UNIVERSITÀ DI PISA

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale in Ingegneria Biomedica

MSc Thesis

3D Exploration of Genomes: A Standardized Hi-C Data Analysis

Relatore:
Ing. ALESSIO BECHINI

Candidato:
RICCARDO CALANDRELLI

Correlatore:
Prof. SHENG ZHONG

Anno Accademico 2015/2016

Declaration of Authorship

I, Riccardo Calandrelli, declare that this thesis titled, '3D Exploration of Genomes: A Standardized Hi-C Data Analysis' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"By going to one more round when you don't think you can, that's what makes all the difference in your life."

Rocky Balboa

UNIVERSITÀ DI PISA

Dipartimento di Ingegneria dell'Informazione
Corso di Laurea Magistrale in Ingegneria Biomedica

Abstract

Riccardo Calandrelli

The biological information of the organisms is stored in the DNA, which folds up into elaborate physical structures inside the cell nucleus. The packing of the genetic material is not only useful to allow spatial compactness, but it assumes also a functional relevance. In such a way, the understanding that nuclear organization plays an important role in the epigenetic regulation poses considerable challenges.
During the past fifteen years, several techniques have been developed to explore the architecture of chromatin within the nucleus, such as Chromosome Conformation Capture (3C) and derived 3C protocols (4C, 5C) or Fluorescence In-Situ Hybridization (FISH). However, a genome-wide analysis was only possible after 2009, when the Hi-C protocol was introduced, which first allowed for a comprehensive mapping of genome interactions. In order to process Hi-C data, several software are needed to perform each step of the analysis, from the preprocessing to the visualization of the data. Moreover, a normalization procedure is required to remove biases, introduced by the experimental protocol itself or related to genome features.
To address these needs we developed *HiCtool*, a standardized bioinformatic pipeline that handles efficiently the Hi-C analysis, from the preprocessing and the normalization of the data to the visualization of heatmaps. HiCtool contains the first pipeline for the data preprocessing and also a section for the topological domains analysis, to allow further investigation about genomes conformations. By using HiCtool, we successfully run several Hi-C datasets of different cell lines and conditions of human and mouse, with the aim of creating the biggest library of standardized processed data ever. We collected all these datasets on *GITAR* (Genome Interaction Tools and Resources), a framework we built to work on and manage genomic interaction data, which is available online at *genomegitar.org*. GITAR contains either a standardized library to process Hi-C data (HiCtool) and the collection of datasets we processed. In such a way, we provide users a powerful and easy tool, both for analysis and epigenetic comparative studies on different species or conditions.

Acknowledgements

Oggi giungo alla fine di un lungo percorso, accademico ma anche di vita, dove finisce un capitolo e ne inizia un altro e sono tante le persone a cui vorrei dedicare un pensiero e dire grazie. Ognuno di voi, in un modo o nell'altro, mi ha reso quello che sono.

Desidero ringraziare il mio relatore Prof. Alessio Bechini per la sua disponibilità, i suoi consigli e avermi sempre seguito pur non avendo compiuto il lavoro di tesi direttamente sotto la sua supervisione.

I would like to thank Prof. Sheng Zhong for having given me the opportunity to fulfill this thesis at University of California San Diego. Thank you for your guidance, suggestions and assistance, I am really glad and proud of having met you and of my experience in your lab. A special "thank you" to Qiuyang, for all the time spent together, a good friend more than a coworker. Thanks also to all the guys in the lab, you made me feel welcomed since the first moment.

Un grazie a tutti quelli con cui ho condiviso la mia esperienza a Pisa.

Grazie a tutti i miei coinquilini, da Simone e Andrea, con cui ho condiviso i primi anni, i più duri ma anche i più belli, a Matteo spesso compagno di viaggio, a Alberto e Osvaldo con cui ho passato l'ultimo anno a Pisa. Mi sono affezionato a ognuno di voi, siete diventati tutti dei buoni amici.

Un grazie a Andrea con cui ho passato tutti gli anni dell'università, compagno di studio e carissimo amico. Tante studiate insieme, tante sudate e preoccupazioni che ci scambiavamo a vicenda, ma senza tutto ciò sarebbe stata più difficile da affrontare. Non dimenticherò mai la tua disponibilità ad aiutare sempre. Un grosso in bocca al lupo per tutto! Sei in gamba, ti meriti tante soddisfazioni.

Grazie anche a Simone, che insieme a Andrea è stato mio compagno di studi durante la triennale. Siamo anche stati insieme al nostro primo concerto di Robbie Williams, esperienza fantastica, indimenticabile! Un grosso in bocca al lupo anche a te.

Grazie a Paolo, divenuto un caro amico e "mentore" per San Diego. Mi mancano le nostre lunghe chiacchierate come quelle durante i viaggi per Pisa, sarà ora di rifarsi?

Grazie a Leonardo, grande persona e amico vero. Quante serate abbiamo fatto insieme, dalla Versilia, a San Diego fino a Las Vegas, ma soprattutto quante ne verranno! Ci conosciamo solo da pochi anni ma l'empatia che abbiamo non ha eguali. Grazie per ogni cosa che hai fatto per me, anche piccola ma dove hai sempre messo il cuore.

Grazie a Rafael, compagno sin dall'infanzia e uno dei miei più grandi amici. Abbiamo passato quasi una vita insieme, dall'asilo al liceo! Grazie per la tua amicizia vera, i tuoi consigli, la tua lealtà. In questi ultimi anni siamo stati insieme raramente, ma il bello di noi è che ogni volta che ci siamo rivisti sembrava come non ci fossimo mai separati. In bocca al lupo per la tua carriera, sarai un grande odontoiatra!

Grazie a Roberto, grande amico e compagno di una miriade di serate. Grazie per tutte

le chiacchierate, le telefonate quando ero a San Diego, le battute che solo tu sai fare. Sei sempre riuscito a tirarmi fuori una risata anche nei momenti più duri.
Grazie a Claudio, grande persona e motivatore. Il tuo celebre "no pain no gain" è divenuto ormai anche un mio marchio di fabbrica. Grazie per i tuoi sempre preziosi consigli.
Un grazie a Paolo e Stefano, miei maestri di ballo ma soprattutto di vita e amici veri. Grazie perché con voi ho passato tutti gli anni dell'adolescenza e mi avete insegnato il valore della disciplina, dello sforzo, della fatica, del fissare gli obbiettivi per poi raggiungerli e superarli. Con voi ho avuto un rapporto speciale e senza quei 9 anni insieme sicuramente oggi non sarei quello che sono.
Un grazie alle mie due ballerine, Fabiana prima e Giorgia poi, con cui ho condiviso quei 9 anni fatti di ore e ore in palestra, di soddisfazioni, gioie e anche dolori. Condividere questa esperienza insieme, comprendere, accettare, imparare ad andare nella stessa direzione quando si hanno caratteri diametralmente opposti, tutto ciò ha formato la mia persona e vi dico grazie.
Grazie alla mia famiglia.
Grazie ai nonni Riccardo e Graziella, Amato e Isolina per l'amore che mi avete sempre dato, per tutte le preghiere spese per me, per tutto ciò che avete fatto per me. Vi voglio tanto bene!
Grazie agli zii Eugenio e Angela, alle cugine Roberta e Sabrina e a Marco che ormai fa parte della famiglia. Praticamente sono cresciuto insieme a voi, grazie a Roberta la mia prima amica e con cui ho sempre avuto un rapporto speciale. Un bacio al piccolo Leonardo, Roberta e Marco avete un bimbo bellissimo!
Grazie a Viola, mia sorella a cui voglio un bene dell'anima. Grazie per tutto ciò che abbiamo passato insieme, per avermi fatto capire cosa sia l'amore fraterno, per quasi una vita spesa insieme a te. Non siamo mai stati di tante parole, non è nemmeno mai servito tra noi, ma per te ci sarò sempre e sappi che la stima che provi per me è la stessa che io ho per te. Sarai una grande infermiera, perché in tutto quello che fai ci metti il cuore. In bocca al lupo per la laurea e una vita ricca di soddisfazioni. Ti voglio bene!
Infine grazie ai miei genitori, le persone più importanti della mia vita, per avermi fatto crescere con amore. Mi avete insegnato i valori dell'impegno, rispetto, lealtà, dell'essere prima di apparire, del guardare avanti ma con un occhio sempre alle spalle. Grazie per avermi sempre supportato, per avermi fatto fare le mie scelte, per avermi sempre dato un consiglio prezioso, per avermi reso la persona che sono e di cui vado fiero.
Mamma, sei sempre stata una presenza costante nella mia vita e nei miei pensieri, grazie per il tuo amore e tutto ciò che hai sempre fatto per me.
Babbo, una volta alla fine di un discorso mi dicesti: "Sono solo il figlio di Amato il pescatore, e ne vado fiero!". Beh, sappi che ieri, oggi e sempre, dovunque andrò, qualunque cosa farò, io sarò fiero di essere il figlio di Zefferino il geometra.

Contents

List of Figures

List of Tables

Acronyms

3C Chromosome Conformation Capture.

4C Circularized Chromosome Conformation Capture.

5C Carbon Copy Chromosome Conformation Capture.

BRE B-Recognition Element.

ChIA-PET Chromatin Interaction Analysis with Paired-End Tag Sequencing.

ChIP Chromatin Immunoprecipitation.

CTCF CCCTC-binding Factor.

CTs Chromosome Territories.

DHSs Deoxyribonuclease I Hypersensitive Sites.

DI Directionality Index.

DPE Downstream Promoter Element.

DSBs Double Strand Breaks.

ES Embryonic Stem.

ESCs Embryonic Stem Cells.

FEND Fragment-end.

FISH Fluorescence In-Situ Hybridisation.

GEO Gene Expression Omnibus.

GRO-seq Global Run on Sequencing.

GWAS Genome-wide Association Studies.

HMEC Human Mammary Epithelial Cells.

HMM Hidden Markov Model.

Inr Initiator.

iPSCs induced Pluripotent Stem Cells.

LCR Locus Control Region.

LMA Ligation Mediated Amplification.

NADs Nucleolus Associated Domains.

NL Nuclear Lamina.

PC Principal Component.

PCA Principal Component Analysis.

PCR Polymerase Chain Reaction.

PET Paired End Tag.

PIC Pre-initiation Complex.

PIL Python Image Library.

PRC2 Polycomb Repressive Complex 2.

RE Restriction Enzyme.

SCNAs Somatic Copy-Number Alterations.

SNPs Single Nucleotide Polymorphisms.

TADs Topologically Associating Domains.

TBP TATA-box Binding Protein.

TREs Transcriptional Regulatory Elements.

TSSs Transcription Start Sites.

Dedicated to my Family, that I love more than any other thing in the world...

Chapter 1

Introduction

In 1953, Watson and Crick's discovery of the DNA two entwined helices and paired organic bases revealed the basic structure of DNA [1]. However, genomes are more than linear sequences [2], with DNA folding up into elaborate physical structures that allow for extreme spatial compactness of the genetic material. Analysis of the spatial organization of chromosomes reveals complex three-dimensional networks of chromosomal interactions. These interactions affect gene expression at multiple levels, including long-range control by distant enhancers and repressors, coordinated expression of genes and modification of epigenetic states [3]. Therefore, the three-dimensional (3D) conformation of chromosomes assumes a central role in epigenetic regulation, being involved in compartmentalizing the nucleus and bringing widely separated functional elements into close spatial proximity [4].

To explore long-range chromatin interactions, several techniques have been utilized, such as Chromosome Conformation Capture (3C) and 3C derived protocols (4C and 5C) or 3D Fluorescence In-Situ Hybridization (3D FISH). However, only in 2009 it has been possible to initiate a 3D genome-wide analysis, when the Hi-C protocol was developed, which first allowed for a comprehensive mapping of genome interactions. The representation of this kind of data usually resorts to *contact matrices* (intra- or inter-chromosomal), where each matrix entry represents the number of ligation products between the two chromosome parts involved. These matrices are typically depicted as heatmaps, where intensity indicates the contact frequency.
From 2009 on, investigation of the three-dimensional organization of genomes by performing Hi-C experiments on human and mouse, led to the identification of large megabase-sized chromatin interaction domains, called *Topological Domains*. They look as highly self-interacting regions, seen as "triangles" in the heatmaps, occupying approximately 91% of the genome. This study gave a prominent contribute to map the three-dimensional

conformation of the DNA at a kilo-base scale, and laid the groundwork for understanding the link between chromatin structure and transcriptional control in mammalian genomes.

During these years, several software applications have been developed for analyzing and visualizing genomic interaction data, with different features and outputs, that we collect on the *4DN software library*[1] . From this survey we realized that, albeit the presence of a large amount of source data and several analysis and visualization applications, a software to perform a standardized processing of Hi-C data was still missing. We have addressed this need by developing *HiCtool*, a bioinformatics tool for Hi-C data analysis, with the aim of creating a standardized and flexible framework to process and visualize Hi-C datasets, and perform a comprehensive intra-chromosomal and topological domains analysis. A complete Hi-C data analysis requires the integrated use of several software tools to perform the data processing; moreover, data are affected by biases, so a normalization procedure should be integrated as well. To deal with these problems, HiCtool provides a complete and exhaustive pipeline, which includes all the software needed for the overall analysis. For each step of the analysis the syntax of the code is shown and clearly explained, with the key advantage that any other software documentation is not required to perform it. This design lets users obtain the results easily and quickly, through a simple, clear, and user-friendly procedure. Thus, HiCtool is a standardized and customizable pipeline to work on and visualize Hi-C data. In such a way, the user, even beginner, is able to understand and manage the Hi-C data processing and this is the big aim achieved. In addition, HiCtool provides the first pipeline to carry out a topological domain study, which enables a comprehensive and deeper analysis about the three-dimensional conformation of genomes.
Then we developed *GITAR*[2] (Genome Interaction Tools and Resources), a comprehensive solution to work on and manage genomic interaction data. GITAR provides users with HiCtool, a standardized way to process and visualize Hi-C data, and an exhaustive collection of processed datasets for different species, cell lines, and conditions. We produce four different outputs per dataset: intra-chromosomal contact matrices (observed, expected and normalized), Directionality Index (a statistic used to identify systematically topological domains in the genome), HMM states for the Directionality Index and topological domains coordinates. In such a way, GITAR allows for the first time to work on and compare different data in a consistent way, providing the largest collection of processed datasets ever. We strongly believe that this could be a major contribute for epigenetic comparative studies, such as cell differentiation and cancer samples analysis.

Finally, in this thesis it will be discussed the role of Hi-C data in important disease-associated studies. Due to the polymer nature of DNA, there is a strong relationship between Hi-C contacts and genomic distance, and this property makes Hi-C a powerful

method to detect large-scale genome aberrations, such as translocations, which are a common feature of cancer genomics. Lastly, possible future works on inter-chromosomal conformations will be outlined, starting from a preliminary analysis we have already performed on Hi-C interchromosomal maps.

[1] http://data.genomegitar.org/4DN_software.php
[2] http://genomegitar.org

Chapter 2

Chromatin structure and function in three-dimensions

2.1 The three-dimensional chromatin structure

Chromosomes are some of the most complex molecular entities in the cell. The molecular composition of the chromatin fibre is highly diverse along its length, and the fibre is intricately folded in three dimensions [5]. The classic illustration of chromosomes (Figure 2.1) appears during mitosis in metaphase, when chromatin is highly condensed. During interphase, chromatin is much less condensed, becoming accessible to the transcriptional machinery, epigenetic factors and DNA repair enzymes. The three-dimensional folding of the DNA is achieved thanks to many structural proteins as histones, that are the chief protein components of chromatin. Histones package and order the DNA into structural units called nucleosomes, acting as spools around which DNA winds.

2.1.1 Chromatin fibre

The DNA molecule consists of two helical chains each coiled around the same axis, and each with a pitch of 3.4 nanometers and a radius of 1 nanometer [1]. The backbone of the DNA strand is made by alternating phosphate and sugar residues [7]. The sugars are joined together by phosphate groups that form phosphodiester bonds between the third and fifth carbon atoms of adjacent sugar rings. The DNA double helix is stabilized primarily by two forces: hydrogen bonds between nucleotides and base-stacking interactions among aromatic nucleobases [8].
Chromatin is a complex of macromolecules found in cells, consisting of DNA, proteins

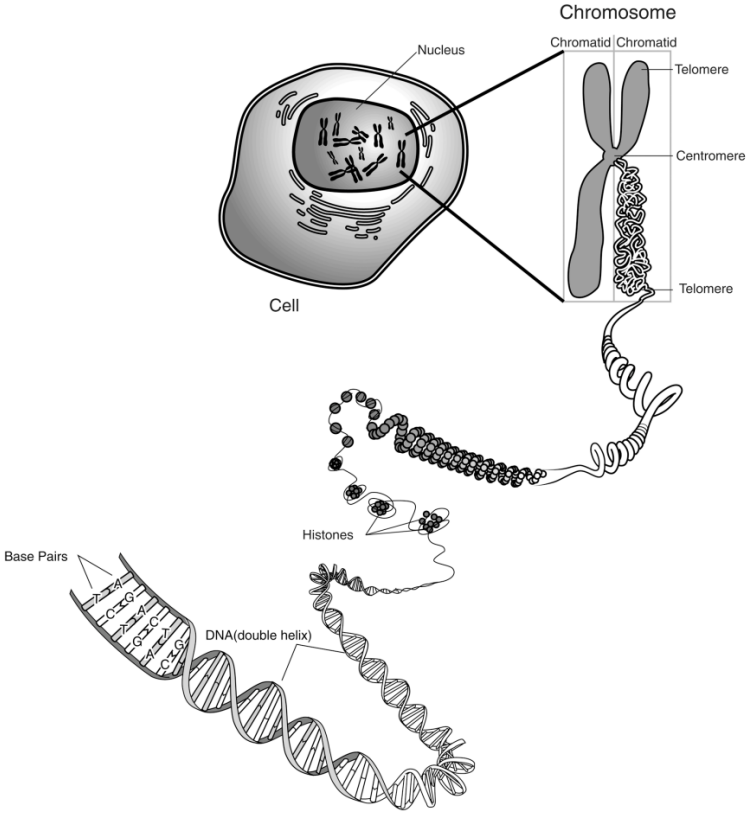

FIGURE 2.1: DNA configuration in the cell nucleus [6]. The highlighted levels from top to down are: chromosome, chromatin fibre, individual histones, the DNA double helix molecule.

and RNA. In general terms, there are three levels of chromatin organization. First, DNA wraps around histone proteins forming nucleosomes, the basic units of DNA packaging in eukaryotes. A nucleosome consists of a segment of DNA coiled in sequence around eight histone protein cores, covering 147 base pairs [9], with linker DNA of around 20-50 bp in length connecting nucleosomes. At a higher level, multiple histones wrap into a 30 nm fibre consisting of nucleosome arrays in their most compact form [10]. Although the "30 nm" chromatin solenoidal fibres have been observed and resolved in vitro, how much of the DNA assumes these formations in vivo is still under debate [11], [12].

2.1.2 Topologically Associating Domains

At the level of hundreds of kilobases to few megabases, DNA folds up into higher level structures named Topologically Associated Domains (TADs) [13], characterized by enriched chromosomal contacts within TADs than between TADs [14]. These highly self-interacting regions have been observed in *Drosophila Melanogaster* [15], *Mus musculus* and *Homo sapiens*, indicating that such spatial organization seems to be a general property of genomes [16], which attests to its importance in nuclear biology. Moreover, TADs

have been indicated to be in relation to gene regulation and other nuclear functions [17].

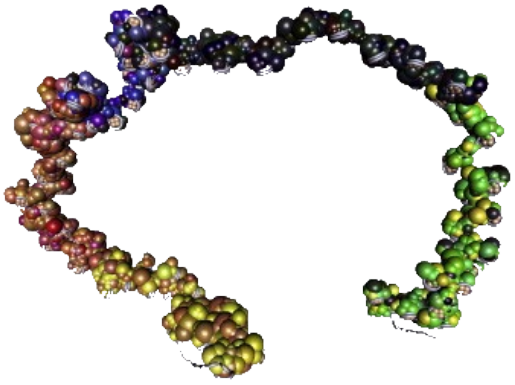

FIGURE 2.2: Cartoon model of a chromosomal fibre [17]. Illustration of a chromosomal fibre segmentation into domains of distinct chromatin types, each consisting of a specific combination of proteins and histone modifications (indicated by colors).

Several studies have indicated that specific regions of chromosomes are located in close proximity to the nuclear lamina (NL) [2]. This has led to the idea that certain genomic elements may be attached to the nuclear lamina, which may contribute to the spatial organization of chromosomes inside the nucleus [18], providing anchoring for chromosomal domains [17].
Several architectural proteins, such as CCCTC-binding factor (CTCF), Cohesin complex and Mediator complex, are important for the establishment and maintenance of a variety of cell type-specific and -invariant genome organizational features, including enhancer-promoter contacts and long-range inter-TAD chromatin contacts, as well as TAD boundaries [19]. The establishment and maintenance of both inter- and intra-TAD chromatin interactions is thought to occur via recruitment of Cohesin, a protein complex that is known for its role in sister chromatid cohesion during mitosis. Cohesin can be recruited by the insulator protein CTCF, which governs cell type-invariant features of genome organization and is required for proper Cohesin localization to CTCF-enriched sites. In such a way, CTCF, Cohesin, and Mediator act as the "architectural" proteins of the nucleus (Figure 2.3). In mouse embryonic stem cells (ESCs) and neural progenitor cells, CTCF, Cohesin, and Mediator are found at more than 80% of chromatin interactions, further supporting the notion that the three proteins play a central role in organizing chromatin [19].

Several recent studies interrogated changes in genome organization upon differentiation of ESCs and during reprogramming of somatic cells to induced pluripotent stem cells (iPSCs), mediated by the expression of the reprogramming factors Oct4, Sox2, Klf4 and cMyc [20]. These reports revealed a large-scale re-organization of long-range, inter-TAD

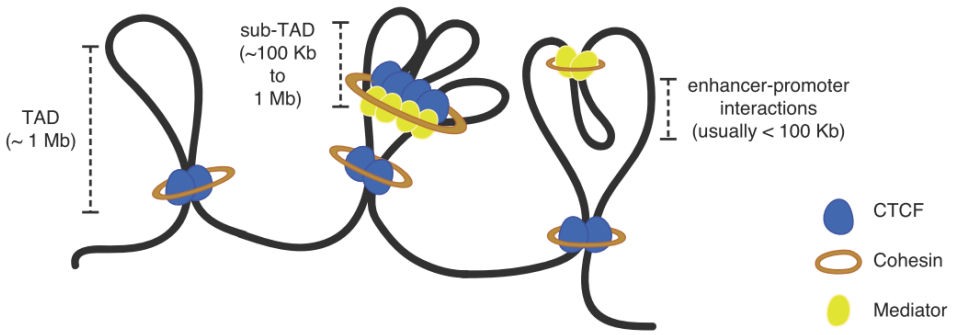

FIGURE 2.3: Architectural proteins act combinatorially to organize chromatin at different length-scales [19]. TAD boundaries are enriched for CTCF and Cohesin, but these proteins can also act in combination with other factors, such as Mediator to partition these large Mb-scale TADs into smaller sub-TADs and facilitate enhancer-promoter interactions.

chromatin contacts of pluripotency loci including the Nanog, Dppa2/4, Oct4 and Sox2 genes during differentiation, and demonstrated that the ESC-specific organization of the genome is re-established upon reprogramming to iPSCs. This pluripotency-specific organization of the mammalian genome suggests a role for pluripotency-associated gene regulatory networks in the organization of long-range chromatin contacts in ESCs and iPSCs. In support of this idea, genomic regions bounded by the master pluripotency transcription factors Oct4, Sox2, and Nanog were found to interact with each other over large distances in the ESC nucleus (Figure 2.4) [19]. Similarly, extended genomic regions enriched for binding by the transcriptionally repressive Polycomb repressive complex 2 (PRC2), which mediates methylation of histone H3 at lysine 27, also co-localize in ESCs, although separately from the pluripotency transcription factors (Figure 2.4) [21].

2.1.3 Chromosome territories

In eukaryotic cell nuclei there is evidence for a compartmentalized nuclear architecture based on chromosome territories (CTs) [22]. A correlation between CT location and human chromosomes size was described, in which smaller chromosomes are generally situated towards the interior and larger chromosomes towards the periphery of the nucleus [23]. However, the finding that CTs with similar DNA content, but with very different gene densities, occupy distinct exterior and interior nuclear positions, indicates that gene content is a key determinant of CT positioning. As example, although both chromosomes 18 and chromosome 19 have a similar DNA content (85 and 67 Mb, respectively), the gene-poor chromosome 18 territories were typically found at the nuclear

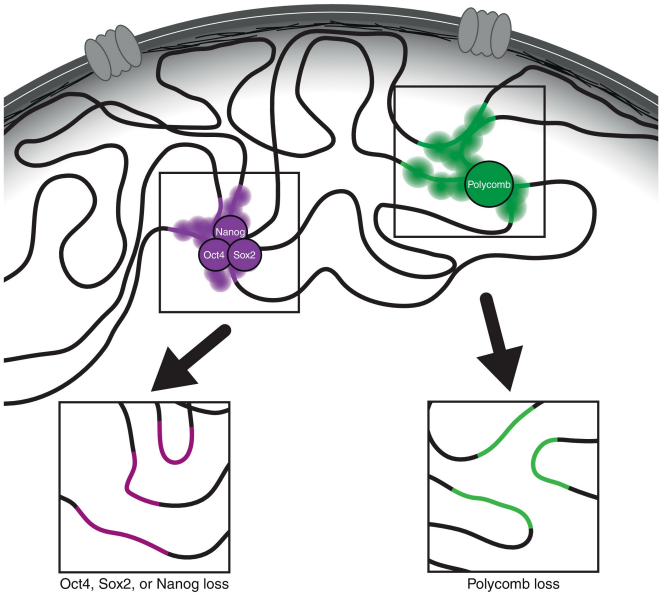

FIGURE 2.4: Gene regulatory factors shape inter-TAD chromatin interactions within the pluripotent nucleus [19]. Chromatin within the ESC nucleus is compartmentalized based on the preferential co-localization of open, transcriptionally permissive 'A' compartment chromatin (white background away from the nuclear periphery) or closed, nuclear lamina-associated 'B' compartment chromatin (gray background, nuclear lamina-associated). Within the 'A' compartment, genomic regions enriched for binding by pluripotency transcription factors (purple) co-localize, as regions enriched for Polycomb proteins and the H3K27me3 histone mark do (green). Loss of the pluripotency transcription factors or the Polycomb repressive complex 2 (arrows) result in loss of inter-TAD interactions, without disrupting the overall A versus B compartmental structure of the nucleus.

periphery, whereas the gene-rich chromosome 19 territories were located in the nuclear interior [22].

2.2 Three-dimensional transcriptional regulation

As stated by the ENCODE project [24], nuclear organization has emerged as an important layer of the epigenetic transcriptional regulation. Specifically, long-range chromosomal associations between genomic regions are an important factor in the regulation of gene expression, by forming loops that often link enhancers with promoters at considerable distances, even located in different chromosomes [25]. DNA methylation and histone modification are involved in epigenetic regulation of gene expression [24], as well as chromatin composition and its distribution along chromosomes [26].

2.2.1 RNA polymerase II core promoters are key components in the regulation of gene expression

The identification and characterization of core promoters and transcription starting sites (TSSs) are crucial to understand how RNA polymerase II transcription is controlled [27]. Most genes have multiple promoters, within which there are multiple TSSs, therefore each promoter usage generates diversity and complexity in the transcriptome and proteome. A set of common DNA sequence elements and patterns are associated with core promoters, which characterize the expression of the downstream genes [27] (see Figure 2.5). The TATA box, located 28-34 bp upstream of the TSSs, is one of the most known transcription factor binding sites. Its consensus sequence, TATAA, binds the TATA-box binding protein (TBP), which is part of the pre-initiation complex (PIC) and this enforces the PIC to select a TSSs in the nearby space. The initiator (Inr) element, defined by the consensus sequence YYANWYY[1], where A is the +1 position, often co-occurs with a TATA-box element, and they are the only known core promoter elements that, alone, can recruit the PIC and initiate transcription. The downstream promoter element (DPE), which lies 28-32 bp downstream of the TSSs, has a similar function of the TATA-box in directing the PIC to a nearby TSSs. The B-recognition element (BRE), lies upstream of the TATA-box and it can either increase or decrease transcription rates in eukaryotes. CpG islands are genomic sequences in which CG dinucleotides are over-represented, and 50% of human promoters are associated with CpG islands.

