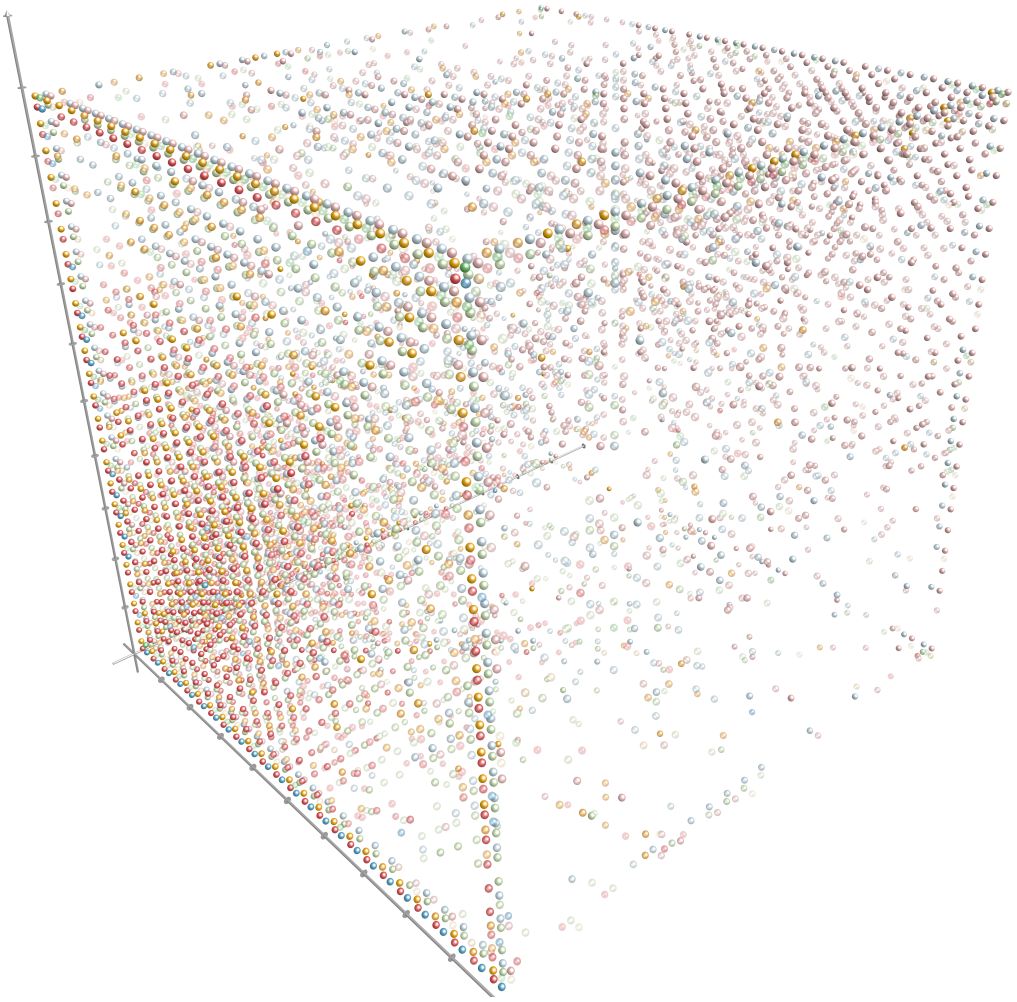
# TiBi-3D

## a Guide through the World of Epigenetics



UNIVERSITÄT LEIPZIG
Fakultät für Mathematik und Informatik
Institut für Informatik

# TiBi-3D

## a Guide through the World of Epigenetics

**Masterarbeit**

Leipzig, den 18. Dezember 2014
Betreuende Hochschullehrer:
Prof. Gerik Scheuermann
Juniorprof. Sonja J. Prohaska

vorgelegt von:

Daniel Gerighausen

Studiengang Informatik

# Contents

# Abstract

In the last two decades the study of changes in the genome function that are not induced by changes in DNA has consolidated a strong research field called ”epigenetics”. Chromatin state changes play an essential role in the regulation of transcription of many genes, thus controlling cell differentiation. A large part of these changes is due to histone modifications that alter the accessibility of the DNA. Current state of the art visualization methods for the analysis of epigenetic data sets are not suited to represent the relationship between the combinatorial pattern of histone modifications and their regulatory effects. A recent strategy to generate a global overview of these interactions is the use of scatterplots. One of the biggest weaknesses of scatterplots is the overplotting. This can be solved using a 2D tiled-binned representation strategy, where dividing a scatterplot into bins consisting of tiles for each modification pattern is possible. However, this 2D strategy does not allow to represent the interaction of more than two histone modifications. Here, *TiBi-3D*, a tool that can visualize the combinatorics of histone modifications with tiled-binned 3D scatterplots, is presented. Two important features of *TiBi-3D* are that tiles are represented with spheres in the scatterplot, and that their position and color encodes the histone modification pattern they represent. *TiBi-3D* also includes a transparency value assigned to each of that spheres to depict the amount of data points in each bin. In addition, to reduce the occlusion in the scatterplot each transparency value is initially filtered by an outlier detection, transformed to log scale, and then normalized. *TiBi-3D* provides features for exploration and interaction with the scatterplot and the data, thus enabling to examine the data set thoroughly. It is also possible to export the results as figures or in bed file format for further processing. By using *TiBi-3D*, for example, it was possible to observe new relations between the CpG-density and histone modifications in different cell types. In conclusion, *TiBi-3D* is an excellent tool for the analysis of global patterns in epigenetic data.

# Chapter 1

# Introduction

During the last years a new research field has been established next to genetics, it investigates changes in genome function: epigenetics. Research in this field studies changes in the genome function that are not induced by changes in DNA sequence. The DNA forms with special proteins, so-called histones, the chromatin, which is the material that builds the chromosome in eukaryotic cells. This histones can be modified by several molecules, thus changing, for example, the readability of the DNA wrapped these proteins. Therefore, these changes of the chromatin states are interesting for studying the differentiation of cell types, since the DNA does not change during the differentiation. Such changes can be seen on a phenotypic level. For example, cells of the nerve system produce biological messengers that are not produced in muscles or stomach. One reason is that the modifications of the histones change between different cell types and therefore, change the readability of the DNA. This influences the regulation of transcription of particular genes and thus, controls the differentiation of the cells. For this reason, studying the changes of the histone modifications is interesting for Bioinformaticians.

Using a special DNA sequencing method, ChIP-seq, it is possible to target and identify only special modified histones and sequences of the DNA bound to that protein. In this thesis, three histone modifications occurring at the histone H3 in mouse are analyzed. These modifications are known to activate or repress transcription. Additionally, the influence of genetic features, like the CpG-density or the length of the DNA segment bound to the histone is also studied. Since the data set has more than 800,000 elements, it is tedious to study it with known available tools, for instance, a genome browser to obtain an overview. Nevertheless, state of the art publications still analyzed histone modifications using genome browsers with stacked tracks for each combination of samples and histone modifications. Using this methods, it is possible to

study the effects of single histone modifications. The effects of combinations of modifications cannot be worked out with them.
Several work has been done before addressing this problem, but none of them has met all the demands, like reasonable results, a good overview, and showing the effects of all combinations of modifications. In my first approach, the global change of histone modifications between different cell types was analyzed using k-means++ clusterings and visualized with starplots and scatterplots [1] to reduce the complexity of the data set and to compare only the centroids of each cluster. Nevertheless, it turned out that the data set does not fulfill the requirements of k-means clustering like well separated clusters. Zeckzer et al. [2] published with my cooperation *TiBi-SPLOM*, a tiled binned 2D scatterplot matrix that visualizes the correlation of histone modifications pairs. Using *TiBi-SPLOM*, the user is able to study data sets in an exploratory manner. It visualizes the distribution of histone modifications between different cell types and the user has to work out the results by studying the scatterplots. However, the data used was generated for the correlation of three histone modifications, and as a result *TiBi-SPLOM* could not show all information of the data set. In this thesis, results are described that were hard to reproduce using *TiBi-SPLOM*. As an alternative *TiBi-3D* was developed to visualize the changes of three histone modifications during the cell differentiation. It extends the tiled binned scatterplots to three dimensions while reducing the effects of occlusion by providing suitable interaction methods with the plot, like rotating and filtering. Additionally, *TiBi-3D* pre-processes the data with a normalization method that tries to reduce the clutter in the plot affected by outliers in the data set. Using *TiBi-3D*, the user is able to filter the data set in many ways, thus being able to expose interesting changes in the plot related to changes in the histone modifications. It is also possible to export the results as a figure and the analyzed data into a bed file for further processing. For a better and easier comparison, it is possible to save the used perspectives of a scatterplot and apply them to a new data set.
In this thesis, the biological background of epigenetics and DNA sequencing is explained to better understand the effects of histone modifications. Then in Chapter 3, the related works are considered. In Chapter 4, all methods, adapted or newly developed for *TiBi-3D*, are described. In Chapter 5, the 3D tiled binned scatterplot is compared with the 2D tiled binned scatterplot by *TiBi-SPLOM*. Several new insights were observed with *TiBi-3D* and analyzed in this thesis.

# Chapter 2

# Background

## 2.1 Epigenetics

Organisms with more than one cell type evolved from unicellular organisms. Each individual cell of these organisms has the same genetic material, the DNA, and the same set of genes, in all of its cells. Nevertheless, the different cell types have different functions and for that purpose produce different transcripts by regulation of gene expression. The process of differentiation into cell types is determined by gene regulation. One part of this research field besides the study of transcription factors and the gene regulation by RNAs is called epigenetics and is defined as "the study of heritable changes in gene expression that are not mediated at the DNA sequence level"[3].
In eukaryotes the DNA is located in the nucleus. It is organized in chromosomes. Human, for example, has 46 chromosomes. As shown in Figure 2.3, the DNA forms a "beads on a string"-like structure with special proteins, the histones. This structure is called the 10nm chromatin fiber.
The histones H3, H4, H2A and H2B in two copies each are forming the histone octamer, a protein complex consisting of eight proteins. The combination of the histone complex and 146bp of DNA wrapped around the protein complex is called nucleosome (shown in Figure 2.1). Figure 2.5c shows the DNA in complex with the histone octamer. To stabilize the DNA between two nucleosomes [4], so called linker DNA, the histone H1 is attached to the DNA at its open ends on the nucleosome (Figure 2.1). The chromatin is packed differently tight as for example the centromeres (shown in Figure 2.3) are very condensed. These condensed areas are called *heterochromatin* [3] and cannot be accessed easily by the *RNA-polymerase*, which transcribes the DNA into RNA. In reverse, parts of the chromosome are packed less dense, the so-called euchromatin [3]

— areas which are often highly transcribed. These two structural units are formed by the histone complexes as they can be packed more tightly or less tightly by modifying their proteins with the help of enzymes. These enzymes attach different molecules to the histones at specific positions and change the conformation of them.

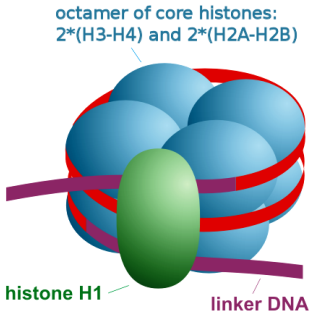



Figure 2.1: The nucleosome [5] consisting of the histone octamer, linker DNA, and the histone H1. 146bp of DNA is wrapped around the histone octamere.

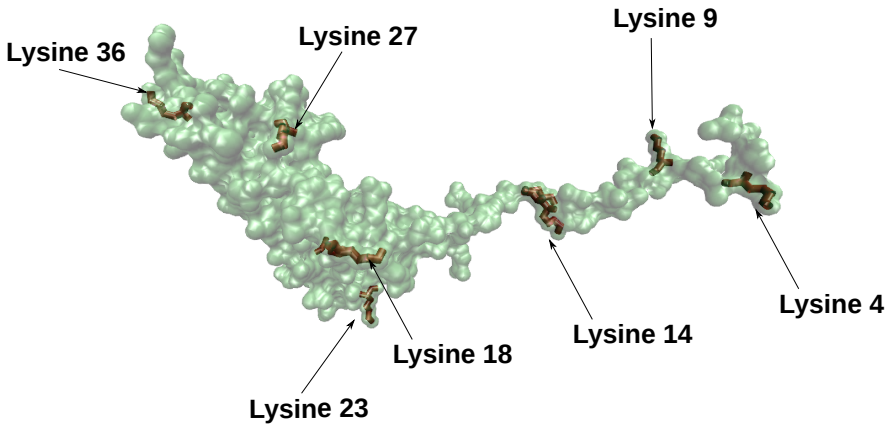



Figure 2.2: Positions of lysine in histone H3

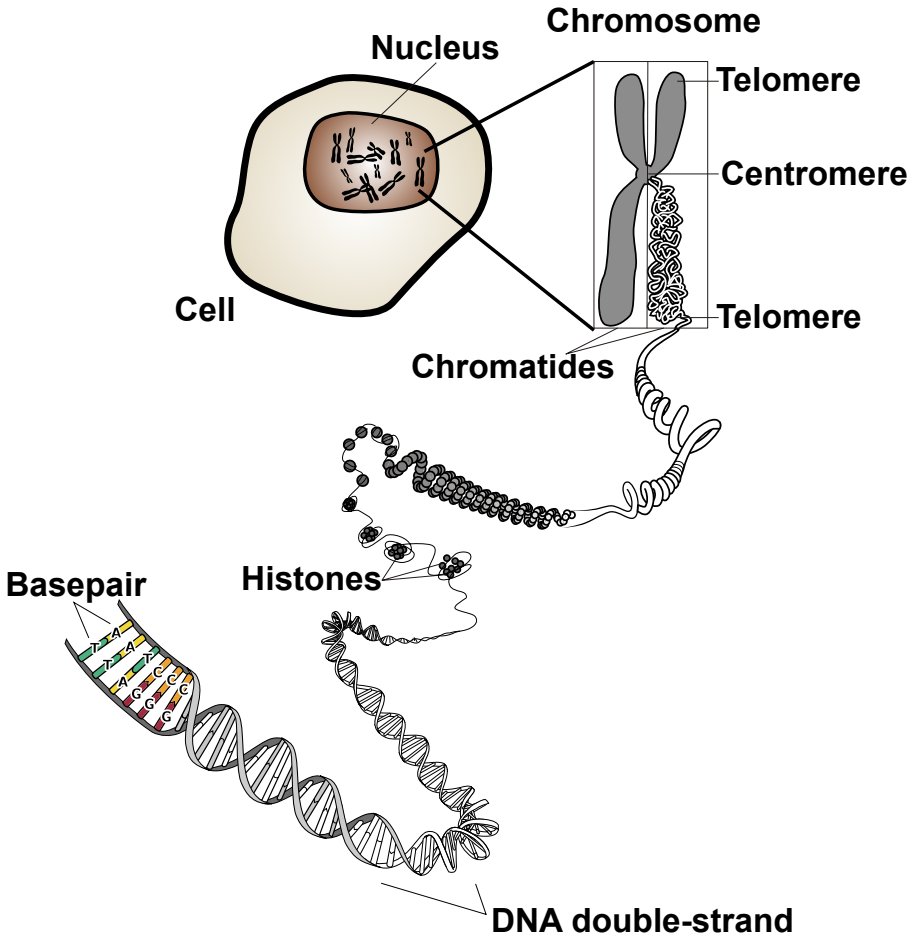



Figure 2.3: DNA and its structural organization in the nucleus of eukaryotic cells [6].

As shown in Figure 2.4, there are many possible modifications on each histone. Acetylation, methylation, phosphorylation, and ubiquitination are the most common. All of these modifications are attached to the histone tails. For example, the histone H3 can be modified by attaching a methylation group to one of the lysines. All positions of lysine are marked in Figure 2.2. All of these modifications can have a different impact on chromatin organization, and it is still a subject of research to reveal their functions. In Section 2.2.4 effects of those modifications relevant for this thesis are described.

## 2.2 Biological data

### 2.2.1 History of DNA Sequencing

After the DNA double helix was first published by Watson and Crick in 1953 [9], its sequencing has drawn much attention to reveal its sequence content. The

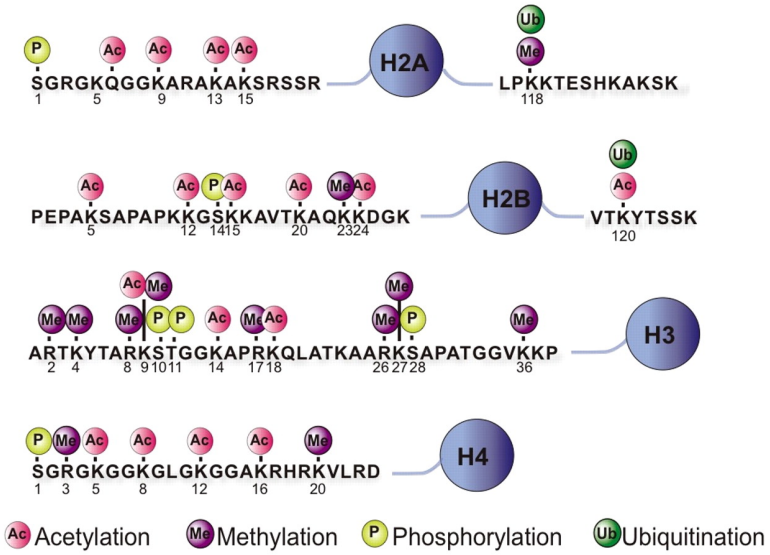
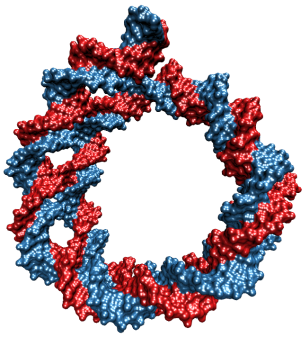



Figure 2.4: Known histone modifications involved in chromatin reorganization [7].

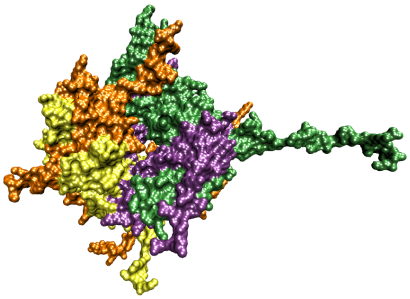
DNA sequence is build up from four nucleotides: deoxyadenosine monophosphate (A), deoxyguanosine monophosphate (G), deoxycytidine monophosphate (C), and thymidine monophosphate (T). All have a 2-deoxyribose sugar and at this sugar binds also a phosphate group. Additionally, one of the four nucleobases adenine, guanine, cytosine, and thymine is attached to to the sugar. The combination of these three correspond to the so-called nucleotides, which form a single strand of the DNA. DNA is double stranded, where nucleotides form *Watson Crick base pairs*: guanine - cytosine and adenine - thymine. The paired double strand makes the helical structure that characterizes the DNA. It is a very stable structure. When wrapped around the histone complexes it forms the chromatin.

### 2.2.2 Basic Sequencing

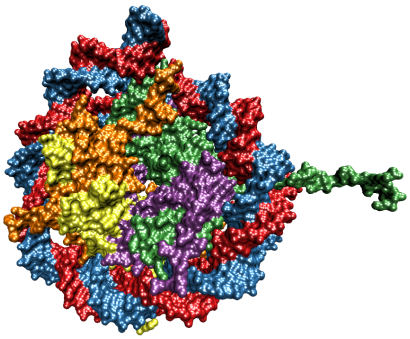
One of the first and widely used methods is *Sanger sequencing*. It was developed by Sanger and Coulson in 1977 [10]. It uses the DNA *polymerase* during the DNA in vitro replication. It is based on the incorporation of modified nucleotides that terminate DNA replication. At the beginning, the DNA has to be denatured, which allows to separate the helix into single strands that will act as templates. Then, a *primer* is added at the beginning of each segment. A *primer* is a short sequence of nucleotides where the DNA *polymerase* binds to start the DNA replication process. In next step, the DNA sample is divided into four parts for the different reactions with the modified nucleotides. For each reaction,



(a) The DNA component of the nucleosome (shown without the histone complex).



(b) The histone complex: H3 is colored in green, H2A in orange, H2B in yellow and H4 in purple (shown without the DNA). The histone tail of the histone H3 can be clearly apprehended.



(c) DNA and the histone complex form the nucleosome.

Figure 2.5: A nucleosome divided into its components. This model was generated from the crystal structure published by Luger et al. [8].

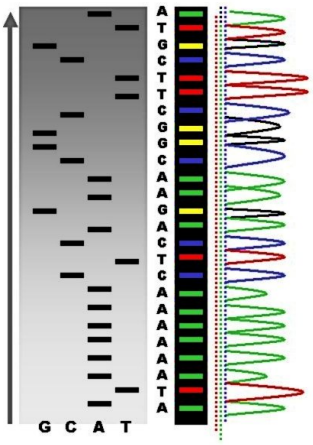



Figure 2.6: Sanger sequencing: Combination of the 4 lanes in one gel image, which reveal the sequence of the investigated segment compared to the fluorescent peaks caused by fluorescent terminators. [11]

four regular and one of the modified nucleotides are added. During replication, the DNA *polymerase* uses all nucleotides but in each reaction only one of four modified nucleotides is used, producing segments with different lengths of the new double stranded DNA. Afterwards, the DNA is again denatured and the segments are separated by length through gel electrophoresis. With the comparison of the four lanes of the gel image as shown in Figure 2.6, it is possible to build the sequence of this segment directly out of these image. It is also possible to use different fluorescent terminators for each type of nucleotide to read the gel image with a computer automatically.

