Statistical Methods For Differential Analysis Of Hi-C And ChIP-Seq Data

by

Duy Nguyen

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Statistics)

at the

UNIVERSITY OF WISCONSIN–MADISON

2018

Date of final oral examination: 05/29/18

The dissertation is approved by the following members of the Final Oral Committee:
Eric Johannsen, Associate Professor, Medicine and Oncology
Sündüz Keleş, Professor, Statistics and of Biostatistics and Medical Informatics
Christina Kendziorski, Professor, Biostatistics and Medical Informatics
Michael Newton, Professor, Statistics and of Biostatistics and Medical Informatics
Kam-Wah Tsui, Professor, Statistics

ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude to my advisor, professor Sündüz Keleş. Without her guidance, this thesis would not have been possible. I would like to thank her for providing challenging yet interesting and exciting works. Most importantly, her patience, flexibility and openness to different research areas have significantly motivated and improved my works. I consider myself very fortunate to have a chance to work and learn from her; the work ethic and passion she has for her research was contagious and motivational, and I am infected by that.

I am thankful to professor Eric Johannsen, professor Christina Kendziorski, professor Michael Newton, and professor Kam-Wah Tsui to be my committee members. I would like to thank them for their time, questions, invaluable discussions and comments. In addition, I wish to thank professor Moo Chung (Biostatistics and Medical Informatics), and professor Jerry Zhu (Computer Sciences) for introducing me to the newly developed and exciting field of Topological Data Analysis (TDA). Through my collaborative works with them, I was able to gain significant insights of TDA, leading to results in Chapter 2.

In addition, I would like to thank my collaborators: Dr. Lam Si Tung Ho (Dalhousie), Dr. Vu Dinh (Delaware), Dr. Cuong Viet Nguyen (Cambridge), Dr. Binh Thanh Nguyen (University of Science, Vietnam), Ye Zheng (UW), and Tien Vo (UW), for their stimulating discussions and collaborations with me on various research topics in statistics as well as machine learning. I also would like to express my gratitude to professor Rick Nordheim (UW), professor Bob Doran (TCU), professor Ken Richardson (TCU), and Ken Ueda. Without their helps outside my academic work, I would not have pursued a Ph.D. in statistics. I want to thank them for their supports, and encouragement, especially during my toughest times in research and personal life.

Lastly, I would like to thank my parents and my younger sister for all their love and encouragement. And most of all for my wife whose love, patience and endless support during the final steps of this Ph.D. is truly appreciated.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

High-throughput sequencing has become a standard method in genomics research and made it possible to address important biological questions. In this thesis, we focus on developing statistical methodologies and software to analyze two important high-throughput sequencing technologies: chromatin conformation capture with high-throughput sequencing (Hi-C), and chromatin immunoprecipitation coupled with high-throughput next generation sequencing (ChIP-seq). Hi-C data provides key insights into the 3D structures of the human genome, while ChIP-seq has been successfully used for genome-wide profiling of transcription factor binding sites, histone modifications, and nucleosome occupancy in many organisms and humans.

This thesis contains three major parts. In the first part, we discuss challenges in quantitative comparison of Hi-C data (often referred as "differential (interaction) analysis" problem) across different cellular conditions. Prior to this work, the state of the art methods for detecting differential interactions largely depends on methods borrowed from RNA-seq data analysis. Such comparisons have critical shortcomings involving testing a large collection of hypotheses in large-scale Hi-C studies. As a result, these existing strategies for detecting differential interactions fail to control the rate of false discovery (FDR) for reported findings in many simulations and experimental Hi-C studies, hindering their comparative analysis. To address these issues, we present TreeHiC, the first hierarchical multiple testing procedure for quantitative comparison applied to Hi-C data. We demonstrate that this framework can detect differential interactions while assuring control of the FDR in complex large-scale Hi-C studies under a wide range of settings. It also is considerably more powerful than existing methods, especially in sparse testing problems where number of hypotheses could be millions with a weak signal-to-noise ratio. Additionally, while the current version of TreeHiC implements methodology pertaining to Hi-C differential analysis, it is easily extendable for other similar data.

For the second part, we investigate statistical challenges in quantitative comparison of histone profiles across different cellular conditions from ChIP-seq data.

Quantitative comparison of histone profiles largely depends on methods borrowed from RNA-seq data analysis. As a result, such comparisons are restricted to the evaluation of differential signal intensity and have critical shortcomings pertaining histone modification marks with diffuse signals and multiple local peaks. To address these problems, we develop TAN, a nonparametric method motivated by the adaptive Neyman test for quantitative comparison of ChIP-seq data. We demonstrate that this framework can detect differential histone mark enrichment under a wide range of settings. Compared to existing methods, TAN shows a better performance in detecting subtle differences in coverage levels between samples, yet is capable of detecting higher order changes such as shape across pre-defined regions. Additionally, TAN is universally applicable to any type of differential ChIP-seq data analysis and is easily extendable for multiple condition comparison.

In the last part, we describe our two novel software in the R packages *TreeHic* and *tan*. Through applications to real Hi-C and ChIP-seq data, we present how these software could reveal biological insights that are not captures in standard data analysis.

1 INTRODUCTION

1.1 Background

High-throughput sequencing has become a standard method in genomics research and made it possible to address important biological questions. Among these high-throughput sequencing technologies, there are two important techniques: (1) chromatin immunoprecipitation coupled with high-throughput next generation sequencing (ChIP-seq), and chromatin conformation capture with high-throughput sequencing (Hi-C). ChIP-seq generates high resolution genome-wide profiles of histone modifications (HMs) (Wang et al., 2008) and transcription factor (TF)-DNA interactions (Kharchenko et al., 2008; Robertson et al., 2007). On the other hand, Hi-C is one of the most widely adopted high-throughput technologies for studying the 3-dimensional (3D) structure of the human genome and assessing the spatial proximity of potentially any pair of genomic regions (Lieberman-Aiden et al., 2009; Rao et al., 2014).

ChIP-seq applications include identifying and comparing genomewide profiles of TFs and HMs across different samples and cellular conditions. Both TF binding and histone modifications play important roles in condition-specific gene regulation. For Hi-C technique, soon after the increasing availability of Hi-C datasets (Rao et al., 2014), there is much interest in comparing interactions that are significantly different across different samples and cellular conditions. Discovery of such interactions aims at to explain the roles of chromatin structure in mechanisms of gene regulation or epigenetic modification differences between cell types, and experimental conditions. Therefore, whether we work with ChIP-seq or Hi-C data, quantitative comparison (often referred as "differential analysis" problem) is necessary to understand the dynamics of these processes. Here, our main interest lies in the development of two separate new methods for differential analysis in the context of ChIP-seq and Hi-C data. In this work, I introduce two newly developed statistical tools to address these challenges.

1.2 Outline of the thesis

In this thesis, we discuss novel statistical methods (Chapters 2 and 3) and their software (Chapter 4). Throughout this thesis, each chapter is an independent work which is self-contained with relevant background for each statistical problem.

Chapter 2 proposes a statistical framework for rigorous detection differential interactions. Specifically, we approach this problem as a hierarchical testing framework and propose a novel algorithm, TreeHiC. Our computational experiments and simulations illustrate that TreeHiC is powered for detecting changes while robustly controlling the false discovery rate (FDR) under a wide range of settings and resolutions. It also is considerably more powerful than existing methods, especially in sparse testing problems where number of hypotheses could be millions with a weak signal-to-noise ratio, giving it a clear advantage over existing differential interaction approaches. We further validate our results by experimental validation.

Chapter 3, which was submitted and is currently under revision in the Bioinformatics, considers a statistical framework for differential enrichment analysis from ChIP-seq data. Specifically, we develop a novel statistical method, TAN, for identifying genomic loci with differential histone modifications across biological conditions with potentially varying peak shapes and structures of ChIP-seq signal. ChIP-seq data is widely used for profiling histone modifications across different conditions to elucidate the chromatin contribution to regulatory dynamics across conditions. Although there are several approaches for identifying differential histone enrichment between experimental conditions, the field largely relies on DEseq (Love et al., 2014) applied to ChIP-seq data. DEseq (and other variations of DEseq with generalized linear models machinery) is exceptionally good at identifying total read count based differences; however, it does not have the functionality to capture enrichment differences due to shape in histone modified regions, e.g., due to nucleosome movements, or other more refined genomic structures. In this work, we develop the first statistical approach that is equally powered to detect both the affinity (total read count) and shape-based differential enrichment. We utilize this approach to analyze H3K27me3 profiles in a conditional activation study of EBV in-

fected B-lymphocytes. Our computational analysis and experimental validation of the results from the H3K27me3 study establish our approach as a widely applicable tool for differential enrichment analysis.

In Chapter 4, we describe two of our software that implement the frameworks proposed in Chapters 2 and 3 in the R packages *TreeHiC* and *tan*, respectively. In addition, our software provide user-friendly interface and computationally efficient implementation. These software are open-sourced, and can be downloaded from `https://github.com/duydnguyen/TreeHiC`, and `https://github.com/duydnguyen/tan`.

2 TREEHIC: HIERARCHICAL TESTING FOR DIFFERENTIAL CHROMATIN INTERACTION ANALYSIS

2.1 Background

Recent years have seen rapid progress in technologies for studying the 3-dimensional (3D) structure in eukaryotes. In addition, 3D genome organization in a growing number of cell types has been studied in increasingly greater detail and scales. Here we provide an overview of the chromosome conformation capture (3C)-coupled sequencing methods referred as C-technologies. In C-technologies, the nucleus is chemically fixed to preserve the 3D chromosome conformation, followed by DNA fragmentation and religation. If the two DNA fragments are spatially close at the time of cross-linking, they can be ligated (Dekker, 2002). To assess the spatial proximity between pairs of loci, the frequency of the ligation products can be assayed by DNA sequencing (Lieberman-Aiden et al., 2009). Among such techniques, 3C (one-to-one) is the earliest version of C-technologies to capture contact frequency between two preselected loci (Dekker, 2002), hence the name 3C (one-to-one). Despite its low output, 3C is widely used to verify long-range interactions due to its high resolution. To extent the 3C technology, there are two different methods in C-technologies to aim at capturing a wider range of contact frequency between loci, namely 4C (one-to-all) and 5C (many to many). 4C allows for genome-wide identification of all possible interacting partners for one specific locus of interest. On the other hand, 5C measures contact frequencies among DNA fragments within a finite number of target loci. Most recently, the demand for an unbiased measurement of all possible interactions across the genome calls for Hi-C (all-to-all) technique. Hi-C combines 3C and next generation DNA sequencing (Lieberman-Aiden et al., 2009). To further improve the resolution and reduce background noise, different variants of Hi-C have been introduced such as DNase Hi-C, micro-C, and in situ Hi-C (Rao et al., 2014; Hsieh et al., 2015). In this work, we are interested in the Hi-C technology due to its power and popularity for studying chromatin folding.

Hi-C is one of the most widely adopted high-throughput technologies for studying the 3-dimensional (3D) structure of the human genome and assessing the spatial proximity of potentially any pair of genomic regions (Lieberman-Aiden et al., 2009; Rao et al., 2014). Computationally, Hi-C data used to obtain 3D structures exhibit many sources of biases due to various experimental conditions. It was clear that these biases substantially affect chromatin interactions. Thus, most methods have been developed for processing data from Hi-C experiments, such as filtering and normalization, to remove biases in Hi-C datasets (Imakaev et al., 2012; Servant et al., 2015; Forcato et al., 2017). Soon after the increasing availability of Hi-C datasets (Rao et al., 2014), there is much interest in comparing interactions that are significantly different across different samples and cellular conditions. Discovery of such interactions aims at to explain the roles of chromatin structure in mechanisms of gene regulation or epigenetic modification differences between cell types, and experimental conditions. For instance in Dixon et al. (2015), it was shown that depletion and enrichment in Hi-C interactions is linked with genes down or up-regulation, respectively. Therefore, quantitative comparison of Hi-C (often referred as "differential (interaction) analysis" problem) is necessary to understand the dynamics of these processes.

Differential interaction (DI) analysis for the two-condition comparison has recently gained considerable interest. However, very few tools perform comparative analysis, visually or statistically, of two Hi-C contact maps (Heinz et al., 2010; Paulsen et al., 2014; Lun and Smyth, 2015; Stansfield and Dozmorov, 2017; Djekidel et al., 2018). There are a number of features that make differential Hi-C analysis different than other analyses such as RNA-seq, which is a comparatively well-studied problem (Pepke et al., 2009). First, in most Hi-C studies, only a very small number of biological replicates (e.g., 1 to 2) are performed owing it to the fact that the primary objective of these analyses is detection of "significant" interactions (peak calling). This makes statistical inference for comparing across multiple conditions a challenging task. Second, many of the existing methods do not scale to high-resolution Hi-C data such as human and mouse. For instance, high-resolution Hi-C studies are involved in testing a large collection of hypotheses. As a result, the

existing strategies for detecting differential interactions fail to control the rate of false discovery (FDR) for reported findings in many simulations and experimental Hi-C studies, hindering their comparative analysis. In addition, the increase in data resolution is requiring new tools to efficiently handle large Hi-C contact maps, while at the same time allowing users to extract meaningful findings at multiple scales.

Two strategies have emerged to address these unique challenges. The first and most straightforward method is the "overlap analysis", which is to apply a peak-calling algorithm (e.g., Rao et al. (2014); Ay et al. (2014a); Xu et al. (2015)) to identify loci with significant changes in the interaction intensity, i.e., peaks, for each of the two conditions. The peak annotation in one condition but without peaks in the other are then deemed as differentially interacted (Rao et al., 2014). However, such a comparison is highly dependent on the thresholds/error rates used for peak calls. Regions (peaks) barely over the threshold in one cell type but under threshold in the other will be declared as condition-specific peaks even if the quantitative differences are small. In addition, this approach completely ignores the quantitative comparison of the genomic regions that are identified as peaks in both cell types (i.e., common peaks) even when the quantitative differences are large.

An alternative strategy to the overlap analysis is to compare read counts (intensity of interactions) of all pairs of loci in the genome, and quantify for differences in the magnitude of the read counts across conditions. Surveying the literature, we noticed an increasing adoption of these count-based methods to detect DIs. Several parametric methods based on Poisson, Negative bionomial and bionomial distributions are in this category (Heinz et al., 2010; Paulsen et al., 2014; Lun and Smyth, 2015; Djekidel et al., 2018). By comparing diffHiC (Lun and Smyth, 2015) to binomial-based methods, Lun and Smyth (2015) showed that diffHiC provides improved sensitivity and error rate control for DI detection, compared to its bionomial counterparts (Heinz et al., 2010; Paulsen et al., 2014). These count-based approaches mostly adapt the methods for RNA-seq differential expression analysis to the more structured Hi-C data. A notable drawback of these approaches is that they were developed to analyze relatively low-resolution Hi-C contact maps (e.g., 50 kb or

more). At low resolutions, regions contain more reads and provide larger counts, increasing precision and power for hypothesis testing (Lun and Smyth, 2015). However, in the case of high-resolution Hi-C protocols (Rao et al., 2014), read pairs are sparsely distributed across the interaction space. In this case, the coverage of interactions between pairs of loci is largely reduced, making the statistical assessment of differences a challenging task. As a result, the resolution is a critical parameter that determines the performance of each tools.

Most recently, as a third alternative, FIND (Djekidel et al., 2018) proposed an approach for differential interaction detection in the case of high-resolution Hi-C contact maps. By using a spatial Poisson process, FIND effectively considers the local spatial dependency between interacting loci. When varying fold-change values in differential interactions, FIND outperforms diffHiC in the case of small fold-change values. However, FIND's performance degrades and shows generally inferior performance to diffHiC in high fold-change differential regions. In practice, Djekidel et al. (2018) suggests to combine FIND with diffHiC, especially for detecting loci that exhibit high fold-change values. This is a major drawback since difference in signal strength between Hi-C samples is probably the most prominent feature for detecting differentially interacted regions. Further, in high-resolution Hi-C studies, differential testing has critical shortcomings involving testing a large collection of hypotheses. As a result, FDR control and sensitivity for reported findings should be studied. To our surprise, these issues have been addressed by diffHiC but not FIND. Note that we have emphasized the importance of FDR control in high-resolution analysis, as it is critical due to the large number of hypotheses under consideration. To resolve these issues, it is essential to develop a new computational method that are both theoretically sound and practically scalable in context of Hi-C differential analysis.

Here, we present a novel approach, TreeHiC, for rigorous detection differential interactions. TreeHiC replies on hierarchical multiple testing framework introduced by Yekutieli (2008) and is the first of its kind applied to Hi-C data. Our computational experiments and simulations illustrate that TreeHiC is powered for detecting changes while robustly controlling the FDR under a wide range of settings and

resolutions. It also is considerably more powerful than existing methods, especially in sparse testing problems where number of hypotheses could be millions with a weak signal-to-noise ratio, giving it a clear advantage over existing differential interaction approaches. Additionally, while the current version of TreeHiC implements methodology pertaining to Hi-C differential analysis, it is easily extendable for other similar data such as ChIA-PET and HiChIP. For developing TreeHiC and studying its operating characteristics, we utilized samples from: (i) three different asexual stages of the malaria parasite *Plasmodium falciparum* 3D7 (Ay et al., 2014b), and (ii) Hi-C contact maps of K562 and GM12878 cells (Rao et al., 2014).

2.2 The TreeHiC modeling framework

Differential analysis is usually performed by a two-step procedure. The first step is to process and normalize Hi-C contact maps over which the differential interaction detection is to be assessed. Since the first step is well-studied in the literature, we will focus on our method for quantitative comparison between two conditions. Note that for our real data analysis, the data processing and normalization steps are discussed in details in Appendix A.3.

