Universitat Politècnica de Catalunya

# Identification of spatial communities in the human genome graph to better understand HIV insertion

Master of Science thesis

in partial Fulfillments for the Master of
**Telecommunications Engineering**

at the
**Universitat Politècnica de Catalunya**

*Author:*
Ricardo García Gutiérrez

*Advisors:*
Dra. Alba Pagès Zamora
Pere Giménez Febrer

June 24, 2019

# Contents

**Abstract** **9**

**1 The 3D genome architecture** **11**
1.1 The spatial organization principles of the genome . . . . . 11
1.1.1 Nuclear DNA . . . . . 11
1.1.2 Decoding the structure of the genome . . . . . 13
1.1.3 Hi-C: High Throughput 3C technology . . . . . 14
1.2 Finding structure in the 3D genome with Hi-C data analysis . . . . . 14
1.2.1 The Hi-C matrix . . . . . 14
1.2.2 Normalization of Hi-C data . . . . . 17
1.2.3 A/B compartments . . . . . 18
1.2.4 Topological associating domains . . . . . 19

**2 Graphs, Graph learning and Spectral Clustering** **21**
2.1 Graph theory . . . . . 21
2.1.1 The Laplacian matrix . . . . . 22
2.1.2 Spectral properties of the Laplacian . . . . . 23
2.2 Graph learning . . . . . 23
2.2.1 Constrained edge pattern . . . . . 24
2.2.2 Automatic parameter selection . . . . . 25
2.3 Spectral clustering . . . . . 25
2.3.1 Spectral clustering analysis . . . . . 28

**3 Clustering pipeline of Hi-C contact maps** **29**
3.1 Hi-C Jurkat cell data and HIV insertion hotspots . . . . . 30
3.2 Pre-processing Hi-C data . . . . . 31
3.3 A/B clustering at Mega base pairs scale . . . . . 32
3.4 Clustering methods . . . . . 35
3.5 Case study: Graph learning applied to chromosome 16 . . . . . 37
3.5.1 Edge mask selection . . . . . 37
3.5.2 Learned weighted adjacency matrix of the intra-chromosomal Hi-C map . . 41

3.6 3D spatial clustering pipeline . . . . . 41
3.6.1 Dimensionality reduction . . . . . 42
3.6.2 Genome wide clustering . . . . . 43
3.6.3 Results . . . . . 45

**4 Conclusions and future work** **51**

# List of Figures


1.1 The spatial configuration and hierarchial structure inside the cell nucleous . . . . . 12
1.2 The genome sequencing high level procedure . . . . . 13
1.3 Overview of the HiC technology. A) Hi-C detects chromatin interaction both within and between chromosomes by covalently crosslinking protein/DNA complexes with formaldehyde. B) The chromatin is digested with a restriction enzyme and the ends are marked with a biotinylated nucleotide. C) The DNA in the crosslinked complexes are ligated to form chimeric DNA molecules. D) Biotin is removed from the ends of linear fragments and the molecules are fragmented to reduce their overall size. E) Molecules with internal biotin incorporation are pulled down with streptavidin coated magnetic beads and modified for deep sequencing. Quantitation of chromatin interactions is achieved through massively parallel deep sequencing. . 15
1.4 Log-scale Intra-chromosomal map of interactions from a particular cell population. The x and y axes represent loci in genomic order and each pixel represents the number of observed interactions between them. . . . . 16
1.5 Distance decay pattern of the interaction frequency in Hi-C data. . . . . 18
1.6 The landscape of structures detected in intra-chromosomal HiC data at a mega base scale . . . . . 19

2.1 A toy graph with a partition into two disjoints subsets A and B. . . . . 26

3.1 Length of each chromosome in genomic bins units . . . . . 30
3.2 Raw Hi-C map of chromosome 14 . . . . . 31
3.3 Log-scale processed Hi-C map after applying O/E normalization and percentile saturation . . . . . 32
3.4 Pre-processed Hi-C matrix of chromosome 16 at 1 Mbps resolution . . . . . 34
3.5 Pearson correlation matrix of the pre-processed Hi-C map of chromosome 14 . . . 34
3.6 The first PC and the Pearson correlation matrix aligned . . . . . 35
3.7 Graph theoretic approach for modelling Hi-C contacts . . . . . 36
3.8 Normalized intra-chromosomal contact matrix $\mathbf{U}$ of chromosome 16 . . . . . 38
3.9 Normalized pairwise Euclidean distance matrix $\mathbf{Z}$ . . . . . 38
3.10 The Pearson correlation matrix for chromosome 16 . . . . . 39
3.11 The histogram of the Pearson correlation matrix for chromosome 16 . . . . . 40

3.12 Smoothed weighted adjacency matrix for different graph sparsity levels . . . . . 41
3.13 Clustering strategy yielding genome wide communities over the set of $N_{chr}$ chromosomes . . . . . 42
3.14 Log-scale heatmap of the inter-chromosomal Hi-C density score matrix . . . . . 43
3.15 Heatmap of the Pearson correlation matrix of $\mathbf{W}_G$ . . . . . 44
3.16 Heatmap of the learned inter-chromosomal adjacency matrix . . . . . 45
3.17 Spectral clustering based on Pearson correlation: k-means cost function evaluation 46
3.18 Spectral clustering based on graph learning: k-means cost function evaluation . . . 46
3.19 Silhouette analysis for both spectral methods . . . . . 47
3.20 Eigenvalue spectrum of the laplacian matrix in the graph learning based clustering 48
3.21 Clusters node distribution for $K_2 = 5$ ordered by community size . . . . . 49
3.22 A/B score distribution for the Correlation based spectral clustering . . . . . 49
3.23 A/B score distribution for the Graph learning based spectral clustering . . . . . 49

# List of Tables


3.1 Sparsity ratio of the edge mask for different Pearson coefficients based thresholds . 40
3.2 Value of $k_s$ selected for different ranges of chromosomes . . . . . 41
3.3 Pearson correlation based spectral clustering: HIV insertions average density and medians of A/B scores for each genome wide cluster label . . . . . 50
3.4 Graph learning based spectral clustering: HIV insertion average density and medians of A/B scores for each genome wide cluster label . . . . . 50

# Abstract

In this work, the 3D spatial organization of a human Jurkat cell, an immune cell who is one of the main targets of the human immunodeficiency virus (HIV), is analyzed through the clustering of genome interactions networks provided by the Hi-C data, a 3D massive sequencing technology capable of quantifying interactions among regions of the genome inside the nucleus of a cell. The data analysis approach consists on a graph theoretic modelling of these networks and the clustering analysis is performed by the use of spectral clustering methods, a family of clustering techniques based on the spectral decomposition of Laplacian matrices of graph networks. By inferring the 3D structure of the Jurkat cell at the nuclear scale, the distribution of HIV integration sites on the Jurkat genome is analyzed and contrasted with the current knowledge of the the integration mechanisms and their relationship with the 3D genomic context. The clustering results are also evaluated through a common set of metrics, which serve to objectively asses the 3D structure of the nucleus of the Jurkat cell. With the proposed data analysis, the main findings are: the 3D spatial structure is not prominent, the global interaction genomic network contains just a few communities and the insertion pattern of HIV, contrasted on the detected communities, confirms the established knowledge of HIV integration mechanisms.

# Chapter 1

# The 3D genome architecture

In this chapter, the biology basic concepts needed to understand this work are introduced. The chapter is structured as follows: A quick review of the spatial organization principles of the 3D genome is given, then the 3D sequencing technology that allows to sample the spatial structure of the genome is introduced, and finally, methods for analyzing 3D sequencing data are presented.

## 1.1 The spatial organization principles of the genome

### 1.1.1 Nuclear DNA

Deoxyribonucleic acid (DNA) is a molecule that contains the information needed for a organism to develop, function and reproduce. These DNA molecules are found in every individual or organism cells. DNA consists of a series of smaller organic molecules called nucleotides. In each nucleotide, there are four types of nitrogen bases: adenine (A), thymine (T), guanine (G) and cytosine (C). The order of these bases determines the genetic information or genes, which are the set of essential instructions for a cell's functioning. A gene is a segment of DNA that is transcribed, i.e. that is converted to ribonucleic acid (RNA), another type of acid nucleic molecule, either to be used as is (structural, catalytic, regulatory RNA, ...) or to guide the synthesis of a protein. Most DNA is located in the cell nucleus, where it is referred to as nuclear DNA. In eukaryotic cells, the large amounts of DNA required to encode all the information needed to sustain cellular life, are packaged into chromosomes. Chromosomes are very long double-stranded DNA molecules found in the cell nucleous, as depicted in Figure 1.1. On the other side, prokaryotic cells typically carry their genes on a single, circular DNA molecule called *bacterial chromosome*, found in the cytoplasm.

The nuclear DNA in human species is packed into 24 different chromosomes, each consisting of a fine thread of DNA and a set of proteins that fold and pack it into a compact structure. Such complex of protein and DNA is called chromatin. The main function of the chromosomes is to carry the genes. The total genetic information carried by all the chromosomes of an organism constitutes its genome. As it may be expected, there exists a correlation between the genome size

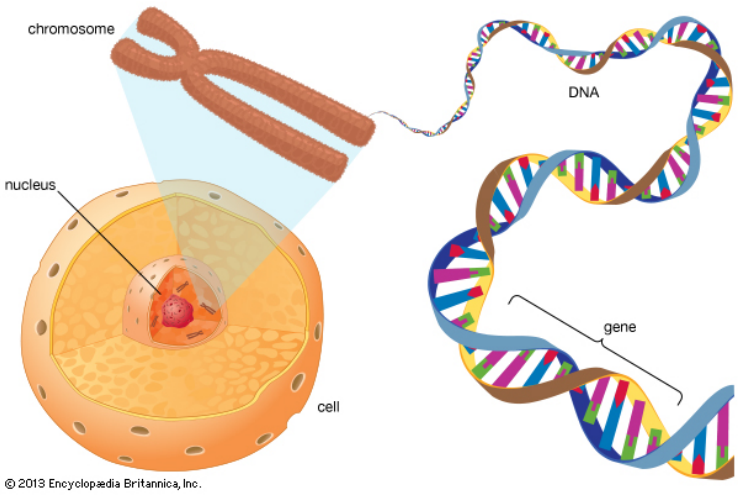


Figure 1.1: The spatial configuration and hierarchial structure inside the cell nucleous

and the complexity of the species.

The long DNA strands of every cells genome are packaged into chromatin in a very confined nuclear volume [1]. The organization of the chromatin in the nucleus is extremely important to biological function at the gene level as well as the global nuclear level.

Noteworthy is that the dimensions of the entire cell would not be sufficient to contain the DNA in a completely stretched form. The largest stretched human chromosome is nearly 3000 times larger than the average-sized cell diameter. Therefore, efficient compaction of DNA is an essential prerequisite for cellular function.

Chromosomes fold in a hierarchy of structures with increasing complexity, from nucleosomes and chromatin fibres to chromatin loops, chromosome domains, chromosome compartments and, finally, chromosome territories. While it is common to think about the genome as a linear object, in reality chromosomes are folded in a highly complex mode with regions located far apart on the same chromosome often coming in contact with one another. For example, many regulatory elements such as enhancers or insulators are physically separated to the genes they target and come in contact via folding. Enhancers are short DNA pieces that amplify transcription levels of certain genome areas, whereas insulators are DNA elements whose function is to prevent inappropriate interactions among adjacent chromosome regions.

Accumulating evidence demonstrates that the three-dimensional (3D) organization of chromatin within the eukaryotic nucleus reflects and influences genomic activities, including transcription, DNA replication, recombination and DNA repair. The study of the packaging of chromatin in the nucleus can shed light on the spatial aspects of gene regulation, i.e. the mechanisms that induce or suppress the expression of the genes.

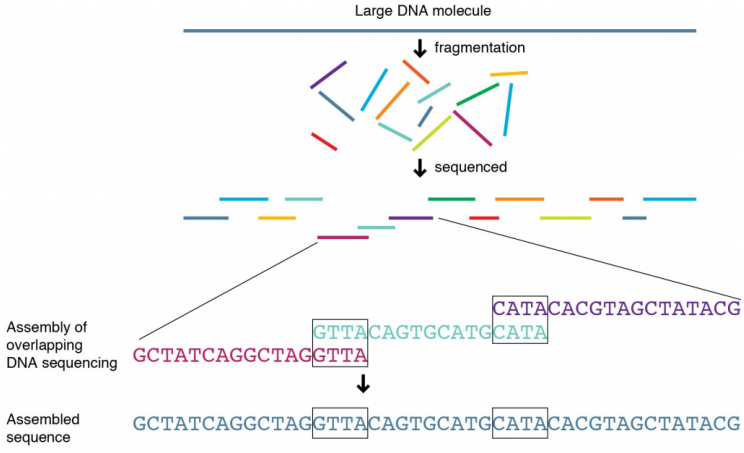


Figure 1.2: The genome sequencing high level procedure

Image source: [2]

### 1.1.2 Decoding the structure of the genome

Early methods for capturing the information encoded in the genome were based on DNA sequencing, a technology that is used to measure the order of the four bases (A,C,G,T) in a strand of DNA.

The whole genome can not be sequenced all at once because methods of DNA sequencing can only handle short regions of DNA at a time. The average length of the human genome is about millions of nucleotides. So instead, the genome is broken into small pieces, called reads, these pieces are sequenced and then reassembled in the proper order. Much of the work involved in sequencing lies in putting together these fragments.

The first sequencing methods, *Chemical* and *Sanger* sequencing, were invented in 1977. These methods were very slow and expensive, requiring enormous human power, but were able to generate reads of several hundreds of nucleotides. Later in the early 2000's, *Next Generation sequencing technologies* were developed, yielding bilions of reads in less than 24 hours.

These DNA-based sequencing methods only capture the linear structure of the genome. So a different set of techniques have been developed and applied to uncover the intrinsic mechanism of the genome architecture. To date, two types of tools have been used to dissect chromosome structure: microscopy-based imaging technologies and more recently developed molecular and biochemical tools. The chromosome conformation capture (3C) and 3C-derived methods, which belong to the latter group, provide a powerful tool for detecting spatial interactions between a single pair of genomic positions, within and between chromosomes.

In particular, Hi-C is the first of the 3C technologies to be truly genome-wide. It was the development of the Hi-C protocol by Liebermann-Aiden et al. 2009, which essentially pushed up the potential of 3C-based technology.

### 1.1.3 Hi-C: High Throughput 3C technology

Hi-C is a chromosome conformation capture technology that allows to inspect the nuclear organization by quantifying the proximity between different pairs of fixed positions in the genome. The Hi-C protocol involves the creation of DNA-protein bonds that cross-link physically interacting DNA loci. The DNA is then processed and filtered to generate a library of products that were spatially close to each other in the nucleus.

For example, consider regions $i$ and $j$ to be two nonadjacent regions on the same chromosome in the DNA fragment captured, sequence from region $i$ is at one end and sequence from region $j$ is at the other. Having captured these fragments, both ends are sequenced so that the nucleotides sequence from region $i$ and region $j$ are obtained. By mapping these sequenced reads back to the reference genome, and by summing up the number of reads with one end in region $i$ and the other end in region $j$, an score for the interaction between those two regions can be derived.