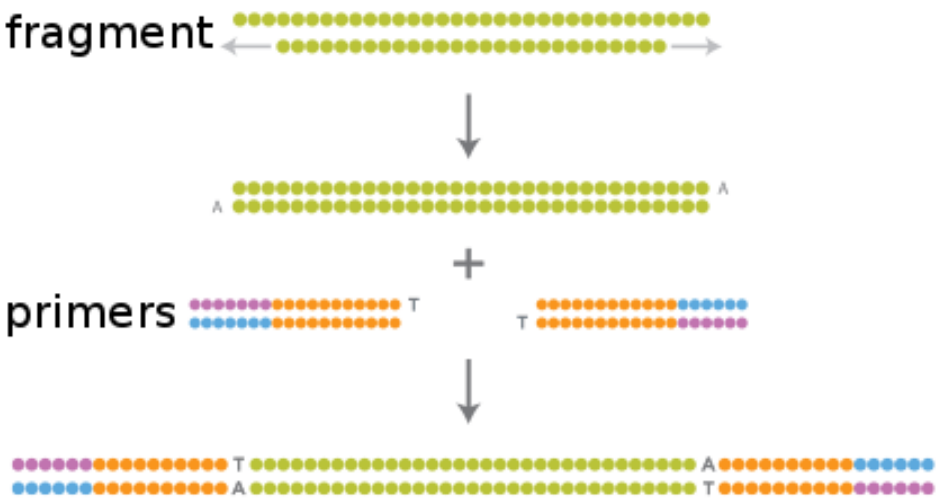
### 2.2.3 Next Generation Sequencing

Sanger sequencing is not well suited for whole genome sequencing for several reasons. For instance, a lot of random primers for binding all parts of the genome are required. Furthermore, it is time consuming and much more expensive. However, next generation sequencing methods make it easy and attractive to sequence whole genomes in less than a week as costs are as low as $1000 [12].

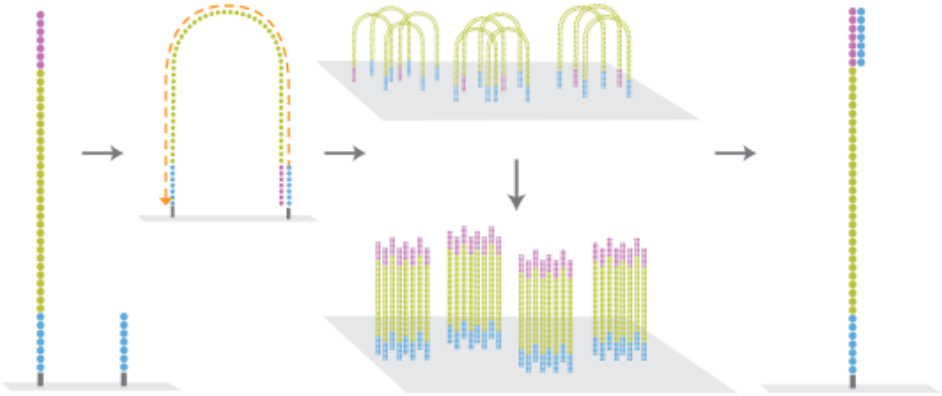
#### Illumina sequencing

The Illumina sequencing methods provide a fast way to sequence whole genomes since it is possible to sequence more than 10 billion nucleotides a day [13]. This method, like the Sanger sequencing, requires fragmented DNA. These fragments are ligated at both ends to the primers (Figure 2.7a), which are necessary

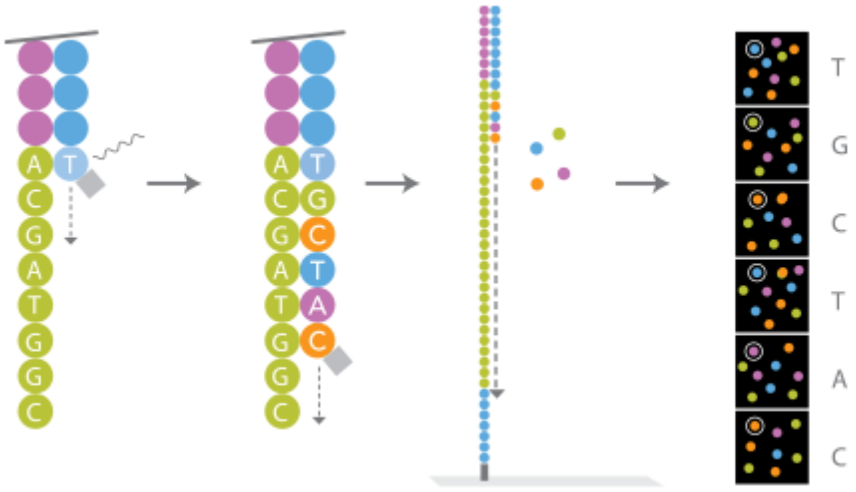
for the amplification step. During the amplification step, the fragments are put on a flow chip through binding to complementary primers. In this way, the DNA binds to the flow chip and can be replicated by a so-called bridge amplification. The flow chip is prepared with primers in abundance for the following replication step. The bound fragments form a "U" shape because both primers are hybridizing to primers on the chip (Figure 2.7b). After one step of replication, the DNA is denatured again and both strands form again a brigde on the chip for the next amplification round. After several rounds, the fragments are forming the so-called *DNA clusters* that contain only one specific fragment. Afterwards, the DNA is denatured again.
In the next step the sequencing is done with modified nucleotides that stop the replication. These nucleotides are fluorescently-labeled to red and green laser light. Since the *polymerase* stops after adding one modified nucleotide, it is possible to iterate over the whole fragments one by one. To recognize which nucleotide is added to a fragment, two lasers cause their specific fluorescent excitation. Each terminal nucleotide has a specific fluorescent behaviour and a camera detects the reaction for each *DNA cluster*. Since A and C are only excited by the red laser and G and T only by the green one, it is necessary to make four pictures of the flow chip each cycle. This pictures are taken with different filters, thus revealing the minimal varieties of spectra between A-C and G-T. Then, the termination element, which is bound to the nucleotide, is chemically washed away and the *polymerase* can again add a new nucleotide. After iterating over the whole length of the fragments, the sequence of each fragment on the flow chip is read. This method is very fast, since it can handle a huge amount of *DNA clusters* on one flow chip. Notwithstanding, it also produces more errors than the Sanger sequencing because an error during the amplification is exponentially distributed in the cluster. Additionally, the differentiation between the A-C and G-T spectra is still a problem and retained termination elements can mislead the results of the following sequencing cycle.




(a) Ligation of the primers to the DNA fragments



(b) Denatured DNA binds with the primers in a "U"-shape on the flow cell and is replicated. After several iterations of denaturation and replication the flow chip is covered with *DNA clusters*.




(c) A modified nucleotide (T) binds to a single DNA strand and emits a specific light spectrum after stimulation by the two lasers. Afterwards, the termination molecule (the gray rectangle) is washed away for the next sequencing cycle.

Figure 2.7: Schematic workflow of the Illumina sequencing [14].

#### ChIP-seq

The previously described methods for DNA sequencing are ”blindly” sequencing the DNA, since the only possibility to select specific parts of the DNA is to create a specific primer to target those pieces. For sequencing, for example, the parts of the DNA that are interacting with a protein, a pre-processing step is necessary for finding these parts of the DNA. One of the methods to do this is chromatin immunoprecipitation (ChIP) combined with DNA sequencing (seq). This method was mainly developed and used for the identification of binding sites for special proteins, so-called *transcription factors* [15]. Since histones are also proteins, it is possible that is wrapped around a histone complex with a particular modification.
Following the *ChIP-seq* protocol, the DNA is cross-linked with the proteins that are bound to the DNA. Then, the DNA is fragmented into pieces of 200-1000 bases by using sonication. Afterwards, specific antibodies are added that bind to their targets as shown in Figure 2.8. The fragments that are not associated with an antibody are washed away. Then, the DNA has to be purified again. Subsequently, the cross-linking is dissolved and the proteins are also washed away for the sequencing step. In the case of histone modifications, it is expected that a large amount of DNA fragments should pass the immunoprecipitation step since histones are spread over the whole genome. Therefore, it is necessary to use a next-generation sequencing methods such as Illumina to get fast results. To detect different histone modifications it is necessary to repeat these steps with a specific antibody for each modification separately and one cycle with an unspecific antibody or whole cell extract (WCE). The results gained from the unspecific antibody/WCE sequencing are used in a later step to normalize the data and reduce the noise, as not all fragments are equally well sequenced by Illumina. This is due to the following: the primers might not properly bind at some DNA fragments or the amplification process might not replicate the DNA fragments to an equal amount.

### 2.2.4 Data Set

The data used in this thesis was generated from three different cell types from mouse. These cell types are embryonic stem cells (ESC), murine embryonic fibroblasts (MEF), and neuronal progenitor cells (NPC). MEF and NPC arise from the ES cells during differentiation. Both have a common last ancestor that was derived from ESC. Mikkelsen et al. [16] used ChIP-seq to sequence three histone modifications at the histone H3. They investigated trimethylation

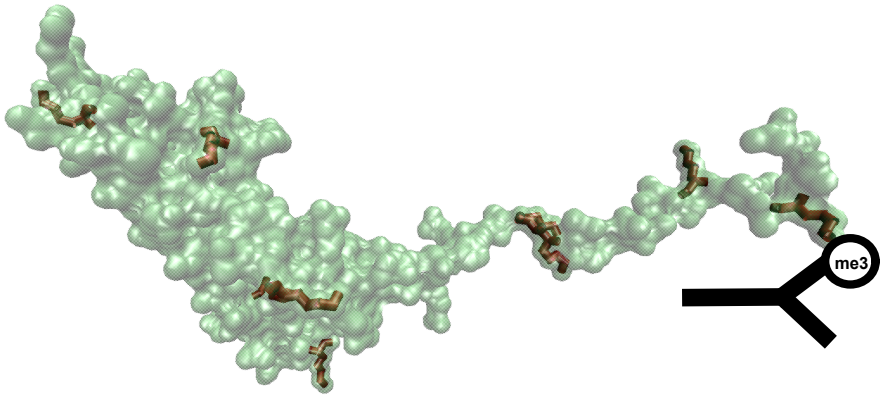



Figure 2.8: A H3 histone is bound to its specific antibody during the ChIP-Seq sequencing.

at three different positions of the amino acid lysine (biological abbreviation: K): K4, K27, and K9. The numbers represent the positions of the lysine on the histone. As they were only looking for trimethylation (me3) at the histone H3, the modification states are called H3K4me3, H3K27me3, and H3K9me3. H3K4me3 is positivly correlated with transcription [17], and it is found at tissue-specific genes and the so-called housekeeping genes. These genes are necessary for the viability of every cell and produce the transcripts for the basic cell functions. H3K4me3 is also frequently found in embryonic stem cells. H3K27me3 is related to heterochromatin, a very condensed part of the chromosome where the DNA is less accessible to RNA polymerase impairing transcription since the RNA polymerase cannot bind to the DNA [18]. H3K9me3 is a repressor of transcription and correlates with DNA methylation silencing the DNA that cannot consequently be transcribed [16]. In addition to the nine histone modification data sets, one whole cell extract was sequenced for each three cell types.

### Mapping

The result of DNA sequencing is just a huge set of short sequences (i.e. reads) derived from the original DNA. Therefore, it is necessary to map them back to the reference genomes. Several programs were developed to perform this task, speed up the mapping, and deal with differences in the sample reads and the reference genome. For mapping the data set to the mouse genome mm9, segemehl [19] was used for each ChIP-seq data set. As described by

Steiner et al. [20], for each modification data set and each position in the genome, the number of mapped reads was divided by the number of reads from the WCE of the corresponding cell type. This intensity score, called ”enrichment”, is a measurement for the significance of the peak with respect to the read distribution. An enrichment value of 3 was selected as a significant peak. Regions of continuous histone modification are identified by peaks of distance 100bp and smaller since it is not possible that another nucleosome can be present between these peaks. Afterwards, all peak regions smaller than 100 basepairs were treated as unmodified since the DNA wrapped around a nucleosome is 145-147 basepairs in length [21].

### Segmentation

In the next step, the data sets were segmented and merged to one data set for further processing. To compare the different cell types, the ESC was selected as reference for the segmentation process. As shown in Figure 2.9, all modification patterns were combined to a modification vector. Three modifications were studied. Therefore, $2^3$ combinatorial patterns are possible. All patterns with a segment length below 200 basepairs were discarded, since they are smaller than the estimated length of a nucleosome ( 150bp) and the indispensable linker DNA( 50bp). The MEF and NPC data sets were projected onto the reference ESC segmentation (shown in Figure 2.10 for MEF) and for each cell type, the overlap of the peak segments was calculated and represented with a value between 0 and 1.

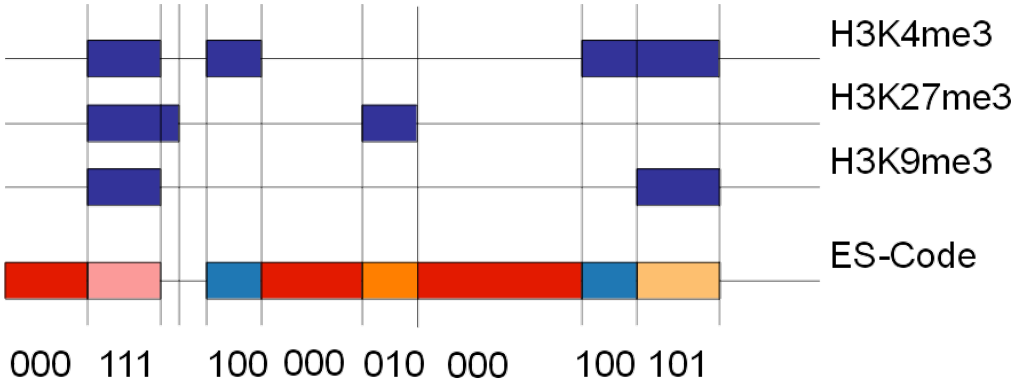



Figure 2.9: Example of the binary code segmentation in ESC. The ES-Code represents which type of modification pattern is present in the segment of the DNA.

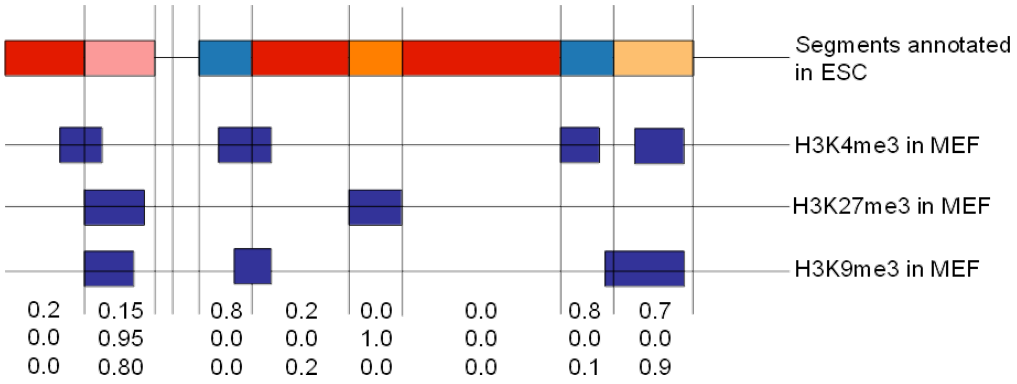



Figure 2.10: Example of code segmentation in MEF mapped to the ESC segmentation. The vectors represents which type of modification pattern is present in this segment relative to ESC.

# Chapter 3

# Related work

Beside *TiBi-SPLOM* another approach was tested to study the combinations of histone modifications: clustering. The initial implementation of clustering for this type of data was implemented by Sarah Seifert [22] during her bachelor thesis. This approach used the k-means++ algorithm for clustering the data set and visualized the results with starplots. However, this implementation was limited to eight clusters. Later on, I extended this implementation (called *ChromatinVis*) during my bachelor thesis [1]. The limitation of eight clusters was removed, so it was possible to test other clustering settings. Additionally, statistical tests for determining the best number of clusters were implemented. In addition to the starplots, scatterplots and tiled-binned scatterplots were implemented for visualizing the clustering results. The results had of course three dimension and principle component analysis was applied to reduce the dimension to two before visualizing them with the scatterplots. Building on this, Daniel Abitz added a consensus clustering during his bachelor thesis [23] to *ChromatinVis*. Using a consensus clustering, it is possible to improve and check the results of a clustering algorithm. The consensus clustering uses different initial setups (for example different numbers of clusters and seeds) and calculates a clustering for each of these setups. Afterwards, it merges the results into one consensus clustering that should result in a better separation between all clusters and increased stability of the found clusters. An improvement of the visualization techniques used by *ChromatinVis* was published by Gerighausen et al. in 2014 [24]. The starplots were discarded and replaced by so-called *windmill charts*. Nevertheless, the clustering approach was discarded, since it did not produce convincing results.

A *self-organizing map* (SOM) can be used also for visualizing high dimensional data. It clusters the data using a neuronal network and reduces thereby the dimensionality of the data set. Using SOMs for visualizing epigenetic data was

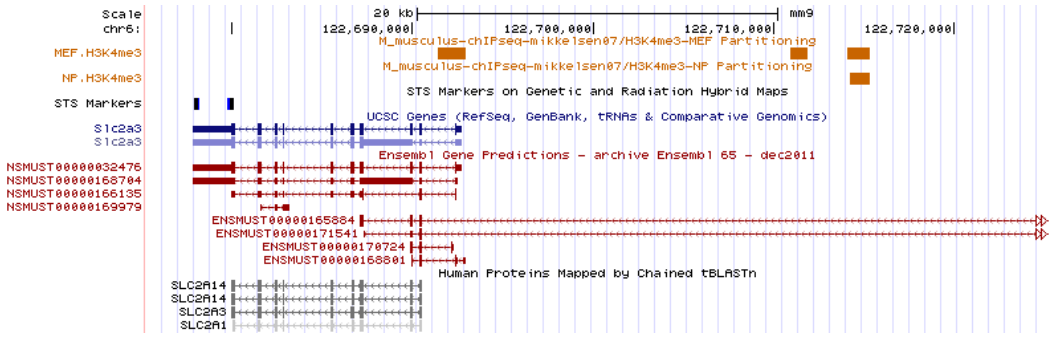



Figure 3.1: Segments with histone modifications annotated to the reference genome using a *genome browser*. It is possible to study the occurrences of histone modifications at a specific position in the genome.

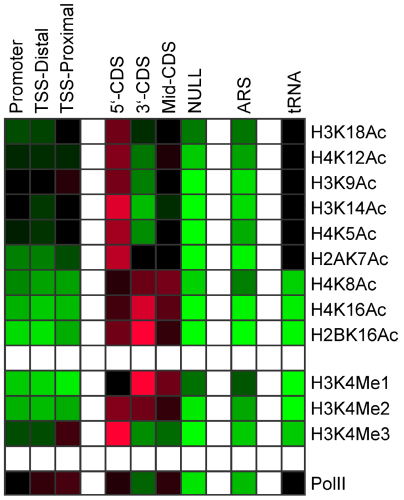



Figure 3.2: A *heatmap* visualizing the occurences of histone modifications at specific genomic locations. [26]

already published by Steiner et al. [25], but the results were hard to interpret and biased by the neighborhood relations in the neuronal network.

Other state of the art publications try to analyze epigenetic data sets mainly with two visualizations techniques: Using a *genome browser* or visualizing the data with *heatmaps*. The *genome browser method* (shown in Figure 3.1) plots the segments as so-called tracks linearly to their chromosome positions. The user is able to study the combination of histone modification at a specific position in the genome but cannot, for example, explore global trends of changes during cell differentiation. In contrast to this, the visualizations using *heatmaps* (shown in Figure 3.2 taken from Liu et al. [26]) try to show the occurences of histone modifications at specific genomic location like promotors or transcription start sites. Each cell in these *heatmaps* corresponds to one combination of a histone modification and genomic location. The color represents the average modification level of nucleosomes with this modification.

Further visualization techniques for high dimensional data were reviewed by Grinstein et al. [27]. However, all of them have issues when visualizing epigenetic data sets. *Parallel coordinates*, for example, use parallel axes for all dimensions of the data set. The range of all values for each axis is scaled to the lower and upper boundary of its axis. A polyline is drawn for each data point between the axes and the intersections mark the value of the data point in this dimension. Circular layouts of this method are called *polarcharts* or starplots. Other variants of this principle are *RadViz* and *PolyViz*. Although these techniques are able to present $n$ dimensions, it is hard to order them. Additionally, the effect of overplotting of the polylines makes it hard to impossible to interpret these visualizations. The starplots were a suitable representation for the clustering results, since the data set was normally clustered with less than 10 clusters and the effect was minimal to non-existent.

# Chapter 4

# Methods and Algorithms

## 4.1 Data Set

To properly handle the data, *TiBi-3D* requires a specific format. The data format is similar to a *CSV* text file, where each line after the header is considered a data item. The header is defined in two lines, as follows:

- !Data

- shortid;segment;code;variable 1;....;variable n;segment length

The body of the file contains these fields filled with values from the segmentation of the histones.

- shortid: the unique id for this item
- segment: the chromosome position for the segment
- code: the calculated histone code (see Table 4.1 for further details)
- variable 1 - (n-1): a value between 0 and 1 that represents the correspondence of the histone modification for this segment between the cell types
- variable n: a value between 0 and 1 that represents the CpG-density for this segment
- segment length: an integer value that represents the number of nucleotids in this segment

The code is calculated as described in Section 2.2.4 and transformed to an integer value. For example, in Table 4.1 is shown how the codes for the histone *H3K4me3*,*H3K27me3*, and *H3K9me3* are calculated.