For a two-sample comparison, we have contact matrices $F_i(x, y)$ and $G_i(x, y)$, $1 \leqslant i \leqslant n_1, n_2$, and $1 \leqslant x, y \leqslant N$. Here F and G denote the two experimental conditions, i represents the index for samples within each condition, and x, y are indices for genomic coordinates. Note that under this notation, collection of $F_i(x, y)$, $1 \leqslant x, y \leqslant N$, where N denotes the length of the genome, represents Hi-C contact matrices as 2D maps. However, an alternate method for visualizing Hi-C maps, and peaks in particular, is to represent them in a 3D parameter space. The x-axis and y-axis represent the location of any two loci, while the height (in the z-axis direction) gives the number of contacts. By considering a single condition, this alternate data visualization scheme proves to be useful for peak-calling algorithms where candidates for significant interactions appear as sharp apexes or mountain-like structures in the corresponding 3D map (Rao et al., 2014). In the case of differential analysis, candidates for differential interactions could be readily extended by considering

the log fold-change function:

$$h(x,y) = \log_2\left(\frac{\bar{G}(x,y)}{\bar{F}(x,y)}\right),$$

$$\text{where } \bar{F}(x,y) = n_1^{-1}\sum_{i=1}^{n_1} F_i(x,y), \quad \bar{G}(x,y) = n_2^{-1}\sum_{i=1}^{n_2} G_i(x,y),$$

i.e., the two mean contact maps for each conditions. Note that for the definition of h to be well-defined, we usually add a small positive quantity $\epsilon > 0$ (e.g., 0.001) to $\bar{F}(x,y)$. Note that we emphasize that the choice of ϵ does not affect our results and the analysis. Though the choice of different ϵ's do change the value of $h(x,y)$, it does not change the coordinates (x,y) where h achieves (local) maxima or minima. We could see this by taking the first derivatives of h and h^*, where h^* is a transformation of h under a different ϵ. By setting the first derivatives to zeros, these two normal equations have the same roots. Therefore, the sets of loci where extrema of h or h^* are achieved are the same. As a result, this transformation does not change the result of our analysis. This is because in our partition and extremum search, we only require the set of (x,y)-loci where extrema of h occur.

When contact data is examined in this manner, differential interactions appear as mountain-like or valley-like structures, which are our features of interest. As a result, searching for these features corresponds directly to finding the local extremas (both maxima and minima) of the log fold-change function $h(x,y)$. In the early stage of Hi-C differential analysis, using a simple fold change as a norm for detection of DIs was generally adopted in Wang et al. (2013); Dixon et al. (2015). We find this procedure inadequate since candidate differential interactions might not always correspond to regions of large fold-change chromatin interactions. Furthermore, by adapting this simple fold-change strategy, differential interactions with small fold-change values would not be considered. On the other hand, as motivated by peak-calling algorithm in Rao et al. (2014), we find that examining extremum output values of the fold-change function provides an appropriate baseline level for differential candidate regions.

Hi-C method of local extremum search

The techniques presented here are based on Morse theory (Milnor, 1963; Edelsbrunner et al., 2001; Gerber et al., 2010, 2012). Here, this section briefly introduces the main concepts. We refer the reader to Edelsbrunner et al. (2001); Gerber et al. (2010, 2012) for a more formal discussion, and Appendix A.1 for a further detailed illustration of the theory. We are aiming to: (i) understand the local maxima and minima (how many there are and their location); and (ii) propose a measure of the significance of each extremal point to appreciate their distributions, and tune our search algorithm. We describe the procedure for local extremum search starting with a function $h : \mathcal{M} \to [0, 1]$ where $\mathcal{M} \subset \mathbb{R}$ or $\mathbb{R}^2$. The sub-level set, noted $h^{-1}(\lambda)$, is defined as the pre-image of the open interval $(-\infty, \lambda)$:

$$h^{-1}(\lambda) = \{p \in \mathcal{M} : h(p) < \lambda\}.$$

As the threshold λ increases, the number of connected components in its sub-level set changes. In particular, it only changes when we pass through extrema of the function. We now see an example of how local extrema characterize the sub-level set of a function $h : \mathcal{M} \subset \mathbb{R} \to [0, 1]$ in Figure 2.1. The blue horizontal line indicating λ moves from 0 to 1. Before the line hits the minimum (point A), the sub-level set (red lines) is empty except for the boundaries. After the threshold λ touches A, the sub-level set becomes a segment on x-axis below A that enlarges as the threshold line keeps increasing upward. When λ reaches B, the two leftmost segments are paired up, making the number of connected components decrease by 1. Continuing with this procedure, as we reach the global maximum (point C), the line segment created at B is merged with the right boundary. As a result, we pair the global minimum value of the function (point D) with the global maximum.

As illustrated by this example, the pairing of minima and maxima introduces a measure of the strength of each extremal point, call *persistence*. Persistence is a measure of the amount of change in the function h required to remove an extremum. Therefore, it also quantifies when two or more partitions are merged. In our example, the highest persistence level γ_1 results in a single partition (α_1, α_5) whose

extrema are points C and D. In contrast, with persistence γ_2, the function has four partitions, namely $(\alpha_1, \alpha_2), (\alpha_2, \alpha_3), (\alpha_3, \alpha_4)$, and (α_4, α_5). In addition, this change in persistence level introduces two extrema points A and B. Therefore, recursively removing the extrema with minimal persistence leads to a nested series of partitions. As a result, at each persistence level, some of the partitions are merged into a single partition until the whole process consists of only a single partition.

To understand how our partition changes with increasing persistence level, we use the persistence graph which plots the number of extremal points as a function of their persistence. In Figure 2.2, we present an example of two-dimensional log fold-change function h described in the previous section, and it corresponding persistence graphs. Here, we only show an upper triangle part of the function since the domain is symmetric in Hi-C applications. Figure 2.2 shows a sequence of increasing persistence levels where each red denotes an extremal point of the log fold-change function, and green vertical lines encoding the selected persistence values. At the highest persistence level, the partitioning of function h's domain only consists of the global maximum and minumum, and the segmentation is the entire domain. As persistence levels decrease, more extrema and corresponding partitions are discovered with respect to their persistence level. Therefore, the graph indicates which extremal points are due to noise and how many can be accurately represented.

For noisy observations such as the two-dimensional log fold-change function h in Hi-C data, the partition algorithm is likely to over-segment the region, and introduces artificial extrema. Thus, an important aspect of the local extremum search is to select the correct persistence level at which the segmentation are reliable and meaningful. In Figure 2.3A, we show the persistence graph resulting in the local extremum search for K562-GM12878 on a sampled region of chromosome 22. Here, the persistence graph exhibits that extrema with low persistence are most likely due to noise while a clear plateau represents a stable number of extrema that need a large amount of change to be simplified. Here, the green vertical line separates noise from features, indicating that the segmentation with 100 extrema requires a function change of approximately 10% percent of the function range or

less to introduce an additional extremum. At this persistence level, these extremal points are not very likely artifacts from noisy observations. To further visualize the application of persistence graph, we provide two modified versions of persistence graph: persistence graph on log-log scale (Figure 2.3B), and its density (Figure 2.3C). All versions of persistence graph show the application of persistence as an important measure of noise on extremal pairs, thereby effectively separating noise from features.

The hierarchical testing tree

Figure 2.4 summarizes TreeHiC's workflow for constructing the hierarchical testing tree from Hi-C data. This workflow consists of two main phases: (1) partition and extremum search, and (ii) testing phases for reporting differential interactions. In phase (i), TreeHiC evaluates the persistence graph. Then it selects a grid of persistence levels (e.g., $\mathrm{pL}_1, \mathrm{pL}_2, \mathrm{pL}_3$). We note that first persistence level pL_1 (solid green vertical line) indicates the most stable persistence level at which the segmentation are reliable and meaningful. Consecutively, TreeHiC constructs the corresponding segmentation based of the selected persistence levels. Here, extremum positions at each segmentation are recorded for subsequent analysis. For the first step, we adapted the Topology ToolKit (Tierny et al., 2017) for persistence-driven segmentation and analysis tasks.

We next sought to examine the generation of hierarchical testing tree resulting from the last analysis. Here, we built our testing tree and conduct testing trees of hypotheses based on hierarchical false discovery rate-controlling schematic in (Yekutieli, 2008). In this hierarchical approach, the set of tested hypotheses (e.g., diamonds and circles in Figure 2.4), is arranged in a tree in L levels where the hypotheses on the first level pL_1 of the tree have no parent hypotheses. Subsequently, each hypothesis on the next level is associated with a single-parent hypothesis. In this setting, we consider those with the same parent hypothesis as a family. In accordance with the segmentation phase, parent hypotheses correspond to partitions at larger persistence level, whereas their downstream children present those with

smaller persistence values. Note that there is a one-to-one correspondence between partitions and their induced extrema, as it has been discussed in previous sessions. Therefore, testing a hypothesis (or partition) amounts to testing its extrema.

Next, we applied this strategy by incorporating our TreeHiC with the novel testing framework from Yekutieli (2008). The hierarchical test of the tree of hypotheses has two main steps: (i) Hypotheses in the same family are tested simultaneously, and (ii) testing starts with the first level pL_1 of the tree, and a family of hypotheses on subsequent levels is tested only if its parent hypothesis is rejected. Furthermore, as each family of hypotheses is tested, the false discovery rate is controlled for a given target value q using the BH procedure (Benjamini and Hochberg, 1995).

We next sought to examine the theoretical guarantees of our hierarchical testing framework based on Yekutieli (2008). Specifically, we discuss the conditions proposed Yekutieli (2008) on which a universal bound for the FDR of the hierarchical testing could be obtained. For the first assumption, the p-values are independently distributed. More specifically, if a hypothesis H is a true null hypothesis, then the corresponding p-values $P \sim U[0,1]$, where U denotes the uniform distribution. Furthermore, for the second condition, if a hypothesis H is false null, the corresponding p-value P satisfies the condition: for all $0 < \alpha_1 \leqslant \alpha_2 \leqslant 1$, $\alpha_1/\alpha_2 \leqslant Pr(P \leqslant \alpha_1 | P \leqslant \alpha_2)$. This states that the conditional marginal distribution of all the p-values is uniform, or stochastically smaller than uniform. An instance where this holds is that the CDF of p-values P's is concave (e.g., under monotone likelihood ratio condition). Future work is planned to investigate the effect of these conditions and how well they are met in our hierarchical testing settings. Under these conditions, Yekutieli (2008) provides a universal bound for the FDR of the hierarchical testing approach, namely $2 \times 1.44 \times q$. For further illustrations of this framework, we include a detailed discussion in Appendix A.4.

Figure 2.4's testing phase presents a schematic drawing of a tree of hypotheses, and the results of the hierarchical test. Green diamonds indicate null hypotheses rejected while red circles are null hypotheses not rejected in hierarchical testing procedure. Note that hypotheses on next levels are tested only if their parent nodes are rejected (green diamond), so the testing procedure stops when all child nodes

fail to reject (red circle).

As emphasized in our workflow, the construction of TreeHiC and hierarchical testing procedure are independent, where TreeHiC's generation only depends on the log fold-change function. Therefore, TreeHiC is practically scalable, as its procedure is readily equipped with different choices of models. Since the limited number of methods available prior to this work, we accompany TreeHiC with a permutation test Stansfield and Dozmorov (2017), and the popular count-based method diffHiC from Lun and Smyth (2015). Permutation test in (Stansfield and Dozmorov, 2017) detects interaction differences based on the fold-change values on a per-unit-length-distance basis. Briefly, fold change values of same distance difference $|x - y|$ are used to provide a reference distribution. Here, the distance is measured in terms of genomic coordinates. To obtain empirical p-values, they compute the probability of observing a fold-change value in the given reference distribution, which is at least at large as the one observed for a given interaction in the comparison between conditions. On the other hand, the diffHiC method is an extension of the popular RNA-seq differential expression method edgeR (Robinson et al., 2009). They use the negative binomial model to detect chromatin interaction differences, showing that diffHiC could outperform bionomial-based methods (Lun and Smyth, 2015).

2.3 Simulations

Several methods to detect differential interactions have been proposed. However, the relative accuracy and sensitivity of these methods are unclear. To evaluate and compare TreeHiC with existing methods, we must first be able to accurately perform simulation studies on Hi-C contacts, with known and controllable structural features. Performance on the simulated data was assessed based on (1) the ability to attain a good power while controlling the false discovery rate (FDR), (2) the ability to detect differential interactions in sparse settings, and (3) the performance of each tool as the number of replicates and signal-to-noise ratios decrease. These three criteria are explored in the next subsections. We benchmarked TreeHiC against the

widely adapted count-based method, diffHiC (Lun and Smyth, 2015), and newly developed method, FIND (Djekidel et al., 2018), and a permutation test (Stansfield and Dozmorov, 2017). Two versions of TreeHiC were used in the simulations: (i) TreeHiC with p-values from permutation (tree-perm), and (ii) TreeHiC with p-values from diffHiC (tree-diffHiC).

Simulation models

To the best of our knowledge, there have been two existing procedures that are able to accurately simulate Hi-C contacts, prior to this work: (1) the first model followed the general simulation outlines of Rao et al. (2014); Stansfield and Dozmorov (2017), and (2) the second model followed the simulation analysis from Djekidel et al. (2018). For the first procedure, matrices of chromatin interaction frequencies (IFs) are generated from parametric models which account for: (i) IFs at each given genomic distance followed a decay power-law distribution, (ii) the distribution of IFs at that distance, and (iii) sparsity which zero IFs occur due to lack of interactions, or insufficient sequencing depths. Specifically, all parameters in the procedure are estimated from real Hi-C data. We note that this simulation model under these settings only aim to detect differential interactions when there is no biological replicate (i.e., $n = 1$). This is an important case as in most Hi-C studies, biological replicates are typically not available. On the other hand, when there are contact frequencies from different replicates, we use the simulation analysis proposed in FIND (Djekidel et al., 2018). In this second simulation model, FIND used the K562 Hi-C heat map as a reference. To induce pairs of differential and nondifferential interactions, they used negative-bionomial distributions with different means and dispersions. Since FIND also accounts for local spatial dependency between interacting loci, they applied a Gaussian smoother to simulate the effect of correlations in the vicinity of each differential interaction loci.

We next describe how simulation parameters are estimated. As mentioned in model 1's setting, at each given genomic distance, interaction frequencies (IFs) are generated by $\mathrm{IFs} \sim \hat{\mathrm{IF}}_d + spead_d + sparsity_d$, where distance $d = |x - y|$. The first

component $\hat{IF}_d$ was estimated by fitting the power-law distribution $IF_d = C * d^{-\alpha}$ by maximum likelihood estimation, resulting α is from 1.8 to 2.2 on GM12878 cell, at resolution from 1Mb to 50kb, on chromosome 1. For the second component, $spread_d$ is estimated using a normal distribution $N(0, SD)$, where $SD \in (1.6, 3.2)$ is the standard deviation of IF_d. This parameter is set to 1.9 in the current simulations. The proportion of zeros was modeled as $P(IF = 0) = \gamma * distance$ where $\gamma = 0.001$ by default. For the second model, negative-binomial distribution is used to induce pairs of interaction (x, y). Nondifferential pairs are sampled from a negative binomial with a mean from the corresponding interactions in K562 cell. On the other hand, the differential pairs are sampled from a negative binomial with a mean equal to the fold change of their corresponding pairwise interaction in K562 contact map. Appendix A.2 further presents detailed illustrations of our simulation configurations and generation of Hi-C contacts under the two proposed models.

Next, we conducted simulations to evaluate model performance under two main settings: (i) the fold-change values to account for signal-to-noise ratios, and (ii) resolution parameter to account for sparsity in Hi-C studies. As a result, the simulation study included 6 signal configurations for each proposed models: 3 levels of signal-to-noise ratios times 2 values of resolutions. Each configuration was run 10 times. For each run, we recorded the observed proportions of false discoveries for FDR thresholding at 0.05. In addition, other operating characteristics such as sensitivity, specificity, and precision were also evaluated.

Power Performance and FDR control

We first investigated the sensitivity, specificity, precision, and the empirical FDR (eFDR) of methods compared. Figures 2.5, 2.6, 2.7, and 2.8 summarize results on different configurations for each proposed models. For each figure, we summarize results on two different measures of performance: (a) at the target FDR of 0.05, and (b) at varying levels of controlled FDR. Note that we have emphasized the important case in which there is lack of biological replicates, as it has been shown that there has not been a globally adopted conventional method performed such analysis.

While some open source implementations for specific algorithms are available, we noticed that they are designed under the availability of biological experiments. For instance, diffHiC (Lun and Smyth, 2015) and FIND (Djekidel et al., 2018) adopted Negative Bionomial and Poisson models, respectively. These adopted count-based tools require sample size n to be at least two, thereby impeding a wider adoption of differential interaction analysis in Hi-C data.

We next corroborated two methods that could be utilized: (i) permutation test (Stansfield and Dozmorov, 2017), and (ii) diffHiC (Lun and Smyth, 2015) with an option to perform DI under $n = 1$. Figures 2.5 and 2.6 recorded different results on the characteristics for model 1. We also evaluated the operator characteristic curves of eFDR and controlled FDR for each method. Overall, both versions of treeHiC show a markedly better performance compared to permutation test and diffHiC under all settings. Performance of these two methods deteriorated under the effect of higher resolution (Figure 2.6), as compared to their treeHiC counterparts. The tree-diffHiC's sensitivity is low, but higher than that of diffHiC while still controlling FDR well. This indicates the effectiveness of proposing a tree-testing structure on an available testing method.