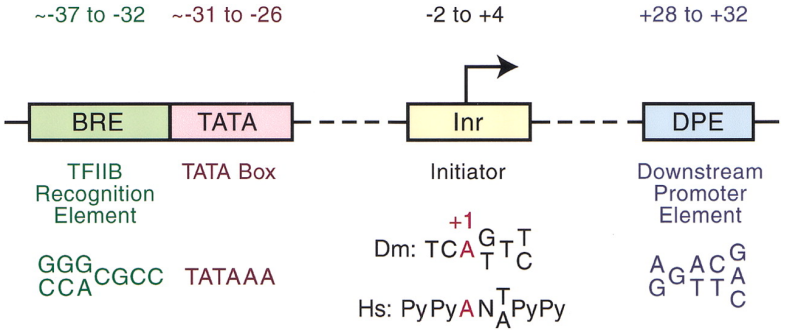

FIGURE 2.5: Core promoter elements [28]. Schematic representation of core promoter elements that can participate in transcription by RNA polymerase II. Each of these elements is found in only a subset of core promoters. The BRE is an upstream extension of a subset of TATA boxes. The DPE requires an Inr and it is located 28-32 bp downstream to the A_{+1} nucleotide in the Inr. The DPE consensus was determined with *Drosophila* transcription factors and core promoters. The Inr consensus sequence is shown for both *Drosophila* (Dm) and *Humans* (Hs).

[1]In nucleic acid notation for DNA, Y (pYrimidine) stands for C/T (cytosine or thymine, which are both pyrimidines), N (Nucleobase) is any of the four bases, W (Weak) stands for A/T (adenine or thymine, which both form only two hydrogen bonds).

2.2.2 Promoter-enhancer interactions

In eukaryotes, gene expression is controlled by short regulatory DNA sequences called enhancers. Classic promoter-enhancer loops are at the MYC gene [29], which has an enhancer ∼400 kb away from the gene promoter, α-globin [29] and β-globin [30], where a specific loop is formed between the β-globin locus control region (LCR) and active globin genes. Understanding how an enhancer selects its promoter (or promoters) is still a big challenge. He *et al.* introduced an integrated method for predicting enhancer targets called IM-PET and, by applying it, they assigned targets for a set of enhancers across 12 cell types in human [31]. They predicted 208,342 enhancers in total, averaging 17,362 enhancers per cell type, and 161,999 active promoters in these cell types. About enhancer-promoter (EP) interactions, 441,879 unique EP pairs across the 12 cell type were estimated, averaging 36,823 interactions per cell type.

2.2.3 Transcription factories

Transcription factories are discrete sites in the nucleus, enriched of multiple active RNA polymerases, where transcription occurs and it is regulated, operating as activators or repressors [32], [33].
While previous studies gave a view of transcription where protein factors are recruited to and move along the chromatin template, a different model is that active RNA polymerase II is concentrated and anchored to a nuclear substructure and the gene loci move to that [34]. To support the idea of factories as self-determining nuclear structures, it was investigated what happens to them when transcription is stopped. It was seen that the foci remain stable after the inhibition of both transcriptional initiation and elongation with heat shock, so visible transcription factories are not simply accumulations of RNA polymerase II on active genes, but they exist as independent nuclear subcompartments [35].

2.3 Techniques to explore chromatin structure

Here, an overview of the experimental techniques used to explore chromatin structure is presented. There are two major approaches to achieve this: the first is based on microscopy and visualization combined with fluorescent labeling, the other is based on the Chromosome Conformation Capture (3C) assay.

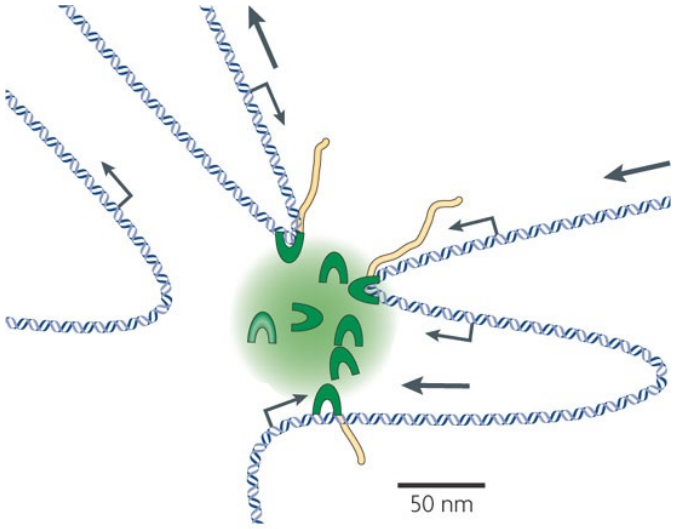

FIGURE 2.6: Model of transcription with RNA polymerase immobilized in transcription factories [34]. A transcription factory with a diameter of ∼70 nm that contains eight RNA polymerases II (green crescents). Gene loci are wrapped to these polymerases (in the direction of large arrows) and as they are transcribed, the nascent RNA (yellow) comes out. Small arrows indicate the direction of transcription at the transcription start site.

2.3.1 Microscopy-based assays

One of the most import microscopy-based techniques is Fluorescence In-Situ Hybridisation (FISH) (Figure 2.7). It detects nucleic acid sequences by using a fluorescent probe, usually between 15 bp and 30 bp in length, that hybridizes specifically to its complementary target sequence within the intact cell [36]. This allows for a targeted visualization of loci at a maximum optical resolution of about 200 nm.

Several variations of FISH have been adapted for various purposes. 3D FISH allows three-dimensional visualization of specific DNA and RNA targets within the nucleus at all stages of the cell cycle. It provides information about the arrangement of chromosome territories and the organization of sub-chromosomal domains, about positions of individual genes and RNA transcripts read from them. Accumulation of such data is necessary for understanding relationships between the spatial organization of the genome and its functioning in the interphase nucleus [37].

Recently, new approaches have created unprecedented new possibilities to investigate the structure and function of cells. Since the first studies of biological structures by early pioneers of microscopy like Robert Hooke and Antoni van Leeuwenhoek in the 17^{th} century, technical developments and improved manufacturing have led to greatly improved image quality but were ultimately faced by a limit in optical resolution. Still, even with perfect lenses, optimal alignment, and large numerical apertures, the optical

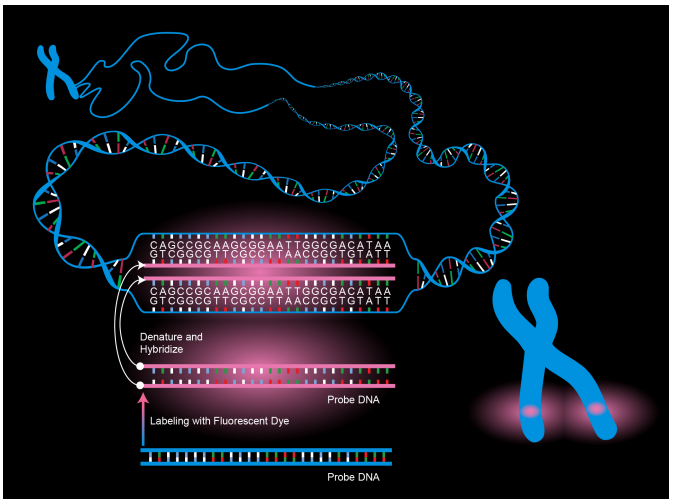

FIGURE 2.7: Fluorescence In-Situ Hybridization (FISH): experimental protocol overview. Laboratory technique for detecting and locating a specific DNA sequence on a chromosome. The technique relies on exposing chromosomes to a small DNA sequence probe, that has a fluorescent molecule attached to it. The probe sequence binds to its corresponding sequence on the chromosome. Source: National Human Genome Research Institute (NHGRI)

resolution of light microscopy was limited to approximately half of the wavelength of the light used. In practical terms, this meant that only cellular structures and objects that were at least 200 to 350 nm apart could be resolved by light microscopy. Much of the fundamental biology of the cell, however, occurs at the level of macromolecular complexes in the size range of tens to few hundreds nm, that is beyond the reach of conventional light microscopy. Super-resolution fluorescent microscopy [38] is able to bypass the diffraction limit of traditional optical microscopes, bringing the resolution potentially up to around 20 nm. This technique is based on stimulating a sample with a series of sinusoidal striped patterns of high spatial frequency, typically generated by laser light passing through a movable optical grating and projected via the objective onto the sample [39]. Then, the response of the fluorophores to excitation is exploited and, by applying the excitation patterns in different orientations and processing all acquired results using computer algorithms, a high-resolution image of the underlying structure can be generated [38]. The multicolor capability of this technique could potentially allow the imaging of the structure of chromatin at high resolution, by labeling short DNA sequences with different colors and then matching the color sequence with the known reference.

2.3.2 Chromosome Conformation Capture and 3C-based assays

Chromosome Conformation Capture (3C) is a high-throughput methodology, which can be used to analyze the overall spatial organization of chromosomes and to investigate

their physical properties at high resolution [40]. The main steps of the protocol are shown on Figure 2.8.

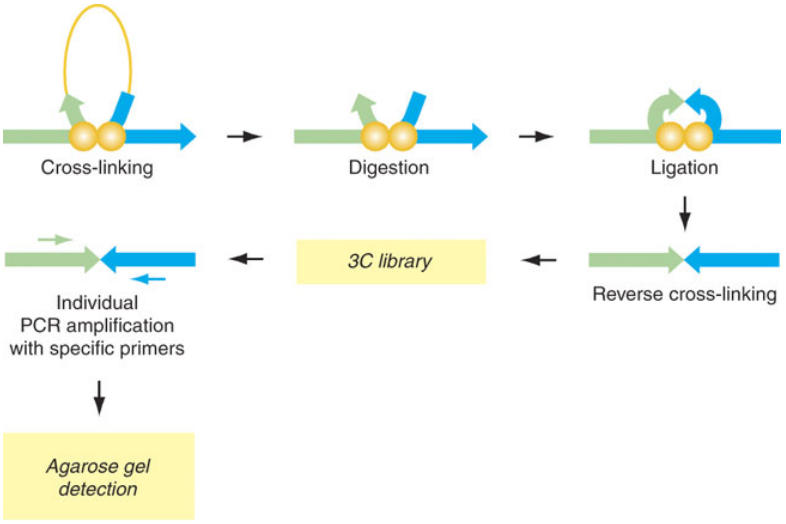

FIGURE 2.8: Schematic representation of the Chromosome Conformation Capture (3C) methodology [41].

Intact nuclei are isolated and subjected to formaldehyde fixation, which cross-links proteins to other proteins and to DNA. The overall result is cross-linking of physically touching segments throughout the genome via contacts between their DNA-bound proteins. Now, cross-linked DNA is digested with a restriction enzyme and then subjected to ligation at very low DNA concentration. Under such conditions, ligation of cross-linked fragments, which is intramolecular, is strongly favored over ligation of random fragments, which is intermolecular. At this point, chromatin parts that were physically in proximity are joined. Cross-linking is then reversed and individual ligation products are detected and quantified by the Polymerase Chain Reaction (PCR) using locus-specific primers. The result is a 3C library composed of stretches of DNA combining two fragments from two distinct genomic locations, that were spatially closed in the three-dimensional conformation.

While useful for specific loci of interest, 3C has very limited throughput. Circularized Chromosome Conformation Capture (4C) [42] is a modification of 3C where the 3C library is cut with a secondary restriction enzyme, and the resulting fragments are circularized with a ligase enzyme and then amplified using Inverse Polymerase Chain Reaction. The main advantage is that the sequence of only one site of interest needs to be known, so this PCR is able to capture all interactions of a single site (one-by-all relationship). The 4C library is then sent to high-throughput sequencing or hybridised with DNA microarrays to make further data analyses [43].

While 4C allows investigation of many unknown interacting sequences, it is still limited in terms of throughput, since only one input sequence can be used per experiment.

Carbon Copy Chromosome Conformation Capture (5C) [44] expands on 3C by allowing parallel analysis of the interaction between many selected loci (many-by-many relationship). After generation of a 3C library, multiplex ligation mediated amplification (LMA) is used to generate a 5C library, which can be then analyzed by microarray hybridization or high-throughput sequencing. LMA is a variation of PCR that permits multiple targets to be amplified with only a single primer pair [45]. Each probe consists of two oligonucleotides, that hybridise to adjacent sites of the target sequence. Specificly, all forward primers feature a common 5'-end tail containing the T7 sequence (TAATACGACTCACTATAGCC), while all reverse primers contain a common 3'-end tail featuring the complementary T3 sequence (ATAATTGGGAGTGATTTCCCT). In this way, all ligated probes have identical end sequences, allowing simultaneous PCR amplification using only one primer pair. In addition, these oligonucleotides can be ligated into a complete probe only when they both are hybridised to their respective targets, with the advantage that only the ligated oligonucleotides, but not the unbound probe oligonucleotides, are amplified. Conversely, if the probes were not split in this way, the primer tails would cause the probes to be amplified regardless of their hybridization to the target DNA, and amplification of probes would not depend on the number of target sequences in the sample.

Another method to explore chromatin interactions is Chromatin Interaction Analysis using Paired-End Tag sequencing (ChIA-PET) [46]. Combining Chromatin Immunoprecipitation (ChIP), proximity ligation and Paired-End Tag (PET) strategy, ChIA-PET provides a global and unbiased interrogation of higher-order chromatin structures associated with certain protein factors, to address the functional relationships between specific subsets of interacting DNA loci. The outcome is functional chromatin structure converted into millions of short tag sequences, resulting in a ChIA-PET library for sequencing analysis. Often the target is RNA Polymerase II, which allows to map the DNA-DNA interactions of actively transcribed genes, but it is also possible to study specific transcriptional regulatory elements (TREs) marked by certain chromatin signatures. Using ChIA-PET, Iouri Chepelev *et al.* [47] defined a global view of enhancer-promoter interactions using H3K4me2 as active enhancer mark. The same approach can be applied to other histone modifications, such as acetilation [48], phosphorylation [49] and ubiquitination [50] to study their individual and collective contribution to 3D chromatin function.

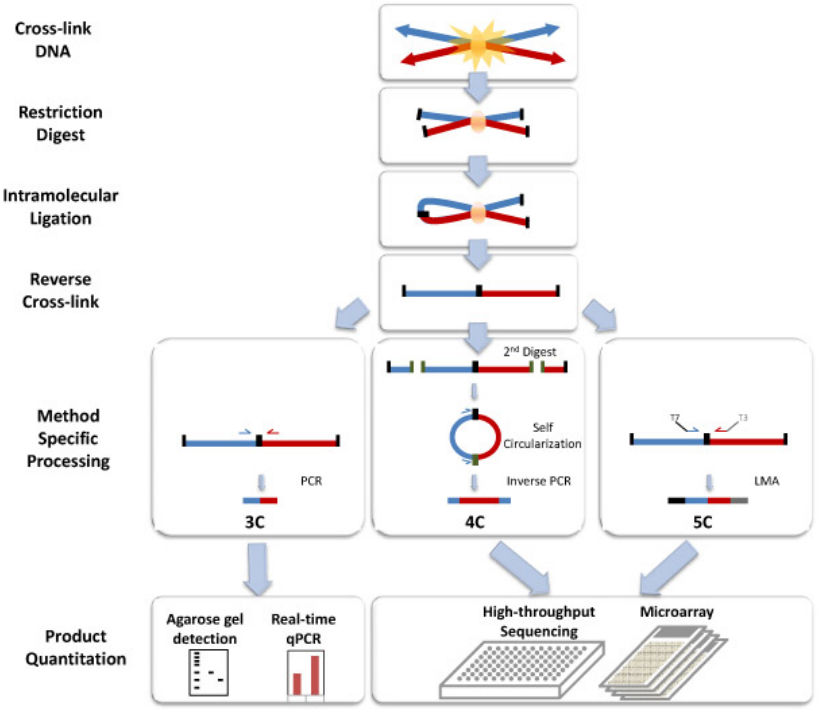

FIGURE 2.9: A comparison of methods used for chromosome conformation capture: 3C, 4C and 5C. While all the methods rely on cross-linking DNA, restriction enzyme digestion and ligation under dilute conditions, 3C analyzes the interaction between two individual loci by PCR, 4C analyzes all loci that interact with one locus by inverse PCR followed by microarray or high-throughput sequencing, and 5C analyzes many parallel interactions by generating a library by amplification with universal primer tags and analysis by microarray or high-throughput sequencing.

2.4 Summary

In this chapter, I presented an overview of the three-dimensional chromatin structure and its role in the epigenetic transcriptional regulation. Then, I introduced the experimental assays to investigate the three-dimensional architecture of genomes, divided into microscopy-based assays (FISH) and Chromosome Conformation Capture (3C) based assays.

In the next chapter I will focus on Hi-C, the most recent and important 3C based assay, which first allowed for a comprehensive three-dimensional genome analysis. I will describe either the experimental protocol and the data computational analysis. Such analysis will then be implemented in Chapter 4, where I present HiCtool, a standardized pipeline that we developed for Hi-C data processing and visualization.

Chapter 3

Hi-C: 3C genome-wide

As seen in the first chapter, long-range interactions between specific pairs of loci can be evaluated with Chromosome Conformation Capture (3C) assay, using spatially constrained ligation followed by locus-specific Polymerase Chain Reaction (PCR). Adaptations of 3C have extended the process with the use of inverse PCR (4C) or multiplexed ligation-mediated amplification (5C). Still, these techniques require choosing a set of target loci and do not allow unbiased genome-wide analysis. To overcome these difficulties in 2009 a genome-wide (all-to-all) version of 3C was developed, termed Hi-C, that probes the three-dimensional architecture of whole genomes, by coupling proximity-based ligation with parallel sequencing [4].

3.1 Hi-C protocol

The Hi-C protocol is identical to 3C up to the restriction enzyme digestion step. The process involves cross-linking of cells using formaldehyde, followed by DNA digestion with a restriction enzyme that leaves 5' overhangs, which are filled in with biotinylated residues. The resulting blunt-end fragments are ligated under dilute conditions to favor ligation events between the cross-linked DNA fragments. The resulting DNA sample contains ligation products consisting of fragments that were originally in close spatial proximity in the nucleus, marked with biotin at the junction. The ligated DNA is then sheared and the biotin-containing fragments are selected using streptavidin beads to yield a library of fragments containing sequences from interacting loci. The library is then analyzed by using massively parallel DNA sequencing, producing a catalog of interacting fragments [4]. In contrast to ChIA-PET, Hi-C is site-neutral, reporting interactions between any pair of close spatial proximity loci in the genome. The entire

process is shown in Figure 3.1.

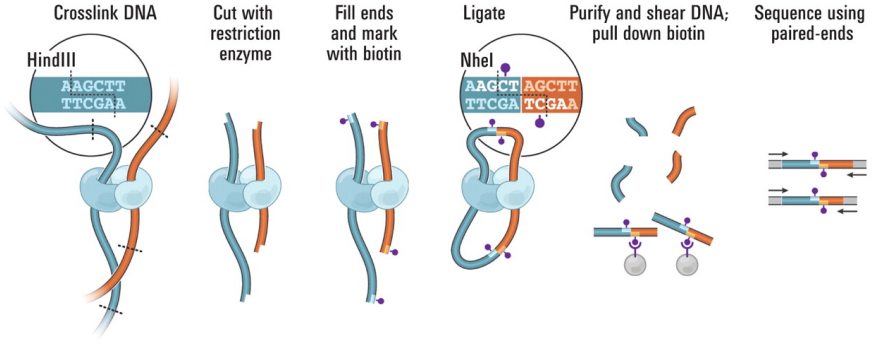

FIGURE 3.1: Overview of the Hi-C process [4]. Cells are cross-linked with formaldehyde, digested with a restriction enzyme, the 5' overhang is filled with a biotinylated residue, blunt-end fragments are ligated under dilute conditions, DNA fragments are sheared and selected with streptavidin beads. The library containing proximity-ligated fragments is analyzed by paired-end high throughput sequencing.

3.2 Hi-C computational analysis

The first step in analyzing Hi-C sequencing data is mapping the paired-end sequence reads on the genome. Then, several kinds of analysis can be carried out like contact frequency analysis, correlation analysis applied to the contact matrices and Principal Component analysis of genomic interactions. Finally, a three-dimensional polymer model of the chromatin structure is derived.

3.2.1 Paired-reads alignment

The process of aligning sequence reads to the genome is becoming a relatively well-established process and there are many programs available for this, like MAQ [51] or Bowtie 2 [52]. In Lieberman-Aiden *et al.* (2009), each end of the paired reads was aligned separately against the human hg18 reference sequence with MAQ, using a mismatch threshold of 150 [4]. For Hi-C data, each of the paired reads should align to the genome to include the sequence to the interaction library, since the goal is to analyze the interaction between these two genomic regions.

The next step is quality control to ensure that the aligned reads are the result of proximity ligation of digested fragments, so they are likely to reflect long range chromatin interactions rather than just random collisions [53]. It should be confirmed that reads align significantly closer to the restriction enzyme sites (HindIII in this case) as compared to randomly generated reads. In Lieberman-Aiden *et al.* (2009) the maximum

fragment size used for sequencing is 500 bp [4]. As shown in Figure 3.2 (A), both the intrachromosomal reads and interchromosomal reads curves decrease rapidly as the distance from the HindIII site increases, until a plateau is reached at a distance of ∼500 bp. The sequence of reads should also be in the correct orientation with respect to the restriction site. Hi-C sequences are expected to point (5'-3') in the direction of the ligation junction and therefore should align in the linear genome to the 3' end of HindIII restriction fragments. This tendency is reflected in ∼80% of reads from both intrachromosomal and interchromosomal interactions, that align near HindIII restriction sites with the correct orientation as shown in Figure 3.2 (B).
Quality control can be performed also for the percentage of reads that map to intrachromosomal and interchromosomal interactions. In a successful experiment, 55% of the alignable read pairs represent interchromosomal interactions, 15% represent intrachromosomal interactions between fragments less than 20 kb apart and 30% are intrachromosomal read pairs that are more than 20 kb apart (Figure 3.2 C) [53].

3.2.2 Contact matrix and correlation analysis

A genome-wide contact matrix M can be constructed by dividing the genome into appropriately sized regions ("loci") and defining each entry m_{ij} to be the number of ligation products between locus i and locus j. This matrix can be visually represented as a heatmap, with intensity indicating the number of contacts (Figure 3.3).
Figure 3.3 shows also the reproducibility of Hi-C results, repeating the experiment using the same initial restriction enzyme (RE) HindIII and then using NcoI. It was observed that the contact matrices generated were very similar to the original contact matrix showing Pearson's $r = 0.990$ (HindIII) and $r = 0.814$ (NcoI) [4].

It was also tested if Hi-C data were consistent with known features of genomic organization, specificly chromosome territories, so distant loci in the same chromosome that are close in space, and patterns in subnuclear position, that is the tendency of certain chromosome pairs to be near one another. The average intrachromosomal contact probability $I_n(s)$ for pairs of loci at genomic distance s on each chromosome n was calculated. $I_n(s)$ decreases monotonically on every chromosome, suggesting that the 3D distance between loci increases with increasing of genomic distance [4]. Also, the average interchromosomal contact probability was computed. In this case, the number of interactions between loci on a pair of chromosomes was divided by the number of possible interactions between the two chromosomes, that is the product of the number of loci on each chromosome. The result shows that even at distances greater than 200

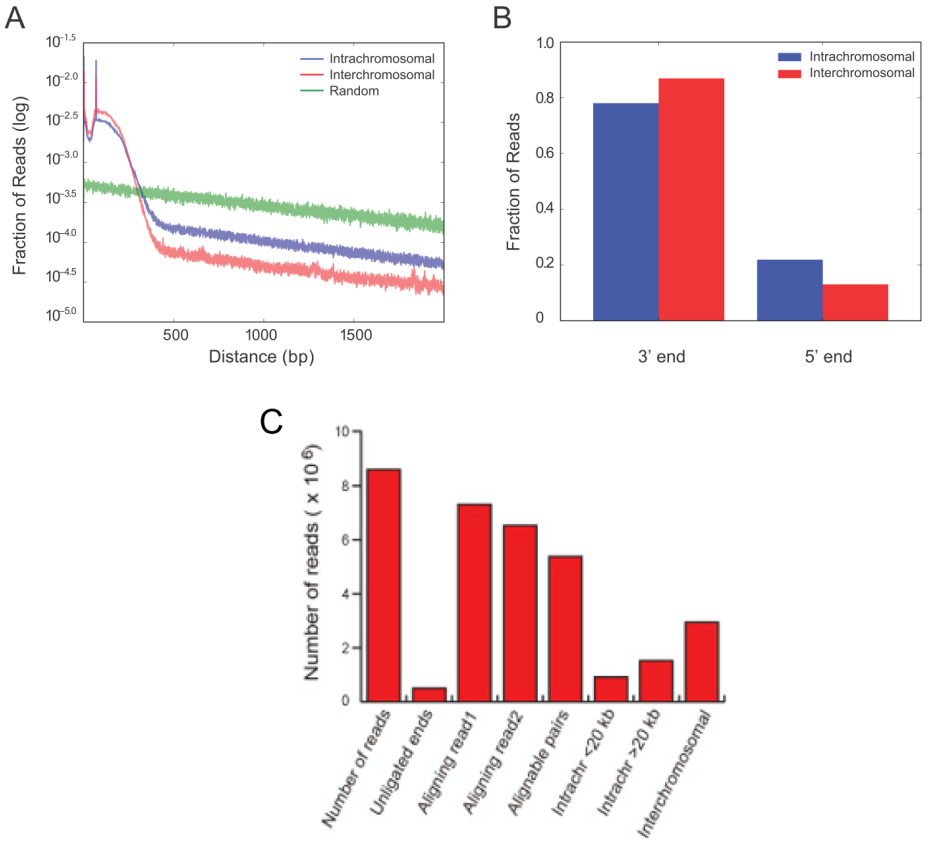

FIGURE 3.2: Hi-C quality control process [4]. **(A)** Reads from fragments corresponding to both intrachromosomal (blue) and interchromosomal (red) interactions align significantly closer to HindIII restriction sites as compared to randomly generated reads (green). **(B)** Orientation of aligned reads with respect to the restriction site. ∼80% of reads from both intrachromosomal (blue) and interchromosomal (red) align to the 3' end of HindIII restriction fragments. **(C)** Percentage of reads that map to intrachromosomal and interchromosomal interactions.

Mb, $I_n(s)$ is always much greater than the average contact probability between different chromosomes (Figure 3.4 A), implying the existence of chromosome territories.

Figure 3.4 (B) displays observed/expected interchromosomal number of contacts between each pair of chromosomes. The expected number of interchromosomal contacts for each chromosome pair i,j was computed by multiplying the fraction of interchromosomal reads containing i with the fraction of interchromosomal reads containing j and multiplying by the total number of interchromosomal reads. The resulting heatmap shows that small, gene-rich chromosomes (like chromosomes 16, 17, 19, 20, 21 and 22) preferentially interact with each other, while chromosome 18, which is small but gene-poor, does not interact frequently with the other chromosomes [4]. This agrees with FISH studies in [54], [55] and [56].

As seen above, sequence proximity strongly influences contact probability, therefore a

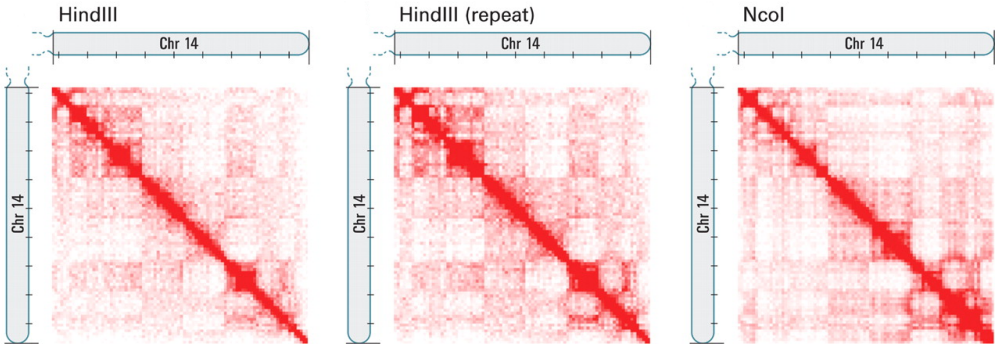

FIGURE 3.3: Hi-C genome-wide contact matrices [4]. The matrices shown here correspond to intrachromosomal interactions on chromosome 14 (chromosome 14 is acrocentric; the short arm is not shown). Each pixel represents all interactions between a 1-Mb locus and another 1-Mb locus; intensity corresponds to the total number of reads (0 to 50). The original experiment is compared also with results from a biological repeat using the same restriction enzyme (HindIII, range from 0 to 50 reads) and with results using a different restriction enzyme (NcoI, range from 0 to 100 reads).