Table 4.1: Example for the code calculation

| Code | H3K4me3 | H3K27me3 | H3K9me3 |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 |
| 7 | 1 | 1 | 1 |

## 4.2 BED Format

The BED format [28] (Browser Extensible Data) is a common file format to store annotations for sequences related to their chromosome position. It is tab separated and each line represents one annotation. This format is used by *TiBi-3D* to export the results for further processing using genome browsers or other programs. Three fields are required:

1. chromosome or scaffold name
2. the starting position of the annotation
3. the ending position of the annotation

The other fields are described as optional, but BED interpreters need each field before the selected field. These are the fields used by *TiBi-3D*:

4. name of the sequence
5. a score between 0 and 1000
6. The strand for the annotation indicated by + and -
7. starting position of the thick line (important for the visualization in the Genome Browser)
8. ending position of the thick line
9. RGB color of the item

## 4.3 Binned Scatter Plots

A scatterplot visualizes the distribution of a data set related to two selected variables. These variables are treated as coordinates in a Cartesian coordinate system and are drawn related to the two axes. With this type of visualization, it is possible to easily recognize the distribution of a data set and the correlation between the two variables. By adding a third axis, the scatterplot can be used to represent the relationship between three variables of a given data set.
As shown in figure 5.2, the 3D scatterplot has a huge overlapping zone around the point of origin. To circumvent this, the idea of the tiled binned scatterplots, proposed by Zeckzer et al.[2] was adapted to 3D. Originally, the data set was divided into equal rectangles in 2D, but now it is possible to divide into equal cubes, called *bins*. Additionally, each of these bins is also divided again into smaller cubes, called *tiles*. Each tile represents one code of the data set, and the amount of data-points that are lying within this bin are represented by a sphere inside this tile. Each sphere has a specific color and it's transparency represents how many data-points of specific code are within this bin. To avoid problems with dyschromatopsia, the colors were taken from the color brewer project [29]. Each tile has a length of $\frac{1}{2}$ of the bin and each sphere has a radius of $\frac{1}{8}$ compared to length of the bin. To avoid occlusion caused by the spheres, each sphere is placed close to the center of its containing bin. Further details of this placement procedure is described in the Algorithm 2, where the position of the sphere is determined for each tile. This binning method uses maximally eight spheres for each bin, which can not overlap, so it has a huge impact on the occlusion of the whole plot. For the binning algorithm (Algorithm 1) the range of values for each axis has to be transform between 1 and the number of bins in the plot, similarly to Equation (Equation (4.1)). If the data is distributed logarithmically, it is possible to scale it first by using the logarithmic transformation 4.3 before applying the binning. If the data is distributed between 0 and a maximum $\leq 1$ the scaling will fail (logarithmic functions start to raise to infinity around zero), therefore, each value is increased by one before normalizing it.

$$normalize(value, data_{max}, data_{min}, norm_{max}, norm_{min}) = \tag{4.1}$$

$$\frac{value - data_{min}}{data_{max} - data_{min}} * (norm_{max} - norm_{min}) + norm_{min} \tag{4.2}$$

Table 4.2: ESC code (binary) or length (logarithmic scale) to map the colors for the TiBi-3D scatterplot. Each row specifies the ESC code, the modifications in ESC, the length, the color, its name, and its RGB values.

| **Code** | **Modifications** | **Length** | **Color** | **Color Name** | **RGB** |
|---|---|---|---|---|---|
| 000 | none | 200-434 | | red | 227, 26, 28 |
| 001 | H3K9me3 | 435-940 | | bright green | 178, 223, 138 |
| 010 | H3K27me3 | 941-2032 | | orange | 255, 127, 0 |
| 011 | H3K27me3, H3K9me3 | 2033-4396 | | bright blue | 166, 206, 227 |
| 100 | H3K4me3 | 4397-9506 | | blue | 31, 120, 180 |
| 101 | H3K4me3, H3K9me3 | 9507-20559 | | bright orange | 253, 191, 111 |
| 110 | H3K4me3, H3K27me3 | 20560-44460 | | green | 51, 160, 44 |
| 111 | H3K4me3, H3K27me3, H3K9me3 | 44461-7000272 | | rose | 251, 154, 153 |

$$logtransform(value, max_{data}) = \frac{log\#(value)}{log\#(max_{data})} \tag{4.3}$$

**Algorithm 1** Bin the normalized data into the scatterplot

**function** CALCULATEBIN(element, dataset)
x = Math.floor(normalize(element.x, $dataset.x_{max}$, $dataset.x_{min}$, bins, 0.0);
y = Math.floor(normalize(element.y, $dataset.y_{max}$, $dataset.y_{min}$, bins, 0.0);
z = Math.floor(normalize(element.z, $dataset.z_{max}$, $dataset.z_{min}$, bins, 0.0);
updateBin(x,y,z, element.code)
**end function**

**Algorithm 2** Determine the position of the sphere according to its containing tile

```
function setPosition( x, y, z, code)
    Point3d position = new Point3d();
    switch code do
        case 0                                                    ▷ Code 000
            position.setX(x + 0.30);
            position.setY(y + 0.30);
            position.setZ(z + 0.30);
        case 1                                                    ▷ Code 001
            position.setX(x + 0.30);
            position.setY(y + 0.30);
            position.setZ(z + 0.70);
        case 2                                                    ▷ Code 010
            position.setX(x + 0.30);
            position.setY(y + 0.70);
            position.setZ(z + 0.30);
        case 3                                                    ▷ Code 011
            position.setX(x + 0.30);
            position.setY(y + 0.70);
            position.setZ(z + 0.70);
        case 4                                                    ▷ Code 100
            position.setX(x + 0.70);
            position.setY(y + 0.30);
            position.setZ(z + 0.30);
        case 5                                                    ▷ Code 101
            position.setX(x + 0.70);
            position.setY(y + 0.30);
            position.setZ(z + 0.70);
        case 6                                                    ▷ Code 110
            position.setX(x + 0.70);
            position.setY(y + 0.70);
            position.setZ(z + 0.30);
        case 7                                                    ▷ Code 111
            position.setX(x + 0.70);
            position.setY(y + 0.70);
            position.setZ(z + 0.70);
    this.position = position   ▷ Overwrite position of the sphere in the class
end function
```

## 4.4 Range normalization

The data used in this thesis has a distribution that is extremely heterogeneous, therefore, the range normalization using Equation (4.1) is not appropriate because of the influence of the outliers. An outlier is described as "an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism" [30]. For this reason, it is better to detect first the outliers and reduce the interval of $[min_{data}; max_{data}]$ to an interval without these outliers.
Zeckzer et al. [2] tried to transform the amount of data points in each bin with a logarithmic scale (Equation (4.3) after binning it. Due to the effects of occlusion in 3D, this transformation alone does not produce considerable results, because even in the logarithmic space the outliers are influencing the visualizations too much. The implemented outlier detection in *TiBi-3D* is adapted from the computation of *box plots* [31]. The *box plot* shows the median as line in a box between the lower and upper quartile. This box is extended by the so called *whisker*, whose ends represent the lowest and highest value within the 1.5 interquartile range (IQR). The IQR is the distance between the lower and the upper quartile. Any data above and below these whiskers is treated as an outlier.
After binning the data set as shown in Algorithm (1) and transforming each bin with Equation (4.3), *TiBi-3D* calculates the upper and lower whisker using Algorithm (3) for each modification patter.
After this outlier detection, *TiBi-3D* calculates, for each tile, the transparency value that represents the amount of elements that are located in the containing bin related to the bin with the most elements of this specific code. Since the transparency values of the spheres in the plot have an interval between 0 and 1, the transparency values are normalized again to the $[0, 1]$ interval (like described in the algorithm 4). The detected outliers are mapped to the lower or the upper whisker.

**Algorithm 3** Implemented box plot outlier detection in *TiBi-3D*.

```
for x = 0; x < bins; x++ do
    for y = 0; y < bins; y++ do
        for z = 0; z < bins; z++ do
            for code = 0; code < #patterns; code++ do
                bin = bins[x][y][z];
                logvalue = logtransform(bin.getElements(code), maxEle-
ments(code));
                boxplots.get(code).add(logvalue);
            end for
        end for
    end for
end for
for all i : boxplots do
    sort(i);
    median = i.length()/2;
    quartile = median/2;
    upperQuartile = i[median + quartile];
    lowerQuartile = i[median - quartile];
    IQR = upperQuartile - lowerQuartile;
    lowerWhisker = lowerQuartile - (IQR * 1.5);
    upperWhisker = upperQuartile + (IQR * 1.5);
end for
```

**Algorithm 4** Calculate the transparency for each tile

```
function calculateTransparency(elements(code), maxElements(code))
    transparency = logTransform(elements, maxElements(code));
    if transparency < lowerWhisker then
        transparency = lowerWhisker;
    end if
    if transparency > upperWhisker then
        transparency = upperWhisker;
    end if
    localmin = lowerWhisker;
    if localmin < 0.0 then
        localmin = 0.0;
    end if
    return normalize(transparency, upperWhisker, localmin,1.0,0.0);
end function
```

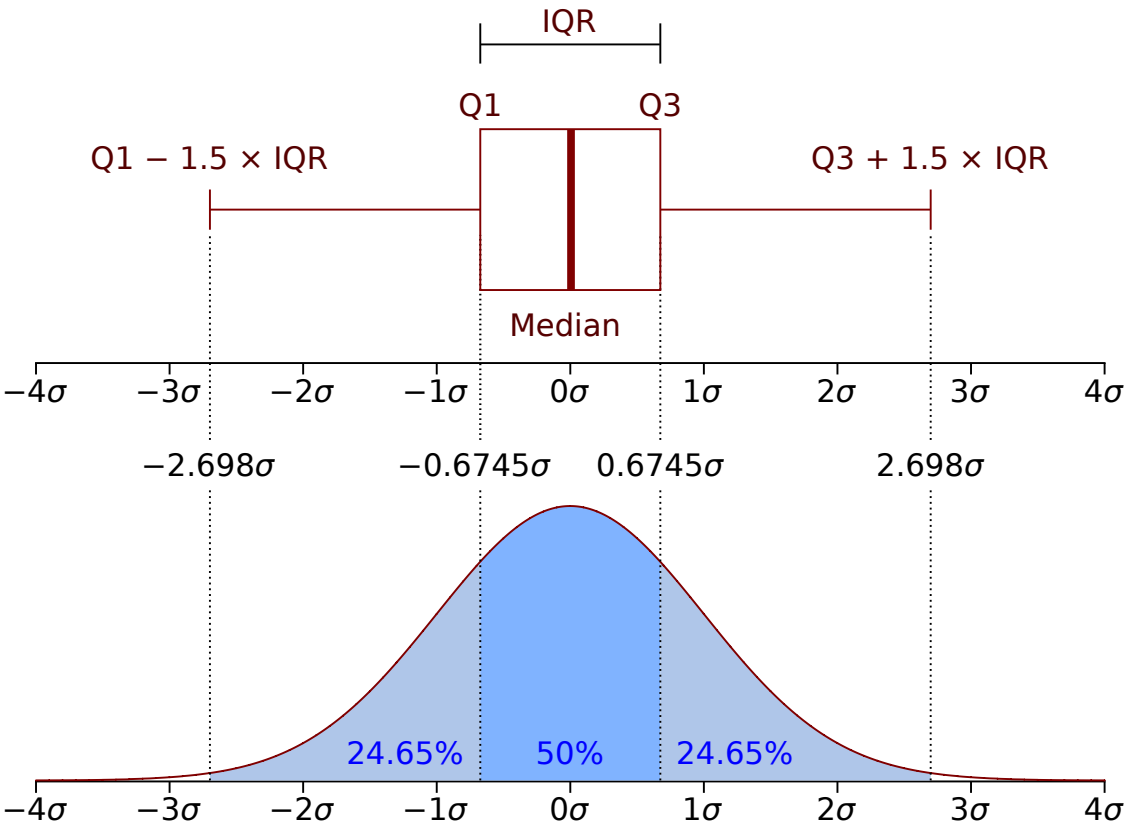



Figure 4.1: A box plot applied on normally distributed data set [32]

## 4.5 Calculation of the right contrast for the background

As the number of elements in a bin is described by its own transparency and the user has the possibility to investigate each bin by choosing it, a detailed information panel was implemented to show the content of the chosen bin. This Panel (see Figure 4.3) shows each possible combination of histone modifications and in the second column, the amount of elements of each of them. The color in the second column corresponds to the color that is used in the bin. For a better perception depending on the background color, the color of the text in these text fields has to change.

*TiBi-3D* uses black and white text color and calculates which color has a greater contrast to the background. For doing this, it transforms the *RGBa* color of the bin into a *RGB* color by implementing the *Alpha Blending* algorithm [33].

With this algorithm, it is possible to combine two transparent images into one:

$$C = \frac{1}{\alpha_C} * (\alpha_A A + (1 - \alpha_A)\alpha_B B) \tag{4.4}$$

where $\alpha_C$ is:

$$\alpha_C = \alpha_A + (1 - \alpha_A)\alpha_B \tag{4.5}$$

$C$ corresponds to the new color for the two combined colors $A$ and $B$, where A is merged over B and $\alpha$ is their alpha value. Since *TiBi-3D* has a complete opaque background, Equation (4.4) is less complex because it is not necessary to calculate $\alpha_C$:

$$C = \alpha_A A + (1 - \alpha_A)B \tag{4.6}$$

With this new color $C$ it is possible to determine if white or black text has a greater contrast to the background.
To do this, another transformation of the color to the *YIQ* [34] colorspace is necessary. The *YIQ* colorspace is an old colorspace normally used by the *National Television System Committee* (NTSC) to encode colored pictures for broadcasting. The *RGB* colorspace is transformed into the *YIQ* colorspace with the transformation matrix:

$$\begin{vmatrix} Y \\ I \\ Q \end{vmatrix} = \begin{vmatrix} 0.299 & 0.587 & 0.114 \\ 0.595716 & -0.274453 & -0.321263 \\ 0.211456 & -0.522591 & 0.311135 \end{vmatrix} * \begin{vmatrix} R \\ G \\ B \end{vmatrix} \tag{4.7}$$

The $Y$ component represents the luminescence of the color, $I$ the difference between cyan and orange and $Q$ the difference between magenta and green. An example of this colorspace is shown in Figure 4.2. For calculating the better foreground color $F$ (black or white), only the $Y$ component is required:

$$Y = \begin{vmatrix} 0.299 & 0.587 & 0.114 \end{vmatrix} * \begin{vmatrix} R \\ G \\ B \end{vmatrix} \tag{4.8}$$

$$F = \begin{cases} white & \text{if } Y < 0.5 \\ black & \text{if } Y \geq 0.5 \end{cases} \tag{4.9}$$

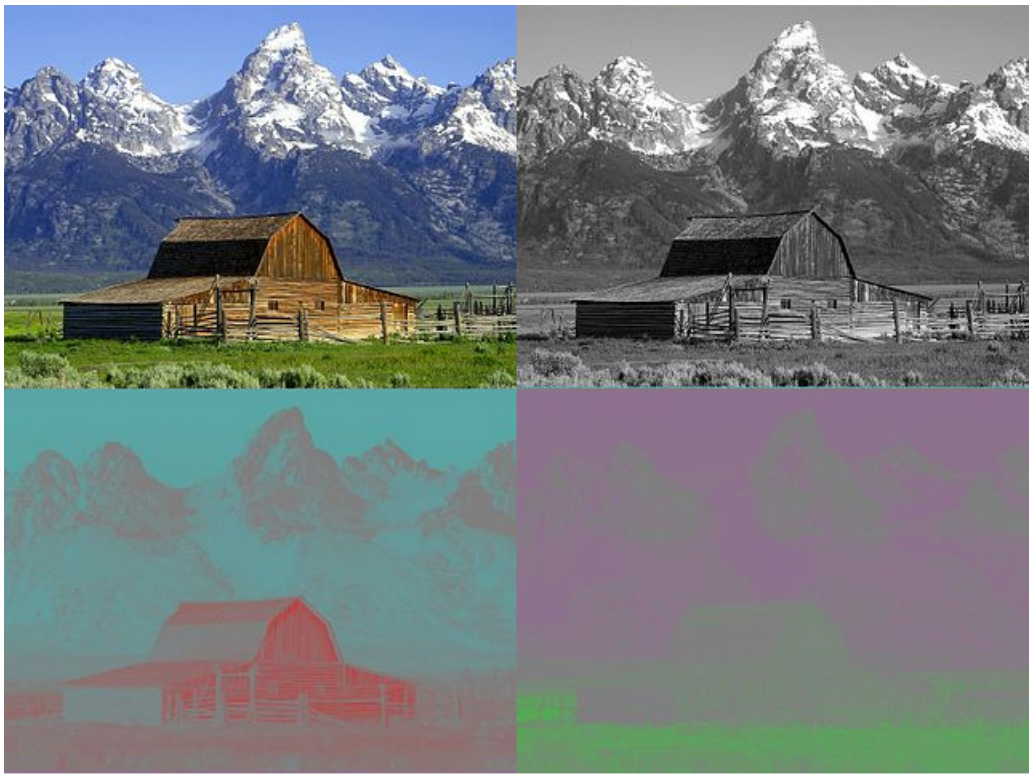


Figure 4.2: A picture [35] (top left) divided into Y (top right), I (bottom left), and Q (bottom right).

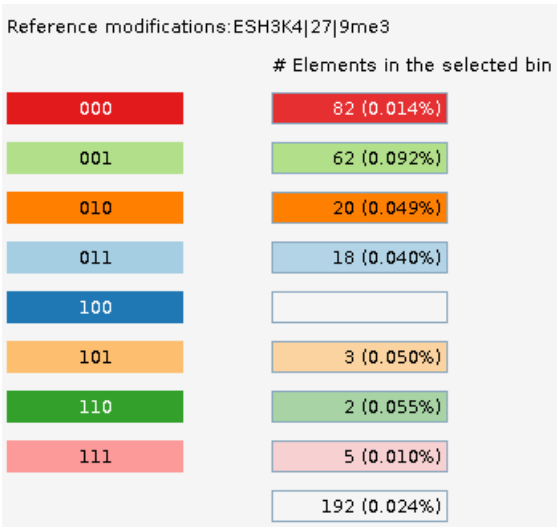



Figure 4.3: Example for the correct contrast and alpha blending in *TiBi-3D*

Figure 4.3 shows how these methods influence the foreground color related to their background color for a better recognition.

## 4.6 Java3D

As explained in Section 4.3, *TiBi-3D* visualizes a given data set with a 3D tiled binned scatterplot. Java is not able to handle 3D components in the viewing interface out of the box. For this special application, an API called Java3D for using OpenGL or Direct3D with Java was developed [30]. Its components are embedded in the normal GUI of a Java program like in TiBi-3D or shown in another window. With this API it is possible to easily model and render 3D objects in a Java program, hence *Java3D* is encapsulating all *OpenGL* functions. *Java3D* stores the whole 3D scene in a *scene graph* that has two main branches:

- Viewing branch
- Content branch

Unlike the *OpenGL* API, *Java3D* stores a camera position (called *ViewingPlatform*) and can transform it. This information is stored in the viewing branch and is used by the perspective saver of *TiBi-3D*. The objects, which are drawn in the scene, are stored in the content branch. Before rendering the scene, *Java3D* optimizes this *scene graph*, therefore, the access to these objects in the scene is restricted via capabilities after the rendering. Since this rendering improves the memory consumption of the scenegraph, it used by *TiBi-3D* to improve its runtime.

# Chapter 5

# Results

## 5.1 Visualization

### 5.1.1 2D vs 3D

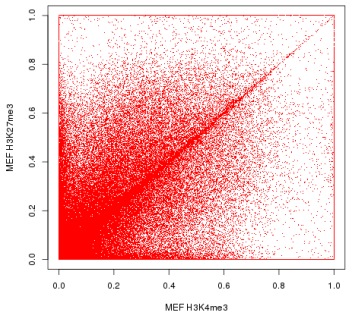
The visualization of information with scatterplots has in general to deal with the effects of overplotting. With a big data set that is not homogeneously distributed, the data points in the scatterplot are overplotted, as shown in Figures 5.1. A lot of information gets lost during the drawing process since each pixel in the visualization can only show one color information. To circumvent this effects, *TiBi-SPLOM* [2] used tiled binned scatterplots in 2D. For example, Figure 5.3 shows the scatterplot for MEF H3K4me3/H3K27me3. However, since some results were already published with this 2D visualization technique, it would be helpful to analyze the relationship of all three histone modifications simultaneously contained in the data set. The three different modifications are influencing each other, therefore, a two dimensional exploration of them does not reveal all information.

Earlier visualizations (Figure 5.2a) with scatterplots in 3D of this data set had to deal with the effects of overplotting like in 2D, as well. However, this visualization already revealed some relations between all three modifications, for instance, a change of modifications between 000 to 111. This result is not easily observable in 2D since the 2D projections are merging the hidden third dimension. Compared to Figure 5.1, there is also a diagonal line between (0,0) and (1,1), but this line shows also the merged diagonal line from (0,0,0) to (1,1,1) in 3D. To explore the relation of all three modifications or just two modifications and the influence of the segment length or the CpG-density, a co-analysis of multiple scatterplots is required in *TiBi-SPLOM*. Without prior knowledge, some of the results described in Section 5.2 were not clearly

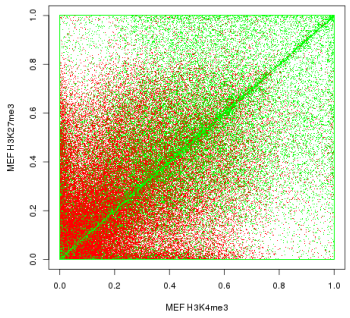
observable. Even data brushing cannot reveal the results of *TiBi-3D* (an example is Figure 5.19).
As described in Section 4.3, the principles of tiled binned scatterplots were adapted to 3D scatterplots to solve the previously described drawbacks of *TiBi-SPLOM*. Unfortunately, 3D visualization has to deal with new difficulties like occlusion and the correct point of view for gaining interesting results. Therefore, interaction with the plot is necessary, for instance, rotating and zooming. Additional features of *TiBi-3D* are described in Section 5.1.4. Occlusion is one of the biggest flaws of 3D visualizations. Thus, *TiBi-3D* tries to reduce it automatically with the normalization process described in Section 4.4 and several filtering options. Compared to *TiBi-SPLOM*, *TiBi-3D* consumes more memory while running, but as shown in Section 5.1.3 it needs approximately 1 GB of memory during the calculation of the scatterplot. However, the benefits of this 3D visualization outweigh the disadvantages of memory consumption, interaction effort by the user, and the 3D specific drawback of the occlusion.

