pp. 9343–9353 (cit. on p. 28).

[80] Eleftherios P Diamandis and Theodore K Christopoulos. “The biotin-(strept) avidin system: principles and applications in biotechnology”. In: *Clinical chemistry* 37.5 (1991), pp. 625–636 (cit. on p. 28).

[81] Mitsutaka Kadota, Osamu Nishimura, Hisashi Miura, et al. “Multifaceted Hi-C benchmarking: what makes a difference in chromosome-scale genome scaffolding?” In: *GigaScience* 9.1 (2020), giz158 (cit. on p. 28).

[82] Guoliang Li, Liuyang Cai, Huidan Chang, et al. “Chromatin interaction analysis with paired-end tag (ChIA-PET) sequencing technology and application”. In: *BMC genomics* 15.12 (2014), pp. 1–10 (cit. on p. 29).

[83] Haiming Chen, Nicholas Comment, Jie Chen, et al. “Chromosome conformation of human fibroblasts grown in 3-dimensional spheroids”. In: *Nucleus* 6.1 (2015), pp. 55–65 (cit. on p. 28).

[84] Susan M Gasser. “Visualizing chromatin dynamics in interphase nuclei”. In: *Science* 296.5572 (2002), pp. 1412–1416 (cit. on p. 28).

[85] Takashi Nagano, Yaniv Lubling, Csilla Várnai, et al. “Cell-cycle dynamics of chromosomal organization at single-cell resolution”. In: *Nature* 547.7661 (2017), pp. 61–67 (cit. on pp. 28, 63, 65, 67–71).

[86] Ilya M Flyamer, Johanna Gassler, Maxim Imakaev, et al. “Single-nucleus Hi-C reveals unique chromatin reorganization at oocyte-to-zygote transition”. In: *Nature* 544.7648 (2017), pp. 110–114 (cit. on pp. 28, 63, 65).

[87] Tim J Stevens, David Lando, Srinjan Basu, et al. “3D structures of individual mammalian genomes studied by single-cell Hi-C”. In: *Nature* 544.7648 (2017), pp. 59–64 (cit. on pp. 28, 63–65).

[88] Longzhi Tan, Dong Xing, Chi-Han Chang, Heng Li, and X Sunney Xie. “Three-dimensional genome structures of single diploid human cells”. In: *Science* 361.6405 (2018), pp. 924–928 (cit. on pp. 28, 63).

[89] Vijay Ramani, Xinxian Deng, Ruolan Qiu, et al. “Sci-Hi-C: a single-cell Hi-C method for mapping 3D genome organization in large number of single cells”. In: *Methods* 170 (2020), pp. 61–68 (cit. on pp. 28, 63).

[90] Tom Misteli. “Chromosome territories: The arrangement of chromosomes in the nucleus”. In: *Nat. Educ* 1.1 (2008) (cit. on p. 30).

[91] C Rabl. “Uber Zelltheilung. Morphol. Jahrb. 10, 214-330”. In: *Rabl21410Morphol. Jahrb* (1885) (cit. on p. 30).

[92] Thomas Cremer, Christoph Cremer, H Baumann, et al. “Rabl’s model of the interphase chromosome arrangement tested in Chinise hamster cells by premature chromosome condensation and laser-UV-microbeam experiments”. In: *Human genetics* 60.1 (1982), pp. 46–56 (cit. on p. 30).

[93] Luis A Parada, Philip G McQueen, Peter J Munson, and Tom Misteli. “Conservation of relative chromosome positioning in normal and cancer cells”. In: *Current Biology* 12.19 (2002), pp. 1692–1697 (cit. on p. 30).

[94] Andreas Bolzer, Gregor Kreth, Irina Solovei, et al. “Three-dimensional maps of all chromosomes in human male fibroblast nuclei and prometaphase rosettes”. In: *PLoS Biol* 3.5 (2005), e157 (cit. on p. 30).

[95] Luis A Parada, Jeffrey J Roix, and Tom Misteli. “An uncertainty principle in chromosome positioning”. In: *Trends in cell biology* 13.8 (2003), pp. 393–396 (cit. on p. 30).

