A method to recognize universal patterns in genome structure using Hi-C

Author
Neil Chowdhury (Phillips Exeter Academy, New Hampshire, USA)

Mentor
Sameer Abraham (Massachusetts Institute of Technology, Massachusetts, USA)

Biology Award

A method to recognize universal patterns in genome structure using Hi-C

Abstract
The expression of genes in cells is a complicated process. Expression levels of a gene are determined not only by its local neighborhood but also by more distal regions, as is the case with enhancer-promoter interactions, which can connect regions millions of bases away [1]. The large-scale organization of DNA within the cellular nucleus plays a substantial role in gene expression and cell fate, with recent developments in biochemical assays (such as Hi-C) generating quantitative maps of the higher-order structure of DNA. The interactions captured by Hi-C have been attributed to several distinct physical processes. One of the processes is that of segregation of DNA into compartmental domains by phase separation. While the current consensus is that there are broadly two types of compartmental domains (A and B), there is some evidence for a larger number of compartmental domains [2]**.** Here a methodology to determine the identity and number of such compartments is presented, and it is observed that there are four distinct compartments within the genome.

Keywords: Hi-C, Clustering, Compartmentalization, Dimensionality Reduction, Stability, ChIP-seq, Repli-seq

Contents

Introduction

Hi-C is a method of measuring the proximity between two regions in the DNA strand [3]. An overview of the Hi-C protocol as well as an example interaction matrix is shown in Figure 1. Each row and column represent regions on the genome and the pixel represents the frequency of interaction between these two regions.

A *region* refers to a position range in the genome. It is designated by chromosome, start position, and end position. A region can be any number of base pairs long. In this paper, a resolution of 1,000,000 base pairs will be used, which is suitable for clustering analysis on Hi-C matrices [4]. This means that all the regions will be 1,000,000 base pairs long.

A *contact probability* is a value that represents how many times two regions are found in “contact” in the cell nucleus, i.e. are physically adjacent within some effective radius that is captured and processed using Hi-C.

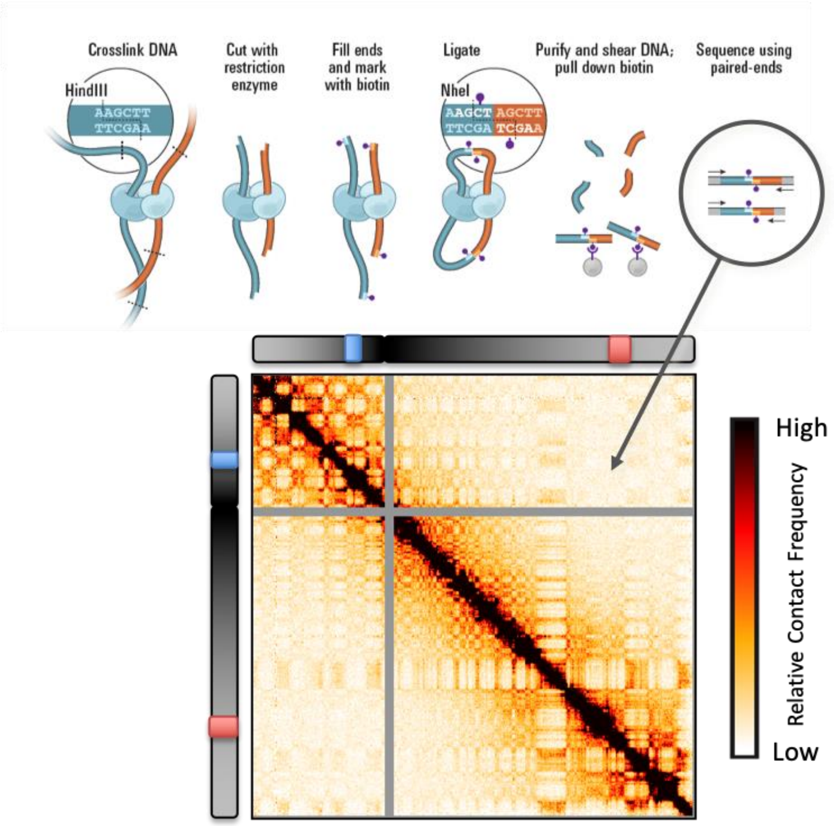

Crosslink DNA
Cut with restriction enzyme
Fill ends and mark with biotin
Ligate
Purify and shear DNA; pull down biotin
Sequence using paired-ends
HindIII
AAGCTT
TTCGAA
NheI
AAGCT AGCTT
TTCGA TCGAA
High
Relative Contact Frequency
Low

Figure 1. An example of a Hi-C matrix (bottom) as well as a description of the process used to create it (top).

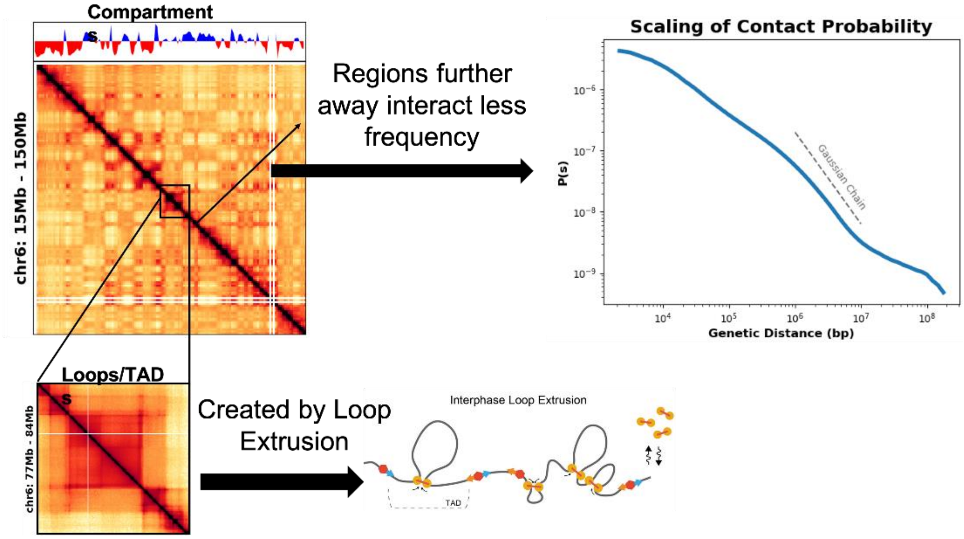

Compartment
chr6: 15Mb - 150Mb
Regions further away interact less frequency
Scaling of Contact Probability
P(s)
Genetic Distance (bp)
Gaussian Chain
Loops/TAD
chr6: 77Mb - 84Mb
Created by Loop Extrusion
Interphase Loop Extrusion
TAD

Figure 2. A visualization of compartmentalization, scaling of contact probability, and loops.

