THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF BIOCHEMISTRY AND MOLECULAR BIOLOGY

VISUALIZING AND UNDERSTANDING CHROMATIN INTERACTIONS USING SELF-ORGANIZING MAPS

TIM KUNZ
SPRING 2017

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Biochemistry & Molecular Biology
with honors in Biochemistry & Molecular Biology

Reviewed and approved* by the following:

Shaun Mahony
Assistant Professor of Biochemistry & Molecular Biology
Thesis Supervisor

Lorraine Santy
Associate Professor of Biochemistry & Molecular Biology
Honors Adviser

Scott Selleck
Department Head for Biochemistry and Molecular Biology

* Signatures are on file in the Schreyer Honors College.

ABSTRACT

The Hi-C chromatin interaction assay is a relatively new procedure for collecting data on the 3D conformation of DNA in the nucleus. There are few tools and methods for analyzing and visualizing this data. Presented herein is a novel approach to chromatin interaction analysis based on the Self-Organizing Map (SOM) algorithm. This approach gives a more intuitive visualization of the data and serves as a platform for assessing correlations between various genomic activities and chromatin structure. The SOM algorithm provides a two-dimensional grid on which chromatin interactions indicated in Hi-C data are visualized. The resulting data structure can then be used to assess the relationships between genomic biochemical activities (e.g. transcription, histone modifications, protein-DNA binding, etc.) and the organization of the chromatin. Given a set of genomic coordinates corresponding to a given biochemical activity, the degree to which this activity is segregated or compartmentalized in chromatin interaction space can be intuitively visualized on the SOM grid and quantified using modified Lorenz curve analysis. We demonstrate the utility of our approach for exploratory analysis of genome compartmentalization using human high-resolution Hi-C datasets.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Gratitude and Appreciation to:

Dr. Mahony
For mentoring me and guiding my growth as a student and researcher

The Mahony Lab
For being an environment of helpful, supportive and insightful individuals

Dr. Santy and Dr. Howell
For advising me and helping me throughout my time at Penn State

Mrs. Bytheway
For convincing me that coding could be a useful skill for someone pursuing biology

Mom, Dad and Ryan
For your ongoing, unending love and support

Chapter 1

Introduction

1.1 Structures of the 3D Genome

The human genome is often thought of as an abstract string of nucleotides, with little consideration given to the complex, 3-dimensional organization of DNA in the nucleus. However, the advent of chromatin capture assays, such as 3C, Hi-C and Micro-C, has allowed for a more comprehensive understanding of the structure of the genome. Chromatin organization is a key driver of many genomic functions. Heterochromatin and euchromatin are key players in gene regulation. Chromosome segregation is associated with the replication cycle. At the end of mitosis, even after the decompression of the chromosomes, they tend to occupy particular regions of the nucleus (Meaburn & Misteli, 2007).

Topologically Associated Domains (TADs) characterize conserved chromatin compartments. TADs represent large domains constituting at least 6 nuclear subcompartments. These subcompartments are correlated with different patterns of histone modifications (Rao et al., 2015).

Lamina Associated Domains (LADs) anchor the DNA to the nuclear envelope. The nuclear envelope drives and maintains the organization of chromatin by this anchoring of peripheral DNA. LADs are associated with low levels of gene expression (Guelen et al., 2013).

Even enhancer-promoter looping involves chromatin structure. Chromatin organization and structure is a three-dimensional process, and so the key to fully understanding its related genomic processes lies in three-dimensional analysis.

1.2 Chromatin Capture

Chromatin capture was first explored in 2002 with the development of the 3C assay, on which other chromatin capture assays are based (Dekker, 2002). The protocol begins with the isolation of intact nuclei, which are then subjected to formaldehyde fixation. The crosslinked DNA is subsequently sequenced, producing hybrid reads aligning to regions of the genome that are physically touching in the nucleus (Dekker, 2002). The assay takes a group of nuclei, the 3D organization of which are not identical, so while two segments of DNA may be proximal enough for crosslinking in one nucleus, this may not be the case in another. The higher the frequency of crosslinked interactions, the more commonly two pieces of DNA occur together in the population of nuclei. The core assumption of the assay is therefore that interaction frequency is directly, positively related to 3-dimensional proximity in the nucleus (Dekker, 2002).

The 3C assay is described as a one-to-one capture procedure as it captures only interactions between two individual loci, for which the sequences of both must already be known (Stadhouders 2013). Several variants of the original assay exist. 4C expands the capture procedure such that all interactions with a known locus of interest are sequenced in a one-to-all capture. 5C explores many loci of interest and captures many of the interactions with each of these loci in a many-to-many capture. Hi-C analysis allows for all-to-all capture of interactions, using high-throughput sequencing technology and genome alignment (Stadhouders 2013).

1.3 Hi-C analysis

Hi-C analysis is a powerful chromatin capture variant in that it captures interactions between loci in a pairwise fashion across the entire genome (Lieberman-Aiden 2009). The fundamental

procedure is the same as that of 3C, with modifications to allow for capture of all pairs of interacting loci. The procedure begins with the isolation of intact nuclei and crosslinking through formaldehyde exposure. The DNA is then cut with restriction enzymes. The free ends are tagged, then ligated. The DNA is then sheared and pulled by its tag. The result is hybrid fragments of DNA that have been ligated together (Rao 2014). These reads have two portions, each aligning to a different portion of the genome. The reads are sequenced and aligned. The aligned sequences represent regions of the genome that were in close enough proximity to one another in the nucleus for an adjacent formaldehyde linkage to occur. This process is illustrated in Figure 1.

Figure 1: Hi-C Protocol

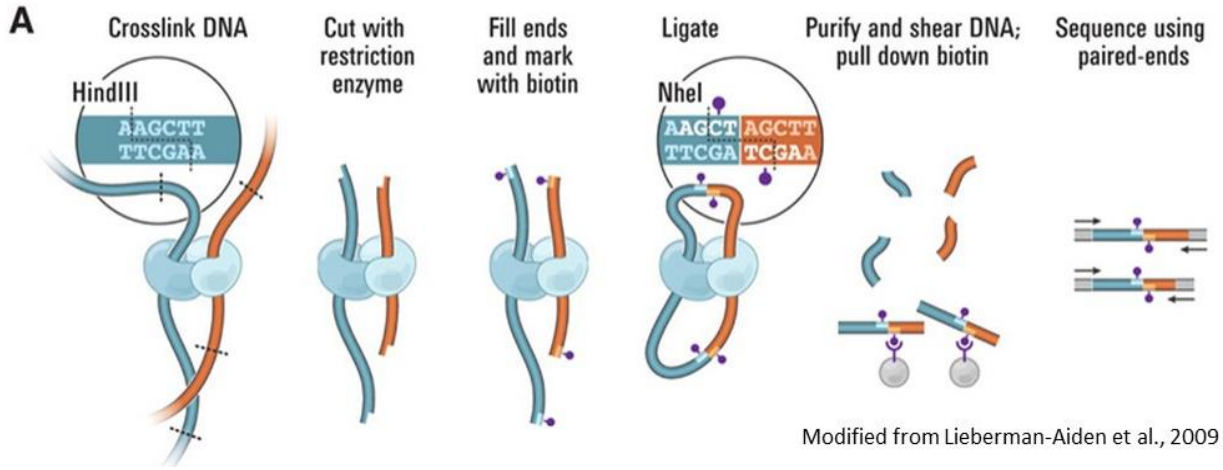


Figure 1: Hi-C Protocol
The procedure is as shown, proceeding left to right. The final paired end reads, once sequenced, are aligned the genome in two locations. That pair of locations then represent an interaction.

1.3.1 Hi-C data

The Hi-C procedure produces billions of reads, even after screening and aligning, each of which represent individual interactions. The data is processed by binning the genome into windows with

the bin size based on the resolution of the data set. Every interacting genomic locus falls into a certain bin. For any given bin, there may be zero or many interacting loci within its genomic window. The interaction frequency of a given bin represents the number of loci within that bin that interact with other loci. The interaction frequency between any pair of bins then represents the frequency with which loci within their respective genomic windows interact.

This data lends itself to representation as a matrix. The matrix has on its axes the bins from the entire genome, or genomic window of interest, and each index is the interaction count between the two bins. The matrices are symmetrical with respect to the diagonal, and sparse. The sparsity is simply due to few interactions occurring between pairs of bins that lie far apart on the genome. The matrices are then displayed as a heat map, with coloring based on the relative magnitudes of the interaction counts Figure 2A (Heinz et al., 2010).

It is important to note that the underlying data structure of the heat map is still a matrix with an interaction count for each pair of nodes. This means that each row or column can be taken as a vector, representing the interaction count between that column's bin and all other bins.

Figure 2: Hi-C Data and Normalization

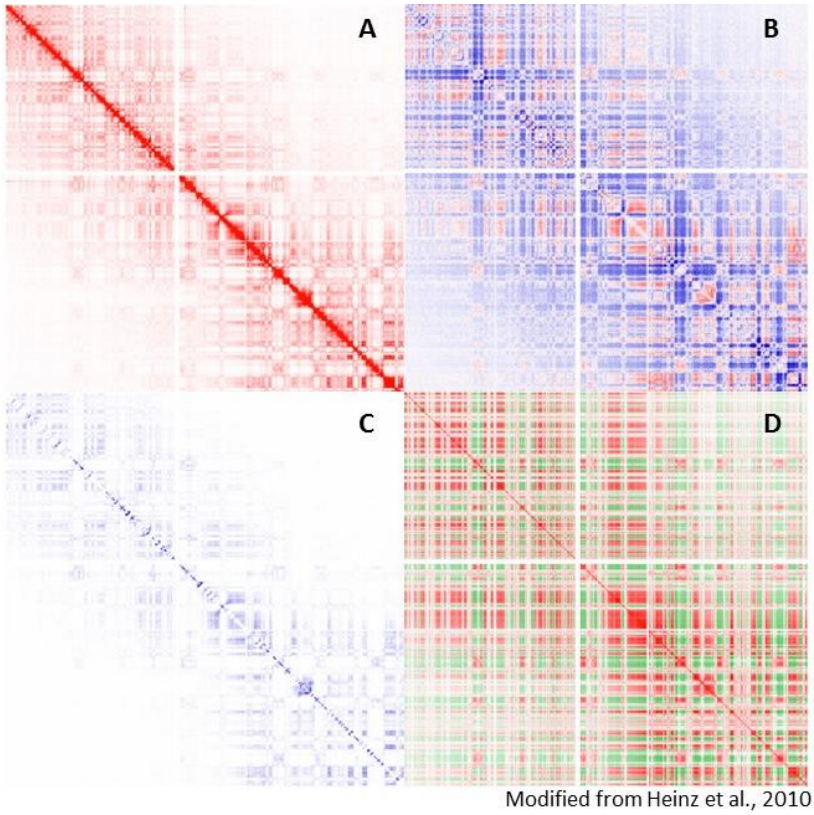


Figure 2: Hi-C Data and Normalization
The Hi-C interaction matrices are often visualized as heat map. **A** represents a raw interaction count matrix, note the pronounced diagonal signal. **B** represents an observed over expected interaction matrix, note that the diagonal signal is no longer present. **C** represents the log scale values of the raw interaction counts. While there is still a diagonal signal present, it is significantly less overpowering. **D** represents a matrix of the correlation of the normalized interaction ratios. All three of these are viable transformations/normalizations for Hi-C data, and there are several other methods. These heat maps depict interactions on chromosome 1 of human embryonic stem cells

1.3.2 Hi-C Data Normalization Methods

As seen in Figure 2, the diagonal of the interaction matrix is the dominant feature. This is because neighboring bins on the genome, whose interactions are represented on the diagonal, are in close 3D proximity due to the linear constraint. DNA that is close together on the linear genome is also close together in the 3D genome, producing an abundance of reads on the diagonal. Additionally, reads along the diagonal are produced by self-ligation effects, in which digested fragments of

crosslinked DNA are ligated to the other sheared end of the linear fragment, rather than ends of the adjacent crosslinked fragment.

This is only one of several factors that can impact Hi-C data. The dominant diagonal is of special interest because the goal of Hi-C assays is often to uncover long-range interactions. The long-range interactions are much more infrequent than the short-range interactions, so normalizing by the expected count can amplify the signal of significant long range interactions. There are several useful normalization and transformation methods, a few of which are illustrated in Figure 2 (Heinz et al., 2010).

In analyzing Hi-C data, data pre-processing can be necessary, and can help extract desired information, but it is important to understand what this preprocessing is doing to the data, as well as what the processing is forcing the data to describe.

Note that the various methods of normalization and the quality of the data are not considered by the tool described herein, it will produce results based on whatever data is input, and the output will reflect the topology of the dataspace. It is therefore necessary to be aware of how data processing is impacting the meaning of the data.

1.4 Dimensionality Reduction

Dimensionality reduction is the process of taking high-dimensional data and generating observations on a low dimensional (2D or 3D) manifold where the observations can be more intuitively understood (Roweis, 2000). It is crucial that the resultant observations on the produced manifold, or description space, are representative of the higher order data structure (Roweis, 2000).

If the goal is to understand the data, then dimensionality reduction is useless if it does not preserve the topology of the high-dimensional space.

The key component of dimensionality reduction algorithms is a function that maintains the relationships between similar data points (Lee, 2007). High-dimensional data vectors can be compared to each other, using Pearson correlation, cosine similarity, Euclidean distance, etc., and those high dimensional vectors most similar to one another should also be close together, at the end of dimensionality reduction, in the description space (Roweis, 2000)

1.5 Self-Organizing Map Algorithm

The Self-Organizing Map (SOM) algorithm is an unsupervised non-linear dimensionality reduction algorithm that also preserves topology (Lee, 2007). The SOM takes the form of a static grid or lattice of nodes. The grid is the 2D space upon which the high-dimensional data set is projected. Each node also has a weight vector of the same dimensionality as those high-dimensional data vectors in a data set, which represents the node's location in the high-dimensional dataspace (Kohonen, 2012). The basic schema of the occupation of a dataspace by a defined gird is shown in Figure 3.

