SigTools: an Exploratory Visualization Tool for Genomic Signals

by

Shohre Masoumi

B.Sc., Amirkabir University of Technology, 2018

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Spring 2021

Declaration of Committee

Name:	**Shohre Masoumi**
Degree:	**Master of Science**
Thesis title:	**SigTools: an Exploratory Visualization Tool for Genomic Signals**
Committee:	**Chair:** Saba Alimadadi Assistant Professor, Computing Science
	Kay C. Wiese Supervisor Associate Professor, Computing Science
	Maxwell W. Libbrecht Committee Member Assistant Professor, Computing Science
	Sheelagh Carpendale Examiner Professor, Computing Science

Abstract

With the advancement of sequencing technologies, genomic data sets are constantly being expanded by high volumes of different data types. One recently introduced data type in genomic science is genomic signals, with genomic coordinates associated with a score or probability indicating some form of biological activity. An example of genomic signals is Epigenomic *marks* which represent short-read coverage measurements over the genome, and are utilized to locate functional and nonfunctional elements in genome annotation studies. To understand and evaluate the results of such studies, one needs to explore and analyze the characteristics of the input data.

Information visualization is an effective approach that leverages human visual ability in data analysis. Several visualization applications have been deployed for this purpose such as the UCSC genome browser, Deeptools, and Segtools. However, we believe there is room for improvement in terms of programming skills requirements and proposed visualizations. Sigtools is an R-based exploratory visualization package, designed to enable the users with limited programming experience to produce statistical plots of continuous genomic data. It consists of several statistical visualizations such as value distribution, correlation, and autocorrelation that provide insights regarding the behavior of a group of signals in large regions – such as a chromosome or the whole genome – as well as visualizing them around a specific point or short region.

To demonstrate Sigtools utilization, first, we visualize five *histone modifications* downloaded from Roadmap Epigenomics data portal and show that Sigtools accurately captures their characteristics. Then, we visualize five *chromatin state features*, probabilistic generated genome annotations, to display how sigtools can assist in the interpretation of new and unknown signals.

Keywords: Genomic Signals; Data Visualization; Epigenomics; Histone Modifications; Chromatin State Features

Acknowledgements

My sincere appreciation goes to my supervisor Prof. Kay Wiese for the continued support, motivation, and knowledge throughout this research.

Special thanks also to Prof Maxwell Libbrecht for the encouragement and insightful comments. To the chair and examiner of my thesis, I am grateful for your constructive feedback and questions. Many thanks to the School of Computing Science at Simon Fraser University for giving me the opportunity to write a master's thesis. Last but not least, I would like to thank my family and friends, without whose love I would not have been able to complete my master's degree.

Table of Contents

List of Tables

List of Figures

Glossary

3D genome architecture In addition to its linear structure, the genome also has a 3-dimensional organization within the cell. Understanding this configuration is an essential step in studying gene regulation. 1

annotation Genome annotation is the process of locating different genomic elements on a genome. 6

assembly Genome assembly is the process of regenerating the original DNA fragment from the outputted sequenced reads. 6

ATAC-seq A sequencing technique that measures chromatin accessibility across the genome. 1

cell The basic structural component of living beings. It comprises several elements namely proteins and nucleic acids which are frequently referred to within this thesis. 1

ChIP-seq A sequencing technique for assessing protein interactions with the genome. 1

chromatin An organization structure for DNA. Consists of protein and DNA. ix

chromatin accessibility Describes how compact the chromatin is which indicates whether proteins such as transcription factors can bind to that piece of DNA. 1

chromatin state Just like histones, a chromatin's function can also be altered when the chromatin is modified by other molecules. To facilitate the study of chromatin modifications, scientists have classified them into separate states. 3

chromosome An organization of DNA. The entire genome of an organism can be arranged into one or many chromosomes. 17

contig Result of a genome assembly process. A long sequence of ordered reads produced by overlapping reads. 6

coverage The coverage of a DNA base denotes the average number of sequenced reads mapped to that base. 1

DNA A double helix molecule that can be considered as a library that carries the entire viability information of an organism. It comprises four bases: A, C, G, and T. These bases are also called nucleotides. 1

DNA methylation When a methyl molecule attaches to a DNA segment. Plays an important role in gene expression. 1

dominant trait A trait that will definitely appear in an offspring once he/she inherits it from his/her parents. 2

downstream A single DNA strand has a direction much like a one-way street. For a DNA strand that contains a gene, the downstream region denotes the side of the gene which is closer to the end of the strand. 3

enhancer A short genomic element that enables expression level increase for a particular gene. xi

gene A strand of DNA that codes for an RNA or protein. x

gene expression The process during which instructions in a gene are converted to functional molecules such as proteins. 2

gene regulation When the expression level of a gene is increased or decreased. 1

genome The complete chain of an organism's DNA. 1

genomic A field in biology. The study of the genome and its contents. iii

genomic feature A segment of the genome that has a function. 6

Genotype-Tissue Expression A genomic dataset that contains gene expression in multiple tissues across people. viii, 7

histone modification When another molecule attaches to a histone protein and the function of that histone is altered. 3

interactional Related to protein interaction. 1

locus A specific location on a chromosome where a gene is positioned. 2

multiple sequence alignment Arranging two or more sequenced reads in a pile to find their similarities. 2

nucleotide A building unit molecule for DNA and RNA. 1

omics A branch of science comprises biology fields that end in *-omics* such as genomics. 6

protein A chain molecule responsible for many vital tasks within an organism. 1

protein interaction When two or more proteins establish physical contact. 1

protein-coding gene A gene that contains the instructions regarding producing a particular protein. 1

read A limited sequence of DNA base pairs that a sequencer machine obtains from a single DNA fragment. A complete sequencing process results in the generation of millions of reads. 1

regulatory element A genomic element that is able to increase or decrease the expression level of a gene. 1

RNA A molecule with a single chain of nucleotides that is responsible for many vital tasks in a cell. x

RNA-Seq A sequencing technique that measures the number of existing RNAs in a given sample. 1

sequencing device A machine that is able to determine the order of the base pairs in a given DNA sequence. 1

single nucleotide polymorphism The variation of a single nucleotide between two DNA fragments that belong to the same position in two different individuals of the same species. 6

super-enhancer A genomic element with multiple enhancers. It has a more significant effect on gene expression than a single enhancer. 2

tissue A body of cells that have similar structure and functions. 2

trait A specific inherited characteristic in a living being. x

transcription binding factor A protein that binds to specific genomic elements and increases or decreases the amount of expression of the adjacent gene. 2

transcriptome The collection of all the transcribed DNA in a cell at this moment. 1

upstream A single DNA strand has a direction much like a one-way street. For a DNA strand that contains a gene, the upstream region denotes the side of the gene which is closer to the beginning of the strand. 3

Chapter 1

Introduction

Within the cell(s) of any organism lies its genome, the entire chain of its DNA that contains all the instructions regarding the viability of that living being. Understanding the structure, behavior, and interactions of those DNA contents has been the objective of hundreds of thousands of studies. Such studies are possible by DNA sequencing technologies that are able to determine the order of the nucleotides (A, C, G, and T) in a DNA sequence, generating various large datasets namely *genomic signals*.

A genomic signal is a continuous variable across the genome indicating the presence of a biological activity such as protein interaction, DNA methylation, and regulatory elements [10]. These activities are called *epigenetic* factors which are environmental changes that modify the genome without changing its underlying DNA sequence [21]. To obtain a signal in a lab, a high-throughput sequencing device sequences DNA fragments bound to a certain protein, hence generating a large number of short reads which are then mapped back to the original genome. Genomic coordinates associated with the coverage measurements of this mapping is recorded as a genomic signal. RNA-Seq, ChIP-seq and ATAC-seq are technologies specifically designed to generate transcriptome, interactional, and chromatin accessibility signals respectively.

Genomic signals are particularly intriguing since they can be utilized in a wide range of studies such as locating protein-coding genes [16, 18], investigating gene regulation in cancer research [14], and understanding 3D genome architecture [23]. However, Sequencing technologies are improving to such an extent that data analysis has become the bottleneck of genome-related studies [25].

The Human visual system provides a great bandwidth for image transmission to the human cognition system, which is excellent at detecting patterns and comparing graphical figures; given that the data has been appropriately mapped to the existing visual channels. Data visualization systematically develops refined presentations of data that would be more comprehensible to the human brain, which would result in acquiring new insights into data and faster analysis, problem detection, message communication and other tasks that could benefit from human involvement.

Several visualization tools have been developed to leverage the human visual system in genomic signals behaviors and characteristics investigation. Many of these publications, namely *browsers* [40, 26, 8], preserve the sequential nature of these signals by presenting them in a linear layout, possibly with parallel arrangements to enable comparison between signals [27]. Such tools are commonly used to investigate local behaviors around specific regions, for example, to depict regulatory elements near a particular gene [13] or displaying read numbers for different signals at a specific locus [34].

Others employ statistical procedures to illustrate the global behavior of the signals [31, 36]. Accordingly, this class of tools usually work with multiple sequence alignment data formats (SAM/BAM), rather than continuous-valued data formats (WIG/bigWig/BED/bedGraph) which contain the actual –or normalized– value of the signals at each position or bin. Example application of these tools include plotting DNA methylation average values over protein-coding genes to investigate their role in organism development [7], generating profiles of transcription binding factors over super-enhancers regions to discuss super-enhancer's cell-specific impacts on genetic risk of disease [39], and illustrating gene expression levels for multiple tissues to examine the cause of dominant traits [30].

Although the reviewed tools above offer a wide selection of processing and visualization features for genomic signals analysis, we identified the need for a set of tools that assist scientists in the early steps of genomic signal analysis, such as the value range, variation, and covariation. Novel genomic signals are being generated both in wet labs by biologists or via learning models by computer scientists. When someone in either of those professions comes across a new signal, some common questions need to be discussed before this signal could be introduced to the genomic society and be employed in other studies: what numeric range does this signal cover? Does it contain noise? If positive, how much noise is there in the data? How much data variation does it have? Does it behave similarly to any previously studied signal? How does it behave in general around specific genomic elements?

Interpreting a signal is to discover which biological activity it represents. Afterward, these signals can be direct representatives of a class of genomic elements or they can be utilized in learning genomic annotation models to identify the location of genomic elements.

This thesis introduces SigTools, an R-based package with four visualization modules for facilitating genomic signals statistical characteristic exploration and global behavior analysis. 1) `sigtools_distribution` module offers three recognized distribution plots for depicting value frequency: empirical cumulative distribution, kernel density plot, and boxplot. These plots uncover insights for later model training, refinement, and development. 2) `sigtools_autocorrelation` module generates a line plot of the input signals' autocorrelation. This plot provides an estimation for the change rate of signal variation. A signal with little variation indicates smaller active regions. 3) `sigtools_correlation` module generates a heatmap of pairwise correlation of for sets of signals. A high correlation coefficient between two signals indicates a high behavioral similarity in the sense that wherever the

value of one of the signals increases, an increase in the other signal's value should be observed. 4) `sigtools_aggregation` module illustrates the overall behavior of a signal over recurring genomic elements. Answering questions such as "does the signal have a high value within gene regions?" "How does the signal behave over enhancer regions?" "Does the signal have a distinguished behavior over genes upstream or downstream?" could be most helpful for the signal's interpretation relative positions. In addition to the visualization modules, SigTools offer several data processing modules to make the tool compatible with different genomic signals format.

Although we implement a novel encoding modification for aggregation plots (read more about this in Subsection 3.2.2), the contribution of this thesis is not in the field of Human-Computer Interaction or visualization as we did not invent a new visualization for genomic signals. The objective of this thesis is to provide a cohesive package that contains all the essential data analysis tasks for scientists who are working with novel genomic signals and do not want to carry out an additional coding workload to their project. Furthermore, this thesis also introduces SigTools-Shiny, a web-based graphical user interface that includes all SigTools preprocessing and visual modules as an alternative for users who want to eliminate command-line interaction in their experience.