Finally, after sequencing, a 2D map of these interaction counts between genome regions, also called genomic loci, is generated. This Hi-C dataset is typically further processed and analyzed in order to discover new biological insights.

The genome-wide power and versatility of Hi-C makes it ideal for the study of the basic biology of genome organization and its implications for health and disease.

## 1.2 Finding structure in the 3D genome with Hi-C data analysis

In order to uncover structure-function relationships, it is necessary first to understand the principles underlying the folding and the 3D arrangement of chromosomes. In this chapter, the main considerations to take into account to process and analyze Hi-C data are explained and a quick review of the general patterns and structures that Hi-C data has already revealed in the human genome.

### 1.2.1 The Hi-C matrix

Data from Hi-C experiments are usually represented by these so-called chromosomal contact maps. A contact map is a matrix whose entries store the population-averaged co-location frequencies between pairs of loci. The genome is divided into equally sized bins or loci. The process that transforms the raw products of the Hi-C experiment into these contacts maps is summarized with these basic steps:

1. Map the Hi-C reads to the reference genome.
2. Filter out Hi-C experiment artifacts and create a contact matrix with valid Hi-C reads.
3. Filter matrix bins with low or zero read coverage.
4. Remove biases from the Hi-C contact matrices.

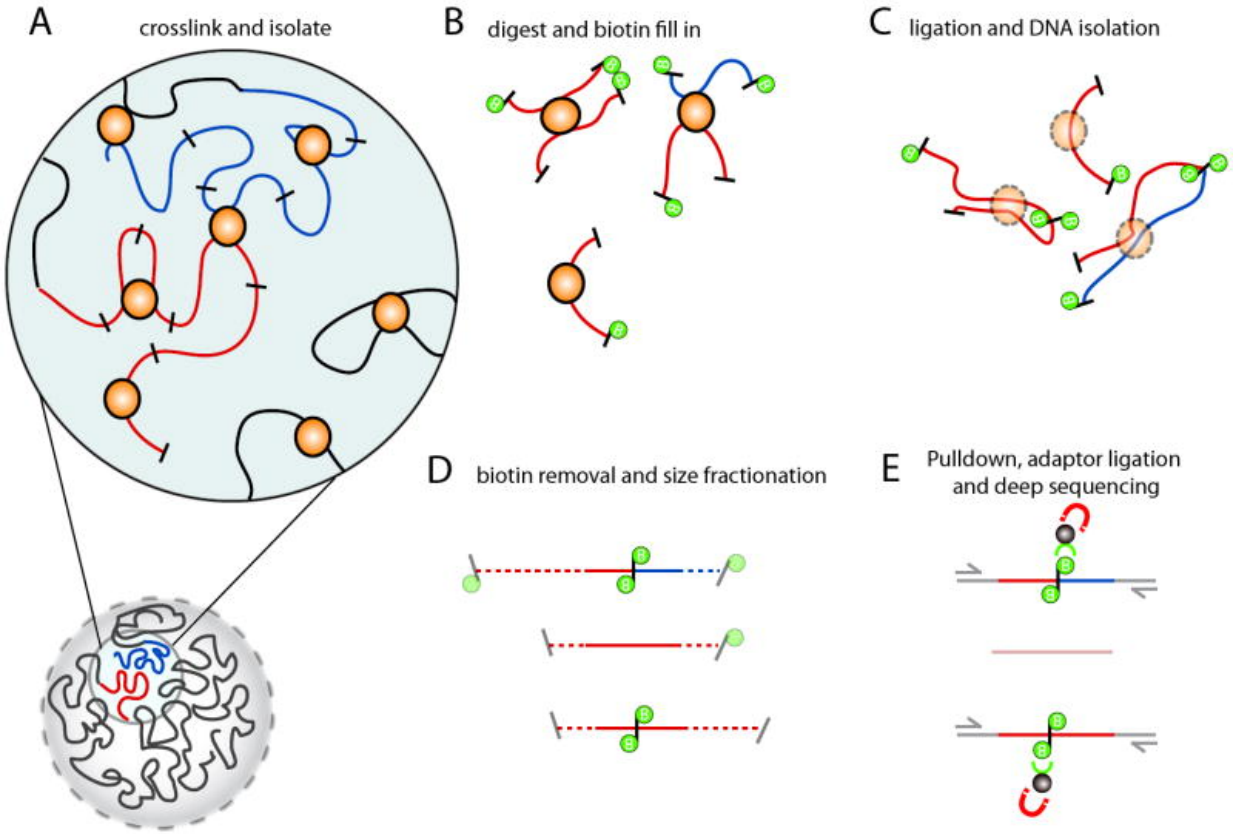


Figure 1.3: Overview of the HiC technology. A) Hi-C detects chromatin interaction both within and between chromosomes by covalently crosslinking protein/DNA complexes with formaldehyde. B) The chromatin is digested with a restriction enzyme and the ends are marked with a biotinylated nucleotide. C) The DNA in the crosslinked complexes are ligated to form chimeric DNA molecules. D) Biotin is removed from the ends of linear fragments and the molecules are fragmented to reduce their overall size. E) Molecules with internal biotin incorporation are pulled down with streptavidin coated magnetic beads and modified for deep sequencing. Quantitation of chromatin interactions is achieved through massively parallel deep sequencing.

Image source: [3]

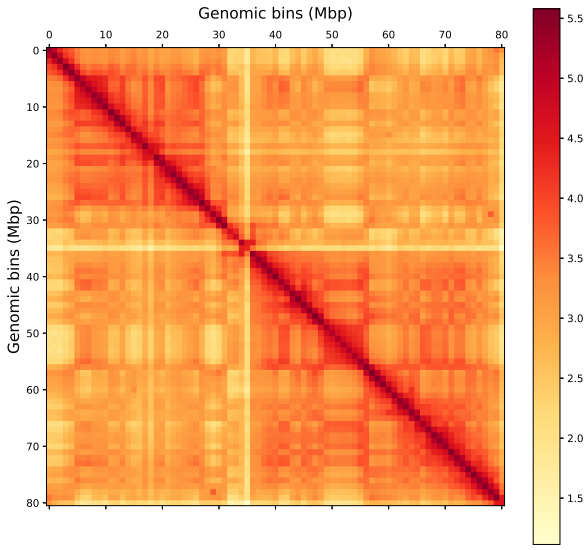


Figure 1.4: Log-scale Intra-chromosomal map of interactions from a particular cell population. The x and y axes represent loci in genomic order and each pixel represents the number of observed interactions between them.

Recent single-cell imaging experiment suggests that the frequency serves as a reasonable proxy of spatial distance [4]. Nonetheless it is important to remember that Hi-C does not measures spatial distance. Formaldehyde crosslinking will occur only between loci which physically interact. Thus, a weak Hi-C signal between two loci indicates that the interaction occurred in a small fraction of the population, but we cannot determine the distance between the two loci without making some simplifying assumptions about how interaction frequencies relate to physical distances. So these HiC maps can also be interpreted as spatial proximity map. It can be visually represented as a heatmap, with intensity indicating contact frequency.

Two kinds of contact maps can be generated, the intra-chromosomal maps and the inter-chromosomal maps, describing read counts within a chromosome region and between chromosomes regions respectively. The matrix of inter-chromosomal contacts is sparse as most of the a priori possible pairings have no associated reads, either because they are not in spatial proximity, or because their contact probability is too low to be reliably detected for a given sequencing depth. The sequencing depth is proportional to the number of reads used for the Hi-C experiment and describes the reach of the experiment.

Hi-C is an unbiased assay of chromatin conformation, resulting in even read coverage across the entire genome. But the fact that most Hi-C reads describe interactions at close linear distance along the chromosomes produces a relatively sparse read coverage for interactions between

individual loci separated by great spatial distance.

Typically, DNA fragments that are very close to each other in the linear genome will have the tendency to interact frequently with each other. This is seen in the intra-chromosomal heatmaps as a prominent diagonal.

Another important aspect of Hi-C data is its resolution. The space of all possible interactions is very large. For example, consider the human genome length to be approximately $10^9$ base-pairs (bp). Using a 6-bp cutting restriction enzyme, there are almost $10^6$ valid cut DNA fragments, leading to an interaction space of $10^{12}$ possible pairwise interactions [3]. Thus, achieving sufficient coverage of the whole genome to support very high resolution maps is a significant challenge.

In light of this, it is critical to establish the goals of the experiment, meaning whether one is most interested in either large-scale genomic conformations (e.g. genomic compartments) or specific small-scale interaction patterns (e.g. promoter-enhancer looping).

A Hi-C dataset is not sequenced deep enough to support maximal data resolution, as it is not yet cost-effective to obtain a sufficient number of reads. Instead, the data is binned into various fixed genomic interval sizes, to aggregate data and smooth out noise. Hi-C restriction fragments are assigned to bins by their midpoint coordinate. Binning the Hi-C data reduces the complexity and number of possible genome wide interactions.

### 1.2.2 Normalization of Hi-C data

Analyzing HiC maps is not an easy task. Hi-C data can contain many different biases, some of known origin and others from an unknown origin. They can be classified as follows:

- ***Read depth per region***: In Hi-C we expect to observe equal read coverage across the genome. However, factors such as the ability to map reads uniquely (e.g. density of genomics repeats)and the number of restriction sites (e.g. where cuts are allowed) in the experimental sample will influence the total number of reads per region.

- ***Linear distance between loci along the chromosome***: Loci closely spaced along a chromosome are almost guaranteed to be 'near' one another for no other reason than their linear DNA separation. As a result, closely spaced loci with have very high Hi-C read counts, regardless of their specific spatial conformation.

- ***Sequencing Bias***: GC content ( e.g. percentage of nitrogenous bases on a DNA), ligation preferences during library construction, normal sequencing problems.

There are two general approaches to Hi-C bias correction: explicit and implicit. Explicit bias models take into account factors such as mappability, GC content, proximity ligation and fragment length. One common explicit approach is the Observed over Expected (O/E) normalization method. This method outputs the ratio of observed to expected interactions by assuming each region has an equal chance of interacting with every other region in the genome by removing the

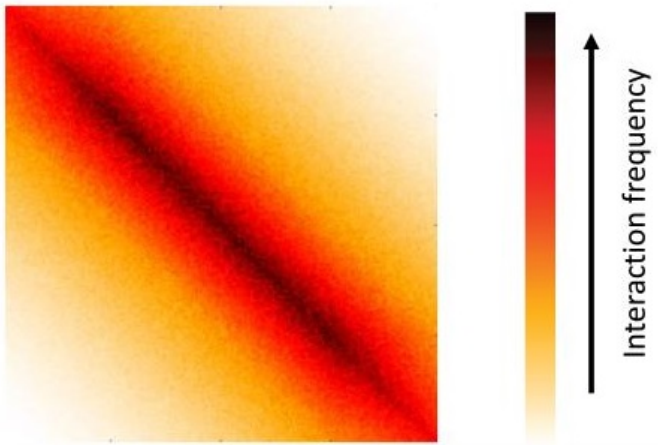


Figure 1.5: Distance decay pattern of the interaction frequency in Hi-C data.

Image source: [3]

linear distance bias, where regions are expected to interact depending on their linear distance along the chromosome.

The interaction frequency between loci within a chromosome decreases, on average, as their genomic distance increases. In the interaction matrix this pattern appears as a gradual decrease of interaction frequency the further one moves away from the diagonal. This effect is observed in figure 1.5.

The implicit methods are based on the fact that it can be quite difficult to know each and every bias, then one can use an implicit approach which we refer to as balancing or also refered as iteractive correction [5]. This procedure attempts to balance the matrix by equalizing the sum of every row/column in the matrix.

### 1.2.3 A/B compartments

Following the mapping, filtering and bias-correction of the Hi-C data, we are left with a binned, genome-wide interaction matrix, where each entry reflects an interaction frequency between two genomic loci. In this section, we briefly comment one of the first identified patterns in the chromosomal maps.

In the seminal work in [6], Hi-C was applied to generate the first comprehensive and unbiased long-range interaction maps of the human genome. Hi-C data revealed known hallmarks of nuclear organization (e.g. formation of chromosome territories, and preferred co-location of particular pairs of chromosomes) as well as novel folding principles of chromosomes. The phenomenon is known as chromosome territories, where chromosomes are physically separated and occupy a distinct volume in the nucleus. A particularly interesting result revealed that the human genome is divided in two spatial compartments, one containing active chromatin, and one containing mostly inactive segments of the genome. This is the so called A/B compartmentalization. Loci found

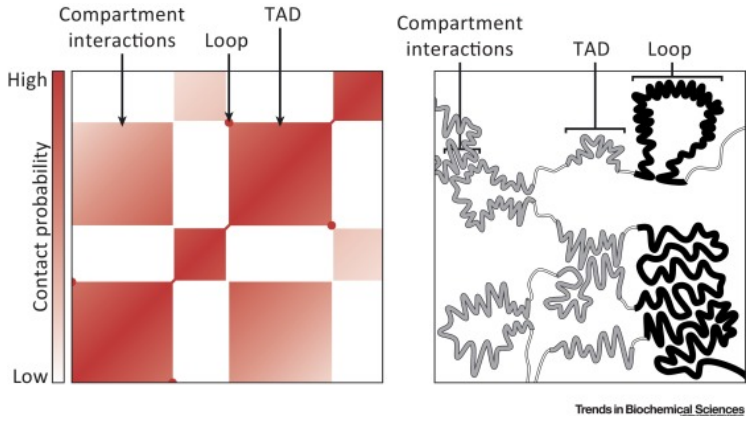


Figure 1.6: The landscape of structures detected in intra-chromosomal HiC data at a mega base scale

Image source: [3]

clustered in A compartments are generally gene rich, meaning that they are transcriptionally active, whereas loci found in B compartments are relatively gene poor, transcriptionally silent.

### 1.2.4 Topological associating domains

Chromosome conformation capture studies suggest that eukaryotic genomes are organized into structures called topologically associating domains (TADs) [7, 8]. TADS are defined as linear units of chromatin that fold as discrete three-dimensional (3D) structures tending to favor internal chromatin interactions. Small regions that are crucial for biochemical functions regulating the genome reside in these structures. For example, a majority of regulatory protein binding sites such enhancers and promoters localize within topological domains.

Detecting the topological domains is thus helpful for studying the relationship between chromosome organization and gene transcription. Topological domains, as regions that have high number of intra-contacts, are characterized by diagonal blocks in the Hi-C matrix. To identify topological domains, in [7] the authors employed a Hidden Markov Model (HMM) on the directionally index from a Hi-C matrix to determine regions initiated by significant downstream chromatin interactions and terminated by a sequence of significant upstream interactions.

The finest pattern identified in Hi-C data are the small scale loops formed by interactions of regulatory elements spanning hundreds kilobases. These are usually seen as peaks in the Hi-C heatmap. Figure 1.6 shows the different architectures dected in Hi-C data.