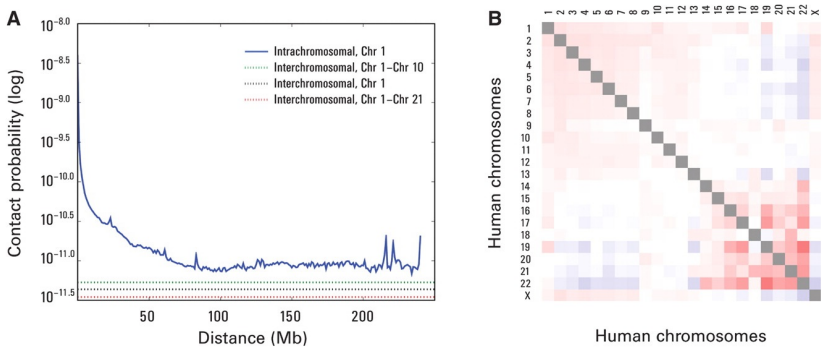

FIGURE 3.4: The presence and organization of chromosome territories [4]. **(A)** Probability of contact decreases as a function of genomic distance on chromosome 1, eventually reaching a plateau at ∼90 Mb (blue). The level of interchromosomal contact (black dashes) differs for different pairs of chromosomes; loci on chromosome 1 are most likely to interact with loci on chromosome 10 (green dashes) and least likely to interact with loci on chromosome 21 (red dashes). Interchromosomal interactions are depleted relative to intrachromosomal interactions. **(B)** Observed/expected number of interchromosomal contacts between all pairs of chromosomes. Red indicates enrichment, and blue indicates depletion (range from 0.5 to 2). Small, gene-rich chromosomes tend to interact more with one another, suggesting that they cluster together in the nucleus.

normalized contact matrix M^* was generated, in which each entry m^*_{ij} is the ratio between the actual number of reads between loci i and j and the number of expected reads at at the genomic distance s between i and j [4]. This observed (Figure 3.5 A) over expected (Figure 3.5 B) matrix can be displayed as a heatmap, that shows many large blocks of more (red) or less (blue) interactions than expected, generating a plaid pattern (Figure 3.5 C). This plaid pattern suggests that the nucleus is segregated into two compartments corresponding to open and closed chromatin.

Further statistical analysis of the data can lead also to the computation of a correlation

matrix. If two loci are nearby in space, it is reasonable that they share neighbors and have correlated interaction profiles. Therefore, a correlation matrix C can be computed, in which each entry c_{ij} is the Pearson correlation between the ith row and the jth column of M^* [4]. This process sharpened the plaid pattern and enhanced the presence of two compartments within the chromosome (Figure 3.5 D).

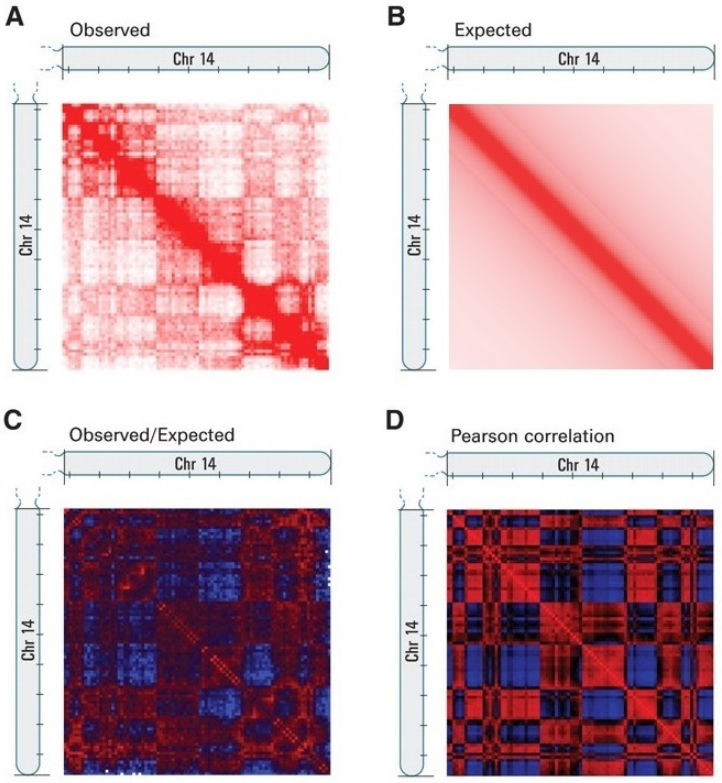

FIGURE 3.5: Heatmaps of chromosome 14 at a resolution of 1 Mb [4]. **(A)** Heatmap based on observed interactions exhibits substructure in the form of an intense diagonal and a constellation of large blocks (range from 0 to 200 reads). Tick marks appear every 10 Mb. **(B)** Heatmap of the expected interaction frequencies based on genomic distance, corresponding to what would be observed if there were no long-range structures. **(C)** The observed/expected matrix shows loci with either more (red) or less (blue) interactions than would be expected, given their genomic distance (range from 0.2 to 5). **(D)** Pearson correlation matrix illustrates the correlation [range from -1 (blue) to +1 (red)] between the intrachromosomal interaction profiles of every pair of 1-Mb loci along chromosome 14.

3.2.3 Principal Component Analysis

Principal Component Analysis (PCA) was used to partition a chromosome into two sets of loci, in which contacts are enriched within each set and depleted between sets [4]. For all but two chromosomes, the first principal component clearly corresponds to the plaid pattern, positive values defining one set (A) and negative values the other (B). For

chromosomes 4 and 5, the first PC corresponds to the two chromosome arms, the second PC to the plaid pattern. The entries of the PC vector reflect the sharp transitions from A to B observed within the plaid heatmaps.

In figure 3.6 four loci on chromosome 14 (L1, L2, L3, L4), that alternate between the two compartments (L1 and L3 in compartment A, L2 and L4 in compartment B), are taken. It is shown that L1 tends to be closer to L3 than L2, despite the fact that L2 is between L1 and L3 in the linear genome sequence (Figure 3.6 A). For the other couple of loci, L2 and L4, the same can be asserted: L4 tends to be closer to L2 than L3 (Figure 3.6 B). To probe these results, a 3D FISH analysis has been used as well (Figure 3.6, bottom).

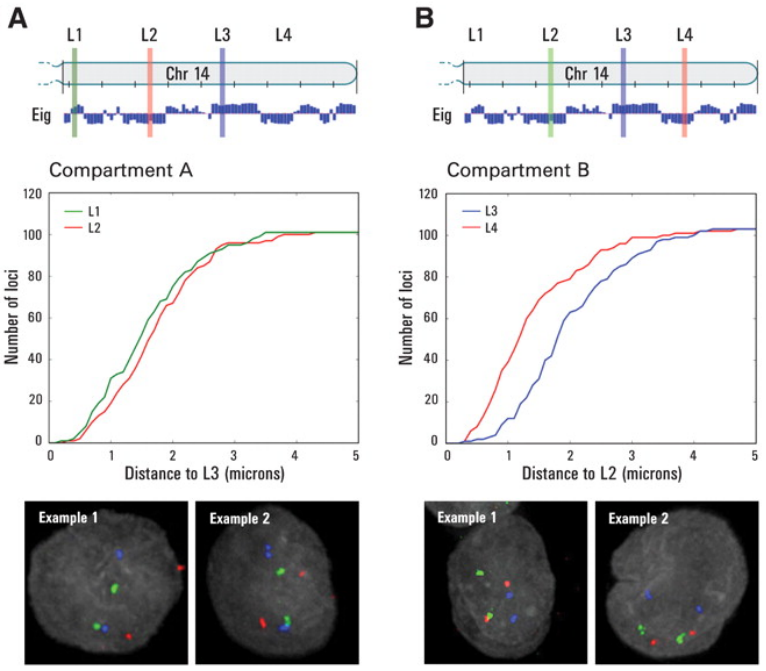

FIGURE 3.6: Principal Component Analysis on chromosome 14 [4]. **(A)** Compartment A analysis: L1 tends to be closer to L3 than L2. **(B)** Compartment B analysis: L4 tends to be closer to L2 than L3. **(Bottom)** 3D FISH experiment results are displayed.

Upon further examination it was found that, at a given genomic distance, pairs of loci belonging to compartment B showed a higher interaction frequency than loci in compartment A, suggesting that compartment B is more densely packed [4], in accord with J. Dekker (2008) [57], who asserted that chromatin fibre exhibits a local variation in compaction. To prove this, it was seen that compartment A correlates strongly with the presence of genes and higher expression, suggesting that it is closely associated with open, accessible and actively transcribed chromatin. These results demonstrate that open and closed chromatin domains occupy different spatial compartments in the nucleus [4].

3.2.4 Three-dimensional chromatin structure modeling

Chromosomal regions can be simulated by three-dimensional polymer models, where the polymer chain of each chromosome is arranged as a highly knotted configuration forming an "equilibrium globule" [58] (Figure 3.7 A, middle).
An alternative model was also proposed, describing the three-dimensional configuration of the chromatin as a "fractal globule" [59] (Figure 3.7 A, bottom). This state is formed by a polymer that folds into a series of small globules, and then again several times creating superior order structures, until a single "globule of globules" remains. The resulting structure resembles a Peano curve [60], a continuous fractal trajectory that fills the 3D space without crossing itself. A fractal globule results as an attractive structure for chromatin segments because it lacks knots and this facilitates the unfolding and refolding, for example during cell cycle, gene activation or gene repression [4].

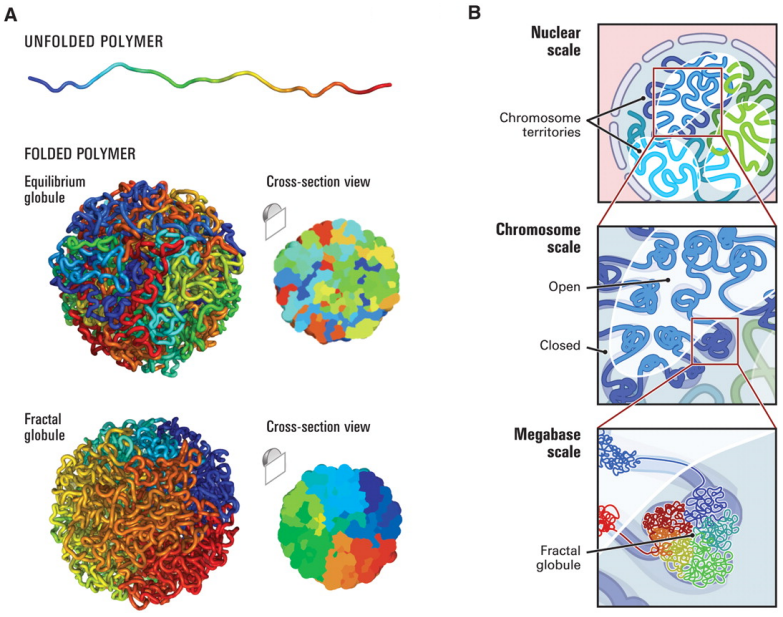

FIGURE 3.7: Three-dimensional chromatin models [4]. **(A)** 3D polymer models. (Top) An unfolded polymer chain, 4000 monomers (4.8 Mb) long; coloration corresponds to distance from one endpoint, ranging from blue to cyan, green, yellow, orange and red. (Middle) An equilibrium globule: loci that are nearby along the contour need to not be nearby in 3D. (Bottom) A fractal globule: nearby loci along the contour tend to be nearby in 3D, leading to monochromatic blocks both on the surface and in the cross section. This structure lacks knots. **(B)** Genome architecture at three scales. (Top) Nuclear scale: chromosomes (blue, cyan and green) occupy different territories. (Middle) Chromosome scale: each chromosome folds back and forth between the open and closed chromatin compartments. (Bottom) Megabase scale: the chromosome consists of fractal globules.

Polymer model structures influence the average behavior of intrachromosomal contact probability as a function of genomic distance. Lieberman-Aiden *et al.* [4] observed that, plotted on log-log axes, $I(s)$ exhibited a power law scaling between ~500 kb and ~7

Mb, where contact probability scaled as $s^{-1.08}$ (Figure 3.8 A). This was in agreement with simulations, that showed a slope of -0.993 for the contact probability of a fractal globule, while an equilibrium globule model predicted that the contact probability scaled as $s^{-1.508}$, which was not observed in the data (Figure 3.8 B).

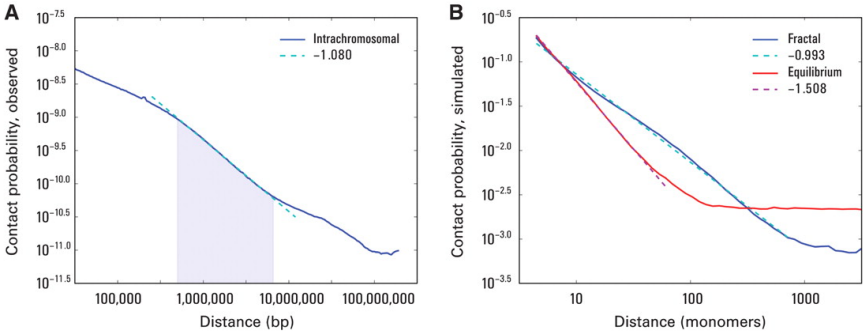

FIGURE 3.8: Contact probability as a function of genomic distance averaged across the genome [4]. **(A)** Observed contact probability shows a power law scaling with a slope of -1.08 between 500 kb and 7 Mb. **(B)** Simulation results for equilibrium (red) and fractal (blue) globules, with a slope respectively of around -3/2 and -1.

3.3 Topological Domains in mammalian genomes

Investigation of the three-dimensional organization of human and mouse genomes by performing Hi-C experiment, led to the identification of large megabase-sized chromatin interaction domains, which were called Topological Domains [16]. This study was performed in mouse embryonic stem (ES) cells, human ES cells and human IMR90 fibroblast, and over 1.7-billion read pairs of Hi-C data were analyzed.

In particular, an analysis of two-dimensional interaction matrices at a bin size less than 100 kb, revealed the presence of these highly self-interacting regions, seen as "triangles" on the heatmaps (Figure 3.9 a). These regions are bounded by segments where the interactions end suddenly, which were called topological boundary regions or unorganized chromatin, depending on their size (the first smaller than the second). To identify systematically all topological domains in the genome, the Directionality Index (DI) was devised [16]. Regions at the periphery of topological domains are highly biased in their interaction frequencies: the most upstream portion of a topological domain is highly biased towards interacting downstream, while the downstream portion is highly biased towards interacting upstream (Figure 3.9 a). Therefore, DI was used to quantify the degree of upstream or downstream interaction bias for each genomic region, using this formula:

$$DI = \left(\frac{B-A}{|B-A|}\right)\left(\frac{(A-E)^2}{E} + \frac{(B-E)^2}{E}\right) \tag{3.1}$$

where, A is the the number of reads that map from a given 40 kb bin to the upstream 2 Mb, B is the the number of reads that map from a given 40 kb bin to the downstream 2 Mb and E is the expected number of contacts for each bin and it equals $\frac{A+B}{2}$. Also a Hidden Markov Model (HMM) based on directionality index can be used to identify biased "states", so to determine the "true" hidden directionality bias (Figure 3.9 a). Furthermore, a subset of boundaries appear to mark the transition between A and B compartment in Lieberman-Aiden *et al.* (2009) [4] (Figure 3.9 e). Generally the A and B compartments are larger than the topological domains, with a median value for the size respectively of 3 Mb and ∼880 kb [16].

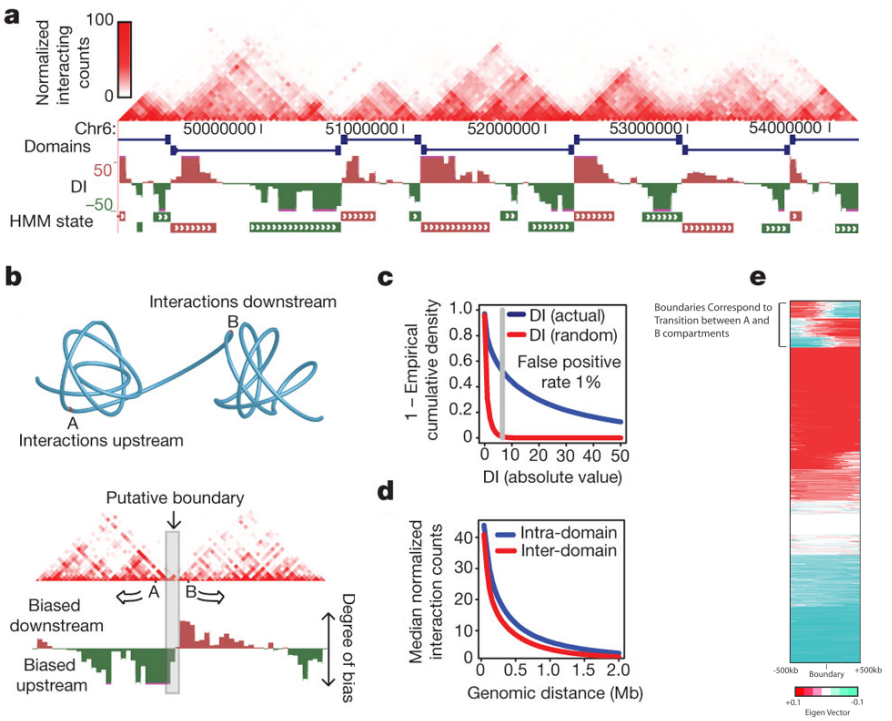

FIGURE 3.9: Topological domains in the mouse ES cells genome [16]. **(a)** Normalized Hi-C interaction frequencies for chromosome 6 displayed as a two-dimensional heatmap, domains, Directionality Index (DI) and HMM bias states. **(b)** Schematic illustration of two topological domains and resulting directionality index. **(c)** Distribution of the directionality index (absolute value, in blue) compared to random (red). 52% of the genome exhibits a directionality index that was not expected by random chance. **(d)** Mean interaction frequencies at all genomic distances between 40 kb and 2 Mb. Above 40 kb, the intra- versus inter-domain interaction frequencies are significantly different. **(e)** Comparison of A and B compartments with topological domains using a heatmap of the Eigen Vector values used to determine A and B sets at a boundary region.

3.3.1 Topological Domains and transcriptional control process

The relationship between topological domains and the transcriptional control process was also investigated. A strong enrichment of CTCF at the boundaries of topological domains was seen, and it was hypothesized that topological boundaries might exhibit insulator or barrier elements behavior [16], since many insulators are bound by CTCF [61]. Although most topological boundaries are enriched for CTCF, only the 15% of CTCF sites are located in boundary regions. This means that CTCF alone is insufficient to identify domain boundaries [16]. Another analysis was related to the distribution of the heterochromatin mark H3K9me3, that is a well known boundary element [62], and it was observed a clear segregation of H3K9me3 around the boundary regions. Further examinations showed that factors associated with active promoters and gene bodies, TSSs and GRO-seq signal were enriched at the boundaries in both mouse and human, while non-promoter-associated marks H3K4me1 (associated with enhancers) and H3K9me3 were respectively not enriched or depleted at boundary regions (Figure 3.10 a). Additionally, tRNA genes, which can function as boundary elements, are enriched at the boundaries, suggesting that a high level of transcription activity may contribute to the boundary formation [16]. Thus, the above observations suggest a potential correlation between topological boundary regions and insulators or barrier elements, revealing a link between topological domains and transcriptional control process.

3.3.2 Boundaries are shared across cell types and conserved in evolution

A comparison between replicates of mouse ES cells and cortex or between human ES cells and IMR90 cells showed that most of the boundary regions are shared between cell types (Figure 3.11 a), suggesting that the global domain structure is largely unchanged [16]. This stability of the domains between cell types led to investigate if the domain structure was invariant across evolution as well. Also in this case, a comparison between domain boundaries of mouse and human ES cells showed that most of the boundaries are shared across evolution (Figure 3.11 c), in particular 53.8% of human boundaries are boundaries in mouse and 75.9% of mouse boundaries are boundaries in human, compared to 21.0% and 29.0% at random [16]. Figure 3.11 (d) shows a high degree of similarity between the domain structure over a syntenic region in the mouse and human ES cells, indicating that there is also a conservation of the higher order structure of DNA.

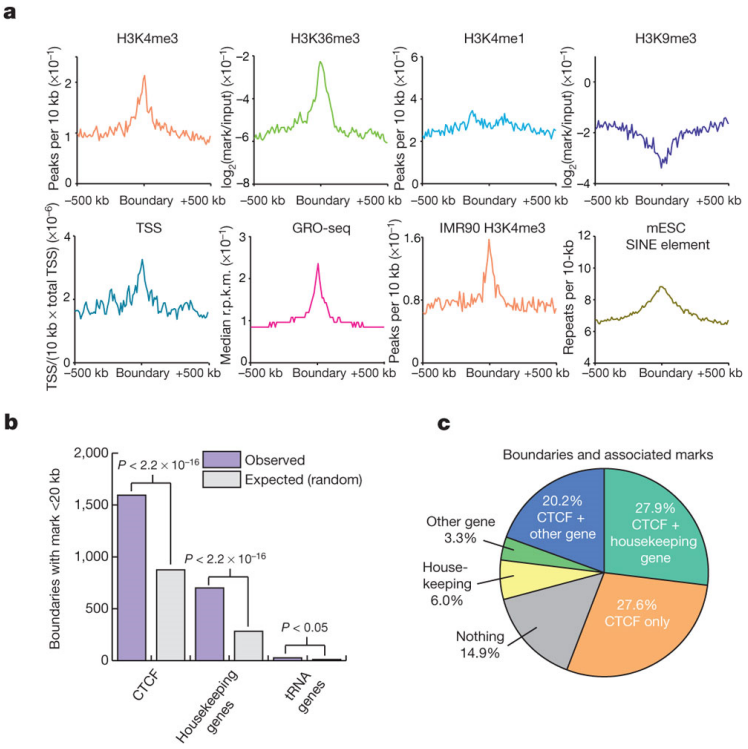

FIGURE 3.10: Factors that may contribute to the formation of topological boundary regions [16]. **(a)** Elements surrounding boundary regions in mouse ES cells or IMR90 cells. **(b)** Boundaries associated with a CTCF binding site, housekeeping gene and tRNA gene (purple) compared to the expected at random (grey). **(c)** Pie chart showing the percentage of boundaries associated with a given mark within 20 kb of the boundaries.

3.4 In situ Hi-C reveals principles of chromatin looping

In 2014, an updated Hi-C protocol called "in situ Hi-C" was developed, to comprehensively map chromatin contacts genome-wide, generating Hi-C maps at unprecedented high resolution. The densest map, in human lymphoblastoid cells, contained 4.9 billion contacts, at 1 kb resolution. Using these maps, it was shown that genomes are partitioned into previously undetected small contact domains, with a median length of 185 kb, that are associated with distinct patterns of histone marks and segregated into six subcompartments. About 10,000 loops were also identified, as pairs of loci that show significantly closer proximity with one another than with the loci lying between them. These loops often link promoters and enhancers, correlate with gene activation and show conservation across cell types and species [29].

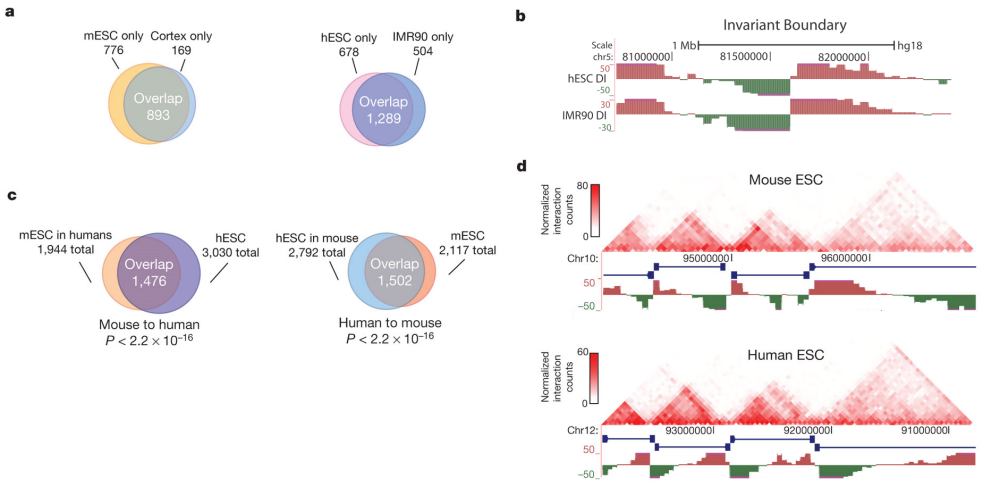

FIGURE 3.11: Boundary regions are shared across cell types and conserved in evolution [16]. **(a)** Overlap of boundaries between cell types for both humans and mouse. **(b)** An invariant boundary region between human ES cell and IMR90. **(c)** Overlap of boundaries between syntenic mouse and human sequences. **(d)** Domain structure over a syntenic region in the mouse and human ES cells.

3.4.1 In situ Hi-C protocol

In situ Hi-C protocol combines the original Hi-C protocol (called from now "dilution Hi-C") [4] with nuclear ligation assay [63]. The methodology involves DNA-DNA proximity ligation in intact nuclei, digestion using a 4-cutter restriction enzyme (like MboI), filling the 5' overhangs including a biotinylated residue, ligating the resulting blunt-end fragments, shearing the DNA, capturing the biotinylated ligation junctions with streptavidin beads and analyzing the resulting fragments with paired-end sequencing (Figure 3.12 A). The updated protocol showed three principal advantages over dilution Hi-C. First, using in situ ligation, the frequency of spurious contacts due to random ligation in dilute solution is reduced. Second, the protocol is faster, requiring 3 days instead of 7. Third, the use of a 4-cutter instead of a 6-cutter enables a more efficient cutting of chromatinized DNA and higher resolution.