[96] Hisashi Tamaru. “Confining euchromatin/heterochromatin territory: jumonji crosses the line”. In: *Genes & development* 24.14 (2010), pp. 1465–1478 (cit. on p. 30).

[97] Brenda R Grimes, Jennifer Babcock, M Katharine Rudd, Brian Chadwick, and Huntington F Willard. “Assembly and characterization of heterochromatin and euchromatin on human artificial chromosomes”. In: *Genome biology* 5.11 (2004), pp. 1–14 (cit. on p. 30).

[98] Yota Murakami. “Heterochromatin and Euchromatin”. In: *Encyclopedia of Systems Biology*. Ed. by Werner Dubitzky, Olaf Wolkenhauer, Kwang-Hyun Cho, and Hiroki Yokota. New York, NY: Springer New York, 2013, pp. 881–884 (cit. on p. 30).

[99] Iain Williamson, Lauren Kane, Paul S Devenney, et al. “Developmentally regulated Shh expression is robust to TAD perturbations”. In: *Development* 146.19 (2019) (cit. on p. 31).

[100] Rachel Patton McCord, Noam Kaplan, and Luca Giorgetti. “Chromosome conformation capture and beyond: toward an integrative view of chromosome structure and function”. In: *Molecular cell* 77.4 (2020), pp. 688–708 (cit. on pp. 31, 40, 47).

[101] Boyan Bonev and Giacomo Cavalli. “Organization and function of the 3D genome”. In: *Nature Reviews Genetics* 17.11 (2016), p. 661 (cit. on pp. 31, 48, 52).

[102] Navneet Matharu and Nadav Ahituv. “Minor loops in major folds: enhancer–promoter looping, chromatin restructuring, and their association with transcriptional regulation and disease”. In: *PLoS genetics* 11.12 (2015), e1005640 (cit. on p. 31).

[103] Tae Hoon Kim, Ziedulla K Abdullaev, Andrew D Smith, et al. “Analysis of the vertebrate insulator protein CTCF-binding sites in the human genome”. In: *Cell* 128.6 (2007), pp. 1231–1245 (cit. on p. 31).

[104] Björn Grüning, Ryan Dale, Andreas Sjödin, et al. “Bioconda: sustainable and comprehensive software distribution for the life sciences”. In: *Nature methods* 15.7 (2018), p. 475 (cit. on p. 33).

[105] David Bernstein. “Containers and cloud: From lxc to docker to kubernetes”. In: *IEEE Cloud Computing* 1.3 (2014), pp. 81–84 (cit. on p. 34).

[106] Ryan Chamberlain. “Using Docker to support reproducible research”. In: (2014) (cit. on p. 34).

[107] Enis Afgan, Dannon Baker, Marius van den Beek, et al. “The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update”. In: *Nucleic acids research* 44.W1 (July 2016), W3–W10 (cit. on p. 34).

[108] Jason A Reuter, Damek V Spacek, and Michael P Snyder. “High-throughput sequencing technologies”. In: *Molecular cell* 58.4 (2015), pp. 586–597 (cit. on p. 35).

[109] 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”. In: *Nucleic acids research* 38.6 (2010), pp. 1767–1771 (cit. on p. 35).

[110] Brent Ewing, LaDeana Hillier, Michael C Wendl, and Phil Green. “Base-calling of automated sequencer traces using Phred. I. Accuracy assessment”. In: *Genome research* 8.3 (1998), pp. 175–185 (cit. on p. 35).

[111] Brent Ewing and Phil Green. “Base-calling of automated sequencer traces using phred. II. Error probabilities”. In: *Genome research* 8.3 (1998), pp. 186–194 (cit. on p. 35).

[112] Daehwan Kim, Ben Langmead, and Steven L Salzberg. “HISAT: a fast spliced aligner with low memory requirements”. In: *Nature methods* 12.4 (2015), pp. 357–360 (cit. on p. 37).

[113] Ben Langmead and Steven L Salzberg. “Fast gapped-read alignment with Bowtie 2”. In: *Nature methods* 9.4 (2012), p. 357 (cit. on p. 37).