# Chapter 2

# Graphs, Graph learning and Spectral Clustering

This chapter reviews important concepts on algebraic graph theory and presents the notation used in the forthcoming chapters. The Hi-C data analysis pipelines proposed on the following chapters are based on a graph theoretic interpretation of Hi-C matrices. Modeling the spatial organization of chromosomes in a nucleus as a graph allows us to use spectral methods to quantitatively study their properties. The structure of this section is the following: Firstly, basic concepts of graph theory are reviewed, then an introduction to the graph learning framework, which later on will be used as a tool to model the Hi-C data. Finally, the core theory of the spectral clustering methods implemented is discussed.

## 2.1 Graph theory

A graph is defined as $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices indexed with $i = 1, \ldots, N$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is an unordered set of pairs of vertices from $\mathcal{V}$ called edges representing a connection between two vertices, where the total number of vertices is $M \leq N^2$. The edge between two vertices $i$ and $j$ is denoted $e_{ij}$ and refers to the information flowing from vertex $j$ to vertex $i$. If a one-way direction is assigned to the edges, the relations are asymmetric and the graph is called a directed graph, or a digraph. On the other hand, if both directions are assigned to the edges, then $e_{ij} \in \mathcal{E} \iff e_{ji} \in \mathcal{E}$ for all pairs $\{i, j\} \in \mathcal{V}$ and the graph is called an undirected graph. An undirected graph is connected when there is a path between every pair of vertices, that is every vertex is within reach from a specific vertex.

A graph is called weighted if a weight is associated with every edge according to a proper map $W : \mathcal{E} \to \mathbb{R}_+$ such that $W(e_{ij}) \neq 0$ if $e_{ij} \in \mathcal{E}$, $W(e_{ij}) = 0$ otherwise.

The edge structure of a graph $G$ with $N$ nodes is described by means of its adjacency matrix. The adjacency matrix $\mathbf{A}$ of $G$ is the matrix with entries $a_{ij}$ given by:

$$a_{ij} = \begin{cases} 1 & if\ e_{ij} \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

,i.e., the $\{ij\}^{th}$ entry of $\mathbf{A}$ is 1 only if vertex $j$ is a neighbor of vertex $i$. If $G$ is undirected, $a_{ij} = a_{ji}$, i.e, $\mathbf{A}$ is symmetric. In this work, only undirected weighted graphs are considered given the nature of the Hi-C data. The weighted adjacency matrix $\mathbf{W}$ of $G$ is a symmetric matrix whose entries $w_{ij} = W(e_{ij})$ for all $e_{ij} \in \mathcal{E}$.

The degree of a vertex $i$ are determined by the sums of the weights of edges, i.e.,

$$d_i = \sum_{i=1}^{N} w_{ij} \tag{2.2}$$

The degree matrix $\mathbf{D}$ of $G$ is the $N \times N$ diagonal matrix with its $\{ij\}^{th}$ entry given by

$$\mathbf{D}_{ij} = \begin{cases} d_i & if\ i = j \\ 0 & otherwise \end{cases} \tag{2.3}$$

The entries of the degree matrix are equal to the row sums of the weighted adjacency matrix, that is

$$\mathbf{D} = \text{diag}(\mathbf{W} \cdot \mathbf{1}) \tag{2.4}$$

where $\mathbf{1} \in \mathbb{R}^N$ is the vector of all ones and $\text{diag}(\mathbf{v})$ refers to the $N \times N$ diagonal matrix whose entries are the elements of a vector $\mathbf{v} \in \mathbb{R}^N$.

### 2.1.1 The Laplacian matrix

This section introduces the Laplacian matrix, which is used for mathematical convenience to describe the connectivity of a graph in a more compact form. In general, the spectral properties of the Laplacian are of interest for the spectral clustering methods. The Laplacian matrix of an undirected and weighted graph $G$ has its $\{ij\}^{th}$ entry given by

$$\mathbf{L}_{ij} = \begin{cases} d_i - w_{ii} & if\ i = j \\ -w_{ij} & i \neq j \end{cases} \tag{2.5}$$

This definition can be expressed in matrix form as follows

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{2.6}$$

and the normalized Laplacian [9] is given by,

$$\mathbf{L_N} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} \tag{2.7}$$

By construction, the Laplacian matrix of an undirected and weighted graph is always symmetric.

### 2.1.2 Spectral properties of the Laplacian

In undirected weighted graphs, the associated Laplacian is positive semidefinite and its eigenvalues can be arranged in an increasing order as follows

$$0 = \lambda_1 \leq \cdots \leq \lambda_N \leq 2d_{max} \tag{2.8}$$

where $\lambda_i$ denotes the $i^{th}$ eigenvalue of $\mathbf{L}$ and $d_{max}$ denotes the maximum degree of $G$. For the normalized Laplacian, the spectrum is normalized as follows

$$0 = \lambda_1 \leq \cdots \leq \lambda_N \leq 2 \tag{2.9}$$

The eigenvalues are real, non-negative and repeated according to their multiplicity $k_i$. In addition, the eigenvector associated with $\lambda_1(\mathbf{L})$ is $\mathbf{1}$ such that $\mathbf{1^T L} = \mathbf{0^T}$, where $\mathbf{1} = [1, \ldots, 1]^T$. The second smallest eigenvalue $\lambda_2$ is known as the Fiedler number, or algebraic connectivity, which characterizes the connectivity and stability of the graph.

Let $G$ be an undirected graph with non-negative weights. Then, the multiplicity of the eigenvalue $\lambda_1(\mathbf{L})$ equals the number of connected components $A_1, \ldots, A_{k_1}$ in the graph. The eigenspace of the eigenvalue $\lambda_1(\mathbf{L})$ is spanned by the indicator vectors $\mathbf{1}_{A_1}, \ldots, \mathbf{1}_{A_k}$ of those components. These indicator vectors are binary vectors that contain ones for points within the components or cluster and zero otherwise. The latter is particularly important for the topic of spectral clustering.

## 2.2 Graph learning

In this section, the framework for graph learning at large scale proposed in [10] is presented. There are many ways to construct a graph from data. Typically they are constructed either by connecting nearest vertices or samples according to some metric, or by learning them from data, solving an optimization problem. While graph learning does achieve a better quality than traditional methods, it also comes with a higher computational cost. In particular, the previous state-of-the-art model cost is $\mathcal{O}(N^2)$ per iteration for $N$ nodes [11]. Furthermore, it needs parameter tweaking to control the graph sparsity, which makes it prohibitive for applications with more than a few thousands of nodes.

The setup for the large scale graph learning framework is the following: given $\mathbf{X} \in \mathbb{R}^{N \times M}$ whose columns reside on the nodes of an unknown weighted undirected graph, the objective is to learn the weight edges $\mathbf{w} \in \mathbb{R}_+^{\frac{M(M-1)}{2}}$ under the smoothness assumption. This assumption states that values change smoothly across adjacent nodes. The smoothness of a set of vectors

$\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^M$ is usually quantified by the Dirichlet energy [12]. Let $\mathbf{W}$ be the adjacency matrix of an unknown weighted undirected graph. The Dirichlet energy is expressed by

$$\mathrm{Tr}(\mathbf{X}^T\mathbf{L}\mathbf{X}) = \frac{1}{2}\sum_{i,j}^{N} w_{ij} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2 \tag{2.10}$$

where $w_{ij} \in \mathbb{R}_+$ denotes the edge weight between node $i$ and $j$ and $\mathrm{Tr}(\cdot)$ is the trace operator. We can learn a graph under the assumption that $\mathbf{X}$ is smooth on it, by minimizing (2.10) w.r.t. $\mathbf{L}$, when $\mathbf{X}$ is given. Recently, [11] proposed an unified model for learning graphs from smooth signals, which solves:

$$\underset{\mathbf{W}\in\mathcal{W}}{\text{minimize}} \left\|\mathbf{W}\circ\mathbf{Z}\right\|_{1,1} + f(\mathbf{W}) \tag{2.11}$$

where $z_{ij} = \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2$ , $\circ$ denotes the Hadamard product, and the first term is equal to $\mathrm{Tr}(\mathbf{X}^T\mathbf{L}\mathbf{X})$. The optimization is over the set $\mathcal{W}$ of valid adjacency matrices, that is, non-negative, symmetric, with zero diagonal. The smoothness term is a weighted $l_1$-norm of $\mathbf{W}$ that penalizes edges connecting dissimilar rows of $\mathbf{X}$. The role of $f(\mathbf{W})$ is to stop $\mathbf{W}$ from obtaining a trivial zero value, regulate sparsity, and impose more structure depending on the application. Kalofolias obtained state-of-the-art results using

$$f(\mathbf{W}) = -\alpha\mathbf{1}^T\log(\mathbf{W}\mathbf{1}) + \frac{\beta}{2}\left\|\mathbf{W}\right\|_F^2 \tag{2.12}$$

Substituting (2.12) in (2.11) forms the so-called log model. It is desirable to have control of how sparse the resulting graph is. To meet these expectations, the parameters $\alpha > 0$ and $\beta \geq 0$ control the magnitude of the weights in $f(\mathbf{W})$. The terms in (2.12) have different roles. The logarithmic barrier acts on the node degree vector $\mathbf{W}\mathbf{1}$, which enforces a positive degree, but does not prevent the weights from becoming zero. This improves the overall connectivity of the graph, without compromising sparsity. Note however, that adding solely a logarithmic term ($\beta = 0$) leads to very sparse graphs, and changing $\alpha$ only changes the scale of the solution and not the sparsity pattern. For this reason, the frobenius norm of $\mathbf{W}$ is added to penalize the formation of big weights. Note that that solely minimizing the smoothness term leads to naturally sparse graphs, so adding an extra $l_1$-norm term has no effect in the sparsity of the solution.

### 2.2.1 Constrained edge pattern

In traditional graph learning [11], all $\binom{N}{2}$ possible edges between $N$ nodes are considered, which results in a cost of $\mathcal{O}(N^2)$ computations per iteration. Often, however, we need graphs with a roughly fixed number of edges per node, like in $k$-NN graphs. It is natural to see whether the cost of graph learning can be reduced, while still reflecting the final desired graph sparsity.

In fact, (2.11) can be solved efficiently when a constrained set $\mathcal{E}_{allowed} \subseteq \mathcal{E}$ of allowed edges is known a priori. In that case, it is enough to solve the modified problem

$$\underset{W \in \widetilde{\mathcal{W}}}{\text{minimize}} \, \|\mathbf{W} \circ \mathbf{Z}\|_{1,1} + f(\mathbf{W}) \tag{2.13}$$

where we optimize in the restricted set of adjacency matrices $W \in \widetilde{W}$. In this form, the problem can be solved by the primal dual techniques by [13]. The cost of this technique is $\mathcal{O}(|\mathcal{E}_{allowed}|)$ instead of $\mathcal{O}(N^2)$ of the initial algorithm by [11], thus reducing the overall complexity.

When approximating the support of the final edges of a graph, it is preferable to begin with an initial support with a larger cardinality than the desired final graph support, and let the weight learning procedure automatically select which edges to set to zero.

### 2.2.2 Automatic parameter selection

A major obstacle in the log model is the choice of convenient parameters $\alpha$, $\beta$, as a grid search increases computation remarkably. Still, the problem can be avoided. The sparsity can depend effectively on a single parameter, and Kalofolias proposes a method to set it automatically for an average number of neighbours per node. The reduction of the grid search to a single parameter tuning the sparsity of the graph is based in a two-step process. Firstly, it is shown in [10] that all graphs that can be learned by model (2.11) can be equivalently computed by multiplying the distances $\mathbf{Z}$ by $\theta = \frac{1}{\sqrt{\alpha\beta}}$ , using them to learn a graph with fixed parameters $\alpha = \beta = 1$, and multiplying all resulting weights by a factor $\delta = \sqrt{\frac{\alpha}{\beta}}$. The factor $\delta$ only changes the scaling of the resulting solution, so in practice, the focus is on tuning parameter $\theta$. This claim is formulated in Theorem 2.2.1.

**Theorem 2.2.1.** *Let $W^*(Z, \alpha, \beta)$ denote the solution of the log model for input distances and parameters $\alpha, \beta > 0$. Then the same solution can be obtained with fixed parameters $\alpha = 1$ and $\beta = 1$, by multiplying the input distances by $\theta = \frac{1}{\sqrt{\alpha\beta}}$ and the resulting edges by $\delta = \sqrt{\frac{\alpha}{\beta}}$.*

$$W^*(Z, \alpha, \beta) = \sqrt{\frac{\alpha}{\beta}} W^*(\frac{1}{\sqrt{\alpha\beta}} Z, 1, 1) = \delta W^*(\theta Z, 1, 1) \tag{2.14}$$

The last step for automating parameter selection is to find a relationship between $\theta$ and the desired graph sparsity $k_s$ (i.e. the average number of neighbors per node).

## 2.3 Spectral clustering

Clustering algorithms provide a useful instrument to explore data structures. The aim of clustering methods is to collect patterns on the basis of a similarity (or dissimilarity) criteria where clusters are sets of similar patterns.

Clustering techniques can be roughly divided into two categories: hierarchical and partitioning. Hierarchical clustering techniques [14] are able to find structures which can be further divided in substructures and so on recursively. The result is a hierarchical structure of groups known as dendrogram.

Partitioning clustering methods try to obtain a single partition of data and are often based on the optimization of an adequate objective function. One example is the well-known K-means clustering.

A particular efficient and popular class of partitioning clustering methods is spectral clustering, which arises from concepts in spectral graph theory. Spectral clustering became popular with, among others, [15, 16]. The basic idea is to construct a weighted graph from the initial data set where each vertex represents a data point and each weighted edge measures the similarity between two data points. Once a graph is built, spectral decomposition methods are employed on the adjacency matrix of the graph to construct spectral embeddings where other classes of partitioning algorithms such as k-means can be used. The clustering problem can be seen as a graph cut problem, which can be addressed by means of the spectral graph theory. In fact, there is a very fine linkage between the graph cut problem and spectral graph theory.

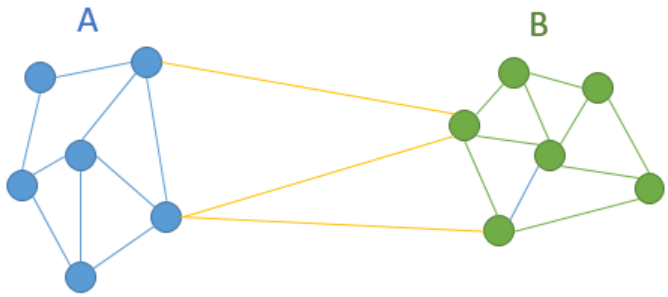


Figure 2.1: A toy graph with a partition into two disjoints subsets A and B.