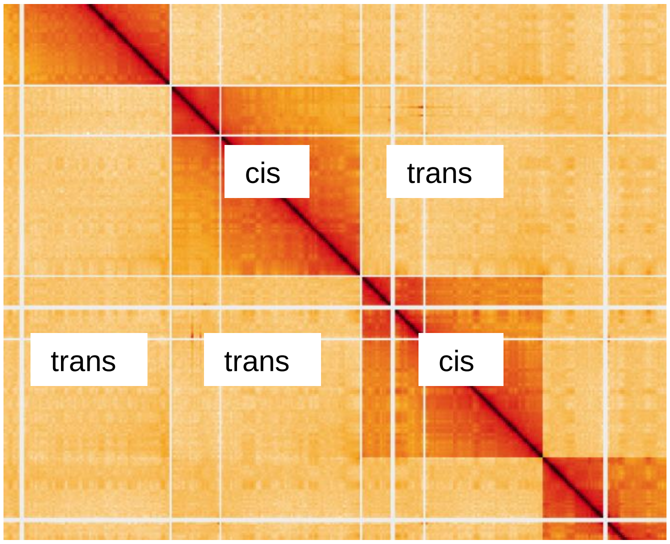


Figure 3. Cis and trans interactions in a section of a Hi-C matrix

Several features are present in Hi-C matrices. On the largest scales, interactions are differentiated into cis and trans interactions. Cis interactions occur within the same chromosome, while trans interactions occur between different chromosomes. Figure 3 illustrates that cis interactions are generally much stronger than trans interactions.

Within cis interacting regions, there are three broad types of features: scaling of contact probability, checkerboarding, and loops (see Figure 2). Segments that are far away from each other along the chromatin fiber compared to segments close together, resulting in a generally smaller contact probability, known as scaling. At large genomic distances, the process of phase separation and compartmentalization results in a checkerboarding pattern. At small scales, one can see enriched squares known as TADs bordered by dots with even higher enrichment (known as a loop). Dots and loops are created by the mechanism of loop extrusion [5], and there is significant evidence for this [6] [7] [8]**.**

In contrast to this, trans interactions contain only the checkerboarding patterns associated with compartmentalization. As the successful clustering relies on identifying only the compartmentalization pattern, trans interactions are the most relevant for this problem.

Context

Hi-C was a procedure invented in 2009, and it has revealed much about the three-dimensional conformation of the chromosomes [9]. Specifically, it is being used to investigate how the human genome is folded and packed within the cell nucleus. Hi-C has also found several other uses, such as in differentiating between different cells [4] and determining long-range chromatin interactions involving colorectal cancer risk loci [10].

Research contributions

Unsupervised clustering of Hi-C matrices can be explored to explain fundamental aspects of chromatin packing in the cell nucleus. The primary research contributions are:

- Presenting a new method of using unsupervised clustering of Hi-C matrices to determine characteristics of the structure chromatin within the cell nucleus.
- Presenting the results of the unsupervised clustering with respect to other biological signals.
- Presenting a strategy for evaluating the clustering of Hi-C matrices, which can be extended to any unsupervised clustering problem.

Experimental Methods

Clustering techniques will be used to establish the identity of compartmental domains. The idea behind this is that different compartments will have different interaction patterns with the rest of the genome – a signal that should be present in Hi-C and should be distinguishable by clustering techniques. Standard techniques used in data science are leveraged to cluster the data. Most clustering techniques treat data as points in a high dimensional vector space. The matrix can be made amenable to such techniques by choosing one axis (row or column) to represent different data points and the other to represent dimensionality.

Datasets Used

Most of the experiments will use the Hi-C matrix of mouse liver cells with the cohesin-loading factor NIPBL removed, created by Schwarzer et al [6]. Cohesin is a strong candidate for the extruder in the loop extrusion model, and Schwarzer et al. [6] observed that when cohesin is prevented from binding, the compartmental strength gets stronger. Even more interestingly, regions that behaved like a single compartment in the presence of cohesin seem to break up into two or more distinct compartments with different interaction patterns. Along with in-silico simulations [11], this seems to indicate that loop extrusion antagonizes the formation of compartmental domains. Thus, focusing on the NIPBL dataset allows development and study of the technique on a system where the signal of interest is the strongest.

Creating a matrix suitable for clustering

The problem with clustering the entire Hi-C matrix is that cis interactions (along the main diagonal) are significantly stronger than trans interactions, and there are other signals in cis interactions that interfere with compartmentalization. A matrix can be created that avoids cis interactions, and that allows us to cluster all the loci at once. The solution is a matrix formed by using odd chromosomes (chr1, chr3, …, chr23) as rows and even chromosomes (chr2, chr4, …, chr22) as columns. This method is described by Rao [2] as a way to analyze trans interactions between rows and columns. This matrix will be referred to as the *odd-even* matrix.

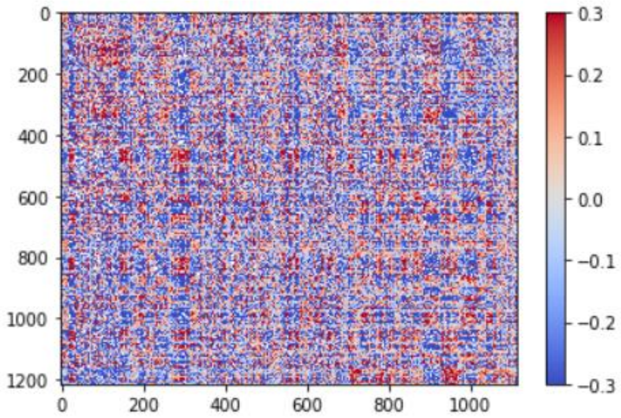


Figure 4. The matrix formed by trans interactions. (The color scale is described below.)

The matrix was normalized using iterative correction, which removes experimental biases from the Hi-C matrix and is part of the standard procedure for analyzing Hi-C matrices [12]. For plotting purposes, entries are divided by the mean of the matrix and then the logarithm is taken to produce data that approximates a Gaussian distribution with mean 0.

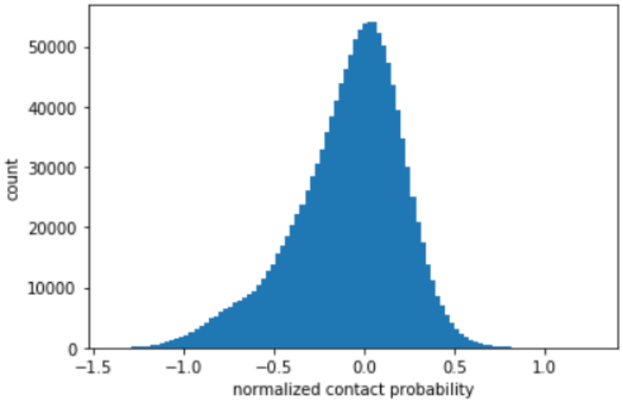


Figure 5. The distribution of contact probabilities after iterative correction, logarithmic transformation, and then dividing by the mean.

Clustering algorithms

The analysis will focus on three major clustering algorithms: k-means, agglomerative, and spectral, as implemented in the Python library Scikit-learn [13] [14]. K-means minimizes the distance from each point in a cluster to the center, agglomerative clustering is a hierarchical clustering algorithm, and spectral clustering embeds the affinity matrix before applying k-means clustering.

K-means works by partitioning the observations into *k* clusters for which each point is in the cluster with the nearest centroid. Initially, *k* points are randomly chosen to be means from which the *k* clusters are computed. At each iteration, the centroids of the *k* clusters are computed and become the new means. The algorithm iterates until convergence.

Spectral clustering starts by constructing an affinity matrix where each element corresponds to the affinity between two of the points, which can be measured using several kernels. The eigenvectors of this affinity matrix are computed, and then k-means is used to cluster the affinity matrix.

In agglomerative clustering, the initial state is each point being in its own cluster. Clusters are repeatedly merged by linkage criterion until the desired number of clusters is reached. For example, when ward linkage is used, the variance within a cluster is minimized (as measured by the sum of the squares of distances between all pairs of points). With average linkage, the pair of clusters with the smallest average of distances between each point is merged, and with single linkage, the two clusters with the closest distance between any pair of points are merged. Additionally, cosine distance is used, which serves as a distance metric for single and average linkage; the “distance” between two points is defined as the cosine of the angle between their position vectors.