[114] Heng Li and Richard Durbin. “Fast and accurate long-read alignment with Burrows–Wheeler transform”. In: *Bioinformatics* 26.5 (2010), pp. 589–595 (cit. on p. 37).

[115] Fidel Ramírez, Vivek Bhardwaj, Laura Arrigoni, et al. “High-resolution TADs reveal DNA sequences underlying genome organization in flies”. In: *Nature communications* 9.1 (2018), pp. 1–15 (cit. on pp. 38, 43).

[116] Nezar Abdennur and Leonid A Mirny. “Cooler: scalable storage for Hi-C data and other genomically labeled arrays”. In: *Bioinformatics* 36.1 (2020), pp. 311–316 (cit. on pp. 38, 64, 75).

[117] Neva C Durand, Muhammad S Shamim, Ido Machol, et al. “Juicer provides a one-click system for analyzing loop-resolution Hi-C experiments”. In: *Cell systems* 3.1 (2016), pp. 95–98 (cit. on p. 38).

[118] Nicolas Servant, Nelle Varoquaux, Bryan R Lajoie, et al. “HiC-Pro: an optimized and flexible pipeline for Hi-C data processing”. In: *Genome biology* 16.1 (2015), pp. 1–11 (cit. on p. 38).

[119] Aaron TL Lun, Malcolm Perry, and Elizabeth Ing-Simmons. “Infrastructure for genomic interactions: Bioconductor classes for Hi-C, ChIA-PET and related experiments”. In: *F1000Research* 5 (2016) (cit. on p. 38).

[120] Steven Wingett, Philip Ewels, Mayra Furlan-Magaril, et al. “HiCUP: pipeline for mapping and processing Hi-C data”. In: *F1000Research* 4 (2015) (cit. on p. 39).

[121] Philip Ewels, Måns Magnusson, Sverker Lundin, and Max Käller. “MultiQC: summarize analysis results for multiple tools and samples in a single report”. In: *Bioinformatics* 32.19 (2016), pp. 3047–3048 (cit. on p. 39).

[122]Hendrik Marks, Hindrik HD Kerstens, Tahsin Stefan Barakat, et al. “Dynamics of gene silencing during X inactivation using allele-specific RNA-seq”. In: *Genome biology* 16.1 (2015), pp. 1–20 (cit. on p. 40).

[123]Marie Zufferey, Daniele Tavernari, Elisa Oricchio, and Giovanni Ciriello. “Comparison of computational methods for the identification of topologically associating domains”. In: *Genome biology* 19.1 (2018), pp. 1–18 (cit. on p. 45).

[124]Zhi-Hua Zhou. “Ensemble learning”. In: *Machine Learning*. Springer, 2021, pp. 181–210 (cit. on p. 46).

[125]Yoav Freund and Robert E Schapire. “A decision-theoretic generalization of on-line learning and an application to boosting”. In: *Journal of computer and system sciences* 55.1 (1997), pp. 119–139 (cit. on p. 46).

[126]Trevor Hastie, Saharon Rosset, Ji Zhu, and Hui Zou. “Multi-class adaboost”. In: *Statistics and its Interface* 2.3 (2009), pp. 349–360 (cit. on p. 46).

[127]Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. “Exploratory undersampling for class-imbalance learning”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.2 (2008), pp. 539–550 (cit. on p. 46).

[128]Leo Breiman. “Bagging predictors”. In: *Machine learning* 24.2 (1996), pp. 123–140 (cit. on p. 46).

[129]Oluwatosin Oluwadare and Jianlin Cheng. “ClusterTAD: an unsupervised machine learning approach to detecting topologically associated domains of chromosomes from Hi-C data”. In: *BMC bioinformatics* 18.1 (2017), pp. 1–14 (cit. on p. 46).

[130]Wenbao Yu, Bing He, and Kai Tan. “Identifying topologically associating domains and subdomains by Gaussian mixture model and proportion test”. In: *Nature communications* 8.1 (2017), pp. 1–9 (cit. on p. 46).