Consider a partition of the weighted graph, $G$, depicted in Figure 2.1 into two disjoint vertices subsets, $A$ and $B$. The Min-Cut problem tries to partition graph into two sets $A$ and $B$ such that weight of the edges connecting vertices in $A$ to vertices in $B$ is minimum. One intuitive goal is to find the partition that minimizes (2.15), the so-called cut cost function.

$$\text{cut}(A, B) = \sum_{i \in A, j \in B} w_{ij} \tag{2.15}$$

This bi-partitioning problem often yields non satisfactory partitions, as it isolates vertices from the rest of the graph for some graph instances. Consider the following function that measures the size of a subset $A \subseteq \mathcal{V}$:

$$\text{vol}(A) = \sum_{i \in A} d_i \tag{2.16}$$

where $d_i$ denotes the node degree of an undirected graph. Intuitively, $\text{vol}(A)$ measures the size of A by summing over the degrees of vertices in $A$. Then the normalized min-cut problem, also called NCut, which takes into account the size of the clusters, is stated as follows

$$\text{Ncut}(A, B) = \text{cut}(A, B)(\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)}) \tag{2.17}$$

The Ncut problem belongs to a complexity class of problems called NP-hard, for which no polynomial-time algorithm capable of solving it exists. It has been demonstrated [9] that spectral clustering with the normalized Laplacian is an approximation to the graph partitioning problem. The semi-optimal solution for this relaxed version of the problem is the second eigenvector of the graph's Laplacian, the Fielder vector. Cuts based on the second eigenvector give a guaranteed approximation to the optimal cut [9]. This method can be extended to finding k clusters by using recursion or computing more eigenvectors.

The core theory of spectral clustering is to find a low-dimensional spectral embedding of the weighted graph, achieved by an eigenvalue decomposition of the Laplacian matrix of the weighted graph where clusters properties are enhanced. Next, the general guidelines to perform spectral clustering inspired by [17] are reviewed.

Given a set of data points $\mathbf{x}_1, \ldots, \mathbf{x}_N$ and some notion of similarity $w_{ij} \geq 0$ between all pairs of data points $\mathbf{x}_i$ and $\mathbf{x}_j$, a way of representing the data is in form of the similarity graph $G = (\mathcal{V}, \mathcal{E})$. Each vertex or node $v_i$ in this graph represents a data point $\mathbf{x_i}$. Two vertices are connected if the similarity $a_{ij}$ between the corresponding data points $\mathbf{x_i}$ and $\mathbf{x_j}$ is positive or larger than a certain threshold, and the edge is weighted by $w_{ij}$. The problem of clustering can now be reformulated using the similarity graph: we want to find a partition of the graph such that the edges between different groups have very low weights (which means that points in different clusters are dissimilar from each other) and the edges within a group have high weights (which means that points within the same cluster are similar to each other).

The main tools for spectral clustering are graph Laplacian matrices. In particular, this algorithm makes use of the normalized Laplacian for its well-known properties. To highlight one of the weakness of unnormalized Laplacians, its spectrum is influenced by the nodes having the highest vertex degree. This can lead to the high degree nodes masking the nodes with lower vertex degrees, and consequently leads to loss of sensitivity to complex structure. The spectral clustering steps are summarized in Algorithm 1.

**Algorithm 1** Spectral Clustering procedure

1: **procedure** SPECTRAL CLUSTERING
2: ***Input***: Weighted adjacency matrix $\mathbf{W}$, number of clusters $k$
3: Build normalized laplacian $\mathbf{L}_N$
4: Compute the first $k$ eigenvectors $\{\mathbf{v_1}, \ldots, \mathbf{v_k}\}$ associated to the k smallest eigenvalues of $\mathbf{L}_N$
5: Build the matrix $\mathbf{V} \in \mathbb{R}^{N \times k}$ with the eigenvectors as columns.
6: For $i = \{1, \ldots, N\}$, let $\mathbf{y}_i \in \mathbb{R}_k$ be the vector corresponding to the i-th row of $\mathbf{V}$
7: Cluster the points $\{\mathbf{y}_i\}_{i=1,\ldots,N}$ with $k$-means into clusters $C_1, \ldots C_k$
8: ***Output***: Clusters $A_1, \ldots, A_k$ with $A_i = \{j | \mathbf{y}_j \in C_i\}$

### 2.3.1 Spectral clustering analysis

Consider an ideal case where some graph $G$ has $k$ connected components . Without loss of generality, assume that the nodes of the graph are ordered according to the connected components they pertain to. In this case, the weighted adjacency matrix $\mathbf{W}$ has a block diagonal form, and the same happens for $\mathbf{L}$

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_1 & & & \\ & \mathbf{L}_2 & & \\ & & \ddots & \\ & & & \mathbf{L}_k \end{pmatrix} \tag{2.18}$$

Notice that each of the blocks $\mathbf{L}_i$ is well-defined Laplacian on its own, specifically the Laplacian corresponding to the subgraph of the i-th connected component. The spectrum of $\mathbf{L}$ is given by the union of the spectra of $\mathbf{L}_i$, and the corresponding eigenvectors of $\mathbf{L}$ are the eigenvectors of $\mathbf{L}_i$, filled with 0 at the locations of the other blocks. As each $\mathbf{L}_i$ is a Laplacian of a connected graph, it is known that every $\mathbf{L}_i$ has eigenvalue 0 with multiplicity 1, and the corresponding eigenvector is the constant one vector $\mathbf{1}$ on the i-th connected component. Consequently, the matrix $\mathbf{L}$ has as many eigenvalues with value 0 as there are connected components, and the correspondent eigenvectors are the vectors indicating the positions of the connected components on the graph. This assumption hols true for the ideal case, but in real-world scenarios, graphs not always contain $k$ connected components. However, spectral clustering still can work for the case of a graph containing weakly connections between components.

# Chapter 3

# Clustering pipeline of Hi-C contact maps

In this chapter, the strategy to explore the 3D spatial structure of the nucleus of a Jurkat cell infected with the human immunodeficiency virus (HIV) is discussed. Given that the role of nuclear architecture in viral infection is known to be important, the main goal is to test the hypothesis that HIV viruses have a non-random insertion pattern in the host's genome [18] by interpreting the relationship between HIV integration spots and genome organization. The viral DNA accesses the chromatin of the cell nucleus and integrates itself in the host chromosomal DNA. The Hi-C data of the Jurkat genome can provide a 3D view of the nuclear organization of the cell. To achieve this task, the Jurkat genome is segmented into spatial clusters. For each chromosome, $K_1$ clusters of loci enriched in self interactions were generated and then coalesced down to $K_2$ genome-wide clusters based on their inter-chromosomal contacts. These genome-wide clusters are evaluated to indicate whether 3D genome organization of Jurkat cells is an important factor in of the HIV insertion process.

A metric named *A/B score* is used to biologically interpret the resulting clusters and test the stated hypothesis. Hi-C data have shown that transcribed genes make preferential contacts with other transcribed genes, forming a spatial cluster known as the A compartment. Reciprocally, silent genes form a spatial cluster known as the B compartment. The loci of the B compartment are usually in contact with the nuclear outer shell, i.e. the periphery of the nucleus. However, the transcribed genes in contact with the nuclear pores are also peripheral, making the nuclear periphery a composite environment, with features of either silent or active chromatin. The *A/B score* will help characterizing the presence of these two spatial clusters within the resulting genome-wide communities.

Two clustering methods are implemented to asses the consistency of the detected structure: a Pearson correlation and a graph learning based spectral clustering. Moreover, the clustering methods are quantitatively evaluated to objectively measure of the true genome-wide structure of the nucleus resulting from this particular Hi-C experiment.

## 3.1 Hi-C Jurkat cell data and HIV insertion hotspots

The available data is a Hi-C assay performed with the protocol published by [19] with modifications. Briefly, one million cells are used for the experiment. Hi-C on uninfected Jurkat cells yielded 1.5 billion informative contacts [20]. The resolution of the Hi-C dataset, determined by the sequencing depth and others factors, is about 5 kilo-base pairs (5kbp) per genome bin. To our knowledge, this dataset constitutes the highest resolution Hi-C experiment presently available in Jurkat cells. The Jurkat cell has two copies of each autosome but only one copy of each sex chromosome, adding up to 23 pairs of chromosomes. In addition, chromosomes X and Y are targeted less frequently than autosomes. In this work, we will consider only the set of autosomes, therefore the number of chromosomes analyzed is $N_{chr} = 22$. Two kinds of Hi-C maps are available: intra-chromosomal and inter-chromosomal. The length of each chromosome as the number of genomic bins at 5kbp per bin is displayed in Figure 3.1, which in fact determines the size of the raw Hi-C matrices.

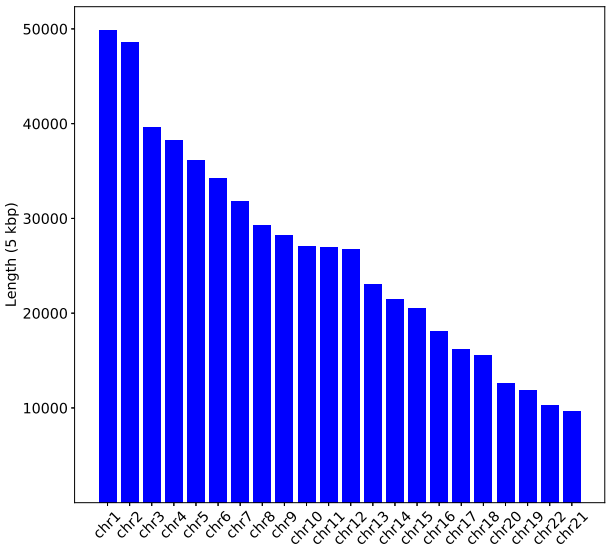


Figure 3.1: Length of each chromosome in genomic bins units

The Hi-C maps of the larger chromosomes are usually sparse matrices. In a parallel experiment [18], a list of integration spots of the HIV virus on a infected Jurkat genome were sequenced. The authors developed a method called Barcoded HIV ensembles (B-HIVE) to map the genomic positions of thousands of individual viruses in an infected cell population. Another list of chromosomal locations or genomic positions of HIV integration spots is obtained from [21], thus providing more samples for the raw data describing HIV insertion spots. These two independent lists, generated on very similar Jurkat cell models, will be analyzed to evaluate the insertion patterns in the genome-wide communities.

## 3.2 Pre-processing Hi-C data

Let $\mathbf{H}_c \in \mathbb{R}^{L_c \times L_c}$ be the symmetric and non-negative matrix of interaction between the genome regions of a given chromosome $c$, where $c = 1, \ldots, N_{chr}$ and $L_c$ is the length of chromosome $c$ in genomic bins. Therefore, the element $h_{ij}^c$ denotes the number of contacts detected between loci $i$ and loci $j$ of the genome. At higher resolution, these matrices are typically very sparse and noisy, so signals or patterns revealing some kind of structure are often hidden.

The intra-chromosomal raw Hi-C matrix of chromosome 14 is depicted as an intensity heatmap in Figure 3.2. The Topologically Associating Domains (TADs) and loop domains are clearly visible on the raw Hi-C map in 5 kbp bins, showing that the Hi-C experiment captures the basic structural features of the Jurkat genome.

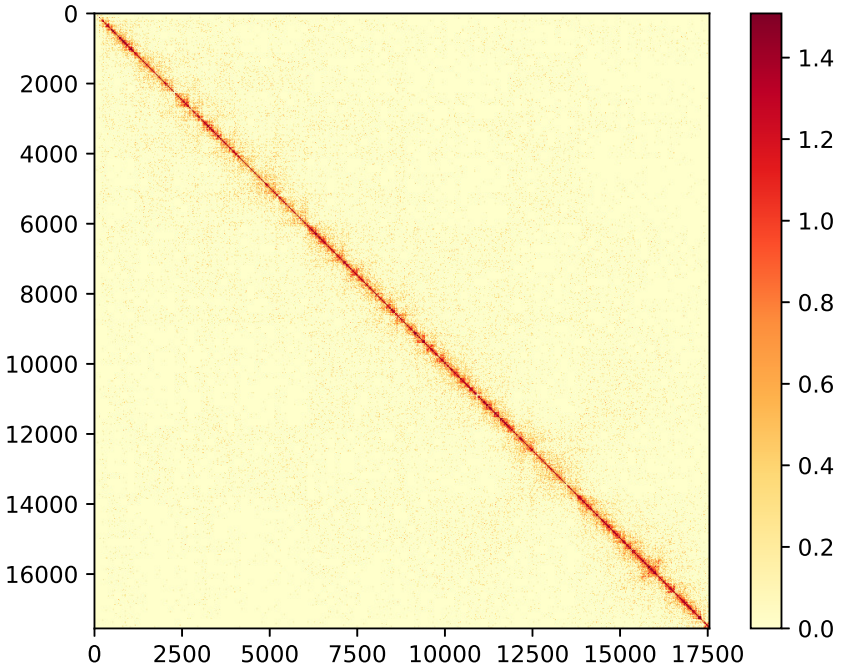


Figure 3.2: Raw Hi-C map of chromosome 14

In order to eliminate possible biases, the HiC matrices are processed as follows: some blank areas that correspond to a part of the chromosome called centromere are removed. This region is unmappable due to the existence of duplicated reads, so no Hi-C information is available.

After removing the centromere region, the Hi-C matrix is normalized by the Observed over Expected (OE) procedure to remove the distance effect. Recall that closely spaced loci are likely to have large Hi-C read counts regardless of their specific conformation. The normalization divides the $(i, j)$-th entry of a Hi-C matrix by the mean of Hi-C reads of all matrix entries at the same distance $d = |i - j|$. Let $\mathbf{U_c} \in \mathbb{R}^{L_c \times L_c}$ be the normalized Hi-C matrix with entries

$$u_{ij}^c = \frac{h_{ij}^c}{f_d} \tag{3.1}$$

where $f_d$ is the so-called expected contact frequency coefficient at distance $d = |i - j|$. These coefficients are computed using all the intra-chromosomal matrices. Mathematically, this step is described as:

$$f_d = \frac{\sum_{c=1}^{N_{chr}} \sum_{(i,j)\in\mathcal{I}_d} h_{ij}^c}{\sum_{c=1}^{N_{chr}} L_{c,d}} \tag{3.2}$$

The term $L_{c,d}$ defines the possible number of pairs of genomic positions separated by $d$ on a given chromosome, and $\mathcal{I}_d = \{(i,j) \mid i - j = d, 0 < i \leq L_c, 0 < j \leq L_c\}$ is the set of matrix elements at a distance $d = |i - j|$. Moreover, it is also common to apply a simple outlier removal method to the normalized Hi-C matrix $\mathbf{U_c}$. The OE matrix is saturated to the 90-th percentile of the matrix data. The dynamic range of the values of the matrix is effectively reduced. The resulting heatmap after pre-processing a raw Hi-C matrix is shown in Figure 3.3.