This thesis is written from a computational biologist's perspective therefore some frequently used vocabularies are obtained from the field of biology, hence might not be familiar to the general computer science audience. A brief explanation of those terms is provided in the glossary section so that this thesis could reach a broader range of audiences.

Chapter 4 includes two use cases to demonstrate SigTools visualization modules' utility. The first use case examines several previously studied genomic signals named histone modifications. This use case is mentioned to demonstrate how SigTools can be beneficial for satisfying scientists' curiosity in exploring and establishing recognized datasets. The second use case examines a dataset of novel chromatin state features which are novel genomic signals generated by a learning model[4]. This use case demonstrates how SigTools can assist in exploring the characteristics and behavior of novel signals towards their interpretation.

The SigTools source code, installation guide, and manual are available on `http://github.com/shohre73`.

The following chapter, Chapter 2, describes the genomic signals data type, format, and characteristics in greater detail together with mentioning some prominent instances of their visualization. Chapter 3 contains SigTools© proposed user tasks and example use cases of how SigTools fulfills them. Afterward, presentation and interaction choices in the SigTools-Shiny app are explained in Chapter 4. Finally, Chapter 5 reviews the tool, its contribution, and future advancements.

Chapter 2

Data and Visualization Approaches

2.1 Genomic Signals Data and Formats

Sequencing Technologies perform a series of physical and biochemical operations to detect the order of nucleotides in a given DNA sequence. Currently, these devices are not able to record the entire given DNA sequence with a single scan. Instead, they generate a great multitude of sub-strings obtained from the many identical copies of the original DNA.

These sub-strings of ordered DNA bases are called *reads* and are banked in text-based formats, FASTA and FASTQ. A brief example of a FASTA file is displayed in Fig. 2.1a. In practice, the size of such files may exceed 1 Gigabyte.

Sequenced reads constitute the underlying data for genome-related studies. Like pieces of a puzzle, these reads also have similar edges. To *align* these reads is to identify their overlaps, and it is the principal method for reducing the size and complexity of their datasets. Fig. 2.1b is an example alignment of reads in Fig. 2.1a. Sequence Alignment/Map (SAM) [22] and its binary equivalent BAM are formats specifically designed for systematically maintaining the alignment of multiple reads to a single reference sequence. Binary formats are compressed versions of their text-based formats that provide faster access to the data. See an example in Fig. 2.1c. In this context, the *coverage* of a position is defined as the number of sequences that extend over that base.

Rather than aiming for uniform coverage, some sequencing methods such as RNA-Seq, ChIP-seq, and ATAC-seq, target specific DNA fragments that are bound to certain proteins and generate a mass of short reads from these isolated regions. Mapping these reads back to a reference sequence results in distinct coverage measurements across its bases. A *genomic signal* contains the exact or normalized values of such coverage. The two most common formats for genomic signals are WIG –or its binary equivalent bigWig– and bedGraph.

```
>ref
AGCATGTTAGATAAGATAGCTGTGCTAGTAGGCAGTCAGCGCCAT
>r001
TTAGATAAAGAGGATACTG
>r002
AAAAGATAAGGGATAAA
>r003
AGCTAA
>r004
ATAGCTCTCAGC
```

(a) An example FASTA file. Lines starting with '>' indicate the name of the subsequent sequence.

```
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT
+r001          TTAGATAAAGGATA*CTG
+r002                AAAAGATAA*GGATA
+r003       GCCTAAGCTAA
+r004                            ATAGCT..............TCAGC
```

(b) An example alignment of the reads in Fig. 2.1a.

```
@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r001 0  ref 7  30 8M2I4M1D3M * 0  0  TTAGATAAAGGATACTG *
r002 0  ref 13 30 3S6M1P1I4M * 0  0  AAAAGATAAGGATA    *
r003 0  ref 4  30 5S6M       * 0  0  GCCTAAGCTAA       *
r004 0  ref 22 30 6M14N5M    * 0  0  ATAGCTTCAGC       *
```

(c) A SAM file can start with an optional header followed by an alignment section. The header gives information about the overall alignments while each line in the alignment section contains information about one specific alignment such as its start position and mapping quality.

```
chr start end signal_value
X   1     3   0
X   4     6   1
X   7     12  2
X   13    14  3
X   15    12  2
X   23    25  2
X   26    28  1
X   29    42  0
X   43    45  1
```

(d) An example bedGrph file.

Figure 2.1: Examples of a FASTA file, multiple sequence alignment, SAM file, and a bedGraph file, inspired by an example in SAMtools documentation.

2.2 Proposed Visualizations

> Computer-based visualization systems provide visual representations of datasets intended to help people carry out some task better. (Munzner [24] p. 3)

As Munzner defines, data visualization provides graphical representations of data to enhance the human comprehension of data parameters. This approach is widely employed in omics studies to empower scientists with a deeper understanding of the voluminous and complex data they are engaging with [12, 28, 27].

Likewise, numerous visualization tools have been developed for genomic signals interpretation and pattern discovery. We classified these developments into two groups of Locus-Specific and Locus-Agnostic visualizations. The rest of this section discusses these two categories in detail and displays some of their instances.

2.2.1 Locus-Specific Visualizations

Visualization tools that map the genomic data to a horizontal axis are commonly denoted as *genome browsers* and are frequently employed for analysis, discussion, and comparison of genomic signals over specific coordinates. Interactivity is one of the essential design elements of these tools since, at each time, only a limited window of data appears on the screen. Navigation (moving back and forward on the genome), zoomin in and out, and selection are common classes of interaction established in these tools.

The UCSC genome browser [40] is a well-known genome browser for visualizing genomic signals together with other genomic feature sets such as annotations, genes, single nucleotide polymorphisms (SNPs), and assembly contigs. These sets are aligned horizontally to create a parallel arrangement for enabling comparison at different resolution levels. Each horizontal strip is called a track, and users can add, remove, or displayed in different modes (showing different levels of details). This genome browser divides the screen into three sections. At the top, there are navigation controls for moving forward or backward, and zooming in or out. Users can also jump to a specific coordinate using the search bar. The dynamic middle section displays genomic feature sets (tracks). The bottom section contains additional optional control for modifying the middle section.

Jbrowse [5] and Ensembl Genome Browser [17] are other instances of genome browsers.

Genome browsers provide an excellent platform for genomic data integration. Yet, due to the extensive length of the genome, visualization tools with a linear layout are commonly employed to display specific genomic coordinates. To reach a general conclusion via these tools requires the user to go through all the instances of occurrence of that specific element. Extending the scale of genomic data analysis enables data comparison across multiple regions simultaneously.

Figure 2.2: A screenshot of the UCSC Genome Browser, extracted from its 2017 update [38], displaying eight RNA-seq tissue-specific read coverage tracks and bar graphs of Genotype-Tissue Expression (GTEx) for five protein-coding genes.

2.2.2 Locus-Agnostic Visualizations

Instead of preserving the sequential nature of the genomic data, global visualizations tools employ statistical methods and aggregation plots in order to support such analyses.

Prior to the introduction of such tools, we ought to review some frequently employed visualization keywords:

- Heatmaps: a heatmap is a two-dimensional matrix with specific categories assigned to its rows and columns. Each row-column pair is associated with a value encoded with color, and the value variety is represented by different hues or intensities. Rows and columns can have either specific or arbitrary ordering and they can be clustered according to the similarity of their vectors.
- Aggregation plot: in the context of genomic signals, an aggregation plot for a specific genomic element is a heatmap generated to display how a signal behaves in regards to that element throughout the genome. Each row of this heatmap is a vector indicating the signal's value over a specific occurrence of that element and each column represents the relative position of each cell to the center or edges of that element.
- Aggregation line chart: a summary line chart is mostly associated with an aggregation plot, and it outlines the commonly observed behavior of that signal over the selected regions. Its x-axis corresponds to the columns of the aggregation plot, and its y-axis corresponds to the signal value.
- Scatter plots: a two-dimensional scatter plot consists of two orthogonal axis and dots that represent the data. The x axis and the y-axis correspond to the value of the first and the second variables, V_1 and V_2, respectively. Each dot corresponds to a pair of set $\{(v_1, v_2)|v_1 \in V_1 \& v_2 \in V_2\}$.
- Correlation plots: another common analysis regarding two signals is to investigate whether it is possible to infer the changes of one signal from the value alteration of the other signal. If two signals positively correlate to each other, and the value of one of them is being increased, it can be included that the value of the other signal is also rising. A heatmap for pairwise correlation values is frequently used in comparative genomic signals analysis.

One of the earliest tools in this category is deepTools [31], a collection of tools developed for exploration, quality control, and visualization of the next-generation sequencing data. Accordingly, its primary input formats are the aligned reads formats (SAM and BAM, see Section 2.1) though, it also accepts and generates coverage formats such as BigWig in certain cases. See examples of deepTools analysis in Fig 2.3. To obtain a visualization in deepTools, users usually need to perform two operations. The first one outputs the underlying computation of the intended visualization, usually in the form of a matrix. The next one maps that computation to a graphic representation. An example of such workflow is to perform `plotCorrelation` after `multiBamSummary` to obtain a correlation matrix of

different coverage samples (Fig 2.3c). Suchlike workflow eliminates repetitive computations in cases where figure regeneration only applies changes in visual parameters. However, it increases the overall complexity of the tool. See Table 2.1 for detains on deepTools command. deepTools is also part of Galaxy [11], a web-based platform for biomedical data investigation, to facilitate accessibility and integration with other NGS data analytics tools.

genomation [1] is an R based assisting studies investigating the association of short-read coverage with discrete genomic features denoted here as genomic *intervals*. It defines multiple functions for importing genomic intervals formats (BED and GFF), detecting overlaps between a new set of genomic intervals and an existing one, visualizing that overlap with summary and heatmap aggregation plots. Fig 2.4 displays an instance of a heatmap generated by genomation. In addition to profiling plots, ngs.plot [36] also provides a database of genomic elements to facilitate region selection. Fig 2.5 shows an example application of this tool.

SegTools [6] enables the interpretation of probabilistic generated genomic segments by facilitating their comparison with known genomic signals at a genome-wide scale. During interpretation, a genomic *label* is assigned to different genomic segments (with different lengths) and declares the role of those segments. SegTools encourages short-read independent segment and signal analysis by using genomedata [15] as the input format for genomic signals. See example of SegTools analysis in Fig 2.6. SegTools commands either provide analysis for a single genomic segment set or enable comparing between a segment set and other related data sets (Table 2.2 and Table 2.3 respectively). You can find a brief description of SigTools Utilities in Table 2.4.

Exploratory Data Analysis (EDA) is an essential step in any data-dependent study that highlights data anomalies, patterns, and provides a deeper understanding of the data.

All the mentioned tools above deliver plots that facilitate such analyses for genomic signals and several other relevant datasets (NGS short reads and genomic intervals). Yet, there are still some aspects of this data that either have not been discussed or have not included a systematic package. Our proposed package, SigTools, contains new modules that uncover other aspects of this data such as range, shape, variation, and covariation. The following list describes each of the mentioned terms:

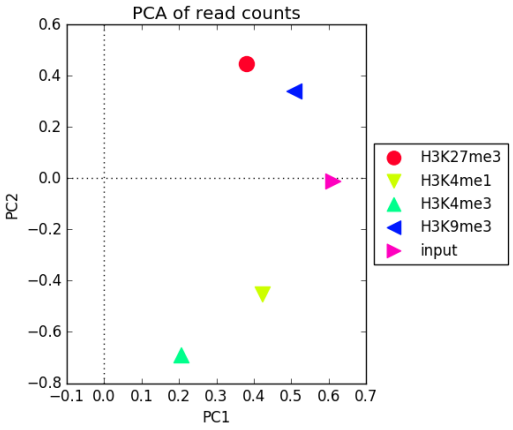
- Data range: the statistical range of a variable is a measure of spread and is defined as the difference between its highest and lowest values. In case the dataset contains extreme values (outliers), other measures of dispersion such as Standard Deviation should be employed.
- Data shape: the shape of a dataset describes how it is spread across a distribution graph such as a histogram or a kernel density curve. The shape of a quantitative variable can be symmetric or skewed, with one or multiple peaks. This helps us to decide which average, mean, or median values best describe the variable's center, or informs us about gaps and outliers in the data.