### 5.1.2 Design alternatives

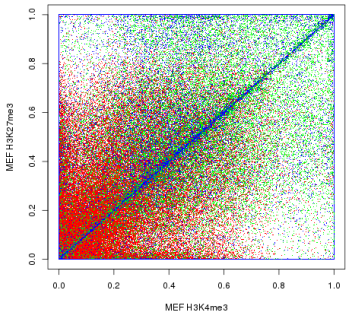
*TiBi-3D* was designed by adapting the 2D tiled-binned scatterplots used by *TiBi-SPLOM*. Nevertheless, the design of the binning was slightly changed to answer new requirements due to adding a third dimension. Therefore, the cube design of each tile was changed to a sphere since a filled cube would hide the tiles behind it in the plot.
The encoding of histone modification patterns could be also implemented by using *glyphs*. Nevertheless, *glyphs* must also represent the amount of points in the tile. This could be done by changing the size of the *glyphs* but it would be difficult to perceive the small *glyphs* in the scatterplot. Moreover, the perception of the contrast between all colors of the tiles is stronger than the perception of different *glyphs*. For that reason, the double encoded histone modification pattern (categorical color and position in the bin) in the spheres provide a better solution. Colin Ware also suggested to use colors for categorical data with less then 10 categories [36].
The color coding is taken from to the color brewer project [29]. One possible alternative was the color coding with binary RGB values for each modification. The pattern 100, for example, would be encoded with the RGB value (1,0,0). The RGB values for the pattern 111 and 000 would be (1,1,1) and (0,0,0). These values represent the colors white and black, respectively. However, also the color of the background of the scatterplot is white. Choosing a different



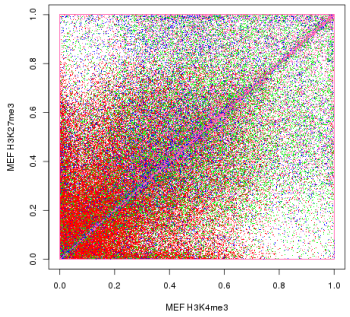

(a) 000



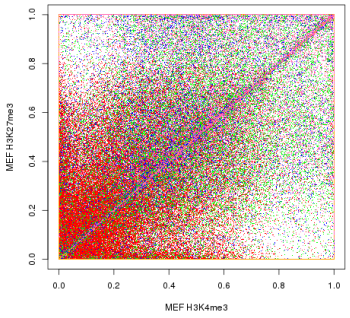

(b) 000, 001



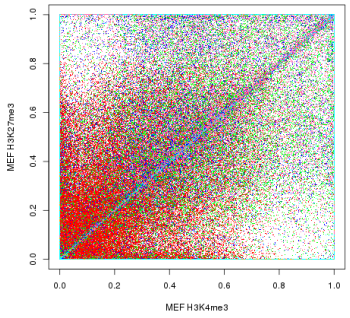

(c) 000, 001, 010



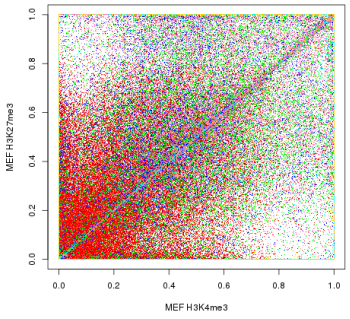

(d) 000, 001, 010, 011



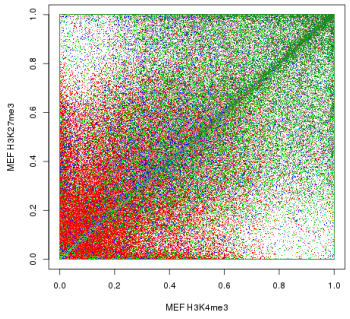

(e) 000, 001, 010, 011, 100




(f) 000, 001, 010, 011, 100, 101

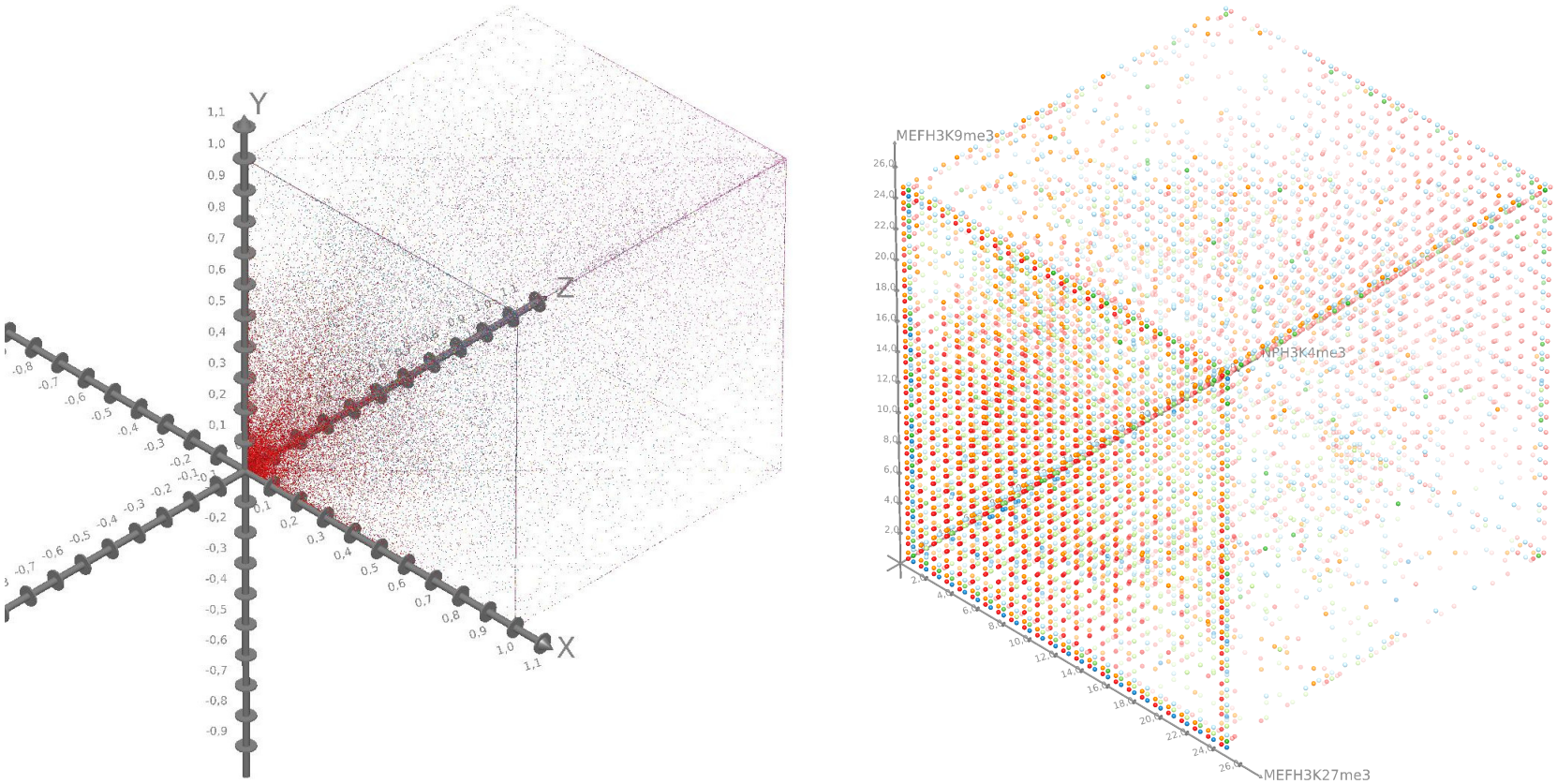



(g) 000, 001, 010, 011, 100, 101, 110




(h) 000, 001, 010, 011, 100, 101, 110, 111

Figure 5.1: The effect of overplotting of the different codes in a 2D scatterplot using the data set by Mikkelsen et al. [16]




(a) A normal 3D scatterplot

(b) A tiled-binned scatterplot from *TiBi-3D* with 25 bins

Figure 5.2: Comparision of a normal and tiled-binned 3D scatterplot

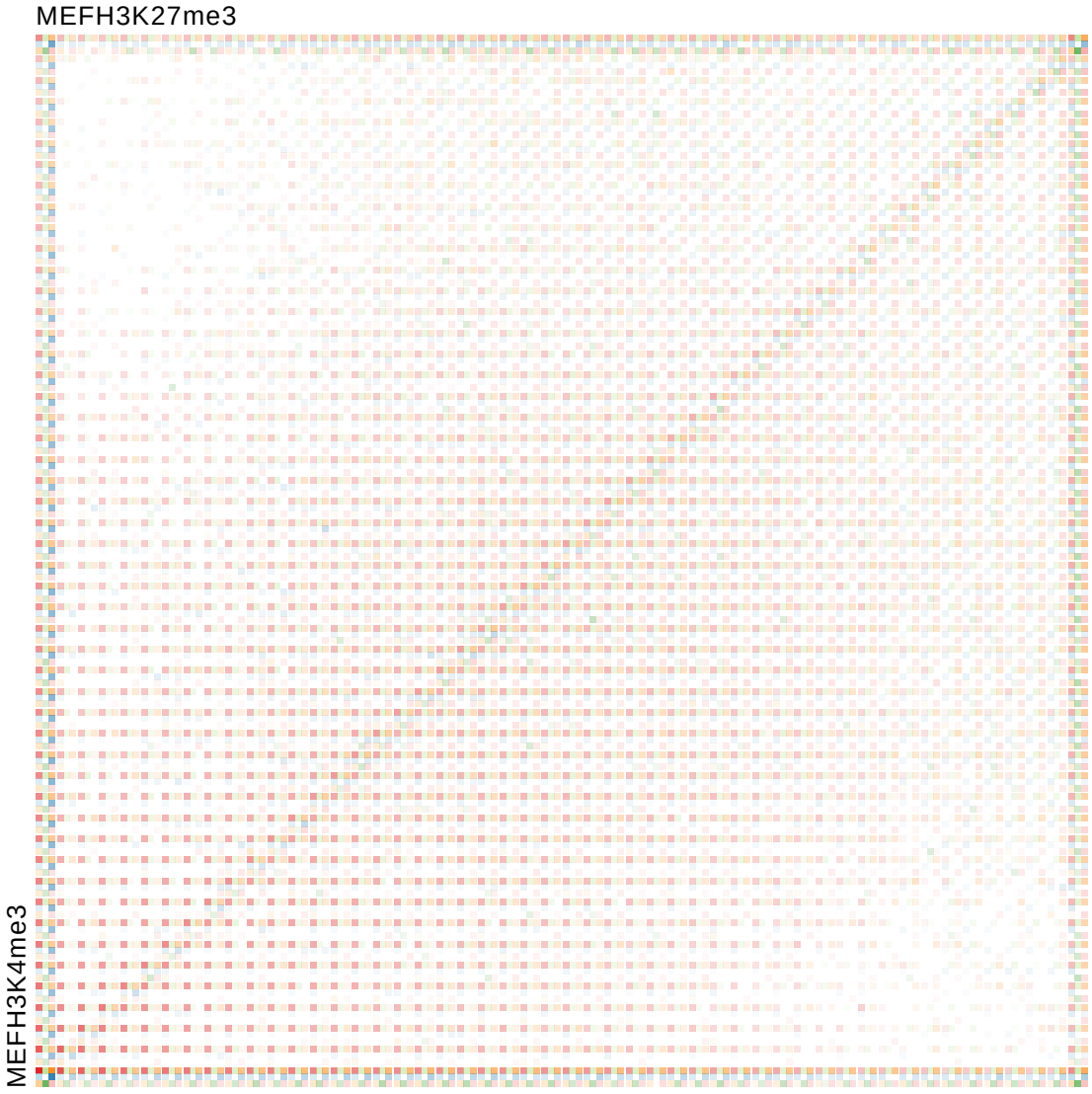



Figure 5.3: A 2D binned scatterplot for MEFH3K4me3 and MEFH3K27me3

background color with a strong contrast to all colors used in the scatterplot is a difficult task. Using a different color scheme for representing these patterns is not intuitive and would interfere with the other colors in the plot.
To encode the size of each bin, a saturation value could also be used instead of transparency. While the transparency values decrease the effects of occlusion, in the scatterplot, but the use of saturation values would not provide this improvement. Therefore, the transparency is preferred over the saturation encoding.
Other design alternatives for representing high dimensional data, in particular epigenetic data sets, are discussed in the Chapter 3.

### 5.1.3 Runtime and memory analysis

*TiBi-3D* is designed as an exploratory software to analyze data with the interaction of the users. Therefore, it should run on a normal computer and provide results in a short amount of computational time. Like described in Section 1 and Section 3, *TiBi-3D* bins the data and generates the statistics for the normalization. The binning process uses $O(n)$ time where $n$ is the number of data items and the normalization works in $O(m)$ where $m$ represents the amount of bins in the scatterplot.

For analyzing the runtime and the memory consumption of *TiBi-3D*, the profiling module of NetBeans [37] was used. This analysis was done with a computer with the following specifications:

- Intel i5-4570 Quadcore CPU with 3.20 GHz
- 8GB RAM
- Nvidia GeForce GTX 650 with 1024MB VRAM

The profiling module plots the different states of the java threads in Figure 5.5. When *TiBi-3D* loads a file, it uses the API foxtrot [38] to prevent that the Swing API of java freezes while the whole computation is done. Since the loading of a file invokes also the binning process and rendering in Java3D, it is possible to determine the required computational time. In Figure 5.5, the foxtrot thread was running for 13.611 ms, thus, *TiBi-3D* needs around 13 seconds on the testing machine to generate the plot. During these computations, it allocates around 2028MB heap space (*TiBi-3D* was started with a maximum allowed heap space of 6GB) and uses a maximum of about 1062MB. The time

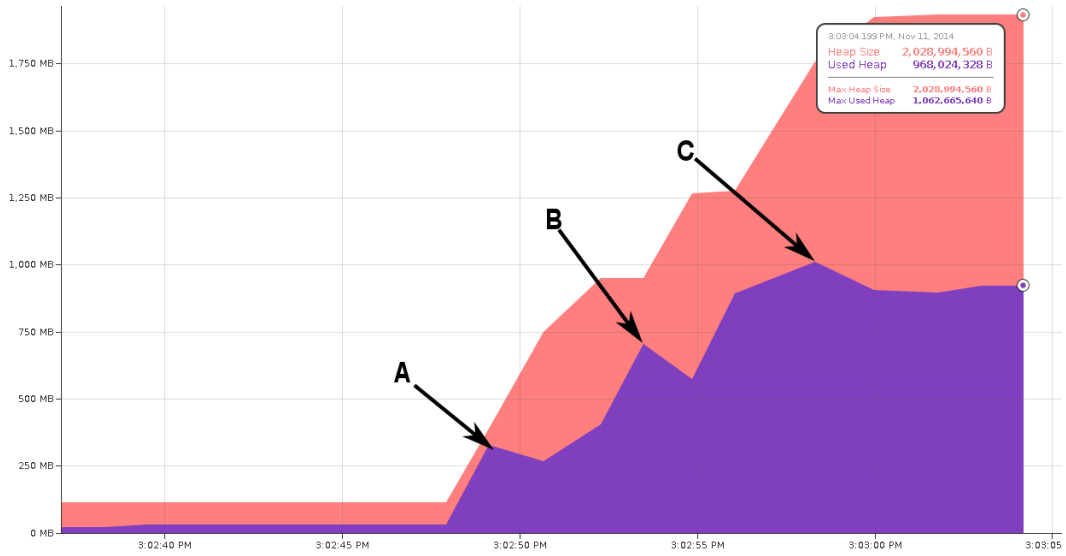



Figure 5.4: Memory consumption of *TiBi-3D* while loading and binning a data set, and generating the Java3D scenegraph.

plot of the memory consumption is shown in Figure 5.4. At label 'A' the loading of the input file is done, at label 'B' the binning process is finished, and at the label 'C' scencegraph is compiled and ready. After each of these stages, the garbage collection of java was invoked manually. Compared to [1], the memory consumption was less, although *TiBi-3D* stores the spheres as a mesh of triangles.

### 5.1.4 Features for Exploration and Interaction

*TiBi-3D* has several features for exploring and manipulating data sets. The user can use the mouse of its computer to interact with the scatterplot thus being able to move, rotate, and zoom in it. *TiBi-3D* allows to rotate the plot automatically and can snap the plot to specific angles. Furthermore, it is possible to switch between parallel and perspective projections. The following features were implemented, while modifying the former 3D scatterplot from Dirk Zeckzer to a 3D binned scatterplot.

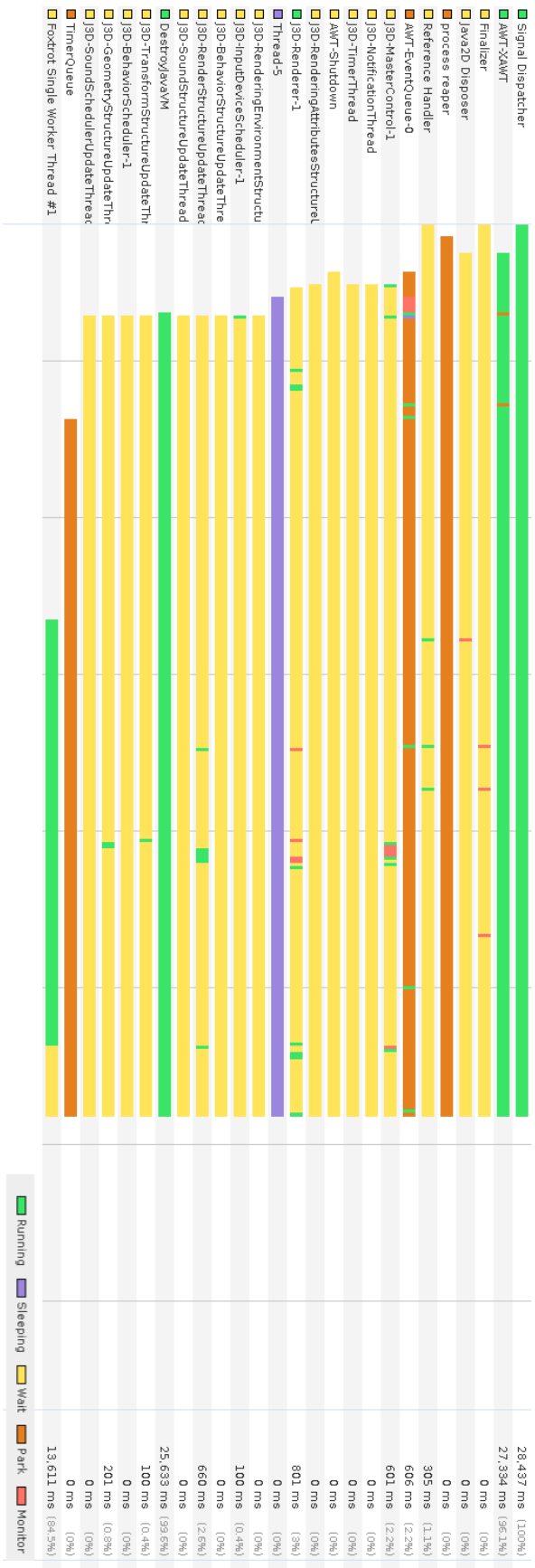



Figure 5.5: Runtime analysis of *TiBi-3D* per thread while loading a data set, binning it, and generating the Java3D scenegraph.

### Filtering

Different filtering options are implemented in *TiBi-3D* for analyzing the scatterplot more easily.

### Cutting planes

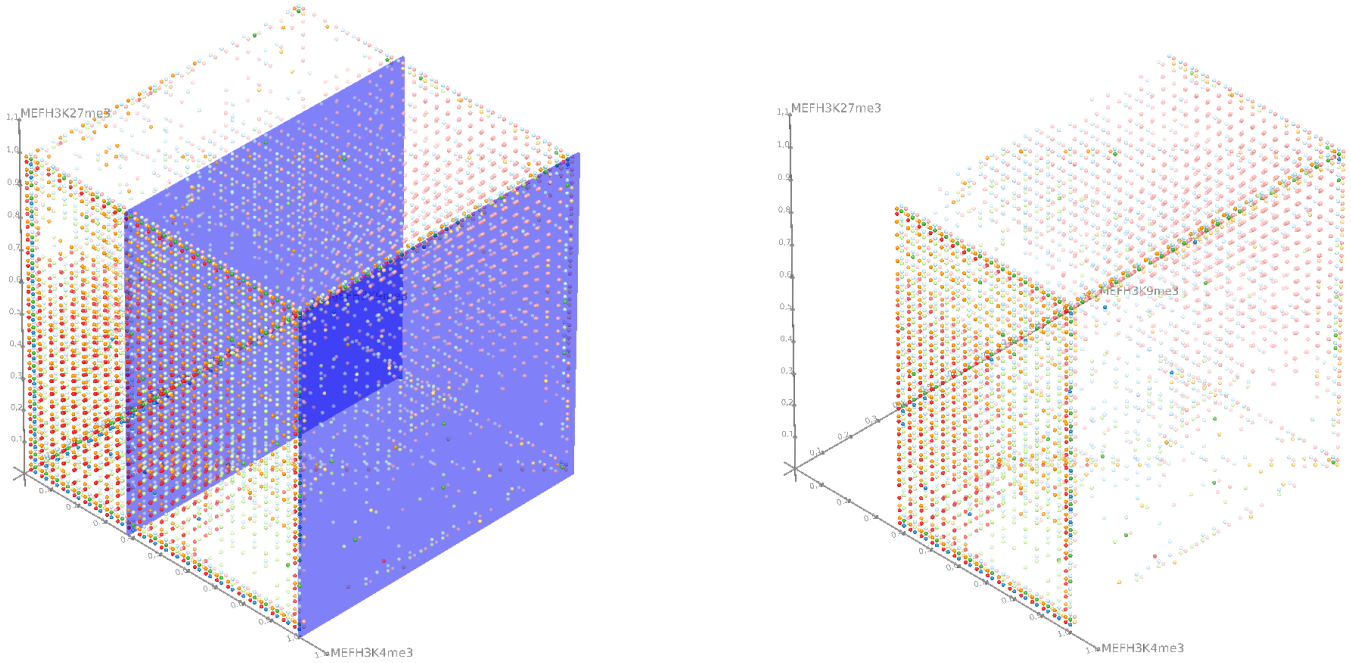
Cutting planes [39] are used to divide a 3 dimensional space into two parts with a plane. In *TiBi-3D* the user is able to divide the drawing space and filter the data set. With these planes, it is also possible to define the minimum and the maximum value of the drawing interval on each axis so that just the values within these planes are drawn by *TiBi-3D*. For selecting the interval to show, a slider with knobs for each axis is provided. The interval is shown with two blue planes in the scatterplot while adjusting the interval. After the user releases the slider, the binned scatterplot will be redrawn with the selected interval. An example of this filtering method is shown in Figure 5.6.