3.4.2 Small contact domains with a median length of 185 kb were detected

In Lieberman-Aiden *et al.* (2009) [4] heatmaps at a resolution of 1 Mb were computed, showing the presence of large squares (5-20 Mb) of higher contact frequency along the diagonal, which were called "megadomains". In addition, each locus of 1 Mb could be assigned to one of two compartments (previously called A and B), with loci showing a higher interaction frequency if they belonged to the same compartment than if they did

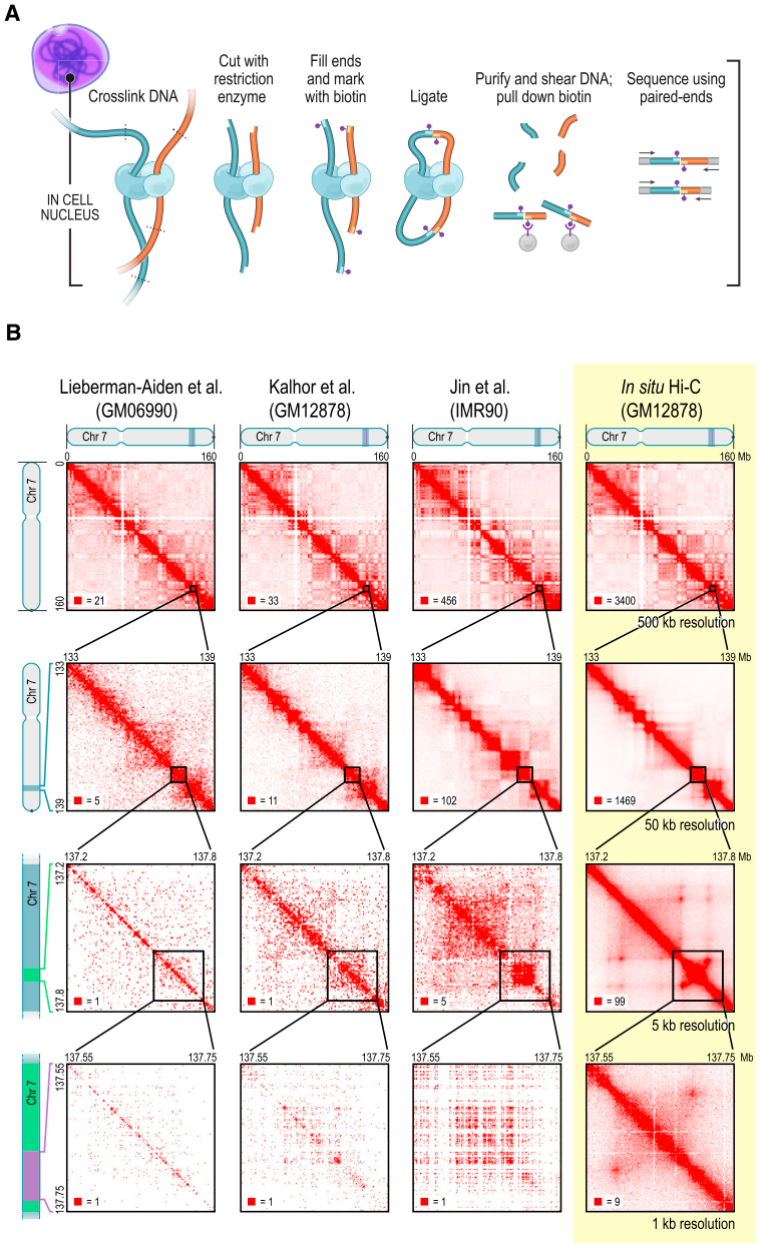

FIGURE 3.12: In situ Hi-C protocol [29]. **(A)** In situ Hi-C protocol combines the original Hi-C protocol with nuclear ligation assay. During in situ Hi-C, DNA-DNA proximity ligation is performed in intact nuclei. **(B)** Comparison of map of chromosome 7 in GM12878 (last column) to earlier Hi-C maps: Lieberman-Aiden *et al.* (2009) [4], Kalhor *et al.* (2012) [64] and Jin *et al.* (2013) [65]. These heatmaps show that the protocol facilitates the generation of much denser Hi-C maps.

not. By using the new higher resolution maps, many small squares of enriched contacts along the diagonal were discovered, whose length was from 40 kb to 3 Mb, with a median size of 185 kb (Figure 3.13 A). Loci within a contact domain showed correlated histone modifications for eight different factors (H3K36me3, H3K27me3, H3K4me1, H3K4me2, H3K4me3, H3K9me3, H3K79me2 and H4K20me1) [29] (Figure 3.13 B). Conversely, loci at similar distance but belonging to different domains showed much less correlation in

chromatin state.
Next, loci were partitioned into categories based on long-range patterns alone, using a manual annotation approach and three different unsupervised clustering algorithms (HMM, K-means and Hierarchical). At a resolution of 25 kb, at least five subcompartments were detected, both within and between chromosomes (Figure 3.13 C) [29]. Two of the five subcompartments are correlated with loci in compartment A, so they were labeled A1 and A2. Both these are gene dense, have highly expressed genes, hold activating chromatin marks such as H3K36me3, H3K79me2, H3K27ac, H3K4me1 and are depleted at the nuclear lamina and at nucleolus-associated domains (NADs). The other three patterns were labeled B1, B2 and B3, since they are correlated with loci in compartment B, and they show different properties. Subcompartment B1 correlates positively with H3K27me3 and negatively with H3K36me3, indicative of facultative heterochromatin. Subcompartments B2 and B3 tend to lack all the marks listed above. Specificly, B2 is enriched at the nuclear lamina (1.8-fold) and at NADs (4.6-fold); subcompartment B3 is enriched at the nuclear lamina (1.6-fold), but strongly depleted at NADs (76-fold). Further visual examinations suggested the presence of a sixth subcompartment correlated with the compartment B, so labeled B4, that spans only 11 Mb, or 0.3% of the entire genome (Figure 3.13 D). Subcompartment B4 comprises few regions, each containing many KRAB-ZNF superfamily genes (130 of the 278 KRAB-ZNF genes in the genome, a 65-fold enrichment). These regions exhibit a strong enrichment for activating chromatin marks (H3K36me3) and heterochromatin-associated marks (H3K9me3 and H4K20me3).

3.4.3 Approximately 10,000 genome-wide loops were identified

The next step was to identify the position of chromatin loops, as pairs of loci that are close together in 3D but separated by a larger genomic distance. This was achieved by searching for "peaks" relative to the local background in Hi-C contact maps, that reflect the presence of chromatin loops, with peak loci being the anchor point for the loop (Figure 3.14 A). Most of the peaks (98%) reflected loops between loci that are at a distance less than 2 Mb. These results were reproducible across all high-resolution Hi-C maps and also conserved across cell types [29].
Next, peaks across species were compared. In particular, 2,927 high-confidence contact domains and 3,331 peaks were identified in CH12-LX mouse B-Lymphoblasts. By examining orthologus regions in GM12878 (Homo Sapiens B-lymphoblastoids), 50% of

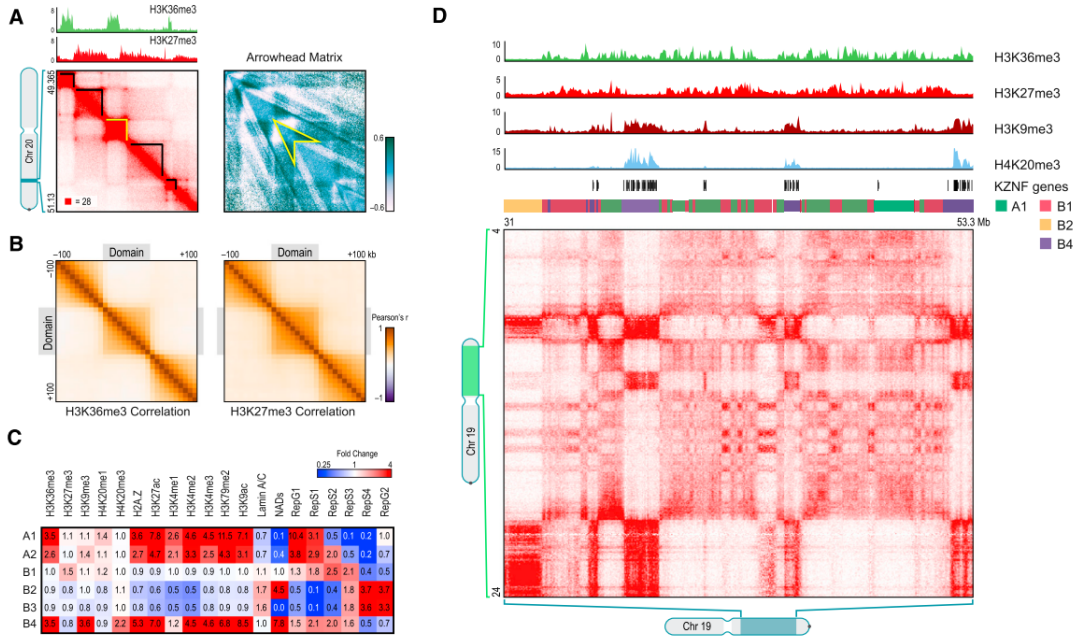

FIGURE 3.13: Genome is partitioned into small contact domains, with loci belonging to six subcompartments correlated to different patterns of histone modifications [29]. **(A)** (Left) Black highlighted contact domains on chromosome 20. (Right) Arrowhead matrix derived using a transformation that replaces domains with an arrowhead-shaped motif, pointing toward the domain's upper-left corner (example in yellow). These arrowheads were identified using dynamic programming. **(B)** Pearson correlation matrices of the histone mark signal between pairs of loci inside and within 100 kb of a domain. (Left) H3K36me3. (Right) H3K27me3. **(C)** Fold-enrichment map of the epigenetic profile of each of the six long-range subcompartments. **(D)** A large continuous region on chromosome 19 which contains intervals in subcompartments A1, B1, B2 and B4.

peaks and 45% of domains in mouse were also detected in humans, suggesting a conservation of the genome three-dimensional structure across the mammals [29] (Figure 3.14 B).

Then, the association between these loops and gene regulation was pointed out by three important findings [29]. First, peaks often present a known promoter at one peak locus and a known enhancer at the other (Figure 3.15 A), like classic enhancer-promoter loops such as at the MYC gene (chr8: 128.35-128.75 Mb, in HMEC) and α-globin (chr16: 0.15-0.22 Mb, in K562). Second, genes whose promoters are associated with a loop are much highly expressed (6-fold) than genes whose promoters are not (Figure 3.15 B). Third, cell type-specific peaks are associated with changes in expression. For example, a loop is anchored at the promoter of the gene encoding L-selectin (SELL), which is expressed in GM12878, where the peak is present, but not in IMR90, where the peak is absent (Figure 3.15 C). In total, 557 loops in GM12878 that were absent in IMR90 were observed; the corresponding peak loci overlapped the promoters of genes that are significantly upregulated (>50-fold) in GM12878, but of only one gene that was markedly upregulated in

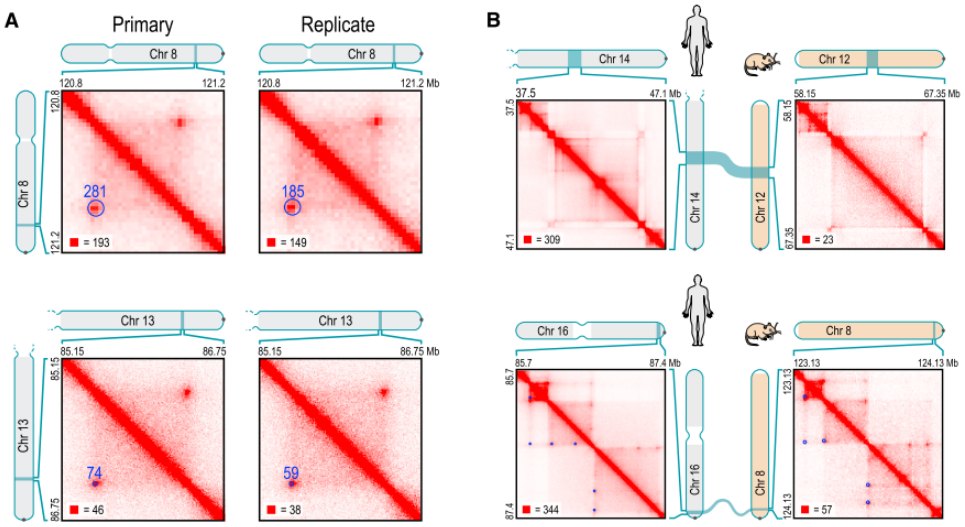

FIGURE 3.14: Thousands of genome-wide loops were detected [29]. **(A)** Peaks were identified by detecting high intensity pixels with respect to the neighborhoods. These "peaks" indicate the presence of a loop and are marked with blue circles (radius = 20 kb) on the heatmaps (10 kb resolution). The number of contacts for each peak is also indicated. **(B)** Loops are also preserved from Human to Mouse. Here, conservation of the 3D structure on synteny regions is displayed. The contact matrix above is shown at 25 kb of resolution, below at 10 kb resolution.

IMR90. On the other side, 510 loops in IMR90 that were absent in GM12878 were found; the corresponding peak loci overlapped the promoters of 94 genes that were upregulated in IMR90, but of only three genes that conversely were upregulated in GM12878 [29].
It was also seen that a large number of peaks (38%) are located at the corner of a contact domain, so at domain boundaries. Vice versa, a large part of the domains (39%) have peaks in their corners, so it was used the term "loop domains" to identify them (Figure 3.15 D).
The next step was to see whether these peaks were associated with specific proteins. It was found that a large part of peak loci are bound by the insulator protein CTCF (86%) and the cohesin subunits RAD21 (86%) and SMC3 (87%). Because most of the loops demarcate domains, this finding was consistent with studies suggesting that CTCF demarcate topological domains [16]. Since the consensus DNA sequence for CTCF-binding sites is usually written as 5'-CCACNAGGTGGCAG-3' and it is not palindromic, a pair of CTCF sites in the same chromosome can have one of the following possible orientations: (1) same direction on one strand, (2) same direction on the other strand, (3) convergent on opposite strands and (4) divergent on opposite strands. By analyzing the 4,322 peaks in GM12878 where the two corresponding peak loci each contained a single CTCF-binding motif, it was found that most of the pairs (92%) were convergent and moreover this kind of orientation was overwhelmingly more frequent than the divergent orientation [29] (Figure 3.15 E). Taken together, these considerations suggest that a pair

of CTCF sites in convergent orientation is required to form a loop.

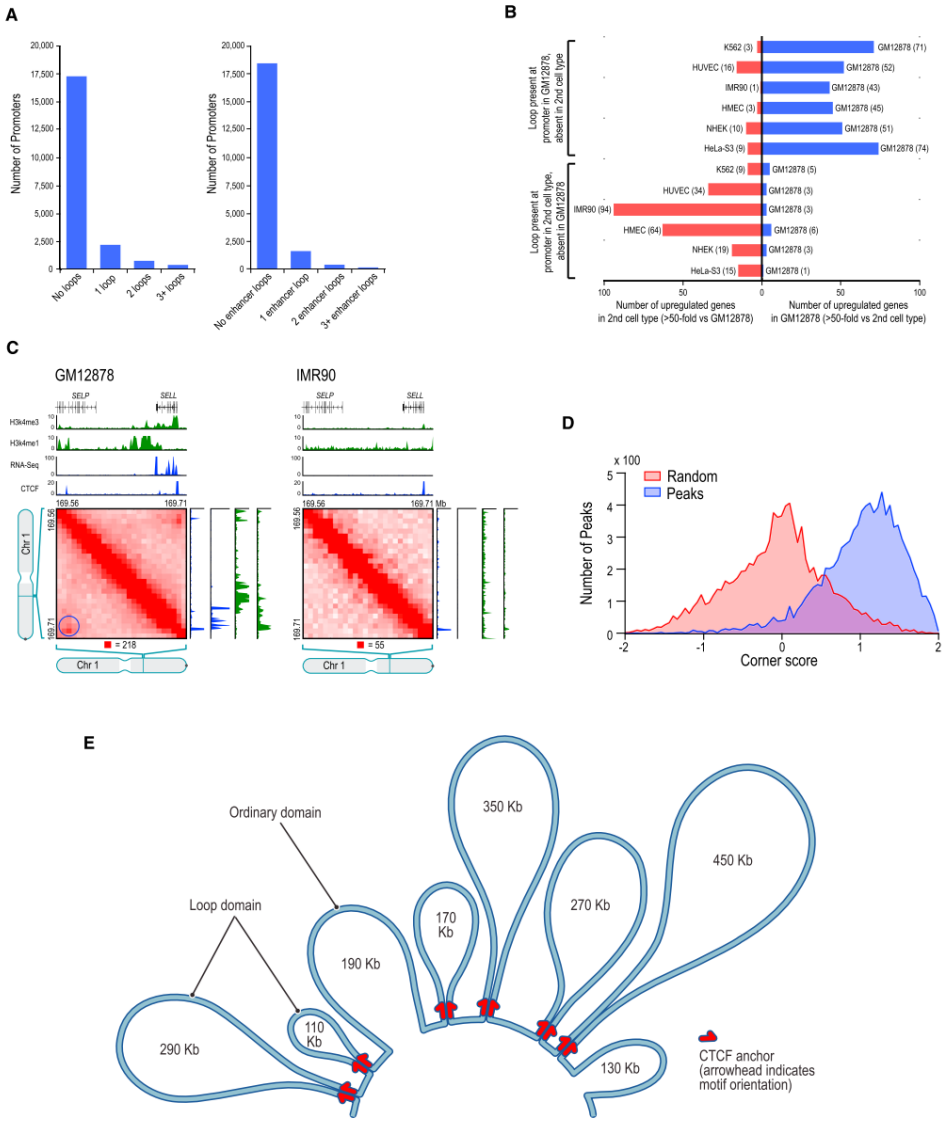

FIGURE 3.15: A strong association between loops and gene regulation was discovered [29]. **(A)** Histograms showing the number of promoters associated with loops (left) and with loops where the distal peak locus contains an enhancer (right). **(B)** Bar plot showing the higher number of upregulated genes whose promoters participate in a loop on GM12878 with respect to a 2nd cell type and vice versa. **(C)** Loop anchored at the SELL promoter in GM12878 where the gene is on (left). The loop is absent in IMR90 where the gene is off (right). **(D)** Histogram of the "corner score" for peak pixels versus random pixels. This correlates with the higher percentage of peaks that locate at a domain corner. **(E)** Most of the pairs of CTCF sites that anchor a loop were found in convergent orientation.

3.5 Summary

In this third chapter the Hi-C protocol has been presented, underlining its major contribute to the mapping of whole genomes. The data analysis has been shown as well. A further analysis based on Hi-C data, regarding topological domains has been also provided. Then, the latest 2014 protocol named "in situ Hi-C" has been introduced, which allows to create Hi-C maps at very high resolution, enabling much deeper genome-wide studies about the three-dimensional architecture of genomes.
In the next chapter, I present the pipeline that we developed for a standardized Hi-C data processing and visualization, and topological domains analysis, named HiCtool.

Chapter 4

HiCtool: a standardized pipeline to analyze Hi-C data

We developed HiCtool, a bioinformatics tool for Hi-C data analysis, with the aim of creating a standardized and flexible framework to process and visualize Hi-C datasets. Albeit the presence of a large amount of source data and several analysis and visualization software, there is still not a standardized way for processing them, which would allow the public to compare different outputs in a consistent way. HiCtool addresses this need providing a complete and exhaustive pipeline that leads the user, even beginner, easily and quickly to the results. For each step of the analysis, the code that is used, the inputs and the outputs are specified. In addition, each section of the documentation contains an explanation that briefly summarizes what it is about, to make all the process clear and user-friendly. Therefore, the key advantage is that users do not needed to read any other software documentation but only follow the steps listed in the related section, providing specific input data to get the output.
The tool is based on Python libraries and packages. Three steps of analysis and four main features were developed to present a clear and concise procedure to analyze and visualize Hi-C data.

4.1 Tool principle and main features

The tool principle is to integrate several software to carry out a comprehensive and standardized Hi-C data analysis, from the source data downloading to the visualization of the heatmaps [4] and the identification of topological domains [16].
The first feature consists of providing functions to save and retrieve the information for each section. This is the way to handle output data on HiCtool or using it for further

analyses.
The second feature is the pre-processing pipeline, which gives a comprehensive guide about how to obtain the input data for the following analysis, which includes contact matrices normalization, heatmaps visualization and topological domains identification.
The third feature is the possibility of a full analysis customization. Since all the source codes are provided, it is possible to update parameters not only related to the output results but also to the data normalization process.
The fourth feature is that two ways of visualizing the data are provided, to improve the feedback for the user. The first way is a normalized intrachromosomal heatmap, where each point shows the interaction frequency between a couple of pre-divided bins along a chromosome with different intensity color. The second way is an enrichment heatmap, where each point represents the log_2 of the observed over expected data based on genomic distance. For each plot there is also the possibility of adding a custom colorbar (specific for the two kinds of heatmaps) and a histogram.

4.2 HiCtool pipeline

HiCtool is composed of a pipeline divided into three sections: preprocessing of the data, data analysis and visualization, topological domains analysis.

4.2.1 Preprocessing of the data

In this section, HiCtool provides an exhaustive pipeline from the downloading of the source data to the final *.bam* files that are used for the following analysis. To implement the preprocessing of the data, HiCtool integrates several software included SRA Toolkit, Bowtie 2, SAMTools and Bedtools. The input data is expected to be one or more *.sra* files downloaded from GEO. It is needed also a *.fa* (or *.fasta*) file of the reference sequence (hg38 in our case) to compute the alignment of the paired reads over the genome.
After downloading the data, the next step of preprocessing is the conversion of the input data from sra to fastq format using SRA Toolkit. The key feature of this step is that the *.fastq* files are splitted per reads, this allows to obtain *.fastq* files each related to one of the mate in the Hi-C data library. The not paired reads are not useful for our analysis. The SRA Toolkit command that implements this function is the following:

```
fastq-dump HiCfile.sra --split-3
```

Now the paired reads are aligned over the reference genome sequence, according to the last step of the Hi-C protocol [4]. To implement this, Bowtie 2 has been used. This

software requires first to build an index for the reference sequence, and then a paired reads alignment can be computed:

```
bowtie2-build hg38.fa index
bowtie2 -x index -1 HiCfile_1.fastq -2 HiCfile_2.fastq -S HiCfile.sam
```

The output *.sam* file is then converted into bam format and sorted by chromosomal coordinates using SRA Toolkit:

```
samtools view -bS HiCfile.sam > HiCfile.bam
samtools sort -m 5000000000 HiCfile.bam HiCfile.sort
```

Potential PCR duplicates are now removed from the resulting aligned *.bam* file using SAMTools. To do this, first it is needed to sort the *.bam* file by reads name (`sort -n`) and fill in the mate coordinates, size and mate-related flags into the *.bam* file (`fixmate`). Lastly, after sorting again by chromosomal coordinates (`sort`), we can remove duplicates from the *.bam* file (`rmdup`):

```
samtools sort -m 5000000000 -n HiCfile.sort.bam HiCfile.namesort
samtools fixmate HiCfile.namesort.bam HiCfile.fixmate_namesort.bam
samtools sort -m 5000000000 HiCfile.fixmate_namesort.bam HiCfile.fixmate_sort
samtools rmdup HiCfile.fixmate_sort.bam HiCfile_noDup.sort.bam
```

The following last two steps are needed for the normalization of the data. First, the *.bam* file is splitted into two *.bam* files, each related to one read of the pairs. Again SAMTools is used and the commands are the following:

```
samtools view -h -f 0x40 HiCfile_noDup.sort.bam > HiCfile_pair1.bam
samtools view -h -f 0x80 HiCfile_noDup.sort.bam > HiCfile_pair2.bam
```

Now a fragment-end (fend) related *.bed* file is created, mapping the restriction enzyme (RE) sites over the reference genome. This is needed in the normalization procedure and it contains restriction sites coordinates and additional information related to fragment properties (i.e. GC content). In order to align all the restriction sites for a certain cutting enzyme (HindIII in our case), a *.fastq* file related to the RE sites has to be provided. In general, for the quality score of the RE sequence a default average score "I" can be added. To locate all the coordinates of the RE site, the multiple alignment command in Bowtie 2 is implemented (`bowtie2 -k`) and finally the alignment file is converted to bed format via SAMTools and Bedtools:

```
echo -e "@HindIII\nAAGCTT\n+\nIIIIII" > HindIII.fq
bowtie2 -k 3000000 -x index -U HindIII.fastq -S restrictionsites.sam
samtools view -bS restrictionsites.sam > restrictionsites.bam
bedtools bamtobed -i restrictionsites.bam > restrictionsites.bed
```

4.2.2 Data analysis and visualization

The complex experimental protocol of Hi-C unavoidably produces numerous biases and experimental artifacts. In this section we address these biases and we provide the pipeline to normalize the data and then plot the heatmaps.
According to Yaffe and Tanay [66], the most significant biases are related to spurious ligation products between fragments, transcription start sites (TSSs) and CTCF-bound sites within topological domains, length and GC content of the fragments. Spurious ligation products are sequence pairs which represent ligation products between non-specific cleavage sites rather than restriction fragment ends. Specificly, they are those paired-reads whose sum of the two distances to the nearest restriction sites is larger than 500 bp. The majority of reads maps within 500 bp to the nearest restriction sites, while 12% and 4% of the reads, respectively for HindIII and NcoI restriction enzymes, represent spurious mapped reads (Figure 4.1 a,b). About TSSs, analysis of the distribution of *cis* contacts involving fragment ends located 0-5 kb upstream of an active TSS (promoter side) showed a strong enrichment from 20 kb to ∼400 kb upstream and, in a weaker fashion, from 20 kb to ∼400 kb downstream, increasing the probability that long-range contacts may be associated with the active transcriptional state (Figure 4.1 c). A specular phenomenon is observed for fragment ends located 0-5 kb downstream of an active TSSs on the gene side (Figure 4.1 d). In addition, fragments located 0-5 kb on one side of a CTCF-binding site displayed *cis* contact enrichment and asymmetry over a range up to ∼400 kb, while contacts that are directing crossing the binding sites are depleted (Figure 4.1 e,f). This could be explained by the correlation between CTCF-binding sites and topological domain boundaries, which show depletion of contacts and whose nearby regions present an asymmetry of contact frequencies. About fragments length, long and short fragments may have a variable ligation efficiency. The probability of contact can be also influenced by the GC content near the ligated fragment ends, up to 200 bp next to the restriction sites [66].

To normalize the data several approaches have been proposed, which are divided into two categories, probabilistic and matrix balancing [67]. The probabilistic approach is implemented in the R software HiCPipe [66] and HiCNorm [68]. HiCPipe uses restriction fend features, partitioning their ranges into bins, and iteratively learns correction

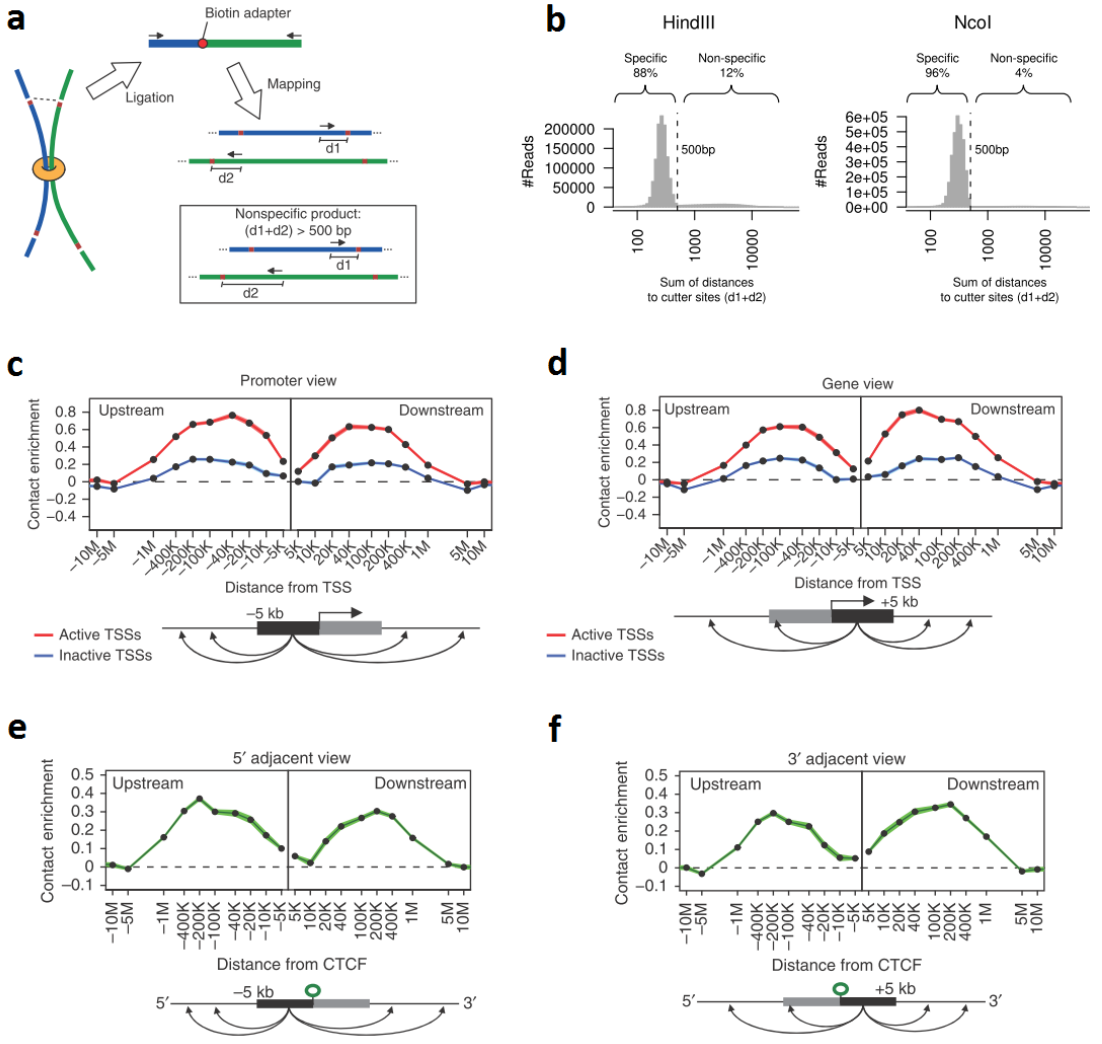

FIGURE 4.1: Hi-C sources of bias [66]. **(a,b)** Spurious ligation products bias. The cartoon shows how to evaluate the distance of a read to the nearest restriction site. The histograms represent the distribution of distances to restriction sites. Two populations of reads are observed for each restriction enzyme: one for normally ligated products (HindIII 88%, NcoI 96%) and one for reads mapped farther away from restriction sites. **(c,d)** NcoI log_2 enrichment values (y axis) of *cis* contacts involving fragment ends up to 5 kb upstream (c) and downstream (d) of a TSS over controls. The data represent the enrichment values for active (red) and inactive (blue) TSSs. **(e,f)** Normalized *cis*-contact profiles for fragment ends located on the 5' side (e) and the 3' site (f) of CTCF sites.

values for each combination of bins based on a binomial distribution of observed versus unobserved fend interactions. HiCNorm uses a Poisson regression model using binned counts instead of binary output. The matrix balancing approach is used by HiCLib [69], included in the R package HiTC [70]. Given a symmetric matrix $A \in \mathbb{R}^{n \times n}, A \geq 0$, which is a heatmap in our case, this method is based on finding the diagonal matrices D_1 and D_2 so that the sum of all the rows and columns of $P = D_1 A D_2$ is one.