- Data variation: variability describes the extent to which data values are different than each other. Weather data points of a variable are accumulated around particular areas or contain a wide range of values is a good measure for comparing different variables.
- Data covariation: the covariance measure describes the relationship between two variables in terms of the expected changes in one variable when the other one changes.

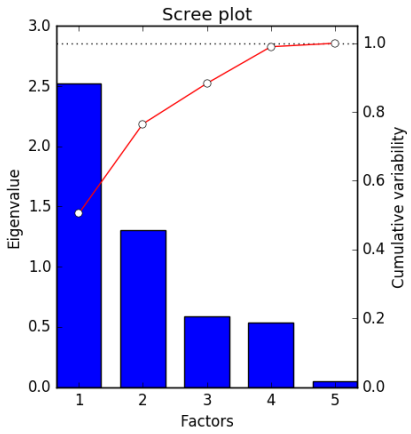
In addition, our corresponding web application, SigTools-Shiny, extends the accessibility scope of these modules to people who are more comfortable working with graphical user interfaces instead of command-line tools.

Table 2.1: Brief descriptions of a number of deepTools modules.

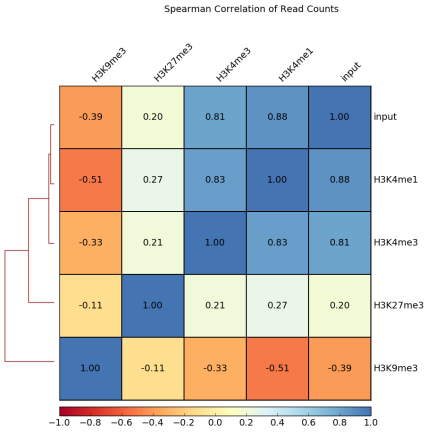
Module name	Description
`multiBamSummary`	Computes the read coverage of one or more BAM files. The coverage measurements are either calculated over regular size windows (bins) over the entire genome, or for specific genomic regions. The second option requires providing a BED file. The output of this module can be forwarded to `plotPCA` or `plotCorrelation` for visualization.
`plotPCA`	Computes and plots principal component analysis (PCA) for read coverage obtained from the `multiBamSummary` module. PCA enables detecting similar clusters in multi-dimensional data, see an example in Figure 2.3a and Figure 2.3b.
`plotCorrelation`	Computes and visualizes the pairwise correlation of samples coverage obtained from `multiBamSummary` module either in the form of a table, scatter plots (Figure 2.3d), or heat-map (Figure 2.3c).
`plotFingerprint`	Plots cumulative coverage over sampled bins. See an example in Figure 2.3g. This plot was specifically proposed for ChIP-seq quality control.
`plotCoverage`	Aggregates the coverage for 1 million sampled base-pairs and generates two plots, the first one displays the distribution curve of this coverage. and the second one is a reverse cumulative distribution(Figure 2.3g).



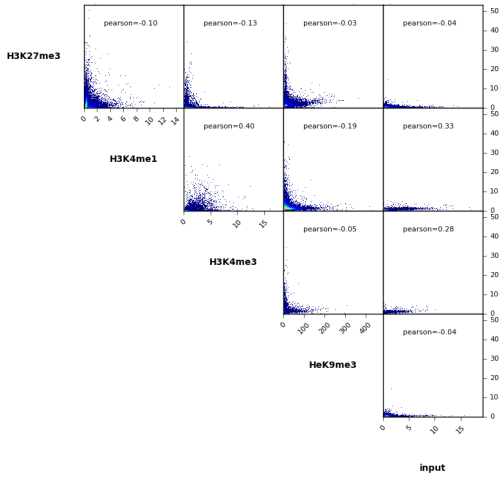
(a) Generated by `plotPCA`, this PCA plot indicates that H3K27me3 and H3K9me3 are very similar in terms of read count over the addressed regions.



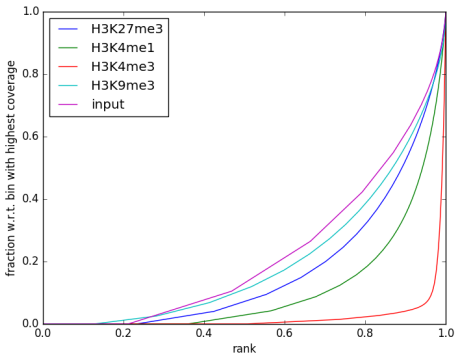
(b) A scree plot generated by `plotPCA`. A scree plot generated by `plotPCA`. The blue bars indicate the amount of variability each PC accounts for. For example, PC1 explains 87% of the dataset variability. The red lines indicate cumulative variability.



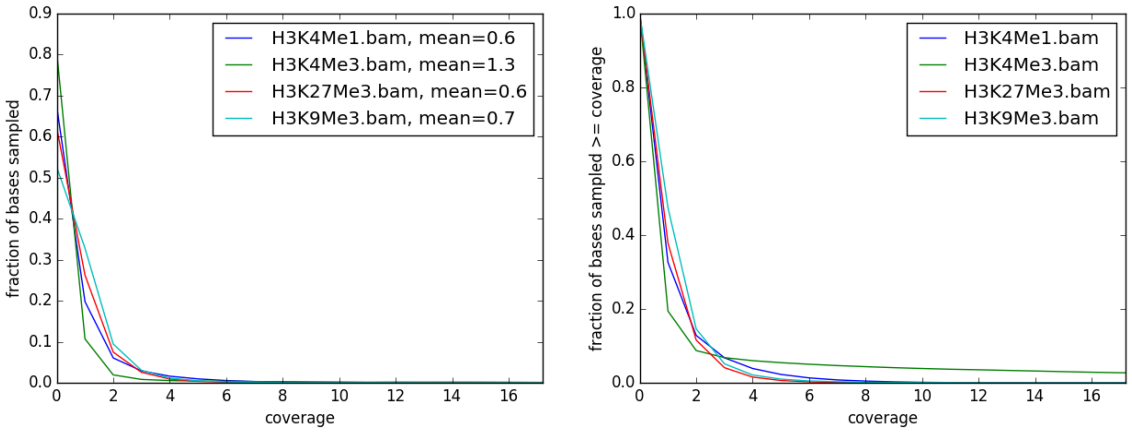
(c) Spearman Correlation heatmap of reads counts for an input sample and several previously studied read coverages. The colors represent correlation coefficients and rows are clustered based on similarity.



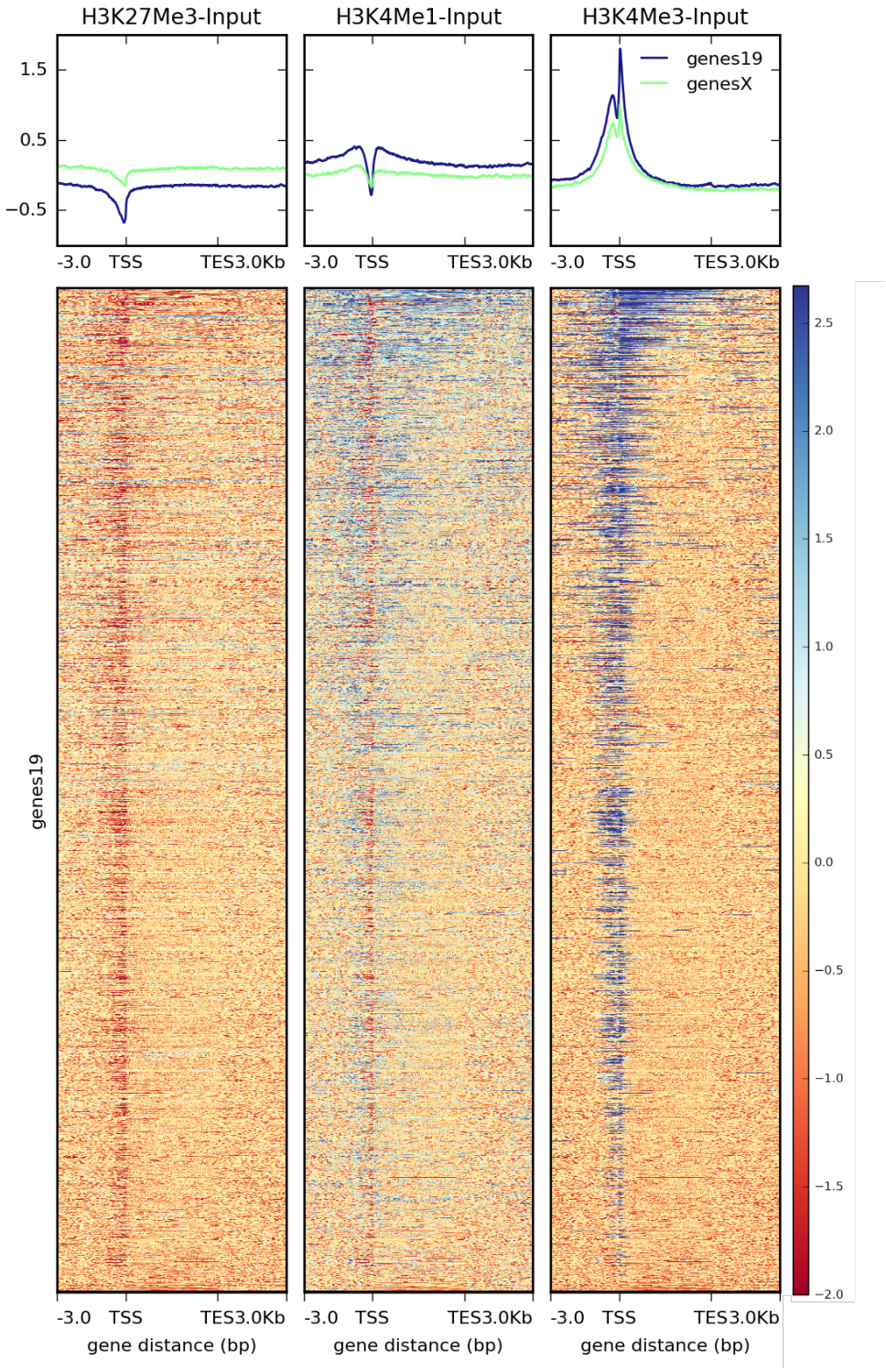
(d) Pairwise scatter plots of samples with Pearson correlation coefficient.



(e) The fingerPrint plot displaying that for H3K4me3, about 97% for the sampled bins have the 5% of the maximum number of reads.



(f) deepTools' `plotCoverage` plots. The left plot indicates that about 1% of the H3K9Me3 sample base-pairs were covered by 2 reads. And the right plot denotes that about 5% of the H3K4Me3 sampled base-pairs were covered at least 16 times.



(g) This module displays the value of the input sample over every recurrence of a specific genomic element. Here you see summary plots (at the top) and heatmaps for three histone modifications over genes at chromosome 19.

Figure 2.3: deepTools visualizations. Reprinted from deepTools: tools for exploring deep sequencing data by the Bioinformatics Facility at the Max Planck Institute for Immunobiology and Epigenetics, Freiburg. Retrieved from https://deeptools.readthedocs.io/en/develop/.