### Color filtering

The user can also exclude modification patterns from the scatterplot by selecting the corresponding check box like shown in Figure 5.14. As default, all patterns are selected for drawing. The unselected patterns are then excluded during the next redrawing, thus the scatterplot shows just the patterns of interest. One advantage of this filtering is it makes possible to exclude unnecessary or uninteresting patterns for a specific task. This filtering also reduces the occlusion in the scatterplot like in Figure 5.7.

### Transparency cutoff

*TiBi-3D* includes a transparency cutoff feature. This allows to show the more significant bins. The transparency of each bin is calculated using the same normalization strategy as described in 4.4. It has values between 0 and 1. Thus, it is possible to filter the spheres of each bin and exclude them from the scatterplot during the drawing process. The user can select the cutoff with a slider in *TiBi-3D* and the scatterplot will be directly redrawn.
Using the transparency cutoff also allows to reduce the clutter of the visualization and to show the spheres which contain a a minimum amount of data points. Therefore, it is possible to detect whether the amount of the modification patterns are changing between the different cell types under analysis. The different levels of the cutoff for MEFH3K4me3, MEFH3K27me3 and MEFH3K9me3 are shown in Figure 5.8. With a cutoff of 100% as shown in Sub-Figure 5.8k, only the outliers of each pattern are shown in the scatterplot.




(a) Selecting the interval on the MEFH3K4me3 axis using cutting planes

(b) The selected interval is redrawn by *TiBi-3D*

Figure 5.6: With the cutting planes it is easy to select the drawing space of the scatterplot for each axis.

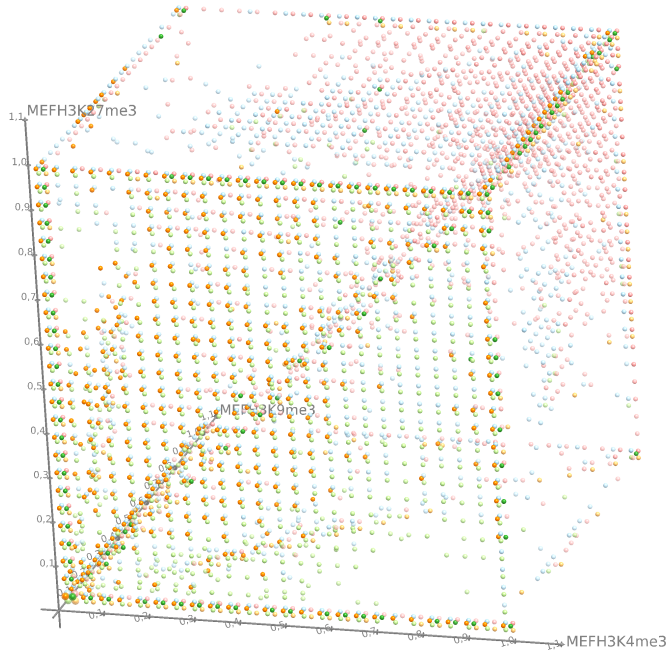


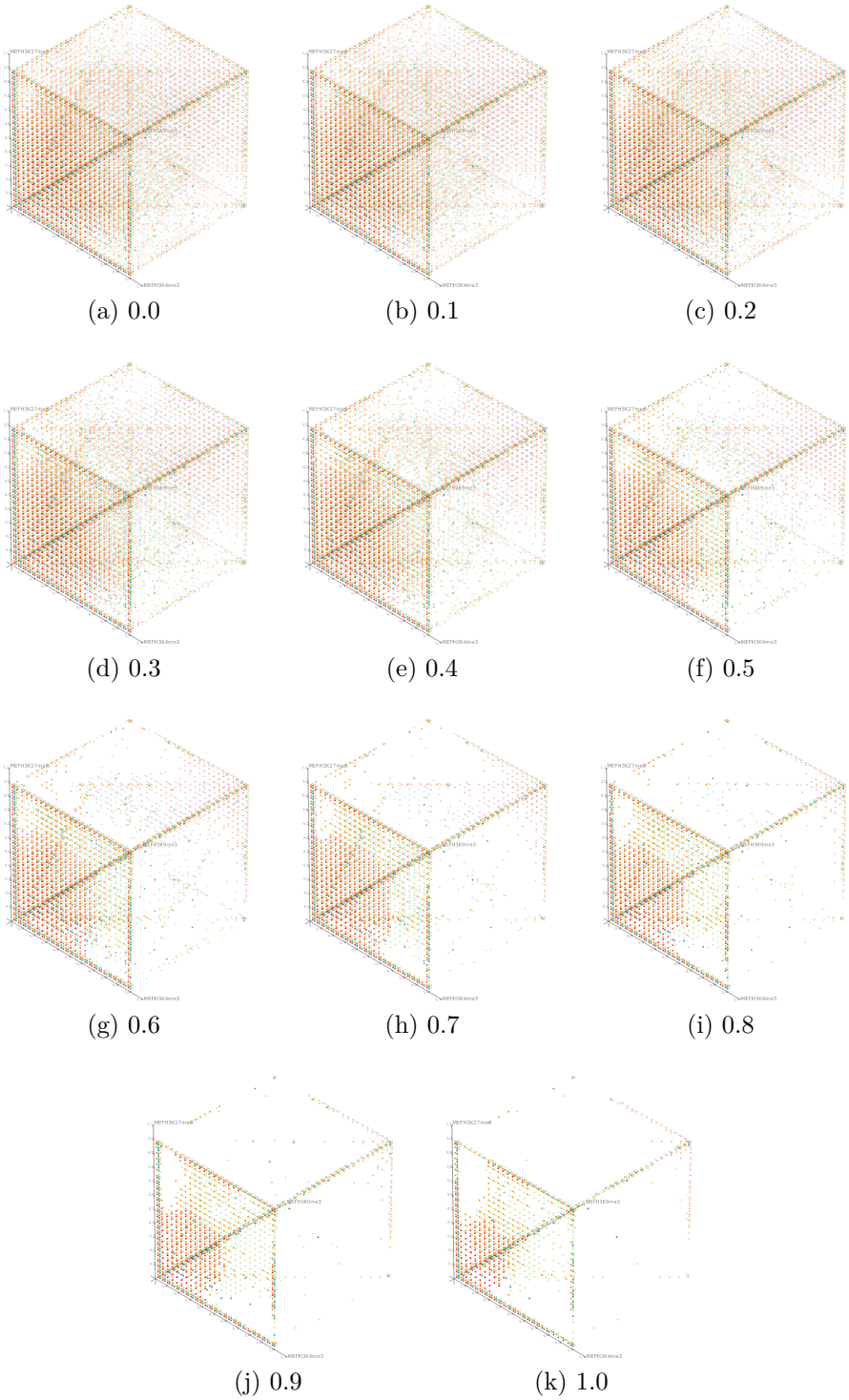

Figure 5.7: MEFH3K4me3, MEFH3K27me3 and MEFH3K9me3 filtered by modifications: the patterns "000" and "100" are filtered.

### Dynamic axis

Considering that the user can change the amount of bins, *TiBi-3D* dynamically calculates the thickness of the axis and adjusts the size of the labels such that they fit properly.
The calculation for the axis and labels are calculated by using normal linear functions with the bin size as the parameter. Two different scatterplots with a bin size of 10 and 25 are shown in Figure 5.9.

### Logarithmic scale

As mentioned before, the data is not homogeneously distributed. Especially the length of each segment and its *CpG-density* is in-homogeneous. Most of the values of this type in the data set are located in a small interval, from the minimum (200bp) up to 1000bp but the maximum value of these dimensions is considerably larger (700,000bp). During the binning process, the bin space is normalized to the minimum and the maximum so that it would produce a large white space with the *CpG-density* drawn with a linear scale (Figure 5.10).
As shown in Figure 5.11, the user can choose for each dimension if a logarithmic scale should be used, using check box. If activated, *TiBi-3D* scales each value



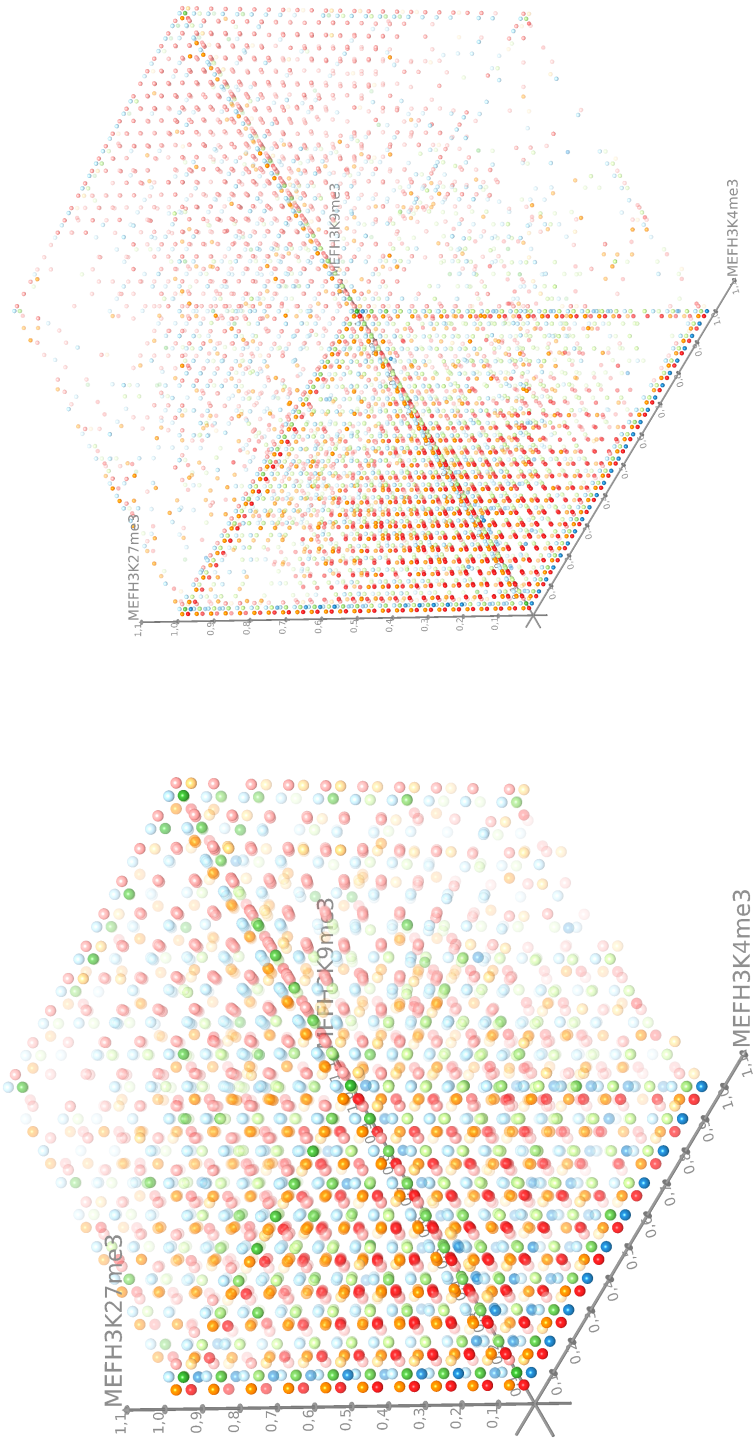
(a) 0.0 (b) 0.1 (c) 0.2

(d) 0.3 (e) 0.4 (f) 0.5

(g) 0.6 (h) 0.7 (i) 0.8

(j) 0.9 (k) 1.0

Figure 5.8: MEFH3K4me3, MEFH3K27me3, and MEFH3K9me3 with different transparency cutoffs from 0.0 (a) to 1.0 (k)




(a) A binned scatterplot with 10 bins

(b) A binned scatterplot with 25 bins

Figure 5.9: The thickness of the axes in *TiBi-3D* is changed and drawn depending on the numbers of bins selected by the user.

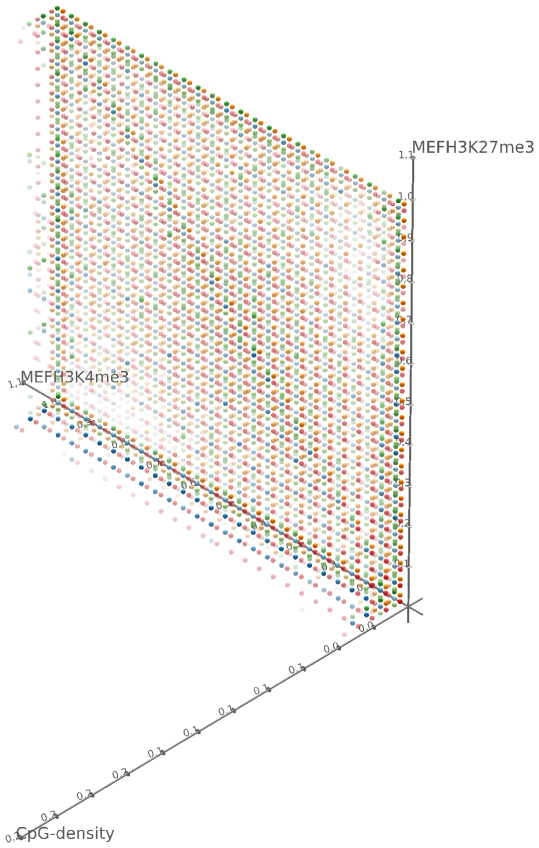



Figure 5.10: MEFH3K4me3, MEFH3K27me3, and *CpG-density* plotted with normal scale on each axis.

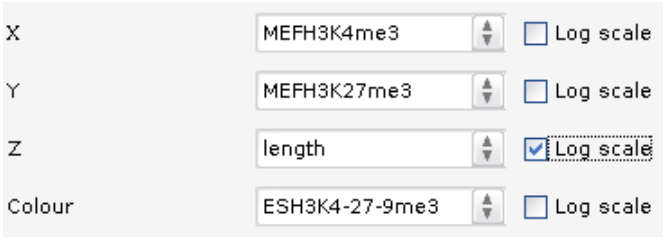



Figure 5.11: The interface for selecting the logarithmical scale for each axis and the color coding

logarithmically, before applying the normal range normalization as described in Section 4.4. If the logarithmic scale for color coding is activated, the interval between the lower and the upper whisker will be divided into eight buckets for the colors, and each logarithmic value will be sorted into one of these buckets for a distinct color choice. Additionally, the color information in the bin summary will be updated with the ranges of each bucket as shown in Figure 5.14.

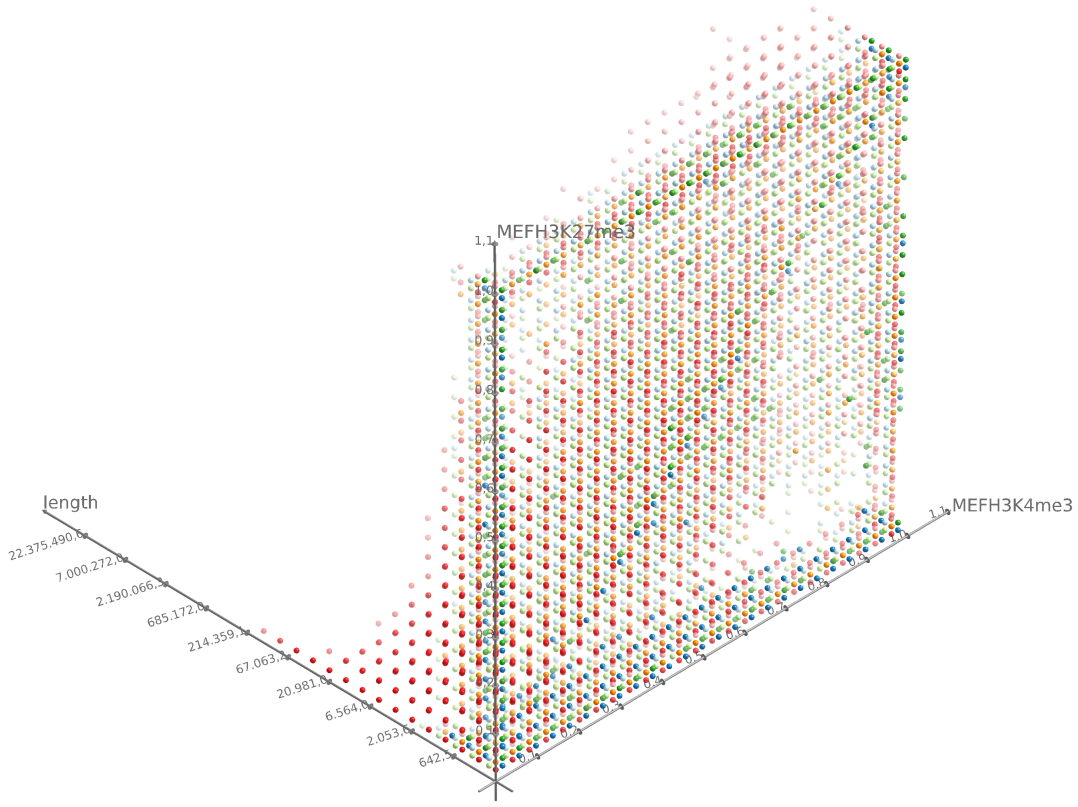



Figure 5.12: Example of the logarithmic scale: the length axes is logarithmic scaled. It allows the user to investigate the values from 200 till 1000 easier.

### Highlighting

*TiBi-3D* provides the possibility to interact with the scatterplot and the data table, for instance, selecting an item from the data table and highlighting the corresponding sphere in the scatterplot. However, in a plot containing a huge amount of spheres, it is complicated to recognize a single static highlighted sphere. Therefore, the highlighting strategy used in *TiBi-3D* attempts to help the viewer using a highlighting bubble as shown in Figure 5.15, where a sphere scales down in one second from a sphere five times as large to its normal radius. Towards Ware's theory of visual perception [36], this movement is recognized during the *pre-attentive stage* of perception that detects the most interesting parts for the later analysis during the second stage. Hence, the spheres are not moving. A moving object in the scatterplot is a great stimulus for gaining attraction for the highlighted sphere since the decreasing of one sphere is recognized as movement.
The user can explore the highlighted item in the *DataInfo Panel* as well (Figure 5.13). It provides all the information about the highlighted item from the data table together with a single independent visualization of its bin in the scatterplot. On the other hand, it is also possible to highlight a bin in the scatterplot and the data table will update the data content to the items of the highlighted bin. Finally, *TiBi-3D* provides a bin summary to show the absolute and the relative values for each sphere and the position of the picked bin (Figure 5.14).

### Export

To produce readily usable results, the user can export the content of the data table into a *BED file* as described in Section 4.2. The program encodes the histone modification patterns together with their related color, such that the user can explore the data in a genome browser, for example *UCSC*, for more in depth exploration of the selected genomic regions (Figure 5.16). *TiBi-3D* is showing epigenetic data, therefore, the regions have no strand information. There is no score information included in the data sets, thus the default values were used for these fields. The thick line fields have the same values as the starting and ending position of the annotation. After exporting the data, the user can import the BED file into a genome browser as shown for example in Figure 5.16. It is also possible to export the whole data table or just a selection.

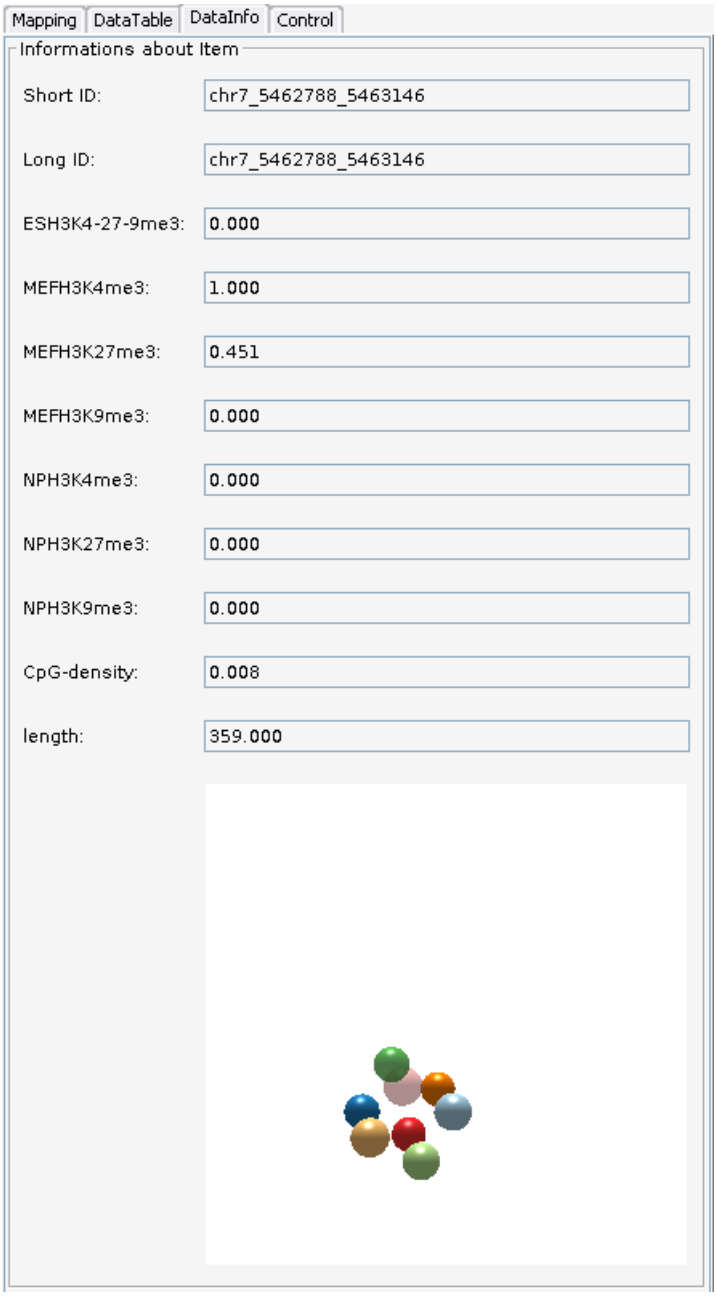



Figure 5.13: The DataInfo tab in *TiBi-3D*, showing the information of the selected item in the table together with an interactive figure of the bin containing this item.