[131]Chin-Tong Ong and Victor G Corces. “CTCF: an architectural protein bridging genome topology and function”. In: *Nature Reviews Genetics* 15.4 (2014), pp. 234–246 (cit. on p. 47).

[132]Vinay Singh Tanwar, Cynthia C Jose, and Suresh Cuddapah. “Role of CTCF in DNA damage response”. In: *Mutation Research/Reviews in Mutation Research* 780 (2019), pp. 61–68 (cit. on p. 47).

[133]Namyoung Jung and Tae-Kyung Kim. “Advances in higher-order chromatin architecture: the move towards 4D genome”. In: *BMB reports* 54.5 (2021), p. 233 (cit. on p. 47).

[134]Frank Wilcoxon. “Individual Comparisons by Ranking Methods”. In: *Biometrics* 1.6 (1945), pp. 80–83 (cit. on p. 47).

[135]Arya Kaul, Sourya Bhattacharyya, and Ferhat Ay. “Identifying statistically significant chromatin contacts from Hi-C data with FitHiC2”. In: *Nature protocols* 15.3 (2020), pp. 991–1012 (cit. on pp. 48, 51).

[136]Tarik J Salameh, Xiaotao Wang, Fan Song, et al. “A supervised learning framework for chromatin loop detection in genome-wide contact maps”. In: *Nature communications* 11.1 (2020), pp. 1–12 (cit. on pp. 48, 51).

[137]A Colin Cameron and Pravin K Trivedi. “Regression-based tests for overdispersion in the Poisson model”. In: *Journal of econometrics* 46.3 (1990), pp. 347–364 (cit. on pp. 48, 49).

[138] Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. “edgeR: a Bioconductor package for differential expression analysis of digital gene expression data”. In: *Bioinformatics* 26.1 (2010), pp. 139–140 (cit. on p. 50).

[139] Davis J McCarthy, Yunshun Chen, and Gordon K Smyth. “Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation”. In: *Nucleic acids research* 40.10 (2012), pp. 4288–4297 (cit. on p. 50).

[140] Cyril Matthey-Doret, Lyam Baudry, Axel Breuer, et al. “Computer vision for pattern detection in chromosome contact maps”. In: *Nature communications* 11.1 (2020), pp. 1–11 (cit. on pp. 51, 75).

[141] Edward J Banigan and Leonid A Mirny. “Loop extrusion: theory meets single-molecule experiments”. In: *Current opinion in cell biology* 64 (2020), pp. 124–138 (cit. on p. 52).

[142] Maxwell R Mumbach, Ansuman T Satpathy, Evan A Boyle, et al. “Enhancer connectome in primary human cells identifies target genes of disease-associated DNA elements”. In: *Nature genetics* 49.11 (2017), pp. 1602–1612 (cit. on p. 52).

[143] Guillaume Andrey, Robert Schöpflin, Ivana Jerković, et al. “Characterization of hundreds of regulatory landscapes in developing limbs reveals two regimes of chromatin folding”. In: *Genome research* 27.2 (2017), pp. 223–233 (cit. on pp. 59, 62).

[144] Vijay Ramani, Xinxian Deng, Ruolan Qiu, et al. “Massively multiplex single-cell Hi-C”. In: *Nature methods* 14.3 (2017), pp. 263–266 (cit. on pp. 63–65, 72).

[145] Johanna Gassler, Hugo B Brandão, Maxim Imakaev, et al. “A mechanism of cohesin-dependent loop extrusion organizes zygotic genome architecture”. In: *The EMBO journal* 36.24 (2017), pp. 3600–3618 (cit. on p. 65).

[146] David Lando, Tim J Stevens, Srinjan Basu, and Ernest D Laue. “Calculation of 3D genome structures for comparison of chromosome conformation capture experiments with microscopy: An evaluation of single-cell Hi-C protocols”. In: *Nucleus* 9.1 (2018), pp. 190–201 (cit. on p. 65).

[147] Kevin Beyer and Jonathan Goldstein. “R. Ramakrishnan, U. Shaft. When is Nearest Neighbors Meaningful”. In: *Proc of ICDT*. Vol. 99. 1999 (cit. on p. 66).