Next, we seek to address diffHiC's performance in this simulation setting. As mentioned previously, model 1 is utilized to quantify methods in the case of no biological replications (e.g., $n = 1$). In their main model description (Lun and Smyth, 2015), it is required that $n \geqslant 2$ to estimate the model's parameters. However, in their implementation in the R package *diffHiC*, they also included an option to perform differential detection when $n = 1$, with a warning that their dispersion parameter estimation is not reliable in this case. Here, we only include diffHiC for the comparison purpose with our tree methods. We acknowledge that diffHiC model's requirement is not satisfied in this scenario. As expected, in model 1's simulation studies, diffHiC failed in both two resolution configurations, namely low resolution (dense) in Figure 2.5, and high resolution (sparse) in Figure 2.6. Furthermore, according to model 1's generation steps in Appendix A.2, the Negative Bionomial does not hold. This is in contrast with model 2 where interaction differences were generated by Negative Bionomial models. This factor of model

robustness also contributes to diffHiC's overall performance.

As an alternative approach for computing p-values, we also included permutation tests proposed in Stansfield and Dozmorov (2017). To our surprise, the implementation in (Stansfield and Dozmorov, 2017) does not adjust for multiple testing. Here, two versions of permutation tests are performed, namely permutation test in Stansfield and Dozmorov (2017), and permutation test adjusted for multiple testing by Benjamini and Hochberg (1995). However, the adjusted version virtually calls no differential interactions in our testing. Thus, we only include the un-adjusted version implemented in Stansfield and Dozmorov (2017) for comparison purpose. Similar to those observed in diffHiC model, the tree version of permutation test, tree-perm, shows a markedly better performance compared to its non-tree version. As expected, un-adjusted permutation inflated FDR control in almost all settings. It is specially worse in the case of sparsity. This shows the effectiveness of hierarchical FDR tree schematic when accompanying with various methods. On simulated data where we know the ground truth, our analysis for $n = 1$ confirms that (1) the TreeHiC framework is able to attain good power while controlling the FDR, and (2) permutation equipped with TreeHiC appears to be well-calibrated, with good power to capture differential interactions in both dense and sparse settings.

We next sought to examine the ability of TreeHiC to identify the set of differential interactions in the second model. In contrast to the first simulation study, we evaluate model performance when there are available biological replicates. Here, we further include FIND in our comparison due to the availability of biological samples. Two versions of FIND were used in the simulations, namely FIND-hardCutof and FIND-quantreg. As mentioned in Djekidel et al. (2018), FIND-quantreg uses quantile regression to select the cutoff for q-value. It therefore disregards a pre-specified FDR level at which multiple testing procedure is applied. Thus, we only include FIND-quantreg for comparing methods at the target FDR of 0.05, whereas for the analysis at varying levels of controlled FDR, we reply on FIND-hardCutof. Figures 2.7, and 2.8 summarize results on the performance for model 2. In general, both version of TreeHiC show a superior performance compared to diffHiC and

FIND under all settings. In the case of low-resolution (Figure 2.7), diffHiC, FIND-quantreg, tree-diffHiC, and tree-perm control FDR well for reported findings. In addition, tree methods outperform diffHiC's sensitivity in the case of small and medium fold-change values. We also observe that the performance of diffHiC and FIND-quantreg was improved in the second model at target FDR of 0.05. Here, we emphasize that under model 2, diffHiC's performance improves markedly. In particular, its performance is comparable with those of tree methods in the case of large fold change (e.g., $\log_2(\text{fold change}) \geqslant 4$), and low resolution in Figure 2.7. Such improvement comes from two factors. First, model 2's generation utilizes Negative Binomial model, which is in agreement with diffHiC's model assumption. Second, dispersion parameters in diffHiC can be estimated due to sample size $n = 2$. However, when moving the high resolution configuration (dense), diffHiC's performance was deteriorated. This might suggest the appropriateness of diffHiC to perform differential analysis when low-resolution is considered (e.g. 50K or more).

On the other hand, performance of non-tree methods deteriorated significantly under the effect of sparsity (Figure 2.8), as compared to their treeHiC counterparts. As expected, in contrast to tree-diffHiC's performance in simulation model 1, tree-diffHiC equipped with biological samples performed substantially better, indicating the effectiveness of biological availability applied to diffHiC. Furthermore, our analysis shows that in both models, FDR control in tree-diffHiC is conservative since its realized FDR is much lower than the nominal one. This explained a decrease in sensitivity in the dense setting. Interestingly, this does not cause much loss in power in the sparse setting. In general, our analysis for $n = 2$ confirms that (1) the TreeHiC framework is required to control FDR, (2) the new testing approach can be considerably more powerful, and (3) TreeHiC equipped with permutation and diffHiC appears to be well-calibrated, with good power to capture findings in both dense and sparse settings.

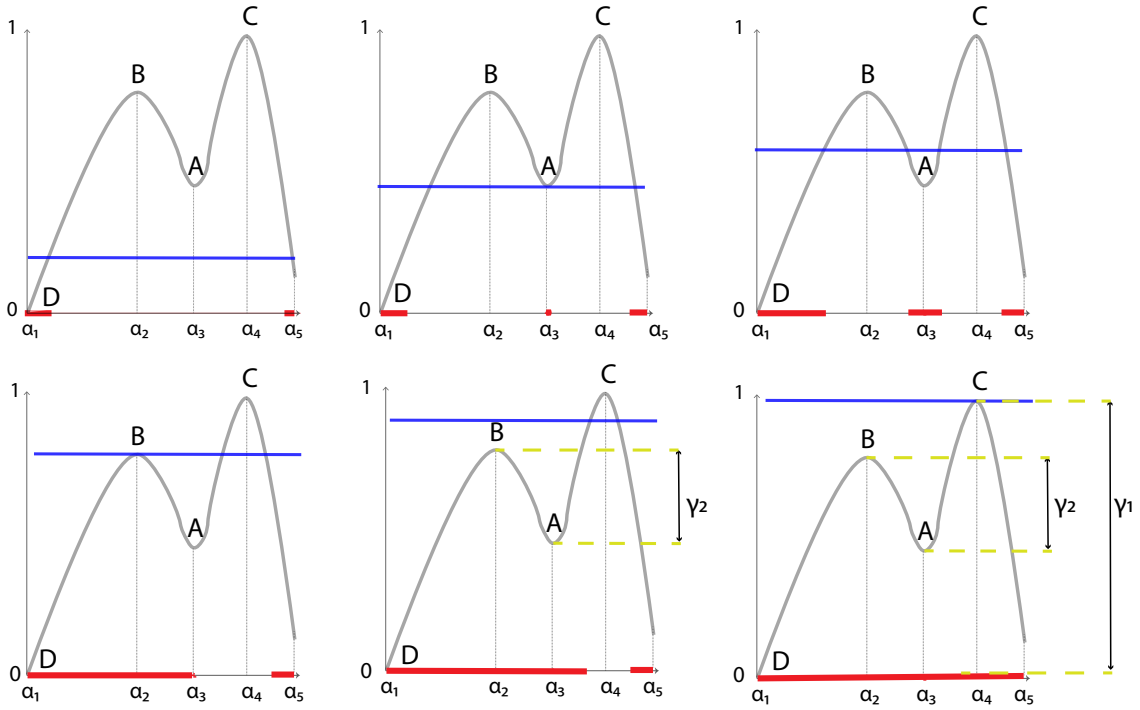


Figure 2.1: A concrete example showing an evolution of connected components in the sub-level set of a one-dimensional function, and the characterization of extrema as a function of persistence values. The blue line indicates the threshold λ moving from the minimum value of the function and up. Red lines present connected components in the sub-level set. x-axis represents the domain of the one-dimensional function. y-axis represents the range of the one-dimensional function, which is from 0 to 1.

2.4 Case study: Differential interactions analysis of K562 and GM12878 cells

Performance as a function of the sample size

In our second set of comparisons, we performed a computational experiment with our actual Hi-C datasets by comparing contact maps of K562 and GM12878 cells (Rao et al., 2014). The data were normalized using square root vanilla coverage

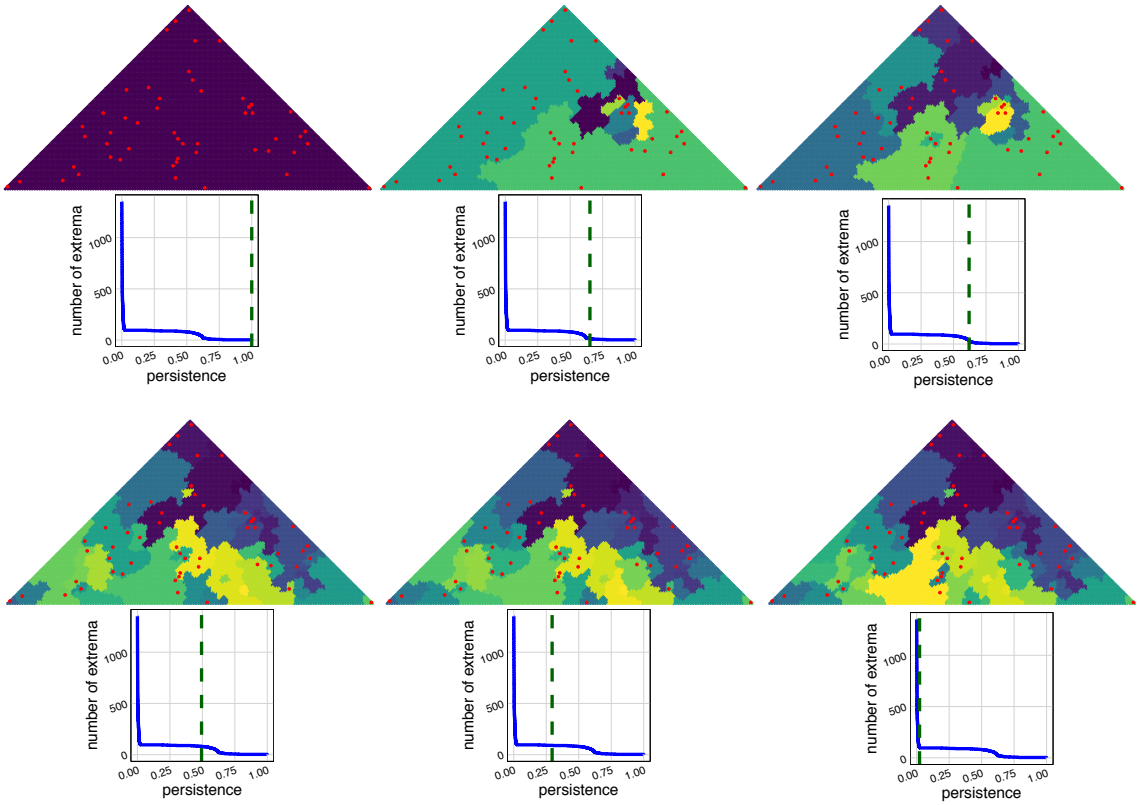


Figure 2.2: An example of 2D log fold-change function from simulated data at different levels of details as persistence varies. Sequence of upper triangles show the domain of Hi-C contact maps. In each triangle, we partition the Hi-C domain at a selected persistence level (green vertical line) where regions with different colors in each triangle present different partitions. Below each triangle, we include a persistence graph whose x-axis shows the number of extrema, and y-axis presents persistence values. Each red dot denotes an extremal point of the function. Green vertical lines encode the selected persistence values.

(VC) normalization (Rao et al., 2014). For our real data analysis, the data processing and normalization steps are discussed in details in Appendix A.3. Furthermore, we complemented this dataset with RNA-seq from the same system, with two replicates per condition. For each of diffHiC, FIND, tree-diffHiC, and tree-perm, we first performed differential interaction analysis with sample sizes $n = 2, 3$ and 4,

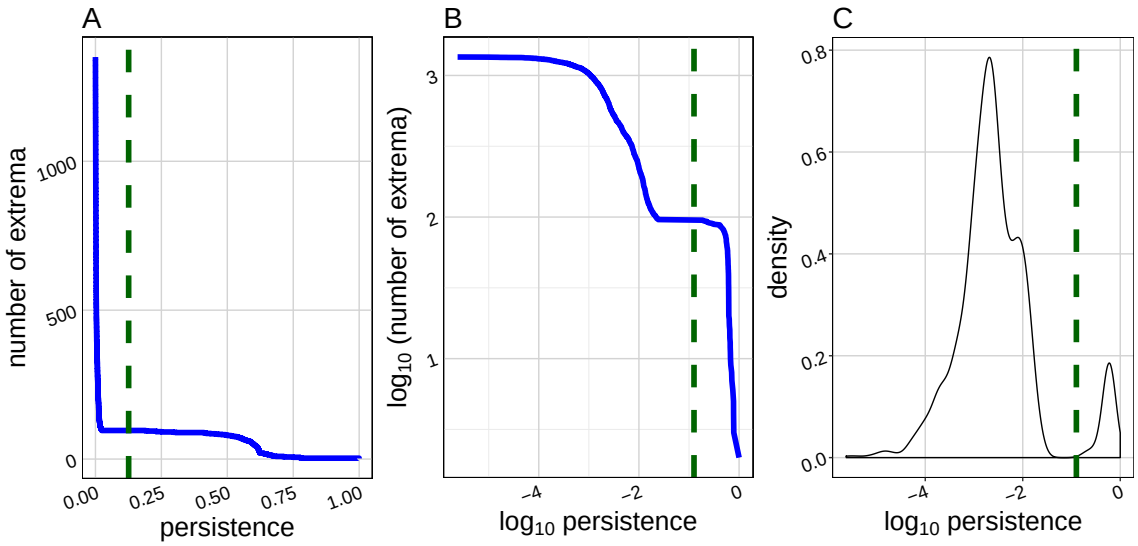


Figure 2.3: Different versions of persistence graph where it exhibits a clear plateau separating noise from features (green vertical lines). Data are from a sampled region of chromosome 22 of K562 and GM12878 cells. (A) Persistence as a function of number of extrema. (B) Persistence graph on log-log scale. (C) Density of $\log_{10}$ (persistence).

according to the settings recommended by the developers of the tools. We utilized the best ranking peaks according to their p-values, and utilized the best ranking peaks identified by sample size of 4 as the gold standard differential interactions. We then report the proportions of peaks that overlap with the gold standard ones when sample size is downscaled. The results are displayed in Figure 2.9. It is clear that both versions of tree methods have larger proportions recovered from the gold standard peaks compared to the other tools. Also, the power to detect differential interactions improves with increased sample size for all methods. Interestingly, both diffHiC and its tree version's reproducible proportions are less robust under change of sample sizes, whereas FIND and tree-perm show less changes in proportions recovered. This suggests that count-based approach diffHiC is more suitable for Hi-C datasets with relatively large sample size.

TreeHiC shows better FDR control than count-based methods on real data

We next performed differential interaction detection by comparing Hi-C contact maps under the same biological condition. We aimed at evaluating the number of interaction findings across methods by considering subsets of samples against each other within a condition. Here, we utilized four biological replicates from K562 (or GM12878). We perform a similar analysis described above by testing two pairwise samples from four replicates. Specifically for this analysis, we perform a single split of the four replicates; we fix the first two samples, and test them against the remaining twos. Here, we also provide sequencing depths for each studied cell lines. The sequencing depths for K562's four samples are 47.2, 48.1, 46.4, and 44.9 where values are $\times 10^6$. For GM12878, number of reads are 197.0, 202.4, 146.1, and 59.7 (values $\times 10^6$). The results are displayed in Figure 2.10. It is clear that tree methods show markedly better FDR control than those of FIND and diffHiC. This observation is similar to that of our simulation studies. We also observe that FIND has a large number of called interactions, especially in GM12878. For this cell line, sequencing depths of its replicates were unbalanced. This would contribute to a large number of false selections.

Differential interactions detected by TreeHiC have higher coverage of differential expressed genes

We next corroborated differential interactions (DIs) with changes in gene expression to evaluate TreeHiC's performance with biological functions. DESeq2 (Love et al., 2014) analysis of RNA-seq read counts from the K562 and GM12878 cell lines was performed at FDR of 0.05. We enlarge the impact regions of each gene by extending 2Kb upstream of the transcription starting site to include the promoters and overlap them with DIs. Differentially expressed genes are identified as those with absolute value of log2 transformation of fold change larger or equal to 2 and adjusted p-value from DESeq2 to be smaller or equal to 0.05. We first assess the fraction of all the

differential express (DE) genes that are associated with DIs. To make it comparable, top 5000 significant DIs for four methods are selected respectively. TreeHiC using permutation (tree-perm) stands out in Figures 2.11A and B indicating that under 10K and 50K resolution, TreeHiC is better at identifying larger proportion of DE genes that reside in regions of differential 3D interaction.

Additionally, we evaluated the power of TreeHiC compared to diffHiC and FIND in detecting DIs coupling with DE genes at FDR 0.05. Figures 2.11C demonstrates TreeHiC, either using diffHiC or permutation p values, has dominated power in detecting DIs that are related to gene regulation which is also stable across different interaction genomic distances. Supplementary Figure 2.12 offers a comprehensive comparison among different settings to define differential expressed genes. Furthermore, the number of DE genes covered by distinct interactions varies. By breaking down the proportion of DIs that have DE genes by the maximum DE gene number of two bins for each interaction, Figure 2.11 revealed that DIs identified by TreeHiC, tree-diffHiC and tree-perm, covers a largeumber of DE genes associated with each contact. Such trends are more obvious for top significant DE genes (Supplementary Figure 2.13). All of these results (Figure 2.11) indicated that TreeHiC tends to detect larger numbers of functional DIs compared to the other methods.

2.5 Conclusions

Very few tools to identify differential interactions between experimental conditions from Hi-C data have been developed in recent years. In general, these tools vastly differ in term of usability and in range of applicability, falling short of fully exploiting the power of Hi-C data. Such shortcomings become apparent as none of the methods formally address the rate of false discovery and their performance for reported findings, especially in high-resolution Hi-C studies which involved in testing a large number of hypotheses. To overcome these limitations, we proposed TreeHiC, a hierarchical testing procedure for quantitative comparison applied to Hi-C data. Different from all the currently available methods, TreeHiC formally addresses and resolves three critical issues in large-scale Hi-C studies: (i) the exis-

tence of sparsity and weak signal-to-noise ratios in high-resolution Hi-C differential analysis, (ii) the FDR control and performance for reported findings, and (iii) lack of biological experiments. Additionally, TreeHiC is practically scalable, as its procedure is readily accompanied with different models. Lastly, while the current version of TreeHiC implements methodology pertaining to Hi-C differential analysis, it is easily extendable for other similar data such as ChIA-PET and HiChIP.