Methods of evaluating clustering quality

Most clustering techniques require specifying the number of clusters to find. Since the true number of different compartment domains present in Hi-C is not known, the following tools must be developed to infer the quality of the clusters:

1. Visualization
2. Eigendecomposition
3. Stability
4. ChromHMM labels
5. Chromatin signals

Visualization

The clusterings in this paper are visualized using the following procedure:

1. Run a clustering technique on the rows and the columns of the odd-even matrix.
2. Sort the matrix by row and column according to cluster label
3. Plot the matrix. The colormap used is the same as previously; blue cells correspond to a negative normalized contact probability, while red cells correspond to a positive normalized contact probability. Thus, red cells represent two regions that are far from each other, while blue cells represent two regions that are close (interact heavily).

Furthermore, the clustering labels are shown in a row above the matrix and a column to the left of the matrix to aid in viewing the discrete clusters. The colors used may be arbitrary.

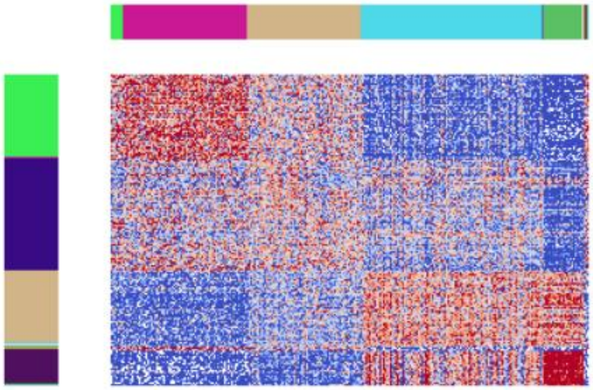


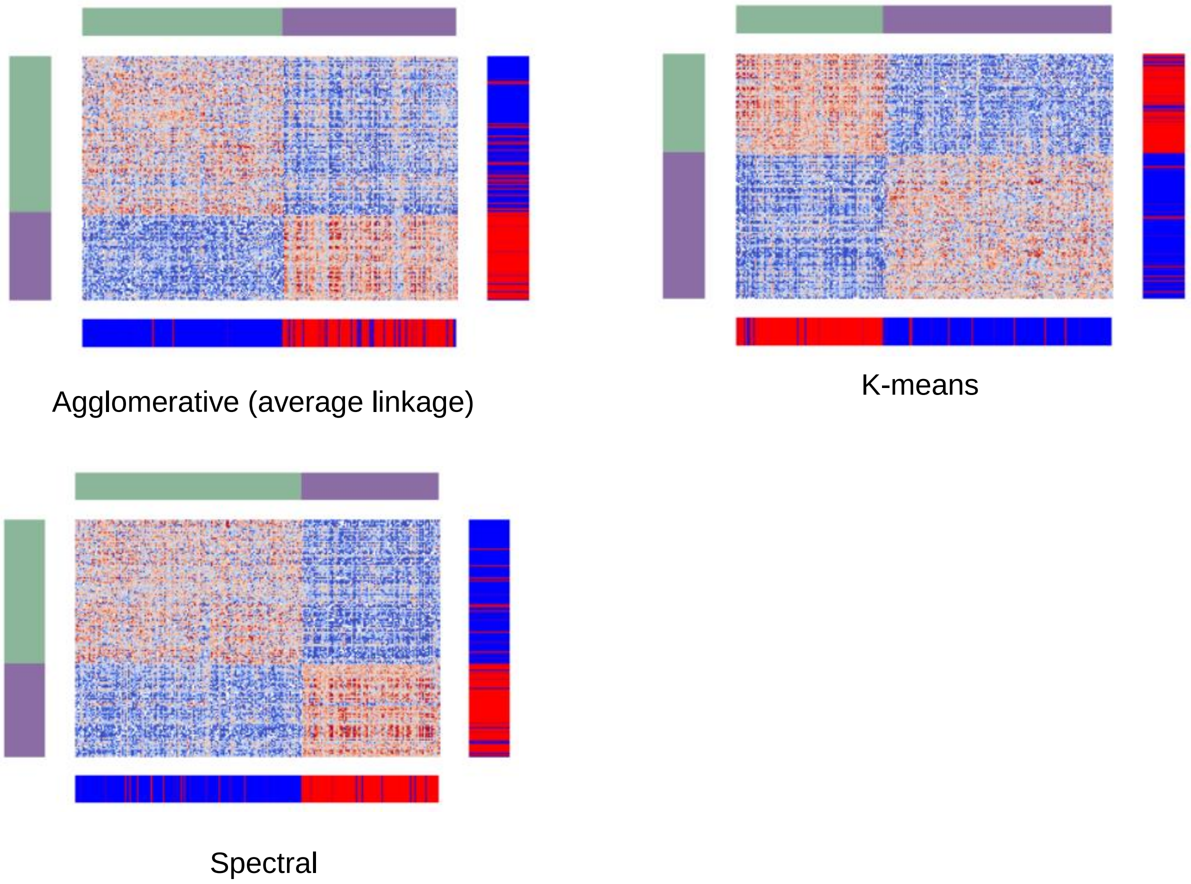
Figure 6. An example visualization.

The visualization example in Figure 6 shows the results of a clusterer that found four large clusters.

Eigendecomposition

The base Hi-C matrix is square and can be eigendecomposed using the library in [12]. The sign of the first eigenvector is taken, and regions with a positive sign are type A and ones with a negative sign are type B. A good clustering creating two clusters should distinguish between the type A and type B compartments.

The clustering methods can be tested by trying to find two clusters and seeing how well they match with type-A and type-B regions [13].



Agglomerative (average linkage)

K-means

Spectral

Figure 7. Results of finding 2 clusters in the odd-even matrix.

The three clustering methods embedded the type A and type B compartments well, as shown in Figure 7.

Evaluating the number of major clusters

Stability

Stability is a metric that can be applied to any clustering algorithm to help determine the number of major clusters in a dataset. In essence, stability analysis works by constructing perturbed versions of the dataset, running a clustering algorithm on both the original and perturbed versions, and then comparing the similarity of the two sets of labels generated. The stability is then compared for differing numbers of clusters to find the most stable clustering. Ulrike proposed several methods for generating perturbed versions of the dataset [15]. Two of them apply to this problem: subsampling and addition of noise.

The training matrix can be subsampled to include only a portion of it (75% is used for further analysis). Then, one clusterer is fit to the subsample and another to the original matrix. The labels for the two clusterers on the subsample are compared using the adjusted Rand index, a measure of correlation between two different labelings of the same data which ranges from 0 to 1 [16]. The process is repeated 25 times and the mean Rand index over all trials is calculated.