Figure 3: SOM Training

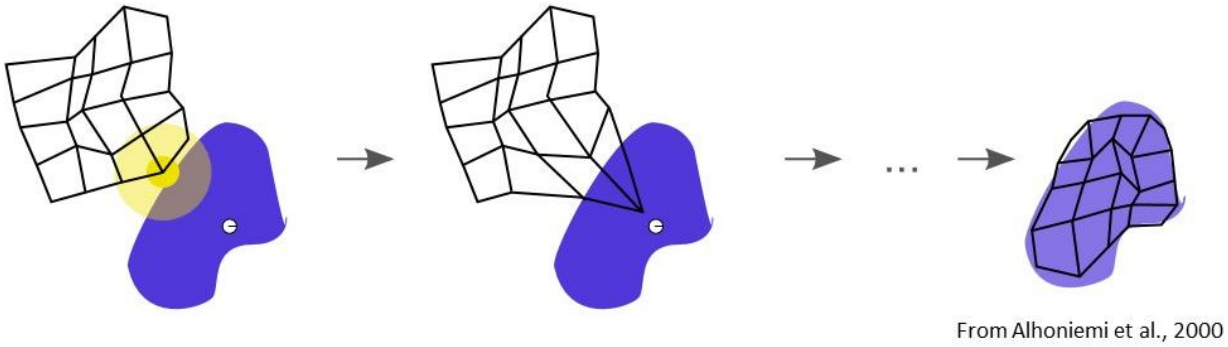


Figure 3: SOM Training
Depicted here is a web of nodes progressing to occupy the data space (blue), as in SOM training. The connectedness and relationship of neighboring nodes remains the same throughout training, but the position in data space of each node changes. The projection space would still be a rectangular grid, but it would represent the topology of the blue polygon on that space.

1.5.1 SOM Initialization

The basic SOM training proceeds as follows. Initializing the nodes begins by first defining the value of the weight vector for each node. This can be done by randomizing the values, or assigning them to be equal to data points in the dataset. Another popular method is to assign the node weight vectors such that they reflect the general structure of the data set from the start; this, however, requires extensive prior understanding of the data set, and is therefore not always possible (Kohonen, 2012).

The initialized nodes also must populate a structured lattice, such that the connections and distance between each are known. This is necessary because the grid onto which the dataset is to be projected aims to represent the topology of the dataset, meaning nodes close together on the 2D projection must also be close together in dataspace. The relationships between each node on the

lattice are used to impose an order to the nodes in dataspace. Before training begins, then, these relationships must be defined. Often, they are simply defined as the distance between the nodes on the projection space. The hexagonal grid is preferred to rectangular and square grids because in the hexagonal case each node has six neighbors with minimum distance, rather than four (see Figure 4**)**. This allows for a more elastic conformation to the dataspace, while maintaining regular structure.

Figure 4: Neighborhoods for SOM Grids

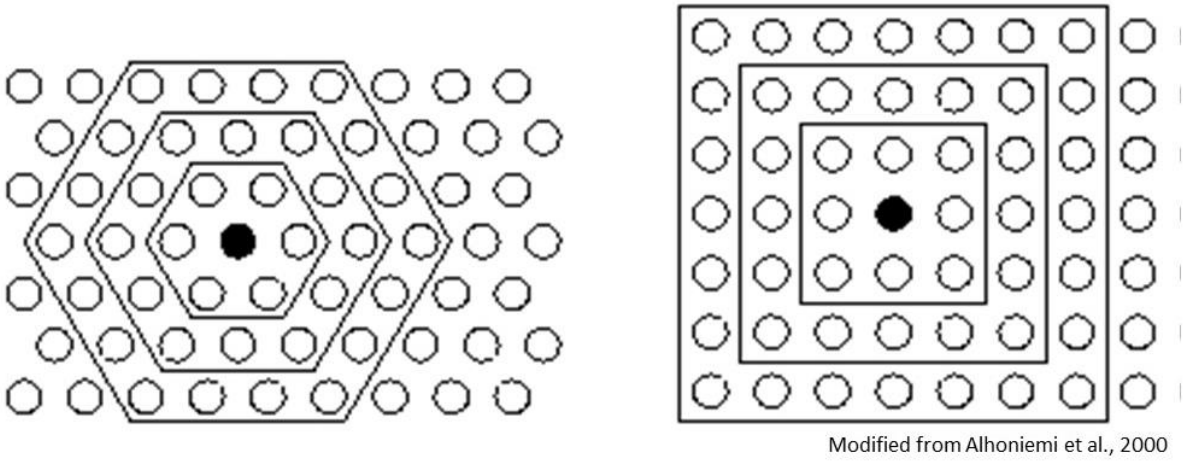


Figure 4: Neighborhoods for SOM Grids
Two neighborhood sides are represented here. The hexagonal grid shows that there are 6 direct neighbors for each node, and that the distance between nodes radiates out in shells. The rectangular grid shows similar shells, but there are only 4 direct neighbors for each node. Note that in the depicted shell there are 4 nodes which are closest to the node in black, and 4, on the corners, which are not of equal distance. This distance disparity, along with the reduced number of direct neighbors, makes the rectangular grid inferior to the hexagonal for SOM projection grids.

1.5.2 SOM Training

There are two main variants of training a Self-Organizing Map: the basic SOM, and the batched SOM. The batched SOM more readily lends itself to threaded training, meaning batched training can be completed faster (**2.5 Training**).

The batched training proceeds in two steps, repeated for a defined number of iterations. In the first step, the assignment step, the data points are assigned to the node whose weight vector is most similar (see 2.5.1.1 Similarity Metric) to the vector of the data point; this vector is then the best matching unit (BMU) (Kohonen, 2012).

The update step comes second, and the weight vector of each node is updated to reflect the assignment of the data points. Each data point is considered in the updating of each node. The weight vectors of each node are updated by:

$$W_v(s+1) = \frac{\sum_{j=1}^{n} h_{i,u,s} x_j}{\sum_{j=1}^{n} h_{i,u,s}}$$

Where:

- ***s*** is the current iteration
- $\boldsymbol{W_v}$ is the weight vector of node ***i***
- ***n*** is the number of data points in the set
- $\boldsymbol{x_j}$ is the data vector of data point ***j***
- $\boldsymbol{h_{i,u,s}}$ is the neighborhood function

The neighborhood function, denoted as $\boldsymbol{h_{i,u,s}}$ is defined as follows:

$$h_{i,u,s} = \alpha(s) * e^{\frac{-o^2}{2\sigma^2}}$$

Where:

- ***i*** is the is the node being updated
- ***u*** is the BMU (node) of the data point being considered
- ***s*** is the current iteration

- o is the distance on the SOM grid between nodes i and u
- σ is the variance of the Gaussian kernel for the current iteration
- $\alpha(s)$ is the learning rate for the current iteration

The learning rate and variance are parameters that can be changed for different training protocols, but, in general, both shrink over time from an initial value to a final value either linearly or exponentially. The gradual reduction of the variance allows for the neighborhood function to assign shrinking weights to further neighbors once the training has proceeded for several iterations. Data points from highly distant nodes have little impact on the weight vector once the map has been allowed to adopt the general structure of the data.

It is necessary to shrink the learning rate over time, because the Self-Organizing Map has no defined end-point in training. Therefore, it must be restricted such that training slows and stops as it reaches the defined number of iterations. This produces the trained SOM (Figure 5), which reflects the structure of the data set.

Figure 5: Trained SOM

Figure 5: Trained SOM
The panels above represent a 2D data space (left) and trained SOM (right). Data points in the data space are colored according to the node to which they are assigned. The two starred nodes are connected to their neighbors by black lines in the data space. The neighboring nodes have assigned data points which are close together in the data space. This is more pronounced in the case of purple node, but indeed the green node is neighboring nodes with relatively similar data points.

1.6 Lorenz Curve Analysis & Gini Coefficient

Assessing resource inequality is an important topic in the fields of economics, sociology and politics, and the Gini coefficient is relatively common way to express resource disparity quantitatively (Catalano et al. 2009). The description of the distribution of resources begins by constructing the Lorenz curve, which plots the cumulative percentage of the resource occupied against the cumulative percentage of the population occupying that resource (Harch 1997). It is important to note that the population is summed from poorest to richest in terms of resource

occupation, producing an increasing trend in cumulative percent resource occupation per percent population.

The Lorenz curve is compared to a specific case of distribution, often simply the case in which the resource is equally distributed (Catalano et al. 2009). The area under the comparison case is maximized in this situation of equal distribution, because, as shown in Figure 6, any unequal distribution of a resource will produce a curve below the line of equal distribution.

The area between the Lorenz and comparison curves is known as the Gini index. A greater area between these curves, which produces a large index, indicates a larger gap between the distribution of the resource and the ideal or expected distribution described by the comparison case. In a common example, the Lorenz curve of income distribution is compared to the case of completely equal distribution. The larger the Gini index, the less equally distributed the income. The Gini index is normalized by dividing the index by the area under the comparison case. This yields the Gini coefficient, which, like the Gini index, grows with the resource disparity. An example of these curves, including the calculation of the coefficients, is shown in Figure 6.

Figure 6: Gini Coefficient

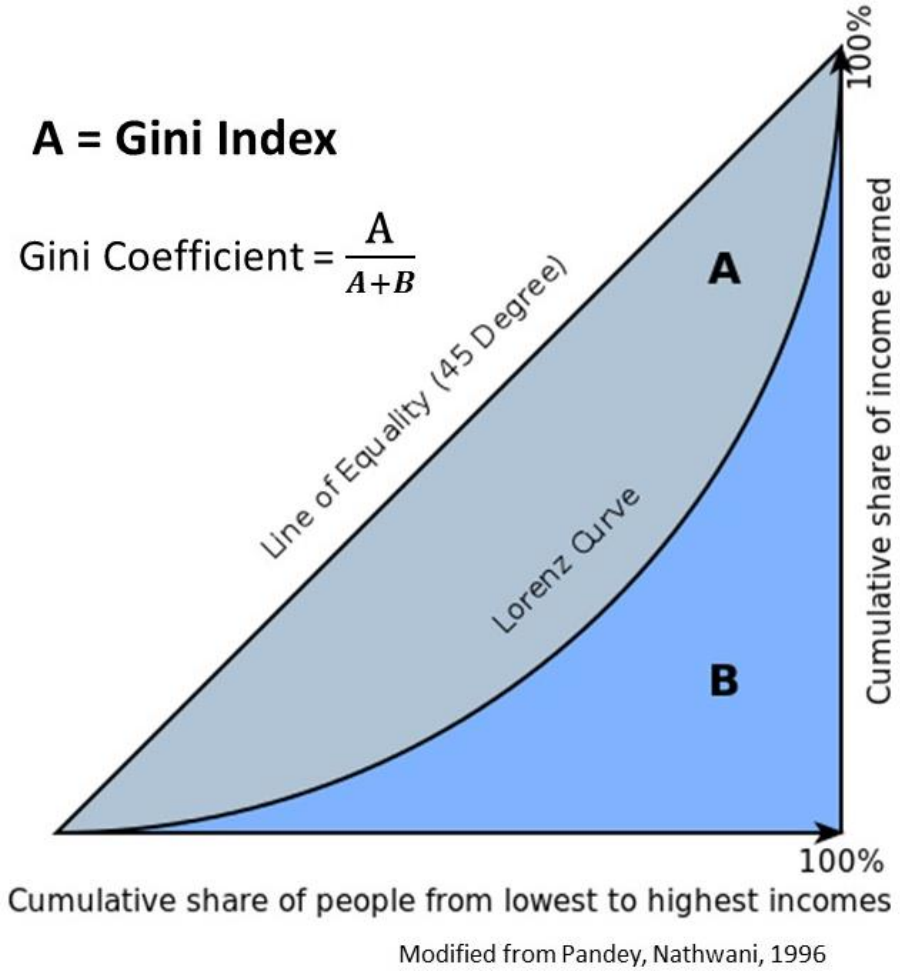
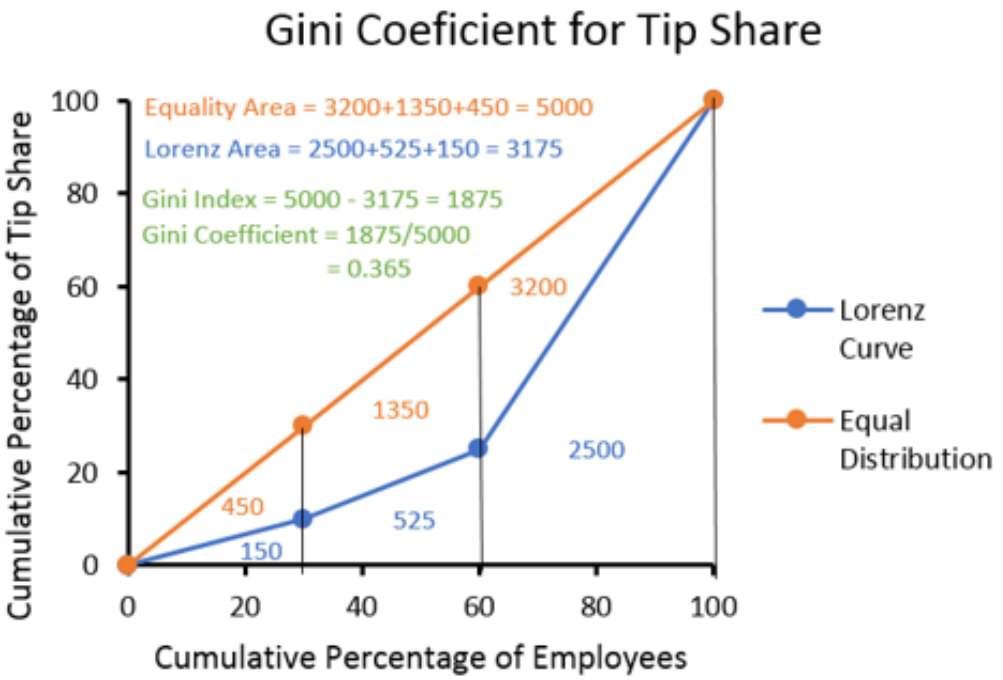


Figure 6: Gini Coefficient
On the graph above, the line of equality represents the comparison curve in the perfect equality case. The Lorenz curve marks the true, unequal distribution of wealth. The grey area marked as A, between the equality comparison and the Lorenz curve, represents the Gini index. The larger the Gini index, the greater the disparity between the equality case and the actual distribution. The Gini coefficient is produced by dividing the Gini index by the total area below the equality comparison. A larger Gini coefficient comes from a large Gini index, which indicates a more unequal distribution of wealth.