Software

TreeHiC is implemented as an R package and is available at `https://github.com/duydnguyen/TreeHiC`.

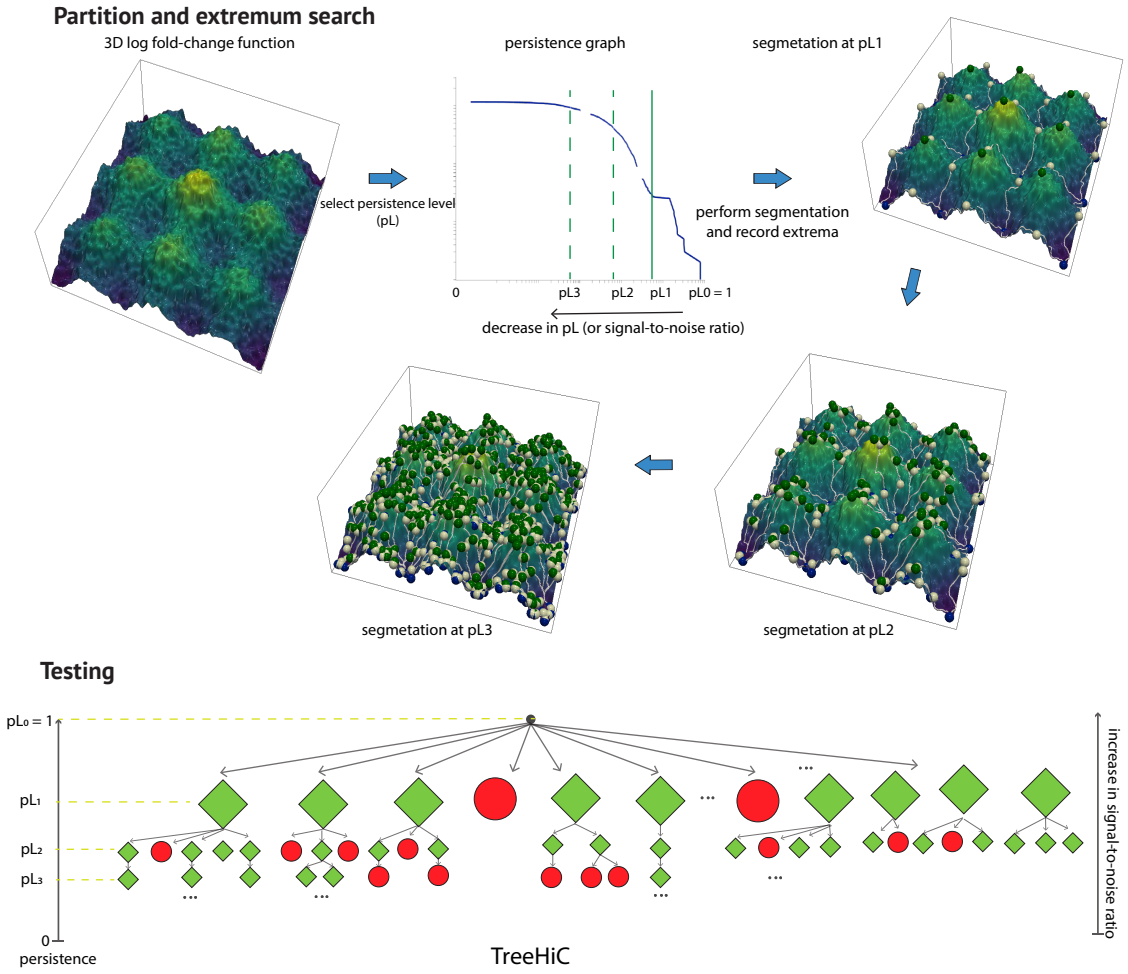


Figure 2.4: A workflow of TreeHiC's framework. (Top) Partition and extremum search. (Bottom) Testing phase . The 2D log-function function was simulated for illustration where green, blue, and white spheres present local maxima, minima, and saddle-points, respectively. Here, for illustration purpose, we elevate the log fold-change function in 3D (3D version of h) to show the extrema. This is useful to see the evolution of extrema at different persistence values.

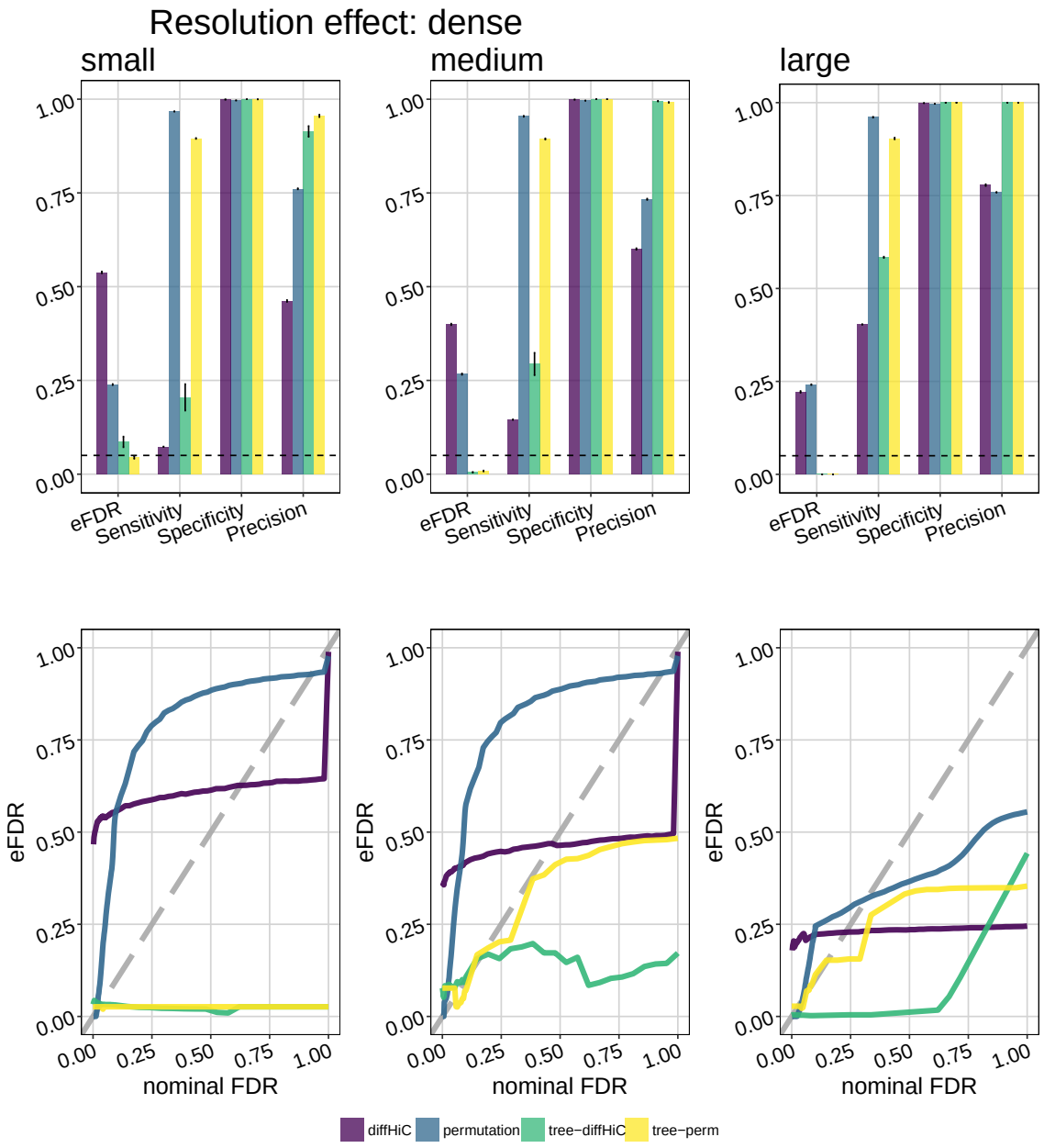


Figure 2.5: Performance summary for simulation model 1 at low resolution on simulation replications at FDR level of 0.05, and at varying level of controlled FDR. FIND is not included since it requires at least 2 replicates per condition. Averages are calculated over 10 runs of simulated data sets. Small, medium, and large present different fold change values in our simulation. eFDR: empirical FDR.

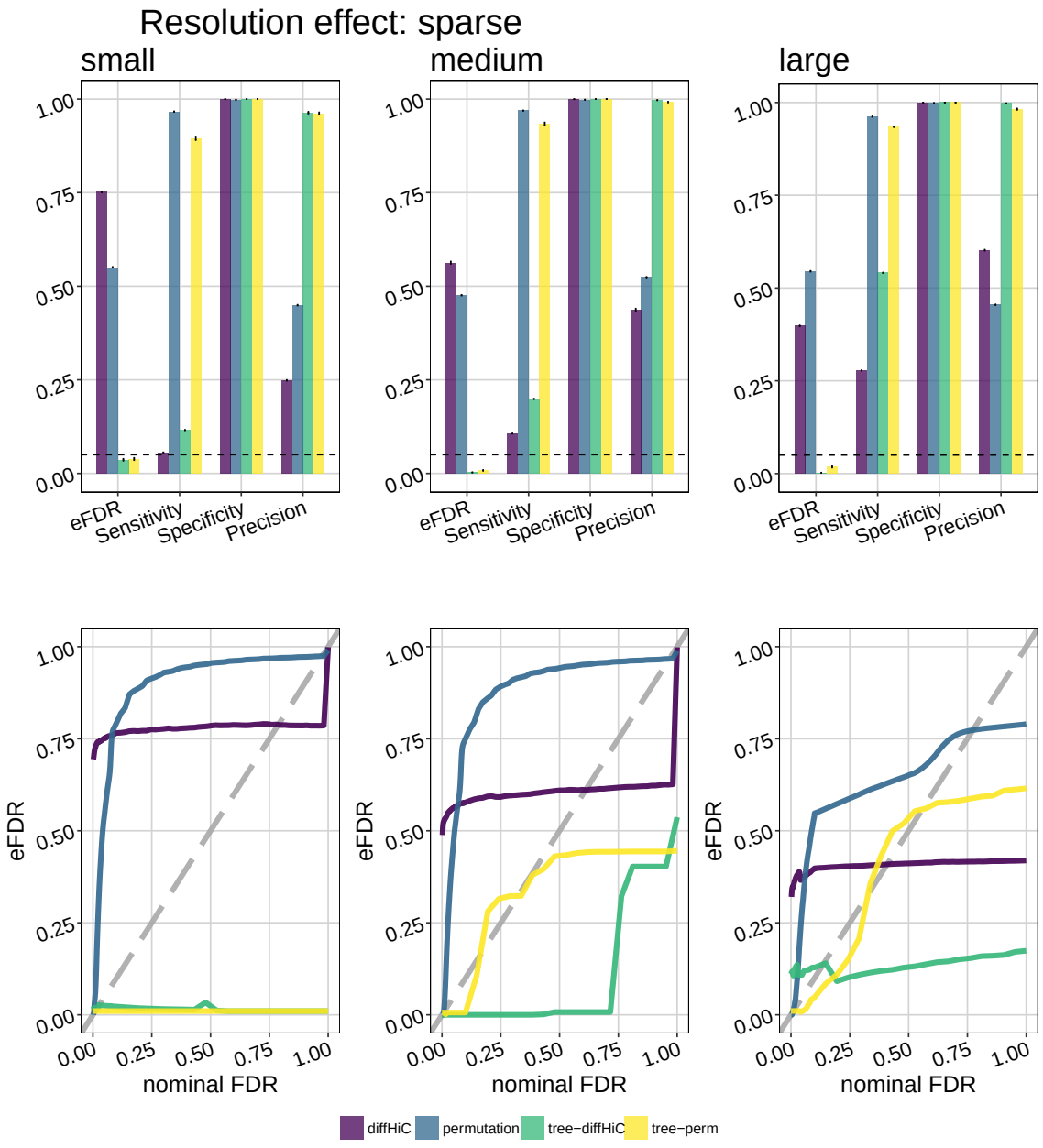


Figure 2.6: Performance summary for simulation model 1 at high resolution on simulation replications at FDR level of 0.05, and at varying level of controlled FDR. FIND is not included since it requires at least 2 replicates per condition. Averages are calculated over 10 runs of simulated data sets. Small, medium, and large present different fold change values in our simulation. eFDR: empirical FDR.

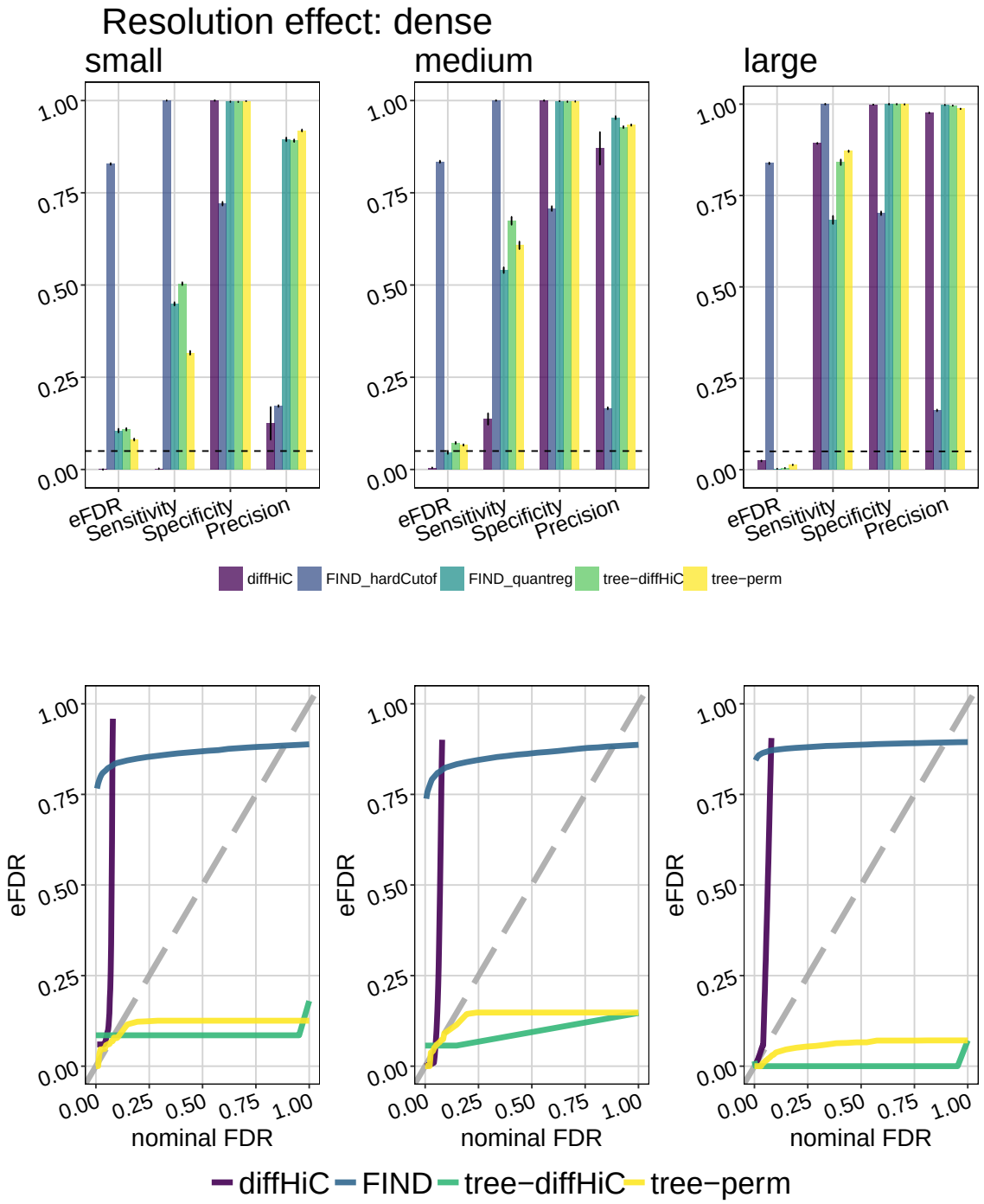


Figure 2.7: Performance summary for simulation model 2 at low resolution on simulation replications at FDR level of 0.05, and at varying level of controlled FDR. Averages are calculated over 10 runs of simulated data sets. Small, medium, and large present different fold change values in our simulation. eFDR: empirical FDR.