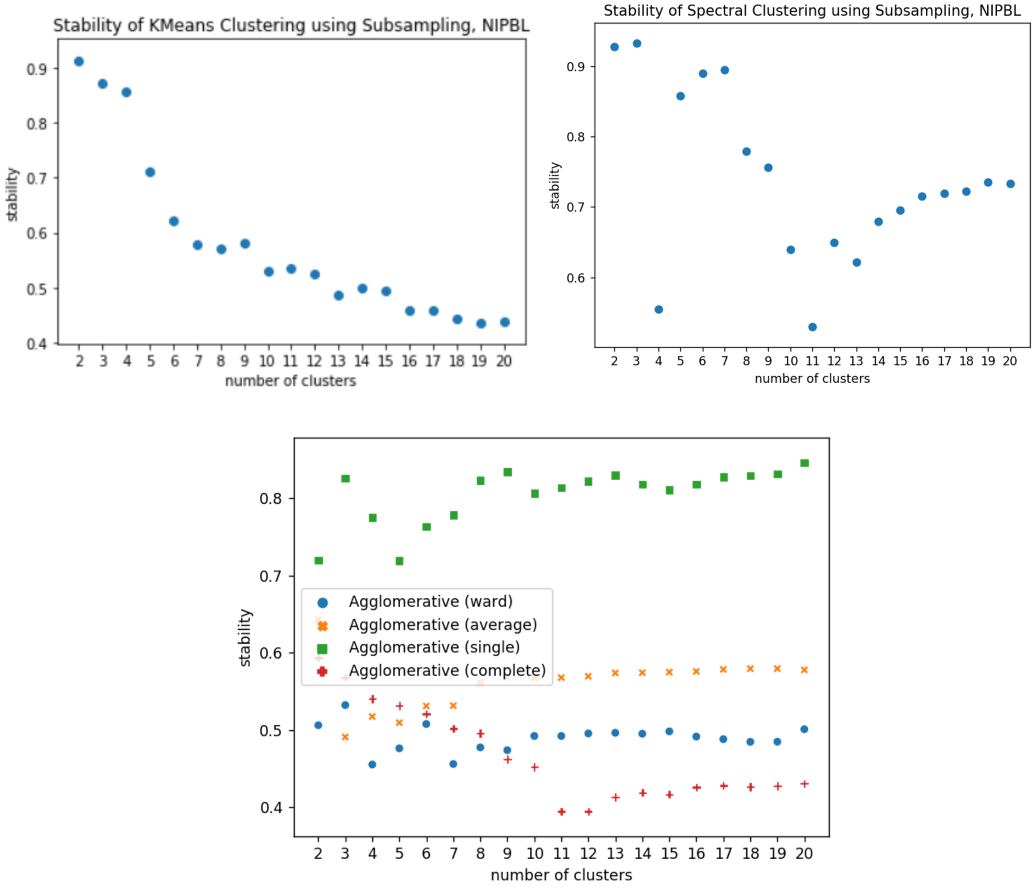


Figure 8. The results of stability analysis using subsampling.

The distribution of stabilities for k-means is the cleanest, where k=2, k=3, and k=4 are stable, while $k \geq 5$ becomes increasingly unstable as *k* increases. Spectral clustering is stable for $k \leq 7$, although it is unexpectedly unstable for k=4. Agglomerative clustering with ward linkage and average linkage have very low average Rand indices, and their stabilities do not diverge enough to decide on the number of clusters. Agglomerative clustering with single linkage, in which the closest pairs of points are combined at each step, seems to suggest that k=2 to k=5 are relatively stable.

Another way to measure the stability of a clustering is by adding noise to the distribution. Since trans contact probabilities are Gaussian distributed in the log space (Figure 5), random Gaussian noise can be added (5% was used) in log space and then converted back to normal space to create a matrix with noise. One clusterer is fit to the matrix with noise and one is fit to the original matrix. The labels of these two clusterers are then compared using the adjusted Rand index. The process is repeated 25 times and then the mean index is taken.

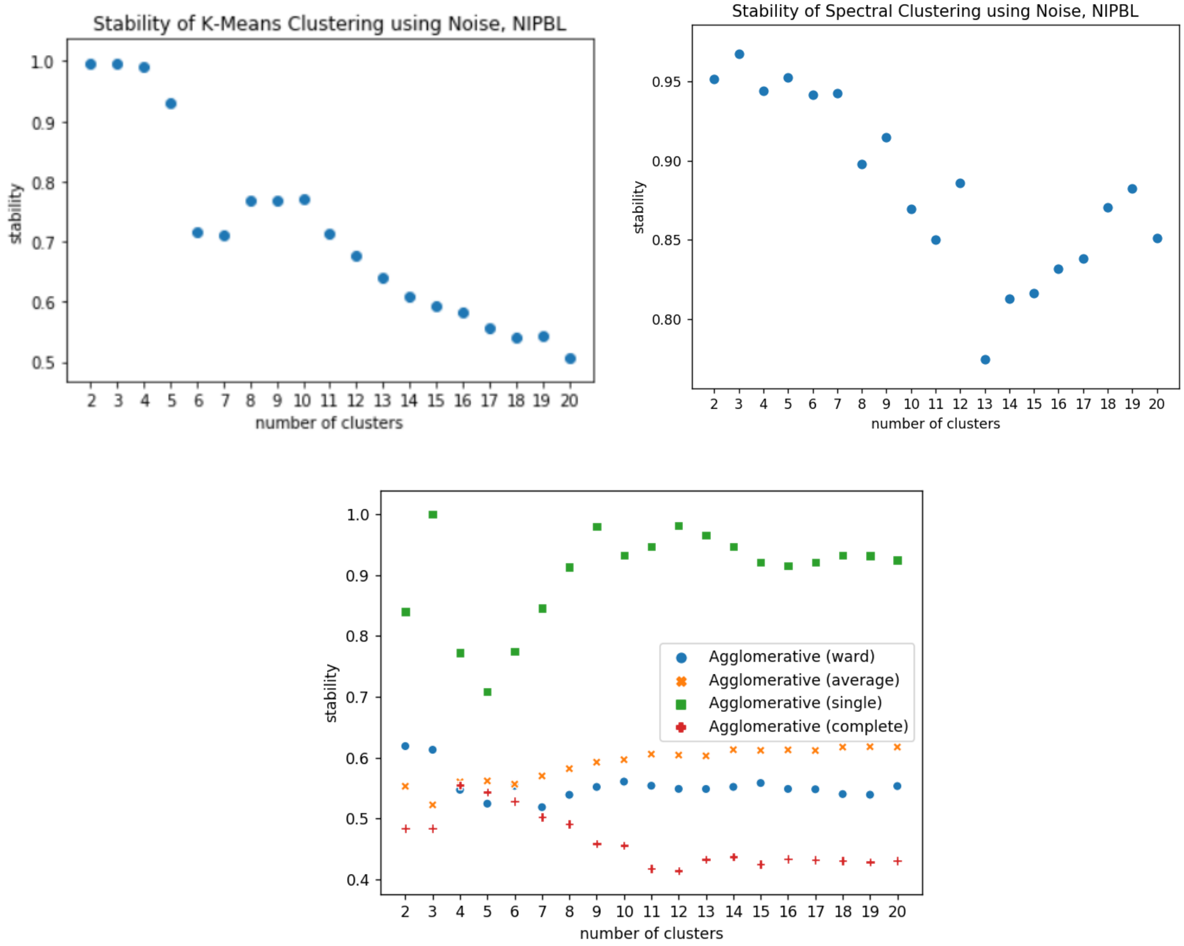


Figure 9. The results of stability analysis by adding noise.

The stability plots for noise suggest that there are between two and five major clusters that are stable for k-means and spectral clustering.

It seemed necessary to further analyze the results of agglomerative clustering (with average linkage) to determine why stability increases as the number of clusters increases.