In cases of cumulative wealth distribution across populations, individuals represent such a small fraction of the total population that the Lorenz curve is effectively continuous, however integration is not required for area under curve calculations; the areas are found geometrically using finite intervals (Figure 7).

Though most commonly the comparison case of interest is that of equality, if another distribution of a resource is expected, the comparison curve can take another form. The Gini coefficient then still represents the degree of difference between the expected or ideal distribution and the observed distribution. In cases of wealth distribution disparity, the United Nations suggest Gini coefficients larger than 0.40 are considered highly unequally distributed (Catalano et al. 2009). However, this cutoff is arbitrary, and so cannot be considered absolute.

Figure 7: Gini Calculation



Position	Percent of Employees	Percent of Tip Share	Cumulative Employee	Cumulative Tip Share
None	0	0	0	0
Busboy	30	10	30	10
Hostess	30	15	60	25
Waiter	40	75	100	100

Figure 7: Gini Calculation
The table above includes sample data for distribution of employee population and tip share among the three front of house restaurant positions. The Lorenz curve based on this data and the linear comparison curve are plotted on the graph above the table. The area under each segment of the curve was calculated geometrically, and is labeled on the graph. The totals of these areas were used to calculate the Gini Coefficient, as shown in green.

Chapter 2

Methods: Training a Chromatin Interaction SOM

2.1 Overview

The Self-Organizing Map implementation described here takes an input matrix of Hi-C chromatin interaction data, raw or normalized, and outputs a trained SOM grid, which is a 2D representation of the chromatin interaction space as depicted by the high dimensional input data. The output grid can then take additional data sets, in the form of genomic coordinates and weights, and provide visual and quantitative analysis about the distribution of those coordinates in chromatin interaction space. The implementation is written in Java.

2.2 Input Data

The input data must be in the form of a tab-delineated DxD matrix, where D is the number of genomic bins in the data set (i.e. related to the resolution of the Hi-C experiments). The matrix must begin with a row of loci, indicating the coordinates of the genomic loci covered by each bin. Every row after that must start with the region covered by the bin, followed by Hi-C interaction counts with every other bin. The interaction count does not have to strictly be a count. Each index takes the form of a double, and can reflect any transformation or normalization performed on the matrix prior to training. A portion of a small sample matrix is shown in Figure 8. Bins with no data, meaning zero interactions with all other bins, are not considered in training, and trained maps will not reflect the genomic loci contained in those bins.

Figure 8: Data Set

	chr1:0-499999	chr1:500000-999999	chr1:1000000-1499999	chr1:1500000-1999999	chr1:2000000-2499999	chr1:2500000-2999999	chr1:3000000-3499999
chr1:0-499999	0.010473129	0.004442765	0.003349629	0.004997421	0.005487543	0.004029231	0.00437567
chr1:500000-999999	0.004442765	0.921265936	1.402032226	0.976721437	0.7415618	0.345715183	0.491566131
chr1:1000000-1499999	0.003349629	1.402032226	1.872505583	1.493553194	0.886890471	0.419783392	0.594765192
chr1:1500000-1999999	0.004997421	0.976721437	1.493553194	2.056700943	1.48304736	0.800104622	1.284059649
chr1:2000000-2499999	0.005487543	0.7415618	0.886890471	1.48304736	2.029409767	1.509390346	2.106155516
chr1:2500000-2999999	0.004029231	0.345715183	0.419783392	0.800104622	1.509390346	1.200779624	1.876728643
chr1:3000000-3499999	0.00437567	0.491566131	0.594765192	1.284059649	2.106155516	1.876728643	1.63331023
chr1:3500000-3999999	0.005289881	0.609676754	0.800090073	1.442379462	1.909292585	1.002604935	1.465224031
chr1:4000000-4499999	0.003113762	0.364120124	0.449964781	0.764402602	1.089066143	1.64109764	1.527126424
chr1:4500000-4999999	0.002413496	0.278162765	0.394977762	0.60675964	0.949747209	1.804979391	1.498115863
chr1:5000000-5499999	0.001383812	0.294446484	0.390258207	0.584531822	0.960459671	2.119375288	1.431826495
chr1:5500000-5999999	0.001538142	0.509242707	0.655264101	1.293249273	1.979297044	2.828141689	3.29475941
chr1:6000000-6499999	0.016844723	1.198212385	1.671644538	3.226843842	3.997032864	2.284346834	3.813396826
chr1:6500000-6999999	0.020116697	1.329048617	1.874994733	3.602061864	3.371653588	1.760313617	2.541787696
chr1:7000000-7499999	0.005894779	0.535835668	0.746221213	1.52891285	1.904124904	2.339884147	2.667456365
chr1:7500000-7999999	0.00835128	0.780075804	1.104589569	2.080323247	2.37027636	2.218250161	2.618642908
chr1:8000000-8499999	0.008938111	1.014680562	1.42653661	2.249412533	2.161177916	1.238204069	1.573393907
chr1:8500000-8999999	0.016537788	1.240162928	1.607621467	2.601562632	2.227467045	1.101644861	1.508917008
chr1:9000000-9499999	0.017244812	1.134019758	1.544058708	2.540877046	2.1535594	1.366106636	1.819230046
chr1:9500000-9999999	0.025306034	1.502762177	2.206613447	2.808801442	2.16506943	0.992287862	1.228947335

Figure 8: Data Set
Shown is only a small portion of an input data set. First row and first column both indicate the genomic location of the bin in question. The indices represent the interaction frequencies. The frequencies are fractional numbers, because this matrix has been normalized.

2.3 Calling the Training Program

The program takes as parameters:

- call for training (String [“Train”])
- x-dimension of the grid (int)
- y-dimension of the grid (int)
- initial variance of the neighborhood kernal (double)
- final variance of the neighborhood kernal (double)
- number of iterations (int)
- number of maps for quality control (int) (see 2.6 Quality Control)
- desired similarity metric (int)
    - 0 for Pearson Correlation

    - 1 for Cosine Similarity
- name of the output map (String)
- path to the input matrix
- number of available processors

Example of the command line call for training:

- `java -Xmx38G MultiMap Train 50 50 1.2 0.2 1000 10 1 "My Map" "My Matrix" 40`

MultiMap is the package in which the code for training resides. The -Xmx38G argument is required to handle memory load. This call will produce a 50x50 map (2500 total nodes) named "My Map" based on the data in the "My Matrix" file. Training will proceed for 1000 iterations, with a kernel variance shrinking from 1.2 to 0.2. Ten maps will be trained for quality control, and training will occupy 40 processors (or the maximum available). Note that learning rate is not taken as a parameter, and ranges from 1 to 0.01 over the course of training in all cases.

2.4 Initialization

The first step in training is to load the data set into memory. The regions covered by each bin are saved in a vector of magnitude D, while interaction counts are saved in a separate DxD matrix. The entire data set is loaded into memory. An additional D-dimensional vector is then created which stores the magnitude of each data vector. This is used in comparing data points to nodes. Unlike the nodes, however, the data point vectors do not change during training, so it is prudent to calculate and save these magnitudes for use throughout training.

Each node on the grid is then given a weight vector equal to that of a random data point. The weight vector of each node is then stored in a NxD matrix, where N is the number of nodes on the grid. Then the neighborhood is calculated. The neighborhood is represented as an NxN matrix in which the pairwise distance between nodes is saved. The grid is static, so these distances do not change throughout training. These neighborhoods are illustrated in Figure 9. Note that the grid is toroidal, so sides and corners are defined to be adjacent.

An additional NxD matrix of Boolean values is created, but not yet filled, to hold the value (true or false) corresponding to whether a data point is assigned to a node. An empty N-dimensional vector is also created to store the number of assigned data points for each node.

Figure 9: Neighborhoods

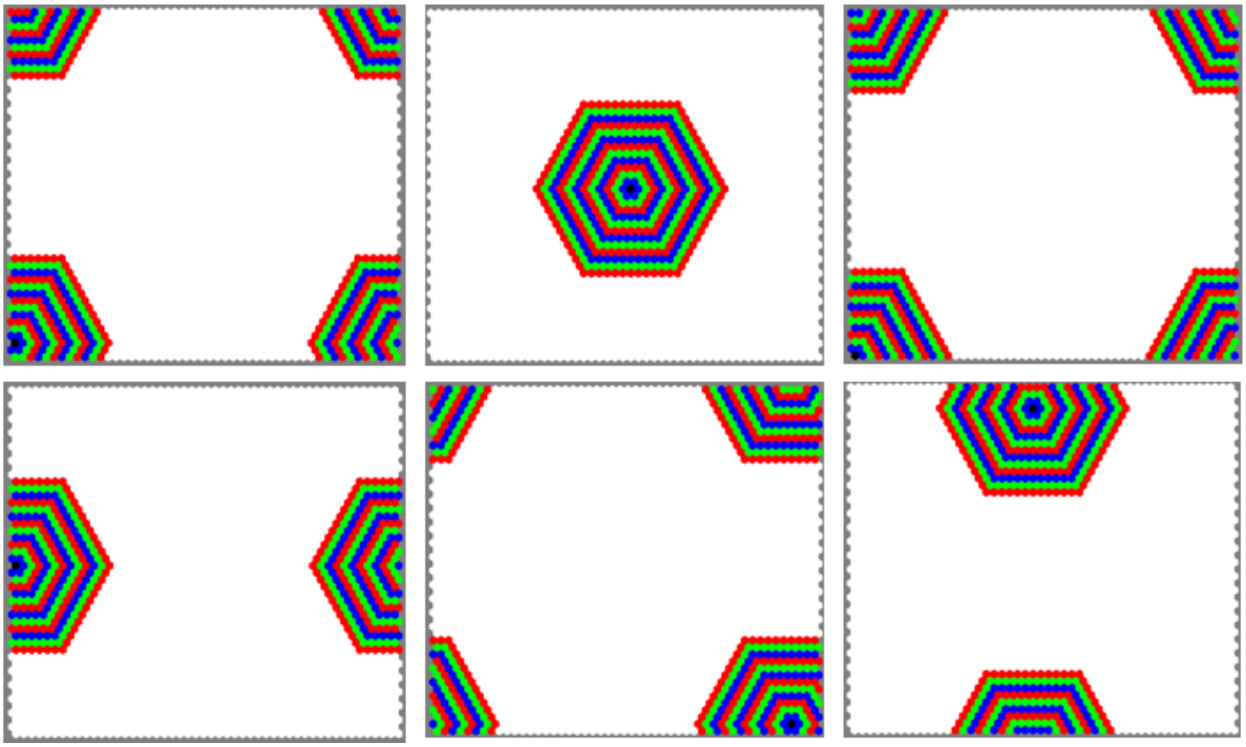


Figure 9: Neighborhoods
On each grid above a different node is indicated in black. For each black node, the neighborhood is indicated by the radiating colors. Each shell represents an increment of distance. Note that the grid is toroidal, so that when radiations reach an edge, they continue on the opposite edge.

2.5 Training

The training protocol is sped up by splitting the jobs required among available processors via multi-threading, in which each thread has a processor with which it computes a fraction of the task at hand. In general, more processors mean faster training. The batched implementation of the training protocol was selected because it lends itself to multi-threading by breaking training into two repeated steps: assignment and update.

2.5.1 Assignment

Assigning data points to nodes proceeds by first calculating and storing the magnitude of each node's weight vector. Then the data points are split among available threads. Each data point is compared to every node. The most similar node is determined. The assignment matrix is then updated to reflect each data point's new BMU, and the vector counting the number of assigned data points for each node is incremented and decremented to reflect the same.

2.5.1.1 Similarity Metric

The BMU of each data point is determined to be the node with the weight vector of the highest similarity to the data vector. There are two similarity metrics available for training: cosine similarity and Pearson correlation. Euclidean distance is not used because with high-dimensional vectors, even sparse vectors, as is the case with Hi-C data, can produce numbers too large to be easily handled by Java. The cosine similarity and Pearson correlation are both invariant to scaling, but only Pearson correlation is invariant to shifts (O'Connor, 2012). However, in Hi-C data, zero

is always the minimum number of interactions, and thus shifts are of no concern. Cosine similarity is used by default, but either can be used effectively, and indeed there is no noticeable difference between maps trained with alternative similarity metrics (4.1 Parameter Optimization**)**. Both scores range from -1 to 1, with 1 reflecting identical vectors.

2.5.2 Update

The update step begins by calculating the neighborhood term for each distance value. This is done all at once because the weights are the same for all equivalent pairwise distances.

$$h_{i,u,s} = \alpha(s) * e^{\frac{-o^2}{2\sigma^2}}$$

Then, node updates are split among all available threads. Each node is updated as follows:

$$W_v(s+1) = \frac{\sum_{j=1}^{n} d_u h_{i,u,s} x_j}{\sum_{j=1}^{n} d_u h_{i,u,s}}$$

The implementation here is different from the basic batch implementation of the update in its inclusion of the $\boldsymbol{d_u}$ term, which reflects the number of data points assigned to the node $\boldsymbol{u}$, which is the BMU of the considered data point. Increasing the weight on nodes with many assigned data points increases their influence on the map. Nodes with many data points, a high data density, are better able to more effectively recruit adjacent nodes to occupy some of the density. This makes for smoother maps with a better spread of data points.

2.6 Quality Control

It can be hard to assess the degree of success of a training run. In general, there are a few values that can describe this success. The first is the *quality*. Two quality scores are generated at the end of training, the cosine similarity quality, and Pearson correlation quality. Both values are the average of the similarity scores between each data point and their assigned nodes. Holding the size of the dataset and the size of the grid constant, a higher quality score indicates a more representative map (Mortazavi et al., 2013).

The second metric is *stability*. The calculation of this metric is the reason for training several maps. Once all maps in the set are trained, the one with the highest quality score is taken as the best map. Then each pair of data points assigned to the same node in the best map is taken. The stability metric is then calculated as the proportion of these pairs that are found on the other trained maps in the same or directly neighboring nodes. Again, with the grid and data set sizes held equal, the higher the value the better. This metric gives an insight into how consistent and reproducible the resulting map is. No two maps will ever produce the exact same output because of the randomized initialization. However, a high stability score indicates that training will reliably produce the same trends if performed again. If the number of maps parameter is set to 1, this quality control metric is set to zero for output, as it cannot be calculated (Mortazavi et al., 2013).