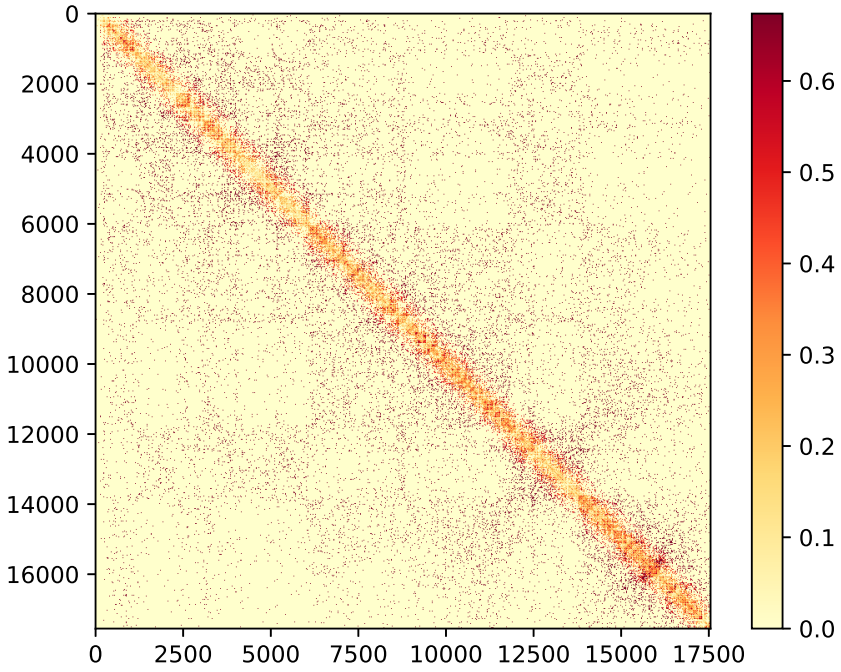


Figure 3.3: Log-scale processed Hi-C map after applying O/E normalization and percentile saturation

## 3.3 A/B clustering at Mega base pairs scale

In this section, an experiment to test that our data is consistent with the A/B compartmentalization of the chromosome territory structure at the mega base pair (Mbp) scale is carried out. At this lower resolution, processing time of the Hi-C matrices is not as significant and the A/B

compartmentalization is highlighted.

Recall that within each chromosome territory, compartments originally termed A and B, each of several mega pair-bases, tend to associate within each single chromosome, reflecting the preferential association of gene-rich regions and their segregation from gene-poor regions. The objective of this section is to explore the method that revealed this insight.

The method described below is based on [6]. The spatial clustering is performed on the Hi-C interaction count matrix of a particular chromosome $c$. The resolution of Hi-C bins is 1 Mbp, meaning that each genomic coordinate or loci spans 1 mega base-pairs of the linear genome. The clustering method follows these general steps:

1. Pre-process the Hi-C matrix.
2. Perform PCA with the Pearson correlation matrix on the normalized Hi-C matrix.
3. Assign A/B compartments to the peaks and valleys of the first principal component with some threshold criteria.

The raw Hi-C matrix of chromosome 16 is firstly pre-processed with the O/E procedure and the matrix entries are saturated at the 90-th percentile of the data. The resulting heatmap is depicted in Figure 3.4.

The Pearson correlation coefficient is a measure for linear relationships between two normal distributed variables. Usually, a Pearson coefficient with a value of 1 represents a perfect positive relationship, -1 a perfect negative relationship, and 0 indicates the absence of a relationship between variables. The coefficients of the Pearson correlation matrix of $\mathbf{U}_c$ are computed as follows:

$$\mathbf{p}^c_{i,j} = \frac{\sum_{k=1}^{L_c} \left(u^c_{i,k} - \bar{u}^c_i\right)\left(u^c_{k,j} - \bar{u}^c_j\right)}{\sqrt{\sum_{k=1}^{L_c} \left(u^c_{i,k} - \bar{u}^c_i\right)^2}\sqrt{\sum_{k=1}^{L_c} \left(u^c_{k,j} - \bar{u}^c_j\right)^2}} \tag{3.3}$$

where $\mathbf{u}^c_j = [u^c_{1,j}, \ldots, u^c_{L_c,j}]^T$ denotes the j-th column of $\mathbf{U_c}$, reflecting how a single loci interacts with the rest of chromosome coordinates and $\bar{u}^c_j = \frac{1}{L_c}\sum_{k=1}^{L_c} u^c_{k,j}$ is the mean value of the j-th column vector. The diag($\mathbf{P}_c$) is set to zero to avoid self-interacting loci, which do not contribute relevant information about spatial organization.

The Pearson correlation matrix $\mathbf{P}_c$ displays a sharpened chessboard pattern in Figure 3.5, highlighting the interactions among far distant regions within the chromosome. The interpretation of the Pearson coefficients is that regions or loci that correlate positively and negatively interact with the rest of loci in a similar manner, indicating that they might lie close in the 3D space, and values close to zero denote that they are not similar at all and might not be anywhere near each other.

The plaid pattern suggests that each chromosome can be decomposed into two sets of loci (arbitrarily labeled A and B) such that contacts within each set are enriched and contacts between sets are reduced.

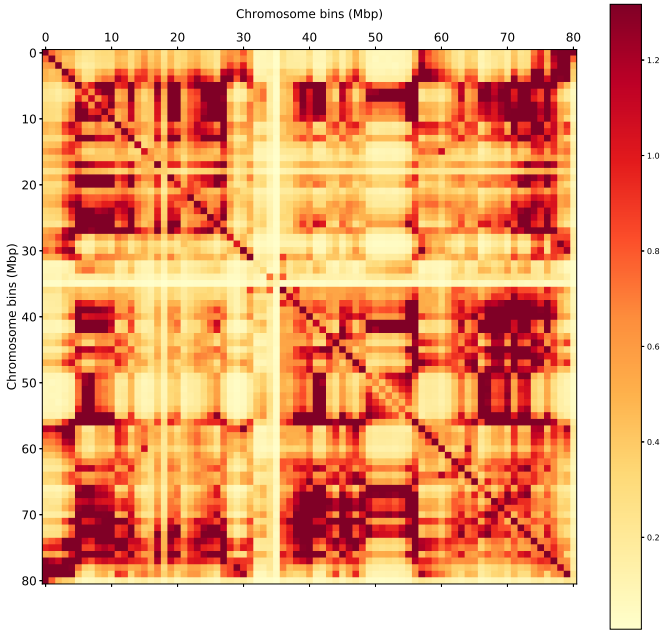


Figure 3.4: Pre-processed Hi-C matrix of chromosome 16 at 1 Mbps resolution

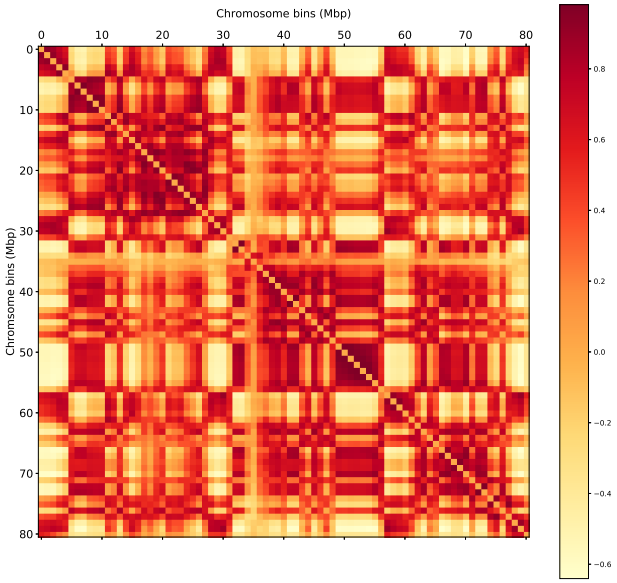


Figure 3.5: Pearson correlation matrix of the pre-processed Hi-C map of chromosome 14

For partitioning the chromosome, a principal component analysis (PCA) approach is taken, by performing the eigendecomposition of the correlation matrix. The typical goal of a PCA is to reduce the dimensionality of the original feature space by projecting it onto a smaller subspace, where the eigenvectors will form the axes. In this case, the first eigenvector corresponding to the largest eigenvalue is kept as the first principal component (PC).

The first principal component (PC) corresponds to the plaid pattern in Figure 3.6 (positive values defining one set, negative values the other).

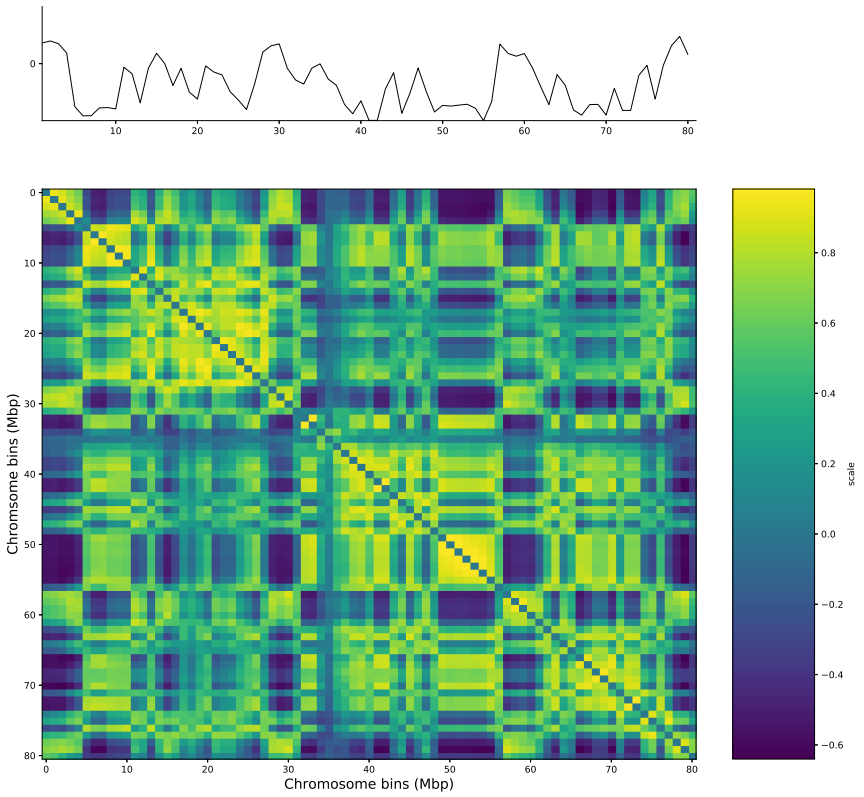


Figure 3.6: The first PC and the Pearson correlation matrix aligned

To explore whether the two spatial compartments correspond to known features of the genome, we would need to compare the compartments identified in our 1 Mb correlation maps to known genetic and epigenetic features of this particular cell type. But, due to lack of availability of genetic/epigenetic data-sets and that the scope of this work is limited to the technicality of the clustering methods, no further experiments are carried on. This spatial configuration has already been confirmed in human cells [6, 22].

## 3.4 Clustering methods

Modeling the spatial organization of chromosomes in a nucleus as a graph allows us to make use spectral methods to quantitatively study their properties. A Hi-C matrix therefore associates a graph to the genome, where vertices are defined by binned loci in the genome, and the edge weight

between a pair of loci is proportional to their contact frequency. In this section, we introduce the two main algorithms that are used in the clustering process of the genome interaction networks. Both methods belong to the family of spectral clustering and are well suited for finding non-compact/non-convex clusters. While the type of weighted adjacency matrix used by the algorithms is different. Recall that the input to a spectral clustering algorithm is a similarity or adjacency matrix and that the high-level main steps of a spectral clustering algorithm are

- Compute a spectral embedding of the similarity/adjacency matrix
- Employ a technique such as k-means to cluster the nodes in the low-dimensional spectral embedding.

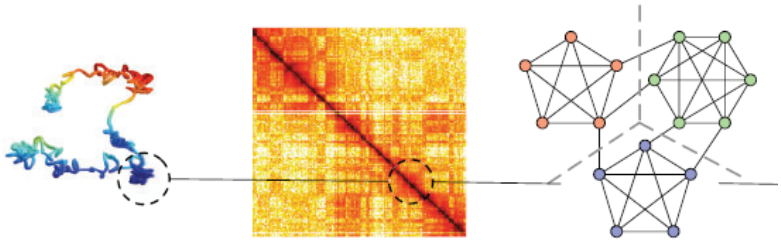

Figure 3.7: Graph theoretic approach for modelling Hi-C contacts

Firstly, a spectral clustering method based on the Pearson correlation proposed in [20] is reviewed. It will serve as a baseline method. This clustering method is very similar to the PCA clustering method described in Section 3.3. The objective is to generate $K$ partitions of the intra-chromosomal interaction network. The general methodology is detailed in Algorithm 2.

**Algorithm 2** Correlation-based Spectral clustering

**procedure** CORRELATION-BASED SPECTRAL CLUSTERING
  ***Input***: normalized Hi-C matrix $\mathbf{U}_c \in \mathbb{R}^{L_c \times L_c}$, number of clusters $K$
  Compute the Pearson correlation matrix, $\mathbf{C}$
  Compute the first $K$ eigenvectors $\{\mathbf{v_1}, \ldots, \mathbf{v_K}\}$ associated to the K largest eigenvalues of $\mathbf{C}$
  Build the matrix $\mathbf{V} \in \mathbb{R}^{N \times K}$ with the eigenvectors as columns.
  For $i = \{1, \ldots, N\}$, let $\mathbf{y}_i \in \mathbb{R}^K$ be the vector corresponding to the $i$-th row of $\mathbf{V}$
  Cluster the points $\{\mathbf{y}_i\}_{i=1,\ldots,N}$ with $k$-means algorithm into clusters $C_1, \ldots C_K$
  ***Output***: Clusters $A_1, \ldots, A_K$ with $A_i = \{j | \mathbf{y}_j \in C_i\}$

Additionally, another spectral clustering algorithm is proposed for comparison, the approach consists on learning a new weighted adjacency matrix $\mathbf{W}$ from the Hi-C matrix through the graph learning method and applying the classic spectral clustering procedure to produce the labeling of the nodes, e.g the coordinates or locus of the chromosomes graphs. In this case, the weight edges of the adjacency matrix are not signed (i.e $w_{i,j} \in \mathbb{R}_+$), such that the formal assumptions

for spectral graph theory are fullfilled. The algorithm computes an eigendecompostion of the Laplacian matrix and selects the $K$ eigenvectors associated to the smallest eigenvalues to form the clustering embedding for k-means. The procedure is detailed in Algorithm 3.

**Algorithm 3** Graph learning based Spectral clustering

**procedure** GRAPH LEARNING BASED SPECTRAL CLUSTERING
  ***Input***: Normalized Hi-C matrix $\mathbf{U}_c \in \mathbb{R}^{L_c \times L_c}$, number of clusters $K$, graph sparsity level k
  Scale and center $\mathbf{U}_c$ to have zero mean and unit variance.
  Compute the Euclidean distance matrix, $\mathbf{Z}_c$, of $\mathbf{U}_c$.
  Apply the graph learning procedure to generate a weighted adjacency matrix, called $\mathbf{W}$, with a graph sparsity level of approximately $k$ neighbours per node.
  Construct the normalized laplacian $\mathbf{L}_N$ of $\mathbf{W}$ .
  Compute the first $K$ eigenvectors $\{\mathbf{v_1}, \ldots, \mathbf{v_K}\}$ associated to the $K$ smallest eigenvalues of $\mathbf{L_N}$
  Build the matrix $\mathbf{V} \in \mathbb{R}^{N \times K}$ with the eigenvectors as columns.
  For $i = \{1, \ldots, N\}$, let $\mathbf{y}_i \in \mathbb{R}^K$ be the vector correspoding to the $i-th$ row of $\mathbf{V}$
  Cluster the points $\{\mathbf{y}_i\}_{i=1,\ldots,N}$ with $k$-means algorithm into clusters $C_1, \ldots C_K$
  ***Output***: Clusters $A_1, \ldots, A_K$ with $A_i = \{j | \mathbf{y}_j \in C_i\}$

As we will see in section 3.6, these two clustering tools will be employed to detect the self-interacting communities in the Hi-C maps.