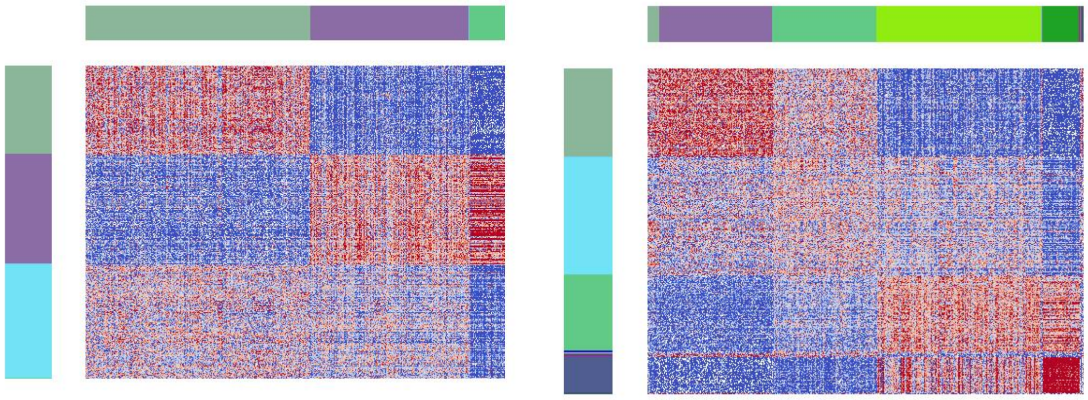


Figure 10. The results of agglomerative clustering with 4 and 20 clusters.

The clusterer does not necessarily determine the number of clusters even when enough clusters are specified, as shown in Figure 10. If agglomerative clustering is used, it must be with a larger number of clusters, such as 20. The clustering with 20 clusters can still be stable, as most of the labels belong to one of four clusters.

Visualization of k-means labels

As k-means was the most stable clustering algorithm, it was chosen to be visualized to better depict the clusters found. $k \leq 7$ clusters are worth analyzing as they have the greatest stability (. The plots below show the result when k-means clustering is applied to the odd-even matrix with differing numbers of clusters (from k=2 to k=7).

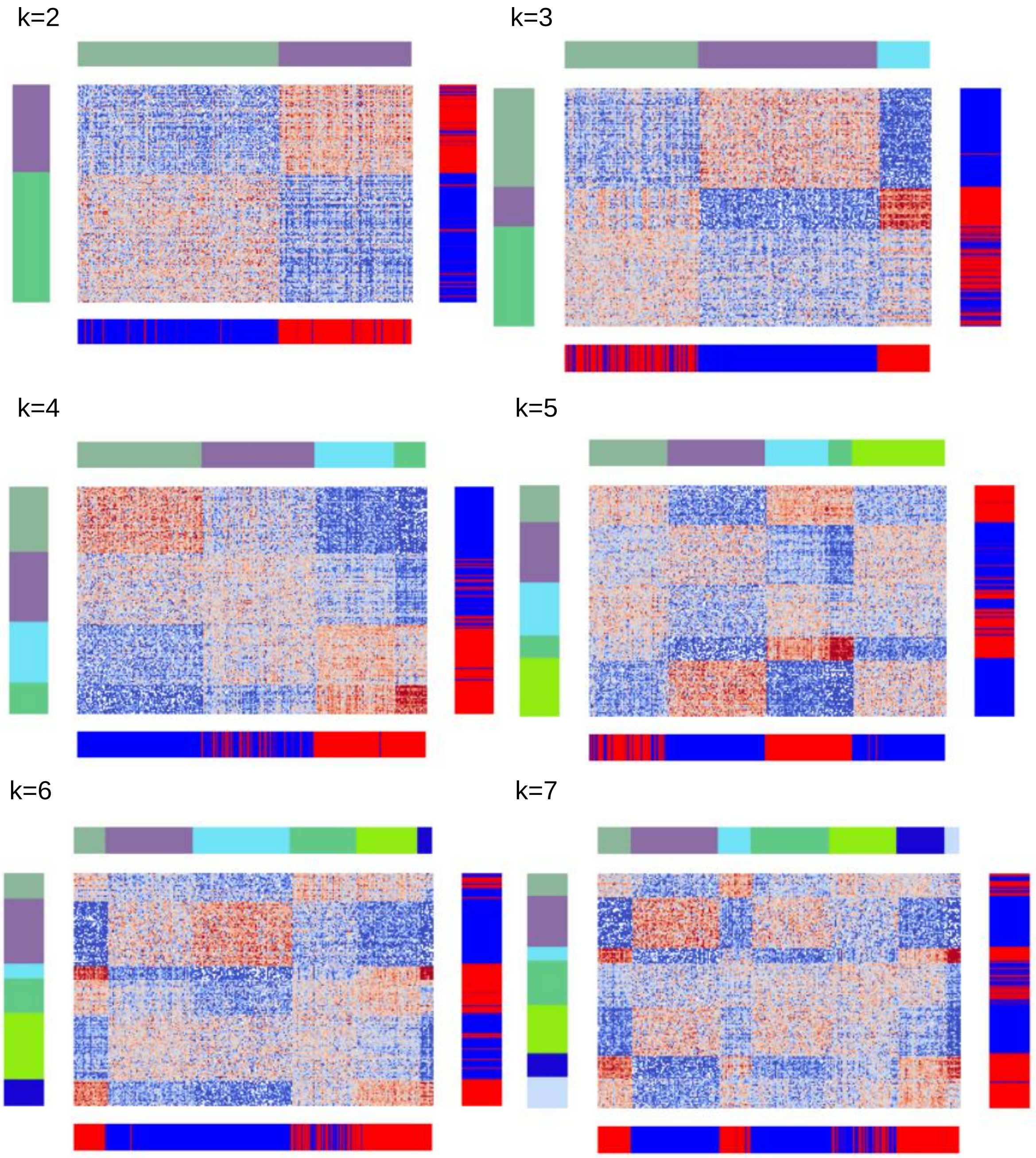


Figure 11. Results of k-means clustering for k=2 to k=7 clusters.

For $k \leq 4$, the clustering gets more discrete as the number of clusters increases. If k is at least 5, the clustering appears to perform worse. The second (purple) and fifth (grass green) clusters in k=5 appear to have the same interaction pattern, which implies that they are not very distinct. Additionally, when k=6, the 3rd and 6th row clusters and the 1st and 6th column clusters are indistinguishable. The separation of the eigenvector classes (red and blue) also appears the best when k=2 or k=4. This concurs with the results from the analysis of stability, which found that $k \leq 4$ was especially stable.

Comparison of spectral and k-means labels

The spectral clustering for k=6 and k=7 seemed unusually stable (Figure 8, Figure 9), so the clusters for k=7 were compared to the k-means labels for k=4.

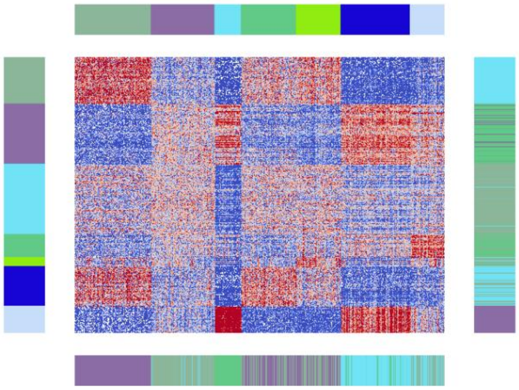


Figure 12. The 7 spectral labels (top, left) compared to the 4 k-means clusters (bottom, right)

Each of the seven spectral labels is part of one of the four k-means clusters. For example, the dark green, grass green, and lime green clusters are part of the purple k-means cluster. The spectral clusters differ only slightly from one another, but these sub-clusters do exist.