The final quality control metric is the proportion of the nodes left empty at the end of training. High data point density can make it hard to interpret the trained SOM, so in general the more spread out the data points are among the nodes, the better. Empty nodes indicate that data points may not be spread especially well. This metric is perhaps least important, because empty nodes can be reflective of the topology of the data set, but all else equal, the smaller this value the better.

2.7 Terminating Training

Convergence is imposed by shrinking the learning rate and kernel variance until the specified number of iterations is reached. At the end of training of all maps, the quality control metrics are calculated. Only the SOM with the highest quality score is output. A text file is generated in the directory from which the program was called, with the indicated name of the SOM, as well as the training parameters and the quality control metrics. In the text file, each node lists its position on the grid, and the genomic bins associated with it. The genomic coordinates of the bins are parsed with commas, and the SOM node grid locations are separated by tabs. All high dimensional data vectors are lost, leaving only the set of genomic loci, and their positions on the grid. An example output is shown in Figure 10. The trained map can then be used for further analyses as described in the next chapter.

It is prudent to discuss, specifically, what the trained map represents. The goal of the map is to reveal structure of the chromatin in the nucleus, which is a 3-dimensional process. However, the topology preserved by the dimensionality reduction is that of the high-dimensional Hi-C input data. The input Hi-C data does not paint a perfect picture of the chromatin organization, which is why data manipulations and normalizations are necessary to make inferences from that data. The information about structure of chromatin in the Hi-C data is preserved through training, but any artifact or other erroneous structures will also be preserved. It is important to remember that, while the organization of the chromatin is of interest, it is the topology of the Hi-C data that is preserved and represented on the output, not the topology of the nucleus itself. The trained SOM therefore only represents the 3D organization of the chromatin in so far as the Hi-C input data does.

Figure 10: Trained SOM Output

A

B

50x50			
Sigma:1.0->0.3			
[0,0]	()	[1,0]	()
[0,1]	(chr9:44500000-44749999, chr9:44750000-44999999,	[1,1]	()
[0,2]	(chr9:25000000-25249999, chr9:31000000-31249999,	[1,2]	()
[0,3]	()	[1,3]	(chr12:42500000-42749999, chr12:42750000-42999999, chr12:45500000-45749999, chr12:46000000-46249999,
[0,4]	()	[1,4]	(chr12:58000000-58249999, chr12:64750000-64999999, chr12:65000000-65249999, chr12:68500000-68749999,
[0,5]	()	[1,5]	()
[0,6]	()	[1,6]	(chr23:152000000-152249999, chr23:152250000-152499999, chr23:152500000-152749999)
[0,7]	()	[1,7]	()
[0,8]	(chr23:152750000-152999999)	[1,8]	(chr23:153000000-153249999)
[0,9]	()	[1,9]	()
[0,10]	(chr23:39250000-39499999)	[1,10]	(chr23:39500000-39749999)
[0,11]	()	[1,11]	(chr23:39750000-39999999)
[0,12]	()	[1,12]	(chr23:40000000-40249999)
[0,13]	()	[1,13]	()
[0,14]	(chr6:38000000-38249999, chr6:38250000-38499999,	[1,14]	()
[0,15]	()	[1,15]	(chr23:41250000-41499999)
[0,16]	(chr23:44500000-44749999, chr23:44750000-449999	[1,16]	()
[0,17]	()	[1,17]	(chr13:19000000-19249999, chr13:32000000-32249999, chr13:33750000-33999999, chr13:34000000-34249999,

Figure 10: Trained SOM Output
A shows the output file naming system. The parameters in the output file name are, from left to right ('X nodes' x 'Y nodes'), 'initial Kernel variance'- 'final Kernel variance', 'iterations', 'number of maps', 'cosine quality', 'Pearson quality', 'stability', 'Percent unoccupied' **B** shows the structure of the output SOM file itself. The output file contains only the coordinate of each node, and which bins are assigned to that node. All interaction frequency and weight vectors are lost, as they are no longer needed

Chapter 3

Methods: Using a Trained Chromatin Interaction SOM

3.1 Viewing

The trained map can be used by invoking the "view" argument in the BatchMap package as follows:

- `java -Xmx38G BatchMap View`
- `java -Xmx38G BatchMap View 0 1`

This will present an empty hexagonal grid with one text field and four buttons. The first invocation specifies no columns for searched data files, while the second one does specify columns. These parameters are discussed further below. The text field is to specify illustration of the whole map, a random sample of the bins, or a single chromosome. Entering 0 into the text field yields a view of the whole map, entering -1 yields a randomized set of bins, and entering any other number presents only bins from that chromosome. Sex chromosomes, assumed to take the form of either X or Y, are referred to numerically as 1 and 2 more than the highest numbered autosome. For example, in a human set, the X chromosome would be referred to as chromosome 23, and the Y chromosome would be number 24. Mitochondrial loci are not included in the training of the map nor the output of the map under the assumption that nuclear DNA cannot physically interact with mitochondrial DNA in cells. If there is no chromosome of the referred to number, the hexagonal grid will be empty. The *chr* button colors the map according to the number entered into the text field. The grid is a heatmap such that the more data points of the indicated criteria, the darker the shade of red. These options are illustrated in Figure 11.

In all cases of projection on the map, the coloring of the nodes is a heat mapping based on the total weight on the node, where darker red indicates a higher weight density. Each node's weight is the sum of the weights of the bins in that node. The weight of a bin can be one of two things, in cases of a weighted dataset being projected, each indicated locus in the data set has both an assigned bin and a weight. That bin's weight is then the sum of assigned locus weights. Alternatively, without weights, the heat mapping is simply a count of loci assigned to the bin, and the node's weight is then the sum of those counts. In the case of chromosome mapping, each bin is considered its own locus, and all bins are weighted equally, so the coloration is a representation of the distribution of bins from the specified chromosome.

Figure 11: Functionality of the Chrome Button

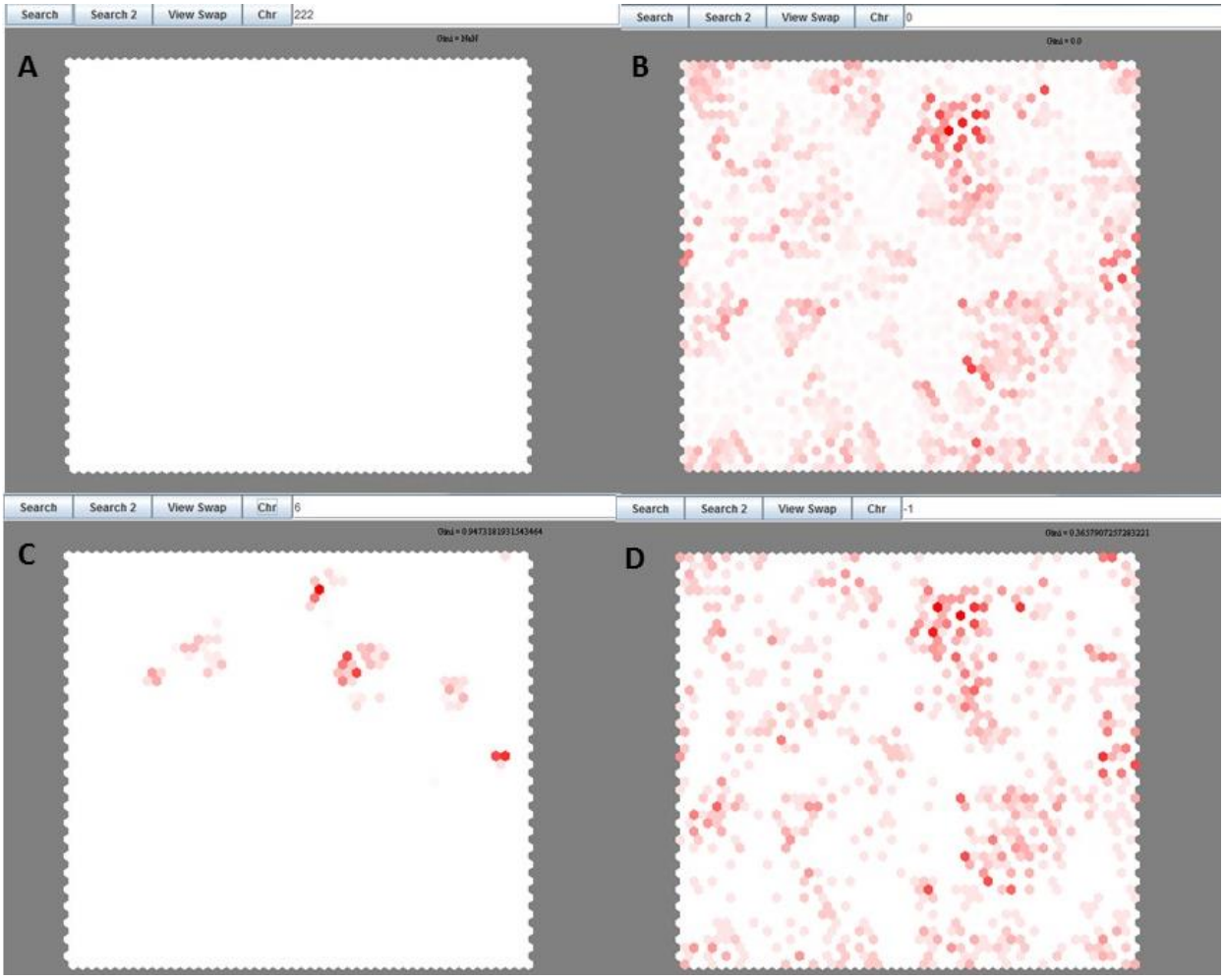


Figure 11: Chrome Button
A shows the empty grid displayed upon opening the SOM file, or upon enter an out of bounds chromosome number. **B** shows the full map, representing all bins. **C** shows only the bins from chromosome 6 of this example K562 SOM. **D** shows a random set of the bins. Note that the structure and density of the random set mirrors that of the full map in **B**.

There are four different options for viewing selected data. The basic one is a simple heat map ranging from white to red. The second is a heat map from white to red, but setting nodes with no qualifying bins, or nodes with weights of less than 1% of the maximally weighted nodes, to match the color of the background. The third and fourth viewing options are the same as the first and second, respectively, except that the edges of the hexagons are painted, and the counts of genomic

bins in each node are displayed. These viewing options are toggled through using the "View Swap" button. These views are illustrated in Figure 12.

Figure 12: Functionality of the View Swap Button

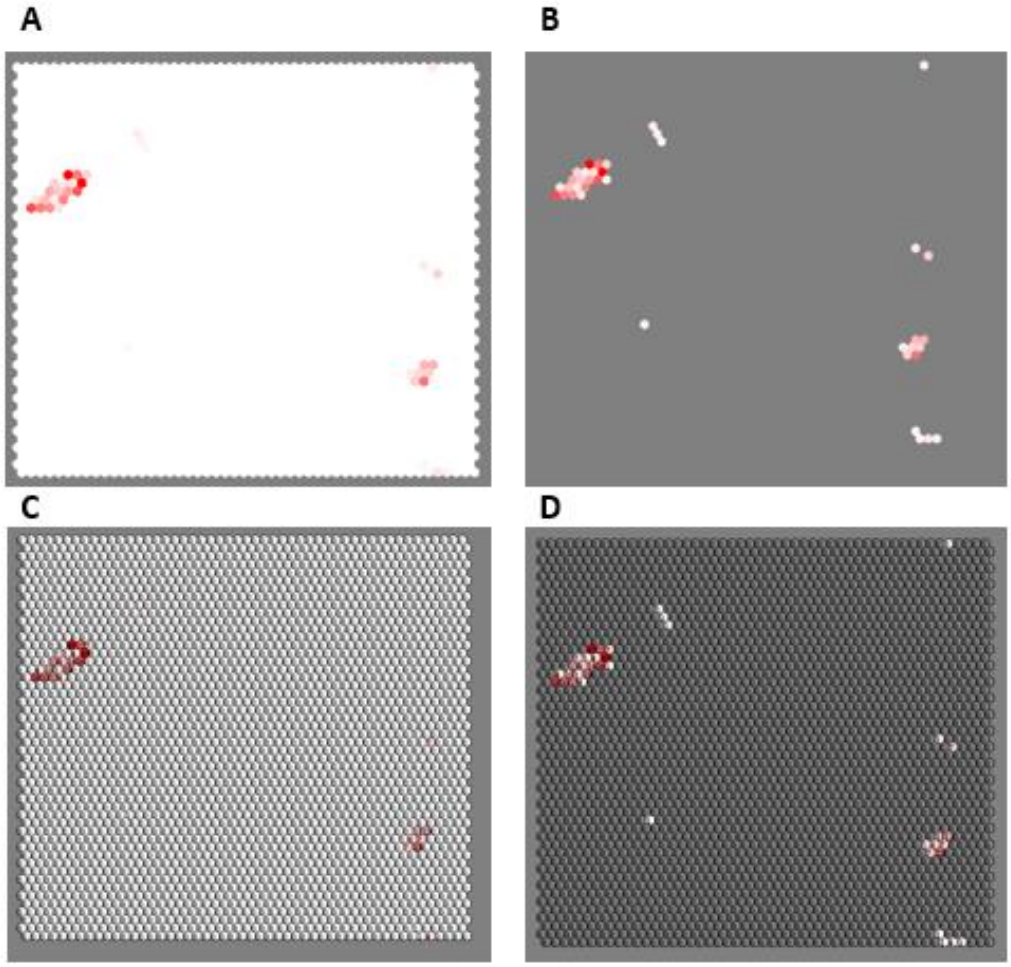


Figure 12: View Swap Button
Each panel shows the display of chromosome 13 from the K562 sample SOM. **A** shows the default view. **B** shows the same view, but with all nodes containing no or few indicated bins colored the same as the background. **C** and **D** show the same views as **A** and **B**, respectively, but the lines of the grid are shown, and the count of bins in each node is are displayed.