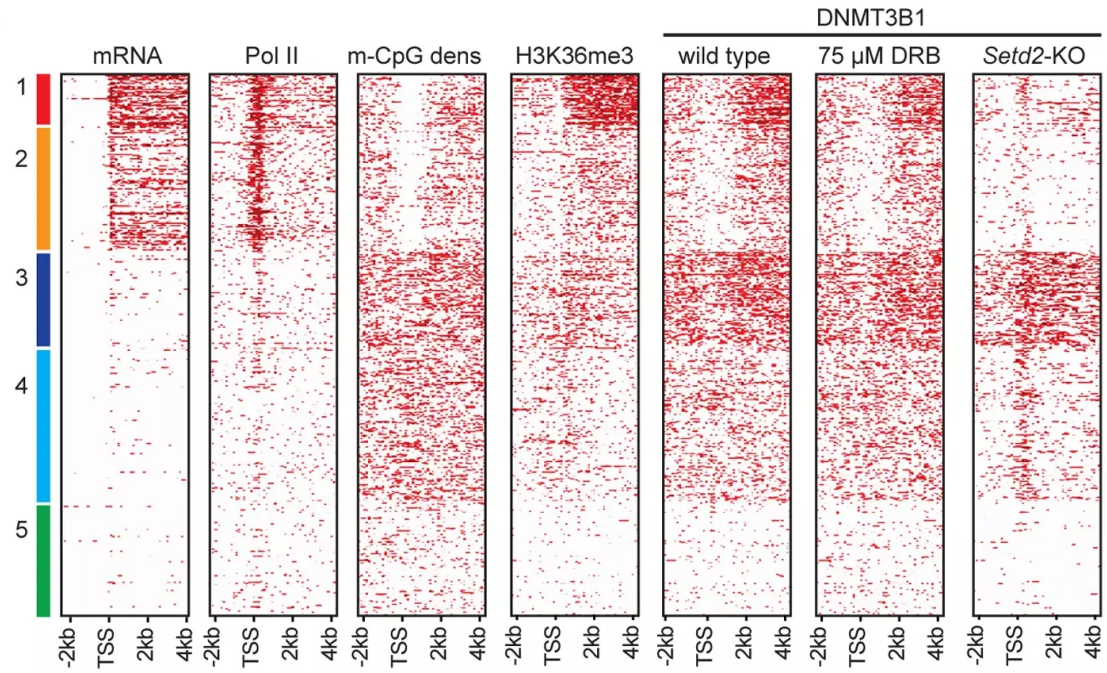


Figure 2.4: genomation application. Heatmaps of coverage values around genes, clustered based on k-means [3].

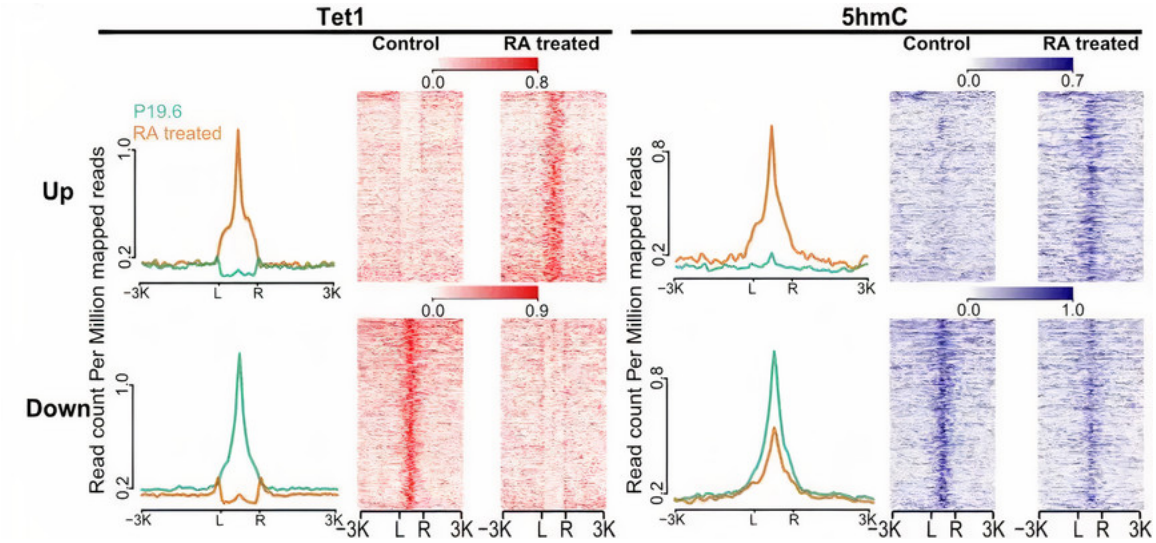
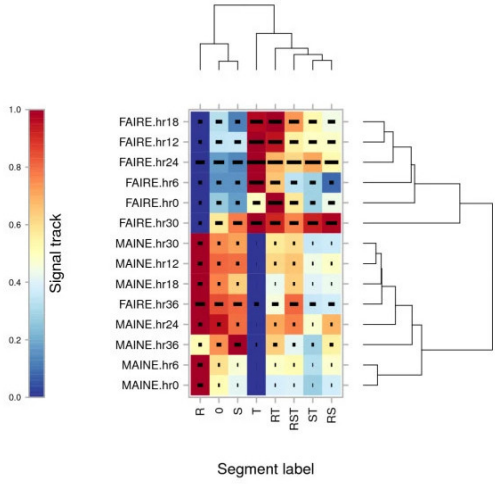
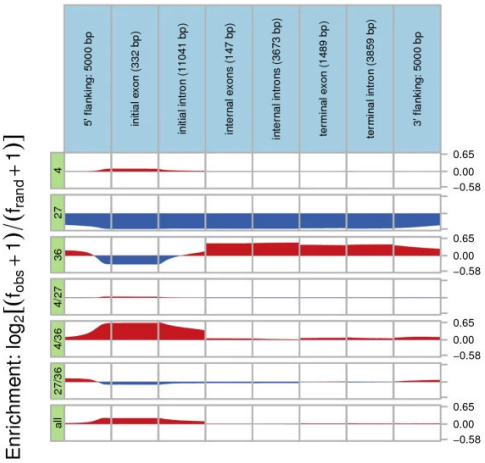


Figure 2.5: ngs.plot application. Comparison between the enrichment of two DNA compounds (Tet1 and 5hmC) at regions showing different Tet1 enrichment before and after a particular treatment [36].



(a) A heatmap with genomic signals as rows and genomic segments as columns. Colour indicates the mean value of a signals associated with specific labels.



(b) Relative occurrence of specific segments over established genomic segments.

Figure 2.6: Examples of SegTools visualizations.

Table 2.2: SegTools commands, single file analyses.

		Commands
Segment analysis	Length	`length-distribution`
	Order	`transition`
	Signal	`signal-distribution`

Table 2.3: SegTools commands, multiple file analyses.

	Commands
Segments & Sequence	`nucleotide-frequency`
Segments & Segments	`compare`
	`flatten`
Segments & Annotations	`aggregation`
	`overlap`
	`feature-distance`

Table 2.4: Brief descriptions of SigTools utilities. Each module outputs a figure and a text file containing the figure's related statistics.

Command name	Description
`length-distribution`	Generates a violin plot that discusses the length distribution for each discrete genomic labels, a horizontal bar chart indicating the fraction of base-pairs that each label covers, and a text file containing statistical measures such as mean, median, and standard deviation length of segments, number of segments, number of covered base-pairs, the fraction of covered base-pairs for each label.
`transition`	Outputs a graph with weighted edges and a heat-map indicating consecutive labels. Every time a segment with label Y appears after a label X, it is counted as a $X \to Y$ transition.
`aggregation`	Given a set of genomic elements (*annotations*) this command displays the number of times each label occurs at those positions. This command has multiple options such as `--normalize` to smooth the plot, `--mode` to be chosen from `point`, `region` or `gene`.
`feature-distance`	The *distance* from one genomic element to another is defined as the minimum number of bases between those two elements. Given two sets of genomic elements (for example, a segmentation and an annotation set) this command generates a histogram of all the calculated distances in addition to a tab-delimited text file containing the distances and elements.
`overlap`	Outputs a confusion matrix that indicates how much a probabilistic generated segmentation overlap with a set of established annotation. A *confusion matrix* describes how much an output of a predicting model meets or disregards the truth.
`signal-distribution`	Generates a heatmap, with signals on rows and labels on the vertical axis. The color indicates the mean value of a signal over each label, the black bar represents the standard deviation of that value. It also displays a clustering for similar rows and columns at the top and the right side of the heatmap. (See Fig 2.6a)
`flatten`	Given multiple segmentations, this command outputs a new segmentation that contains all the input segments in addition to new labels for regions covered by two or more segments.

Chapter 3

SigTools

Publications mentioned in Section 2 are all well-known tools enabling the exploration of genomic signals data sets. Along with the aggregation plots, which the most prominent visual representations for genomic signals analysis, Sigtools includes several modules that were previous neglected in developments. Additionally, SigTools-Shiny offers these modules through a graphical user interface.

3.1 Input Format

Recent studies [4] promote computational genomic signals to direct representatives of genomic elements rather than the mere indication of protein binding regions. Hence, SigTools pursues alignment-free genomic signal analysis by choosing the *multi-column bedGraph* (`.mulColBedg`) as its primary input format for the visual tasks. Multi-column bedGraph format is a tab-delimited text-based file that contains several signals associated with regular-size stretches over the entire chromosome (see an example in Figure 3.1). This format boosts later comparative tasks since it includes several signals data within one file, hence eliminating the need to work with multiple files.

```
chr start end H3K9me3 H3K27me3 H3K36me3 H3K4me3 H3K4me1 H3K27ac
21 37262800 37263000   0.399     0.000     0.000    0.000    0.000    0.000
21 37263000 37263200   0.069     0.000     0.094    0.353    0.000    0.391
21 37263200 37263400   0.314     0.222     1.127    0.051    0.000    0.206
21 37263400 37263600   0.592     0.501     0.211    0.026    0.960    1.562
21 37263600 37263800   0.413     1.955     1.074    0.595    1.257    0.363
```

Figure 3.1: A few lines of a multi-column bedGraph file.

3.2 Implementation and modules

A software package is a platform to organize several related functions and user documentation to create a unified set of tools to carry specific tasks. All SigTools' code and associated files are contained within an R package, hence facilitating sharing this utility. Additionally, the package automatically manages the dependencies and required libraries. Sigtools should be installed and loaded in an R session, which automatically get activated by opening Rstudio.

```
> install.packages(sigtools)
> library(sigtools)
```

SigTools' user manuals are maintained in `.Rd` formats under `man` directory, and are generated by roxygen2 which is an R package designed to generate standard user manuals. The `.Rd` is the format for *R documentation*, is automatically converted into LaTeX and HTML format when the package is installed. Either of the first two following commands will convert the `.Rd` file of the specified function to a readable document. The last command runs an example of that function. (See an example in figure 3.2)

```
> ?function_name
> help("function_name")
> example("function_name")
```

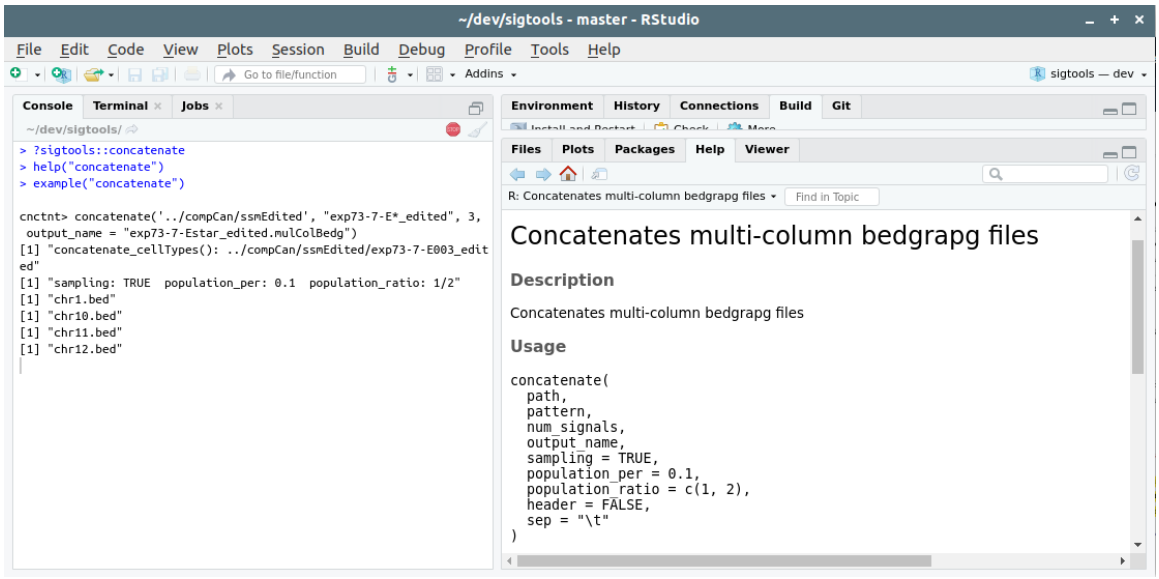


Figure 3.2: generated user manual for a particular SigTools module and a running example of that function.