In HiCtool, we normalized the data according to the probabilistic model of Yaffe and

Tanay [66], performed by using the Binning algorithm of the Python package HiFive [67]. We chose this normalization method because it derives from a comprehensive biological background about Hi-C potential sources of biases and, therefore, it is one of the most popular approaches. We did not use HiCPipe because HiFive's Binning algorithm has a more consistent performance across all binning resolutions and also, at bin sizes lower than 50 kb, HiCPipe performs even worse [67]. This is a critical point for our pipeline since, according to Dixon *et al.* [16], we process the data at a resolution of 40 kb to enable also a topological domains analysis (Dixon *et al.* used Yaffe and Tanay's method as well). In addition, HiFive's capability of handling high-resolution data makes it able to process the last generation of Hi-C datasets, derived from the "in situ Hi-C" protocol. Lastly, about the running time, HiFive's Binning algorithm performs faster not only than HiCPipe, but also than the other software mentioned above (Figure 4.2). After data normalization, for plotting the heatmaps we used the Python Imaging Library (PIL), resulting in a better visualization and understanding than the plots that HiFive produced.

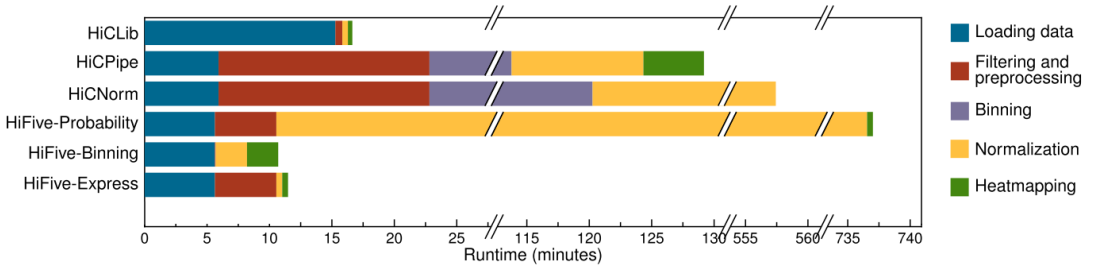

FIGURE 4.2: Running time for Hi-C data analysis software [67]. For each software or method, the runtime in minutes is displayed, partitioned for each stage of the processing. All times are determined for chromosome 1 of mouse (restriction enzyme HindIII) producing a heatmap at a resolution of 10 kb, and using a single processor.

To normalize the data using HiFive, a Fragment-end (FEND) object (hdf5 format) is required, containing the information about the fragments created by digestion of a genome using a specific restriction enzyme (HindIII in our case). This feature allows a full customization of the process, related to the restriction enzyme used in the experiment, the reference genome, or even adding other possible sources of biases to be taken into account. To create a Fend object, information RE fragments are supplied in the form of a *bed*-formatted file. In our case, it contains the information about the location of the RE sites for the target genome for each chromosome and the GC content of the 200 bp upstream and downstream of each RE site (Figure 4.3). To create a Fend object, the HiFive function `hifive.fend` is used:

```
import hifive
```

```
fend = hifive.Fend('fend_object.hdf5', mode='w')
fend.load_fends(RE_data_filename, re_name='HindIII', format='bed')
fend.save()
```

chr	start	stop	name	score	strand	gc
chr1	80518483	80518489	0	1	-	0.2500000,0.2050000
chr1	224801750	224801756	0	1	+	0.2350000,0.4250000
chr1	152542628	152542634	0	1	-	0.5500000,0.5250000
chr1	242747160	242747166	0	1	-	0.4750000,0.3800000

FIGURE 4.3: Fragment-end related *bed* file. First four lines of the fend *bed* file. The chromosome, start and end coordinate of the RE site, strand and GC content of the fragment are displayed. The fields "name" and "score" have been added automatically by Bedtools and they are not used in the analysis.

The second step is to create a HiC dataset (HiCData object) from both the Fend file and mapped data in bam format (two separated *.bam* files, related to the first and the second mates of the paired reads). This is done using the function `hifive.HiCData`. This function allows also to insert a cutoff (`maxinsert`) for filtering paired-end reads whose total distance to their respective restriction sites exceeds that value, so we choose 500 bp according to [66]. In such a way, we remove the bias related to spurious ligation products.

```
import hifive

data = hifive.HiCData('HiC_data_object.hdf5', mode='w')
data.load_data_from_bam('fend_object.hdf5',
                        [BAM_file_1, BAM_file_2],
                        maxinsert=500)
data.save()
```

Now a HiC project object is created. It contains links to HiCData object and Fend object, information about which fends to include in the analysis, model parameters and learned model values. This is the standard way of working with Hi-C data in HiFive and this object will be used for learning the correction model and downstream analysis.

```
import hifive

hic = hifive.HiC('HiC_project_object.hdf5', 'w')
hic.load_data('HiC_data_object.hdf5')
hic.save()
```

According to Yaffe and Tanay [66], to do not take into account of biased regions up to ∼400 kb upstream or downstream of an active transcription start site (TSS) or CTCF-binding site, we filter out fragments that interact within a distance of 500 kb

(`mindistance`) before learning the correction parameters related to fend biases (length and GC content). We choose 500 kb to be more confident of not considering biased regions, however the number of fends removed inserting a cutoff of 400 kb is not significantly different (1,802,066 than 1,808,135 respectively for a cutoff of 400 kb and 500 kb).

```
import hifive

hic = hifive.HiC('HiC_project_object.hdf5')
hic.filter_fends(mininteractions=1, mindistance=500000, maxdistance=0)
hic.save()
```

With the filtering using a cutoff of 500 kb, 1,808,135 over 3,441,412 fends are removed (Human ES cells H1, GSM862723).
The HiC project object is used now to estimate the distance-dependence relationship from the data prior to normalization, in order to avoid biases that may result due to restriction site distribution characteristics or the influence of distance/signal relationship. Restriction sites throughout the genome are unequally distributed, resulting in greatly varying sets of distances between fragments and their neighbors. Interaction signal is strongly related to inter-fragment distance, so this unequal distribution means that fragments with lots of shorter adjacent fragments have a nearby neighborhood of higher interaction values than fragments surrounded by longer fragments, simply due to cutsite variation. To find the HiC distance function, `find_distance_parameters` is used. This function requires three input parameters: `numbins` is the number of bins the range of interaction distances is broken into to compute the analysis; `minsize` is used to specify the maximum size of all the interaction distances which are covered by the first `numbins` bin; `maxsize` sets the maximum size of interaction distances taken into account. Setting `maxsize` to zero, as in the following function call, means that the maximum size is equal to the longest interaction distance.

```
import hifive

hic = hifive.HiC('HiC_project_object.hdf5')
hic.find_distance_parameters(numbins=90, minsize=200, maxsize=0)
hic.save('HiC_distance_function.hdf5')
```

To learn the correction parameters related to fragment length and GC content biases, Yaffe and Tanay's method is used [66]. They defined a multiplicative probabilistic model that computes the prior probability of a contact between two fragment ends given their properties (we consider the fragment length and GC content in our analysis). For each source of bias, a seed correction matrix is computed which contains the coverage enrichment, defined as the ratio between the observed number of *cis* contacts and the total

number of assayed fragment pairs (Figure 4.4). To build each matrix, the lengths and GC contents range is divided into 20 bins, such that each bin contains the same number of fragments. Then, each entry of the seed matrix for the fragments length (the same for GC content) is computed as:

$$S_{len}[i,j] = (1/P_{prior}) \cdot \frac{O_{len}[i,j]}{T_{len}[i,j]} \tag{4.1}$$

where, P_{prior} is the prior probability to observe a pair and it is equal to the total number of observed *cis* pairs divided by the total number of possible *cis* pairs. $O_{len}[i,j]$ is the number of observed *cis* pairs such that one fragment end is in bin i and the other is in bin j. $T_{len}[i,j]$ is the total number of possible unique *cis* pairs such that one fragment end is in bin i and the other is in bin j.

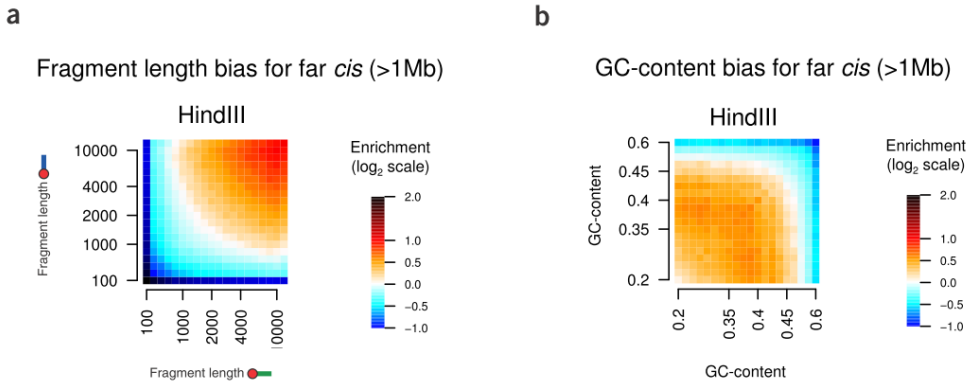

FIGURE 4.4: Seed matrices for fragments length and GC content [66]. **(a)** Length biases for *cis* contacts. Enrichment ratios for fragment ends divided into 20 bins according to the fragment length. **(b)** GC biases for *cis* contacts. Similar to (a) but here fragment ends are grouped according to their local GC content.

Now, given two fragment ends a,b, the probability $P(X_{a,b})$ to observe them in a paired-end read is defined as:

$$P(X_{a,b}) = P_{prior} \cdot F_{len}(a_{len}, b_{len}) \cdot F_{gc}(a_{gc}, b_{gc}) \tag{4.2}$$

where $a_{len}, b_{len}, a_{gc}, b_{gc}$ are the fragment length bins and GC content bins of the two ends, while $F_{len}(a_{len}, b_{len}), F_{gc}(a_{gc}, b_{gc})$ are the two seed matrices adjusted using a maximum likelihood optimization procedure. The likelihood function is:

$$\begin{aligned} L(F_{len}, F_{gc}) &= \prod_{\{a,b\} \in I} P(X_{a,b}) \cdot \prod_{\{a,b\} \notin I} (1 - P(X_{a,b})) \\ &= \prod_{c=(a_{len}, a_{gc}, b_{len}, b_{gc})} P(X_{a,b})^{n_c} \cdot [1 - P(X_{a,b})]^{m_c} \end{aligned} \tag{4.3}$$

where I is the set of observed fragment end pairs, n_c is the number of observed pairs that match the bin criteria of c, and m_c is the number of pairs that match the criteria but were not observed. The likelihood function is then maximized by alternating between the optimization of the two matrices, until the improvement in the log-likelihood is smaller than a threshold:

$$\begin{aligned} F_{len}^{n+1} &= \arg\max_{F_{len}} L(F_{len}, F_{gc}^{n}), \quad F_{gc}^{n+1} = F_{gc}^{n} \\ F_{gc}^{n+1} &= \arg\max_{F_{gc}} L(F_{len}^{n}, F_{gc}), \quad F_{len}^{n+1} = F_{len}^{n} \end{aligned} \tag{4.4}$$

The resulting correction matrices are similar but not identical to the seed matrices, because of the adjustment derived by the likelihood optimization procedure which is due to the cross-correlation of fragment length and GC content.
To learn the correction values for fragment length and GC content according to this algorithm, the function `find_binning_fend_corrections` is used:

```
import hifive

hic = hifive.HiC('HiC_distance_function.hdf5')
hic.find_binning_fend_corrections(max_iterations=1000,
                                  mindistance=500000,
                                  maxdistance=0,
                                  num_bins=[20,20],
                                  model=['len','gc'],
                                  parameters=['even','even'],
                                  usereads='cis',
                                  learning_threshold=1.0)
hic.save('HiC_norm_binning.hdf5')
```

This function performs the optimization of correction values using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) non-linear algorithm of the Python package Scipy. According to this function, the algorithm can terminate because of `learning_threshold` (chosen as 1 according to Yaffe and Tanay) or `max_iterations`, which is the maximum number of iterations before terminating. `mindistance` and `maxdistance` set respectively the minimum and the maximum inter-fragments distance to be included in modeling (setting `maxdistance` to zero means that the maximum size is equal to the longest interaction distance). `num_bins` is the number of equally sized bins the range of interaction distances is broken into (the value 20 is chosen according to Yaffe and Tanay). `model` is a list which specifies the parameters taken into account for the normalization (length and GC content). `parameters` is a list of types, one for each element in `model`. The value `'even'` means that each parameter bin (i.e. each bin of the correction matrices)

should contain approximately even numbers of fragments. `usereads` specifies which set of interactions is used to learn the correction parameters (`'cis'` means that we are considering intrachromosomal interactions).

After learning the correction parameters, for any arbitrary division of the genome, we calculate two matrices: an observed contact matrix and a fend expected contact matrix. The observed intrachromosomal contact matrix *O[i,j]* contains the observed reads count. The fend expected intrachromosomal contact matrix *E[i,j]* contains the sum of corrections for all the paired-reads in each bin, according to the previous model. Then, the normalized reads count is calculated as:

$$N[i,j] = \frac{O[i,j]}{E[i,j]} \tag{4.5}$$

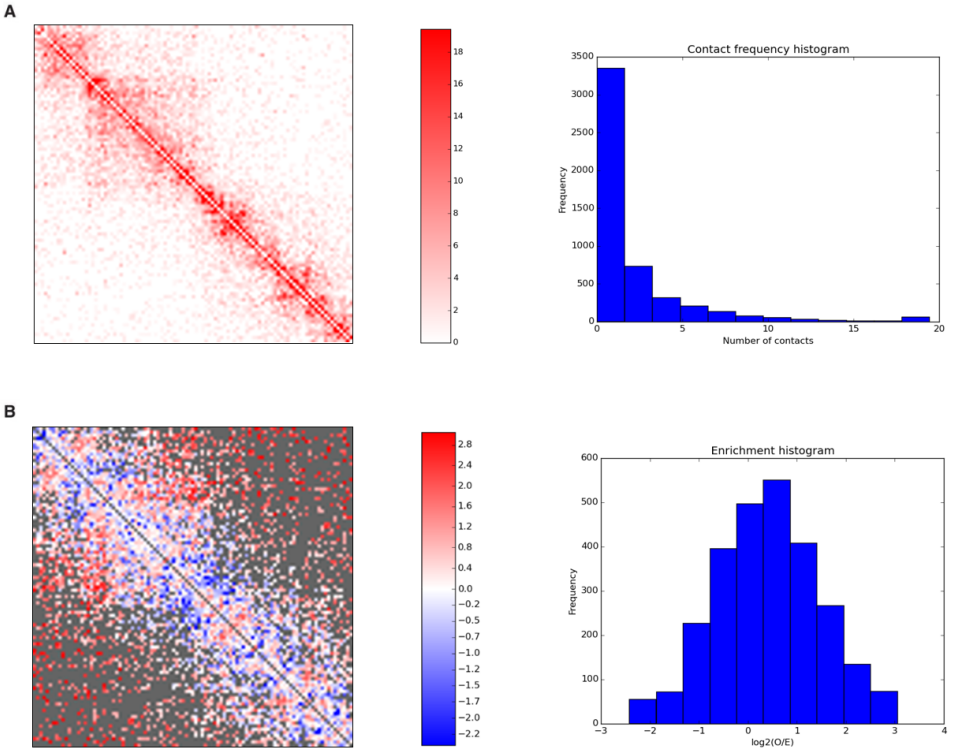

FIGURE 4.5: Normalized heatmaps of Chr 6: 50-54 Mb at a bin size of 40 kb. **(A)** Normalized read counts (range from 0 to 19 reads). The heatmap represents the 98^{th} percentile of the non-zero data. **(B)** $log_2(enrichment)$ (range from -2.421 to 3.047). The gray pixels represent non-valid $log_2(enrichment)$ values, i.e. where the corresponding expected value is 0. The heatmap represents the 99^{th} percentile of the positive and negative log_2 values. (GEO accession number: GSM862723)

In addition, we calculate also an enrichment expected contact matrix, which contains the expected read counts considering the distance between fends and the learned correction parameters. The data enrichment is then calculated as the ratio between the observed and the enrichment expected data, as above.

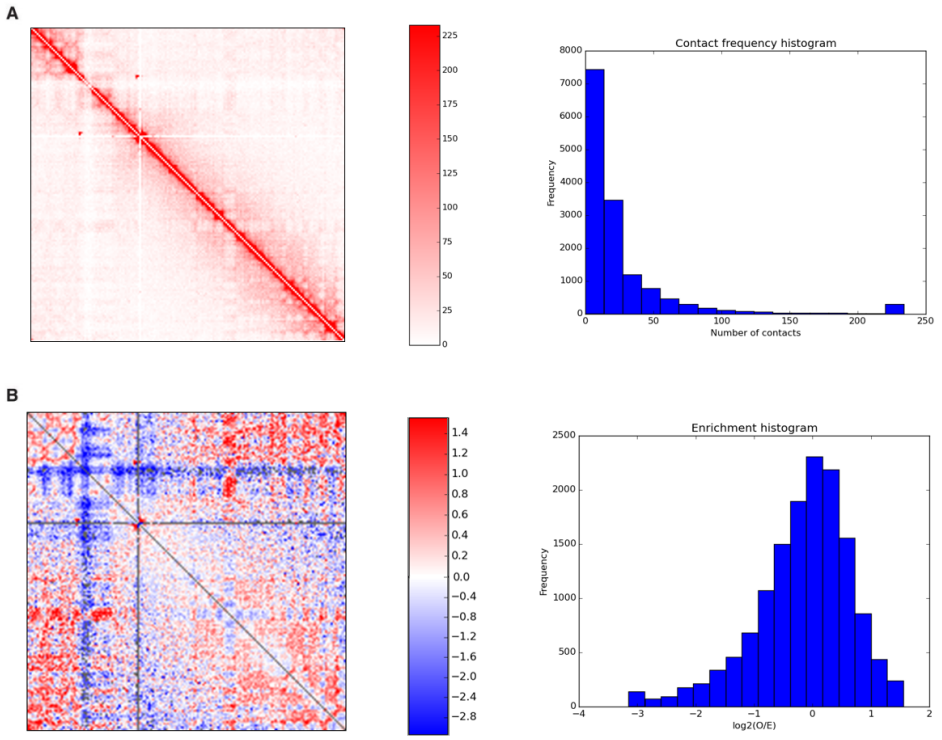

FIGURE 4.6: Normalized heatmaps of Chr 6: 0-171 Mb at a bin size of 1 Mb. **(A)** Normalized read counts (range from 0 to 233 reads). The heatmap represents the 98^{th} percentile of the non-zero data. **(B)** $log_2(enrichment)$ (range from -3.147 to 1.554). The gray pixels represent non-valid $log_2(enrichment)$ values, i.e. pixels where the corresponding expected value is 0. The heatmap represents the 99^{th} percentile of the positive and negative log_2 values. (GEO accession number: GSM862723)

After the normalization, heatmaps are plotted (see the code in Appendix A, section A.2). For the normalized data, we plot the 98^{th} percentile of the non-zero data to be confident of not considering potential outliers and have a better visualization (i.e. setting a proper heatmap color range). A histogram of the data was added to the heatmaps for a complete understanding. Figures 4.5, 4.6 show heatmaps and histograms derived from the normalized data with the pipeline displayed above. Figure 4.7 shows the effect of normalization on the observed data, displaying either the contact matrices before and after normalization, for a region of chromosome 6 of Human ES cells H1. From the output data, it is clear that the normalization does not alter significantly the chromatin conformation. This means that the effect of the corrected read counts is just to shift a little the topological domain coordinates.

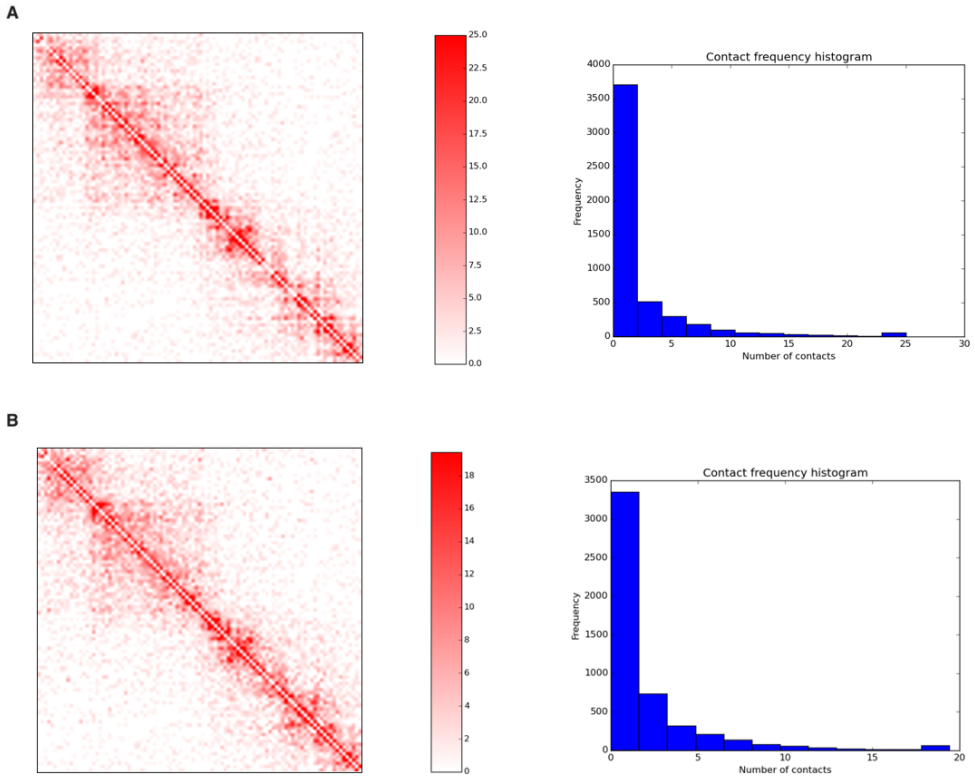

FIGURE 4.7: Comparison between observed and normalized heatmaps of Chr 6: 50-54 Mb at a bin size of 40 kb. **(A)** Observed data (range from 0 to 25 reads). **(B)** Normalized data (range from 0 to 19 reads). Both the heatmap represent the 98^{th} percentile of the non-zero data. (GEO accession number: GSM862723)

4.2.3 Topological Domains analysis

The Topological Domains analysis section provides the code to calculate the DI and visualize it (Appendix A, section A.3). It allows to calculate both the observed DI (3.1) and the "true DI" using a Hidden Markov Model implemented with MATLAB. The resulting plot shows both the observed DI and the true DI states, therefore it is possible to infer about the presence of topological domains and boundaries over the genome, whose coordinates are calculated as well.

First, the contact data are loaded and the observed DI values are calculated and saved (subsection A.3.1). Then, the observed DI data are used to calculate the "true DI" states with a HMM, which allow to identify the locations of the topological domains in the genome (MATLAB code in Appendix A, subsection A.3.2). Specifically, a domain is initiated at the beginning of a single downstream biased HMM state (red color in Figure 4.8) and it is continuous throughout any consecutive downstream biased states. The domain will then end when the last in a series of upstream biased states (green color in Figure 4.8) are reached, with the domain ending at the end of the last HMM upstream

biased state [16].

Chromosome	Start coordinate	End coordinate
Chr 6	50.12 Mb	50.60 Mb
Chr 6	50.68 Mb	52.08 Mb
Chr 6	52.20 Mb	52.84 Mb
Chr 6	52.88 Mb	53.08 Mb
Chr 6	53.12 Mb	53.72 Mb
Chr 6	53.76 Mb	53.96 Mb

TABLE 4.1: Topological domains coordinates for Chr 6: 50-54 Mb at a bin size of 40 kb. (GEO accession number: GSM862723)

FIGURE 4.8: Topological Domains on Chr 6: 50-54 Mb. The heatmap shows topological domains (blue highlighted) on chromosome 6, with coordinates from 50 Mb to 54 Mb at a bin size of 40 kb (GSM862723). Observed DI (bar plot) and "true DI" states (below) for chromosome 6, with coordinates from 50 Mb to 54 Mb at a bin size of 40 kb . The plot shows 6 topological domains and 7 topological domain boundaries according to the HMM states shifts (the coordinates are listed on Table 4.1).

To calculate the topological domains coordinates (Table 4.1), first we extract all the potential start and end coordinates according to the definition stated above, and then we

evaluate a list of conditions to take into account the possible presence of gaps between a series of positive or negative state values (Figure 4.9). In the first step, the condition to extract the start coordinate of the first topological domain is checked. In particular, it is checked if there are some negative states before the first positive state: if "yes", we shift to the next negative state and we check again, until we find a negative state that is after the first positive state value, i.e. we have found the first topological domain coordinate. Since we assume to not take account of gaps between positive or negative states, a check if there are any gaps between positive states is now performed: if "yes", the gaps are removed and then the same check is performed for negative states as well. After having removed all the gaps, the domain coordinates are recorded and the next couple of positive and negative states is analyzed, starting again from the second step of the process.

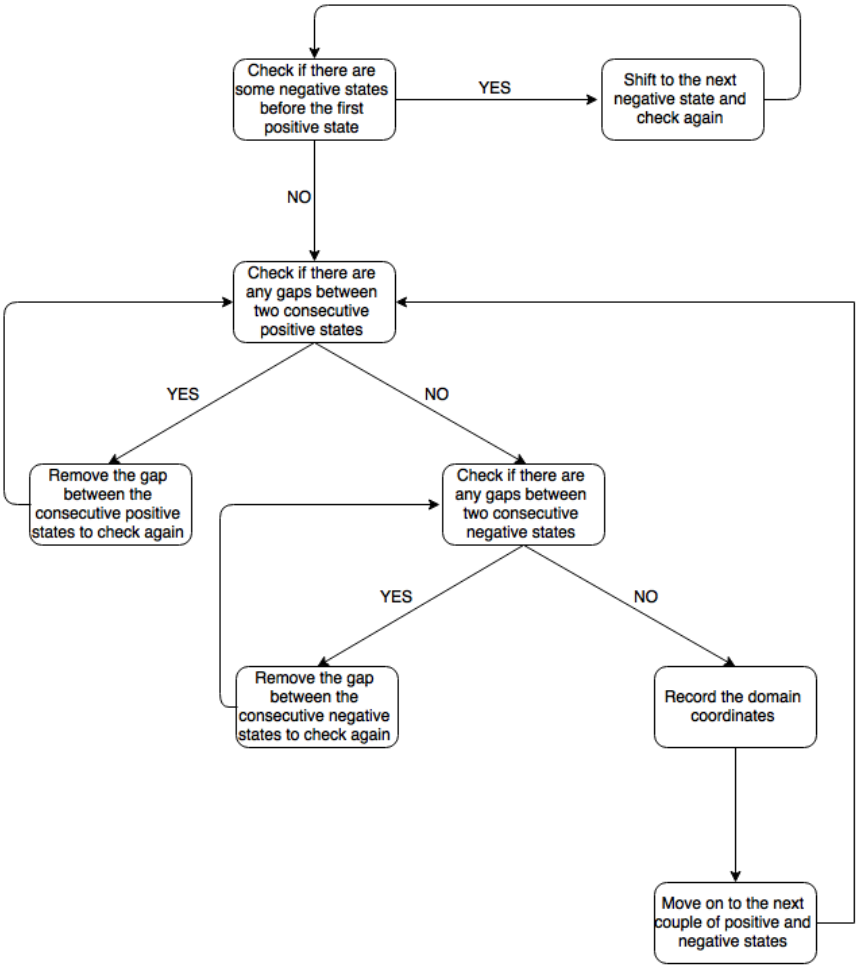

FIGURE 4.9: Topological Domains coordinates flowchart.