The "search" button opens a dialog where the user can select a set of genomic coordinates to view on the map. If no integer parameters were included in the *View* invocation, each search data file is assumed to contain loci in the first column, and weights in the second column. If integer parameters are used, the first integer refers to the column containing the loci, and the second integer refers to

the weights. If the second integer is -1, then the loci are assumed to be equally weighted. Each line in search files indicates an instance of a genomic activity. The accepted format is tab-delimited, where the locus column describes the location in the form "chr#:locus" and the weight column indicates the weight or count of that activity as a double or integer. If signals in the experiment cover a range, rather than a single position, then the location can be described as "chr#:start locus-end locus" and each bin contained in that range will be given an equal share of the weight from that range. A sample projection of a genomic activity is shown in Figure 13**.**

Figure 13: Functionality of the Search Button

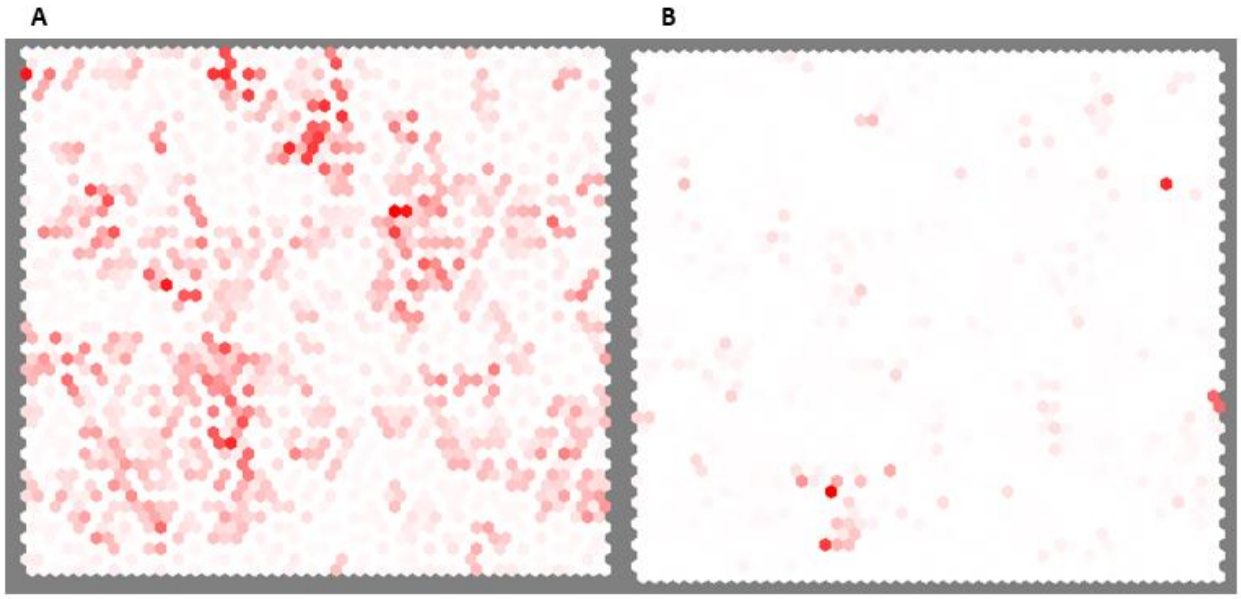


Figure 13: Search Button
A shows the default view of the full map of a K562 sample SOM. **B** is displaying the relative weight of bins assigned to each node, as indicated by a sample ChIP-Seq file, weighted c-Jun in this case. The relatively unequal distribution of the weight on the SOM suggests that there may be an unequal distribution of this transcription factor in the nucleus.

The "search 2" button opens a dialog that allows for the selection of two files. Both are then displayed on the grid, one colored in blue, the other in red. This enables an assessment of whether two activities are co-localized in chromatin interaction space. The same file restrictions for the "search" button hold true for "search 2" button. A view of the "search 2" function is illustrated in

Figure 14. These views generate a basic Gini coefficient, and display it in the top right corner (see 3.2.1).

Figure 14: Functionality of the Search 2 Button

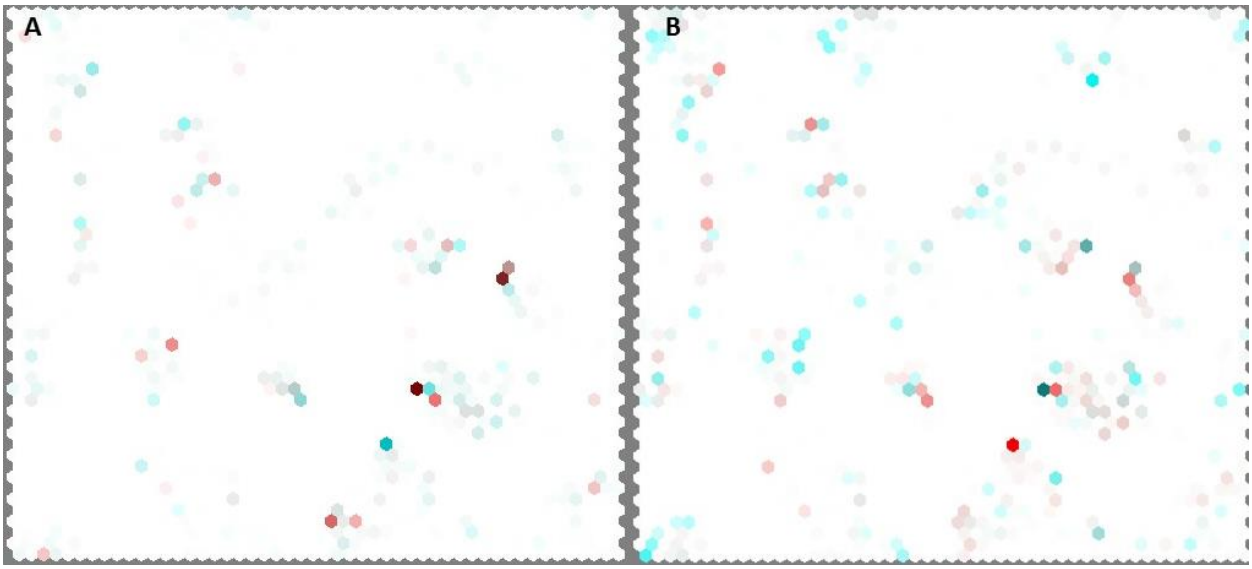


Figure 14: Search 2 Button
A shows the projection of two separate ChIP-Seq experiments for the same protein, Pol2. The high degree of greyish purple on the map indicates a large degree of overlap. There is a lot of red, without a lot of blue, which suggests also that the experiment projected in red had higher coverage. **B** shows two separate projected ChIP-Seq experiments, Pol2 and p300, with only a small amount of overlap, yielding distinct areas of blue and red.

3.2 Batched Data Projection

The Use function of the tool is meant to expedite the discovery of potential 3D structures or chromatin segregations in the nucleus. The function is meant to be invoked from the command line, without needing to open any GUI. Given a directory of files and a map, the Use function will search each file in the directory and project that data set onto the map. It then calculates Gini coefficient value for each data set, and saves a text file tabulating these values for each file in the directory. Additionally, if desired, a PNG-format image will be saved in a newly created directory

for each file in the directory. The image will be the same in appearance as if the View function were being used, and the Search button were invoked on the respective data set (Figure 13)

The Use function can be used by invoking the “Use” argument in the BatchMap package as follows:

- `java -Xmx38G BatchMap Use “My Map” “My Data Directory” 0 1 1`

The directory containing the data sets must have datasets in the same form as those in the Search button function, containing the location and weight of signals specified by the first and second integer arguments, respectively. Again, if the second integer is set to -1, all indicated loci receive equal weights. The third integer argument specifies whether images for each dataset are desired: a value of 1 will generate images, a value of 0 will not.

3.2.1 Gini Coefficient Calculations

For each data set, two Gini coefficients are calculated, a basic and normalized coefficient. Both calculations start from the mapping of the data set on to the trained SOM. Each signal in the data set has a genomic locus and a weight. Each node is given a weight equal to the sum of the weights of the loci contained in the bins assigned to that node. The Lorenz curve for both calculated Gini coefficients are constructed with the nodes as the population, and the weight as the wealth.

For the basic Gini coefficient calculation, the Lorenz curve is compared to a curve as if each bin contains equal weight. The bins are not evenly distributed amongst the nodes on the map due to the structure of the genome indicated by the Hi-C input data. It is therefore not desirable to calculate Gini coefficients based on the comparison curve of the diagonal. A more reasonable

comparison is to distribution of bins among nodes; this way differences in distribution due to the distribution of bins on the map is not considered, leaving only differences in distribution attributable to location in the data space.

There is, however, one issue with the basic Gini coefficient when working with weighted data; in a given weighted data set, since the weight of each signal differs, it is expected that there will be an unequal distribution of the weight simply due to these differences. Even if the loci are approximately evenly distributed across the bins, one especially strong signal can enrich a single node, inflating the Gini coefficient. The normalized Gini coefficient guards against this by comparing to an area which includes an expected distribution given the set of weights. The comparison curve is constructed by taking the weights of each signal from the data set and assigning that weight to a random bin, then using the same protocol as the Lorenz curve construction. A thousand of these randomized areas are calculated and averaged, the standard deviation is also calculated to be used later. This average represents the expected degree of unequal distribution attributable to both the unequal distribution of bins among nodes and the unequal distribution of weights among signals.

Another way to avoid this issue is to invoke the -1 option for the weight column. This causes the program to treat all indicated loci as having equal weights, removing any issues caused by the distribution of weights among loci. This is a reasonable approach if, for the dataset at hand, there is an expectation that all loci have contributed a significant signal. For example, if a dataset is filtered to remove all loci that are not statistically significant, it may be apt to treat all remaining data points as equivalent.

The Gini coefficients are then calculated as the difference between the area under the Lorenz curve and the area under the comparison curves, divided by the area under the comparison curves. A third value is also calculated as the difference between the Lorenz area and mean area as calculated for the normalized Gini coefficient, divided by the calculated standard deviation. Though the distribution of weight among random bins is not explicitly normal, this calculation yields a Z-score-like value that represents the likelihood that the unequal distribution of weight could have arisen strictly by chance. Higher values of the Z-score and the two Gini coefficients indicate higher degrees of unequal distribution of signal, which in turn suggests localization or clustering of some kind for the assessed genomic activity. A sample graph of the curves used to calculate the Gini coefficients is shown in Figure 15.

Figure 15: Sample Gini Coefficient Calculation Plot

Figure 15: Gini Coefficient Calculation Plot
This graph depicts the curves used to calculate the Gini coefficients for a projected genomic activity dataset (c-Jun ChIP-Seq used in this example). The blue curve plots the distribution of weight among the nodes. The orange curve plots the expected unequal distribution due only to the distribution of the bins, and the green curve plots the unequal distribution expected from both the unequal distribution of bins among the nodes and the unequal distribution of weights among the bins. The area between the orange and blue curves is used to calculate the basic Gini coefficient, and the area between the green and blue curves is used to calculate the normalized Gini Coefficient.

3.2.2 Batch Use Output

After Z-values and Gini coefficients have been calculated for each dataset in the directory, the program will output a text file. This text file will be tab delineated, with each line containing the file name of the dataset, the basic Gini coefficient, the normalized Gini coefficient and the Z-value for each file.

If the argument to produce pictures is invoked, the file will be saved in a newly created directory named “Lorenz Analysis,” along with the generated PNG images. Otherwise, if pictures are not desired, the text file will be saved in the directory from which the program is invoked.

Chapter 4

Results

4.1 Parameter Optimization

To determine which parameters are best for training the Self-Organizing Map on Hi-C data sets, several sets of parameters were tested on various test sets. The parameters tested included grid size, learning rate, Gaussian Kernel Variance, similarity metric, and number of iterations. The best set of parameters was originally defined as that which produced, in order of importance, the highest quality scores, highest stability score, and had the lowest percentage of empty nodes. Additionally, maps were checked by eye to assure that each combination produced no aberrant results not apparent in the summary statistics.

In general, the implementation shows a robust preservation of results across parameter settings. The similarity metrics each produced almost identical quality scores with other parameters held constant, as shown in Table 1. Number of iterations, though more is always better, tends to show no significant improvement in terms of output after 1000 iterations (Kohonen, 2012). In order to ensure training time is not too long, 1000 iterations has been set as the default. However, with larger data sets, in the interest of time, fewer may be successfully employed.

Table 1: Parameter Optimization

Similarity metric	X	Y	Cos	Pears	Stability	Percent
Pearson	12	12	0.6281	0.6071	0.9068	0.5833
Cosine	12	12	0.6294	0.6091	0.885	0.618
Pearson	14	14	0.6748	0.6575	0.9212	0.6224
Cosine	14	14	0.6764	0.659	0.9039	0.6275
Pearson	16	16	0.7183	0.7037	0.9471	0.625
Cosine	16	16	0.722	0.7087	0.9384	0.621
Pearson	18	18	0.7552	0.7438	0.9604	0.5956
Cosine	18	18	0.7509	0.7384	0.9547	0.6358
Pearson	20	20	0.7811	0.7713	0.9574	0.5975
Cosine	20	20	0.7804	0.7699	0.9573	0.61
Pearson	50	50	0.8701	0.7444	0.8236	0.4536
Cosine	50	50	0.8719	0.7472	0.8348	0.436

Table 1: Parameter Optimization
The similarity metric column indicates, for the tested map, which similarity metric was used to train. The X and Y columns indicate the dimensions of the trained map. Cos and Pears columns indicate the cosine quality and Pearson quality, respectively, for the trained map. Stability indicates the stability metric calculated for the trained map. Percent indicates the percent unoccupied quality metric.

The data used to train the maps sized 12x12 to 20x20 were from a Yeast data set, with only 3000 bins (Hsieh et al., 2015). This data set was used for the smaller maps to ensure the trained maps did not produce erroneous metrics due to overpopulation of the nodes; too small a map may not be

able to adequately capture the topology of a large dataset. The 50x50 maps were trained on the same, larger human K562 dataset presented in 4.2 K562 Hi-C Data Analysis.

Note that the table appears to present a trend of fractionally better stability scores for maps trained with the Pearson correlation similarity metric, and a fractionally lower percent of unoccupied nodes for those trained with the cosine similarity metrics. However, these trends are not present in all pairings, and the fractional differences are small enough to not be considered significant.