3.2.1 Preprocessing

SigTools offers several modules that prepare the available data for future visualization tasks.

`sigtools_convertToMultiColBedg`

This module converts several bedGraph files with different bin sizes to a single `mulColBedg` file at the desired resolution. Changing the resolution of genomic signals data sets to a larger bin size is a beneficial strategy for reducing data points particularly for visualizations that discuss an entire chromosome or genome.

`sigtools_sampling`

Genomic signal data sets are generally large. Besides increasing signals' resolution (enabled by `sigtools_convertToMultiColBedg`), choosing random stretches of signals is a technique that can be employed to reduce signals' file size. This approach is particularly useful for obtaining quick results. `sigtools_sampling` provides two parameters to *enable* different sampling approaches:

- `pop_per`: the relative size of the sample to the original file. The default argument for this parameter is 0.01 which means unless indicated otherwise, the number of rows in the obtained sample will be one-hundredth of the number of rows in the original file.
- `pop_ratio`: for any given file, a generated sample constitutes a matrix that has the randomly picked stretches as its rows. Hence this matrix has as many rows as the count of the picked stretches and as many columns as the width of these stretches (all stretches have the same length). The sample matrix generated by `pop_ratio = c(2,1)` has twice as many rows as its columns. To have only one long random stretch from each file, the user can set the parameter to `NA`.

`sigtools_concatenation`

For a particular cell type, the signal data for different chromosomes is usually stored in separate files. This module appends multiple input files together and outputs a single large `.mulColBedg` file that can be used for whole-genome or multiple cell-type analysis. `sigtools_sampling` can optionally be incorporated into this process, preventing the final file to become too heavy.

`sigtools_stats`

This module outputs a `.csv` containing the five-number summary (min, lower quartile, median, mean, upper quartile, max) of the present signals in a `.mulColBedg` file. Although SigTools focuses on visual analytics, the text-based output of this module is beneficial in pipeline and learning model development.

3.2.2 Visualizations

SigTools employs ggplot2 to conduct its visualization tasks. Each of the following functions has a range of parameters that modify either the underlying data or the visualization. Here is the list of share parameters among all visualization modules and their description:

- `path_mulColBedg`: the path to the input multi-column bedgraph file
- `outdir`: name of the output directory [default: `function_name`]
- `header`: does the input file include headers? [default: `FALSE`]
- `prefix`: if the input file does not have a header, this prefix is use for naming the signals [default: `'s'`]. This is the name that signals will be represented with on SigTools plots.
- `sep`: the field separator character in the input file [default: `'\t'`]
- `img_title`: plot title.
- `font_size`: plot font size.
- `x_label`: the x axis label.
- `y_label`: the y axis label.
- `img_width`: image width.
- `img_height`: image height

`sigtools_distribution`

Depicting the value frequency of a variable is a quick approach to get an estimation of its primary characteristics; namely the existence of multiple local maxima, the overall range, and the outliers. SigTools generates several distribution plots for genomic signals. The Kernel Density plot prints a curve giving an estimation of the estimated recurrence count of observed values over a continuous data range, presenting the spread and shape of the signals.

$$kDensityDist(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sigma}K(\frac{x-x_i}{\sigma}) \tag{3.1}$$

In the empirical cumulative distribution plot or ECDF, for each value, the count of all instances is equal or less than the plotted value. This enables quick detection of most repeated numbers.

$$ECDF(x) = \frac{1}{n}\sum_{i=1}^{n}1_{x_i-x} \tag{3.2}$$

And the box-plot is best at presenting the summary statistics over a horizontal axis.

The options for the distribution plots are removing a certain upper percent of the data with `percentile`, removing the zero population with `nozero`, and plotting the enrichment distribution with `enrichment`.

`sigtools_autocorrelation`

SigTools includes autocorrelation plot as a measure of the dependency among consecutive bins, within the signals. Signals with higher autocorrelation have smoother picks and valleys. SigTools uses R `acf()` for autocorrelation value calculation. The autocorrelation plot is an orthogonal plot with x-axis presenting lags (shift), and y-axis presenting the value of autocorrelation at each lag, which is the correlation of the signal with itself when shifted *lag* times.

$$autoc(lag) = corr(sig, sig_{+lag}) \tag{3.3}$$

If the input data is the result of `concatenation` or `sampling` of several chromosomes, meaning the successive sequences may not necessarily be adjacent in the original data, the autocorrelation at each lag (shift) is the correlation of all the pairs of different sequences shifted by that lag.

Autocorrelation can also be used to measure the dependency of genomic signals among adjacent elements such as neighboring genes [9]. To use this module in such way, users can set the `mode` to `"regions"`.

`sigtools_correlation`

The correlation between a pair of variables calculates the linear association of those variables. SigTools `correlation` outputs a heat-map presenting the pair-wise correspondence of two sets of different signals.

$$\rho_{sig_1,sig_1} = \frac{cov(sig_1, sig_1)}{\sigma_{sig_1}\sigma_{sig_2}} \tag{3.4}$$

`sigtools_aggregation`

This feature inspects signal values upon every occurrence of a specific element for an entire chromosome or genome and illustrates that signal's general behavior. Such figures are essential in signal interpretation and assessment and have been suggested by multiple tools [29, 31, 1]. In addition to interval data retrieval and aggregation, the underlying computations for such plots need to perform data modification and normalization for the resulting plot to be meaningful and beneficial.

The primary information of the elements under examination – such as the chromosome they are located on, their starting and ending coordinate, and direction – should be inputted as a `.bed` or `.gene_info` file. To generate an aggregation plot for a signal S, an aggregation matrix needs to be computed. For each indicated element, an array of S's value over that element is retrieved. As the input elements vary in length, these arrays do not have the

same length, so unless they undergo an operation that unifies their length, they can not be assembled into a matrix.

The `mode` parameter controls the length unification process in `sigtools_aggregation` module. This parameter accepts two arguments: `point` and `region`. If `point` is chosen, all arrays will be centrally aligned and the smaller arrays will be padded with zeros. As indicated by the name, this option results in value accumulation over a specific point hence a biased conclusion over the edges of the elements. On another note, the `region` option unifies the arrays' length by stretching shorter arrays using smooth interpolation.

Our novel approach to the conventional aggregation plots is highlighting the differences between high and low signal values by adding a **shifted origin** line to the plot, and use different color encoding for upper and lower regions of this line. Different approaches can be pursued for choosing the value of the introduced shifted origin line (l), but not all of them disclose what is truly happening within that data set.

Genomic signals data sets generally contain a large population of zeros. Accordingly the mean value of a signal S is a small number close to zero. Setting the shifted origin to a signal's mean ($l = mean(S)$) will cause most of the aggregated points to stand above the line, hence resulting in an overrated aggregation plot.

Assigning the mean of non-zero signal values ($mean(\{s \in S | s > 0\})$) to l results in most of the aggregated points to stand below the l. A large count of zeros included in genomic signal data sets belongs to quiescent genomic regions, hence do not bear any more information than telling us these regions do not have *any* functionality. Yet, there still exist some important zeros that indicate the absence of a specific activity, and removing them results in an underrated aggregation plot.

Following the calculation of the aggregation frame (the average signal values over the specified repeated element), another potential choice for the shifted origin line (l) is the mean of the present values in the aggregation frame. However, the drawback of neglecting the signal values that did not participate in the creation of the aggregation matrix is that the shifted origin does not reveal any indications of how this signal is behaving over the specified elements in comparison to the rest of the chromosome or the genome.

Blacklist genomic regions have empirically shown to only commit artifact data in next-generation sequencing. Accordingly, the dismissal of these regions has proved to improve the result of several genomic signal related studies [2]. To capture an accurate aggregation plot, `sigtools_aggregation` offers an option for excluding the values that lie within blacklist regions, hence eliminating the bias that the large zero population or redundant extreme values introduce to the mean value.

The discussed approaches result in different shifted origins for different genomic signals while having a unified shifted origin value enables an effortless comparison between multiple aggregation plots. Towards this notion, `sigtools_aggregation` contains the option to gen-

erated the aggregation plot with the *enriched* signal, which is the input signal normalized by its mean value. The shifted origin in this case is set to be 1.

3.3 Web Application Interface

Command-line packages are necessary for pipeline development yet a large number of prompt parameters often discourage users to explore their data. Graphical user interfaces are ideal choices for users with limited command-line experiences to interact with their data. Shiny is an R package enabling the assembly of an interactive web-based user interface from R scripts [33]. The increasing number of Shiny apps in data visualization, particularly genomic data visualization [41, 19, 32, 42], indicates the effectiveness of this approach in enhancing the accessibility of developed packages. All SigTools utilities are also embodied in an interactive Shiny web application.

To start SigTools Shiny application, users need to click on the project file `sigtools-shiny.Rproj` to open Rstudio, then click on the *Run App* button or enter the following command in Rstudio's console:

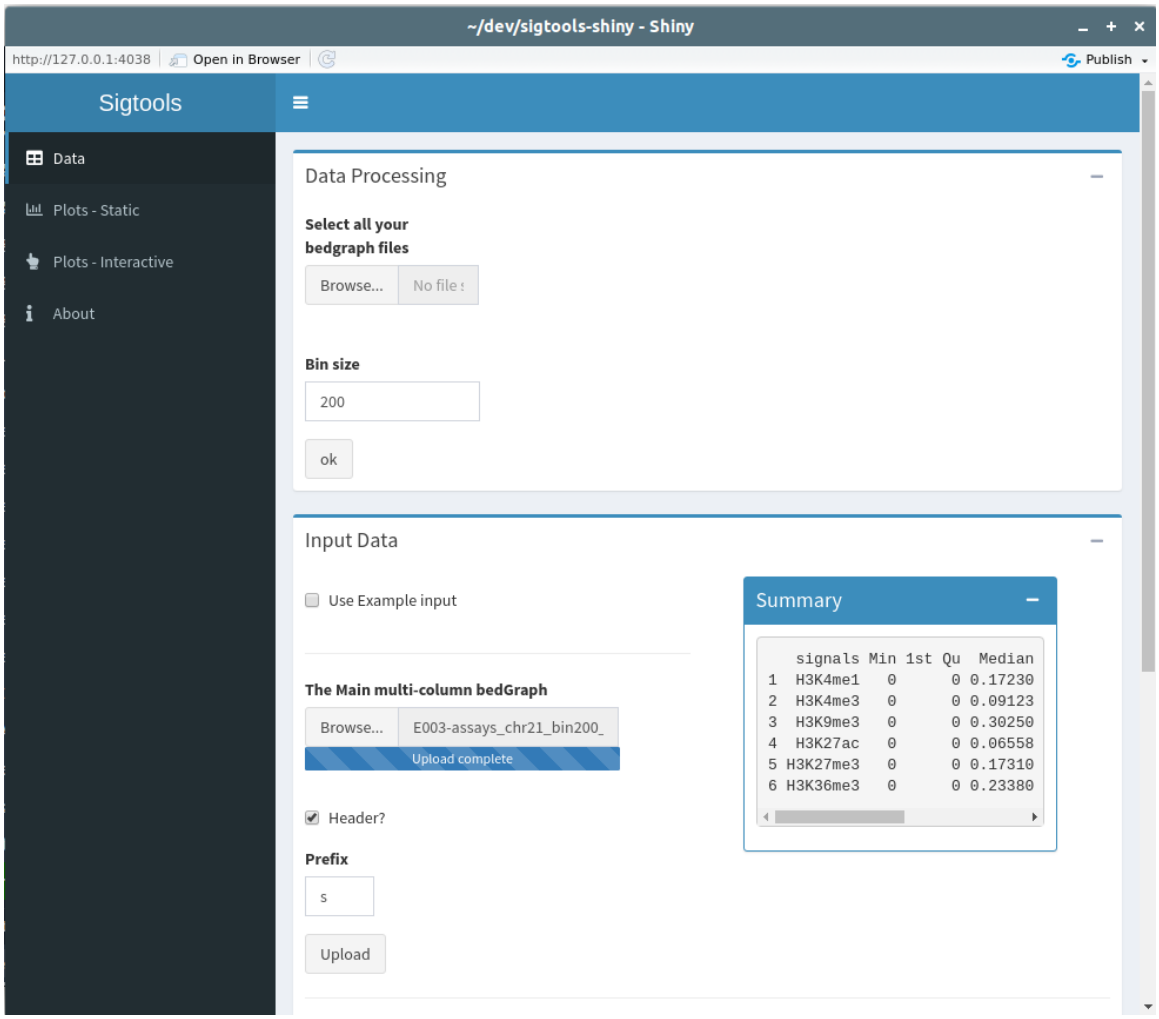
```
> runApp('sigtools.R')
```

SigTools-Shiny is structured based on a Shiny *dashboard* which consists of a vertical navigation bar and a main body. The navigation bar provides access to SigTools-Shiny's four pages: *Data*, *Plots-static*, *Plots-Interactive*, and *About*. By clicking on each of these sidebar items, their corresponding page appears in the main body.