4.2.4 Data results

We validated HiCtool comparing our results to Lieberman-Aiden *et al.* data, which are available on Hi-C Data Browser[1]. We observed a high degree of correlation between the observed contact matrices generated with our pipeline and those generated by Lieberman-Aiden *et al.*, with an average Pearson's $r = 0.8164$ (see Table 4.2). This result demonstrates the quality of HiCtool to process and reproduce accurately Hi-C data.

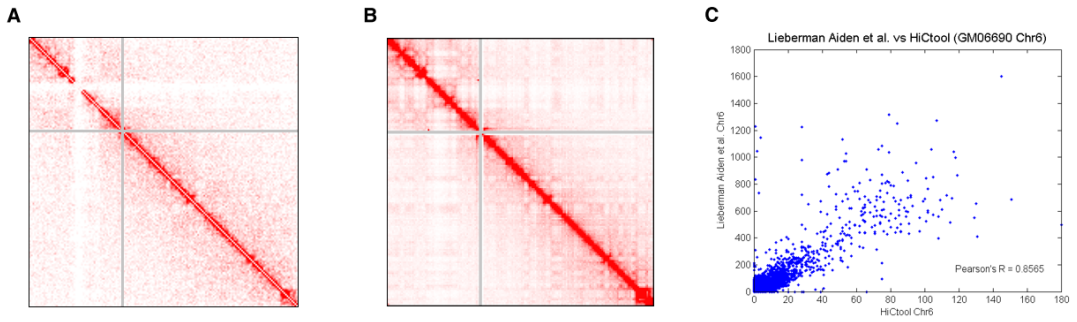

FIGURE 4.10: Comparison between HiCtool and Lieberman-Aiden *et al.* data. Heatmaps and scatter plot for chromosome 6 at a bin size of 1 Mb (Human EBV-transformed lymphoblastoid GM06690). **(A)** HiCtool heatmap. **(B)** Lieberman-Aiden *et al.* heatmap. **(C)** Scatter plot showing the high degree of correlation between HiCtool and Lieberman-Aiden *et al.* data (Pearson's $r = 0.8565$).

Next, we compared our topological domains to Dixon *et al.* domains [16] (see Table 4.3). Specificly, we found a greater number of domains than Dixon *et al.* for each chromosome and the additional domains are validated by the Directionality Index shifts over the genome (Figure 4.8). About the domains listed on Table 4.1, only two of that (50.12-50.60 Mb, 50.68-52.08 Mb) have corresponding domains in Dixon *et al.* data (respectively 50.16-50.68 Mb, 50.76-52.04 Mb); the following four domains we found (in the range from 52.20 Mb to 53.96 Mb) are included in a region covered by only one domain in Dixon *et al.* data (52.24-53.48 Mb).

The pipeline was tested on different cell lines and datasets of Homo Sapiens (hg38) and Mus Musculus (mm10), taken from the library that we compiled on 4D Nucleome Web Portal[2] (see Table 4.4 for details). The 4DN library is a collection of genome interaction papers related to the Chromosome Conformation Capture (3C) based assays (4C, 5C and Hi-C). Intrachromosomal contact matrices, observed DI and HMM states values are given in txt format. We provide the Python functions to save and load these files, in order to allow the usage for further analyses. The topological domains coordinates are saved in a *.txt* file as well but already formatted, to make the user able to read it directly to compare the data to the plot.

[1] http://hic.umassmed.edu/welcome/welcome.php
[2] http://www.4dnucl.org/

Chromosome	**Pearson's r**
Chr 1	0.8117
Chr 2	0.8587
Chr 3	0.8576
Chr 4	0.8575
Chr 5	0.8673
Chr 6	0.8565
Chr 7	0.8381
Chr 8	0.8335
Chr 9	0.7515
Chr 10	0.8099
Chr 11	0.8511
Chr 12	0.8288
Chr 13	0.8912
Chr 14	0.8413
Chr 15	0.7734
Chr 16	0.7690
Chr 17	0.7334
Chr 18	0.8473
Chr 19	0.6286
Chr 20	0.8462
Chr 21	0.7355
Chr 22	0.8231
Chr X	0.8656

TABLE 4.2: Comparison between HiCtool and Lieberman-Aiden *et al.* data. Pearson's r coefficients calculated for intra-chromosomal contact matrices for each chromosome of Human EBV-transformed lymphoblastoid GM06690.

Chromosome	**HiCtool number of domains**	**Dixon number of domains**
Chr 1	322	208
Chr 2	298	183
Chr 3	259	169
Chr 4	254	139
Chr 5	223	136
Chr 6	204	129
Chr 7	191	133
Chr 8	172	106
Chr 9	157	134
Chr 10	169	99
Chr 11	184	113
Chr 12	163	100
Chr 13	117	71
Chr 14	113	58
Chr 15	108	67
Chr 16	84	76
Chr 17	99	67
Chr 18	80	55
Chr 19	71	70
Chr 20	74	51
Chr 21	55	27
Chr 22	52	22
Chr X	176	89

TABLE 4.3: Topological domains number. For each chromosome it is shown the number of domains we found using HiCtool and the number of domains found by Dixon *et al.* [16]. (GEO accession number: GSM862723)

GEO Accession Number	RE	Species	Cell Line	Reference
GSM455133	HindIII	Homo Sapiens	EBV-transformed lymphoblastoid GM06990	[4]
GSM862723	HindIII	Homo Sapiens	Human Embryonic Stem Cells H1	[16]
GSM862724	HindIII	Homo Sapiens	Fetal lung fibroblast IMR90	[16]
GSM862720	HindIII	Mus Musculus	Mouse Embryonic Stem Cells J1	[16]
GSM1551550	MboI	Homo Sapiens	B-lymphoblastoids GM12878	[29]
GSM1551599	MboI	Homo Sapiens	Lung Fibroblasts IMR90 (CCL-186)	[29]
GSM1551633	MboI	Mus Musculus	B-lymphoblasts CH12-LX	[29]
GSM1055800	HindIII	Homo Sapiens	Fetal lung fibroblast IMR90	[65]
GSM1055805	HindIII	Homo Sapiens	Embryonic stem cells H1	[65]
GSM927075	HindIII	Homo Sapiens	ERG prostate epithelial cell line RWPE1	[71]
GSM1267196	HindIII	Homo Sapiens	Embryonic stem cells H1	[72]
GSM1267200	HindIII	Homo Sapiens	H1 Mesenchymal stem cells	[72]
GSM1294038	HindIII	Homo Sapiens	Breast cancer cell line T47D-MTVL, unstimulated	[73]
GSM1294039	HindIII	Homo Sapiens	Breast cancer cell line T47D-MTVL, progestin R5020-stimulated	[73]
GSM1608505	HindIII	Homo Sapiens	Lymphoblastoid GM12878	[74]
GSM1718021	HindIII	Homo Sapiens	Human Embryonic Stem Cells H9	[75]
GSM1906332	HindIII	Homo Sapiens	B-cell Follicular Lymphoma RL	[76]
GSM1906333	HindIII	Homo Sapiens	Primary TumorB-cell acute lymphocytic leukemia B-ALL	[76]
GSM1906334	HindIII	Homo Sapiens	MHH-CALL-4 B-cell acute lymphocytic leukemia CALL4 H1	[76]
GSM1909121	MboI	Homo Sapiens	Haploid fibroblast-like Hap1	[77]

Table 4.4: Datasets run by using HiCtool.

Chromosome	Length	Bin size	Number of bins
Chr 1	249 Mb	40 kb	6225
Chr 2	243 Mb	40 kb	6075
Chr 3	198 Mb	40 kb	4950
Chr 4	191 Mb	40 kb	4775
Chr 5	180 Mb	40 kb	4500
Chr 6	171 Mb	40 kb	4275
Chr 7	159 Mb	40 kb	3975
Chr 8	146 Mb	40 kb	3650
Chr 9	141 Mb	40 kb	3525
Chr 10	135 Mb	40 kb	3375
Chr 11	135 Mb	40 kb	3375
Chr 12	133 Mb	40 kb	3325
Chr 13	115 Mb	40 kb	2875
Chr 14	107 Mb	40 kb	2675
Chr 15	102 Mb	40 kb	2550
Chr 16	89 Mb	40 kb	2225
Chr 17	81 Mb	40 kb	2025
Chr 18	77 Mb	40 kb	1925
Chr 19	59 Mb	40 kb	1475
Chr 20	61 Mb	40 kb	1525
Chr 21	48 Mb	40 kb	1200
Chr 22	51 Mb	40 kb	1275
Chr X	155 Mb	40 kb	3875
Chr Y	57 Mb	40 kb	1425

TABLE 4.5: Chromosomes and matrices dimensions used in HiCtool. Chromosomes lengths in terms of Mega-base pairs, bin sizes (kilo-base pairs) and contact matrices dimensions in terms of number of bins for each chromosome used to run the datasets on Table 4.4.

Chromosome	**Length**	**Bin size**	98^{th} **percentile**
Chr 1	249 Mb	1 Mb	135
Chr 2	243 Mb	1 Mb	189
Chr 3	198 Mb	1 Mb	213
Chr 4	191 Mb	1 Mb	227
Chr 5	180 Mb	1 Mb	241
Chr 6	171 Mb	1 Mb	234
Chr 7	159 Mb	1 Mb	281
Chr 8	146 Mb	1 Mb	312
Chr 9	141 Mb	1 Mb	470
Chr 10	135 Mb	1 Mb	307
Chr 11	135 Mb	1 Mb	285
Chr 12	133 Mb	1 Mb	316
Chr 13	115 Mb	1 Mb	458
Chr 14	107 Mb	1 Mb	447
Chr 15	102 Mb	1 Mb	444
Chr 16	89 Mb	1 Mb	429
Chr 17	81 Mb	1 Mb	601
Chr 18	77 Mb	1 Mb	802
Chr 19	59 Mb	1 Mb	337
Chr 20	61 Mb	1 Mb	674
Chr 21	48 Mb	1 Mb	592
Chr 22	51 Mb	1 Mb	302
Chr X	155 Mb	1 Mb	152
Chr Y	57 Mb	1 Mb	266

TABLE 4.6: Human Embryonic Stem Cell genome data features. Chromosomes lengths in terms of Mega-base pairs, bin sizes (Mega-base pairs) and 98^{th} percentile of the non-zero data of normalized intrachromosomal contact matrices. (GEO accession number: GSM862723)

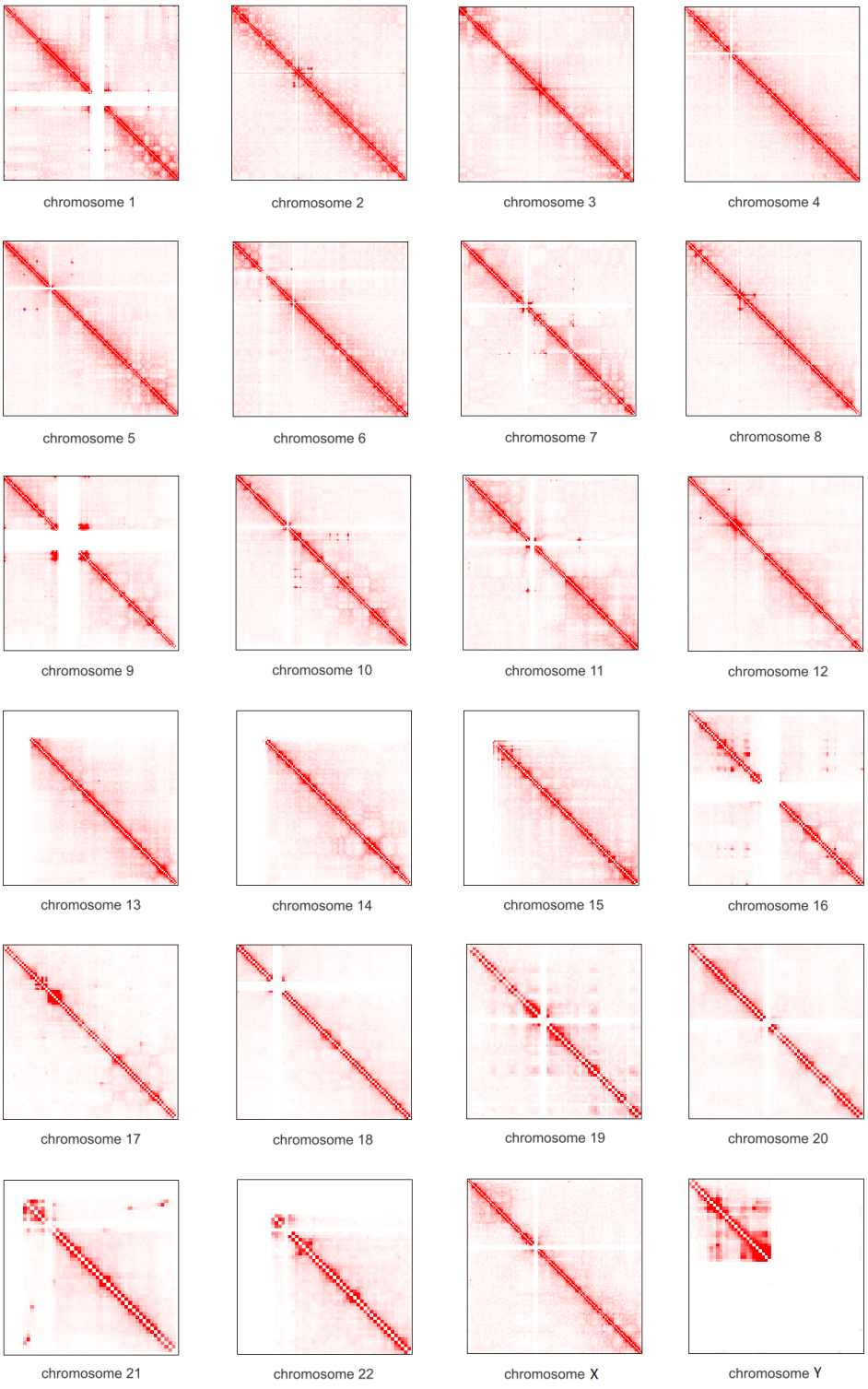

FIGURE 4.11: Human Embryonic Stem cells genome. Normalized intrachromosomal contact maps (1 Mb resolution), obtained using HiCtool, are shown for each chromosome. The dimensions of the maps are not proportional to the chromosomes length, but they were adjusted to have the same dimension of chromosome 1. All the maps display the 98^{th} percentile of the non-zero data (see Table 4.6 for more details). (GEO accession number: GSM862723)

4.3 GITAR: Genome Interaction Tools and Resources

We developed GITAR[3] (Genome Interaction Tools and Resources), a tool to provide the public a comprehensive solution to manage genomic interaction data from processing to storage and visualization. GITAR is based on three modules: HiCtool, processed data and GIVe (Genomic Interaction Visualizer).

The first module is HiCtool, a standardized Python library for processing and visualizing Hi-C data, which was entirely presented above and that we developed. HiCtool establishes a standardized, flexible and easy way to work on genomic interaction data.

The second module is a collection of datasets processed using HiCtool (see Table 4.4). The results include intrachromosomal contact matrices (observed, expected and normalized), Directionality Index (DI), HMM states for the DI and topological domains coordinates, which are stored into zip files available for downloading. In such a way, GITAR provides the largest collection of processed Hi-C data ever. This allows the users to make comparative analyses or manipulation. The actual collection of datasets will be updated as soon as more data are available.

The third module includes GIVe, a powerful web-based visualization engine for genomic interaction data, which clearly displays topological domains over the genome (developed by Dr. Xiaoyi Cao).

[3] http://www.genomegitar.org

Chapter 5

Discussion

Since nuclear organization has emerged as an important layer of epigenetic transcriptional regulation, during the last ten years a big effort has been produced to explore the three-dimensional architecture of chromatin and understand the mechanisms that regulate the DNA folding process. To investigate genomes conformation, several techniques have been used such as FISH and chromosome conformation capture (3C) or 3C derived assays (4C, 5C), until in 2009 the development of Hi-C first allowed for a comprehensive mapping of genome interactions. In such a way, Hi-C gave a significant contribute to the understanding of the principles of high order chromatin organization and its functional role in genome regulation.
In the following section, I present ongoing disease-associated studies, pointing out the big contribute that Hi-C can give in this field of research. Finally, I also introduce potential studies related to interchromosomal conformations based on Hi-C data, which could be significant for future research.

5.1 Disease-associated studies based on three-dimensional genome structure

The extent to which high-order chromatin aberrations are involved in cancer genomics is an important question. Chromosomal rearrangements, including translocations, are common pathogenetic events in cancers, such as leukemias, lymphomas and sarcomas. These events disrupt the integrity of the genome and require formation and joining of DNA double strand breaks (DSBs) [78]. The modeling of three-dimensional structure of chromatin as a fractal globule implies a strong relationship between contact probability and genomic distance, where interaction frequency scales as $s^{-1.08}$ between $\sim$500 kb and $\sim$7 Mb, although a similar behavior is evident for the entire length of a chromosome (s is

the genomic distance between two loci) [4]. In order of this, since insertions, deletions or translocations alter the distances between regions involved in such events, an unusually contact frequency appears compared to the reference sequence. Specifically for translocations, the new parts in contact would show a stronger intra- or inter-chromosomal interaction, which can be detected from the Hi-C maps.

Several studies investigated the hypothesis that three-dimensional spatial organization influences the set of somatic copy-number alterations (SCNAs) in cancer [79]. SCNAs are among the most common genomic alterations observed in human cancers, and the identification of regions that show frequent SCNAs is a robust way to find out key genes involved in oncogenesis [80]. Albeit the big amount of data on structural variation in cancer genomes, at first these studies were limited by the characterization of 3D chromatin architecture. Later, with the development of the Hi-C protocol, it was possible to compare SCNA maps and genome-wide maps, to prove the existence of a spatial relationship between the three-dimensional genome conformation and the chromosomal alterations in cancer. To quantitatively determined the relationship between 3D genomic structure and SCNA, Fundenberg *et al.* [79] performed a study converting both the datasets in the same form. For each chromosome, they built a Hi-C contact matrix, where each pixel stands for the number of spatial contacts between loci i and j, and a SCNA matrix across 3,131 tumors, where each pixel represents the number of amplifications or deletions that start at genomic location i and end at location j. From these maps it was seen that regions enriched for 3D interactions were more likely to undergo frequent SCNAs (see Figure 5.1). The achieved results also argued that the probability of a 3D contact between two loci based on the fractal globule model explains the length distribution of SCNAs better than other mechanistic models.

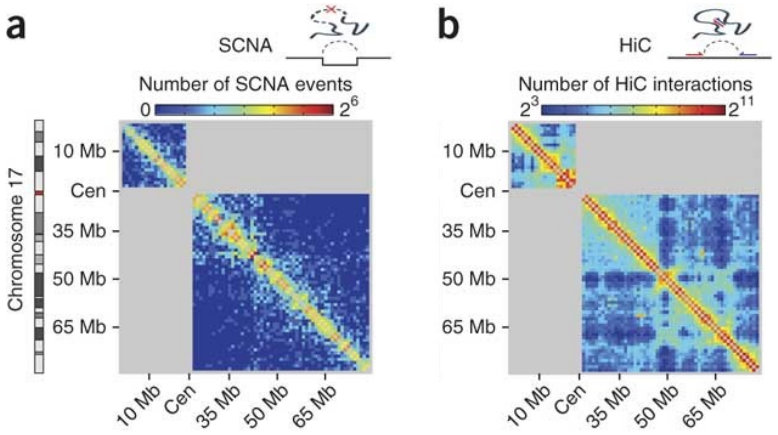

FIGURE 5.1: SCNA and contact maps for chromosome 17 at 1 Mb resolution [79]. **(a)** SCNA heatmap. Each pixel (i,j) is the number of SCNAs starting at genomic location i (vertical axis) and ending at location j (horizontal axis) on the same chromosome. **(b)** Hi-C contact heatmap. Each pixel (i,j) is the number of interactions between the loci i and j on the same chromosome.

During the last ten years, many GWAS (Genome-wide association studies) have been conducted to investigate genomic features of disease-associated SNPs (Single Nucleotide Polymorphisms) [81]. Specifically, Maurano *et al.* [82] showed that these variants are concentrated in regulatory DNA, marked by deoxyribonuclease I (DNase I) hypersensitive sites (DHSs). Since the results suggest an involvement of these variants in transcriptional regulatory mechanisms, including modulation of promoters and enhancers, associating them to their targets is crucial to understand how they affect gene function. In order to detect DHSs partners, Hi-C genome-wide protocol represents a powerful way to perform such analysis.

5.2 Future perspectives of Hi-C studies

During the last years, studies on functional implications of three-dimensional chromatin structure have been focused mostly on the intra-chromosomal contacts. As seen above, several analyses have been performed to understand the relationship between chromosomal interactions and transcriptional control processes. Dixon *et al.* [16] demonstrated the existence of topological domains throughout the genome and showed how topological domain boundaries are correlated with factors associated with active promoters and gene bodies (section 3.3). Lately, Rao *et al.* [29] carried out an analysis at higher resolution, exploiting in situ Hi-C protocol (section 3.4). They identified many chromatin loops, as spots of higher interactions in the heatmaps, which often link known enhancers and promoters and demarcate topological domains. This showed the strong association between loops and gene regulation, and the consistency with previous studies about topological domains and transcriptional control.

Besides intrachromosomal contacts, other studies showed that also inter-chromosomal interactions assume an important role in transcriptional regulation, allowing for example an enhancer to modulate the expression of a gene located in a different chromosome. In this case, can an enhancer activate either a *cis*- or a *trans*-promoter? If it can, how does an enhancer choose a target promoter when presented both in *cis* and *trans*? Bateman *et al.* [83] tried to answer these questions performing a study based on a transgenic approach on *Drosophila Melanogaster* using the enhancer GMR. They demonstrated that the enhancer can activate promoters in the same or different chromosomes and that promoters compete for the activity of an enhancer. Specificly, the enhancer was biased toward a promoter in *cis* than in *trans*, demonstrated by the strong reduction in *trans*-activation when a *cis*-promoter was present.

Since enhancers have been largely categorized in various cell types [24], promising studies could be performed using Hi-C data to understand how interchromosomal conformations

are involved in transcriptional regulation. The power of Hi-C to explore interchromosomal contacts at a high resolution would enable a comprehensive detection of *trans*-enhancers target regions.

A macroscopic study related to interchromosomal conformations could be conducted to discover if stable three-dimensional structures exist in the nucleus. To do this, interchromosomal Hi-C maps could be explored with the first goal of finding out potential patterns of high contacts. A high-contact pattern is intended as a defined region which shows enriched interaction frequencies with respect to the nearby loci. Then, a comparative analysis could be performed between different cell lines or conditions of a specie or even different species. From a preliminary analysis of the human interchromosomal maps that we computed (see Appendix B, section B.1), we derived the presence of at least two types of high-contact patterns, that we named "stripes pattern" and "spot pattern" (see Figure 5.2). We saw a correlation between the position of the patterns in an interchromosomal heatmap and the spatial organization of the *cis* maps of the two chromosomes involved. In particular, these patterns appear between loci which show high self-contacts in the intrachromosomal maps, meaning that they are generated by interactions between these "mega-domains" along each chromosome. If confirmed, the presence of stable interchromosomal interaction patterns may be the first step for studying the functional roles of *trans* contacts, which may pave the way to comprehensive and quantitative three-dimensional models of genome regulation.

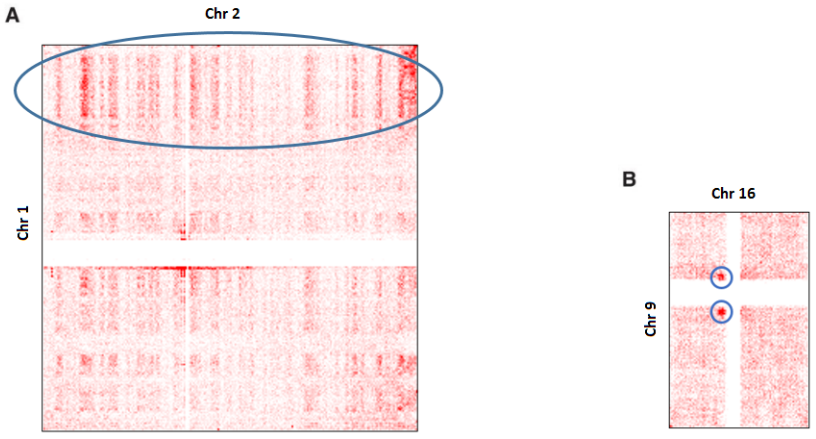

FIGURE 5.2: Interchromosomal patterns. **(A)** Interchromosomal heatmap of Chr 1: 0-250 Mb, Chr 2: 0:244 Mb at a bin size of 1 Mb. Range from 0 to 19 reads (GEO accession number: GSM1551550). Highlighted is what we called "stripes pattern". **(B)** Interchromosomal heatmap of Chr 9: 0-142 Mb, Chr 16: 0:90 Mb at a bin size of 1 Mb. Range from 0 to 8 reads (GEO accession number: GSM862723). Highlighted are those we called "spot patterns". Both the heatmaps show the 98^{th} percentile of the non-zero data.

5.3 Final summary

In this thesis we made a comprehensive study about the three-dimensional architecture of chromatin, from the physical structure to its role in transcriptional regulation and the techniques to explore it. We particularly focused on Hi-C, a novel method which first allowed for a genome-wide mapping of long-range interactions, showing both the Hi-C computational analysis and the main following studies performed by using this protocol.
In Chapter 4 we introduced HiCtool, the pipeline we developed for a standardized processing and visualization of Hi-C data and topological domains analysis. We showed its ability to handle Hi-C data in each step of the processing and how it is easy to get the results, even for a beginner user. We also clearly explained every step of the analysis, either the code and the reasoning behind the choice of each specific parameter of the processing. This makes HiCtool a really powerful and complete software for analyzing and visualizing intrachromosomal Hi-C data.
Then, we built GITAR (Genome Interaction Tools and Resources), that is our solution to work on and manage Hi-C interaction data. With GITAR, we provide to the public either a standardized pipeline to process Hi-C data (HiCtool) and the first exhaustive collection of processed datasets ever, which allows for the first time users to work on and compare different datasets in a consistent way. Actually, eighteen datasets of Homo Sapiens (hg38) and two datasets of mouse (mm10) are ready for downloading, and our goal is to process even more datasets in the near future.
Finally, we have highlighted the contribute of Hi-C in studies related to common pathogenetic conditions, like cancers. This could potentially have a role in understanding what goes wrong in cancer cells and which therapies could have the best probability of success.

Appendix A

HiCtool sources

A.1 Preprocessing of the data

This section requires the software: SRA Toolkit, Bowtie 2, SAMTools and Bedtools.

Before preprocessing the data with the following *.bash* file, the index for the reference genome has to be already built:

```
bowtie2-build ref.fa index
```

`ref.fa` is the *.fasta* file of the reference sequence (we used hg38). This step has to be performed only once, for each reference genome.

To automatically preprocess the data, the following *.bash* file can be used:

```
# 2. Converting from sra format to fastq format

for i in $(ls *.sra); do
    fastq-dump $i --split-3
done

rm *.sra


# 3. Mapping paired-end reads to a reference sequence

num1=0

for i in $(ls *_1.fastq); do
    num1=$(( $num1 + 1))
    string1+=$i,
done
```

```
length1=${#string1}-1
mate1s=${string1:0:$length1}

num2=0

for i in $(ls *_2.fastq); do
    num2=$(( $num2 + 1))
    string2+=$i,
done

length2=${#string2}-1
mate2s=${string2:0:$length2}

bowtie2 -p 8 -x index -1 $mate1s -2 $mate2s -S HiCfile.sam

rm *.fastq
samtools view -bS HiCfile.sam > HiCfile.bam
rm *.sam
samtools sort -m 5000000000 HiCfile.bam HiCfile.sort
rm HiCfile.bam


# 4. Removing PCR duplicates from the bam file

samtools sort -m 5000000000 -n HiCfile.sort.bam HiCfile.namesort
rm HiCfile.sort.bam

samtools fixmate HiCfile.namesort.bam HiCfile.fixmate_namesort.bam
rm HiCfile.namesort.bam

samtools sort -m 5000000000 HiCfile.fixmate_namesort.bam HiCfile.fixmate_sort
rm HiCfile.fixmate_namesort.bam

samtools rmdup HiCfile.fixmate_sort.bam HiCfile_noDup.sort.bam
rm HiCfile.fixmate_sort.bam


# 5. Splitting the bam file to separate the two reads in a pair

samtools view -h -f 0x40 HiCfile_noDup.sort.bam > HiCfile_pair1.bam
samtools view -h -f 0x80 HiCfile_noDup.sort.bam > HiCfile_pair2.bam
```

The code to create the fragment-end *.bed* file was not included in the *.bash* file because this part has to be run only once for each restriction enzyme (HindIII in this case) and a specific reference genome:

```
echo -e "@HindIII\nAAGCTT\n+\nIIIIII" > HindIII.fastq

bowtie2 -p 8 -k 3000000 -x index -U HindIII.fastq -S restrictionsites.sam
samtools view -bS restrictionsites.sam > restrictionsites.bam
bedtools bamtobed -i restrictionsites.bam > restrictionsites.bed
```

A.2 Data analysis and visualization

This section requires python libraries: math, numpy, matplotlib, matplotlib.pyplot, PIL.
This section requires python package: HiFive [67].