The three validation tools (stability, k-means visualization, and spectral clustering) suggest that there are four major clusters. The four k-means clusters seem to represent four compartments within the genome, and these compartments will be analyzed further for the remainder of the paper.
This analysis demonstrates that there are four large clusters, which are not initially separated when the clusterer attempts to find four clusters. This is a drawback of using hierarchical clustering on the Hi-C data. Combined with the stability analysis, it can be concluded that there are four intrinsic types of regions within DNA.

These clusters are best represented by the k-means plot for k=4 (Figure 11). The dark green, purple, light blue, and light green clusters can be named B1, B2, A2, and A1, respectively. B1 interacts with B1 strongly, and A1 interacts with A1 especially strongly. B2 interacts with B1, B2, and A2 about equally, but has a highly negative interaction with A1. Similarly, A2 interacts with B2, A2, and A2 strongly, but has a highly negative interaction with B1.

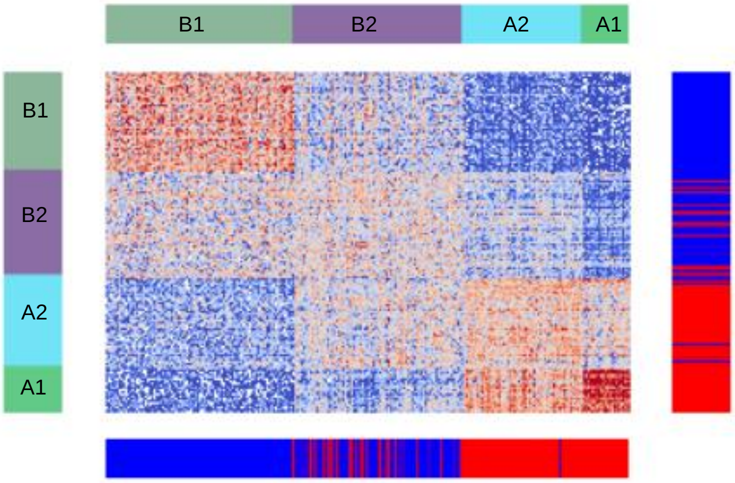


Figure 13. The four clusters generated from k-means.

Dimensionality reduction

As the odd-even matrix has thousands of dimensions, dimensionality reduction was considered for improving the quality of clusters. For this, principal component analysis (PCA) was used to reduce the odd-even matrix to 25 dimensions. A parallel coordinates plot was made to visualize the original k-means labels on the plot. In this plot, each zig-zag line represents a single row of the matrix, and the lines are colored by label.

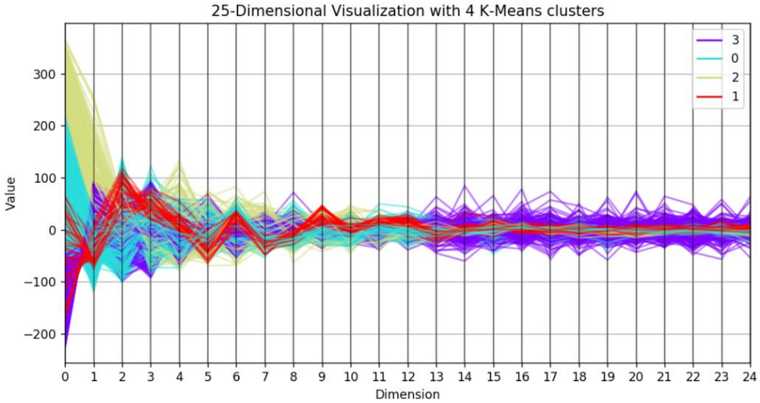


Figure 14. Parallel coordinates plot of PCA for 25 dimensions, annotated by k-means labels.

The dimensions past about 12 are likely noise, as there is minimal variance, so it is possible to do clustering on the 12-dimension reduction instead of the original matrix without compromising on quality.

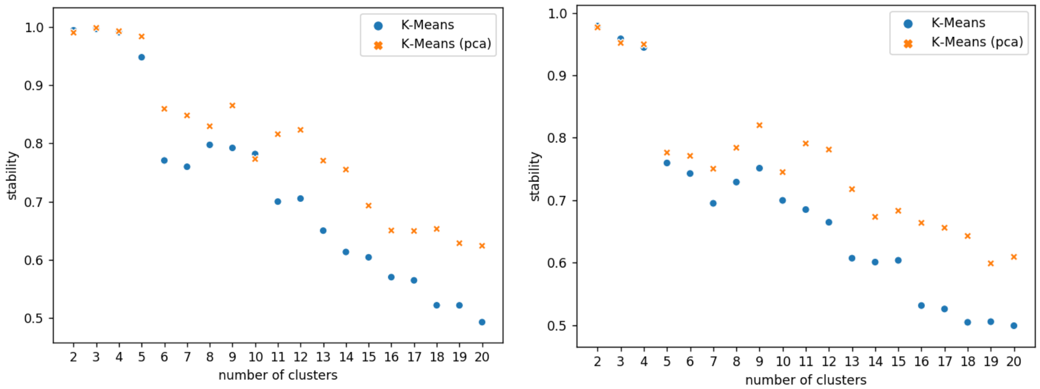


Figure 15. The stability using subsampling (left) and noise (right) for k-means with and without PCA.

Dimensionality reduction using PCA is slightly more stable than without, so it might be useful to use PCA, especially when finding larger numbers of clusters. The clustering results with and without PCA may be compared by computing the adjusted Rand index between the clustering labels for each number of clusters.

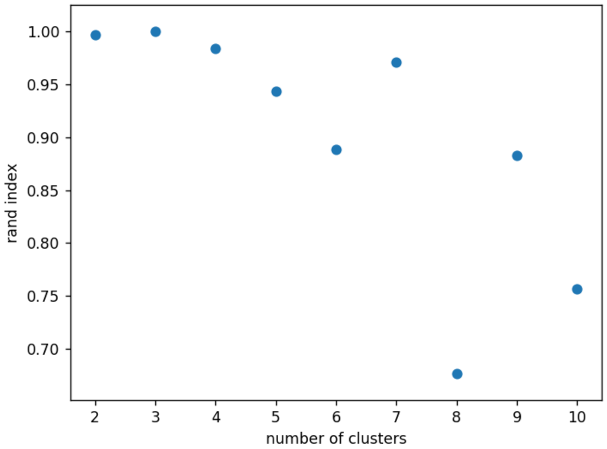


Figure 16. Adjusted Rand index between the labels of k-means with and without PCA.

It is found that the clustering results are very similar for 2, 3, and 4 clusters. When finding different numbers of clusters, PCA may be beneficial. Additionally, clustering on a lower dimensionality dataset can be much faster than the original dataset, which makes PCA useful for larger-scale analysis.

Analysis of ChromHMM

ChIP-seq data corresponds to proteins within the chromatin structure. ChromHMM [17], which is essentially a hidden Markov model algorithm running on ChIP-seq tracks, can help classify DNA regions as promoters, enhancers, quiescent, etc. ChromHMM runs separately from Hi-C. It classifies regions that are 200 base pairs long, so each of the regions for clustering (1,000,000 base pairs) contains 5,000 ChromHMM regions.