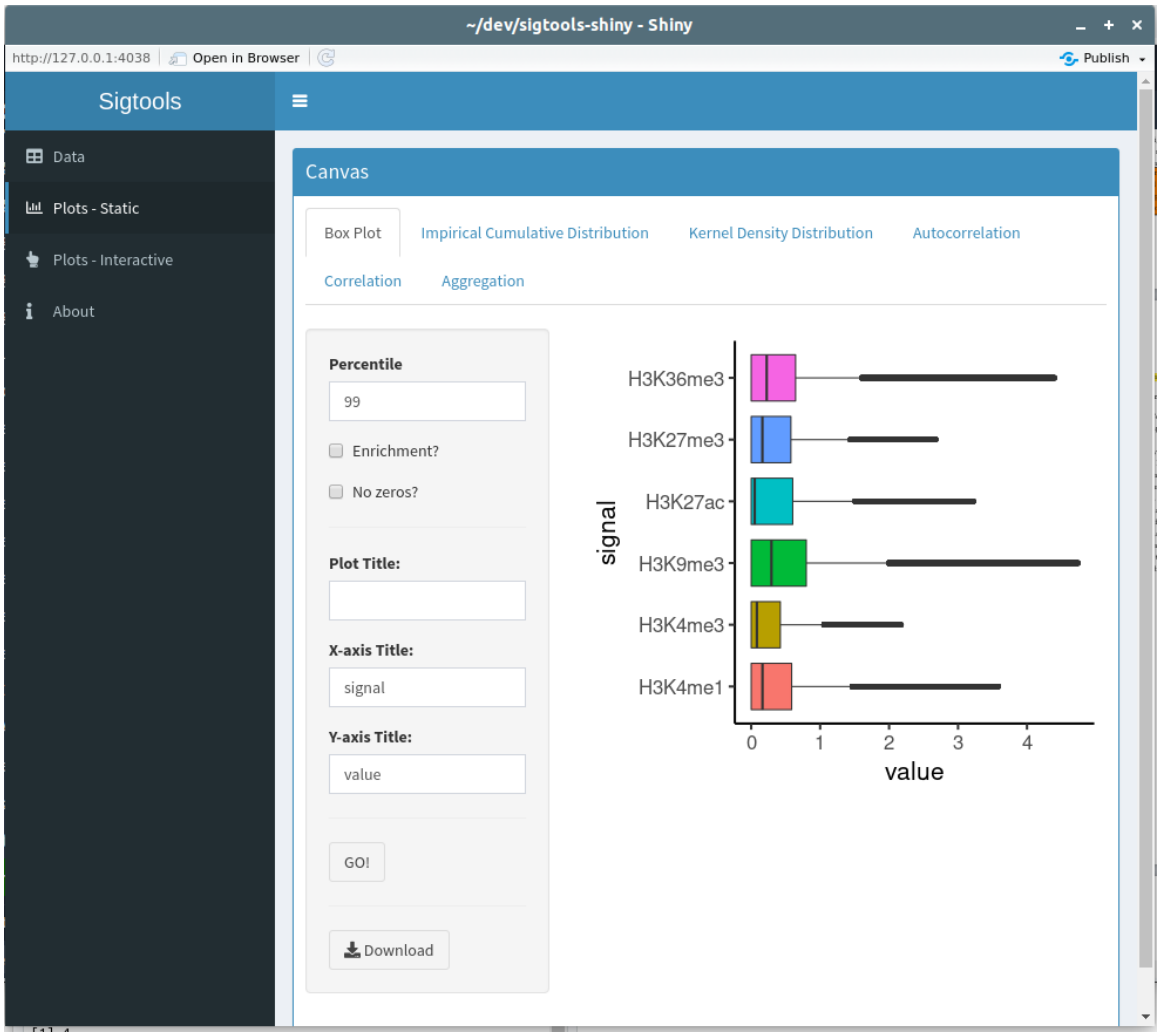
The *Data* page (Figure 3.3a) contains several boxes, one for each data operation discussed in Section 3.2.1 and one for data import controls. When a file is uploaded, the five-number summary of its contents appears in a box on the right side of the window, confirming that the uploading process was successful. This summary file can be downloaded using the `Download` button.

The *Plots-Static* page (3.3b) contains only one box named *Canvas* which contains several tabs each for a specific plot: *Boxplot*, *Impirical Cummulative Distribution*, *Kernel Density Distribution*, *Autocorrelation*, *Correlation*, and *Aggregation*. Each tab has a sidebar and a main panel. All the data and plot modification options are located on the sidebar panel. The `GO!` button initiates the plot generation process and eventually the plots is displayed on the main panel.

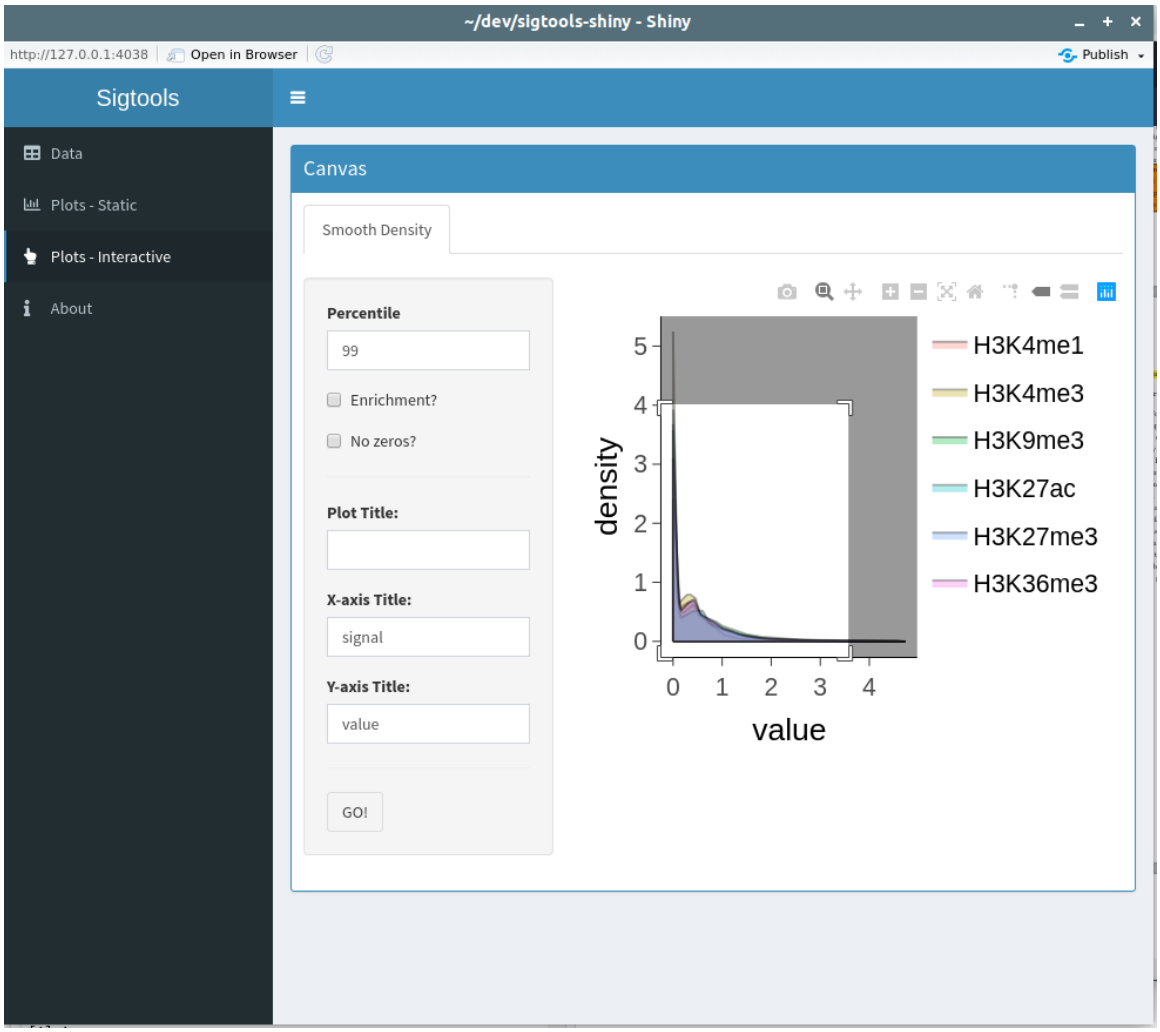
Much like the *Plot - Static* page, the *Plot - Interactive* page (Figure 3.3c) also consists of only one canvas, with multiple tabs. Yet, the plots in this page offer interactions such as zooming in and out and data selection. These plots are generated by Plotly, an R package that enables creating interactive web-based figures.



(a) A screenshot of SigTools-Shiny's first page, `Data`. A click on the `Browse` button opens an upload control and the users can select their file. Here the successful upload of a multi-column bedgraph file with a header is indicated by the summary table displayed on the right.



(b) A screenshot of SigTools-Shiny's second page, `Plots - Static`. A click on the `GO!` button generates a static box plot from the 99th percentile of the data, including zeros, with "signal" as the y-axis label and "value" as the x-axis label. The `Download` button opens a download control, allowing to store the generated image in a chosen location.



(c) A screenshot of SigTools-Shiny's third page, `Plots - interative`. A click on the `GO!` button generates an interactive distribution density curve from the 99th percentile of the data, including zeros. The x-axis indicates signal values and the y-axis indicates the observed frequency of each value. Different curves are distinguished with different colors. By hovered over data points a tooltip appears displaying their exact data values. The interaction also enables zooming in and out.

Figure 3.3: Three pages of SigTools Shiny App.

Chapter 4

Results

The two case studies presented in this section demonstrate SigTools effectiveness in the interpretation and evaluation of genomic signals data sets. The first case study explores several previously studied genomic signals and concludes that SigTools accurately captures their characteristics. The second case study investigates the attributes of a novel set of genomic signals and exhibits how SigTools can reveal their associate biological function.

4.1 Case Study 1 – Exploratory analysis of known histone modifications

Histones are proteins that play a crucial role in DNA structure and chromosome organization. It is possible for these proteins to be modified by connecting to a methyl group (methylation) or acetyl group (acetylation) and these modification heavily impact gene expression. Accordingly, many studies have been focused on histone modification and what particular activity they represent.

Our first dataset contains six modification signals of protein histone H3 –*H3K4me1*, *H3K4me3*, *H3K9me3*, *H3K27ac*, *H3K27me3*, and *H3K36me3*– over chromosome 21 of the human genome. Table 4.1 displays the elements and their associated locations that each of these signals represents. The signals were downloaded from Roadmap Epigenomic Data Portal [20] in indexed binary format (bigWig) with single base-pair resolution. These files were processed into a multi-column bedGraph file with 200bp resolution using `sigtools_convertToMultiColBedg` function.

The following mentioned analyses are subfigures of Fig 4.1, and in this case, they were proposed to explore and acknowledge the already established insights.