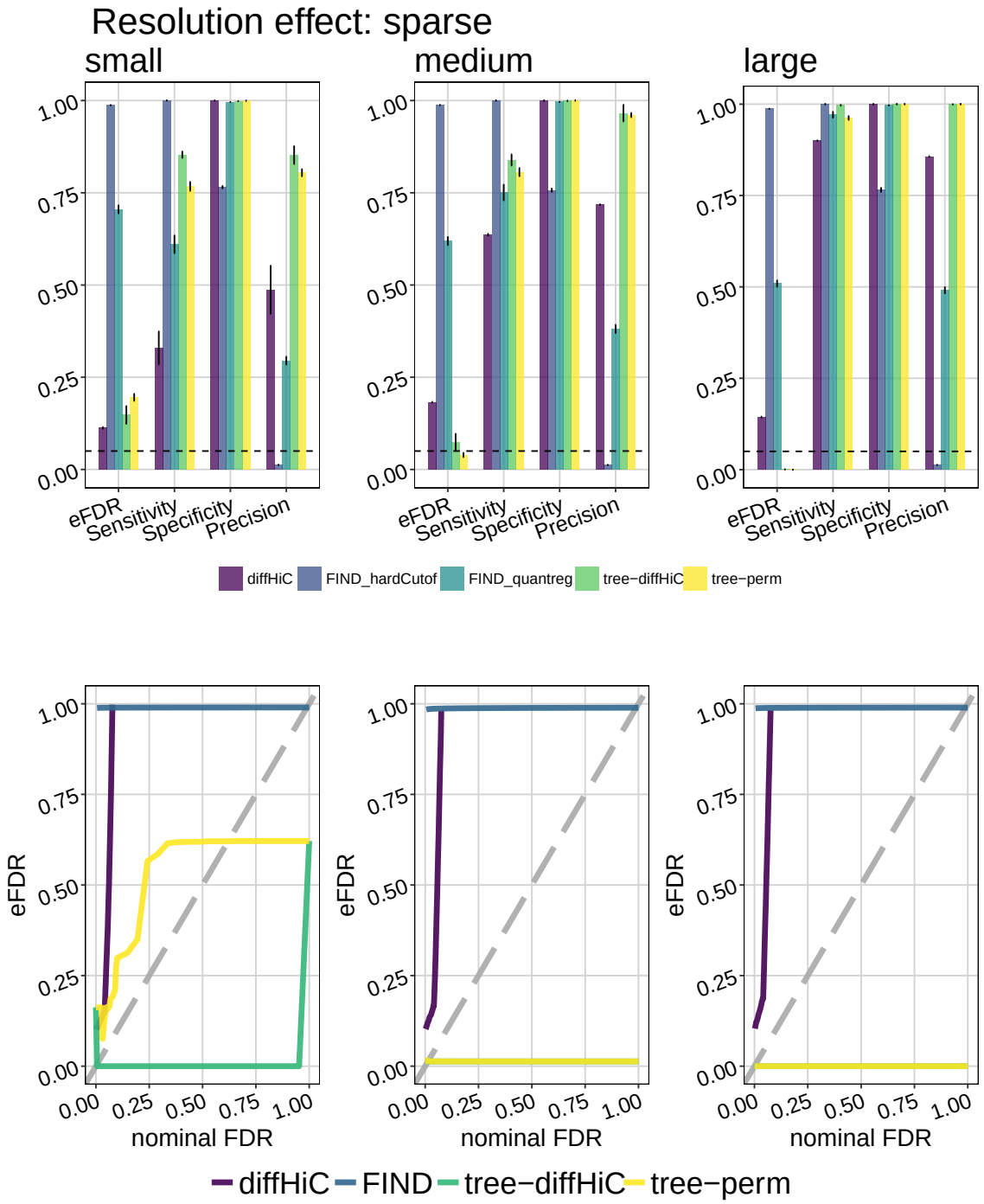


Figure 2.8: Performance summary for simulation model 2 at high resolution on simulation replications at FDR level of 0.05, and at varying level of controlled FDR. Averages are calculated over 10 runs of simulated data sets. Small, medium, and large present different fold change values in our simulation. eFDR: empirical FDR.

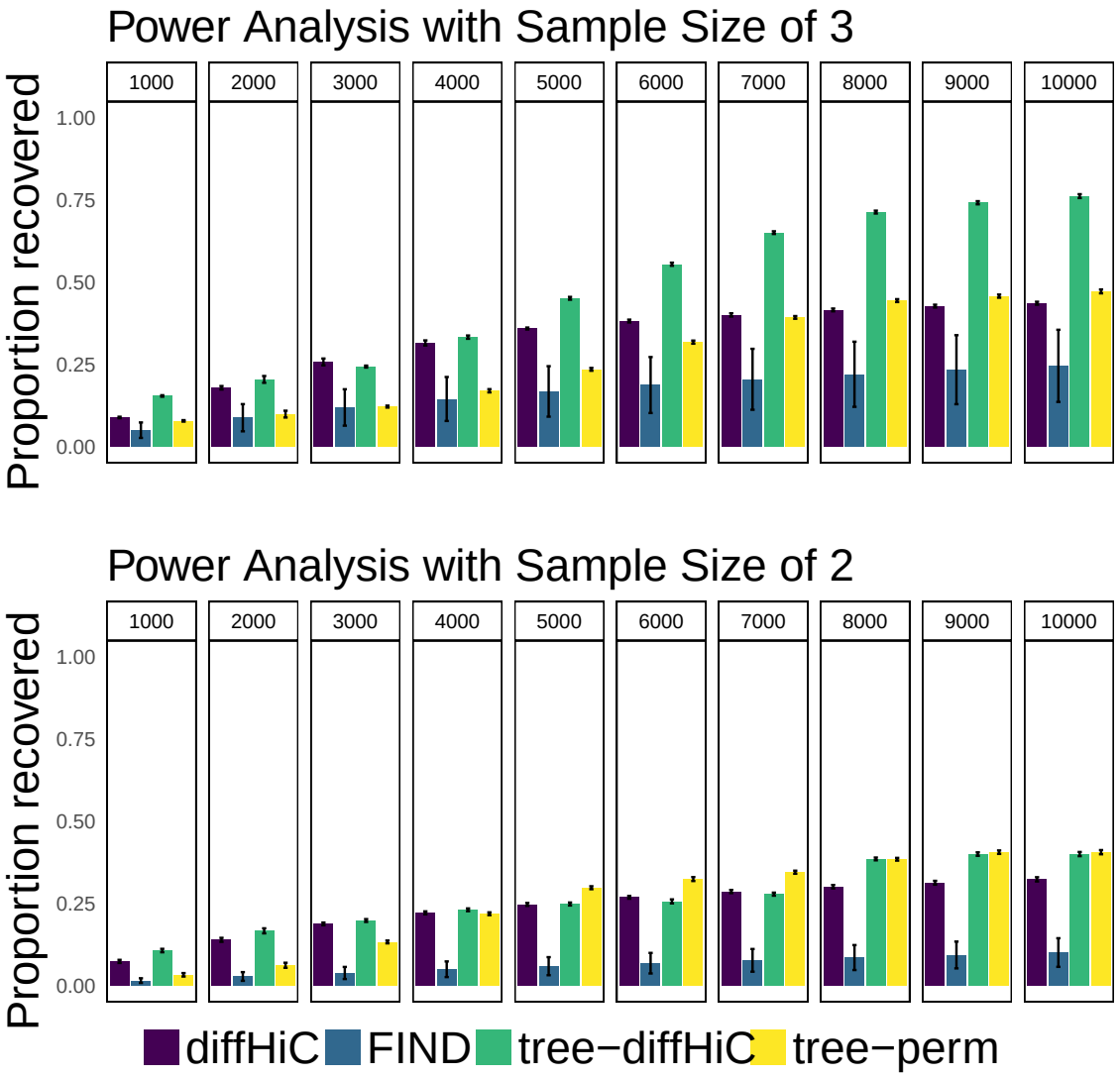


Figure 2.9: Reproducibility of differential interaction detection across methods at 50-kb resolution. (Top) Power analysis with sample size of 3. (Bottom) Power analysis with sample size of 2. In each vertical panel, top peaks are ranked based on p-values. Those of sample size 4 are considered gold standard differential interactions. For each sample size, the proportions of peaks that overlap with the gold standard ones are shown on the y-axis. The average proportions over different choices of sample sizes are shown, together with their error bars.

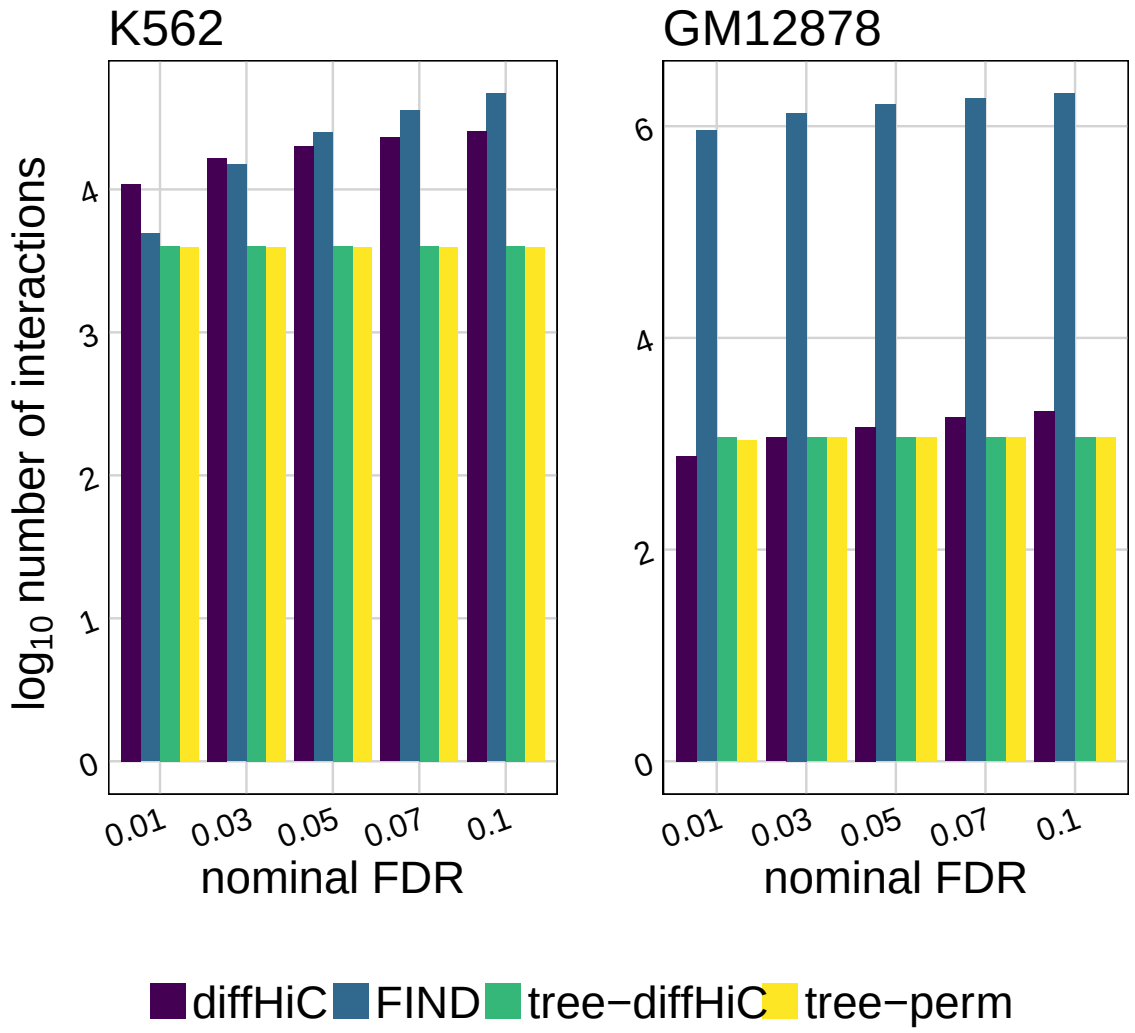


Figure 2.10: FDR control across methods at 50-kb resolution Hi-C contact maps of K562 (Left) and GM12878 (Right).

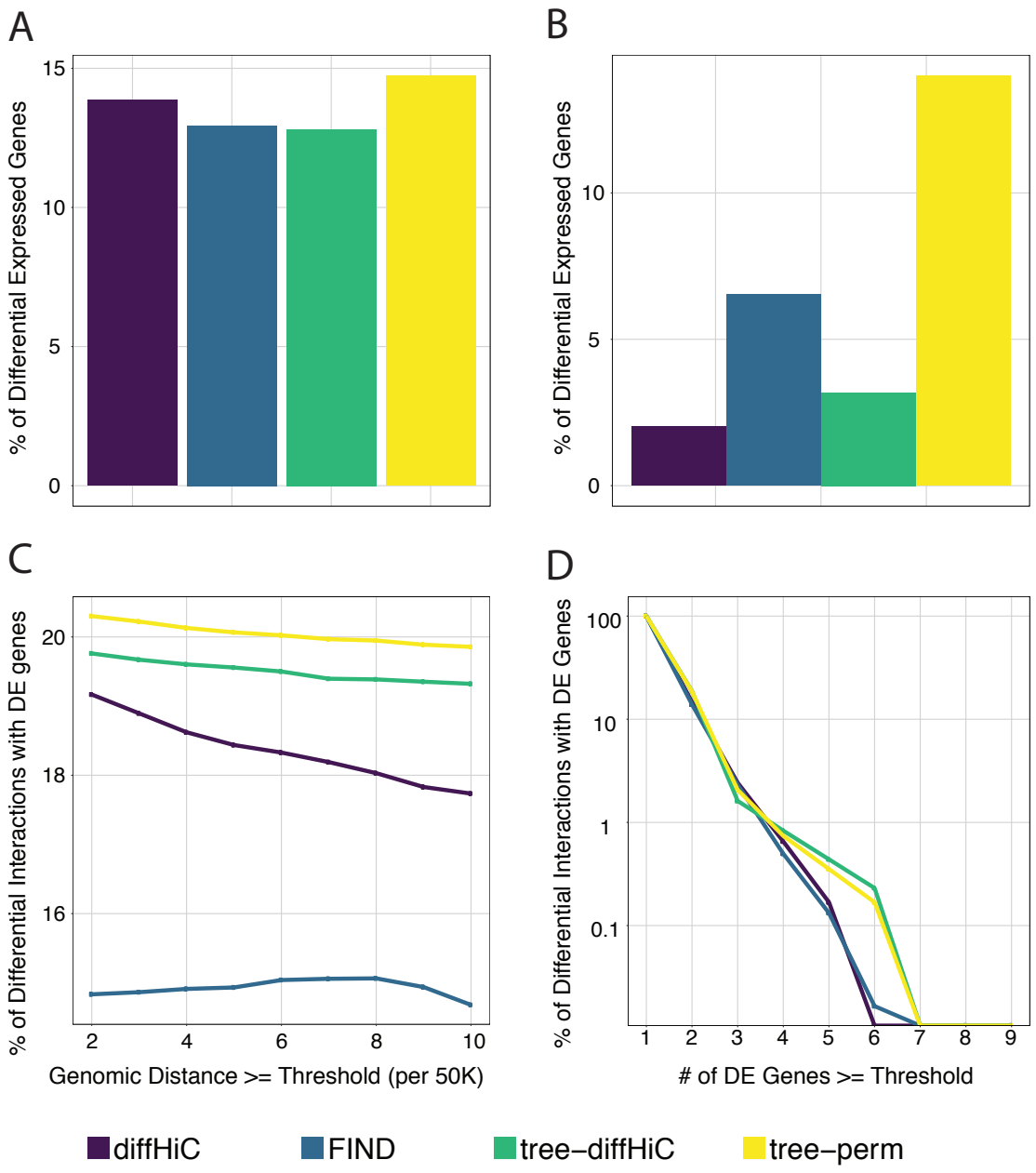


Figure 2.11: Differentially expressed (DE) genes involved in Hi-C differential interactions (DIs) of K562 and GM12878 cells. (A-B) The proportion of DE genes located within top 5000 significant DIs at 50K (A) and 10K (B) resolution. (C) The proportion of DIs that have DE genes in either of the two contacting bins at 50K resolution with respect to different interaction genomic distances. (D) The proportion of DIs that have DE genes in either of the two contacting bins at 50K resolution with respect to different numbers of DE genes involved in each interaction.

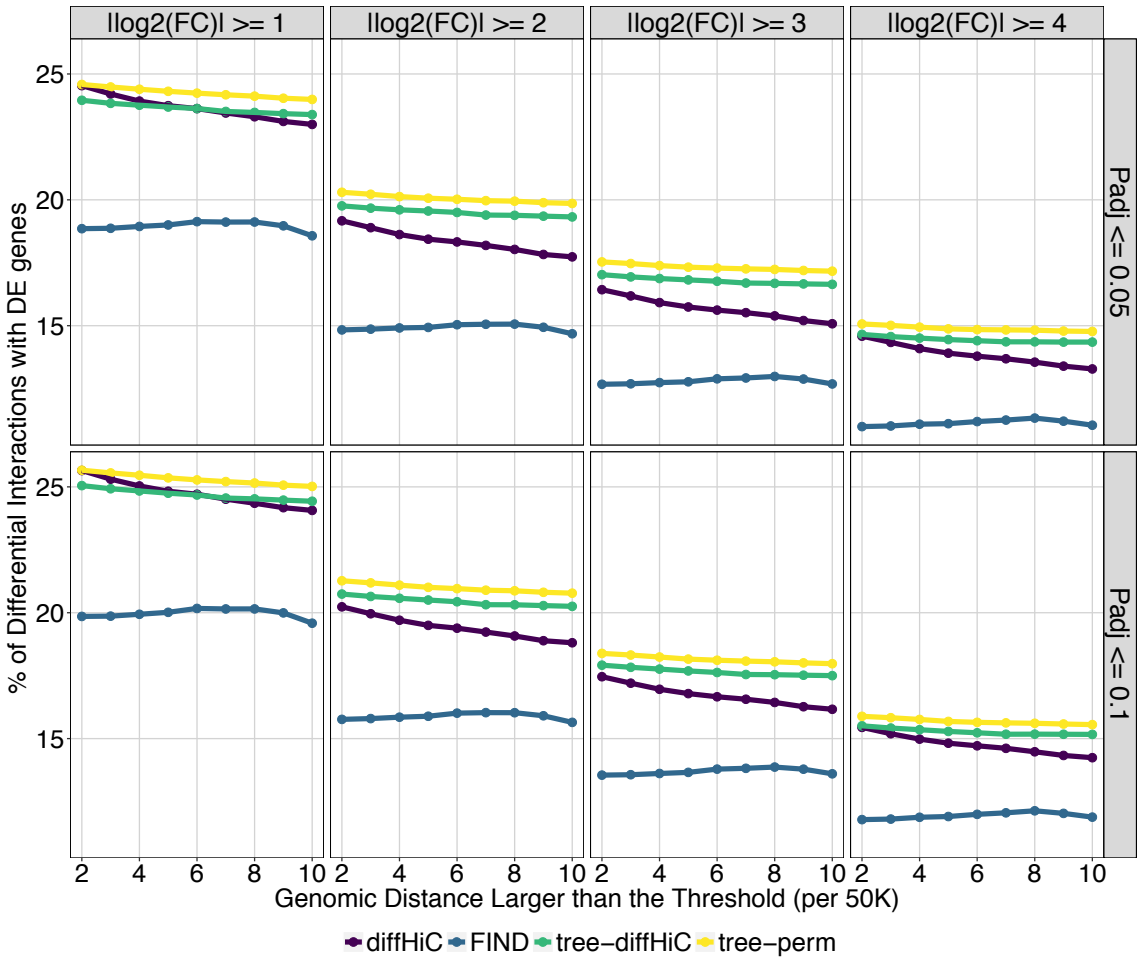


Figure 2.12: The proportion of DIs that have DE genes in either of the two contacting bins at 50K resolution with respect to different interaction genomic distances for K562 and GM12878 cell lines. Differential expressed genes are defined under different log2 fold change and adjusted p values thresholding settings.