## 3.5 Case study: Graph learning applied to chromosome 16

In this sub-section, a walk-through of the graph learning step of the spectral clustering algorithm is given. The initial dataset is the normalized intra-chromosomal contact matrix, $\mathbf{U}$, generated from the Hi-C experiments for chromosome 16. The heatmap of the normalized intra-chromosomal matrix can be visualized in Figure 3.8, the dynamic range of the intensity values has been reduced for better figure representation.

The first step towards learning a graph from the intra-chromosomal map is to compute the pairwise distances matrix , $\mathbf{Z}$, where $z_{i,j} = ||\mathbf{u_i} - \mathbf{u_j}||_2$ and $\mathbf{u}_i$ is the i-th column vector of $\mathbf{U}$.

The next step consists on applying the learning procedure defined in Equation (2.13), the so-called log model, to generate a weighted adjacency matrix $\mathbf{W}$. The inputs to this method are the Euclidean distance matrix $\mathbf{Z}$ and parameter $\theta$, that is automatically set by specifying the desired graph sparsity level, $k_s$, as explained in Section 2.2.2. The convex optimization solver for this procedure is explained in [10]. The edge mask used to restrict the set of allowed edges is defined in Section 3.5.1.

### 3.5.1 Edge mask selection

Recall that the intra-chromosomal maps are matrices of moderate size. In this sub-section, a binary edge mask based on the structure of the Pearson correlation matrix is introduced, the masked

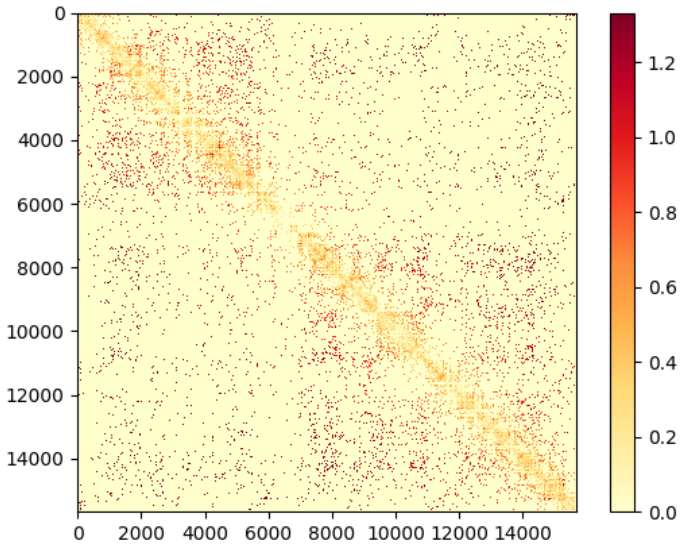


Figure 3.8: Normalized intra-chromosomal contact matrix **U** of chromosome 16

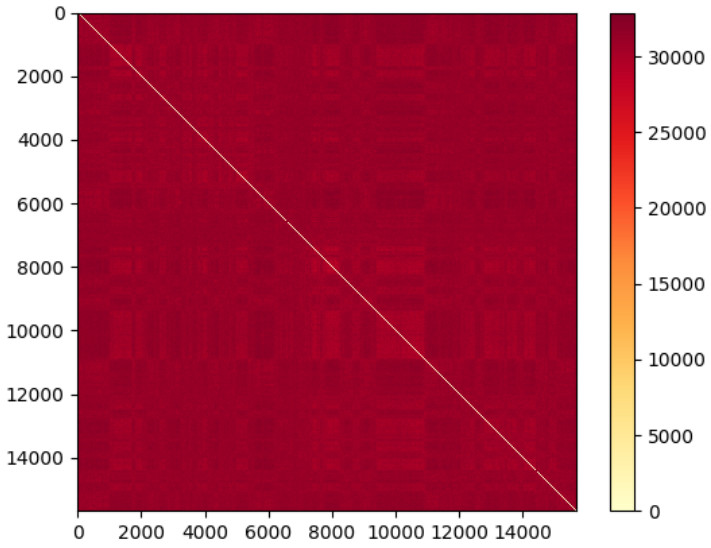


Figure 3.9: Normalized pairwise Euclidean distance matrix **Z**

edges are ignored by the learning algorithm, thus alleviating the computational complexity of the process. At first glance, the matrix has very well defined plaid pattern that reveals community structure between nodes. We will use this prior information to filter out edges that are not needed for our enhanced weighted adjacency matrix.

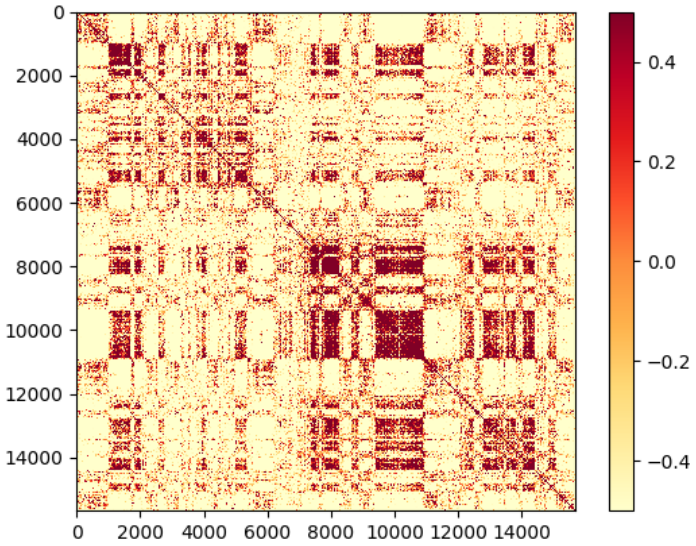


Figure 3.10: The Pearson correlation matrix for chromosome 16

Since we want to connect pairs of loci with similar interaction patterns with the rest of the genome, a threshold $\gamma$ is used to filter out low values of the pairwise correlation between bins. Let $\mathbf{M} \in \mathbb{Z}_2^{L_c \times L_c}$ be the binary edge mask and $\mathbf{C}$ the Pearson correlation matrix, the entries of the mask are computed as follows

$$m_{i,j} = \begin{cases} 1 & if\ c_{i,j} \geq \gamma \\ 0 & otherwise \end{cases} \tag{3.4}$$

The sparsity of the resulting binary edge mask, defined as the number of non-zeros divided by the total number of entries, denotes the fraction of allowed edges for the optimization algorithm.. For different values of $\gamma$, the resulting sparsities are shown in Table 3.1.

To get an idea of the overall distribution of the Pearson correlation coefficients, we plot the histogram of the Pearson correlation in Figure 3.11.

The threshold $\gamma$ is set to have sufficient coverage for the parameter $k_s$ controlling the number of neighbouring edges per node in the optimization procedure. The cardinality of the set of allowed edges in the mask must be larger than $k_s L_c$, where $L_c$ is the number of nodes in the adjacency matrix of the intra-chromosomal Hi-C map.

| $\gamma$ (threshold) | Sparsity ratio of edge mask |
|---|---|
| -0.10 | 0.2374 |
| 0.05 | 0.1893 |
| 0.25 | 0.1373 |
| 0.5 | 0.089 |

Table 3.1: Sparsity ratio of the edge mask for different Pearson coefficients based thresholds

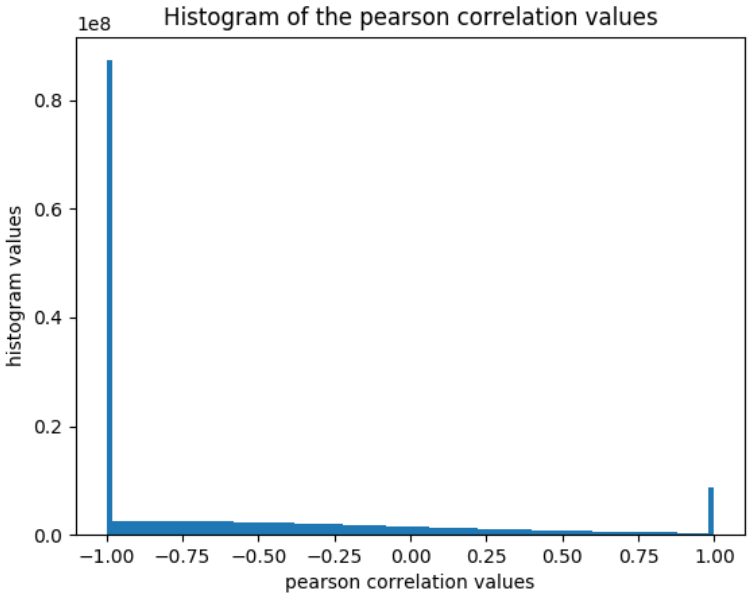


Figure 3.11: The histogram of the Pearson correlation matrix for chromosome 16

### 3.5.2 Learned weighted adjacency matrix of the intra-chromosomal Hi-C map

Under the Pearson correlation threshold criteria, for a threshold $\gamma = 0.05$, we present the learned weighted adjacency matrices for different values of $k_s$. It is worth noting that with the use of an edge mask the running time of the procedure is significantly reduced.

Figure 3.12: Smoothed weighted adjacency matrix for different graph sparsity levels

A value of $k_s$ must be selected in order to proceed with the clustering algorithm. The value $k_s$ must be kept low to maintain the sparnesses of the adjacency matrices. For the range of chromosomes $1, \ldots, N_{chr}$, the parameter $k_s$ for each chromosome is selected empirically and shown in Table 3.2.

| Chromosomes | $k_s$ |
|---|---|
| 1-12 | 3000 |
| 13-18 | 1500 |
| 19-22 | 500 |

Table 3.2: Value of $k_s$ selected for different ranges of chromosomes

The threshold $\gamma$ of the correlation matrix to form the edges mask of each chromosome is automatically set to give enough room for the learning method to output the desired final sparsity.

## 3.6 3D spatial clustering pipeline

In this section, the strategy followed to inspect the 3D genome snapshot of this Jurkat-wild type cell at the 5 kbp resolution of the available HiC maps is presented. A data processing approach inspired in [20] is used to obtain a few global genome communities at the nuclear level, beyond the intra-chromosomal level interactions, that describe the 3D structure of the analyzed cell. The clustering strategy is divided into two steps,also represented in Figure 3.13:

- **Dimensionality reduction**: the first step consists inreducing the high dimensionality of the intra-chromosomal maps by producing a set of clusters $\{A_1, \ldots, A_{K_1}\}$ for each intra-chromosomal Hi-C map. This procedure is repeated for each of the chromosomal Hi-C maps, a total of $N_{chr}$ times.

- **Genome wide clustering**: By summing up the interactions among the clustered chromosomal loci of all chromosomes, a normalized inter-chromosomal map describing the density of contacts between loci in different chromosome is generated. Then, by using a clustering method of choice, a set of global compartments $\{I_1, \ldots, I_{K_2}\}$ at genome-wide scale are identified on this inter-chromosomal map.

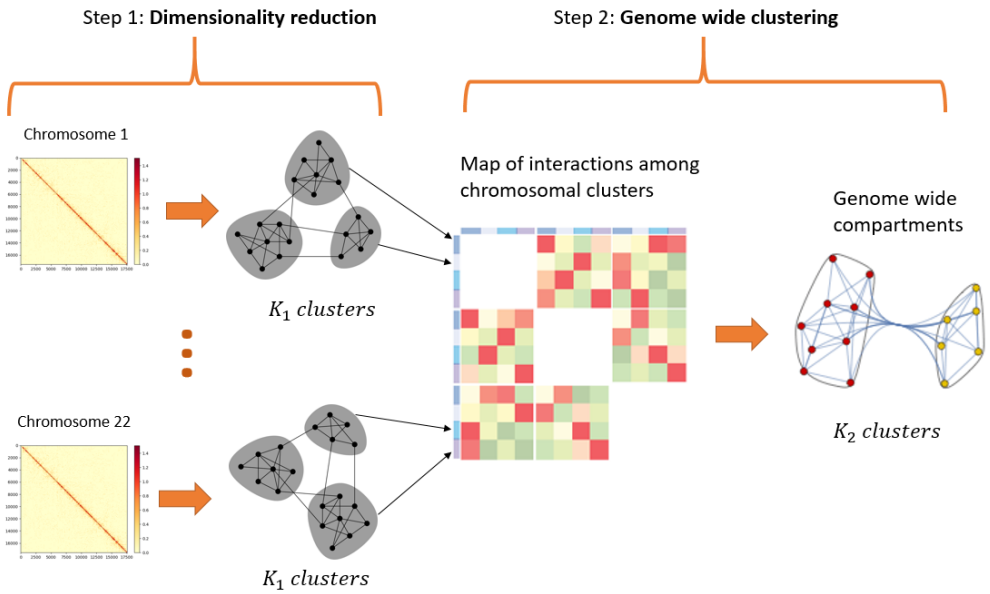


Figure 3.13: Clustering strategy yielding genome wide communities over the set of $N_{chr}$ chromosomes

### 3.6.1 Dimensionality reduction

The spatial segmentation of the Jurkat genome begins by applying a spectral clustering method on the intra-chromosomal maps of all chromosomes. The parameter $K_1$, defining the number of clusters to detect, dictates the dimensionality reduction factor, thus representing the coarseness of the grid where all inter-chromosomal combinations are represented. The intra-chromosomal Hi-C matrices are pre-processed, as explained in Section 3.2, to add robustness to the posterior clustering and eliminate bias such as the linear genome distance effect. Depending on the clustering method of choice, different kinds of adjacency or distance matrices are firstly generated.

If correlation based clustering is used, the Pearson correlation matrix of the normalized intra-chromosomal Hi-C map $\mathbf{U}_c$ of a given chromosome $c$ is computed. On the other hand, when employing the graph learning procedure, an enhanced adjacency matrix is derived from $\mathbf{U}_c$ through an optimization method. In order to reduce the computational complexity of the learning algorithm at this step where the high dimensionality of the Hi-C data imposes moderate and large matrix sizes, an edge mask based on the Pearson correlation of these maps is used to reduce this complexity and filter out unnecessary edges, which are automatically set to zero during the learning process.

The first round of clustering contains the major computational bottlenecks of the whole pipeline. The length of chromosomes can be in the order of tens of thousands of 5kbp bins. For example, the length of chromosome 1 results in $L_1 \approx 50000$. In particular, the clustering phase is dominated by the spectral decomposition of the adjacency matrix $\mathbf{W}$ and the computation of the pairwise Euclidean distance matrices and the Pearson correlation matrices. Their footprint in

memory can also be notable. This eigenvalue decomposition computation can be performed more efficiently by only computing a few of the first smallest eigenvalues and the associated eigenvectors instead of the whole spectrum.