The size of the grid was not optimized. The grid size, as it expands, will tend to increase the quality and decrease the stability. The stability decreases because it only considers nodes and direct neighbors, and on larger grids, even somewhat similar data points may be split apart by the number of available nodes. However, in cases when the maps in comparison is too small, the stability will increase as size increases, as seen in Table 1. This occurs because, when there are too few nodes to capture the data space, the trained map cannot faithfully reflect the topology of the dataset. Therefore, two maps of that size, trained separately, cannot be expected to capture the dataset in the same way.

The quality score goes up for the same reason, with more nodes to occupy the dataspace, it is easier for differing data points to occupy a single node, and shape that node's weight vector. A larger network is generally preferred, as this allows for a better spread and a more intuitive layout. The drawbacks, however, are that larger maps take longer to train, and may introduce unnecessary empty space. 50x50 is the default grid size in the current implementation, but some experimentation may be necessary to find a network that has the right balance of empty space and spread for any given dataset.

A linear decay rate was chosen as the optimum decay rate, over an exponential decay rate, because the resultant maps were identical in terms of quality and stability, and as well as to the eye. The exponential decay functions took several additional operations to complete at each iteration, so in the interest of training time, linear decay was selected.

The Gaussian kernel variance showed a similar pattern to that of the similarity metrics. Table 2 shows the weights by distance for several kernel variances.

Table 2: Kernel Variance Weights

Variance	Distance	Weight
5	0	1
5	1	0.980199
5	2	0.923116
5	3	0.83527
2	0	1
2	1	0.882497
2	2	0.606531
2	3	0.324652
1	0	1
1	1	0.606531
1	2	0.135335
0.8	0	1
0.8	1	0.457833
0.8	2	0.043937
0.8	3	0.000884
0.4	0	1
0.4	1	0.043937
0.4	2	3.73E-06
0.4	3	6.1E-13
0.2	0	1
0.2	1	3.73E-06
0.2	2	1.93E-22

As is shown in the table, there is a significant difference in the weight contributed by neighboring nodes from the neighborhood function between a variance of 5 and a variance of 0.2. The objective

of the neighborhood function, and the variance, is to start out by giving somewhat significant weight to surrounding nodes, and shrinking to almost insignificant weights from surrounding nodes as training proceeds. This is achieved appropriately by the default behavior of variance shrinking from 1.0 to 0.2. There have been no significant differences observed when changing the variance starting and ending parameters slightly. This breaks down with extreme values, for example, maps with starting and ending variances of greater than 4 show a marked reduction in stability. However, because training was found to be robust to the slight changing of this parameter, it is changeable for users. Varying this parameter may help to achieve a desired result with certain data sets.

4.2 K562 Hi-C Data Analysis

To explore the genomic processes of human K562 cells, the Hi-C data set from Rao, *et al.* was first normalized according to a very basic normalization protocol, outlined in 4.2.1 Data Normalization. The data set contained 12,155 non-empty chromatin-interaction vectors, covering the human genome with a bin size of 250Kbp. The data set was then used to train a 50 x 50 SOM over 1000 iterations, with n=2 (to train two maps for a stability metric), and a Kernel variance ranging from 1 to 0.2. The Pearson correlation similarity metric was used. The training process took 46 hours employing 40 processors. This produced a map with a 0.8704 cosine quality score, 0.7442 Pearson quality score, 0.8184 stability score, and 44.16% empty nodes. The whole map can be seen in Figure 16, as well as individual chromosomes projected onto the whole map.

Figure 16: K562 Trained SOM

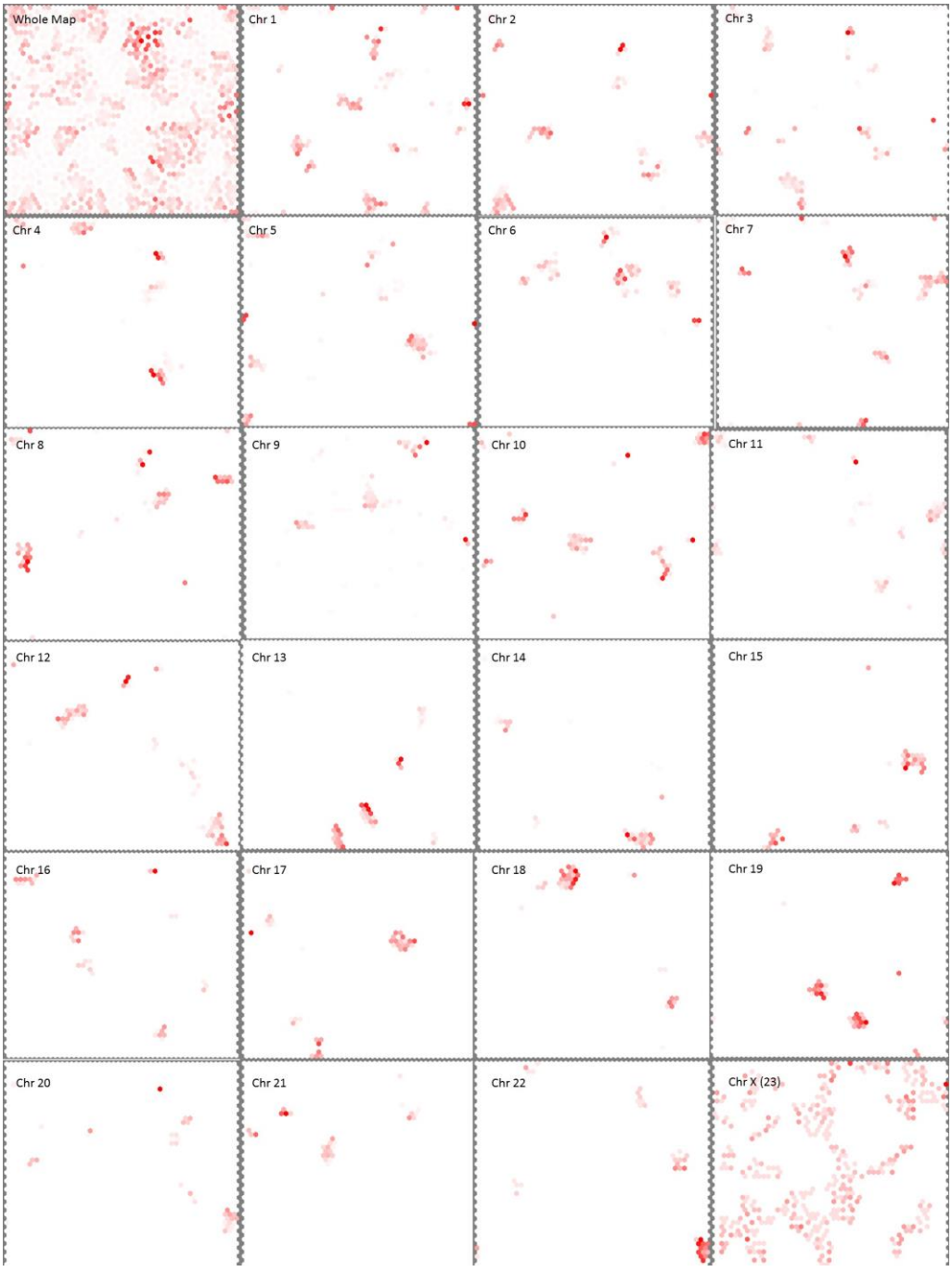


Figure 15: K562 Trained SOM
This figure presents the whole map (top left) produced from training on the O/E normalized K562 dataset, as well as each individual chromosome. Chromosomes are known to be segregated in the nucleus, so they are expected to be segregated on the SOM, even in O/E sets when the diagonal signal has been removed.

4.2.1 Data Normalization

The K562 Hi-C data set used to construct the matrix on which the K562 SOM was trained was obtained from Rao, *et al.* The 250Kbp resolution data set contained interaction frequency counts between 250Kbp bins. Original code was employed to assemble a matrix containing the pairwise interaction frequencies between bins.

Original code was again employed to do a very basic normalization of the data. As discussed, non-normalized Hi-C interaction matrices can have a dominating effect on the diagonal. The interest in the Hi-C assay is often on long-range interactions, which reflect the larger-scale organization of the nucleus, rather than the simple linear proximity of loci. Therefore, in order to deflate the diagonal signal, each index on the matrix was divided by the average interaction count between bins of that linear distance. Inter-chromosomal counts were divided by a grand average of all inter-chromosomal counts. These averages can be thought of as a very simplistic measure of expected count, yielding an observed over expected matrix upon division. The resulting matrix had a significantly smaller diagonal signal, and was then suitable for SOM training. This normalization protocol is quite simple, and there are many tools available for more sophisticated normalization protocols, but even this simple measure was able to remove the strong diagonal signal.

4.2.2 ChIP-Seq Analysis

After training a SOM on the K562 Hi-C dataset, we attempted to find any genomic activities that seemed to be segregated or otherwise unequally distributed within chromatin interaction space. A total of 128 K562 ChIP-Seq experiments were projected on to the map, and some of the top normalized Gini coefficient producing sets are described in

Table 3. A table of the full set of values is provided in Appendix A. Data sets for projection for both this and the following section were obtained from the ENCODE project (ENCODE Consortium, 2012).

Table 3: Chromatin Segregation of Protein-DNA Interactions

Protein	Basic Gini	Normalized Gini	Z-Value
Pol3	0.989092078	0.612437224	5017.452144
BCL3	0.698342201	0.507026076	239.3984998
BCLAF1	0.606362385	0.473601177	249.3337756
SIX5	0.785693902	0.422140268	409.2638844
HDAC8	0.917119511	0.339445583	1023.238001

The values depicted in the table suggest that for the K562 cell type, by virtue of their high Gini coefficients, these proteins are significantly unequally distributed in the nucleus. The output images for each are shown in Figure 17.

Figure 17: Example Projections of ChIP-seq Data onto the Trained K562 SOM

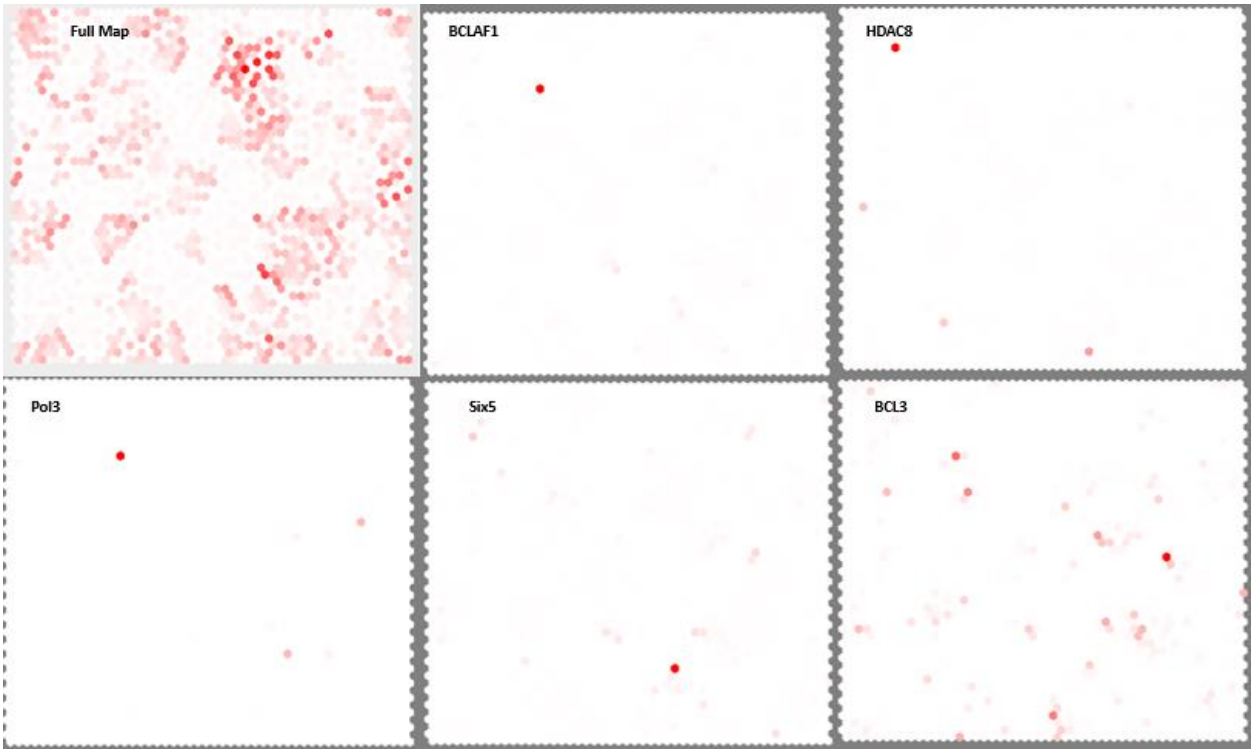


Figure 17: Example Projections of ChIP-seq Data onto the Trained K562 SOM
Shown are 5 highly segregated datasets. The data sets are weighted, and the projections use those weights for heat mapping (as opposed to equally weighting the loci)

Note that for some high scores, the results may be slightly dubious. In the case of Pol3, the ChIP-Seq experiment found 250 peaks, and of these peaks, four had extremely high weight (9000+), while the rest have relatively very little weight (on the order of ~20-30). This allows for these four peaks to dominate the output image, and though there are enough peaks to otherwise produce a reliable output, these outlier weights effectively reduce the number of peaks considered by the Gini coefficient to 4 by dwarfing any contribution by the other peaks. The normalized Gini coefficient should be robust to these effects, but it can be useful for the sake of confidence in the results to perform the same analysis with the equal weighting parameter invoked. Equal weighing produced similarly high Gini coefficients for Pol3 of 0.90708 for the basic Gini, 0.74688 for the

normalized Gini and a Z-value of 364.13. The dataset, when projected with equally assigned weights, produced very similar projections, shown in (Figure 18).