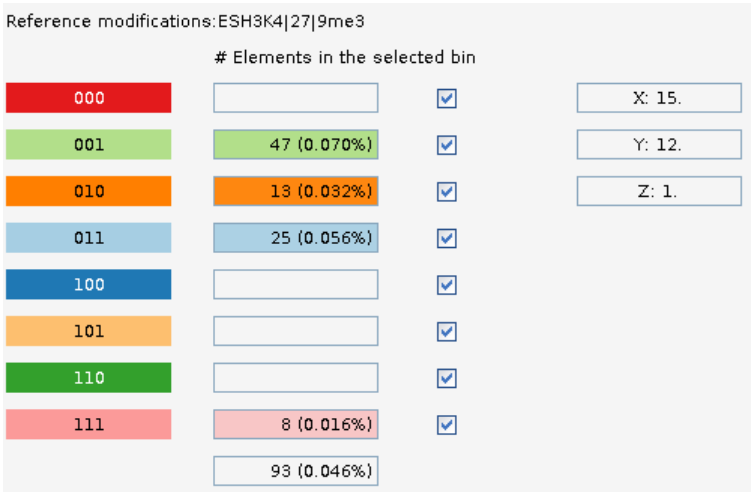



Figure 5.14: The information panel showing a particular bin in *TiBi-3D*

Additionally, *TiBi-3D* supports to produce high resolution snapshots of the scatterplot in the *PNG format*, thus providing an easy way to save interesting views of the scatterplot.

### Perspective Saver

*TiBi-3D* can store specific camera positions for the user permanently. This feature allows the user to compare different data sets using the same specific camera position. As previously mentioned in Section 4.6, *Java3D* stores camera positions similar to object positions. To do this, the program has to store a *transformation matrix* for the camera position. An affine transformation matrix is required to rotate and translate an object in a space. To calculate this affine transformation, the rotation matrix has to be concatenated with the translation matrix obtaining an *augmented matrix*. In addition, a fourth row is added for storing the *homogeneous component* [40] to the translation vector. The remaining of the row is filled up with zeros. The element $m_{44}$ in the Table 5.1 is this component. The *homogeneous component* determines how to project infinity in a finite space.
*TiBi-3D* can access this transformation matrices of the camera and stores them in a serialized objects [41]. These objects can be exported, thus these camera perspectives can be imported later by *TiBi-3D* again.

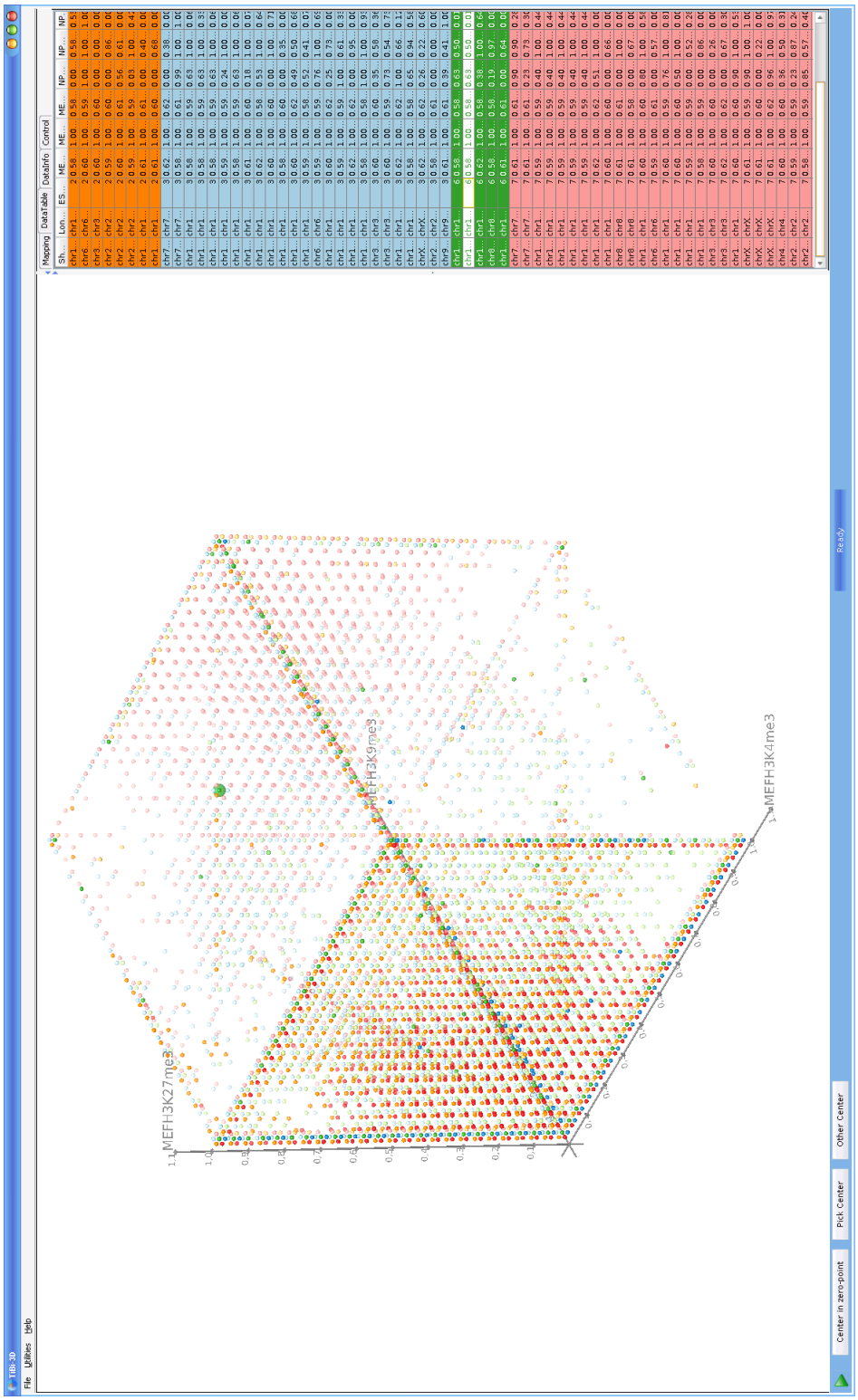



Figure 5.15: The bigger sphere scaled down by time. It indicates the sphere that contains the selected item from the table.

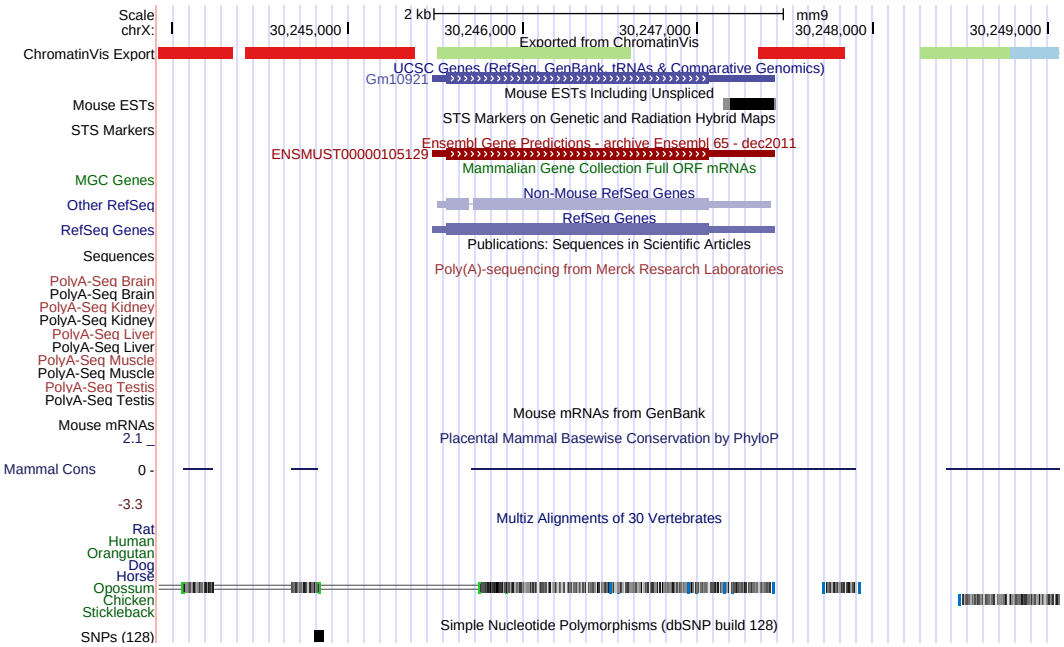



Figure 5.16: An example of visualizing the exported data using the UCSC Genome Browser

Table 5.1: An affine transformation matrix for 3 dimensions

| **x dimension** | **y dimension** | **z dimension** | **translation vector** |
|---|---|---|---|
| $m_{11}$ | $m_{12}$ | $m_{13}$ | $m_{14}$ |
| $m_{21}$ | $m_{22}$ | $m_{23}$ | $m_{24}$ |
| $m_{31}$ | $m_{32}$ | $m_{33}$ | $m_{34}$ |
| $m_{41}$ | $m_{42}$ | $m_{43}$ | $m_{44}$ |

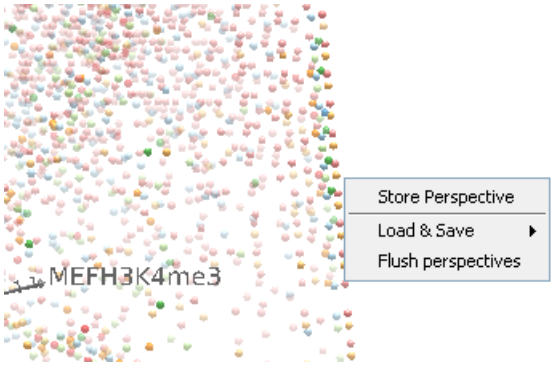



Figure 5.17: An example of how the camera view can be saved

## 5.2 Biology

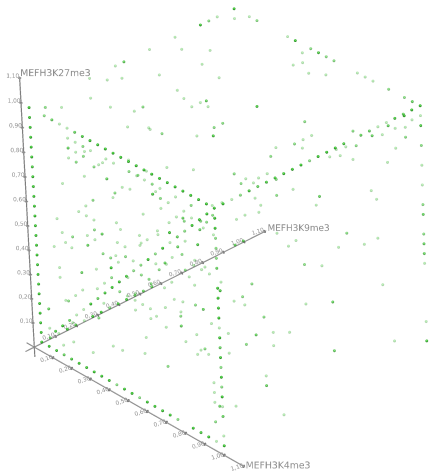
With *TiBi-3D* it was possible to get new insights into the data set compared to [2]. As *TiBi-3D* is able to show the relation of three histone modifications simultaneously, the results cannot (easily) be reproduced using *TiBi-SPLOM*.

### 5.2.1 H3K4me3-H3K27me3 switch

In 2012, a so called H3K4me3-H3K27me3-switch was published by Cui et al. [42]. It was desribed that H3K4me3-H3K27me3 modifications in ESC is changed to H3K4me3 and H3K27un (un for unmodified) in fully differentiated tissues through the process of differentiation. As H3K27me3 is a repressor and H3K4me3 an activator of gene expression, it was assumed that the DNA associated with the modifications contains genes that are important for differentiation.
With *TiBi-3D* and the data set from Mikkelsen et al. [16], it was not possible to recreate this results. As shown in Figure 5.18 and 5.19, there is no clear trend showing that the H3K4me3-H3K27me3 (code 110) modification patterns are changing to H3K4me3 (code 100) modification patterns. In this data set, it was found that these modification patterns are changing to all possible modifications through the cell differentiation from ESC either to MEF or NPC.

### 5.2.2 The ”H3K9me3 hole”

As previously described in Section 2.2.4, H3K9me3 is related to a methylation that silences the expression of the methylated DNA segment. This is a powerful mechanism to shutdown the expression of genes not needed in a specific cell type, because this repressive mark is copied during the replication of the DNA [43]. Since embryonic stem cells can still be differentiated to all cell types not many genes should be marked with H3K9me3. In the differentiated cell types, the amount of H3K9me3 modifications should increase as a result of not all genes are necessary for these cell types.
Based on the given data set it was not possible to validate the reported trend in MEF or NPC. Instead rather the opposite effect is observable. In Figure 5.20 and Figure 5.21, the H3K9me3 modification area is marked. It is easy to see that only few segments are located around the H3K9me3 pattern in MEF or in NPC. Using *TiBi-SPLOM* it was not possible to observe this ”H3K9me3 hole”. Two possible explanations for this behavior could be a lower amount of



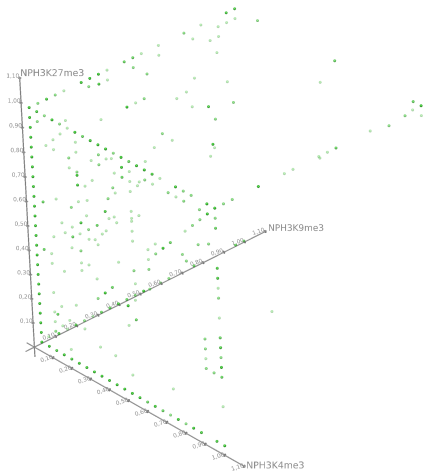

(a) Tiled binned 3D scatterplot showing the correlation of MEF H3K4me3, MEF H3K27me3, and MEF H3K9me3 for 110 modification pattern


(b) Tiled binned 2D scatterplot showing the correlation of H3K4me3 and H3K27me3 in MEF for 110 modification pattern

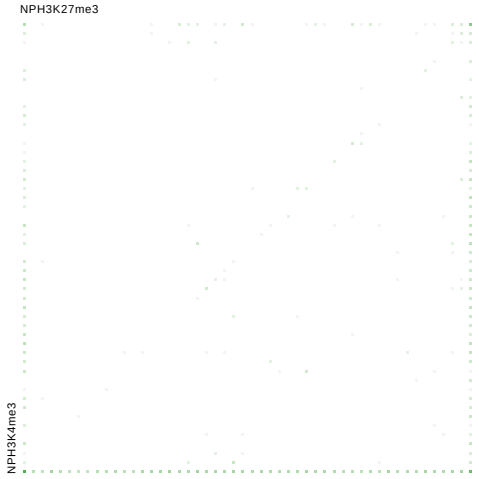

(c) Tiled binned 2D scatterplot showing the correlation of H3K4me3 and H3K9me3 in MEF for 110 modification pattern


(d) Tiled binned 2D scatterplot showing the correlation of H3K9me3 and H3K27me3 in MEF for 110 modification pattern

Figure 5.18: The 110 modification pattern does not switch to 100 in MEF. Using *TiBi-3D*, it is possible to perceive the distribution of 110 patterns. The normalization process of *TiBI-SPLOM* is influenced by the outlier bins in each corner of the figures and reduces the visibility of the other bins.




(a) Tiled binned 3D scatterplot showing the correlation of NPC H3K4me3, NPC H3K27me3, and NPC H3K9me3 for 110 modification pattern

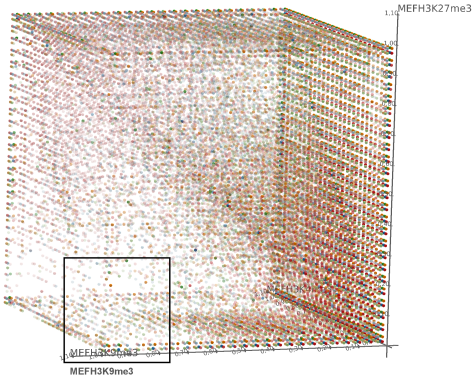



(b) Tiled binned 2D scatterplot showing the correlation of H3K4me3 and H3K27me3 in NPC for 110 modification pattern

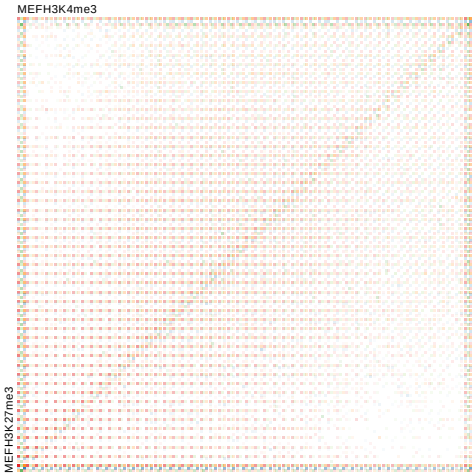

(c) Tiled binned 2D scatterplot showing the correlation of H3K4me3 and H3K9me3 in NPC for 110 modification pattern


(d) Tiled binned 2D scatterplot showing the correlation of H3K9me3 and H3K27me3 in NPC for 110 modification pattern

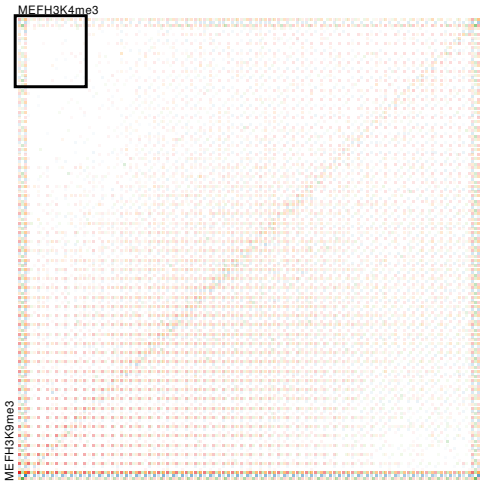
Figure 5.19: The 110 modification pattern does not switch to 100 in NPC. Using *TiBi-3D*, it is possible to perceive the distribution of 110 patterns. The normalization process of *TiBI-SPLOM* is influenced by the outlier bins in each corner of the figures and reduces the visibility of the other bins.



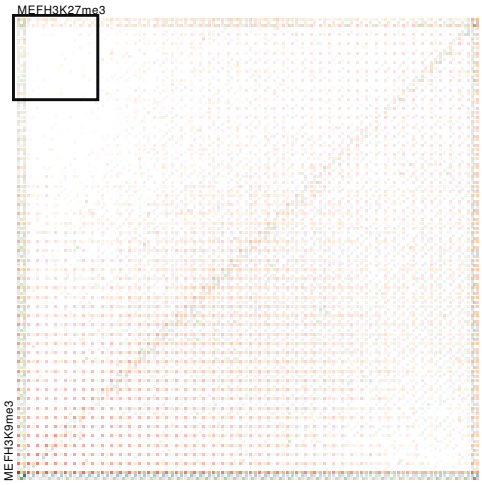

(a) Tiled binned 3D scatterplot showing the correlation of MEF H3K4me3, MEF H3K27me3, and MEF H3K9me3




(b) Tiled binned 2D scatterplot showing the correlation of MEF H3K4me3, MEF H3K27me3



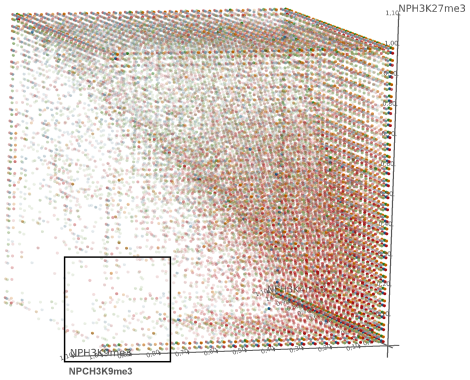

(c) Tiled binned 2D scatterplot showing the correlation of MEF H3K4me3, MEF H3K9me3



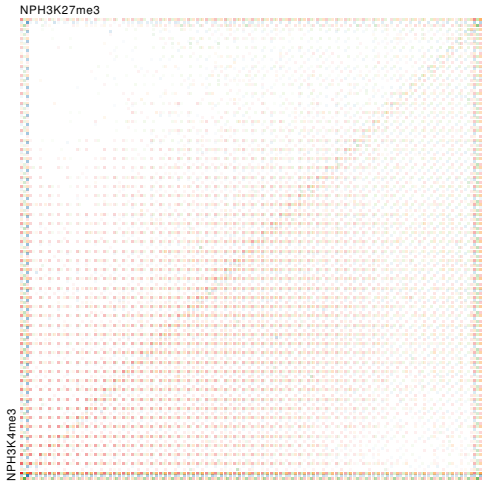

(d) Tiled binned 2D scatterplot showing the correlation of MEF H3K27me3, MEF H3K9me3

Figure 5.20: H3K9me3 is not recruitable in MEF. The labeled rectangles in the plots are showing the ”H3K9me3 hole”.

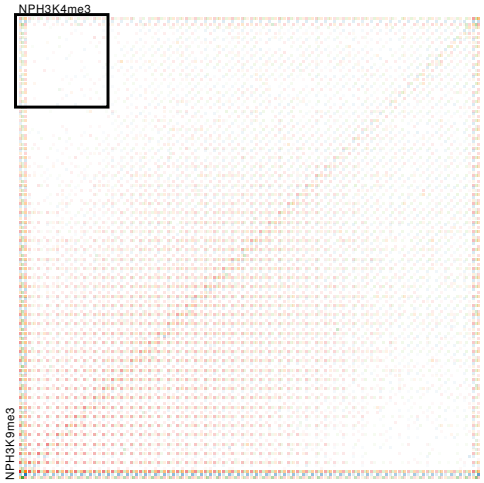
H3K9me3 modifications (around 5950 of 815000 segments) in the data set, or that the course of differentiation is already set in the ESC. This hole is also an explanation of the necessary ”vacuum cleaner” cluster in [1] in NPC since this H3K9me3 hole is more distinct in NPC.