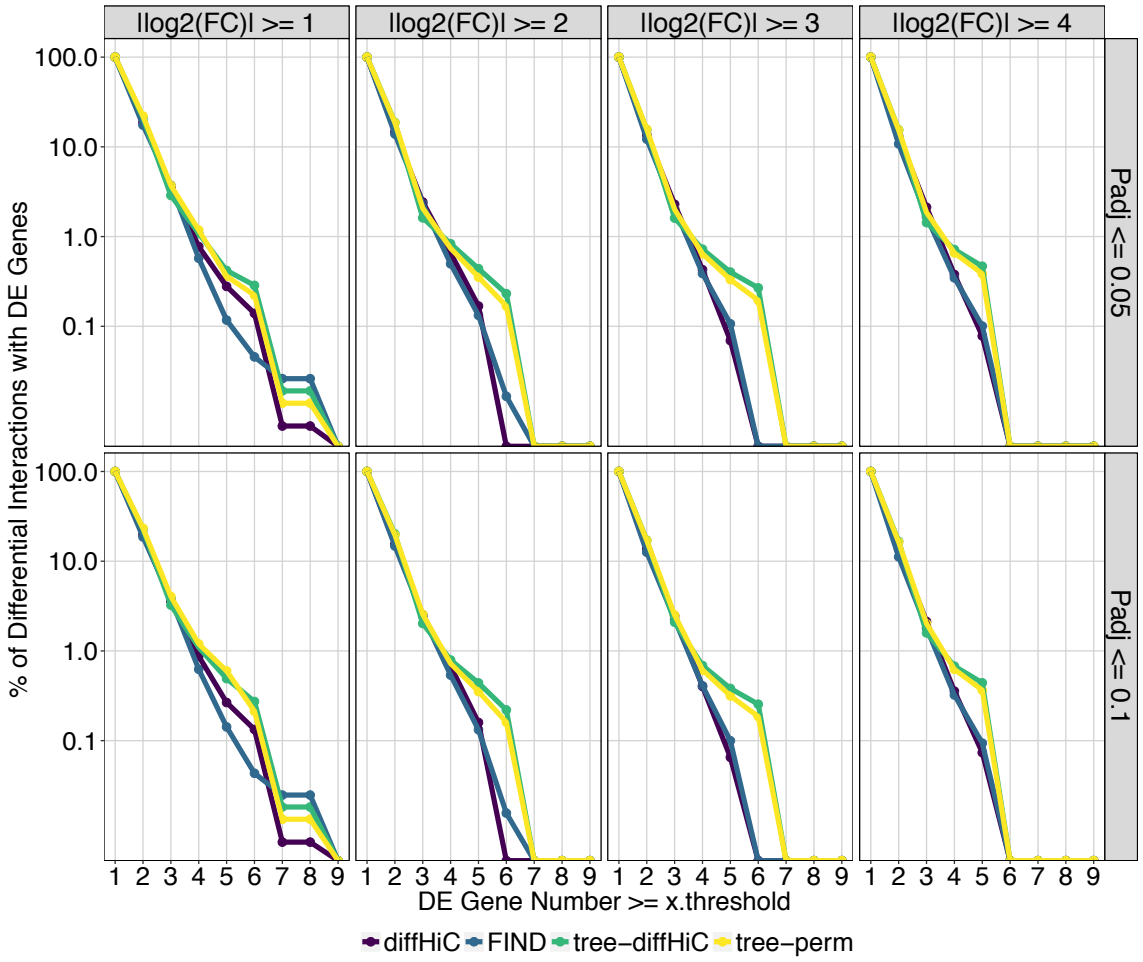


Figure 2.13: The proportion of DIs that have DE genes in either of the two contacting bins at 50K resolution with respect to different numbers of DE genes involved in each interaction for K562 and GM12878 cell lines. Differential expressed genes are defined under different log2 fold change and adjusted p values thresholding settings.

3 IDENTIFYING DIFFERENTIAL HISTONE MODIFICATIONS FROM CHIP-SEQ DATA

3.1 Background

Chromatin immunoprecipitation followed by high throughput sequencing (ChIP-seq) generates high resolution genome-wide profiles of histone modifications (HMs) (Wang et al., 2008) and transcription factor (TF)-DNA interactions (Kharchenko et al., 2008; Robertson et al., 2007). Many methods have been developed for processing data from ChIP-seq experiments. Popular applications of ChIP-seq include identifying and comparing genomewide profiles of TFs and HMs across different samples and cellular conditions. Both TF binding and histone modifications play important roles in condition-specific gene regulation; therefore, quantitative comparison of ChIP-seq (often referred as "differential (enrichment) analysis" problem) is necessary to understand the dynamics of these processes.

Differential enrichment analysis has recently gained considerable interest and led to development of several methods for the two-condition comparison (Zhang et al., 2008; Shao et al., 2012; Heinz et al., 2010; Liu et al., 2013; Shen et al., 2013; Chen et al., 2015; Ross-Innes et al., 2012). We present in Figure 3.1 a decision tree indicating the applicability of several tools for specific tasks in differential enrichment analysis. There are a number of features that make the differential ChIP-seq analysis different than differential expression analysis of RNA-seq, which is a comparatively well-studied problem (Pepke et al., 2009; Schweikert et al., 2013). First, in most ChIP-seq studies, only a very small number of biological replicate experiments, e.g., 2 to 3, are performed owing it to the fact that the primary objective of these studies is detection of peaks. This makes statistical inference for comparing across multiple conditions a challenging task. Second, the signal from HMs present spatially distributed patterns localized to specific regions of the genome (O'Geen et al., 2011). These are more challenging to identify compared to TF binding sites by directly applying the existing peak-calling methods due to their varying lengths

and lower signal to noise. In addition, the search space is not limited to a particular region of the genome (e.g., transcripts, genes) since most histone modifications are observed in non-coding regions. This implies that the regions of interest, for which differential enrichment analysis is performed, need to be defined first.

Two strategies have emerged to address these unique challenges. The first and most straightforward method is the "overlap analysis", which is to apply a peak-calling algorithm (e.g., Zhang et al. (2008); Kuan et al. (2011)) to identify regions with significant ChIP signal, i.e., peaks, for each of the two conditions. The regions with peaks in one condition but without peaks in the other are then deemed as differentially enriched (Schmidt et al., 2010; Chikina and Troyanskaya, 2012; Nostrand and Kim, 2013). However, such a comparison is highly dependent on the thresholds/error rates used for peak calls. Regions (peaks) barely over the threshold in one sample but under threshold in the other will be declared as condition-specific peaks even if the quantitative differences are small. In addition, this approach completely ignores the quantitative comparison of the genomic regions that are identified as peaks in both samples (i.e., common peaks) even when the quantitative differences are large.

An alternative strategy to the overlap analysis is to compare the read counts of predefined regions or peaks identified in individual samples and test for differences in the magnitude of the read counts across conditions. Several parametric methods (DiffBind (Start and Brown, 2011) and DESeq (Anders and Huber, 2010)) based on Poisson and Negative binomial distributions are in this category. The inputs to these methods rely on identifying the ChIP-enriched regions and quantifying their read counts. These *count-based* approaches mostly adapt the methods for RNA-seq differential expression analysis to the more structured ChIP-seq data. A notable drawback of this approach is that the properties of the enriched regions in ChIP-seq differ considerably depending on the protein or epigenetic modification targeted by immunoprecipitation. The count-based methods represent a peak by a single number, i.e., the (normalized) total counts of reads mapping to the candidate peak region. As a result, important spatial profile differences such as shifts of the signal region or shapes of signals or other higher order information that is part of peak

are prone to being lost.

Spatial structure in ChIP-seq signal is particularly evident in the case of peaks associated with epigenetic marks. For example, trimethylation of lysine 4 on histone H3 (H3K4me3) is known to form distinct bimodal peaks at transcription start sites, e.g. (Barski et al., 2007). Interestingly, at a given genomic location, the shape of enrichment peaks tend to be highly reproducible across biological replicates and increasing evidence hints towards a functional role of these profile structures (Consortium et al., 2012; Bieberstein et al., 2012). Peak shape properties different from signal intensity have been used in peak calling (Hower et al., 2011; Mendoza-Parra et al., 2013; Mahony et al., 2014) and peak ranking (Wu and Ji, 2014). Most recently, the shapes of peaks have been investigated for testing the hypothesis that peak shape is influenced by the organization and interactions of the proteins bound to the DNA (Cremona et al., 2015). This study demonstrated that ChIP-seq profiles include information regarding the binding of other proteins beside the one used for immunoprecipitation. In particular, peak shape provides new insights into cooperative transcriptional regulation and is correlated with gene expression. Thus, there is mounting evidence that peak shapes have a functional role and a biological meaning. Focusing exclusively on total read counts of peaks is a significant limitation for performing differential analysis of epigenomic modifications.

As a third and under-utilized alternative, (Schweikert et al., 2013) proposed an approach for testing the differences in profiles of peaks in different conditions. MMDiff shows a better performance compared to other count-based methods in detecting localized changes that alter the shape of peaks. However, when considering affinity (total counts of reads) changes, MMDiff is significantly inferior to count-based methods. In practice, (Schweikert et al., 2013) suggests to combine MMDiff with a count-based method, especially for detecting regions that exhibit differential signal in terms of both shape and affinity. This is a major drawback since difference in signal strength between ChIP-seq samples is probably the most prominent feature for detecting differentially enriched regions. In addition, complementary use of MMDiff can result in loss of false discovery rate (FDR) control compared with a single testing framework. Another and, perhaps more practical, drawback

of MMDiff is that the *maximum mean discrepancy* (MMD) statistic (Schweikert et al., 2013) it utilizes is computationally intensive. As a result, MMDiff is not practical for differential analysis of a large set of broad peaks (Steinhauser et al., 2016).

Here we describe a novel approach, TAN, for differential enrichment analysis of histone modification ChIP-seq data. TAN relies on a new test based on the adaptive Neyman test introduced by (Fan and Lin, 1998) and is applicable when the study design includes two or more biological replicates across two or more biological conditions. Our computational experiments and simulations illustrate that TAN is powered for detecting both changes in affinity and higher order changes such as peak shapes, giving it a clear advantage over existing differential enrichment approaches. We utilized TAN to study of the role of Epstein-Barr nuclear protein EBNA3C in gene repression with H3K27me3 ChIP-seq from LCLs in which a conditional EBNA3C is active or inactive. TAN identified differentially modified regions that are missed by an affinity based differential analysis. Integrating these regions with RNA-seq from the same conditions and experimental validation with ChIP-qPCR highlight the power of TAN for ChIP-seq differential analysis in low signal to noise settings.

3.2 The TAN modeling framework

Differential analysis is usually performed by a two-step procedure. The first step is to identify a set of genomic intervals over which the differential enrichment is to be assessed. The TAN framework allows input peaks from any peak calling algorithm and defaults to using mosaics (Kuan et al., 2011) in the absence of a user-defined set of peaks. We then utilize the union of peaks identified from all datasets as the candidate regions. Since the first step is well-studied in the literature, we will focus on our method for quantitative comparison between two conditions. Extension to multiple conditions is discussed in Appendix B.1.

For a two sample comparison, we have observations denoted by $X_{ij}(t)$ and $Y_{ij}(t)$, $1 \leqslant i \leqslant N$, $1 \leqslant j \leqslant n_1, n_2$, and $1 \leqslant t \leqslant T$. Here X and Y denote the two experimental conditions, i represents index for genomic regions, j is the index for

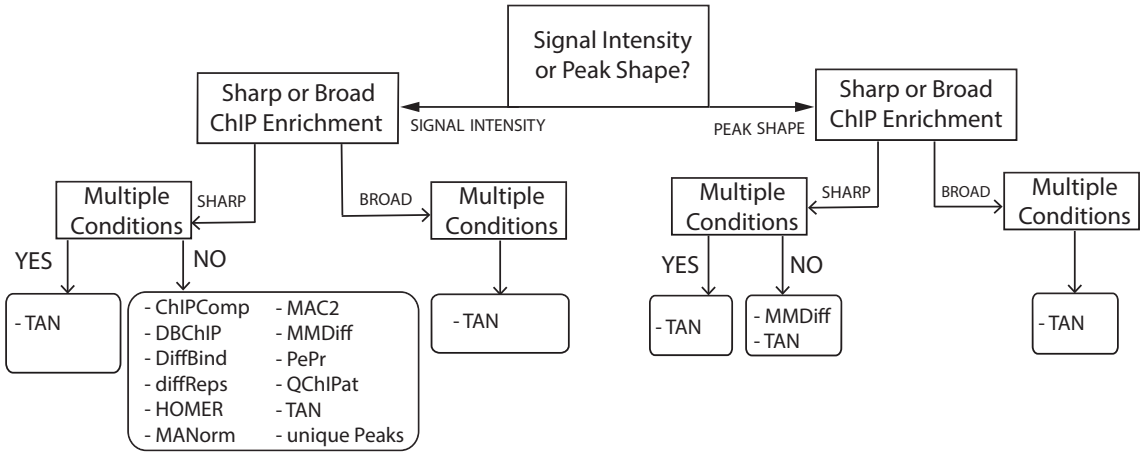


Figure 3.1: A decision tree indicating the applicability of various algorithms in ChIP-seq analysis for peak feature detection (signal intensity or peak shape), shape of the signal (sharp or broad peaks) and the presence of multiple conditions (two or more than two conditions).

samples within each condition, and t is the index for genomic positions ($1 \leqslant t \leqslant T$). Note that under this notation, collection of $X_{ij}(t)$, $t = 1, \cdots, T$, where T denotes the length of the genome, represents a ChIP-seq curve. We consider observations (i.e., normalized read counts) $X_{ij}(t)$, where $\mathbb{E}[X_{ij}(t)] = f_{i1}(t)$ and $\text{Var}[X_{ij}(t)] = \sigma^2_{i1}(t)$. Similarly for the second condition, we observe $Y_{ij}(t)$, where $\mathbb{E}[Y_{ij}(t)] = f_{i2}(t)$ and $\text{Var}[Y_{ij}(t)] = \sigma^2_{i2}(t)$.

Negative binomial distribution is commonly used to model count data in the presence of over-dispersion (Steinhauser et al., 2016; Anders and Huber, 2010; Robinson et al., 2009). Under this model, $X_{ij}(t)$ and $Y_{ij}(t)$ follow $\text{NB}(\mu_{ij}(t), \sigma^2_{ij}(t))$, with two parameters: the mean $\mu_{ij}(t)$ and the variance $\sigma^2_{ij}(t)$. Assuming $X_{ij}(t)$ and $Y_{ij}(t)$ for $t = 1, \ldots, T$ follow "signal+white noise" model, we can write $X_{ij}(t) = f_{i1}(t) + \epsilon_{ij}(t)$ where $\epsilon_{ij}(t) \sim N(0, \sigma^2_{i1}(t))$. The expression for $Y_{ij}(t)$ is similarly defined.

Testing differential modification at peak i is equivalent to testing $H_0 : f_{i1} = f_{i2}$ where f_{i1} and f_{i2} are the mean functions for their respective conditions. A standard approach for this testing problem in multivariate analysis is Hotelling's T-squared

test, based on the test statistics $T^2 = (\bar{X} - \bar{Y})'\hat{\Sigma}^{-1}(\bar{X} - \bar{Y})$. In this setting, $\bar{X}$ and $\bar{Y}$ are the sample means from both conditions, and $\hat{\Sigma}$ denotes a "pooled" covariance estimator. The key challenge in applying Hotelling's T-squared test in ChIP-seq context is that the dimension T of the data vectors is arbitrary, and $T > n_1, n_2$. In a typical ChIP-seq differential analysis, $\hat{\Sigma}$ is singular and the Hotelling's T-squared test is undefined. Hence, the choice of the dimension T is critical. To resolve this issue, we develop a novel test based on the adaptive Neyman methodology (Fan and Lin, 1998). Furthermore, our framework does not assume any parametric distributions on X_{ij} and Y_{ij}. All the parameter estimation of mean functions f_1, f_2, and variances $\sigma_{i1}^2, \sigma_{i2}^2$ are performed under a distribution-free setting.