### 3.6.2 Genome wide clustering

After the first round of clustering, a new weighted adjacency matrix $\mathbf{W}_G$ is built, representing a graph where its nodes are the intra-chromosomals clusters and the weighted edges reflect a normalized interaction score between communities of different chromosomes. Each $(i, j)$-th entry of $\mathbf{W}_G$ is the aggregated count of inter Hi-C reads between the set of loci in community $A_k$ of chromosome $i$ and the set of loci in community $A_l$ of chromosome $j$, where $i \neq j$ and $k, l \in \{1, \ldots, K_1\}$. Recall that this information is made available via the inter-chromosomal Hi-C maps of the Jurkat cell. This aggregated count is normalized by the sizes of the communities such that the units of the matrix values are [Hi-C reads / $kbp^2$], allowing communities of different loci spans to be fairly compared. This normalization results into measuring densities of contacts in a 2D region. The matrix $\mathbf{W}_G \in \mathbb{R}^{(K_1 N_{chr}) \times (K_1 N_{chr})}$ is symmetric and can be interpreted as the adjacency matrix of a graph. The heatmap of $\mathbf{W}_G$ is shown in Figure 3.14.

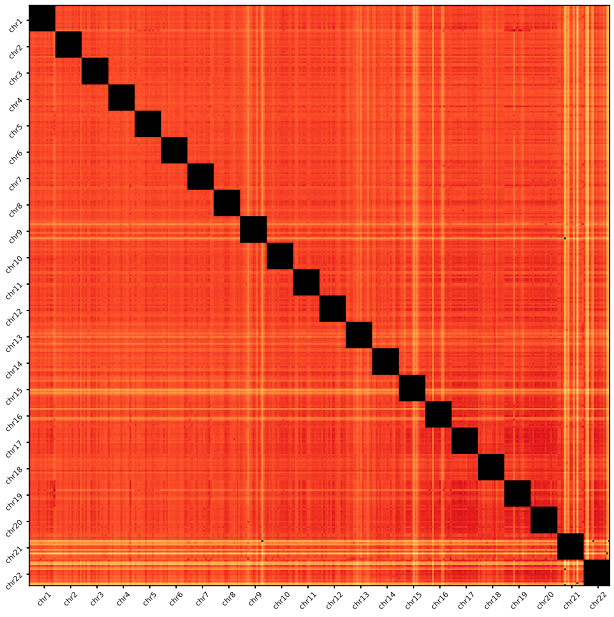


Figure 3.14: Log-scale heatmap of the inter-chromosomal Hi-C density score matrix

Please notice that the aggregated interactions among intra-communities are not considered and furthermore eliminated as they add no relevant information of the overall organization at the

nuclear genome scale and they would mask patterns

The final round of clustering is performed on $\mathbf{W}_G$. The goal is to find $K_2$ clusters describing the 3D overall architecture of the genome. At this stage, there is no need for OE normalization as we are only considering inter-connections between locus of different chromosomes, so the linear genome distance effect is not present.

As in the first stage of the 3D spatial clustering approach, depending on the requirements of the clustering method, a different type of adjacency matrix is employed as input to the spectral decomposition procedure. Therefore, when using the correlation-based clustering algorithm, the Pearson correlation matrix of the inter-chromosomal map $\mathbf{W}_G$ is computed. The Pearson correlation matrix, shown in Figure 3.15, can be interpreted as a signed weighted adjacency matrix that highlights hubs of chromosomal communities in the whole genome map and sharpens transitions among chromosomal clusters behaving differently with the rest of the nucleus, indicating they might not be co-located. A higher correlation coefficient in absolute value terms shows that these chromosomal clusters might lie close together in the 3D space of the nucleus.

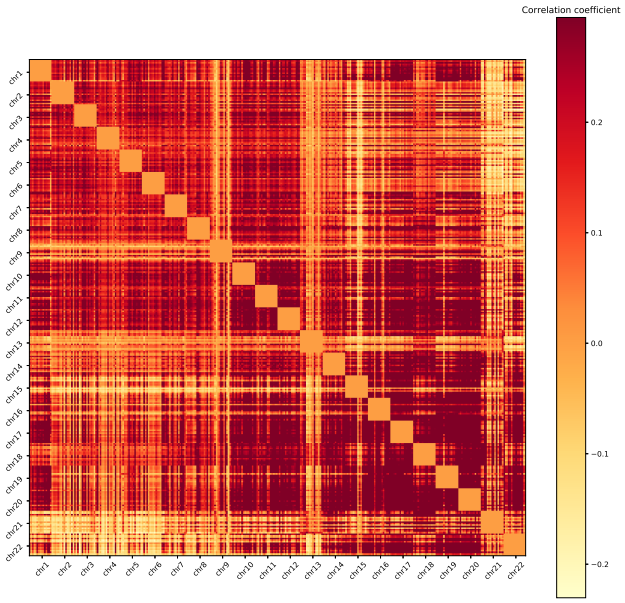


Figure 3.15: Heatmap of the Pearson correlation matrix of $\mathbf{W}_G$

The spectral clustering algorithm based on graph learning applies an optimization procedure to learn a weighted adjacency matrix from data prior to the spectral decomposition phase. In this case, the data is the matrix $\mathbf{W}_G$ and the goal is to produce an enhanced weighted adjacency matrix version, more suited for a clustering method to discover structure. Recall that the desired

graph sparsity level must be given as input to the learning algorithm. A value of $k_s = 50$ is empirically chosen. It is preferable to set $k_s$ large enough, so that the algorithm itself decides whether an edge weight should be zero or not. The heatmap of the learned adjacency matrix is depicted in Figure 3.16.

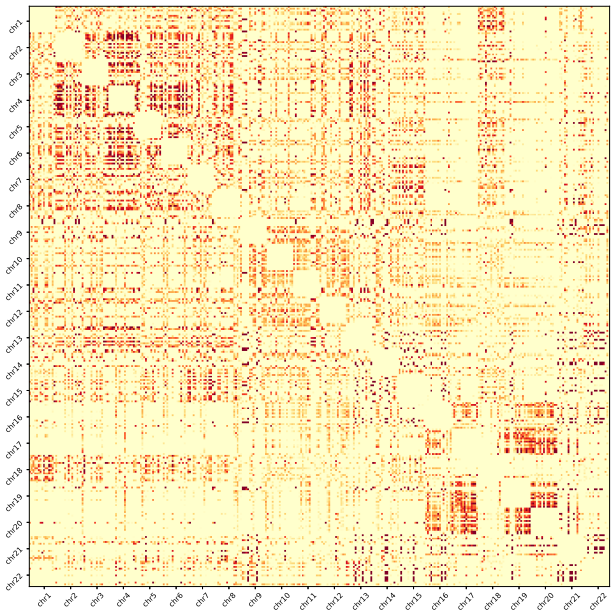


Figure 3.16: Heatmap of the learned inter-chromosomal adjacency matrix

### 3.6.3 Results

In this section, we conduct experiments to evaluate the genome wide clustering. Both spectral clustering algorithms are analyzed and compared. It is particularly interesting to extract insights from the final round of clustering. The main target of our work to find community structure on the overall spatial organization of the Jurkat cell. Several methods exists to test the performance the clustering methods; we will focus mainly on a quantitative characterization of the results of the k-means partitioning by exploring the optimal number of clusters $K_2$ through metrics such as the silhouette index, the k-means objective function and the analysis of the eigenvalue spectrum of the Laplacian matrix. Although it is important to be cautious about what exactly is optimal in this biological experiment setting, a quantitative analysis will help us establishing the ground rules for algorithm comparison. Then, a biological inspired metric called A/B score will be used to quantify the two types of chromatin present in the genome wide communities, such that an active/non-active genomic score can be assigned to each detected cluster. The latter will help us

identify whether meaningful structure is captured or not. Finally, the HIV insertion distribution throughout the genome-wide communities will be studied.

In the first round of clustering, namely the dimensionality reduction stage, k-means is employed on the spectral embedding of the intra-chromosomal adjacency matrices of all chromosomes with the number of clusters to search for set to $K_1 = 15$. As it is known, the quality of k-means clustering is quite sensitive to the selection of initial number of cluster centers, thus the algorithm is randomly initiated ten rounds to add robustness to the final clustering.

In the second round of clustering, k-means with ten random restarts is also employed on the spectral embeddings obtained with both clustering methods. At this point, $K_2$ needs to be set manually which poses a significant challenge if the cluster structure is not pronounced. To begin with the clustering analysis, we start by analyzing the well-known elbow method to evaluate the consistency and the goodness of the K-means procedure. It also helps identifying the appropriate number of clusters. This method evaluates the K-means objective function that the clustering procedure is trying to minimize with the number of clusters as a hyper-parameter. For both spectral clustering methods, the evaluation of the K-means cost function with the number of clusters is represented in Figures 3.17 and 3.18. Results indicate that the number of communities that minimize the cost function is very low and both algorithms show a very clear trend, increasing the number of clusters does not result on a better k-means score.

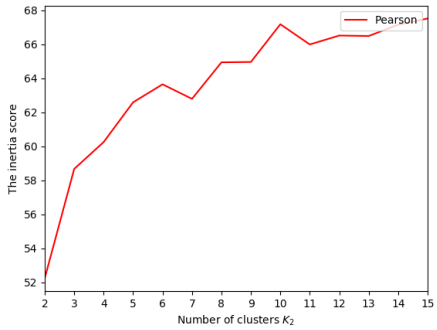


Figure 3.17: Spectral clustering based on Pearson correlation: k-means cost function evaluation

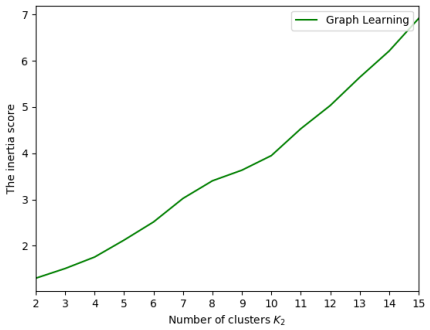


Figure 3.18: Spectral clustering based on graph learning: k-means cost function evaluation

Another tool to analyze the consistency within the detected clusters is the silhouette analysis, which explores the separation distance between the resulting clusters. The silhouette coefficient quantifies how close each sample in one cluster is to samples in the neighboring clusters and thus provides a way to assess the number of clusters visually. In this work, samples refer to the $N_{chr} \times K_1$ chromosomal communities. The measure has a range of [-1, 1]. To interpret the measure, positive values show that the sample is far away from the neighboring clusters. A value of 0 indicates that the sample is very close to the decision boundary between two neighboring clusters and negative values indicate that those samples might have been assigned to the wrong cluster.

The silhouette metric is computed as the average of the silhouette coefficients of all samples. The silhouette plot is displayed in figure 3.19. Results from both spectral clustering methods are consistent with the elbow method, a higher number of clusters does not correspond with better silhouette score, rather indicating that the number of clusters present on the inter-chromosomal graph is low.

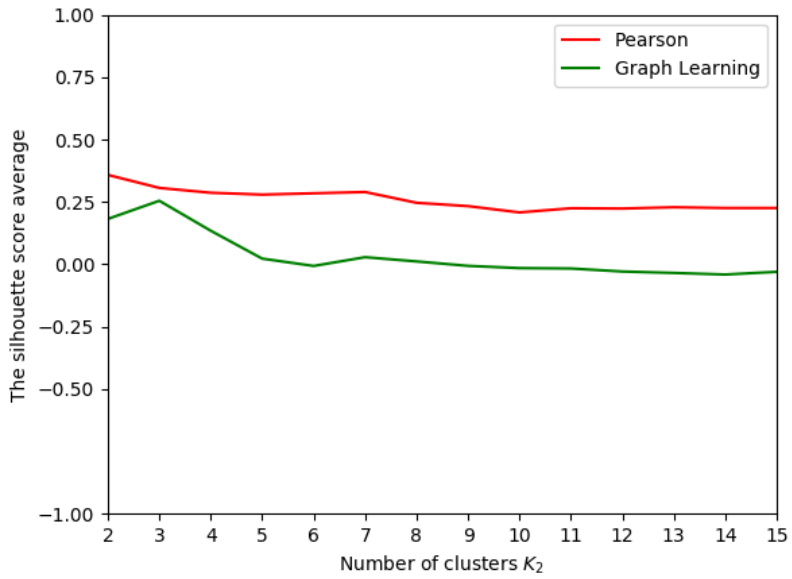


Figure 3.19: Silhouette analysis for both spectral methods

In the graph learning based spectral clustering algorithm, the Laplacian matrix of the adjacency matrix is computed. The eigenvalue spectrum of the Laplacian matrix provides another way of estimating a proper $K_2$ through an heuristic called eigengap [17]. Intuitively, only the first eigenvalue of this Laplacian matrix equals exactly 0, $K_2 - 1$ eigenvalues are practically equal to 0, and the rest is significantly different from 0. Thus, indicating that the position with the largest absolute difference value between successive eigenvalues is the right number of clusters. The eigengap is displayed in Figure 3.20. The greatest absolute difference between successive eigenvalues is found at a $K_2 = 1$, indicating that the Laplacian has a single block structure. The first eigenvalue of the Laplacian matrix clearly corresponds to 0.

The clustering validation performed previously reveals that there is no clear definition of the optimal number of clusters, concluding that no strong connected components appear on the genome wide graph. To continue with the analysis, we manually set $K_2 = 5$, a relatively low number of clusters. A particularly interesting property is how the first round chromosomal communitiesare distributed throughout the detected $K_2$ communities, in both the spectral graph clustering methods. The distribution is plotted in Figure 3.21 and shows that there are no prominent clusters acquiring most of nodes, which in turn is reasonable because k-means tends to output compact clusters. Both algorithms achieve a fairly similar distribution of nodes in their respective

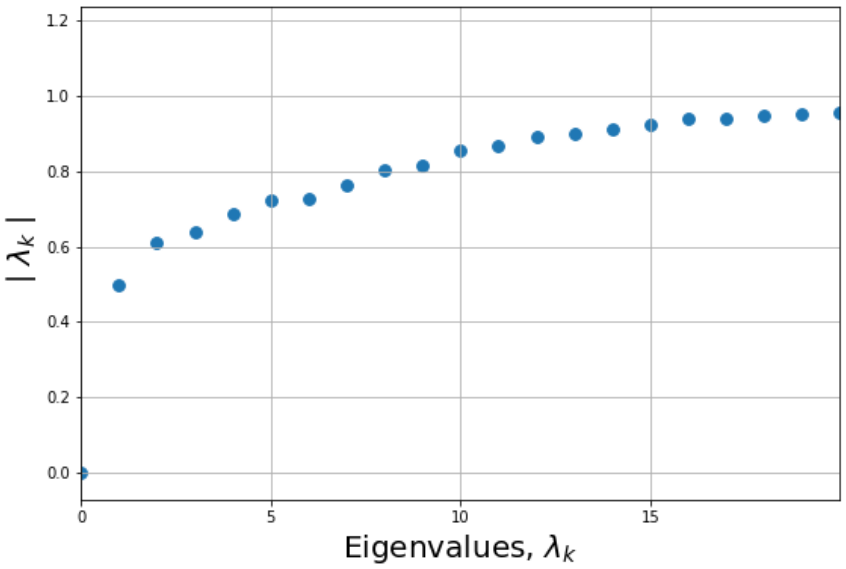


Figure 3.20: Eigenvalue spectrum of the laplacian matrix in the graph learning based clustering

communities. Notice that clusters are ordered and labeled by size.