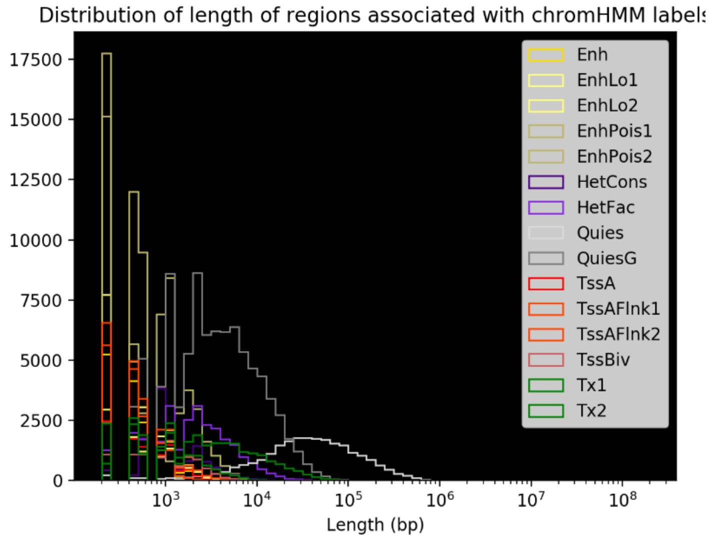


Figure 17. The ChromHMM states used for analysis plotted with the average length of consecutive states.

One way to assign a ChromHMM state is to take the most common string of consecutive ChromHMM states (Figure 17) out of all the states in a region. The results of this method can be plotted similarly to the eigenvectors. The B1 cluster correlated almost perfectly with the Quies state (white). The other states were mainly composed of the QuiesG (dark gray) state. This provides further evidence that the clustering represented a pattern fundamental to the structure of chromatin.

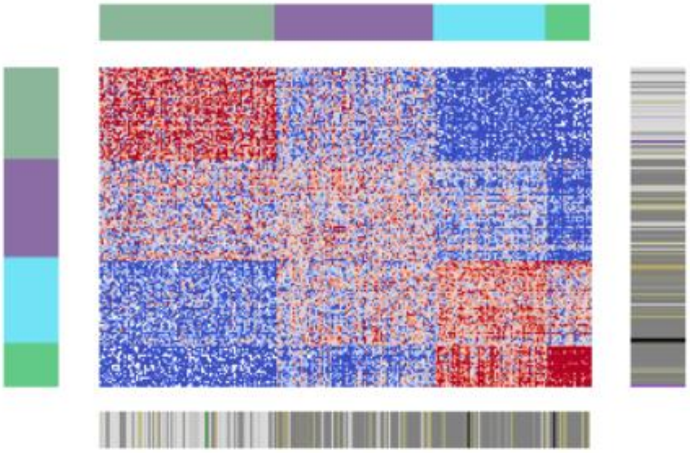


Figure 18. Comparison of clusters with ChromHMM states

Clusters can also be distinguished by comparing the average composition of regions by the states present. To compute the composition, the mean count of each of the 15 states within the regions for a cluster are computed, and then these counts are divided by 5000, the number of ChromHMM labels per clustering region. Figure 19 shows the composition visualized as percentages. The percentages may not add up to 100% due to independent rounding.

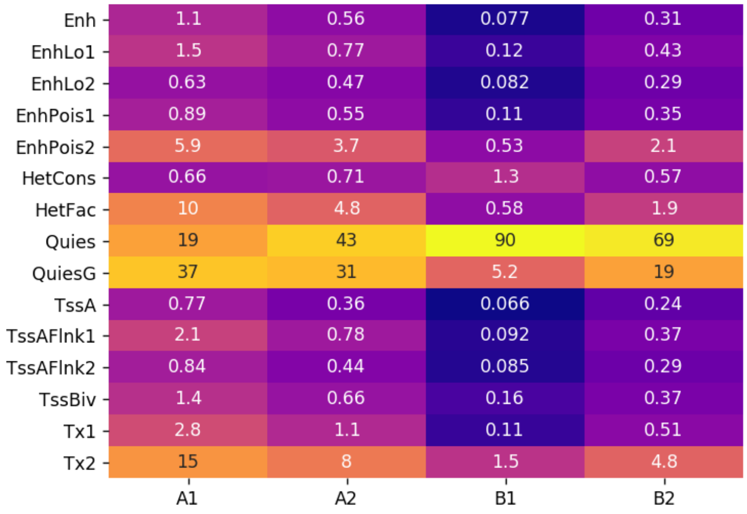


Figure 19. Composition of clusters by ChromHMM state

B1 regions are composed primarily of Quies and are deficient in every other state compared to the other regions. Furthermore, A1, A2, and B2 exhibit slightly different patterns in terms of their composition. For example, A1 and A2 are especially strong in enhancer states (Enh, EnhLo1, EnhLo2, EnhPois1, EnhPois2).

Chromatin signals

Histones are the primary component of chromatin, and they are known to control chromosome structure. Some proteins act as histone modifiers, and these modifiers manipulate DNA expression [18]. The profiles of histone modifications across the genome can be measured using ChIP-seq [19]. Rao et al. [2] suggested that the average expression of these histone modifications can be compared by cluster type.

Several different protein markers exist. The clustering of a GM12878 (human) cell from Rao et al. [2] was tested and compared it to the associated histone modifiers.

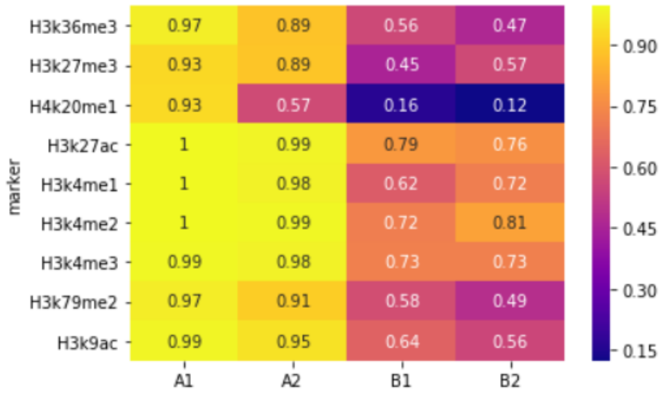


Figure 20. Histone Modifications for Clusters in GM12878 (human)

A1 and A2 have similar histone modifications, as do B1 and B2. However, there are slight differences. For example, the A2 region is more deficient in H4k20me1, which is involved in transcriptional repression. Additionally, there are small differences in the modifications between the type-B regions.

Repli-seq is a tool to analyze how different regions in the genome replicate at different times [20]. The cell cycle goes as follows: G1, S1, S2, S3, S4, G2, and the activity of genomic regions is measured at each of these times. Repli-seq can be visualized similarly to ChIP-seq.

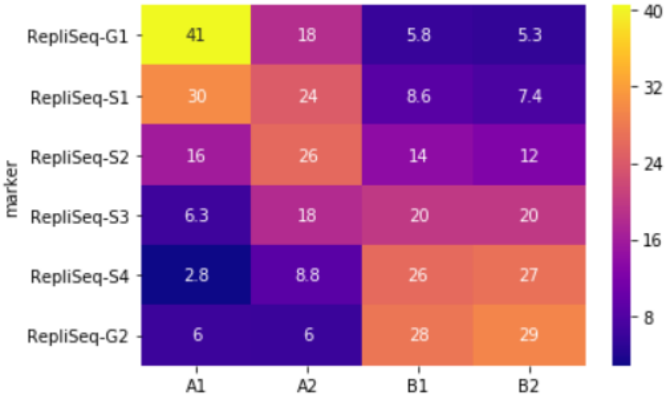


Figure 21. Repli-Seq for Clusters in GM12878 (human)

Both A1 and A2 replicate at an earlier time than B1 and B2, on average. However, A2 replicates on average later than A1. The difference between B1 and B2 is less observable, but B2 replicates slightly later than B1. This provides further evidence that there are structural differences between A1 and A2 regions.