Adaptive Neyman test statistics

We now describe the procedure for constructing the Adaptive Neyman test in the two sample setting starting with the mean curves:

$$\bar{X}_i(t) = n_1^{-1} \sum_{j=1}^{n_1} X_{ij}(t), \quad \bar{Y}_i(t) = n_2^{-1} \sum_{j=1}^{n_2} Y_{ij}(t),$$

and the variance estimators

$$\hat{\sigma}_{i1}^2(t) = \frac{1}{n_1 - 1} \sum_{j=1}^{n_1} \left(X_{ij}(t) - \bar{X}_i(t)\right)^2, \quad \hat{\sigma}_{i2}^2(t) = \frac{1}{n_2 - 1} \sum_{j=1}^{n_2} \left(Y_{ij}(t) - \bar{Y}_i(t)\right)^2.$$

Define the standardized difference as

$$Z(t) = \frac{\bar{X}_i(t) - \bar{Y}_i(t)}{\sqrt{\frac{\hat{\sigma}_{i1}^2(t)}{n_1} + \frac{\hat{\sigma}_{i2}^2(t)}{n_2}}}$$

and let $\mathbf{Z} = (Z(1), \ldots, Z(T))'$. When n_1 and n_2 are large and under the null hypothesis, we have approximately

$$\mathbf{Z} \sim N(\mathbf{0}, \mathbf{I}_T).$$

Now, the maximum likelihood ratio test for the above problem is $\|Z\|_2^2$, which tests for all components of $\mathbf{Z}$. However, it might be sufficient to test only some components since testing many dimensions aggregates noise and leads to loss of power of the test. For instance, Fan and Lin (1998) proposed a test that uses truncation of the components of a test statistics, and they reject H_0 for large values of $\sum_{t=1}^{m} Z(t)^2$. Fan and Lin (1998) proposed choosing m by

$$\hat{m} = \arg\max_{m} \left\{ m^{-\frac{1}{2}} \sum_{t=1}^{m} \left(Z(t)^2 - 1 \right) \right\}.$$

The adaptive Neyman test statistic in (Fan and Lin, 1998) is then defined as

$$T^*_{AN} = (\sqrt{2\hat{m}})^{-1} \sum_{t=1}^{\hat{m}} (Z(t)^2 - 1).$$

The term "adaptive" is used here to refer to this choice of m. We provide an illustration of this test with adaptive $\hat{m}$ in Figure 3.2. Here, sequencing depth-normalized H3K27me3 ChIP-seq coverage is displayed for gene bodies of TTYH2 and CLEC2D where the shaded regions depict the actual peak tested, and later validated by our ChIP-qPCR experiments. In addition, the dotted lines highlight values of $\hat{m}$. As illustrated by this example, there are broad regions of low enrichment with small log fold-change values: 0.36 and 0.48 for TTYH2 and CLEC2D, respectively. In this case the total coverage of the whole regions are very similar under the two conditions and makes the statistical assessment of differences a challenging task. Next, we consider testing the null hypothesis H_0 of equality of mean functions by the following testing procedures: the adaptive Neyman test where $m = \hat{m}$, the "ordinary" or "total" Neyman test where $m = T$, and the widely adapted count-based method, DESeq.

For the proposed adaptive Neyman test, we obtain $T^*_{AN} = 5.4$ for TTYH2 and 25.1 for CLEC2D. This test is equivalent to rejecting H_0 when T^*_{AN} is too large. As an alternative approach, the "total" Neyman test uses the whole peak region, yielding values of -7.1 and 12.8 for TTYH2 and CLEC2D, respectively. Note that there are

large regions where both signals show little differences, so the "total" Neyman tests are smaller than those in the adaptive version. This observation shows that testing too many dimensions accumulates large stochastic noise and hence decreases the discriminative power of the test. The price is reflected in magnitudes of the "total" Neyman test statistics.

As a second alternative, we consider affinity (total counts of reads) changes in the tested regions. DESeq yields p-values of 0.24 for TTYH2 and 0.45 for CLEC2D. With the proposed adaptive test, we obtain p-values of 0.0013 and 0.002 for TTYH2 and CLEC2D, respectively, yielding evidence of the tested regions to be differential between the two conditions. Here, our p-values are obtained by the testing framework discussed in the next section. Note that these changes might have been identified by DESeq or other count-based methods if regions with more accurate boundaries could have been considered. As illustrated in this example, the proposed approach has an attractive property, as it adaptively truncates ChIP-seq signals where the two conditions 4HT- and 4HT+ are sufficiently different. In particular, this proves to be useful when identifying broad regions with localized changes.

With some standardization, the asymptotic distribution of T^*_{AN} under H_0 is given by the standard extreme value distribution. This asymptotic distribution poses several challenges that prevent a direct application of the methodology in Fan and Lin (1998) for differential ChIP-seq analysis. The first challenge arises from the slow convergence rate of T_{AN} as noted by Fan and Lin (1998). Hence, the asymptotic distribution of T_{AN} is not well-calibrated for computing p-values. Another difficulty is due to the sample sizes typically encountered in ChIP-seq experiments. These are typically too low to get reliable variance estimates, especially for regions with very small counts. Furthermore, large variability in biological replicates is prevalent in real datasets. This implies that distributions of the same peak in different biological replicates might be more different than expected. The above testing procedure rejects the null hypothesis in almost all comparisons between biological replicates.

To illustrate these observations, we utilized a random set of $5,000$ peaks from our H3K27me3 ChIP-seq datasets. The data are from LCLs expressing a conditional EBNA3C protein which is functional in the presence (4HT+) and inactive in the

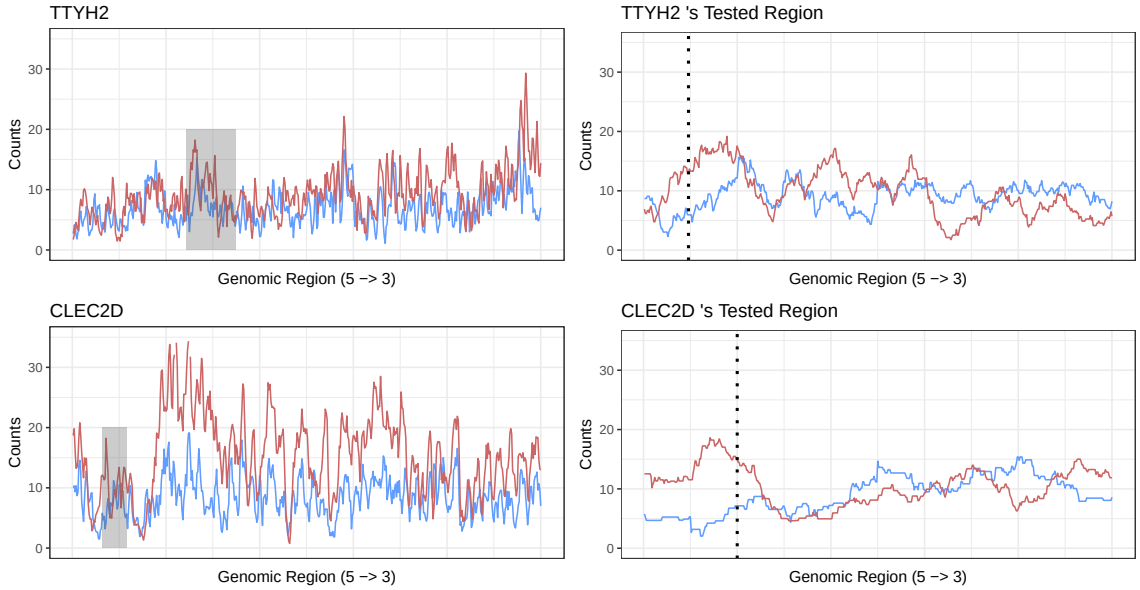


Figure 3.2: An illustration of adaptive Neyman tests in ChIP-seq context. Average H3K27me3 signals for 4HT-(blue) and 4HT+(red) over (Left) TTYH2 and CLEC2D gene bodies, and (Right) tested peak regions by our proposed method. Shaded regions mark the actual peaks being tested. Dotted lines represent the *adaptive* $\hat{m}$.

absence (4HT-) of 4-hydroxytamoxifen. To estimate the distribution of the test statistic under the null hypothesis of no differential enrichment, we compute the T_{AN} statistic for testing within biological replicates. Specifically, we divide the 4 replicates of each condition into two conditions of sample size two. The T_{AN} statistic is then evaluated for each peak, resulting in the empirical null distribution of T_{AN}. Figures 3.3A-B highlight that with a small number of replicates, the distribution of the test statistic under the null visibly deviates from the theoretical null. Specifically, we observe that T_{AN} for testing among biological replicates results in heavy-tailed density curves that are not centralized at 0 (Figure 3.3A). This is in contrast to asymptotic distribution of T_{AN} which exhibits a unimodal shape centralized at 0 (Figure 3.3B).

Comparison of the empirical null distribution of T_{AN} with the asymptotic null

distribution indicates that p-values from the extreme value distribution will reject the null hypothesis in almost all comparisons between biological replicates (Figure 3.3C). In Figure 3.3C, each data point corresponds to adaptive Neyman statistics when testing between pairs of biological replicates while the black line represents significant level α of 5% from the asymptotic distribution. It is evident that majority of points lie above this bound, resulting in a large number of differentially enriched regions in this null setting of biological replicate comparison setting.

As an alternative procedure, one could rely on permutations to get a null distribution. The permutation methodology is simple to understand and easy to implement. Thus, we randomly permute the condition label, and recompute the test statistics. The proportion of the values of T_{AN}'s larger than the test statistics for the observed data gives an estimated p-value. In Figure 3.3C, each red line presents an FDR cutoff at 5% obtained from permutation tests. Although permutation provides a small improvement compared to the asymptotic distribution, a large number of biological replicate comparisons are still declared differentially enriched. An explanation of this drawback is that permutation test requires the null to assume equal distribution (not just equal mean functions) for both conditions (Good, 2004). This is a stringent assumption, which is generally not held in ChIP-seq data analysis. As shown in Figure 3.3A , the two null distributions from different conditions deviate from each other. Further, while we note that this permutation scheme has problems due to small number of replicates within each group, similar behavior is still prevalent when we mitigate the sample size issue by a pooling scheme across genomic regions with similar total counts (see Section Testing for differential enrichment).

Variance estimation

Adaptive Neyman statistics require estimation of the variance for each genomic region. A direct approach is to use the sample variances $\hat{\sigma}^2_{i1}(t)$ and $\hat{\sigma}^2_{i2}(t)$ for genomic regions $i = 1, \ldots, N$. Small sample sizes of ChIP-seq study designs does not lead stable variance estimates. To improve precision, we pool data from regions

with similar counts under the assumption that the per-position variance parameter $\sigma^2_{i1}(t)$ and $\sigma^2_{i2}(t)$ are smooth functions of mean counts at position t,

$$\sigma^2_{i1}(t) = \nu_{i1}(f_{i1}(t)), \text{ and } \sigma^2_{i2}(t) = \nu_{i2}(f_{i2}(t)).$$

This assumption is similar to that used by DESeq (Anders and Huber, 2010). To obtain the smooth functions ν_{i1} and ν_{i2}, we use the following variance estimator

$$\hat{\sigma}^2_{i1}(t) = \text{ median }_{\text{ peak } p \in \text{Bin}(\text{ peak } i)}\{\hat{\sigma}^2_{p1}(t)\},$$

where $\texttt{Bin}(\text{peak } i)$ is the set of peaks with similar mean counts as peak i. Estimation of $\sigma^2_{i2}(t)$ is conducted in a similar manner. Although local and parametric regression can be alternative approaches, this procedure has an advantage over the approaches from (Anders and Huber, 2010) in terms of computational cost. Furthermore, variance estimation from Anders and Huber (2010) might lead to convergence issues (Landau and Liu, 2013). The proposed variance estimation performs well in our simulation studies.

Testing for differential enrichment

Figure 3.4 summarizes TAN's workflow for estimating the null distributions empirically from data. This workflow consists of two main phases: (i) generating empirical null distributions, and (ii) testing phase for reporting differentially enriched regions. In phase (i), TAN evaluates the (within) Neyman statistics between pairs of replicates within the same conditions. Then, it clusters these peaks into bins based on their total counts. These bins are simply partitions of read counts across the pre-defined genomic regions. Consecutively, TAN constructs the corresponding null distributions based on the statistics available from each bin (or partition) of counts. For the testing phase, TAN constructs the (between) Neyman statistics by testing replicates from different conditions. It then maps the regions being tested to their corresponding empirical null distributions based on regions' total read counts. Consequently, TAN provides the p-values for each peak, leading to the

identification of differentially enriched regions after multiple testing correction. This completes the testing phase.

Next we present the calculation of p-values in detail. Let n_1, n_2 be the sample sizes for two conditions, respectively. Furthermore, let $\{m_1^i, \ldots, m_{n_1}^i\}$ and $\{p_1^i, \ldots, p_{n_2}^i\}$ be labels for replicates from conditions 1 and 2 from peak i, respectively. Without loss of generality and to simplify the notation, we assume that the sample sizes are at least 4. For smaller sample sizes, we provide the details in Appendix B.2. Phase (i), the null distribution estimation phase, mimics a permutation test to generate an empirical null distribution. By considering subsets of samples against each other within a condition, it takes into account of biological variability within conditions as opposed to getting a single test from all the samples in each conditions.

For condition 1, we generate the "within" adaptive Neyman tests as follows. We create $\binom{n_1}{2}\binom{n_1-2}{2}/2$ pairwise samples from n_1 replicates to obtain adaptive Neyman tests T_j^{cond1}, for $j = 1, \ldots, \binom{n_1}{2}\binom{n_1-2}{2}/2$. The adaptive Neyman tests T_j^{cond2}, are computed in a similar way with samples from the other condition. Under the null hypothesis of no DE, we could consider that T^{cond1}'s and T^{cond2}'s are sampled from the same null distribution.

Let $S_- = \{(m_j^i, m_{j'}^i) : \text{where } (m_j^i, m_{j'}^i) \text{ are from pairwise samples in condition 1}\}$, and $S_+ = \{(p_j^i, p_{j'}^i) : \text{where } (p_j^i, p_{j'}^i) \text{ are from pairwise samples in condition 2}\}$. The "between" condition adaptive Neyman tests $T_j^{bt}, j = 1, \ldots, \binom{n_1}{2}\binom{n_2}{2}$ result from testing one pair of sample in S_- versus a pair in S_+. The corresponding p-value for T_j^{bt} is

$$p_j^{bt}\text{-value} = \frac{\left|t \in \{T^{cond1}\text{'s}, T^{cond2}\text{'s}\} : t \geqslant T_j^{bt}\right|}{|\{T^{cond1}\text{'s}, T^{cond2}\text{'s}\}|}.$$

The estimated p-values are expected to be highly variable owing to the small number of observations for p-value calculation. To alleviate this problem, we pool peaks with similar total counts to generate robust estimates of the p-values. Specifically, peaks are binned into pre-specified quantiles determined by the mean read counts across replicates. To obtain empirical p-values, we compute the probability of observing an adaptive Neyman test statistics between biological replicates in

the given bin, which is at least as large as the one observed for a given peak in the comparison between conditions. This leads to the following estimator:

$$p_j^{bt}\text{-value} = \frac{\left| t \in \cup_{\text{peak } p \in \text{Bin(test } j)} \{T_{\text{peak } p}^{\text{cond1}}\text{'s}, T_{\text{peak } p}^{\text{cond2}}\text{'s}\} : t \geqslant T_j^{bt} \right|}{\left| \cup_{\text{peak } p \in \text{Bin(test } j)} \{T_{\text{peak } p}^{\text{cond1}}\text{'s}, T_{\text{peak } p}^{\text{cond2}}\text{'s}\} \right|},$$

where $\text{Bin(test } j)$ is the set of peaks with similar mean counts as the replicates are used to compute T_j^{bt}. The final p-value for testing the hypothesis of differential enrichment at peak i is taken as the median of p_j^{bt}'s. The false discovery rate (FDR) is controlled for a given target value using the method of (Benjamini and Hochberg, 1995). Furthermore, we evaluated the performance of our method under various quantiles of obtaining the final p-values (Appendix B: Supplementary Tab. B.1 and B.2).

3.3 Simulations

We performed simulation studies to evaluate operating characteristics of TAN. Performance on the simulated data was assessed based on (1) the ability to attain a good power while controlling the false discovery rate (FDR), (2) the performance of each tool as the number of replicates decreases, and (3) the ability to detect diffuse signals and multiple local peaks. These three criteria are explored in the next subsections. We benchmarked TAN against the widely adapted count-based method, DESeq (Anders and Huber, 2010), and shape-based method, MMDiff (Schweikert et al., 2013), and a permutation test. Two versions of adaptive Neyman tests were used in the simulations: (i) the adaptive Neyman test with pooled variance (pTAN), and (ii) the adaptive Neyman test with unpooled variance (TAN).

Simulation 1

To evaluate model performance, we conducted a simulation with a parameter setting generated from our actual data. We followed the general simulation outline

of Schweikert et al. (2013) and simulated histone ChIP-seq across two conditions with 10,000 candidate regions (peaks). For each condition, reads for four replicates were generated following a Poisson process where the between-sample variation followed a gamma distribution. Specifically, we assigned a *true affinity count* to each peak. The affinity counts were sampled according to the distribution of total counts in our H3K27me3 ChIP-seq data. To generate sample-specific affinity values for each peak, we used a Gamma distribution with mean given by the affinity count for that peak. We further assumed the peak profiles to be bimodal. This was ensured by using a mixture of two Gaussians with different means, variances, and mixing parameters. For differential regions (or a "treatment" set), we randomly chose 100 peaks and introduced changes in their base affinity values. We also selected 100 peaks and changed their base profiles by altering the mixing parameter. We simulated the parameters of the noise level of the peaks by picking a random value from Gaussian distributions with means given by the true base values. In all of our simulations, we set a minimum range of 2 to 3 for fold change between signal and background. We repeated this procedure 10 times and reported results over these runs. Appendix B: Supplementary Fig. B.1 further presents a detailed illustration of generating a set of ChIP-seq peaks.