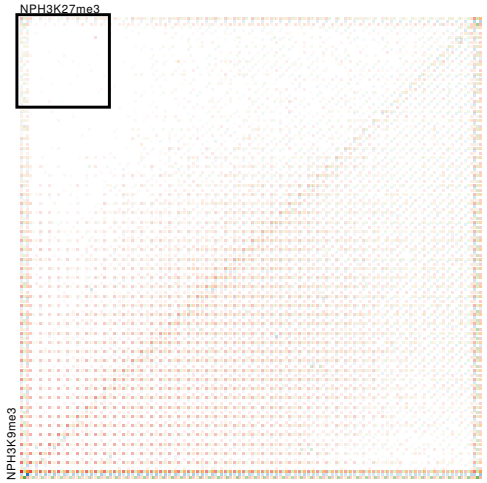

(a) Tiled binned 3D scatterplot showing the correlation of NPC H3K4me3, NPC H3K27me3, and NPC H3K9me3




(b) [Tiled binned 2D scatterplot showing the correlation of NPC H3K4me3, NPC H3K27me3




(c) Tiled binned 2D scatterplot showing the correlation of NPC H3K4me3, NPC H3K9me3




(d) Tiled binned 2D scatterplot showing the correlation of NPC H3K27me3, NPC H3K9me3

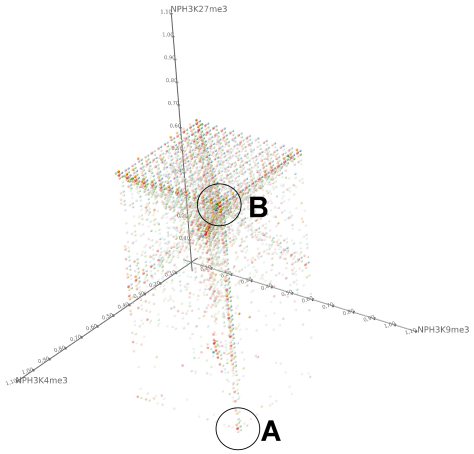
Figure 5.21: H3K9me3 is not recruitable in NPC. The labeled rectangles in the plots are showing the ”H3K9me3 hole”.

### 5.2.3 Correlation of histone modifications in NPC

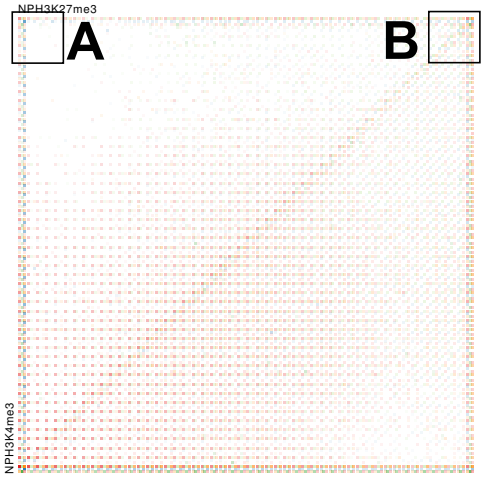
The correlation between H3K4me3 and H3K9me3 with H3K27me3 shows an interesting behavior in NPC since there are only few segments modified with the 101 pattern in NPC (the A labeled area in 5.22a). However, in the area around "B" with the 111 pattern there are many more segments. This leads to the assumption that in NPC H3K4me3-H3K9me3 are not that likely as fully 111 modified segments. This result shows a strong correlation between H3K4me3-H3K9me3 modifications with H3K27me3 that are not detectable in *TiBi-SPLOM* as shown in Figure 5.22. Since H3K4me3-H3K9me3 modifications cannot occur together in NPC without an additional H3K27me3 there must be a cause that is used by mechanisms that are setting all three trimethylations. A more detailed analyses of the bins (25,25,25) and (25,01,25) in MEF and NPC (Figure 5.23) showed that this correlations is stronger NPC than in MEF.

### 5.2.4 Cordilleras in MEF and NPC plots for the pattern 111

In this section, the influence of length and CpG-density on different modifications is investigated. Using the modification filtering, the 111 pattern was analyzed. This revealed a special distribution in MEF and NPC. In MEF (Figures 5.24 and 5.25), each plot has a peak: the area labeled with "A". MEF H3K4me3 and H3K27me3 are forming the same stable shape, however, H3K9me3 is thinned out (Figure 5.25e). This "111 mountain" occurs in an area with segments longer than 6000bp and a CpG-density around 0.3. Compared to NPC, these segments are changed to non modified segments. This cordilleras reveal a specific modification mechanism of H3K4me3, H3K27me3, and H3K9me3 histone marks in MEF and NPC. Since these segments were modified with all histone modifications in ESC, a significant part of them with long length and high CpG-density are conserving this state in MEF but change it completely to being non-modified in NPC.

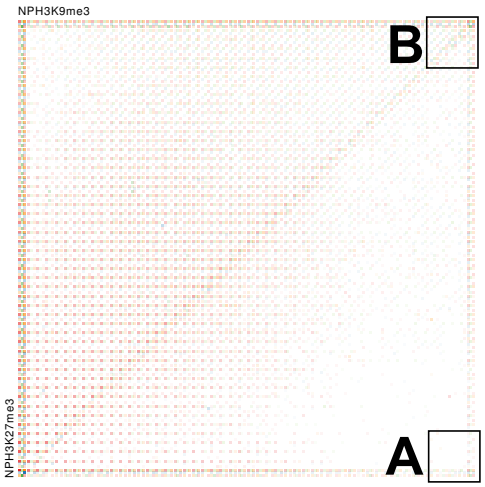



(a) Tiled binned 3D scatterplot showing the correlation of NPC H3K4me3, NPC H3K27me3, and NPC H3K9me3. 101 modification patterns (Label "A") are not likely as fully 111 pattern (Label "B"). The scatterplot was cutted with cutting planes to expose the 110/111 patterns.



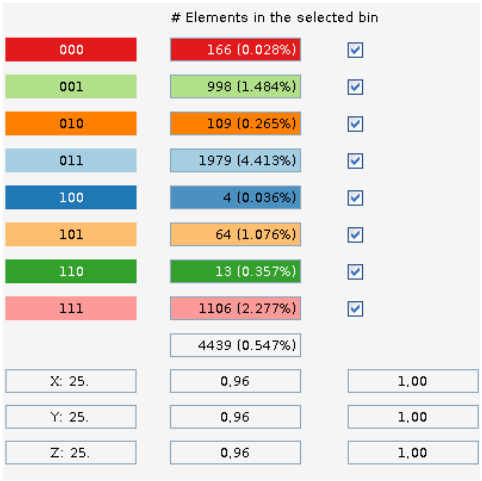

(b) Tiled binned 2D scatterplot showing the correlation of NPC H3K4me3, NPC H3K27me3. Label "A" shows the 110 modification patterns and Label "B" shows the 111 patterns.


(c) Tiled binned 2D scatterplot showing the correlation of NPC H3K4me3, NPC H3K9me3. Label "A" shows the 110 modification patterns and Label "B" shows the 111 patterns. In this plot, both patterns are at the same spot.

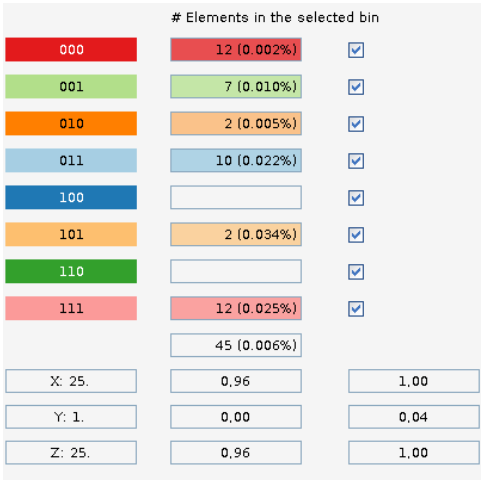



(d) Tiled binned 2D scatterplot showing the correlation of NPC H3K27me3, NPC H3K9me3. Label "A" shows the 110 modification patterns and Label "B" shows the 111 patterns.

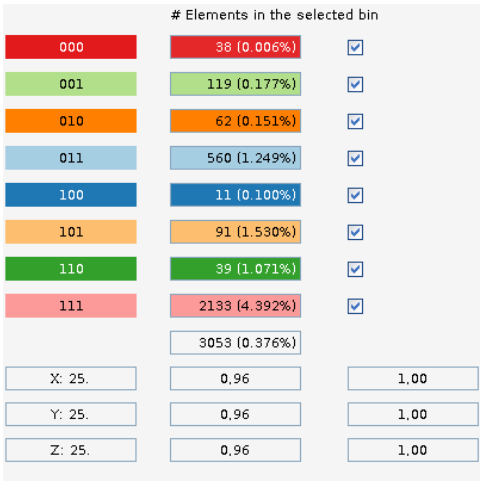
Figure 5.22: The modification pattern 111 is more likely to occur than 110 in NPC. H3K4me3-H3K9me3 modifications have a strong correlation with H3K27me3.



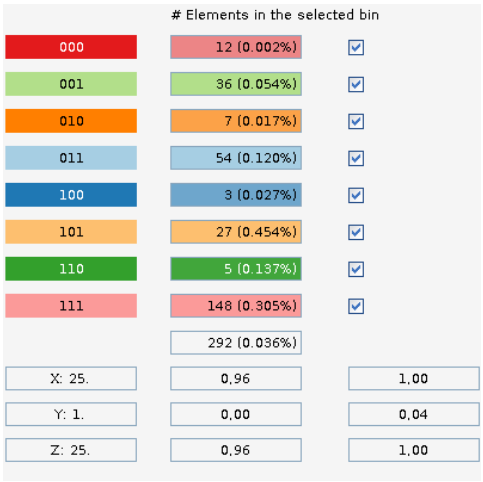

(a) Itemview of bin (25,25,25) in NPC of the plot shown in Figure 5.22a




(b) Itemview of bin (25,1,25) in NPC of the plot shown in Figure 5.22a

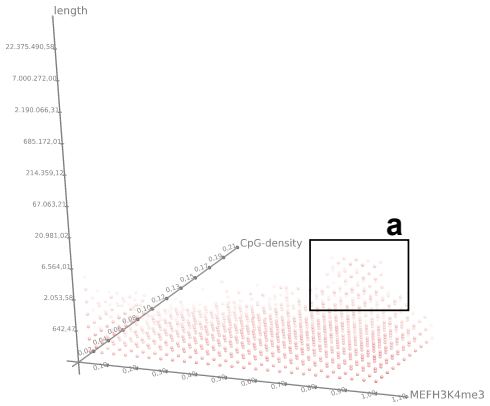



(c) Itemview of bin (25,25,25) in MEF

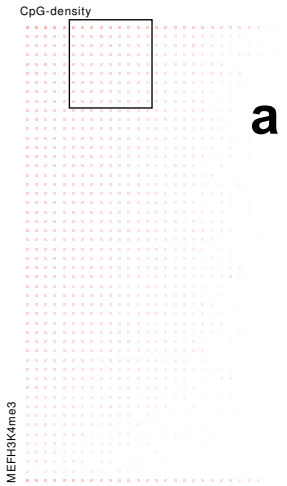



(d) Itemview of bin (25,1,25) of in MEF

Figure 5.23: Comparison of the item view of the bins (25,25,25) and (25,1,25) in MEF and NPC.

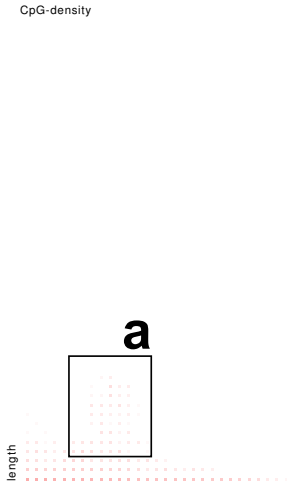



(a) TiBi-3D for MEF H3K4me3, CpG-density, and length filtered for code 111 (rose). The peak of the "111 mountain" is labeled with "a".



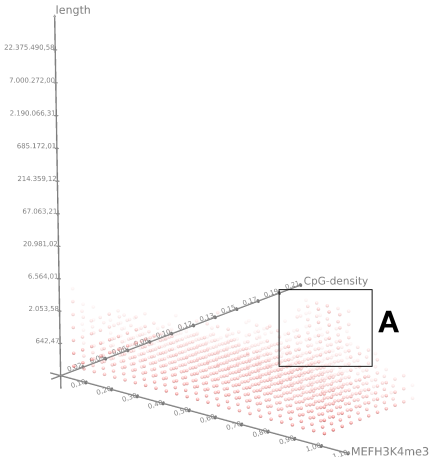

(b) *TiBi-SPLOM* for CpG-density and MEF H3K4me3 filtered for code 111 (rose).


(c) *TiBi-SPLOM* for length and MEF H3K4me3 filtered for code 111 (rose).

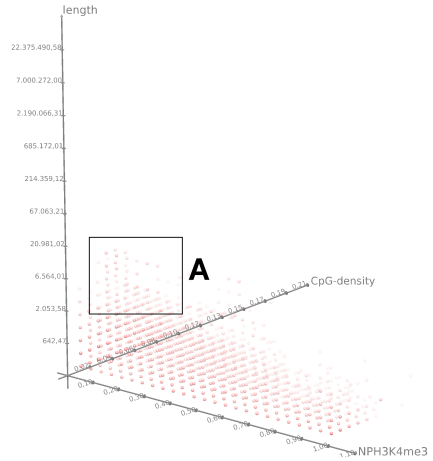



(d) *TiBi-SPLOM* for CpG-density and length filtered for code 111 (rose).

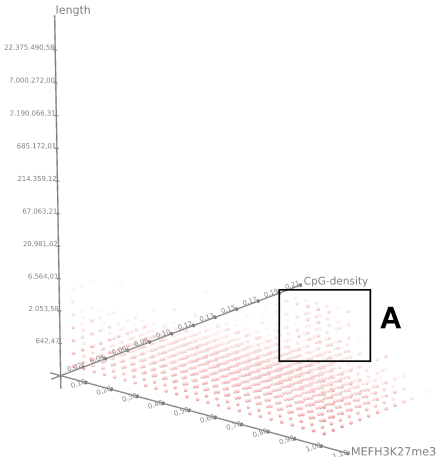
Figure 5.24: The 111 peak in MEF with H3K4me3, CpG-density, and length. Although the peak can be seen using *TiBi-3D*, it is difficult to identify them in 2D.



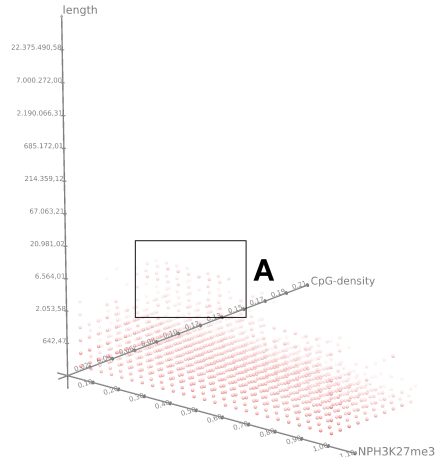

(a) MEF H3K4me3



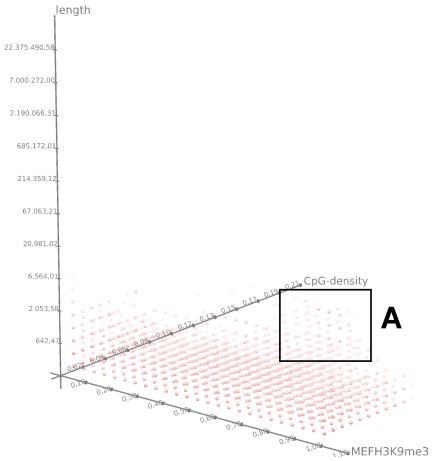

(b) NPC H3K4me3



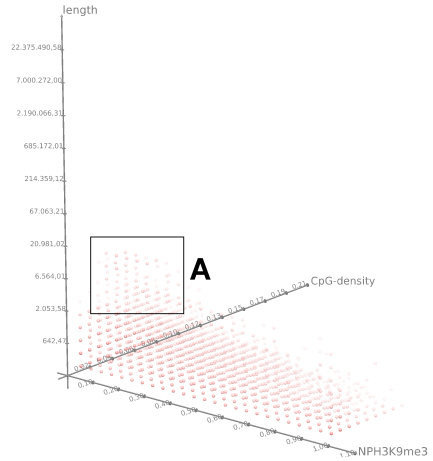

(c) MEF H3K27me3




(d) NPC H3K27me3




(e) MEF H3K9me3




(f) NPC H3K9me3

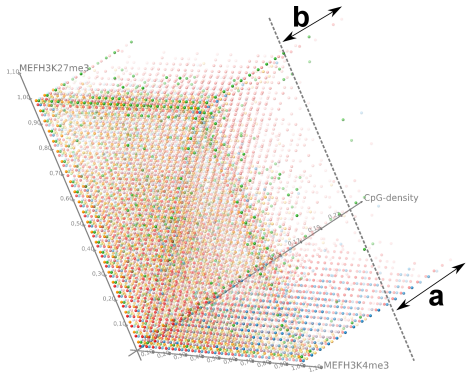
Figure 5.25: The 111 cordilleras in all three histone modifications in MEF and NPC. The peak shifts from an attached trimethylation in MEF to unmodified segments in NPC.

### 5.2.5 Correlation of 111 modifications and CpG-density in MEF/NPC

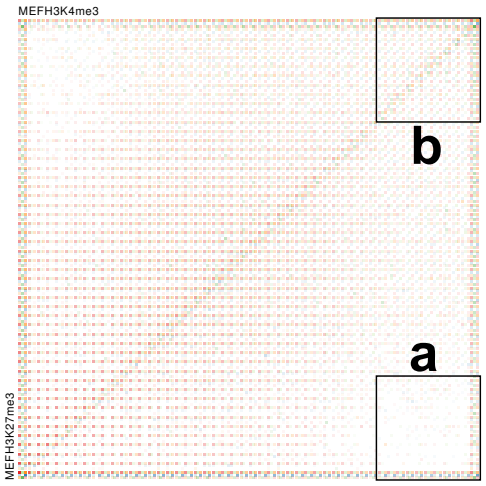
In Figure 5.26, the relation between H3K4me3, H3K27me3, and CpG-density in MEF is shown. The CpG-density is an interesting genomic feature that may indicate binding sites for histone modification enzymes. It is clearly recognizable that only H3K4me3 appears with high CpG-density (mark "A") and that H3K27me3 marks cannot be found with this level of CpG-density (mark "B"). High CpG-density occurs in MEF only without H3K27me3 marks. Around the label "A" the majority of the bins contains only segments having red and blue color. As these colors represent the non-modified and H3K4me3 modified elements in ESC, it indicates a part of these segments were modified after the differentiation into MEFs. This modification pattern seems to be specific for H3K4me3 in MEF in relation to high CpG-density since many segments with H3K4me3 and H3K27me3 modifications were found in ESC. As described in Section 2.2.4, H3K4me3 modifications act as an activator of transcription, thus it can be assumed that this special segments are involved in controlling MEF-specific cell functions.

### 5.2.6 Influence of CpG-density towards H3K4me3 modifications in murine fibroblasts

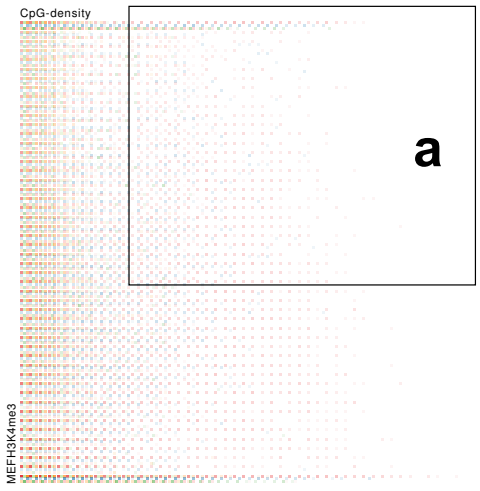
The influence of the CpG-density towards H3K4me3 in MEF was analyzed in depth. In Figure 5.27, the coloring of the bins is changed from the ESC code to show the CpG-density. It is clearly visible that the segments having a high CpG-density are located around the corner of H3K4me3 in the plot. For further analysis, the data was exported as a bed file from *TiBi-3D* and uploaded to the *UCSC genome browser*. In Figure 5.28a, the genegraph of these segments is shown for all chromosomes. It shows the distribution of these segments where a peak visualizes a locus with a lot of segments. A short analysis of this genegraph does not reveal any specific pattern since the data is more or less homogeneous distributed over all chromosomes.



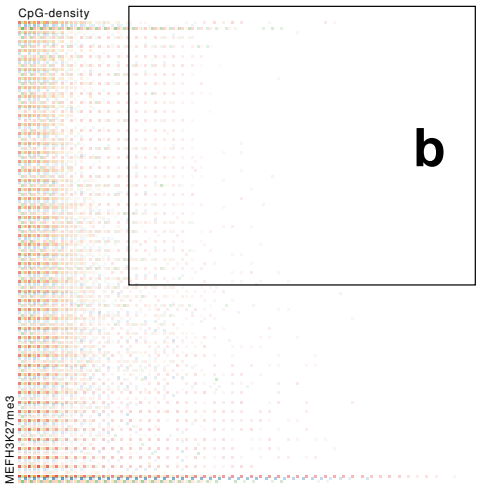

(a) Tiled binned 3D scatterplot showing the correlation of MEF H3K4me3, MEF H3K27me3, and CpG-density. The region labeled with "a" shows segments with H3K4me3 modifications and high CpG-density, while the region labeled with "b" shows the region with H3K27me3 marks and a less CpG-density.




(b) Tiled binned 2D scatterplot showing the correlation of MEF H3K4me3 and MEF H3K27me3. The labels correspond to Figure 5.26a




(c) Tiled binned 2D scatterplot showing the correlation of CpG-density and MEF H3K4me3. The label "a" corresponds to Figure 5.26a




(d) Tiled binned 2D scatterplot showing the correlation of CpG-density and MEF H3K27me3. The label "b" corresponds to Figure 5.26a

Figure 5.26: Comparison between MEF H3K4me3, MEF H3K27me3, and CpG-density using *TiBi-3D* and *TiBi-SPLOM*.