A.2.1 HiFive functions

To learn the correction parameters with HiFive, the following code is used:

```
import hifive

# Creating a Fend object
fend = hifive.Fend('fend_object.hdf5', mode='w')
fend.load_fends('HindIII_hg38_gc.bed', re_name='HindIII', format='bed')
fend.save()

# Creating a HiCData object
data = hifive.HiCData('HiC_data_object.hdf5', mode='w')
data.load_data_from_bam('fend_object.hdf5',
                        ['HiCfile_pair1.bam','HiCfile_pair2.bam'],
                        maxinsert=500)
data.save()

# Creating a HiC Project object
hic = hifive.HiC('HiC_project_object.hdf5', 'w')
hic.load_data('HiC_data_object.hdf5')
hic.save()

# Filtering HiC fends
hic = hifive.HiC('HiC_project_object.hdf5')
hic.filter_fends(mininteractions=1, mindistance=500000, maxdistance=0)
hic.save()

# Finding HiC distance function
hic = hifive.HiC('HiC_project_object.hdf5')
hic.find_distance_parameters(numbins=90, minsize=200, maxsize=0)
hic.save('HiC_distance_function.hdf5')
```

```
# Learning the correction model
hic = hifive.HiC('HiC_distance_function.hdf5')
hic.find_binning_fend_corrections(max_iterations=1000,
                                  mindistance=500000,
                                  maxdistance=0,
                                  num_bins=[20,20],
                                  model=['len','gc'],
                                  parameters=['even','even'],
                                  usereads='cis',
                                  learning_threshold=1.0)
hic.save('HiC_norm_binning.hdf5')
```

A.2.2 Normalizing the data

To normalize the data, we need the observed data and the correction parameters to remove the biases to obtain the corrected read counts. In order to perform this, we calculate the observed contact matrix and the fend expected contact matrix. In addition, we calculate also the enrichment expected contact matrix to compute the observed over expected enrichment values, considering also the distance between fends. For each chromosome, the following five matrices are computed at a bin size of 40 kb. Every contact matrix is saved in txt format using the function `save_matrix`.

- The **observed data** contain the observed reads count for each bin.
- The **fend expected data** contain the learned correction value to remove biases related to fends for each bin.
- The **enrichment expected data** contain the expected reads count for each bin, considering the distance between fends and the learned correction parameters.
- The **normalized fend data** contain the corrected reads count for each bin.
- The **normalized enrichment data** contain the enrichment value (O/E) for each bin.

Before calculating the contact matrices, the function to save the data and importing numpy are needed:

```
import numpy as np
```

```
def save_matrix(n, matrix, out_file):
    """
    Function to save a square matrix in a .txt file.
    The matrix is reshaped by rows and saved in a vector.
    Inputs:
        n:          number of rows (or columns) of the matrix.
        matrix:     input matrix to be saved.
        out_file:   output file name in .txt format
    Output:
        .txt file containing the saved list
    """
    vect = []
    for row in xrange(n):
        for col in xrange(n):
            vect.append(matrix[row,col])

    with open (out_file,'w') as fout:
        for i in xrange(n**2):
            fout.write('%s\n' %vect[i])
```

To calculate and save the observed data and expected enrichment data the following code is used:

```
import hifive

# hg38
chromosomes = {'1':249250621,
               '2':243199373,
               '3':198022430,
               '4':191154276,
               '5':180915260,
               '6':171115067,
               '7':159138663,
               '8':146364022,
               '9':141213431,
               '10':135534747,
               '11':135006516,
               '12':133851895,
               '13':115169878,
```

```
                '14':107349540,
                '15':102531392,
                '16':89354753,
                '17':81195210,
                '18':77077248,
                '19':59128983,
                '20':61025520,
                '21':48129895,
                '22':51304566,
                'X':155270560,
                'Y':57373566}

ch = '6'
chromosome = 'chr' + ch
bin_size = 40000
start_pos = 0
end_pos = (chromosomes[ch]/1000000)*1000000

start_part = str(float(start_pos)/float(1000000))
end_part = str(float(end_pos)/float(1000000))
binsize_str = str(float(bin_size)/float(1000000))

# Enrichment data
hic = hifive.HiC('HiC_norm_binning.hdf5')
heatmap_enrich = hic.cis_heatmap(chrom=chromosome,
                                 start=start_pos,
                                 stop=end_pos,
                                 binsize=bin_size,
                                 arraytype='full',
                                 datatype='enrichment')

# Observed data
observed = heatmap_enrich[:,:,0] # observed contact data
n = len(observed)
save_matrix(n, observed, 'HiCtool_observed_contact_matrix_' + chromosome +
'_' + binsize_str + 'mb_' + start_part + 'mb_' + end_part + 'mb.txt')

# Expected enrichment data (fend corrections and distance property)
expected_enrich = heatmap_enrich[:,:,1] # expected enrichment contact data
```

```
n = len(expected_enrich)
save_matrix(n, expected_enrich, 'HiCtool_expected_enrich_contact_matrix_' +
chromosome + '_' + binsize_str + 'mb_' + start_part + 'mb_' +
end_part + 'mb.txt')
```

Before calculating the fend expected data, we need to calculate the raw expected data, which contain the number of possible fend interactions. We need this to scale the fend expected data by the mean fend pairs in each bin. To calculate the scaling factor, the following code is used:

```
import hifive

# Raw data
hic = hifive.HiC('HiC_norm_binning.hdf5')
heatmap_raw = hic.cis_heatmap(chrom=chromosome,
                              start=start_pos,
                              stop=end_pos,
                              binsize=bin_size,
                              arraytype='full',
                              datatype='raw')

# Expected raw (number of possible fend interactions)
expected_raw = heatmap_raw[:,:,1]
n = len(expected_raw)
scaling_factor = (np.sum(expected_raw)/2)/(n*(n-1)/2)
# mean fend pairs in each bin
```

To calculate and save the expected fend data (i.e. the correction values for fend biases) the code is the following:

```
import hifive

# Fend data
hic = hifive.HiC('HiC_norm_binning.hdf5')
heatmap_fend = hic.cis_heatmap(chrom=chromosome,
                               start=start_pos,
                               stop=end_pos,
                               binsize=bin_size,
                               arraytype='full',
```

```
                                    datatype='fend')

# Expected fend (fend corrections)
expected_fend = heatmap_fend[:,:,1]/scaling_factor # fend correction values
n = len(expected_fend)
save_matrix(n, expected_fend, 'HiCtool_expected_fend_contact_matrix_' +
chromosome + '_' + binsize_str + 'mb_' + start_part + 'mb_' +
end_part + 'mb.txt')
```

In the above calls, all valid possible interactions are queried from chromosome 'chrom' between 'start' and 'stop' parameters. The 'arraytype' parameter determines what shape of array data are returned in: 'full' returns a square, symmetric array of size NxNx2. The 'datatype' parameter specifies which kind of data to extract. The observed counts are in the first index of the last dimension of the returned array (the same for every 'datatype'), while the expected counts are in the second index of the last dimension. To normalize and save the data, the code is the following:

```
# Normalized fend contact matrix
normalized_fend = np.zeros((n,n))
for i in xrange(n):
    for j in xrange(n):
        if expected_fend[i][j] == 0:
            normalized_fend[i][j] = 0
        else:
            normalized_fend[i][j] = float(observed[i][j])/
            float(expected_fend[i][j])

save_matrix(n, normalized_fend, 'HiCtool_normalized_fend_contact_matrix_' +
chromosome + '_' + binsize_str + 'mb_' + start_part + 'mb_' +
end_part + 'mb.txt')

# Normalized enrichment contact matrix
normalized_enrich = np.zeros((n,n))
for i in xrange(n):
    for j in xrange(n):
        if expected_enrich[i][j] == 0:
            normalized_enrich[i][j] = 0
        else:
            normalized_enrich[i][j] = float(observed[i][j])/
```

```
            float(expected_enrich[i][j])

save_matrix(n, normalized_enrich, 'HiCtool_normalized_enrich_contact_matrix_' +
chromosome + '_' + binsize_str + 'mb_' + start_part + 'mb_' +
end_part + 'mb.txt')
```

A.2.3 Visualizing the normalized data

The following code is to plot heatmaps and histograms of the data.
Before plotting, we need the functions to load the data and generate the colorbar, and importing the modules we are using:

```
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import numpy as np
import math
from PIL import Image

def load_matrix(input_file):
    """
    Function to load a list by a .txt file
    Input:
        input_file:   input file name in .txt format
    Output:
        output_matrix:  array containing all the reshaped values stored
        in the input .txt file
    """

    import numpy as np

    with open (input_file,'r') as infile:
        lines = infile.readlines()
        matrix_vect = []
        for i in lines:
            j = i[:-1]
            matrix_vect.append(float(j))

        matrix_vect_size = len(matrix_vect)
```

```
        matrix_size = int(np.sqrt(matrix_vect_size))
        output_matrix = np.reshape(matrix_vect,(matrix_size,matrix_size))
        return output_matrix

def make_cmap(colors):
    '''
    make_cmap takes a list of tuples which contain RGB values and returns a
    cmap with equally spaced colors.
    Arrange your tuples so that the first color is the lowest value for the
    colorbar and the last is the highest.
    '''
    import matplotlib as mpl
    import numpy as np

    bit_rgb = np.linspace(0,1,256)
    position = np.linspace(0,1,len(colors))
    for i in range(len(colors)):
        colors[i] = (bit_rgb[colors[i][0]],
                     bit_rgb[colors[i][1]],
                     bit_rgb[colors[i][2]])
    cdict = {'red':[], 'green':[], 'blue':[]}
    for pos, color in zip(position, colors):
        cdict['red'].append((pos, color[0], color[0]))
        cdict['green'].append((pos, color[1], color[1]))
        cdict['blue'].append((pos, color[2], color[2]))
    cmap = mpl.colors.LinearSegmentedColormap('my_colormap',cdict,256)
    return cmap
```

In the following code, only a part of chromosome 6, from 50 to 54 Mb at a bin size of 40 kb is plotted. To plot the entire chromosome, commenting the lines of the code below "selecting a part" is needed.
First, the heatmap, histogram and colorbar for the fend normalized data are plotted. To have a better visualization (i.e. setting a proper heatmap color range), only the 98^{th} percentile of the non-zero data is plotted. Then, in order to obtain a heatmap whose colors span from white (RGB[255,255,255]) to red (RGB[255,0,0]), the data are normalized between 0 and 255 before plotting. To plot and save the heatmap in png format, the code is the following:

```
# hg38
chromosomes = {'1':249250621,
               '2':243199373,
               '3':198022430,
               '4':191154276,
               '5':180915260,
               '6':171115067,
               '7':159138663,
               '8':146364022,
               '9':141213431,
               '10':135534747,
               '11':135006516,
               '12':133851895,
               '13':115169878,
               '14':107349540,
               '15':102531392,
               '16':89354753,
               '17':81195210,
               '18':77077248,
               '19':59128983,
               '20':61025520,
               '21':48129895,
               '22':51304566,
               'X':155270560,
               'Y':57373566}

ch = '6'
chromosome = 'chr' + ch
bin_size = 40000
start_pos = 0
end_pos = (chromosomes[ch]/1000000)*1000000

start_part = str(float(start_pos)/float(1000000))
end_part = str(float(end_pos)/float(1000000))
binsize_str = str(float(bin_size)/float(1000000))

# Plotting of the fend normalized data
matrix_data_full = load_matrix('HiCtool_normalized_fend_contact_matrix_' +
chromosome + '_' + binsize_str + 'mb_' + start_part + 'mb_' +
```

```
end_part + 'mb.txt')

# Selecting a part
start_coord = 50000000
end_coord = 54000000
start_bin = start_coord/bin_size
end_bin = end_coord/bin_size

start_part = str(float(start_coord)/float(1000000))
end_part = str(float(end_coord)/float(1000000))
matrix_data_full = matrix_data_full[start_bin:end_bin+1,start_bin:end_bin+1]
###################

n = len(matrix_data_full)
output_vect = np.reshape(matrix_data_full,n*n,1)
non_zero = np.nonzero(output_vect)
perc = np.percentile(output_vect[non_zero[0]],98)
for i in xrange(len(matrix_data_full)):
    for j in xrange(len(matrix_data_full)):
        if matrix_data_full[i][j] > perc:
            matrix_data_full[i][j] = perc

# Heatmap
max_value = np.max(matrix_data_full)
min_value = np.min(matrix_data_full)
norm_matrix_data_full = 255-((matrix_data_full-min_value)/
(max_value-min_value))*255

img = Image.new('RGB',(n,n))
newData = []
for i in xrange(n):
    for j in xrange(n):
        value = int(norm_matrix_data_full[i][j])
        newData.append((255,value,value))

img.putdata(newData)
img.save('HiCtool_normalized_fend_contact_matrix_' + chromosome + '_' +
binsize_str + 'mb_' + start_part + 'mb_' + end_part + 'mb.png','PNG')
```

To plot and save the histogram, the following code is used:

```
histogram = []
k = 1
for i in xrange(n):
    row = matrix_data_full[i][k:]
    for j in row:
        histogram.append(j)
    k += 1

plt.close("all")
histogram_bins = int(pow(len(histogram),0.3))
plt.hist(histogram, bins=histogram_bins)
plt.title('Contact frequency histogram')
plt.xlabel('Number of contacts')
plt.ylabel('Frequency')
plt.savefig('HiCtool_normalized_fend_histogram_' + chromosome + '_' +
binsize_str + 'mb_' + start_part + 'mb_' + end_part + 'mb.png')
```

To plot and save the colorbar, the following code is used:

```
bar_min = min(histogram)
bar_max = max(histogram)
colors = []
for i in xrange(256):
    color = (255,255-i,255-i)
    colors.append(color)

plt.close("all")
fig = plt.figure(figsize=(1.5, 7))
ax = fig.add_axes([0.3, 0.08, 0.4, 0.9])
cmap = make_cmap(colors)
norm = matplotlib.colors.Normalize(vmin=bar_min, vmax=bar_max)
cb = matplotlib.colorbar.ColorbarBase(ax, cmap=cmap,
                                      norm=norm,
                                      orientation='vertical')
plt.savefig('HiCtool_normalized_fend_colorbar_' + chromosome + '_' +
binsize_str + 'mb_' + start_part + 'mb_' + end_part + 'mb.png')
```

Now, the heatmap, histogram and colorbar for the enrichment normalized data are plotted. The log_2 of the data is plotted, to quantify the positive enrichment (red) and the negative enrichment (blue). The zero values before performing the log_2 are shown in gray. The 99^{th} percentile of the data, computed either on positive logs and negative logs is plotted. To plot and save the heatmap in png format, the following code is used:

```
# hg38
chromosomes = {'1':249250621,
               '2':243199373,
               '3':198022430,
               '4':191154276,
               '5':180915260,
               '6':171115067,
               '7':159138663,
               '8':146364022,
               '9':141213431,
               '10':135534747,
               '11':135006516,
               '12':133851895,
               '13':115169878,
               '14':107349540,
               '15':102531392,
               '16':89354753,
               '17':81195210,
               '18':77077248,
               '19':59128983,
               '20':61025520,
               '21':48129895,
               '22':51304566,
               'X':155270560,
               'Y':57373566}

ch = '6'
chromosome = 'chr' + ch
bin_size = 40000
start_pos = 0
end_pos = (chromosomes[ch]/1000000)*1000000

start_part = str(float(start_pos)/float(1000000))
```

```
end_part = str(float(end_pos)/float(1000000))
binsize_str = str(float(bin_size)/float(1000000))

matrix_data_full = load_matrix('HiCtool_normalized_enrich_contact_matrix_' +
chromosome + '_' + binsize_str + 'mb_' + start_part + 'mb_' +
end_part + 'mb.txt')

# Selecting a part
start_coord = 50000000
end_coord = 54000000
start_bin = start_coord/bin_size
end_bin = end_coord/bin_size

start_part = str(float(start_coord)/float(1000000))
end_part = str(float(end_coord)/float(1000000))
matrix_data_full = matrix_data_full[start_bin:end_bin+1,start_bin:end_bin+1]
##################

n = len(matrix_data_full)
output_vect = np.reshape(matrix_data_full,n*n,1)
non_zero = np.nonzero(output_vect)
non_zero_values = output_vect[non_zero[0]]
positive_logs = []
negative_logs = []
negative_logs_abs = []
zero_logs = []
for i in non_zero_values:
    log_value = math.log(i,2)
    if log_value == 0:
        zero_logs.append(log_value)
    if log_value > 0:
        positive_logs.append(log_value)
    if log_value < 0:
        negative_logs.append(log_value)
        negative_logs_abs.append(abs(log_value))

max_value = np.percentile(positive_logs,99)
min_value = np.percentile(negative_logs_abs,99)
min_value = -min_value
```

```
# Heatmap
img = Image.new('RGB',(n,n))
newData = []
for i in xrange(n):
    for j in xrange(n):
        value = matrix_data_full[i][j]
        if value==0:
            newData.append((100,100,100))
            continue
        log_value = math.log(value,2)
        if log_value < 0:
            if log_value < min_value: log_value = min_value
            color_value = int(((log_value-min_value)/(abs(min_value)))*255)
            newData.append((color_value,color_value,255))
        if log_value >= 0:
            if log_value > max_value: log_value = max_value
            color_value = int((log_value/max_value)*255)
            newData.append((255,255-color_value,255-color_value))

img.putdata(newData)
img.save('HiCtool_normalized_enrich_contact_matrix_' + chromosome + '_' +
binsize_str + 'mb_' + start_part + 'mb_' + end_part + 'mb.png','PNG')
```

To plot and save the histogram, the following code is used:

```
for i in xrange(len(positive_logs)):
    if positive_logs[i] > max_value:
        positive_logs[i] = max_value
for i in xrange(len(negative_logs)):
    if negative_logs[i] < min_value:
        negative_logs[i] = min_value

logs_values = positive_logs + negative_logs + zero_logs
s = set([x for x in logs_values if logs_values.count(x) > 1])
for i in s:
    c = logs_values.count(i)
    for j in xrange(c/2):
        logs_values.remove(i)
```

```
plt.close("all")
histogram_bins = int(pow(len(logs_values),0.3))
plt.hist(logs_values, bins=histogram_bins)
plt.title('Enrichment histogram')
plt.xlabel('log2(O/E)')
plt.ylabel('Frequency')
plt.savefig('HiCtool_normalized_enrich_histogram_' + chromosome + '_' +
binsize_str + 'mb_' + start_part + 'mb_' + end_part + 'mb.png')
```

To plot and save the colorbars (red for positive logs and blue for negative logs), the following code is used:

```
# Positive logs
bar_min = 0
bar_max = max_value
colors = []
for i in xrange(256):
    color = (255,255-i,255-i)
    colors.append(color)

plt.close("all")
fig = plt.figure(figsize=(1.5, 3.5))
ax = fig.add_axes([0.3, 0.08, 0.4, 0.8])
cmap = make_cmap(colors)
norm = matplotlib.colors.Normalize(vmin=bar_min, vmax=bar_max)
cb = matplotlib.colorbar.ColorbarBase(ax, cmap=cmap,
                                      norm=norm,
                                      orientation='vertical')
plt.savefig('HiCtool_normalized_enrich_colorbar_red_' + chromosome + '_' +
binsize_str + 'mb_' + start_part + 'mb_' + end_part + 'mb.png')

# Negative logs
bar_min = min_value
bar_max = 0
colors = []
for i in xrange(256):
    color = (i,i,255)
    colors.append(color)
```

```
plt.close("all")
fig = plt.figure(figsize=(1.5, 3.5))
ax = fig.add_axes([0.3, 0.08, 0.4, 0.8])
cmap = make_cmap(colors)
norm = matplotlib.colors.Normalize(vmin=bar_min, vmax=bar_max)
cb = matplotlib.colorbar.ColorbarBase(ax, cmap=cmap,
                                      norm=norm,
                                      orientation='vertical')
plt.savefig('HiCtool_normalized_enrich_colorbar_blue_' + chromosome + '_' +
binsize_str + 'mb_' + start_part + 'mb_' + end_part + 'mb.png')
```

A.3 Topological Domains analysis

This section requires python package: HiFive [67].
This section requires the software MATLAB.

A.3.1 Calculating the observed DI

Before calculating the DI values, the functions to load the contact data and save the output are needed:

```
def save_vector(n, vect, out_file):
    """
    Function to save a list in a .txt file
    Inputs:
        n:          number of elements of the list to save
        vect:       name of the list to save
        out_file:   output file name in .txt format
    Output:
        .txt file containing the saved list
    """
    with open (out_file,'w') as fout:
        for i in xrange(n):
            fout.write('%s\n' %vect[i])


def load_matrix(input_file):
```

```
    """
    Function to load a list by a .txt file
    Input:
        input_file:   input file name in .txt format
    Output:
        output_matrix:  array containing all the reshaped values stored
        in the input .txt file
    """

    import numpy as np

    with open (input_file,'r') as infile:
        lines = infile.readlines()
        matrix_vect = []
        for i in lines:
            j = i[:-1]
            matrix_vect.append(float(j))

        matrix_vect_size = len(matrix_vect)
        matrix_size = int(np.sqrt(matrix_vect_size))
        output_matrix = np.reshape(matrix_vect,(matrix_size,matrix_size))
        return output_matrix
```

First, the contact data for the selected chromosome are loaded:

```
# hg38
chromosomes = {'1':249250621,
               '2':243199373,
               '3':198022430,
               '4':191154276,
               '5':180915260,
               '6':171115067,
               '7':159138663,
               '8':146364022,
               '9':141213431,
               '10':135534747,
               '11':135006516,
               '12':133851895,
               '13':115169878,
```

```
                '14':107349540,
                '15':102531392,
                '16':90354753,
                '17':81195210,
                '18':78077248,
                '19':59128983,
                '20':63025520,
                '21':48129895,
                '22':51304566,
                'X':155270560,
                'Y':59373566}

ch = '6'
chromosome = 'chr' + ch
bin_size = 40000
start_pos = 0
end_pos = (chromosomes[ch]/1000000)*1000000

start_part = str(float(start_pos)/float(1000000))
end_part = str(float(end_pos)/float(1000000))
binsize_str = str(float(bin_size)/float(1000000))

contact_matrix = load_matrix('HiCtool_normalized_fend_contact_matrix_' +
chromosome + '_' + binsize_str + 'mb_' + start_part + 'mb_' +
end_part + 'mb.txt')
n = contact_matrix.shape[0]
```

Now the observed DI values are calculated according to the formula 3.1, and then saved into a *.txt* file using the function `save_vector`:

```
DI = []
len_var = 2000000/40000

for locus in xrange(n):
    if locus < len_var:
        A = sum(contact_matrix[locus][:locus])
        B = sum(contact_matrix[locus][locus+1:locus+len_var+1])
    elif locus >= n-len_var:
        A = sum(contact_matrix[locus][locus-len_var:locus])
```

```
        B = sum(contact_matrix[locus][locus+1:])
    else:
        A = sum(contact_matrix[locus][locus-len_var:locus])
        B = sum(contact_matrix[locus][locus+1:locus+len_var+1])

    E = (A+B)/2

    try:
        di = ((B-A)/(abs(B-A)))*((((A-E)**2)/E)+(((B-E)**2)/E))
    except ZeroDivisionError:
        di = 0

    DI.append(di)

save_vector(n,DI,'HiCtool_' + str(chromosome) + '_DI.txt')
```

Now we have all the DI values of every bin saved into `chrN_DI.txt`.

A.3.2 Calculating the true DI states using a HMM

After selecting an input file (`chrN_DI.txt`) containing the DI values of a chromosome, the code allows to insert the start base position, the end base position and the bin size for the following analysis and to plot the DI distributions. The calculation of the true DI is implemented using the HMM functions included into the Statistics and Machine Learning Toolbox of MATLAB[1]. For true DI calculation, we consider the Emission Sequence as the observed DI values and the Transition Matrix, Emission Matrix and initial State Sequence as unknown. We have three states 1, 2, 3 corresponding to a positive (1), negative (2) or zero (3) value of the true DI. In our analysis, we associate to the state '3' all the absolute DI values under a threshold (default 0.4). So, first we estimate both the Transition and the Emission matrices, and then the most probable sequence of states.

To load the data and select the region of interest, the MATLAB code is the following:

```
chromosome = '6';
input_string = strcat('HiCtool_chr',chromosome,'_DI.txt');
f = fopen(input_string);
formatSpec = '%f';
A = fscanf(f,formatSpec);
```

[1] http://it.mathworks.com/help/stats/hidden-markov-models-hmm.html

```
bin_size = int_binsize_value;
start_pos = int_base_start;
end_pos = int_base_end;
start_index = round(start_pos/bin_size);
end_index = round(end_pos/bin_size);
x = (start_pos:bin_size:end_pos);

DI_part = A(start_index:end_index);
```

`DI_part` contains the observed DI values of the selected part to be plotted.
Since the initial sequence of states is unknown, we can guess the initial Transition and Emission matrices and estimate them using the function `hmmtrain` of the HMM package.

First, each DI value is associated to the corresponding state:

```
seq = zeros(1,length(A));
zero_threshold = 0.4;

for i = (1:length(seq))
    if A(i) >= zero_threshold
        seq(i) = 1;
    elseif A(i) <= -zero_threshold
        seq(i) = 2;
    else
        seq(i) = 3;
    end
end
```

`seq` contains the sequence of states associated to each DI value.
Now the Transition and Emission matrices are estimated using the function `hmmtrain`. This function calculates maximum likelihood estimates of transition and emission probabilities from a sequence of emissions. It uses an iterative algorithm that alters `TRANS_GUESS` and `EMIS_GUESS` so that at each step the adjusted matrices are more likely to generate the observed sequence `seq`. The algorithm halts when the matrices in two successive iterations are within "tolerance" of each other.