To obtain a quick grasp of the data distribution, we generated an ECDF plot using the 99 percentile of the data (Fig 4.1a) using `sigtools_distribution` function with `ecdf` option. We observe that repeated zero values constitute almost 50 percent of all the signals, which is not surprising since a great portion of human DNA is non-coding sequences with largely unknown functionality, though some contain regulatory elements [35]. Next, we

Table 4.1: The function and the location of the signals investigated in the first case study.

Assay	Acting as:	Location
H3K4me1	Primed enhancer	Spatially close to the promoter, though it might lineary be away from the gene.
H3K4me3	Active Promoter	Found near the beginning of the gene.
H3K9me3	Heterochromatin regions	Contains very few genes.
H3K27ac	Active enhancer	Commonly found near the transcription start site (TSS).
H3K27me3	Active promoter	Found near the beginning of the gene.
H3K36me3	Exons (occasionally)	Gene bodies

examine non-zero values within the 99 percentile by generating a kernel density plot (Fig 4.1b, `sigtools_distribution` function with `curve` option) which uncovers that most of the remaining population rest within the $(0, 2)$ interval. Having an estimation of variable ranges is particularly necessary when deciding whether to apply any normalization techniques on data before directing it to a learning algorithm.

Having studied value variation, the autocorrelation plot indicates how sudden or smooth the values shift in consecutive bins. Fig 4.1c displays that out of the six modifications, H3K4me3 has the sharpest picks and deepest valleys, hence it has the smallest active regions. Accordingly, the signal with the highest autocorrelation, H3K36me3, has the the largest active regions since not much value transformation is indicated.

To understand if there is a linear association between these value variations, we generated the correlation plot, which uses two visual variables (size and color) to encode Pearson correlation for all pairs of given signals. In this case, Figure 4.1d displays a high correlation between H3K27ac—indicator of active enhancers—and H3K4me1—a signal representing all enhancers.

The remaining subplots of Fig 4.1 discuss the average enriched behavior of two of the mentioned histone modifications over gene bodies of chromosome 21. H3K4me3 exhibits high values near the beginning of genes as a promoter does (Fig 4.1e) And H3K9me3 has little variability throughout the considered regions which comply with characteristics of heterochromatin regions (Fig 4.1f).

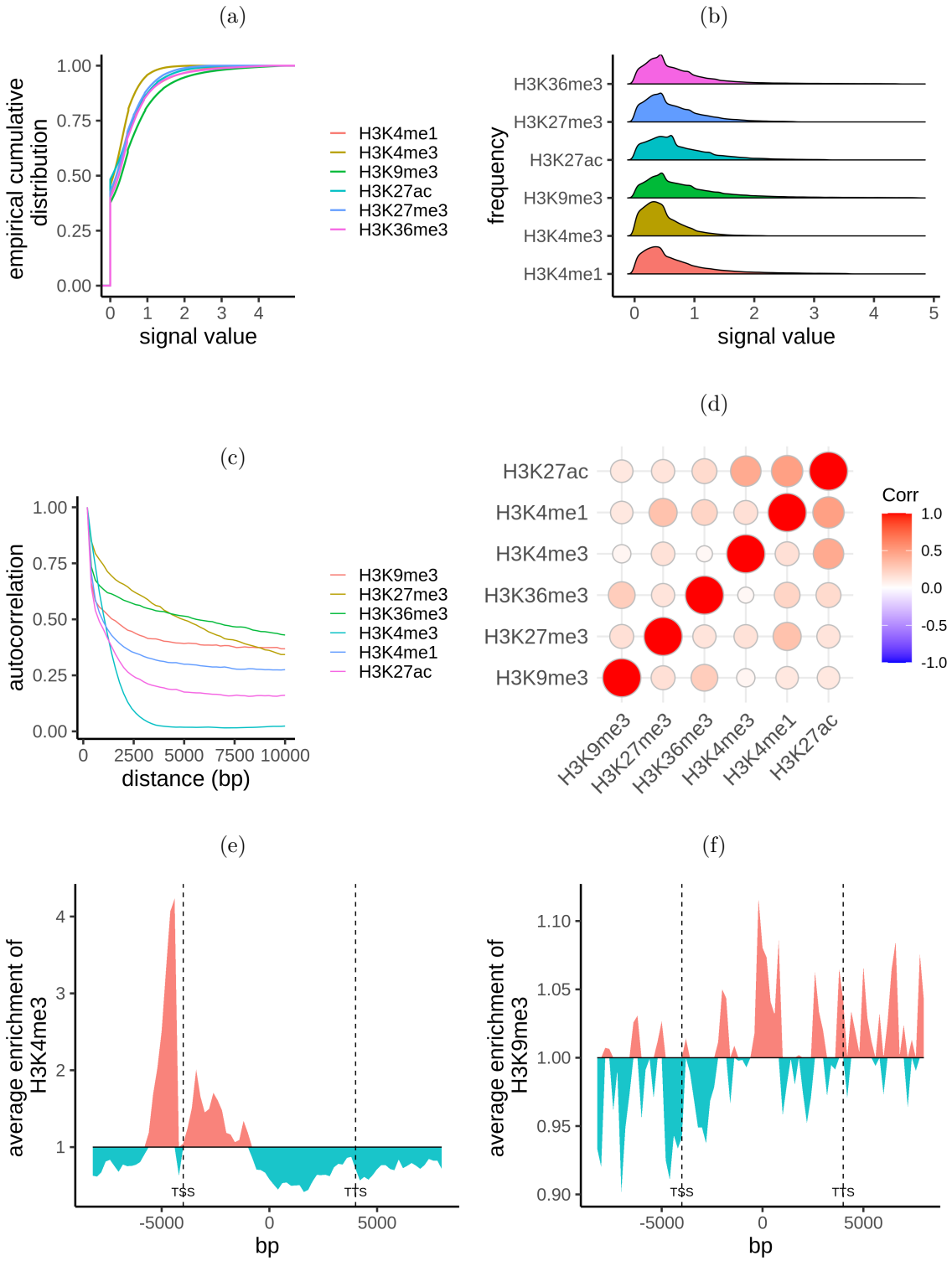


Figure 4.1: Exploratory analysis of histone modifications. a) Empirical Cumulative Distribution b) Kernel Density Distribution c) Autocorrelation d) Correlation HeatMap e) H3K4me3 Aggregation f) H3K9me3 Aggregation

4.2 Case Study 2 – Interpreting Chromatin State Feature

A recent study [4] proposes chromatin state features for capturing genomic elements instead of discrete annotation. These features are continuous genomic signals which are obtained from histone modifications refined by a Kalman filter state-space model. We chose a set of three features for this case study to demonstrate how SigTools can assist in interpretation of novel genomic signals.

Since these features are to project characteristics of histone modifications, it is expected when the ECDF plot (Fig 4.2a) of the 99 percentile of feature data displays a large population of zeros in the dataset. We can also obtain an estimation about the range of the data which is about $[0, 0.5)$ for all the signals.

Removing the zero population and the distribution curve plot (Fig 4.2b) displays that out of the three features, feature1 is denser within the range of $(0, 1)$. Despite this smaller variation, the autocorrelation plot (Fig 4.2c) displays that feature1 contains more sudden changes.

Fig 4.2d displays that feature1 mainly correlates with H3K27ac, H3K4me1 and H3K4me3 which are responsible for transcription enhancement. Accordingly, these three assays drop the most in autocorrelation. Plotting the average enrichment of this feature over gene body regions (Fig 4.2e) make an even stronger argument that feature1 also represents enhancer activity.

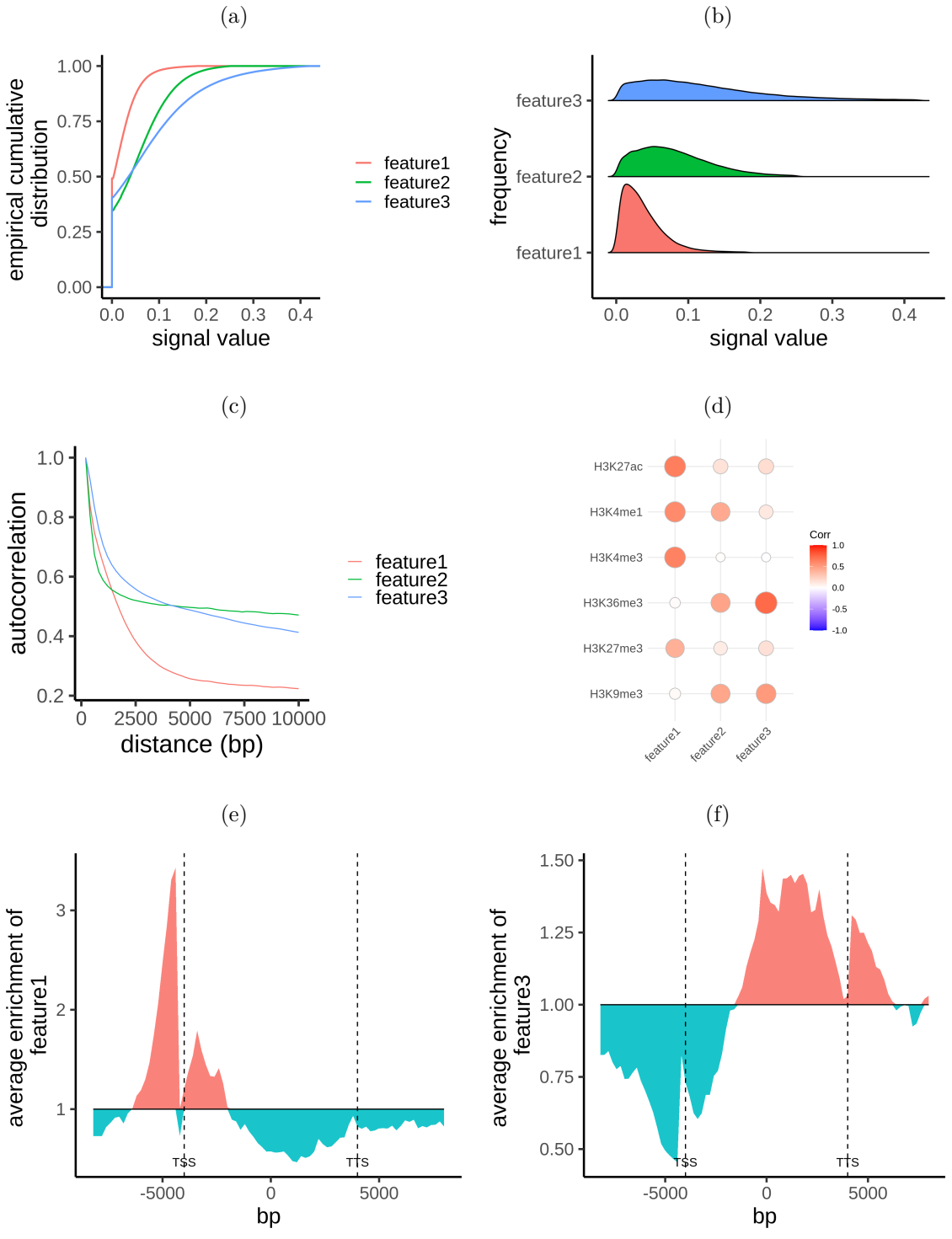


Figure 4.2: Towards the interpretation of Chromatin State Feature. a) Empirical Cumulative Distribution b) Kernel Density Distribution c) Autocorrelation d) Correlation HeatMap e) Feature1 Aggregation f) feature3 Aggregation

Chapter 5

Discussion and Conclusion

> Exploratory data analysis can never be the whole story, but nothing else can serve as the foundation stone – as the first step. (Tukey [37] p. 3)

An ever-increasing number of genomic signals are being generated by next-generation sequencing technologies and are being widely utilized in studies such as genome annotation [10, 16], cell development, and gene regulation. The analysis of these signals is often associated with the analysis of short-read sequences or genomic elements. However, in recent annotation studies [4] theoretically generated signals have been promoted to be a direct representative of biological activities. Regarding the importance and the increasing number of genomic signals and their novel applications we believe there is a growing need for refined tools that enable convenient exploratory analysis and facilitate genomic signals' interpretation and assessment.