Conclusion

It has been demonstrated that Hi-C matrices can be used to cluster chromosomal regions to reveal additional compartments domains past the consensus on an A/B split. Furthermore, the existence of these compartments has been justified through stability analysis, comparison to the A/B regions, and comparison with ChIP-seq and Repli-seq markers.

This process can be extended to any analysis of Hi-C data or other biological signals as these continue to be developed and refined. Additionally, the methods in this process can be used to aid in approaching any unsupervised clustering problem, even those not related to the field of genetics.

Future exploration

In the future, the clustering labels can be compared to computational models of chromatin structure to see whether they correlate with the 3-D structure of chromosomes. Additionally, the information about the types of clusters could be used to write a program optimized to find the A1, A2, B1, and B2 clusters from any Hi-C matrix, which other research groups could utilize.

Another line of exploration is a fine-grained analysis of the clusters to see if there are regions that seem to be a combination of multiple clusters and to identify smaller clusters that were missed by clustering. A major drawback in this exploration is that the low resolutions explored might have missed intricate detail in the Hi-C matrix.

Acknowledgments

This paper relies on datasets published by the ENCODE Project Consortium [21] [22] and created by the John Stamatoyannopoulos lab (UW) for Repli-seq and ChIP-seq (experiment accession numbers: ENCSR000DRW, ENCSR000CXJ).

I would like to thank my mentor, Sameer Abraham, for teaching me about the compartmentalization problem and about Hi-C matrices in general and assisting me throughout my exploration of finding clusters in the genome. Also, I would like to thank Professor Leonid Mirny at MIT for providing computational resources and assistance.

References

[1] I. . Krivega and A. . Dean, "Enhancer and promoter interactions — long distance calls," *Current Opinion in Genetics & Development,* vol. 22, no. 2, pp. 79-85, 2012.

[2] S. S. P. 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 and E. L. Aiden, "A 3D Map of the Human Genome at Kilobase Resolution Reveals Principles of Chromatin Looping," *Cell,* vol. 156, no. 7, pp. 1665-1680, 2014.

[3] N. van Berkum, E. Lieberman-Aiden, L. Williams, M. Imakaev, A. Gnirke, L. A. Mirney, J. Dekker and E. S. Lander, "Hi-C: a method to study the three-dimensional architecture of genomes," *J Vis Exp,* vol. 39, no. 1869, 2010.

[4] J. Zhou, J. Ma, Y. Chen, C. Cheng, B. Bao, J. Peng, T. J. Sejnowski, J. R. Dixon and J. R. Ecker, "Robust single-cell Hi-C clustering by convolution- and random-walk–based imputation," *PNAS,* vol. 116, no. 28, pp. 14011-14018, 2019.

[5] G. Fudenberg, M. Imakaev, C. Lu, A. Goloborodko, A. A and L. Mirny, "Formation of Chromosomal Domains by Loop Extrusion.," *Cell Reports,* vol. 15, no. 9, pp. 2038-2049, 2016.

[6] W. Schwarzer, N. Abdennur, Goloborodko, A. Pekowska, G. Fudenberg, Y. Loe-Mie, N. A. Fonseca, W. H. C. H. Huber, L. Mirny and F. Spitz, "Two independent modes of chromatin organization revealed by cohesin removal," *Nature,* no. 551, pp. 51-56, 2017.

[7] J. Haarhuis, R. van der Wilde, B. VA, T. Brummelkamp, E. de Wit and B. Rowland, "The Cohesin Release Factor WAPL Restricts Chromatin Loop Extension," *Cell,* vol. 169, no. 4, pp. 693-707.E14, 2017.

[8] E. Nora, A. Goloborodko, A. Valton, J. Gibcus, A. Uebersohn, N. Abdennur, J. Dekker, L. Mirny and B. Bruneau, "Targeted Degradation of CTCF Decouples Local Insulation of Chromosome Domains from Genomic Compartmentalization," *Cell,* vol. 169, no. 5, pp. 930-944.E22, 2017.

[9] 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, R. Sandstrom and B. Bernstein, "Comprehensive mapping of long range interactions reveals folding principles of the human genome," *Science,* vol. 326, no. 5950, pp. 289-293, 2009.

[10] R. Jager, M. G, M. Henrion, K. R, S. H. E, A. Heindl and N. Whiffin, "Capture Hi-C identifies the chromatin interactome of colorectal cancer risk loci," *Nature Communications,* vol. 6, p. 6178, 2015.

[11] J. Nuebler, G. Fudenberg, M. Imakaev, N. Abdennur and L. A. Mirny, "Chromatin organization by an interplay of loop extrusion and compartmental segregation," *PNAS,* vol. 115, no. 29, pp. E6697-E6706, 2018.

[12] 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," *Nat Methods,* vol. 10, pp. 999-1003, 2012.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort and V. Michel, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, 2011.

[14] Scikit-learn, "Clustering," [Online]. Available: https://scikit-learn.org/stable/modules/clustering.html. [Accessed 3 September 2019].

[15] U. v. Luxburg, "Clustering Stability: An Overview," *Foundations and Trends in Machine Learning,* vol. 2, no. 3, pp. 235-274, 2010.

[16] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association,* vol. 66, no. 336, pp. 846-850, 1971.

[17] J. Ernst and M. Kellis, "ChromHMM: automating chromatin-state discovery and characterization," *Nature Methods,* vol. 9, pp. 215-216, 2012.

[18] A. J. Bannister and T. Kouzarides, "Regulation of chromatin by histone modifications," *Cell Research,* vol. 21, pp. 381-395, 2011.

[19] P. J. Park, "ChIP–seq: advantages and challenges of a maturing technology," *Nature Reviews Genetics,* vol. 10, pp. 669-680, 2009.

[20] C. Marchal, T. Sasaki, D. Vera, W. Korey, S. Jiao, J. C. Rivera-Mulia, C. Trevilla-Garcia, C. Nogues, E. Nafie and D. M. Gilbert, "Genome-wide analysis of replication timing by next-generation sequencing with E/L Repli-seq," *Nature Protocols,* vol. 13, pp. 819-839, 2018.

[21] The ENCODE Project Consortium, "An integrated encyclopedia of DNA elements in the human genome," *Nature,* vol. 489, pp. 57-74, 2012.

[22] C. A. Davis, B. C. Hitz, C. A. Sloan, E. T. Chan, J. M. Davidson, I. Gabdank, J. A. Hilton, K. Jain, U. K. Baymuradov, A. K. Narayanan, K. C. Onate, K. Graham, S. R. Miyasato and T. R. Dreszer, "The Encyclopedia of DNA elements (ENCODE): data portal update," *Nucleic Acids Res.,* vol. 46, no. D1, p. D794–D801, 2018.

A method to recognize universal patterns in genome structure using Hi-C

Author
Neil Chowdhury (Phillips Exeter Academy, New Hampshire, USA)

Mentor
Sameer Abraham (Massachusetts Institute of Technology, Massachusetts, USA)

Biology Award