The code that was used to perform the estimation is the following:

```
% Initial guessed Transition Matrix
A_pp = 0.4;
A_pn = 0.3;
A_pz = 0.3;

A_np = 0.3;
A_nn = 0.4;
A_nz = 0.3;
```

```
A_zp = 0.3;
A_zn = 0.3;
A_zz = 0.4;

TRANS_GUESS = [A_pp, A_pn, A_pz; A_np, A_nn, A_nz; A_zp, A_zn, A_zz];

% Initial guessed Emission Matrix
B_pp = 0.4;
B_pn = 0.3;
B_pz = 0.3;

B_np = 0.3;
B_nn = 0.4;
B_nz = 0.3;

B_zp = 0.3;
B_zn = 0.3;
B_zz = 0.4;

EMIS_GUESS = [B_pp, B_pn, B_pz; B_np, B_nn, B_nz; B_zp, B_zn, B_zz];

% 'hmmtrain' function
[TRANS_EST, EMIS_EST] = hmmtrain(seq, TRANS_GUESS, EMIS_GUESS, 'tolerance', '
    0.00001');
```

Given the estimated Transition and Emission matrices, we use the Viterbi algorithm to compute the most likely sequence of states the model would go through to generate a given sequence of emissions. The function `hmmviterbi` performs this estimation, giving as output `likelystates` that is the most likely produced sequence of states. Then, the output vector of the HMM states is saved and two custom values are assigned to the positive and negative states to give a good visual feedback in the plot to identify topological domains.

```
likelystates = hmmviterbi(seq, TRANS_EST, EMIS_EST);

HMM_string = strcat('HMM_states_chr',chromosome,'.txt');
output_file = fopen(HMM_string,'w');
fprintf(output_file,'%d\n',likelystates');
fclose(output_file);

DI_true = zeros(1,length(seq));

for i = (1:length(seq))
    if likelystates(i) == 1
        DI_true(i) = min(DI_part)-12;
    elseif likelystates(i) == 2
        DI_true(i) = min(DI_part)-15;
    else
        DI_true(i) = 0;
    end
end
```

```
DI_true_part = DI_real(start_index:end_index);
```

The following is the code used for the identification of the topological domains coordinates (see Figure 4.9). The result is saved in `topological_domains_chrN.txt`.

```
% Start coordinates of the domains
k1 = 1;
for i = (2:length(seq))
    if (likelystates(i) == 1 && likelystates(i-1) == 2) || (likelystates(i) == 1
    && likelystates(i-1) == 3)
        p(k1) = i * bin_size;
        k1 = k1 + 1;
    end
end

% End coordinates of the domains
k2 = 1;
for i = (2:length(seq))
    if (likelystates(i) == 2 && likelystates(i+1) == 1) || (likelystates(i) == 2
    && likelystates(i+1) == 3)
        n(k2) = i * bin_size;
        k2 = k2 + 1;
    end
end

k = 1;
p1 = 1;
n1 = 1;
p2 = 2;
n2 = 2;

% Step 1: checking if the first negative values are greater than the first
    positive value.
while n(n1) < p(p1)
    n1 = n1 + 1;
    n2 = n2 + 1;
end
% Now we have removed all the first negative values before the first positive one
    .

while p1 < length(p) && n1 < length(n)

    % Step 2: checking if there are two consecutive positive values.
    while n(n1) > p(p2) && p2 < length(p)
        p2 = p2 + 1;
    end
    % Now we have removed the possible gaps between consecutive positive states.

    % Step 3: checking if there are two consecutive negative values.
    while n(n2) < p(p2) && n2 < length(n)
        n1 = n1 + 1;
        n2 = n2 + 1;
```

```
    end
    % Now we have removed the possible gaps between consecutive negative states.

    % Step 4: identification of the Topological Domain.
    topological_domains(k,1) = p(p1);
    topological_domains(k,2) = n(n1);

    k = k + 1;
    p1 = p2;
    n1 = n2;
    p2 = p1 + 1;
    n2 = n1 + 1;

end

% Saving of the output on a .txt file. This result can be loaded again using '
    import data' option.

topological_domains_string = strcat('topological_domains_chr',chromosome,'.txt');
output_file1 = fopen(topological_domains_string,'w');
fprintf(output_file1,'%16s\t%14s\n','Start coordinate','End coordinate');
fprintf(output_file1,'%9i\t\t\t%9i\n',topological_domains');
fclose(output_file1);
```

Finally, this is the code to plot both the observed DI and the "true DI":

```
pidx = find(DI_part>0);
nidx = find(DI_part<0);

pidx_true = find(DI_true_part==min(DI_part)-12);
nidx_true = find(DI_true_part==min(DI_part)-15);

figure,
suptitle(strcat('Directionality Index Chr',chromosome)),
bar(x(pidx),DI_part(pidx),'r'),hold on
bar(x(nidx),DI_part(nidx),'FaceColor',[0.0 0.5 0.0]),hold on
plot(x(pidx_true),DI_true_part(pidx_true),'>','Color','r','LineWidth',3),hold on
plot(x(nidx_true),DI_true_part(nidx_true),'<','Color',[0.0 0.5 0.0],'LineWidth'
    ,3),grid on
xlabel('Base coordinates'),legend('Positive DI (downstream biases)','Negative DI
    (upstream biases)','Positive HMM state','Negative HMM state')
```

Appendix B

Other sources

B.1 Interchromosomal maps

This is the code to generate, save and plot observed interchromosomal Hi-C maps derived from a standardized data processing.
In order to generate interchromosomal heatmaps for a specific datasets, the same steps listed in the section A.2 has to be followed until the object `HiC_distance_funtion.hdf5` is obtained. Then, to learn the correction parameters the code is the following:

```
import hifive

hic = hifive.HiC('HiC_distance_function.hdf5')
hic.find_binning_fend_corrections(max_iterations=1000,
                                  mindistance=500000,
                                  maxdistance=0,
                                  num_bins=[20,20],
                                  model=['len','gc'],
                                  parameters=['even','even'],
                                  usereads='trans',
                                  learning_threshold=1.0)
hic.save('HiC_norm_binning_trans.hdf5')
```

The function `find_binning_fend_corrections` allows to learn the correction parameters for trans contacts, setting the parameter `usereads='trans'`.
Before creating the heatmap, the function to save the data is needed:

```
def save_rectangular_matrix(n, m, matrix, out_file):
    """
    Function to save a rectangular matrix in a .txt file.
    The matrix is reshaped by rows and saved in a vector.
    Inputs:
        n:          number of rows
        m:          number of columns
        matrix:     input matrix to be saved
        out_file:   output file name in .txt format
    Output:
        .txt file containing the saved list
    """
    vect = []
    for row in xrange(n):
        for col in xrange(m):
            vect.append(matrix[row,col])

    with open (out_file,'w') as fout:
        for i in xrange(n*m):
            fout.write('%s\n' %vect[i])
```

Then, an interchromosomal heatmap is generated and plotted (chr1-chr2 in the code):

```
import hifive
import numpy as np
from PIL import Image

# hg38
chromosomes = {'1':249250621,
               '2':243199373,
               '3':198022430,
               '4':191154276,
               '5':180915260,
               '6':171115067,
               '7':159138663,
               '8':146364022,
               '9':141213431,
               '10':135534747,
               '11':135006516,
```

```
                '12':133851895,
                '13':115169878,
                '14':107349540,
                '15':102531392,
                '16':89354753,
                '17':81195210,
                '18':77077248,
                '19':59128983,
                '20':61025520,
                '21':48129895,
                '22':51304566,
                'X':155270560,
                'Y':57373566}

ch1 = '1'
ch2 = '2'
bin_size = 1000000
binsize_str = str(float(bin_size)/float(1000000))

hic = hifive.HiC('HiC_norm_binning_trans.hdf5')
heatmap = hic.trans_heatmap('chr' + ch1, 'chr' + ch2,
                            start1=0, stop1=chromosomes[ch1],
                            startfend1=None, stopfend1=None,
                            binbounds1=None,
                            start2=0, stop2=chromosomes[ch2],
                            startfend2=None, stopfend2=None,
                            binbounds2=None,
                            binsize=bin_size,
                            datatype='enrichment')

matrix_data_full = heatmap[:,:,0]

row = matrix_data_full.shape[0]
col = matrix_data_full.shape[1]
row_str = str(row)
col_str = str(col)

filename = 'chr' + ch1 +'_chr' + ch2 + '_' + binsize_str + 'mb_' + row_str +
'x' + col_str + '_observed'
```

```
save_rectangular_matrix(row, col, matrix_data_full, filename + '.txt')

output_vect = np.reshape(matrix_data_full,row*col,1)
non_zero = np.nonzero(output_vect)
perc = np.percentile(output_vect[non_zero[0]],98)
for i in xrange(row):
    for j in xrange(col):
        if matrix_data_full[i][j] > perc:
            matrix_data_full[i][j] = perc

max_value = np.max(matrix_data_full)
min_value = np.min(matrix_data_full)
norm_matrix_data_full = 255-((matrix_data_full-min_value)/
(max_value-min_value))*255

row = norm_matrix_data_full.shape[0]
col = norm_matrix_data_full.shape[1]

img = Image.new('RGB',(col,row))
newData = []
for i in xrange(row):
    for j in xrange(col):
        value = int(norm_matrix_data_full[i][j])
        newData.append((255,value,value))

img.putdata(newData)
img.save(filename + '.png','PNG')
```

The function `trans_heatmap` allows to generate an interchromosomal heatmap object, selecting the two chromosomes involved, the start and end coordinates and the bin size. Like for the intrachromosomal heatmaps, here we plot the 98^{th} percentile of the non-zero data.

Bibliography

[1] James D Watson, Francis HC Crick, et al. "Molecular structure of nucleic acids". In: *Nature* 171.4356 (1953), pp. 737–738.

[2] Tom Misteli. "Beyond the sequence: cellular organization of genome function". In: *Cell* 128.4 (2007), pp. 787–800.

[3] Job Dekker. "Gene regulation in the third dimension". In: *Science* 319.5871 (2008), pp. 1793–1794.

[4] Erez Lieberman-Aiden, Nynke L van Berkum, Louise Williams, Maxim Imakaev, Tobias Ragoczy, Agnes Telling, Ido Amit, Bryan R Lajoie, Peter J Sabo, Michael O Dorschner, et al. "Comprehensive mapping of long-range interactions reveals folding principles of the human genome". In: *science* 326.5950 (2009), pp. 289–293.

[5] Job Dekker, Marc A Marti-Renom, and Leonid A Mirny. "Exploring the three-dimensional organization of genomes: interpreting chromatin interaction data". In: *Nature Reviews Genetics* 14.6 (2013), pp. 390–403.

[6] Dorothy Buck. *Applications of Knot Theory: American Mathematical Society, Short Course, January 4-5, 2008, San Diego, California*. Vol. 66. American Mathematical Soc., 2009.

[7] Anirban Ghosh and Manju Bansal. "A glossary of DNA structures from A to Z". In: *Biological Crystallography* 59.4 (2003), pp. 620–626.

[8] Peter Yakovchuk, Ekaterina Protozanova, and Maxim D Frank-Kamenetskii. "Base-stacking and base-pairing contributions into thermal stability of the DNA double helix". In: *Nucleic acids research* 34.2 (2006), pp. 564–574.

[9] Karolin Luger, Armin W Mäder, Robin K Richmond, David F Sargent, and Timothy J Richmond. "Crystal structure of the nucleosome core particle at 2.8 Å resolution". In: *Nature* 389.6648 (1997), pp. 251–260.

[10] Frado Woodcock. "Structure of the 30 nm chromatin fiber". In: (1986).

[11] Sergei A Grigoryev and Christopher L Woodcock. "Chromatin organization—The 30nm fiber". In: *Experimental cell research* 318.12 (2012), pp. 1448–1455.

[12] Galip Gürkan Yardımcı and William Stafford Noble. "Predictive model of 3D domain formation via CTCF-mediated extrusion". In: *Proceedings of the National Academy of Sciences* 112.47 (2015), pp. 14404–14405.

[13] Elphège P Nora, Bryan R Lajoie, Edda G Schulz, Luca Giorgetti, Ikuhiro Okamoto, Nicolas Servant, Tristan Piolot, Nynke L van Berkum, Johannes Meisig, John Sedat, et al. "Spatial partitioning of the regulatory landscape of the X-inactivation centre". In: *Nature* 485.7398 (2012), pp. 381–385.

[14] Darren J Burgess. "Chromosomes: Dynamically in the loop". In: *Nature Reviews Genetics* 15.7 (2014), pp. 440–440.

[15] Tom Sexton, Eitan Yaffe, Ephraim Kenigsberg, Frédéric Bantignies, Benjamin Leblanc, Michael Hoichman, Hugues Parrinello, Amos Tanay, and Giacomo Cavalli. "Three-dimensional folding and functional organization principles of the Drosophila genome". In: *Cell* 148.3 (2012), pp. 458–472.

[16] Jesse R Dixon, Siddarth Selvaraj, Feng Yue, Audrey Kim, Yan Li, Yin Shen, Ming Hu, Jun S Liu, and Bing Ren. "Topological domains in mammalian genomes identified by analysis of chromatin interactions". In: *Nature* 485.7398 (2012), pp. 376–380.

[17] Wendy A Bickmore and Bas van Steensel. "Genome architecture: domain organization of interphase chromosomes". In: *Cell* 152.6 (2013), pp. 1270–1284.

[18] Lars Guelen, Ludo Pagie, Emilie Brasset, Wouter Meuleman, Marius B Faza, Wendy Talhout, Bert H Eussen, Annelies de Klein, Lodewyk Wessels, Wouter de Laat, et al. "Domain organization of human chromosomes revealed by mapping of nuclear lamina interactions". In: *Nature* 453.7197 (2008), pp. 948–951.

[19] Giancarlo Bonora, Kathrin Plath, and Matthew Denholtz. "A mechanistic link between gene regulation and genome architecture in mammalian development". In: *Current opinion in genetics & development* 27 (2014), pp. 92–101.

[20] Kazutoshi Takahashi and Shinya Yamanaka. "Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors". In: *cell* 126.4 (2006), pp. 663–676.

[21] Matthew Denholtz, Giancarlo Bonora, Constantinos Chronis, Erik Splinter, Wouter de Laat, Jason Ernst, Matteo Pellegrini, and Kathrin Plath. "Long-range chromatin contacts in embryonic stem cells reveal a role for pluripotency factors and polycomb proteins in genome organization". In: *Cell stem cell* 13.5 (2013), pp. 602–616.

[22] Thomas Cremer and Christoph Cremer. "Chromosome territories, nuclear architecture and gene regulation in mammalian cells". In: *Nature reviews genetics* 2.4 (2001), pp. 292–301.

[23] 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.

[24] ENCODE Project Consortium et al. "An integrated encyclopedia of DNA elements in the human genome". In: *Nature* 489.7414 (2012), pp. 57–74.

[25] Stefan Schoenfelder, Ieuan Clay, and Peter Fraser. "The transcriptional interactome: gene expression in 3D". In: *Current opinion in genetics & development* 20.2 (2010), pp. 127–133.

[26] Guillaume J Filion, Joke G van Bemmel, Ulrich Braunschweig, Wendy Talhout, Jop Kind, Lucas D Ward, Wim Brugman, Inês J de Castro, Ron M Kerkhoven, Harmen J Bussemaker, et al. "Systematic protein location mapping reveals five principal chromatin types in Drosophila cells". In: *Cell* 143.2 (2010), pp. 212–224.

[27] Albin Sandelin, Piero Carninci, Boris Lenhard, Jasmina Ponjavic, Yoshihide Hayashizaki, and David A Hume. "Mammalian RNA polymerase II core promoters: insights from genome-wide studies". In: *Nature Reviews Genetics* 8.6 (2007), pp. 424–436.

[28] Jennifer EF Butler and James T Kadonaga. "The RNA polymerase II core promoter: a key component in the regulation of gene expression". In: *Genes & development* 16.20 (2002), pp. 2583–2592.

[29] Suhas SP Rao, Miriam H Huntley, Neva C Durand, Elena K Stamenova, Ivan D Bochkov, James T Robinson, Adrian L Sanborn, Ido Machol, Arina D Omer, Eric S Lander, 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.

[30] Won Ju Yun, Yea Woon Kim, Yujin Kang, Jungbae Lee, Ann Dean, and AeRi Kim. "The hematopoietic regulator TAL1 is required for chromatin looping between the β-globin LCR and human γ-globin genes to activate transcription". In: *Nucleic acids research* 42.7 (2014), pp. 4283–4293.

[31] Bing He, Changya Chen, Li Teng, and Kai Tan. "Global view of enhancer–promoter interactome in human cells". In: *Proceedings of the National Academy of Sciences* 111.21 (2014), E2191–E2199.

[32] Francisco J Iborra, Ana Pombo, Dean A Jackson, and Peter R Cook. "Active RNA polymerases are localized within discrete transcription "factories' in human nuclei". In: *Journal of cell science* 109.6 (1996), pp. 1427–1436.

[33] David S Latchman. "Transcription factors: an overview". In: *The international journal of biochemistry & cell biology* 29.12 (1997), pp. 1305–1312.

[34] Heidi Sutherland and Wendy A Bickmore. "Transcription factories: gene expression in unions?" In: *Nature Reviews Genetics* 10.7 (2009), pp. 457–466.

[35] Jennifer A Mitchell and Peter Fraser. "Transcription factories are nuclear subcompartments that remain in the absence of transcription". In: *Genes & development* 22.1 (2008), pp. 20–25.

[36] Annette Moter and Ulf B Göbel. "Fluorescence in situ hybridization (FISH) for direct visualization of microorganisms". In: *Journal of microbiological methods* 41.2 (2000), pp. 85–112.

[37] I Solovei, J Walter, M Cremer, F Habermann, L Schermelleh, and T Cremer. "FISH on three-dimensionally preserved nuclei". In: *FISH: a practical approach* (2002), pp. 119–157.

[38] Lothar Schermelleh, Rainer Heintzmann, and Heinrich Leonhardt. "A guide to super-resolution fluorescence microscopy". In: *The Journal of cell biology* 190.2 (2010), pp. 165–175.

[39] Rainer Heintzmann and Christoph G Cremer. "Laterally modulated excitation microscopy: improvement of resolution by using a diffraction grating". In: *BiOS Europe'98*. International Society for Optics and Photonics. 1999, pp. 185–196.

[40] Job Dekker, Karsten Rippe, Martijn Dekker, and Nancy Kleckner. "Capturing chromosome conformation". In: *Science* 295.5558 (2002), pp. 1306–1311.

[41] Josée Dostie and Job Dekker. "Mapping networks of physical interactions between genomic elements using 5C technology". In: *Nature protocols* 2.4 (2007), pp. 988–1002.

[42] Zhihu Zhao, Gholamreza Tavoosidana, Mikael Sjölinder, Anita Göndör, Piero Mariano, Sha Wang, Chandrasekhar Kanduri, Magda Lezcano, Kuljeet Singh Sandhu, Umashankar Singh, et al. "Circular chromosome conformation capture (4C) uncovers extensive networks of epigenetically regulated intra-and interchromosomal interactions". In: *Nature genetics* 38.11 (2006), pp. 1341–1347.

[43] Harmen JG van de Werken, Gilad Landan, Sjoerd JB Holwerda, Michael Hoichman, Petra Klous, Ran Chachik, Erik Splinter, Christian Valdes-Quezada, Yuva Öz, Britta AM Bouwman, et al. "Robust 4C-seq data analysis to screen for regulatory DNA interactions". In: *Nature methods* 9.10 (2012), pp. 969–972.

[44] Josée Dostie, Todd A Richmond, Ramy A Arnaout, Rebecca R Selzer, William L Lee, Tracey A Honan, Eric D Rubio, Anton Krumm, Justin Lamb, Chad Nusbaum, 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.

[45] Jan P Schouten, Cathal J McElgunn, Raymond Waaijer, Danny Zwijnenburg, Filip Diepvens, and Gerard Pals. "Relative quantification of 40 nucleic acid sequences by multiplex ligation-dependent probe amplification". In: *Nucleic acids research* 30.12 (2002), e57–e57.

[46] Jingyao Zhang, Huay Mei Poh, Su Qin Peh, Yee Yen Sia, Guoliang Li, Fabianus Hendriyan Mulawadi, Yufen Goh, Melissa J Fullwood, Wing-Kin Sung, Xiaoan Ruan, et al. "ChIA-PET analysis of transcriptional chromatin interactions". In: *Methods* 58.3 (2012), pp. 289–299.

[47] Iouri Chepelev, Gang Wei, Dara Wangsa, Qingsong Tang, and Keji Zhao. "Characterization of genome-wide enhancer-promoter interactions reveals co-expression of interacting genes and modes of higher order chromatin organization". In: *Cell research* 22.3 (2012), pp. 490–503.

[48] Michael Grunstein. "Histone acetylation in chromatin structure and transcription". In: *Nature* 389.6649 (1997), pp. 349–352.

[49] JR Paulson and SS Taylor. "Phosphorylation of histones 1 and 3 and nonhistone high mobility group 14 by an endogenous kinase in HeLa metaphase chromosomes." In: *Journal of Biological Chemistry* 257.11 (1982), pp. 6064–6072.

[50] Louis Levinger and Alexander Varshavsky. "Selective arrangement of ubiquitinated and D1 protein-containing nucleosomes within the Drosophila genome". In: *Cell* 28.2 (1982), pp. 375–385.

[51] Heng Li, Jue Ruan, and Richard Durbin. "Mapping short DNA sequencing reads and calling variants using mapping quality scores". In: *Genome research* 18.11 (2008), pp. 1851–1858.

[52] Ben Langmead and Steven L Salzberg. "Fast gapped-read alignment with Bowtie 2". In: *Nature methods* 9.4 (2012), pp. 357–359.

[53] Nynke L Van Berkum, Erez Lieberman-Aiden, Louise Williams, Maxim Imakaev, Andreas Gnirke, Leonid A Mirny, Job Dekker, and Eric S Lander. "Hi-C: a method to study the three-dimensional architecture of genomes". In: (2010).

[54] Jenny A Croft, Joanna M Bridger, Shelagh Boyle, Paul Perry, Peter Teague, and Wendy A Bickmore. "Differences in the localization and morphology of chromosomes in the human nucleus". In: *The Journal of cell biology* 145.6 (1999), pp. 1119–1131.

[55] Shelagh Boyle, Susan Gilchrist, Joanna M Bridger, Nicola L Mahy, Juliet A Ellis, and Wendy A Bickmore. "The spatial organization of human chromosomes within the nuclei of normal and emerin-mutant cells". In: *Human molecular genetics* 10.3 (2001), pp. 211–219.

[56] Hideyuki Tanabe, Felix A Habermann, Irina Solovei, Marion Cremer, and Thomas Cremer. "Non-random radial arrangements of interphase chromosome territories: evolutionary considerations and functional implications". In: *Mutation Research/-Fundamental and Molecular Mechanisms of Mutagenesis* 504.1 (2002), pp. 37–45.

[57] Job Dekker. "Mapping in vivo chromatin interactions in yeast suggests an extended chromatin fiber with regional variation in compaction". In: *Journal of biological chemistry* 283.50 (2008), pp. 34532–34540.

[58] Christian Münkel and Jörg Langowski. "Chromosome structure predicted by a polymer model". In: *Physical Review E* 57.5 (1998), p. 5888.

[59] A Grosberg, Y Rabin, S Havlin, and A Neer. "Crumpled globule model of the three-dimensional structure of DNA". In: *EPL (Europhysics Letters)* 23.5 (1993), p. 373.

[60] BB Mandelbrot. "The fractal geometry of natureFreeman and Company". In: *New York* (1983).

[61] Jennifer E Phillips and Victor G Corces. "CTCF: master weaver of the genome". In: *Cell* 137.7 (2009), pp. 1194–1211.

[62] Tony Kouzarides. "Chromatin modifications and their function". In: *Cell* 128.4 (2007), pp. 693–705.

[63] Katherine E Cullen, Michael P Kladde, and Mark A Seyfred. "Interaction between transcription regulatory regions of prolactin chromatin". In: *Science* 261.5118 (1993), pp. 203–206.

[64] Reza Kalhor, Harianto Tjong, Nimanthi Jayathilaka, Frank Alber, and Lin Chen. "Genome architectures revealed by tethered chromosome conformation capture and population-based modeling". In: *Nature biotechnology* 30.1 (2012), pp. 90–98.

[65] Fulai Jin, Yan Li, Jesse R Dixon, Siddarth Selvaraj, Zhen Ye, Ah Young Lee, Chia-An Yen, Anthony D Schmitt, Celso A Espinoza, and Bing Ren. "A high-resolution map of the three-dimensional chromatin interactome in human cells". In: *Nature* 503.7475 (2013), pp. 290–294.

[66] Eitan Yaffe and Amos Tanay. "Probabilistic modeling of Hi-C contact maps eliminates systematic biases to characterize global chromosomal architecture". In: *Nature genetics* 43.11 (2011), pp. 1059–1065.

[67] Michael EG Sauria, Jennifer E Phillips-Cremins, Victor G Corces, and James Taylor. "HiFive: a tool suite for easy and efficient HiC and 5C data analysis". In: *Genome biology* 16.1 (2015), pp. 1–10.

[68] Ming Hu, Ke Deng, Siddarth Selvaraj, Zhaohui Qin, Bing Ren, and Jun S Liu. "HiCNorm: removing biases in Hi-C data via Poisson regression". In: *Bioinformatics* 28.23 (2012), pp. 3131–3133.

[69] Maxim Imakaev, Geoffrey Fudenberg, Rachel Patton McCord, Natalia Naumova, Anton Goloborodko, Bryan R Lajoie, Job Dekker, and Leonid A Mirny. "Iterative correction of Hi-C data reveals hallmarks of chromosome organization". In: *Nature methods* 9.10 (2012), pp. 999–1003.

[70] Nicolas Servant, Bryan R Lajoie, Elphège P Nora, Luca Giorgetti, Chong-Jian Chen, Edith Heard, Job Dekker, and Emmanuel Barillot. "HiTC: exploration of high-throughput 'C'experiments". In: *Bioinformatics* 28.21 (2012), pp. 2843–2844.

[71] David S Rickman, T David Soong, Benjamin Moss, Juan Miguel Mosquera, Jan Dlabal, Stéphane Terry, Theresa Y MacDonald, Joseph Tripodi, Karen Bunting, Vesna Najfeld, et al. "Oncogene-mediated alterations in chromatin conformation". In: *Proceedings of the National Academy of Sciences* 109.23 (2012), pp. 9083–9088.

[72] Jesse R Dixon, Inkyung Jung, Siddarth Selvaraj, Yin Shen, Jessica E Antosiewicz-Bourget, Ah Young Lee, Zhen Ye, Audrey Kim, Nisha Rajagopal, Wei Xie, et al. "Chromatin architecture reorganization during stem cell differentiation". In: *Nature* 518.7539 (2015), pp. 331–336.

[73] François Le Dily, Davide Baù, Andy Pohl, Guillermo P Vicent, François Serra, Daniel Soronellas, Giancarlo Castellano, Roni HG Wright, Cecilia Ballare, Guillaume Filion, et al. "Distinct structural transitions of chromatin topological domains correlate with coordinated hormone-induced gene regulation". In: *Genes & development* 28.19 (2014), pp. 2151–2162.

[74] Adrian L Sanborn, Suhas SP Rao, Su-Chen Huang, Neva C Durand, Miriam H Huntley, Andrew I Jewett, Ivan D Bochkov, Dharmaraj Chinnappan, Ashok Cutkosky, Jian Li, 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.

[75] Takashi Nagano, Csilla Várnai, Stefan Schoenfelder, Biola-Maria Javierre, Steven W Wingett, and Peter Fraser. "Comparison of Hi-C results using in-solution versus in-nucleus ligation". In: *Genome biology* 16.1 (2015), pp. 1–13.

[76] Zheng Wang, Renzhi Cao, Kristen Taylor, Aaron Briley, Charles Caldwell, Jianlin Cheng, et al. "The properties of genome conformation and spatial gene interaction and regulation networks of normal and malignant human cell types". In: *PloS one* 8.3 (2013), e58793.

[77] Fabian Grubert, Judith B Zaugg, Maya Kasowski, Oana Ursu, Damek V Spacek, Alicia R Martin, Peyton Greenside, Rohith Srivas, Doug H Phanstiel, Aleksandra Pekowska, et al. "Genetic control of chromatin states in humans involves local and distal chromosomal interactions". In: *Cell* 162.5 (2015), pp. 1051–1065.

[78] Isaac A Klein, Wolfgang Resch, Mila Jankovic, Thiago Oliveira, Arito Yamane, Hirotaka Nakahashi, Michela Di Virgilio, Anne Bothmer, Andre Nussenzweig, Davide F Robbiani, et al. "Translocation-capture sequencing reveals the extent and nature of chromosomal rearrangements in B lymphocytes". In: *Cell* 147.1 (2011), pp. 95–106.

[79] Geoff Fudenberg, Gad Getz, Matthew Meyerson, and Leonid A Mirny. "High order chromatin architecture shapes the landscape of chromosomal alterations in cancer". In: *Nature biotechnology* 29.12 (2011), pp. 1109–1113.

[80] Rameen Beroukhim, Craig H Mermel, Dale Porter, Guo Wei, Soumya Raychaudhuri, Jerry Donovan, Jordi Barretina, Jesse S Boehm, Jennifer Dobson, Mitsuyoshi Urashima, et al. "The landscape of somatic copy-number alteration across human cancers". In: *Nature* 463.7283 (2010), pp. 899–905.

[81] Lucia A Hindorff, Praveen Sethupathy, Heather A Junkins, Erin M Ramos, Jayashri P Mehta, Francis S Collins, and Teri A Manolio. "Potential etiologic and functional implications of genome-wide association loci for human diseases and traits". In: *Proceedings of the National Academy of Sciences* 106.23 (2009), pp. 9362–9367.

[82] Matthew T Maurano, Richard Humbert, Eric Rynes, Robert E Thurman, Eric Haugen, Hao Wang, Alex P Reynolds, Richard Sandstrom, Hongzhu Qu, Jennifer Brody, et al. "Systematic localization of common disease-associated variation in regulatory DNA". In: *Science* 337.6099 (2012), pp. 1190–1195.

[83] Jack R Bateman, Justine E Johnson, and Melissa N Locke. "Comparing enhancer action in cis and in trans". In: *Genetics* 191.4 (2012), pp. 1143–1155.