The A/B score is derived from the eigenvector associated to the highest eigenvalue of the Pearson correlation matrix, the vector $\mathbf{v}_1 \in \mathbb{R}^{L_c}$. Recall that, at low base pairs resolution, these intra-chromosomal maps exhibit a binary community structure, namely two regions called A and B can be found. The A/B reference regions are set to be the top/bottom 10% entries of $\mathbf{v}_1$ respectively. This results in two sets $\mathcal{A} = \{i : v_1(i) \geq P_{90}\}$ and $\mathcal{B} = \{i : v_1(i) \leq P_{10}\}$, where $P_p$ denotes the $p$-th percentile of the vector $\mathbf{v}_1$, denoting the genomic bins within the A and B reference regions respectively. To compute the A/B score, for each row of the Pearson correlation matrix, the following quantities are computed as the sum of the correlation row values over the A and B reference regions, respectively, as

$$\begin{aligned} \mathbf{a}_s(i) &= \sum_{j\in\mathcal{A}} \mathbf{C}_{ij}\,, \quad \mathbf{b}_s(i) = \sum_{j\in\mathcal{B}} \mathbf{C}_{ij} \\ \mathbf{a}_m(i) &= \sum_{j\in\mathcal{A}} \left|\mathbf{C}_{ij}\right|\,, \quad \mathbf{b}_m(i) = \sum_{j\in\mathcal{B}} \left|\mathbf{C}_{ij}\right| \end{aligned} \tag{3.5}$$

Finally, the A/B score is calculated as follows

$$\mathbf{AB}(i) = \frac{\mathbf{a}_s(i) - \mathbf{b}_s(i)}{\mathbf{a}_m(i) + \mathbf{b}_m(i)} \cdot 100 \tag{3.6}$$

where $\mathbf{AB} \in \mathbb{R}^{L_c}$ is the A/B score vector of a given chromosomal Pearson correlation matrix. This vector is obtained for all the $N_{chr}$ chromosomes, producing an A/B quantity for the all the individual loci available from all chromosomes. This measure has a range of [-100,100], yielding values between 100 for A-like regions and -100 for B-like regions. Ambiguous regions that are

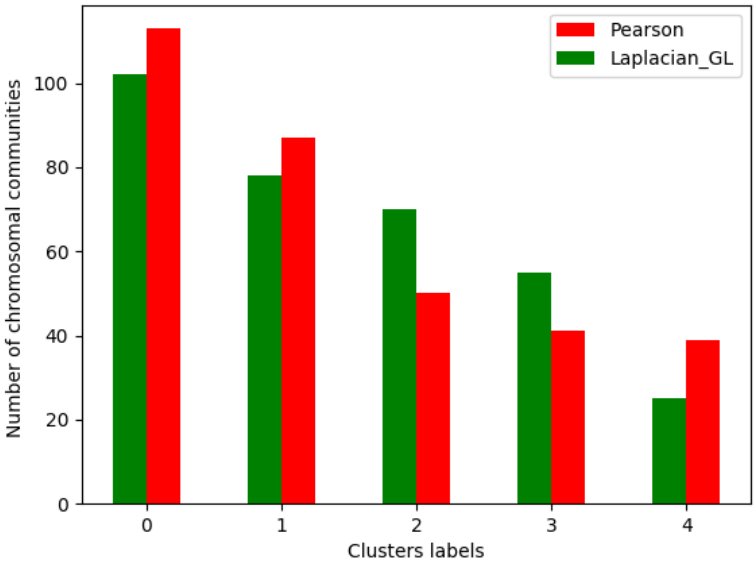


Figure 3.21: Clusters node distribution for $K_2 = 5$ ordered by community size

identically in contact with the reference A and B regions, or that might not be in contact with them at all, will have AB scores close to 0. Up to this point, we have access to a catalogue of A/B scores for each loci or genomic bins of all the chromosomes. To study the A/B properties of the resulting clusters, we analyze the distribution of A/B score in the genomic loci of the genome-wide communities through the statistical tool named box plot. A box plot is a standardized way of displaying the distribution of data that summarizes key properties such as the median, the minimum and maximum, how tightly the data is, and if and how data is skewed.

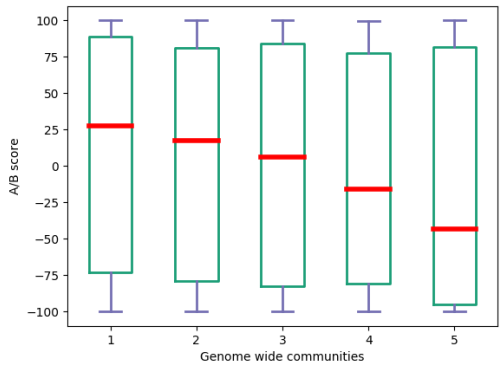


Figure 3.22: A/B score distribution for the Correlation based spectral clustering

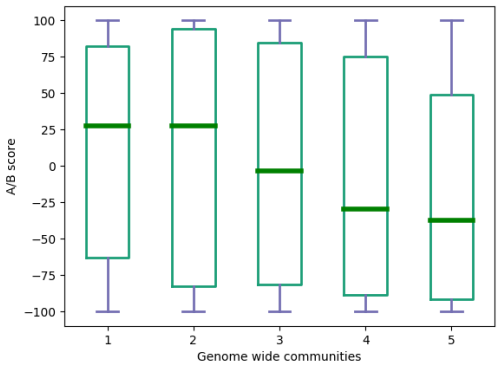


Figure 3.23: A/B score distribution for the Graph learning based spectral clustering

The red and green lines represent the medians of the A/B score distribution. The top and bottom sides of the rectangles represent the lowest and highest A/B values. The A/B scores are

ordered by the value of their medians. These A/B scores help identifying what kind of chromatin composes the genome wide communities. Ideally, a partitioning that reveals very concentrated A/B scores could indicate some form of meaningful clustering. But, even if the spread of values is significant in Figures 3.22, the medians still show a diverse scenario of the A/B genomic feature in this clustering. There are clusters that tend to have more A-like regions than B-like and the opposite as well. The A/B scores are ordered by the median value for better visualization. It is noticeable that both spectral algorithms yield very similar results.

Finally, we proceed to evaluate the HIV insertion distribution on these genome wide communities and see how the distribution correlates with the A/B score properties. To do so, the two independent lists of HIV pro-viruses insertion sites are processed. The genomic coordinates on the $N_{chr}$ chromosomes of each insertion site at the 5kbp resolution is available, thus a count of the total number of insertions per genome wide cluster can be extracted. Tables 3.3 and 3.4 reflect the analysis of HIV insertion on the detected communities for each spectral clustering method. The average density of HIV insertions on a particular community (i.e. normalized by community size in base pairs) compared to its A/B score median allows to asses whether the HIV virus has a tendency to cluster around A-like regions. The HIV insertion average density is computed as the average of HIV insertion densities of the two independent HIV insertion lists. The results indicate that communities with higher A/B score median have a higher density of HIV insertions. Note that clusters are arbitrarily labeled.

| **Pearson correlation** | | | | |
|---|---|---|---|---|
| Cluster labels | Cluster size (bins) | A/B score median | HIV insertions | HIV insertion average density |
| 0 | 40970 | 5.95 | 6194 | 0.076 |
| 1 | 68400 | 17.64 | 5915 | 0.043 |
| 2 | 51281 | -43.34 | 3668 | 0.037 |
| 3 | 213915 | -15.87 | 17453 | 0.041 |
| 4 | 158662 | 27.5 | 19064 | 0.061 |

Table 3.3: Pearson correlation based spectral clustering: HIV insertions average density and medians of A/B scores for each genome wide cluster label

| **Graph learning** | | | | |
|---|---|---|---|---|
| Cluster labels | Cluster size (bins) | A/B score median | HIV insertions | HIV insertion average density |
| 0 | 151221 | -3.2 | 10328 | 0.035 |
| 1 | 151092 | -29.61 | 19062 | 0.060 |
| 2 | 151193 | 27.33 | 8244 | 0.025 |
| 3 | 23198 | -37.04 | 254 | 0.005 |
| 4 | 56524 | 27.32 | 14406 | 0.13 |

Table 3.4: Graph learning based spectral clustering: HIV insertion average density and medians of A/B scores for each genome wide cluster label

# Chapter 4

# Conclusions and future work

In this work, the 3D structure of the genome of a human Jurkat cell is characterized through a graph theoretic approach, by using spectral clustering methods, of the Hi-C available data. The Hi-C data describes the intra-chromosomals and inter-chromosomals interaction networks and serves as the starting point for modelling the 3D spatial organization of the genome. A two-step procedure is carried out to detect the genome-wide communities present in the nucleus of the cell, where most of DNA chromatin lies. Then by analyzing the resulting genome-wide communities, the overall 3D cluster structure turns out to be not so distinguished but for a relatively small number of clusters, results partially confirm a widely known assumption, the integration of HIV virus in the human genome is non-random. The provirus preferentially targets gene-rich chromosomes, chromatin regions that are transcriptionally active. The 3D spatial organization for the genome is known to have a major role on the behaviour of the HIV virus and understanding this relationship could be key to design a cure for it. The main obstacle to cure HIV is the presence of latent or silent virus in infected patients, the mechanisms of latency are far more complex than the mechanisms of HIV integration, but they may also be induced by chromatin structure context. Thus the expression sites of HIV do not exactly correspond with the insertion sites of HIV. In general, these insights may help construct better antiretroviral therapies (ART), the standard method to suppress HIV, and others strategies. Current therapies are able to suppress HIV replication mechanisms to undetectable levels in the blood, however reservoirs of silent virus are invisible to this treatment and HIV can rapidly bounce back into replicating. In summary, the HIV latency/expression patterns (from silent to active or reverse) are far from being understood with this simple analysis of the 3D genome context.

Future work could follow many lines of research, but if the focus is put on mathematical modelling, the most promising are:

- Construct a different approach, instead of clustering intra-chromosomal Hi-C maps, cluster the inter-chromosomals Hi-c maps at a lower resolution, so that the interaction maps are less sparse, to detect the genome-wide communities.

- Explore different normalization procedures to enhance the Hi-C matrices. The pre-processing

of the Hi-C data is a fundamental step of this whole analysis.

# Bibliography


[1] T. Cremer, M. Cremer, S. Dietzel, S. Müller, I. Solovei, and S. Fakan, "Chromosome territories–a functional nuclear landscape," *Current opinion in cell biology*, vol. 18, no. 3, pp. 307–316, 2006.

[2] N. H. G. R. Institute, "Whole genome sequencing." [Online]. Available: http://knowgenetics.org/whole-genome-sequencing/

[3] B. R. Lajoie, J. Dekker, and N. Kaplan, "The hitchhikers guide to hi-c analysis: practical guidelines," *Methods*, vol. 72, pp. 65–75, 2015.

[4] S. Wang, J.-H. Su, B. J. Beliveau, B. Bintu, J. R. Moffitt, C.-t. Wu, and X. Zhuang, "Spatial organization of chromatin domains and compartments in single chromosomes," *Science*, vol. 353, no. 6299, pp. 598–602, 2016.

[5] M. Imakaev, G. Fudenberg, R. P. McCord, N. Naumova, A. Goloborodko, B. R. Lajoie, J. Dekker, and L. A. Mirny, "Iterative correction of hi-c data reveals hallmarks of chromosome organization," *Nature methods*, vol. 9, no. 10, p. 999, 2012.

[6] E. Lieberman-Aiden, N. L. Van Berkum, L. Williams, M. Imakaev, T. Ragoczy, A. Telling, I. Amit, B. R. Lajoie, P. J. Sabo, M. O. Dorschner *et al.*, "Comprehensive mapping of long-range interactions reveals folding principles of the human genome," *science*, vol. 326, no. 5950, pp. 289–293, 2009.

[7] J. R. Dixon, S. Selvaraj, F. Yue, A. Kim, Y. Li, Y. Shen, M. Hu, J. S. Liu, and B. Ren, "Topological domains in mammalian genomes identified by analysis of chromatin interactions," *Nature*, vol. 485, no. 7398, p. 376, 2012.

[8] E. P. Nora, B. R. Lajoie, E. G. Schulz, L. Giorgetti, I. Okamoto, N. Servant, T. Piolot, N. L. van Berkum, J. Meisig, J. Sedat *et al.*, "Spatial partitioning of the regulatory landscape of the x-inactivation centre," *Nature*, vol. 485, no. 7398, p. 381, 2012.

[9] F. R. Chung and F. C. Graham, *Spectral graph theory*. American Mathematical Soc., 1997, no. 92.

[10] V. Kalofolias and N. Perraudin, "Large scale graph learning from smooth signals," *arXiv preprint arXiv:1710.05654*, 2017.

[11] V. Kalofolias, "How to learn a graph from smooth signals," in *Artificial Intelligence and Statistics*, 2016, pp. 920–929.

[12] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in neural information processing systems*, 2002, pp. 585–591.

[13] N. Komodakis and J.-C. Pesquet, "Playing with duality: An overview of recent primal? dual approaches for solving large-scale optimization problems," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 31–54, 2015.

[14] R. Xu and D. C. Wunsch, "Survey of clustering algorithms," 2005.

[15] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Departmental Papers (CIS)*, p. 107, 2000.

[16] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[17] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[18] H.-C. Chen, J. P. Martinez, E. Zorita, A. Meyerhans, and G. J. Filion, "Position effects influence hiv latency reversal," *Nature structural & molecular biology*, vol. 24, no. 1, p. 47, 2017.

[19] S. S. Rao, M. H. Huntley, N. C. Durand, E. K. Stamenova, I. D. Bochkov, J. T. Robinson, A. L. Sanborn, I. Machol, A. D. Omer, E. S. Lander *et al.*, "A 3d map of the human genome at kilobase resolution reveals principles of chromatin looping," *Cell*, vol. 159, no. 7, pp. 1665–1680, 2014.

[20] B. Lucic, H.-C. Chen, M. Kuzman, E. Zorita, J. Wegner, V. Minnerker, V. Roukos, M. Benkirane, W. Weng, M. Schmidt *et al.*, "Spatially clustered loci with multiple enhancers are frequent targets of hiv-1," *bioRxiv*, p. 287896, 2018.

[21] G. P. Wang, A. Ciuffi, J. Leipzig, C. C. Berry, and F. D. Bushman, "Hiv integration site selection: analysis by massively parallel pyrosequencing reveals association with epigenetic modifications," *Genome research*, vol. 17, no. 8, pp. 1186–1194, 2007.

[22] J.-P. Fortin and K. D. Hansen, "Reconstructing a/b compartments as revealed by hi-c using long-range correlations in epigenetic data," *Genome biology*, vol. 16, no. 1, p. 180, 2015.