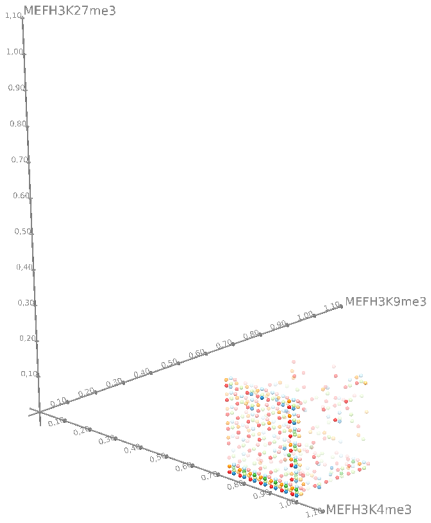
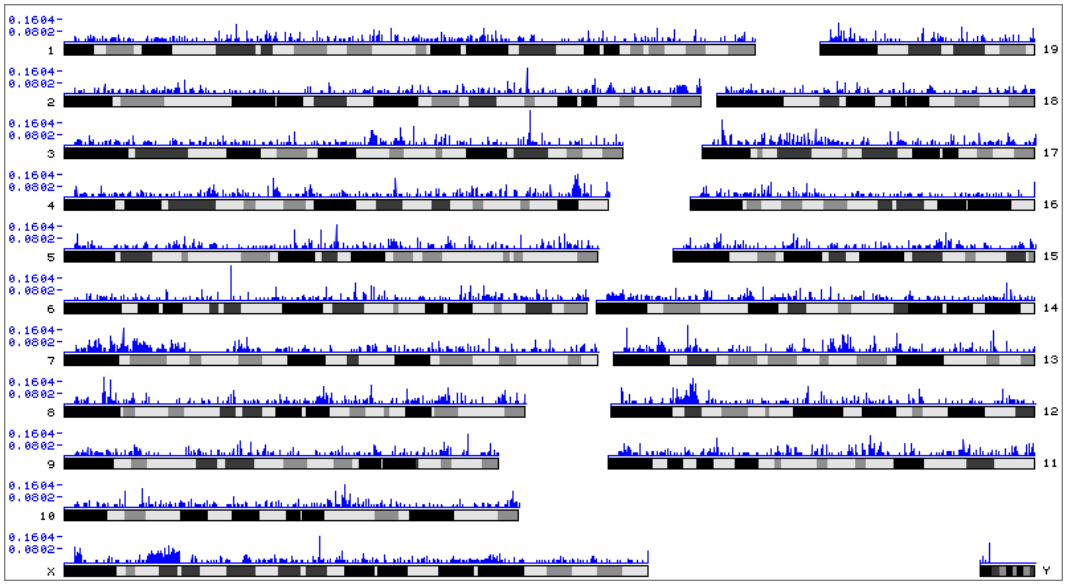
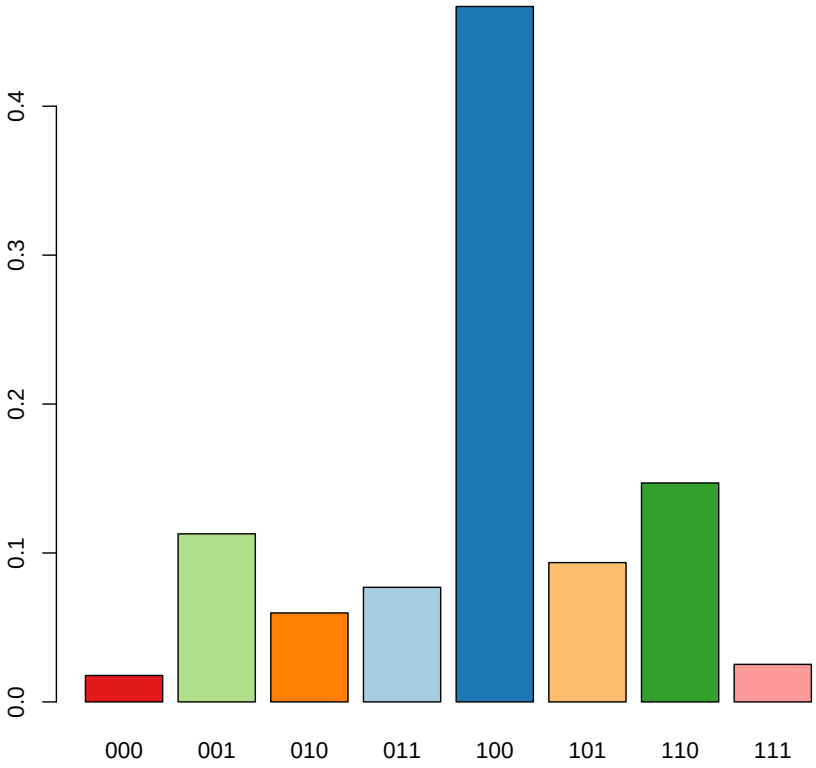



Figure 5.27: Corner of MEF H3K4me3, H3K27me3, and H3K9me3 with CpG-density as coloring

In the next step, the origin of the segments was analyzed. Figure 5.28b shows the origin of all segments in the H3K4me3 corner. Around 45% of all segments were modified with H3K4me3 before in ESC, but only 3% were acquired from non-modified segments in ESC. Subsequently, the data set was again filtered for those 1000 segments with the highest CpG-density. Since the bed file generated by *TiBi-3D* does not contain anymore the CpG-density of the segments but the segment ID as described in Section 4.1, the forth column was cut out using the bash tool cut (`"cut -n 4"`) and the elements of this new file were used for grepping these segments out of the whole data set using the bash tool grep (`"grep -f cutted-bedfile dataset-file"`). This generated a data set file contains only the elements of the corner and were sorted with `"sort -n -f ";" -k10,10"` since the 10. column stores the CpG-density. Appending the command `" | tail -1000"` only the 1000 segments with the highest CpG-density were saved. Figure 5.29a shows the origin of these segments and reveals that around 70% of the segments were not modified in ESC. This result supports the assumption that in MEF are special segments related to specific cell functions. A genegraph generated by the *UCSC genome browser* for these 1000 segments is shown in Figure 5.29b. However, it does not show significant peaks over all chromosomes, but compared to the genegraph with all segments (Figure 5.28a), it is noticeable that no peaks occur on chromosome Y and only a few on chromosome X. The segments with the highest CpG-density are apparently not on these chromosomes.

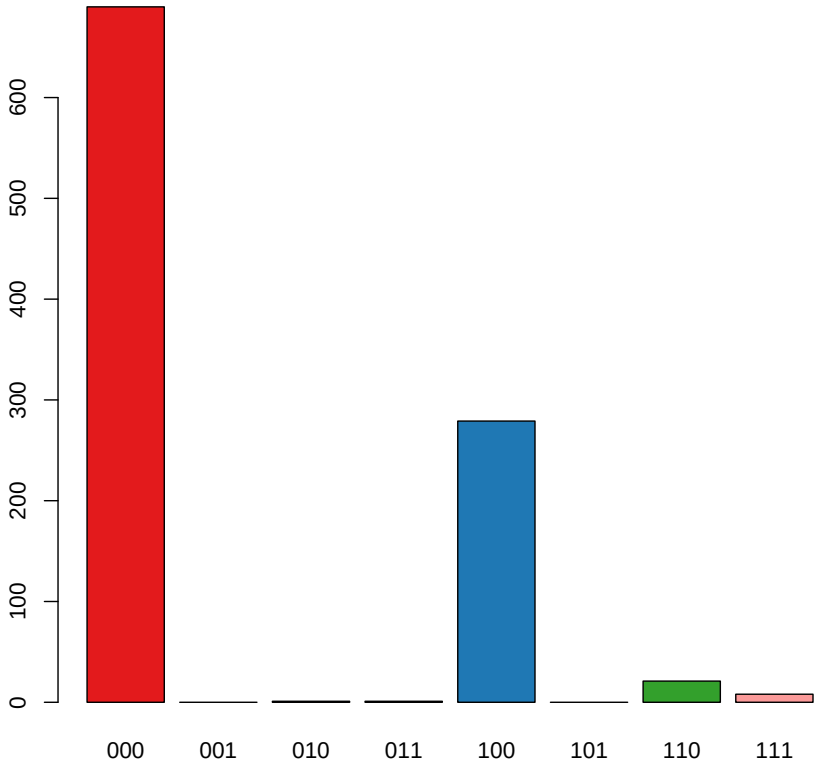


(a) Genegraph of all segments in the H3K4me3 corner in MEF

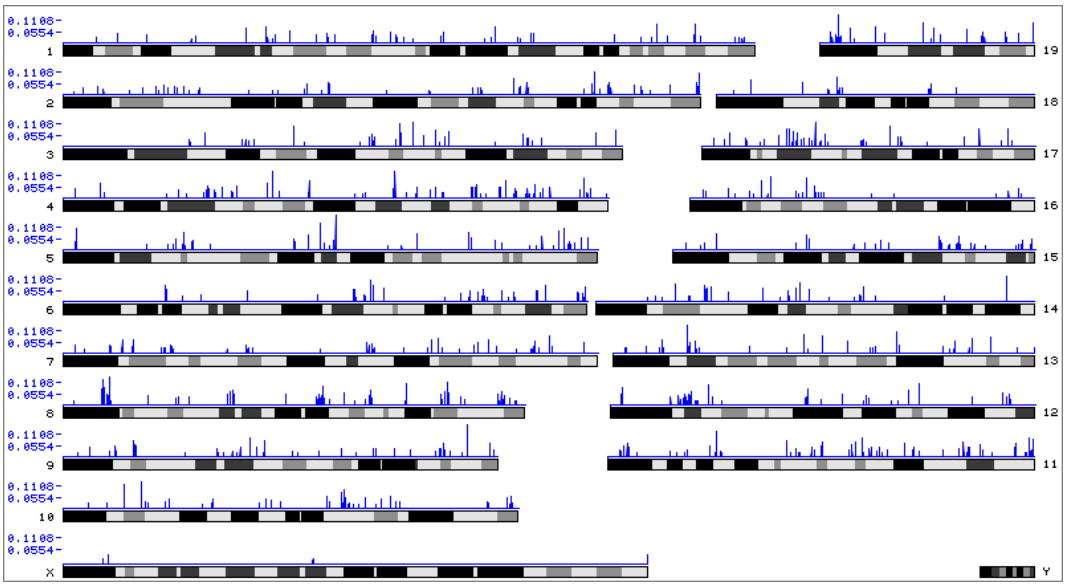



(b) Origin of all segments in the H3K4me3 corner in MEF

Figure 5.28: Analysis of pure H3K4me3 modifications in MEF




(a) Origin of 1000 segments with the highest CpG-density in the H3K4me3 corner in MEF




(b) Genegraph for the 1000 segments with the highest CpG-density in the H3K4me3 corner in MEF

Figure 5.29: Analysis of 1000 pure H3K4me3 modifications with high CpG-density in MEF

# Chapter 6

# Conclusion

From a biological perspective, the results of *TiBi-3D* shows this is a useful and strong exploration tool for analyzing epigenetic data sets. Providing various possibilities to interact with the scatter plot like filtering, rotating, and data brushing, it can help bioinformaticians and biologists to get a deeper understanding of the relationships of histone modifications. It can also reveal interesting spots in data sets and then export them as figures or as bed files for later analysis. Compared to *TiBi-SPLOM*, *TiBi-3D* can visualize more complex correlations, since it can visualize three dimensions simultaneously. However, 3D visualizations have to circumvent the effects of occlusion. The overlapping of data points in the 3D tiled-binned scatterplots is less than in the 2D visualizations, because the third hidden dimension is accumulated in the two visible dimensions. Additionally, *TiBi-3D* allows the user to explore changes of the histone modifications, that are hard or impossible to observe in 2D, for instance, the effect of the CpG-density towards trimethylation of H3K4, H3K27, and H3K9 in MEF and NPC. *TiBi-3D* also has an efficient memory management for 3D visualizations, since the first approach to visualize the used data set in 3D shown in Figure 5.2a consumed more than 14 GB RAM at runtime. Using the same data set used in this thesis, *TiBi-3D* allocates only 2 GB of memory for its heap and uses only 1 GB for the calculation. Therefore, *TiBi-3D* could still be used on older machines with a 32 bit operating system. Since *TiBi-3D* is written in Java, it can be executed on every operating system without additional compilation.

Every data set is pre-processed as described in Section 2.2.4 and can thus be analyzed with *TiBi-3D*. The results of *TiBi-3D* give a better understanding of the interaction of histone modifications during the differentiation of the cell type and the influence of attributes of the DNA wrapped around the histones, for instance, length or CpG-density of the segment.

# Chapter 7

# Future work

Although *TiBi-3D* produces meaningful results, there is still space for improvements. For example, the color space is still not perfect, since some of the used colors can be confusing. However, eight colors are necessary to encode all possible modification patterns and the position of each sphere also encodes the modification pattern. *TiBi-3D* is able to visualize just three histone modifications in one scatter plot. Since many histone modifications as shown in Figure 2.4 are related only to the chromatin reorganization, it would be useful to extend *TiBi-3D* to analyze more dimensions simultaneously. Additional data mining is necessary for extending *TiBi-3D* as it is impossible to visualize more than three dimensions with scatter plots. Additionally, a better integration between the UCSC genome browser and *TiBi-3D* would be useful. As described in Section 5.2.6, it is tedious to re-import the exported bed files in *TiBi-3D* for further analysis. For example, a filtering mechanism for segment positions with a given bed file would improve future analysis, since the user could filter the data set for interesting chromosomes or regions for known gene families, e.g., the "housekeeping genes". Furthermore, a deeper analysis of the result described in Section 5.2.6 like a GO analysis could provide more information to support the assumption that these segments contain genes that are responsible for MEF specific cell functions.

# Bibliography


[1] Daniel Gerighausen. *Kombination von K-means++ Clustering und PCA zur Analyse von Chromatin-Daten*. Universität Leipzig, 2013.

[2] Dirk Zeckzer, Daniel Gerighausen, Lydia Steiner, and Sonja J. Prohaska. Analyzing chromatin using tiled binned scatterplot matrices. *CoRR*, abs/1407.2084, 2014.

[3] P. Cheung and P. Lau. Epigenetic regulation by histone methylation and histone variants. *Molecular endocrinology (Baltimore, Md.)*, 19(3):563–573, March 2005.

[4] Ronald Berezney and K.W. Jeon. *Nuclear Matrix: Structural and Functional Organization*. A Single Volume Reprint of Volumes 162 a and B in International Review of Cytology Series. Academic Press, 1995.

[5] Nucleosome organization. Last accessed at 22.10.2014 and modified: http://en.wikipedia.org/wiki/Nucleosome#mediaviewer/File:Nucleosome_organization.png.

[6] Model of the fine structure of a chromosome. Last accessed at 10.11.2014 and modified (changed the telomere sequence): http://upload.wikimedia.org/wikipedia/commons/1/1a/Chromosom.svg.

[7] Histone modifications involved in chromatin reorganization. Last accessed at 10.11.2014: http://www.nature.com/bonekey/knowledgeenvironment/2010/1009/bonekey20100464/images/bonekey20100464-f3.jpg.

[8] Karolin Luger, Armin W Mäder, Robin K Richmond, David F Sargent, and Timothy J Richmond. Crystal structure of the nucleosome core particle at 2.8 å resolution. *Nature*, 389(6648):251–260, 1997.

[9] James D Watson and Francis HC Crick. The structure of dna. In *Cold Spring Harbor Symposia on Quantitative Biology*, volume 18, pages 123–131. Cold Spring Harbor Laboratory Press, 1953.

[10] F. Sanger and A.R. Coulson. A rapid method for determining sequences in dna by primed synthesis with dna polymerase. *Journal of Molecular Biology*, 94(3):441 – 448, 1975.

[11] Gel image compared with flourescent peaks. Last accessed at 09.12.2014: http://en.wikipedia.org/wiki/Sanger_sequencing#mediaviewer/File:Radioactive_Fluorescent_Seq.jpg.

[12] EC Hayden. Is the $1000 genome for real. *Nature News*, 2014.

[13] Martin Kircher. Understanding and improving high-throughput sequencing data production and analysis. 2011.

[14] Schematic workflow of the Illumina sequencing. Last accessed at 30.10.2014: http://www.dkfz.de/gpcf/hiseq_technology.html.

[15] PJ Mitchell and R Tjian. Transcriptional regulation in mammalian cells by sequence-specific dna binding proteins. *Science*, 245(4916):371–378, 1989.

[16] Tarjei S. Mikkelsen and et al. Genome-wide maps of chromatin state in pluripotent and lineage-committed cells. *Nature*, 448(7153):553–560, 08 2007.

[17] Schübeler et al. The histone modification pattern of active genes revealed through genome-wide chromatin analysis of a higher eukaryote. *Genes Dev*, 18(11):1263–71, Jun 2004.

[18] R Cao, L Wang, H Wang, L Xia, H Erdjument-Bromage, P Tempst, R S Jones, and Y Zhang. Role of histone H3 lysine 27 methylation in Polycomb-group silencing. *Science*, 298(5595):1039–43, Nov 2002.

[19] Steve Hoffmann, Christian Otto, Stefan Kurtz, Cynthia M Sharma, Philipp Khaitovich, Jörg Vogel, Peter F Stadler, and Jörg Hackermüller. Fast mapping of short sequences with mismatches, insertions and deletions using index structures. *PLoS computational biology*, 5(9):e1000502, 2009.

[20] Lydia Steiner, Lydia Hopp, Henry Wirth, Jörg Galle, Hans Binder, Sonja J. Prohaska, and Thimo Rohlf. A Global Genome Segmentation Method for Exploration of Epigenetic Patterns. *PLOS ONE*, 7(10):e46811, 2012.

[21] Karolin Luger, Mekonnen L. Dechassa, and David J. Tremethick. New insights into nucleosome and chromatin structure: an ordered state or a disordered affair? *Naturereviews Molecular Cell Biology*, 13:436–447, July 2012.

[22] Sarah Seifert. *Visualization of chromatin data using k-means++ clustering and starplots*. Universität Leipzig, 2012.

[23] Daniel Abitz. *Analyse von Chromatin durch den K-median und das Consensus Clustering*. Universität Leipzig, 2014.

[24] Daniel Gerighausen, Dirk Zeckzer, Lydia Steiner, and Sonja J. Prohaska. *ChromatinVis: a tool for analyzing epigenetic data*. Poster presented at the the VIZBI 2014, 2014.

[25] Lydia Steiner, Lydia Hopp, Henry Wirth, Jörg Galle, Hans Binder, Sonja Prohaska, and Thimo Rohlf. *A global genome segmentation method for exploration of epigenetic patterns*. PLOS ONE 2012, Leipzig, 2012.

[26] Chih Long Liu, Tommy Kaplan, Minkyu Kim, Stephen Buratowski, Stuart L Schreiber, Nir Friedman, and Oliver J Rando. Single-nucleosome mapping of histone modifications in s. cerevisiae. *PLoS Biol*, 3(10):e328, 08 2005.

[27] Georges Grinstein, Marjan Trutschl, and Urška Cvek. High-Dimensional Visualizations. In *Proceedings of the VII Data Mining Conference KDD Workshop 2001*, pages 7–19, San Francisco-CA, USA, 2001. ACM Press, New York.

[28] W. James Kent, Charles W. Sugnet, Terrence S. Furey, Krishna M. Roskin, Tom H. Pringle, Alan M. Zahler, and David Haussler. The Human Genome Browser at UCSC. *Genome Research*, 12(6):996–1006, 2002.

[29] Cynthia A. Brewer. Color brewer, 2014. http://colorbrewer2.org/?type=qualitative&scheme=Paired&n=#8, accessed April 5th, 2014.

[30] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[31] J.W. Tukey. *Exploratory Data Analysis*. Addison-Wesley series in behavioral sciences. Addison-Wesley Publishing Company, 1977.

[32] A box plot. Last accessed at 29.07.2014 and modified: http://commons.wikimedia.org/wiki/File:Boxplot_vs_PDF.svg#mediaviewer/File:Boxplot_vs_PDF.svg.

[33] Thomas Porter and Tom Duff. Compositing digital images. *SIGGRAPH Comput. Graph.*, 18(3):253–259, January 1984.

[34] R. W. G. Hunt. *The Reproduction of Colour*. Fountain Press, 1957.

[35] Example for YIQ colorspace. Last accessed at 16.06.2014 and modified: http://upload.wikimedia.org/wikipedia/commons/thumb/0/01/YIQ_components.jpg/320px-YIQ_components.jpg.

[36] C. Ware. *Information Visualization: Perception for Design*. Interactive Technologies. Elsevier Science, 2004.

[37] Netbeans IDE 8. Last accessed at 12.11.2014: https://netbeans.org/.

[38] Foxtrot. Last accessed at 12.11.2014: http://foxtrot.sourceforge.net/.

[39] Dennis Lieu and Sheryl Sorby. *Visualization, modeling, and graphics for engineering design*. Cengage Learning, 2008.

[40] A.F. Möbius, R. Baltzer, and F. Klein. *Gesammelte Werke*. Number Bd. 1 in Gesammelte Werke. Sändig Reprint, 1967.

[41] Documentation serializable. Last accesed at 16.06.2014: http://docs.oracle.com/javase/7/docs/api/java/io/Serializable.html.

[42] Peng Cui, Wanfei Liu, Yuhui Zhao, Qiang Lin, Daoyong Zhang, Feng Ding, Chengqi Xin, Zhang Zhang, Shuhui Song, Fanglin Sun, Jun Yu, and Songnian Hu. Comparative analyses of h3k4 and h3k27 trimethylations between the mouse cerebrum and testis. *Genomics, Proteomics and Bioinformatics*, 10(2):82 – 93, 2012.

[43] Julie A Law and Steven E Jacobsen. Establishing, maintaining and modifying dna methylation patterns in plants and animals. *Nature Reviews Genetics*, 11(3):204–220, 2010.

# List of Figures

# List of Tables

# Chapter 8

# Acknowledgments

I would like to thank Dirk, Lydia, and Sonja for supervising me for more then two years. My thanks is also due to the whole Bioinf for providing me an office and a nice and friendly working environment. Thank you to proof-reading my thesis, Alvaro!
Special thanks go to my family. Without your patience and help it would not have been possible to study and write this thesis.

**Thanks!**

This thesis is powered by Yerba-Mate.
*Several mint plants were harmed during this thesis for making mojitos.*

# Chapter 9

# Eidesstattliche Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann.

Leipzig, den 18. Dezember 2014 _____

Ort, Datum Unterschrift