[148] Alexander Hinneburg, Charu C Aggarwal, and Daniel A Keim. “What is the nearest neighbor in high dimensional spaces?” In: *26th Internat. Conference on Very Large Databases*. 2000, pp. 506–515 (cit. on p. 66).

[149] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. “On the surprising behavior of distance metrics in high dimensional space”. In: *International conference on database theory*. Springer. 2001, pp. 420–434 (cit. on p. 66).

[150] Andrei Z Broder. “On the resemblance and containment of documents”. In: *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*. IEEE. 1997, pp. 21–29 (cit. on p. 66).

[151] Peter Kerpedjiev, Nezar Abdennur, Fritz Lekschas, et al. “HiGlass: web-based visual exploration and analysis of genome interaction maps”. In: *Genome biology* 19.1 (2018), pp. 1–12 (cit. on p. 72).

[152] Fidel Ramírez, Friederike Dündar, Sarah Diehl, Björn A Grüning, and Thomas Manke. “deepTools: a flexible platform for exploring deep-sequencing data”. In: *Nucleic acids research* 42.W1 (2014), W187–W191 (cit. on p. 72).

[153]Johannes Köster and Sven Rahmann. “Snakemake—a scalable bioinformatics workflow engine”. In: *Bioinformatics* 28.19 (2012), pp. 2520–2522 (cit. on p. 73).

[154]Peter Amstutz, Michael R Crusoe, Nebojša Tijanić, et al. “Common workflow language, v1. 0”. In: (2016) (cit. on p. 73).

[155]Bérénice Batut, Saskia Hiltemann, Andrea Bagnacani, et al. “Community-driven data analysis training for biology”. In: *Cell systems* 6.6 (2018), pp. 752–758 (cit. on p. 73).

[156]Adrian L Sanborn, Suhas SP Rao, Su-Chen Huang, et al. “Chromatin extrusion explains key features of loop and domain formation in wild-type and engineered genomes”. In: *Proceedings of the National Academy of Sciences* 112.47 (2015), E6456–E6465 (cit. on p. 77).

[157]Sylvain Foissac, Sarah Djebali, Kylie Munyard, et al. “Multi-species annotation of transcriptome and chromatin structure in domesticated animals”. In: *BMC biology* 17.1 (2019), pp. 1–25 (cit. on p. 77).

[158]Jason D Buenrostro, Paul G Giresi, Lisa C Zaba, Howard Y Chang, and William J Greenleaf. “Transposition of native chromatin for multimodal regulatory analysis and personal epigenomics”. In: *Nature methods* 10.12 (2013), p. 1213 (cit. on p. 77).

[159]Robert A Beagrie, Antonio Scialdone, Markus Schueler, et al. “Complex multi-enhancer contacts captured by genome architecture mapping”. In: *Nature* 543.7646 (2017), pp. 519–524 (cit. on p. 77).

[160]Sofia A Quinodoz, Noah Ollikainen, Barbara Tabak, et al. “Higher-order inter-chromosomal hubs shape 3D genome organization in the nucleus”. In: *Cell* 174.3 (2018), pp. 744–757 (cit. on p. 77).

[161]Jochen Graw. *Genetik, 6. Auflage*. Springer Berlin Heidelberg, 2015 (cit. on p. 77).

[162]Michael Boutros and Julie Ahringer. “The art and design of genetic screens: RNA interference”. In: *Nature Reviews Genetics* 9.7 (2008), pp. 554–566 (cit. on p. 77).

[163]Shilu Zhang, Deborah Chasman, Sara Knaack, and Sushmita Roy. “In silico prediction of high-resolution Hi-C interaction matrices”. In: *Nature communications* 10.1 (2019), pp. 1–18 (cit. on p. 78).

[164]Ron Schwessinger, Matthew Gosden, Damien Downes, et al. “DeepC: predicting 3D genome folding using megabase-scale transfer learning”. In: *Nature Methods* 17.11 (2020), pp. 1118–1124 (cit. on p. 78).