Figure 18: Pol3 ChIP-seq Projections

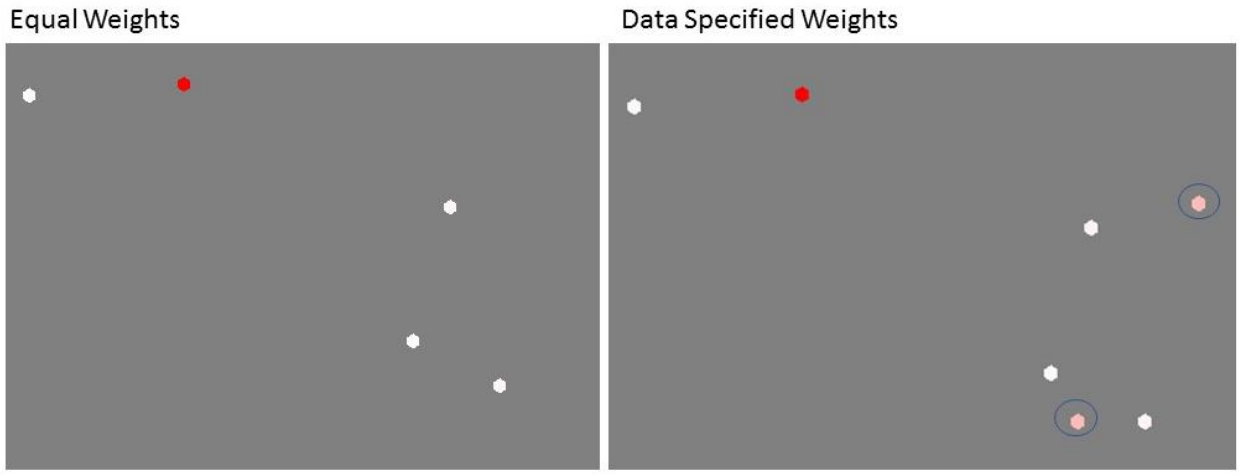


Figure 18: Pol3 ChIP-seq Projections
The projections of the same data set are shown. On the left, each data point is assigned an equal weight, while on the right the weights are specified by the dataset. Circled are two nodes which each have a single data weighted bin assigned, these bins contained a single heavily weighted locus. When weighting equally, these nodes drop below the 1% weight threshold for heat mapping. The other 2 highly weighted loci are assigned to the deep red node. Beside these two circled nodes, the projections are clearly very similar between the two weighting schemes.

Among the transcription factors and proteins that produced high Gini coefficients are several which are expected to be segregated in the nucleus. For example, Pol3 itself is expected to occupy a specific region in the nucleus. The nucleolus, where ribosomes are constructed, is segregated in the nucleus, and localized ribosomal RNA transcription contributes to the formation of this structure (Warner, 1990). Pol3 transcribes rRNA, so it is reasonable to expect Pol3 to be segregated in the nucleus just as the nucleolus is.

Another interesting result here is that BCL3 is highly segregated. B cell lymphoma-3, or BCL3, is a lymphocyte proliferation factor, and is associated with many leukemias. The K562 cells used to train the SOM come from an immortalized leukemia line, and K562 cells have been found to have

BCL3 highly upregulated (Kim et al., 2011). The apparent segregation of BCL3 indicated by the Gini coefficient may be related to this association between the BCL3 upregulation and the immortalization of the K562 line.

In a similar vein, BCLAF1, also known as BtF, is known to be an important apoptosis regulator (Kasof et al., 1999). In the absence of other factors, BCLAF1 is a potent pro-apoptotic factor, but in the presence of certain viral proteins, or BCL2, BCLAF1 functions as an anti-apoptotic factor (Kasof et al., 1999). Apoptosis dysregulation is an important aspect in the immortalization of cell lines as well as the development of cancers. Again, it is interesting that the K562 cell line, which is a cancer cell line, produced results indicating that immortality related factors are segregated in the chromatin.

Some of the other factors, like HDAC8, are associated with chromatin organization. Chromatin remodeling and modifying factors are expected to be segregated for the same reason that histone marks are expected to be segregated. This is discussed further below. Of the other high scoring factors, there are none that jump out as having an intuitive reason for their segregation. While the purpose of their segregation is not apparent, it may be that the localization of these factors is driven by or driving cellular functions.

4.2.3 Histone Mark Analysis

The same map used to project the K562 transcription factor ChIP-Seq experiments, was also used to project K562 histone modification ChIP-Seq experiments. Twenty datasets were projected on to the map, and the results are shown in Table 4.

Table 4: Nuclear Segregation of Histone Marks

Histone Mark	Basic Gini	Normalized Gini	Z-Value
H3K36me3	0.487263313	0.466556984	473.1949
H3K27me3	0.493183403	0.460288164	453.7636
H3K9me1	0.496800498	0.452492936	356.5716
H4K20me1	0.487943901	0.450871578	378.079
H3K4me1	0.460259407	0.440922798	436.0645
H3K9acB	0.441977299	0.423582736	409.2338
H3K36me3	0.422154058	0.409797141	492.9958
H3K27me3	0.384840697	0.367691206	429.0847
H3K4me1	0.380962527	0.366724547	367.9419
H3K9me3	0.419880485	0.365299433	247.2153
EZH2	0.379429003	0.357487124	365.0068
H3K27ac	0.393375994	0.350667651	226.804
H3K4me3B	0.400814897	0.34644021	185.2663
H3K9ac	0.38904972	0.344508236	218.7526
H3K79me2	0.420648096	0.342916102	184.2818
H3K4me3	0.388072296	0.334651807	185.6876
H3K27me3B	0.336349845	0.322359967	375.5205
H3K4me2	0.351426418	0.318707207	210.9459
H3K4me3	0.357422613	0.317237004	192.1909
H2A.Z	0.329730966	0.316115769	316.5803

Histone marks are known to be significant drivers of chromatin organization, so it is perhaps unsurprising that the modifications are segregated in the nucleus. The output images of some of these histone marks is shown in Figure 19.

Figure 19: Example Projections of Histone Mark ChIP-seq onto The K562 SOM

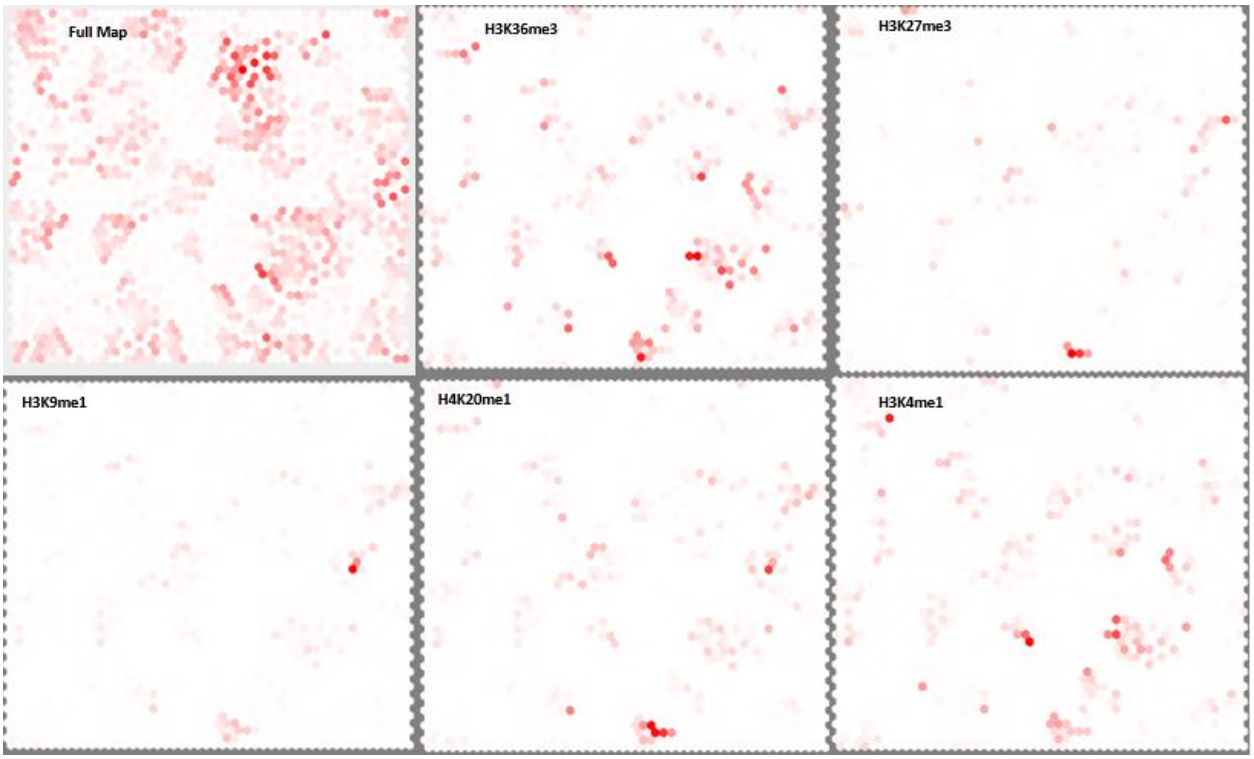


Figure 19: Example Projections of Histone Mark ChIP-seq onto The K562 SOM
Shown are 5 highly segregated histone mark datasets. The data sets are weighted, and the projections use those weights for heat mapping (as opposed to equally weighting the loci). Histone mark loci are expressed as regions. In cases where regions span multiple bins, the weigh is distributed equally among the bins spanned by the indicated region.

The main driver of nuclear organization in mammalian cells is the packaging of chromatin (Bártová et al., 2008). Packaging is organized based on the regions of similarly modified histone modifications. Histone modifications can recruit several other factors to change the architectural landscape of the surrounding chromatin, including histone modification enzymes to reciprocally proliferate a certain set of marks, and larger chromatin remodeling complexes to assist in

packaging or unpacking of regions. Specific histone marks can also recruit transcription factors (Mattout et al., 2015). Similarly marked histones will tend to exist near one another due to both the nature of the packaging, and the reciprocal nature of modification. Because of the integral role played by histone modifications in the chromatin organization, it makes sense that many of these marks, and the factors which modify histones to create these marks, produced high Gini coefficients, indicating high degrees of segregation. These results are consistent with those of Rao and colleagues, showing that observed nuclear subcompartment organization is associated with histone modification patterns.

H3K36me3, for example, is associated with heterochromatin (Chantalat, 2011). Heterochromatin contributes to transcriptional silencing by packaging DNA sufficiently tightly so that it is inaccessible for transcription. Heterochromatin regions are segregated from transcriptionally active regions, so the associated marks, like H3K36me3 are expected to be segregated.

It must be noted, however, that the chromatin marks with the highest Gini coefficients are significantly lower than some of the top transcription factors. This may be because histone modifications are more broadly applied to regions of the genome, creating several, large architecturally similar regions in the genome, not necessarily together. This would lead to the observed general lowering of Gini coefficients. It should also be noted that, unlike the transcription factor datasets, which produced several insignificant or even negative Gini coefficients, every histone ChIP-seq dataset produced a Gini coefficient indicating at least some degree of segregation.

Chapter 5

Discussion

The Self-Organizing Map algorithm can be used to map high-dimensional data space onto a 2D grid. The algorithm is especially useful for this task due to its neighborhood function, which preserves the topology of the input data space. This means that the grid onto which the data is eventually projected is representative of the structure of the data, and that similar data points in the high-dimensional data space will tend to be closer together on the projection space.

The Hi-C chromatin capture assay produces a dataset that can be thought of as high-dimensional. In a binned genome, interaction frequencies can be used to create an all-by-all data matrix in which the pairwise interaction frequency between bins is described. Rows or columns of this matrix can then be taken as high-dimensional data vectors, which represent the frequency with which individual bins interact with every other bin along the genome. Such vectors describe a data space. This data space is representative of the average 3D organization of the genomes on which the Hi-C assay is performed, because only proximal DNA can form the crosslinks used to identify interactions. Therefore, higher interaction frequencies indicate that two bins are closer together in the nucleus more often. The resulting data vectors are therefore related to the consensus relative distances between a given genomic bin and every other bin.

The Self-Organizing Map algorithm can be employed to project the Hi-C data space onto a 2D grid that captures the organization of the genome from which the chromatin interaction data was collected. Chromosomes, which are known to be clustered together in the nucleus, are faithfully represented as clusters in the projection space (Figure 16).

The trained SOM can then be efficiently employed to project a given genomic activity. This produces a view of the map that shows only the distribution of that genomic activity in chromatin interaction space, which can reveal the degree of its segregation in the nucleus. Such segregation can also be found quantitatively by analyzing the distribution of the weight of the genomic activity data set on the map. This is quantified using a Gini coefficient. The Gini coefficient has indeed been successful in deconvolving segregations of genomic activities that may not be apparent by eye when the data set was projected on the map.

It is important to consider the data being used both to train the map and to project on to the map. Hi-C data has biases and a strong diagonal signal. Often the goal when using Hi-C data is to understand interactions that are not on the diagonal, so normalization of the interaction matrix may be necessary. This does, however, change the topology of the data space, which in turn changes how that topology will be projected by the trained SOM. It is necessary to be careful of the degree to which the topology is being changed, and in what ways, when employing normalization methods.

In a similar vein, it is always necessary to inspect and understand genomic activity data sets which are being projected on to the SOM, as poorly distributed weights in those data sets, as well as sets with too few data points, can produce untrustworthy, high Gini coefficient values.

It is possible to project multiple sets of genomic activity data on a trained SOM at once. This produces a projection where co-clustering can be estimated by eye. However, a quantitative way to describe the degree of co-clustering, as the Gini coefficient does of single set segregation, has not yet been developed. This is an interesting point for future development.

I have developed a novel approach to visualization chromatin interaction data on a 2D grid using an implementation of the Self-Organizing Map algorithm. This approach allows for efficient an intuitive visualizations and quantifications of distribution and segregation of biochemical activities in chromatin interaction space. The utility of this method has been demonstrated using human Hi-C data and exploring the distribution of ChIP-seq data. I hope that this software will be useful in future analyses of chromatin organization and compartmentalization.