The primary contributions of this work are briefly listed below:

- The design, development, and implementation of an R-based data analysis package, SigTools, to be used by biologists or computer scientists who work with known and novel genomic signals. This package includes several **recognized** statistical plots that are frequently employed for exploratory data analysis in genomics and other fields. Table 5.1 is an overview of SigTools visualization modules and their availability in other genomic signal analysis tools. This table indicates that no other tool offers a function for generating distribution and correlation plots for this type of data, the correlation plot is offered only by one other tool, and the aggregation plot is offered by all of them.
- An aggregation plot is a powerful visualization that has been frequently employed in genomic signals analysis. Table 5.1 displays that the aggregation plot is incorporated in all locus-agnostic visualization tools discussed in Section 2.2.2. In this thesis, we *implement* a novel visual encoding for this plot. By introducing a *shifted origin* line to this plot, we aimed to highlight the difference between high and low signal values and enable comparison between different aggregation plots for one signal over different sets of elements, or multiple signals across the same elements.

- Offering a web-based application, SigTools-Shiny. This graphical user interface would be an additional option for users who prefer to limit their interaction with a command-line environment and feel more comfortable inspecting their data with different combinations of plots and parameters through a GUI. SigTools-Shiny also includes some interactive versions of SigTools visualizations, these interactive Java-Script plots are generated by an R package named *Plotly*.

SigTools enables users who work with both experimental or statistical generated genomic signals to obtain text-based or graphical statistical summaries of their datasets, to understand what activities their novel signals represent, and investigate the relation of their recently obtained signals with previously studied signals.

As for working with any other extensive dataset, the large size of genomic signals is the challenge that requires close consideration. For obtaining faster results, SigTools offers two solutions: working with modified data to a bigger resolution size, or working with a random subset of the data. The file size can particularly cause issues in web applications when users have to pause their analysis due to multiple uploads when working with diverse datasets. To overcome this issue, some frameworks such as GALAXY [11] offer cloud workstations to their users, hence uploaded data is stored in the user's account and it can be accessed at any time. As a part of GALAXY, deepTools users benefit from such an online work station. Being a stand-alone tool allowed SigTools to have a flexible user interface design, yet finding a solution for reducing the number of uploads should be included in SigTools future versions. Future versions of SigTools should also focus on including an additional number of visualization and enable comparison between continuous and discrete genomic data.

Table 5.1: Sigtools' features and their availability in other tools.

Tools	distibution	correlation	autocorrelation	aggregation (summary)	web-app
SigTools	✓	✓	✓	✓	✓
deepTools		✓		✓	✓
genomation				✓	
ngs.plot				✓	
SegTools				✓	

Bibliography

[1] Altuna Akalin, Vedran Franke, Kristian Vlahoviček, Christopher E. Mason, and Dirk Schübeler. genomation: a toolkit to summarize, annotate and visualize genomic intervals. *Bioinformatics*, 31(7):1127–1129, 11 2014.

[2] Kundaje A. Boyle A.P. Amemiya, H.M. Identification of Problematic Regions of the Genome. *Sci Rep*, 9(9354), 2019.

[3] Baubec, T., Colombo, D., Wirbelauer, C. *et al.* Genomic profiling of DNA methyltransferases reveals a role for DNMT3B in genic methylation. *Nature*, 520(243-247), 2015.

[4] Bowen Chen, Neda Shokraneh Kenari, Habib Daneshpajouh, Kay C. Wiese and Maxwell W. Libbrecht. Continuous chromatin state feature annotation of the human epigenome. *International Conference on Machine Learning ICML 2019-Workshop on Computational Biology*, 2019.

[5] Buels, R., Yao, E., Diesh, C.M. *et al.* JBrowse: a dynamic web platform for genome visualization and analysis. *Genome Biol*, 17(66), 2016.

[6] Buske, O.J., Hoffman, M.M., Ponts, N. *et al.* Exploratory analysis of genomic segmentations with segtools. *BMC Bioinformatics*, 12(415), 2011.

[7] Celton J. Linsmith G. *et al* Daccord, N. High-quality de novo assembly of the apple genome and methylome dynamics of early fruit development. *Nat Genet*, 49(1099–1106), 2017.

[8] Deepak Purushotham Renee L Sears Ting Wang Daofeng Li, Silas Hsu. Washu epigenome browser update 2019. *Nucleic Acids Research*, 47(W158–W165), 2019.

[9] Yangyang Deng, Xianhua Dai, Qian Xiang, Zhiming Dai, Caisheng He, Jiang Wang, and Jihua Feng. Genome-wide analysis of the effect of histone modifications on the coexpression of neighboring genes in saccharomyces cerevisiae. *BMC Genomics*, 11(1):550, Oct 2010.

[10] ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489:57, 2012.

[11] Enis Afgan, Dannon Baker, Bérénice Batut, *et al.* The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update,. *Nucleic Acids Research*, 46(W537–W544), 2018.

[12] Gehlenborg, N., O'Donoghue, S., Baliga, N. *et al.* Visualization of omics data for systems biology. *Nat Methods*, 7(S56–S68), 2010.

[13] David Gosselin, Dylan Skola, Nicole G. Coufal, Inge R. Holtman, Johannes C. M. Schlachetzki, Eniko Sajti, Baptiste N. Jaeger, Carolyn O'Connor, Conor Fitzpatrick, Martina P. Pasillas, Monique Pena, Amy Adair, David D. Gonda, Michael L. Levy, Richard M. Ransohoff, Fred H. Gage, and Christopher K. Glass. An environment-dependent transcriptional network specifies human microglia identity. *Science*, 356(6344), 2017.

[14] Sheraz Gul. Epigenetic assays for chemical biology and drug discovery. *Clinical Epigenetics*, 9(1):41, Apr 2017.

[15] Michael M. Hoffman, Orion J. Buske, and William Stafford Noble. The Genomedata format for storing large-scale functional genomics data. *Bioinformatics*, 26(11):1458–1459, 04 2010.

[16] Michael M. Hoffman, Jason Ernst, Steven P. Wilder, Anshul Kundaje, Robert S. Harris, Max Libbrecht, Belinda Giardine, Paul M. Ellenbogen, Jeffrey A. Bilmes, Ewan Birney, Ross C. Hardison, Ian Dunham, Manolis Kellis, and William Stafford Noble. Integrative annotation of chromatin elements from ENCODE data. *Nucleic Acids Research*, 41(2):827–841, 12 2012.

[17] Birney E Cameron G Chen Y Clark L *et al.* Hubbard T, Barker D. The Ensembl genome database project. *Genome Biol*, 30(38-41), 2002.

[18] Ernst Jason and Kellis Manolis. Chromatin-state discovery and genome annotation with ChromHMM. *Nature Protocols*, 12:2478, 2017.

[19] Aziz Khan and Anthony Mathelier. Intervene: a tool for intersection and visualization of multiple gene or genomic region sets. *BMC Bioinformatics*, 18(1):287, May 2017.

[20] Meuleman W. Ernst *et al* Kundaje, A. Integrative analysis of 111 reference human epigenomes. *Nature*, 518(317–330), 2015.

[21] B. Laura. Epigenomics: The new tool in studying complex diseases. *Nature Education*, 1((1):178), 2008.

[22] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, and 1000 Genome Project Data Processing Subgroup. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 06 2009.

[23] Arpit Mishra and R. David Hawkins. Three-dimensional genome architecture and emerging technologies: looping in disease. *Genome Medicine*, 9(1):87, Sep 2017.

[24] Tamara Munzner. *Visualization Analysis and Design.* A K Peters visualization series. A K Peters/CRC Press, Florida, 1 edition, 2015.

[25] Fábio C. P. Navarro, Hussein Mohsen, Chengfei Yan, Shantao Li, Mengting Gu, William Meyerson, and Mark Gerstein. Genomics and data science: an application within an umbrella. *Genome Biology*, 20:109, 2019.

[26] Ann E. Loraine Nowlan H. Freese, David C. Norris. Integrated genome browser: visual analytics platform for genomics. *Bioinformatics*, 32(2089–2095), 2016.

[27] S Nusrat, T Harbig, and N Gehlenborg. Tasks, Techniques, and Tools for Genomic Data Visualization. *Computer Graphics Forum*, 28:781–805, 2019.

[28] Nguyen QV Zhou Y Catchpoole DR. Qu Z, Lau CW. Visual Analytics of Genomic and Cancer Data: A Systematic Review. *Cancer Informatics*, 2019.

[29] Aaron R. Quinlan and Ira M. Hall. Bedtools: a flexible suite of utilities for comparing genomic features. *Bioinformatics (Oxford, England)*, 26(6):841–842, Mar 2010. 20110278[pmid].

[30] Ramírez-González, R. H. and Borrill, *et al.* The transcriptional landscape of polyploid wheat.

[31] Ramírez, Fidel, Devon P. Ryan, Björn Grüning, *et al.* deeptools2: a next generation web server for deep-sequencing data analysis. *Nucleic Acids Research*, 2016.

[32] Silva T.C. Coetzee S.G. *et al* Reyes, A.L.P. GENAVi: a shiny web application for gene expression normalization, analysis and visualization. *BMC Genomics*, 20(745), 2019.

[33] RStudio, Inc. *Easy web applications in R.*, 2013. URL: http://www.rstudio.com/shiny/.

[34] Benjamin R. Sabari, Alessandra Dall'Agnese, Ann Boija, Isaac A. Klein, Eliot L. Coffey, Krishna Shrinivas, Brian J. Abraham, Nancy M. Hannett, Alicia V. Zamudio, John C. Manteiga, Charles H. Li, Yang E. Guo, Daniel S. Day, Jurian Schuijers, Eliza Vasile, Sohail Malik, Denes Hnisz, Tong Ihn Lee, Ibrahim I. Cisse, Robert G. Roeder, Phillip A. Sharp, Arup K. Chakraborty, and Richard A. Young. Coactivator condensation at super-enhancers links phase separation and gene control. *Science*, 361(6400), 2018.

[35] Pramanayagam S ShanmugamA, Nagarajan A. Non-coding DNA – a brief review. *J App Biol Biotech*, 5(05)(42-47), 2017.

[36] Shen, L., Shao, N., Liu, X. *et al.* ngs.plot: Quick mining and visualization of next-generation sequencing data by integrating genomic databases. *BMC Genomics*, 15(284), 2014.

[37] John W. (John Wilder) Tukey. *Exploratory data analysis / John W. Tukey.* Addison-Wesley series in behavioral science. Addison-Wesley Pub. Co., 1977.

[38] Cath Tyner, Galt P. Barber, Jonathan Casper, Hiram Clawson, Mark Diekhans, Christopher Eisenhart, Clayton M. Fischer, David Gibson, Jairo Navarro Gonzalez, Luvina Guruvadoo, Maximilian Haeussler, Steve Heitner, Angie S. Hinrichs, Donna Karolchik, Brian T. Lee, Christopher M. Lee, Parisa Nejad, Brian J. Raney, Kate R. Rosenbloom, Matthew L. Speir, Chris Villarreal, John Vivian, Ann S. Zweig, David Haussler, Robert M. Kuhn, and W. James Kent. The UCSC Genome Browser database: 2017 update. *Nucleic Acids Research*, 45(D1):D626–D634, 11 2016.

[39] Kanno Y. Furumoto Y. *et al* Vahedi, G. Super-enhancers delineate disease-associated regulatory nodes in T cells. *Nature*, 520(558–562), 2015.

[40] Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, and Haussler D. The human genome browser at UCSC. *Genome research*, 12:996, 2002.

[41] Wen Yao Yiming Yu, Yidan Ouyang. shinyCircos: an R/Shiny application for interactive creation of Circos plot. *Bioinformatics*, 34(7), 2018.

[42] Yiming Yu, Wen Yao, Yuping Wang, and Fangfang Huang. shinyChromosome: An R/Shiny Application for Interactive Creation of Non-circular Plots of Whole Genomes. *Genomics, Proteomics & Bioinformatics*, 17(5), 2019.