Appendix A

Complete K562 ChIP-seq Gini Coefficient Table

Table 5: K562 ChIP-seq Gini Coefficients

Protein	Basic Gini	Normalized Gini	Z-Value
BDP1	0.929997	0.759391	535.6443
RPC155	0.904941	0.748938	521.5559
ZNF274	0.97811	0.700124	2821.109
BRF1	0.952528	0.686506	728.5776
Pol3	0.989092	0.612437	5017.452
BCL3	0.698342	0.507026	239.3985
TFIIIC-110	0.717938	0.499411	374.2803
BCLAF1	0.606362	0.473601	249.3338
SIX5 (rep1)	0.785694	0.42214	409.2639
SIX5 (rep2)	0.627573	0.415966	188.0904
Sin3Ak	0.716469	0.383033	580.3553
Pol2	0.661036	0.35898	583.699
c-Myc	0.39913	0.355038	234.655
HMGN3	0.387195	0.354502	259.1805
Mxi1--AF4185	0.812003	0.344365	649.5257
HDAC8	0.91712	0.339446	1023.238
ETS1-	0.66589	0.335761	564.528
Ini1	0.650645	0.335758	264.2173

TAF7	0.801346	0.334554	456.8543
SETDB1	0.848103	0.33026	672.8543
Pol2	0.427616	0.329732	277.8437
SP2	0.83464	0.328402	525.5105
SP1	0.47401	0.326855	151.7415
Brg1	0.613832	0.322425	238.0082
CCNT2	0.469292	0.316369	275.2136
CREB1	0.385058	0.310581	167.5595
GTF2F1	0.793623	0.307144	466.6026
CBX2	0.826534	0.300106	302.3202
THAP1	0.741069	0.298818	385.546
GTF2B	0.703948	0.298147	233.5379
ZNF274	0.68065	0.29652	117.4255
PML	0.331963	0.295332	208.0949
HDAC2	0.672742	0.293673	415.0973
GABP	0.526447	0.290444	159.6096
TAF1	0.529883	0.287758	244.0777
Max	0.689836	0.284581	292.9195
SAP30	0.659742	0.282049	111.6647
c-Myc	0.55856	0.282012	100.3675
GATA2	0.362539	0.280635	154.9091
ATF3	0.700876	0.280613	431.1431

CHD1	0.67057	0.278819	111.0141
SIRT6	0.934964	0.277724	745.1334
ELK1	0.879703	0.276616	662.5498
SIRT6	0.517899	0.276118	103.2604
ARID3A	0.719178	0.271235	502.4899
UBF	0.837851	0.270313	402.0929
c-Fos	0.640867	0.267275	267.9905
HCFC1	0.686435	0.264061	250.592
RFX5	0.927882	0.262569	887.2031
CEBPD	0.30083	0.259212	148.7201
Nrf1	0.858475	0.258295	564.8609
TBLR1	0.861181	0.254331	819.4916
Pol2-b	0.681148	0.250869	119.7497
CBX8	0.776907	0.250455	193.8224
TBP	0.673642	0.247754	205.8908
ZBTB33	0.514528	0.247301	92.75603
COREST	0.843809	0.246678	543.6441
UBTF	0.714102	0.245594	158.3439
PCAF	0.95973	0.244652	866.4432
TRIM28	0.281493	0.243168	134.2031
REST	0.484909	0.242884	81.81789
TBLR1	0.750304	0.242024	517.6984

HDAC1	0.597723	0.2411	84.8049
MAZ	0.649896	0.240444	130.4769
c-Jun	0.51647	0.239553	216.338
CEBPB	0.269773	0.238424	142.2692
GATA-2	0.673343	0.238325	319.0314
FOSL1	0.73708	0.230334	391.2874
YY1	0.775294	0.227752	251.6008
STAT5A	0.307333	0.227653	89.25976
MEF2A-	0.840459	0.227323	486.4112
c-Myc	0.668342	0.225476	154.8053
ZNF-MIZD	0.812267	0.221058	489.8853
RNF2	0.820912	0.220879	195.5301
CHD4-Mi2	0.966295	0.217552	1029.558
NSD2	0.840279	0.217112	245.6138
GATA-1	0.780103	0.216448	344.9818
FOS	0.267758	0.213215	91.48029
p300	0.818439	0.210783	309.6566
USF2	0.891604	0.210747	606.2701
NR4A1	0.915632	0.210056	464.1784
Egr-1	0.601987	0.209533	94.97379
Max	0.579224	0.208293	96.48813
SRF	0.648918	0.207268	154.4871

YY1	0.587135	0.206178	141.7057
GATA2	0.781632	0.205825	235.9099
E2F4	0.6613	0.205391	134.6408
E2F6	0.479464	0.199691	68.73391
HDAC2	0.664511	0.198779	77.1464
JunD	0.431591	0.195945	212.266
TR4	0.933587	0.195723	610.7738
JunB	0.519429	0.195606	54.80904
CDP	0.867951	0.193695	599.226
CHD7	0.916404	0.192204	422.8463
ZNF263	0.720514	0.191894	137.5779
ATF1	0.486709	0.191751	119.6784
NCoR	0.962287	0.181118	1040.399
EZH2	0.75777	0.178754	89.04758
ELF1	0.557753	0.178319	123.9114
LSD1	0.725417	0.172652	102.2189
USF-1	0.589304	0.16458	111.1037
CTCFL	0.708615	0.164538	206.9253
NF-E2	0.642799	0.164159	105.6914
BHLHE40	0.666715	0.157291	74.7865
KAP1	0.836517	0.152586	266.3005
PU.1	0.274967	0.152575	47.36097

COREST	0.542173	0.149816	49.34809
NRSF	0.499426	0.148238	53.28469
ZNF384	0.626337	0.14783	51.61222
CBX3	0.822373	0.143561	94.01569
Znf143	0.688417	0.13805	87.43964
GATA1	0.776371	0.13158	160.5638
p300	0.685863	0.13051	96.8667
c-Jun	0.721656	0.124568	86.02239
E2F6	0.47944	0.124484	32.76252
YY1	0.558539	0.123762	53.71988
CTCF	0.294742	0.120079	16.5843
SETDB1	0.785905	0.120022	107.176
CHD2	0.745968	0.116175	91.41211
JunD	0.545246	0.109932	29.24979
TAL1	0.593949	0.106208	38.83517
CBP	0.649908	0.098362	25.83986
CTCF	0.24939	0.097736	-5.38232
CTCF	0.325857	0.094857	-0.56524
NF-YA	0.724294	0.078927	33.2174
SUZ12	0.778655	0.076269	34.15671
NF-YB	0.281545	0.066152	-18.1749
CTCF	0.458816	0.063138	-6.1462

Rad21	0.440345	0.050738	-38.9125
SMC3	0.681578	0.042126	-0.75668
HDAC6	0.81277	0.036786	10.26391
MafF	0.614797	0.033708	-10.484
CTCF	0.162732	0.02202	-54.6784
Rad21	0.47063	0.010859	-45.429
MafK	0.638982	-1.61E-04	-31.038
CEBPB	0.405048	-3.35E-04	-38.2519
SETDB1	0.877345	-0.01	-40.7606
p300	0.771383	-0.01037	-30.4297

Appendix B

Code

The tool developed and described herein can be accessed for free at GitHub at this address:

- https://github.com/seqcode/somatic

BIBLIOGRAPHY

1. Alhoniemi, E., Himberg, J., Parhankangas, J., and Vesanto, J. SOM Toolbox: implementation of the algorithm.
2. Bártová, E., Krejčí, J., Harničarová, A., Galiová, G., and Kozubek, S. (2008). Histone Modifications and Nuclear Architecture: A Review. J Histochem Cytochem 56, 711–721.
3. Catalano, M., Leise, T., and Pfaff, T. (2009). Measuring Resource Inequality: The Gini Coefficient. Numeracy 2.
4. Chantalat, S., Depaux, A., Héry, P., Barral, S., Thuret, J.-Y., Dimitrov, S., and Gérard, M. (2011). Histone H3 trimethylation at lysine 36 is associated with constitutive and facultative heterochromatin. Genome Res 21, 1426–1437.
5. Consortium, T.E.P. (2012). An integrated encyclopedia of DNA elements in the human genome. Nature 489, 57–74.
6. Dekker, Jo., Rippe, K., Dekker, M., and Kleckner, N. (2002). Capturing chromosome conformation. Science; Washington 295, 1306–1311.
7. Duan, Z., Andronescu, M., Schutz, K., McIlwain, S., Kim, Y.J., Lee, C., Shendure, J., Fields, S., Blau, C.A., and Noble, W.S. (2010). A three-dimensional model of the yeast genome. Nature 465, 363–367.
8. Guelen, L., Pagie, L., Brasset, E., Meuleman, W., Faza, M.B., Talhout, W., Eussen, B.H., de Klein, A., Wessels, L., de Laat, W., et al. (2008). Domain organization of human chromosomes revealed by mapping of nuclear lamina interactions. Nature 453, 948–951.
9. Graczyk, P.P. (2007). Gini Coefficient: A New Way To Express Selectivity of Kinase Inhibitors against a Family of Kinases. J. Med. Chem. 50, 5773–5779.
10. Harch, B.D., Correll, R.L., Meech, W., Kirkby, C.A., and Pankhurst, C.E. (1997). Using the Gini coefficient with BIOLOG substrate utilisation data to provide an alternative quantitative measure for comparing bacterial soil communities. Journal of Microbiological Methods 30, 91–101.

11. Heinz, S., Benner, C., Spann, N., Bertolino, E., Lin, Y.C., Laslo, P., Cheng, J.X., Murre, C., Singh, H., and Glass, C.K. (2010). Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. Mol. Cell 38, 576–589.
12. Hsieh, T.-H.S., Weiner, A., Lajoie, B., Dekker, J., Friedman, N., and Rando, O.J. (2015). Mapping Nucleosome Resolution Chromosome Folding in Yeast by Micro-C. Cell 162, 108–119.
13. Kasof, G.M., Goyal, L., and White, E. (1999). Btf, a Novel Death-Promoting Transcriptional Repressor That Interacts with Bcl-2-Related Proteins. Mol Cell Biol 19, 4390–4404.
14. Kim, B., Chang, H.G., and Kim, S.J. (2011). Human lactoferrin upregulates BCL-3 in the K562 erythroleukemia cell. BioChip J 5, 362–368.
15. Kohonen, T. (2012). Self-Organizing Maps (Springer Science & Business Media).
16. Lee, J.A., and Verleysen, M. (2007). Nonlinear Dimensionality Reduction (New York: Springer).
17. Lieberman-Aiden, E., van Berkum, N.L., Williams, L., Imakaev, M., Ragoczy, T., Telling, A., Amit, I., Lajoie, B.R., Sabo, P.J., Dorschner, M.O., et al. (2009). Comprehensive mapping of long range interactions reveals folding principles of the human genome. Science 326, 289–293.
18. Mattout, A., Cabianca, D.S., and Gasser, S.M. (2015). Chromatin states and nuclear organization in development — a view from the nuclear lamina. Genome Biology 16, 174.
19. Meaburn, K.J., and Misteli, T. (2007). Cell biology: Chromosome territories. Nature 445, 379–381.
20. Moindrot, B., Audit, B., Klous, P., Baker, A., Thermes, C., Laat, W. de, Bouvet, P., Mongelard, F., and Arneodo, A. (2012). 3D chromatin conformation correlates with replication timing and is conserved in resting cells. Nucl. Acids Res. gks736.
21. Mortazavi, A., Pepke, S., Jansen, C., Marinov, G.K., Ernst, J., Kellis, M., Hardison, R.C., Myers, R.M., and Wold, B.J. (2013). Integrating and mining the chromatin landscape of cell-type specificity using self-organizing maps. Genome Res. 23, 2136–2148.

22. Pandey, M.D., and Nathwani, J.S. (1996). Measurement of socio-economic inequality using the Life-Quality Index. Soc Indic Res 39, 187–202.

23. Pope, B.D., Ryba, T., Dileep, V., Yue, F., Wu, W., Denas, O., Vera, D.L., Wang, Y., Hansen, R.S., Canfield, T.K., et al. (2014). Topologically associating domains are stable units of replication-timing regulation. Nature 515, 402–405.

24. Rao, S.S.P., Huntley, M.H., Durand, N.C., Stamenova, E.K., Bochkov, I.D., Robinson, J.T., Sanborn, A.L., Machol, I., Omer, A.D., Lander, E.S., et al. (2014). A 3D Map of the Human Genome at Kilobase Resolution Reveals Principles of Chromatin Looping. Cell 159, 1665–1680.

25. Reddy, K.L., and Feinberg, A.P. (2013). Higher order chromatin organization in cancer. Semin Cancer Biol 23, 109–115.

26. Roweis, S.T., and Saul, L.K. (2000). Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science 290, 2323–2326.

27. Stadhouders, R., Kolovos, P., Brouwer, R., Zuin, J., van den Heuvel, A., Kockx, C., Palstra, R.-J., Wendt, K.S., Grosveld, F., van Ijcken, W., et al. (2013). Multiplexed chromosome conformation capture sequencing for rapid genome-scale high-resolution detection of long-range chromatin interactions. Nat Protoc 8, 509–524.

28. Warner, J.R. (1990). The nucleolus and ribosome formation. Current Opinion in Cell Biology 2, 521–527.

29. Yitzhaki, S. (1979). Relative Deprivation and the Gini Coefficient. The Quarterly Journal of Economics 93, 321–324.

30. Zhang, Z., Li, G., Toh, K.-C., and Sung, W.-K. (2013). 3D Chromosome Modeling with Semi-Definite Programming and Hi-C Data. Journal of Computational Biology 20, 831–846.

31. Cosine similarity, Pearson correlation, and OLS coefficients | AI and Social Science – Brendan O'Connor.

ACADEMIC VITA

Timothy Robert Kunz
trk5150@gmail.com
411 Hillcrest Avenue, State College, PA, 16803

Education

The Pennsylvania State University
Eberly College of Science, Schreyer Honors College — University Park, PA
Bachelor of Science, Biochemistry & Molecular Biology — August 2013-May 2017
Minor in Statistics
Honors in Biochemistry and Molecular Biology

Visualizing and Understanding Chromatin Interactions Using Self-Organizing Maps
Shaun Mahony

Work Experience

May 2016 - August 2016
Bioinformatics Programmer
Pennsylvania State University
Shaun Mahony

June 2011 – May 2015
Server, Busboy
Texas Roadhouse, State College, PA
Amanda McGonigle

Activities

Microbiology Learning Assistant (January 2016 - May 2016)
Pennsylvania State Certified EMT-B (August 2013)
Operations Coordinator for Movin' On Music Festival (August 2017 - May 2017)
Math Tutoring (August 2015 - December 2015)

Honors and Awards

Edward B. Nelson Undergraduate Research Fund (2016)
Erickson Discovery Grant (2015)
NIH ABI Grant 1564466 (2016)
Dean's List, The Pennsylvania State University (August 2013- May 2017)