Power Performance and FDR control

We first investigated the sensitivity, specificity, and the empirical FDR of methods compared. Figure 3.5A summarizes results on the affinity and profile changes separately at target FDR of 0.05. We also evaluated the receiver operator characteristic (ROC) curves for each method (Figures 3.5B-D) by varying the threshold for the corresponding test statistics or p-values. Overall, both versions of adaptive tests show a markedly better performance compared to DESeq and MMDiff (Figure 3.5B). In the case of affinity change (Figure 3.5C), DESeq performs best, and pTAN shows a competitive performance. Good performance for DESeq was expected since the count data in these simulations were generated from negative binomial distribution satisfying DESeq modeling assumption.

The unpooled TAN's sensitivity is low, but higher than that of MMDiff. Interestingly, at FDR of 0.05, MMDiff's sensitivity is smaller than that of pTAN, indicating pTAN's better performance in detecting profile changes while controlling FDR. As expected, count-based method DESeq cannot capture shape-based changes; its ROC curve essentially lies close to the diagonal (Figure 3.5D). As an alternative approach for computing p-values, we also included permutation tests. Here, two versions of permutation tests are also included in the simulations: (i) the unpooled permutation test (Perm), and (ii) the permutation test with a pooling scheme across genomic regions with similar total counts (pPerm). Similar to those observed in TAN and pTAN, the pooling version of permutation test shows a markedly better performance compared to its unpooled counterpart. However, both versions of permutation tests do not work well in either testing category, whereas pTAN performs well in both categories while controlling FDR well. This shows the effectiveness of the empirical information sharing by our pooling scheme across genomic regions. On simulated data where we know the ground truth, our analysis confirms that (1) the pooled version of TAN is required to attain good power while controlling the FDR, and (2) the adaptive Neyman test appears to be well-calibrated, with good power to capture both profile and affinity changes.

Performance as a function of the sample size

Next, we examined the ability of TAN to identify the set of non-null peaks at various sample sizes, and compared it to DESeq (Anders and Huber, 2010), and MMDiff (Schweikert et al., 2013). For each method, the target FDR was set at 5%. Figure 3.6 summarizes the performances as a function of sample size set at 2, 3, and 4, per condition. Our analysis on simulated data confirms that TAN with larger sample size has a higher rate of false negatives but markedly fewer false positives. As sample size increases, TAN identifies a much smaller number of differentially enriched regions, and achieves a much better precision, yet at the expense of the recall. These properties are similar to those of the tools that accommodate replicates (Steinhauser et al., 2016). In general, TAN controls FDR

better while maintaining good sensitivity as sample size increases. In the earlier simulation setting, our approach showed comparable performance to DESeq for detecting affinity-change peaks, but significantly higher power to capture shape-based changes. This simulation study also shows that TAN works very well in comparison with shape-based method MMDiff (Schweikert et al., 2013). We note that count-based method DESeq does not aim to detect peaks with diffuse signal or multiple local peaks, so it is expected that TAN will outperform this method at detecting differentially enriched peaks in this category.

Simulation 2

In our second set of comparisons, we performed a computational experiment with our actual ChIP-seq datasets. For each of TAN, MMDiff, and DESeq, we first performed differential enrichment analysis with sample sizes $n = 2, 3$, and 4, according to the settings recommended by the developers of the tools. We considered the $10,000$ best ranking peaks according to their p-values, and utilized the best ranking peaks identified by sample size of 4 as the gold standard differentially enriched peaks. We then report the proportions of ranked peaks that overlap with the gold standard ones when sample size is downscaled. The results are displayed in Figure 3.7A. It is clear that TAN has larger proportions recovered from the gold standard peaks compared to the other tools. Also, the power to detect differential regions improves with increased sample size for all three methods. This observation is similar to that of our simulation study. Interestingly, DESeq has large variability in the recovered proportions, especially in top-ranked peaks, whereas TAN and MMDiff has much lower variability. This suggests that the count-based approach DESeq is less robust for datasets with large variability among replicates (e.g., ChIP-seq data). Its performance deteriorated especially in the top-ranked peaks, whereas shape-based method MMDiff performed well in such scenario. This is anticipated since shape-changing peaks do not show much variability as compared to their affinity-changing counterparts. Our testing framework for TAN explicitly takes into account both the "within" and "between" variations of the

samples. This seems to be a key point when handling ChIP-seq data. As a result, its performance is more robust when there is large variation among replicates.

Ability to detect diffuse signals and multiple local peaks

Next, we examined the ability of TAN to identify diffuse signals and multiple local peaks while still maintaining the sensitivity of affinity-changing peaks. In this comparison, we also examine how TAN can universally identify differential regions with both changes in affinity and profile in our case study. We asked to what extent the sets of differentially enriched regions overlapped among tools. We performed a similar procedure of detecting differentially enriched peaks for a sample size of 4 as discussed in simulation 2. Peaks were ranked according to their p-values. We then computed the proportions of ranked peaks that overlap between methods. Figure 3.7B shows that TAN's overlap proportions with MMDiff and DESeq are higher than that of MMDiff with DESeq. This pattern mirrors the analyses from the previous section, and suggests that TAN is indeed able to detect both types of changes .

3.4 Case study: Differential enrichment analysis of H3K27me3 ChIP-seq in LCLs with conditionally active EBNA3C

Epstein-Barr virus (EBV) is a herpesvirus that establishes lifelong asymptomatic infection in up to 95% of the human population (Kieff and Rickinson, 2007). In vitro EBV infection of resting B lymphocytes drives them to proliferate as lymphoblastoid cell lines (LCLs) (Henle et al., 1967; Pope et al., 1968). EBV expresses limited genes, which include six nuclear proteins (EBNAs), three integral membrane proteins (LMPs), and more than 30 micro RNAs in LCLs (Kieff and Rickinson, 2007). EBNA3C, one of the six nuclear proteins, is transcription factor and regulates host cell genes (Ohashi et al., 2015). The growth effects of EBNA3C appear

to be primarily due to suppression of the CDKN2A gene products, p16INK4A and p14ARF (Maruo et al., 2011; Skalska et al., 2013). Conditional inactivation of EBNA3C results in increasing p16 expression and this is accompanied by the substantial reduction of the repressive H3K27me3 modification at the CDKN2A promoter and cell cycle arrest (Maruo et al., 2011). To further elucidate the relationship between EBNA3C and H3K27me3 modification, we profiled H3K27me3 by ChIP-seq in CDKN2A knockout EBNA3CHT LCLs in the presence (4HT+) or absence (4HT-) of 4-hydroxytamoxifen, with a total of 4 biological replicates per condition. We complemented this dataset with RNA-seq from the same system, with 8 replicates per condition. While we provide genome-wide differential ChIP-seq analysis results of these datasets, in what follows we focus our discussion on gene proximal regions to take advantage of the RNA-seq data.

Differential H3K27me3 modification and differential gene expression

We used MOSAiCS (MOdel-based one and two Sample Analysis and inference for ChIP-seq Data) (Kuan et al., 2011) (version `2.9.9`) at an FDR level of 0.05 to identify replicate-specific peaks. Union of the peak sets across replicates resulted in a total of 262,786 consensus peaks for further analysis. Overall, broad regions of low enrichment seem to be prevalent in this comparison in addition to some punctuated peaks with large fold changes are also present. Using an FDR threshold of 0.05, TAN identified 10,861 differential peaks while DESeq identified 3,253, with a total of 928 peaks identified as differential by both.

We next corroborated differentially modified peaks with changes in gene expression to both evaluate TAN's performance and also identify novel genes affected by conditional activation of EBNA3C. EBSeq (Leng et al., 2013) analysis of RNA-seq data from the 4HT+ and 4HT- conditions identified 362 differentially expressed (DE) genes at FDR of 0.05. We then considered all the peaks within the gene body of a given gene as peaks specific to that gene. 176 of the peaks identified as differentially modified by TAN mapped to 101 DE gene bodies. In comparison, only 55 DE

genes had at least one DEseq identified differentially modified peak. Figure 3.8A summarizes this comparison and highlights 39 genes in common. To the best of our knowledge, this is the first large scale differential enrichment of a repressive chromatin mark upon conditional activation of EBNA3C. Among the differentially modified genes detected by both methods, COBLL1 and CDKN2A have been shown to accumulate the repressive mark upon activation of EBNA3C (Maruo et al., 2011; Kalchschmidt et al., 2016), of which CDKN2A is uniquely detected by TAN.

ChIP-qPCR validation

We next performed ChIP-qPCR to validate a subset of the differentially modified peaks. Although Kalchschmidt et al. (2016); Maruo et al. (2011) have identified COBLL1 and CDKN2A as repressed with the activation of EBNA3C, we took an unbiased approach in selecting our ChIP-qPCR targets and considered both repressed and induced genes as a result of conditional activation of EBNA3C. We restricted our analysis to those with smallest TAN p-values to avoid ChIP-qPCR detection limits. We subsequently chose to validate 9 genes. These spanned genes with smallest p-values, and also larger p-values but high fold changes. Figures 3.8B and C display genes deemed differentially modified by TAN and DESeq with their respective differential ChIP-seq analysis p-values and fold change statistics. Data points annotated with gene names in these panels are the set of genes selected for the ChIP-qPCR experiments. In general, genes with smallest TAN p-values, such as CDKN2C, CDKN2A, and BZW2, correspond to largest absolute fold changes (Figure 3.8B). In contrast, DESeq target genes do not show this trend, and genes with small p-values do not correspond to large fold changes (Figure 3.8C). Further, to avoid selection biases towards large fold changes, we also performed ChIP-qPCR for genes TTYH2, NCALD, and COBLL1 with small fold changes and small p-values.

The H3K27me3 ChIP-qPCR results are presented in Figure 3.9. We see significant changes in the qPCR signals associated with the selected EBNA3C genes. Once again, we compared the results of TAN and DESeq among the differentially modified genes. In Figure 3.8C, annotated genes were selected by DEseq based on

ranking of p-values and fold changes statistics. It is apparent that, due to large fold change, majority of these genes are also identified by TAN, showing a good agreement with DESeq. In contrast, genes like CLEC2D and TTYH2 identified by TAN and not by DESeq exhibit broad regions of low enrichment with small fold changes, further showing the complementary nature of our method to count-based methods. To our surprise, despite a large fold change, CDKN2A was detected by TAN but not DESeq. Note that we have emphasized the importance of CDKN2A in this analysis, as it has been shown to accumulate the repressive mark upon activation of EBNA3C (Maruo et al., 2011). We further include coverage plots of H3K27me3 signals to support our findings in Figure 3.10. In general, regions with small TAN p-values correspond to large fold changes of ChIP peaks. Figure 3.10A displays the normalized condition-specific average coverage plots for 4 genes with the smallest p-values. These genes show visibly strong differential H3K27me3 enrichment levels. In addition, we also observed that some genes with small fold changes exhibited small TAN p-values (Figure 3.10B). Gene such as NCALD showed a clear shift of peaks between two conditions and highlighted the effect which was not captured by simply using total read counts as in fold changes and DESeq statistics. This eludes to the importance of modeling spatial ChIP enrichment profiles. For completeness, additional plots of H3K27me3 signals for genes ANKMY2 and PTAFR are provided in Appendix B: Supplementary Fig.B.9.

3.5 Conclusions

Many different tools to identify differential ChIP enrichment between experimental conditions have been developed in recent years. However, these tools vastly differ in terms of usability and in range of applicability, reflecting the complex structure of ChIP-seq data. As a result, the choice of method impacts the quantity and characteristics of the identified differential regions. To overcome these limitations, we proposed a nonparametric method, TAN, to identify differentially enriched regions based on the ChIP-seq data. Different from all the currently available methods, TAN models the spatial histone enrichment profiles, rather than simply

considering the total read counts in a given region. Together with its ANOVA-based testing setting suitable for multiple conditions, TAN is widely applicable to diverse types of ChIP-seq data, avoiding pitfalls of making an improper choice of tool.

Software

TAN is implemented as an R package and is available at `https://github.com/duydnguyen/tan`.

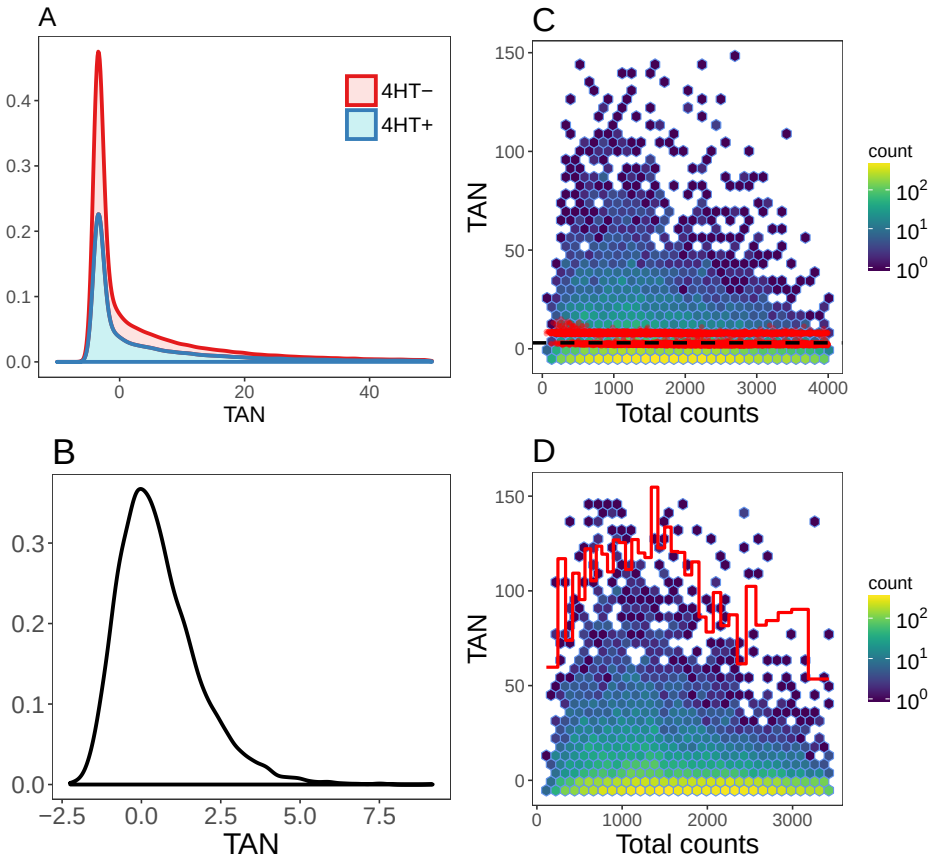


Figure 3.3: Different null distributions (left panels) and TAN as functions of read counts (right panels). (A) Distributions of Neyman statistics from 4HT- and 4HT+ when testing between pairs of biological replicates for a set of peaks. (B) The asymptotic distribution of the Neyman statistics under the null hypothesis of no DE. (C) TAN as a function of total counts where each dot corresponds to Neyman statistics when testing between pairs of biological replicates. Each red dot denotes the FDR cutoff at 5% obtained from the permutation null distribution. The black line represents the cutoff for FDR of 5% from the asymptotic distribution. (D) TAN with pooled variance as function of total counts. The red lines of FDR at 5% are obtained by our framework.

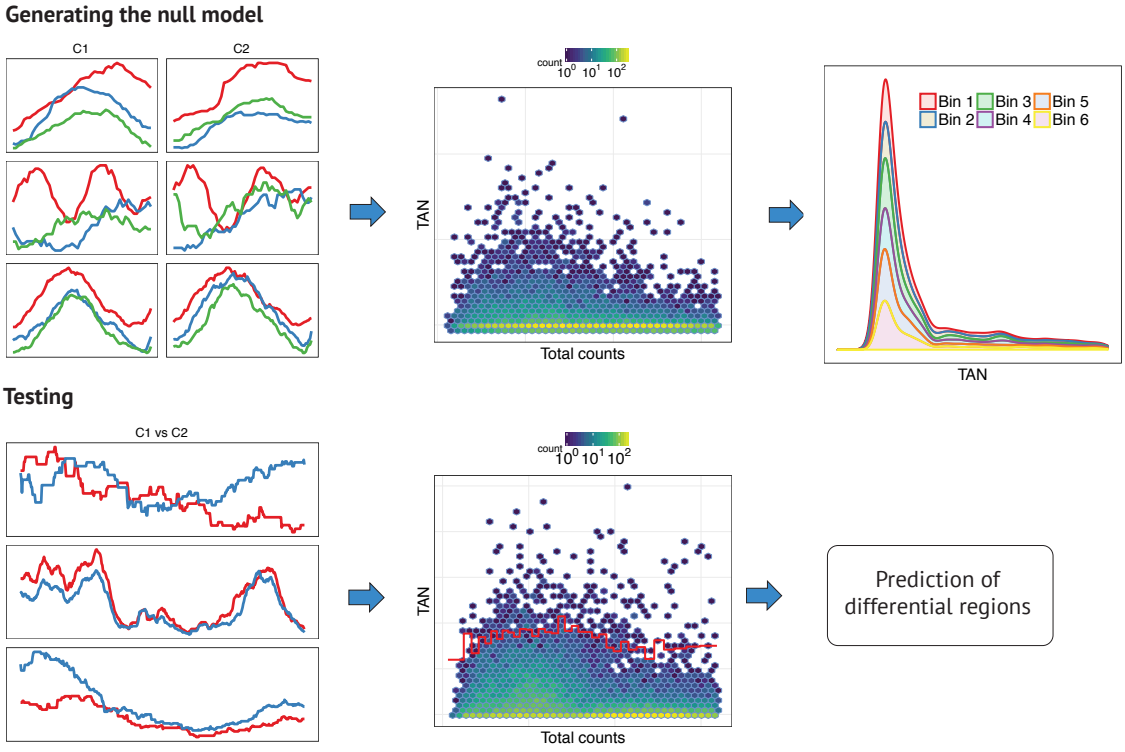


Figure 3.4: A workflow of TAN's framework. (Top) Generating null distributions. (Bottom) Testing phase

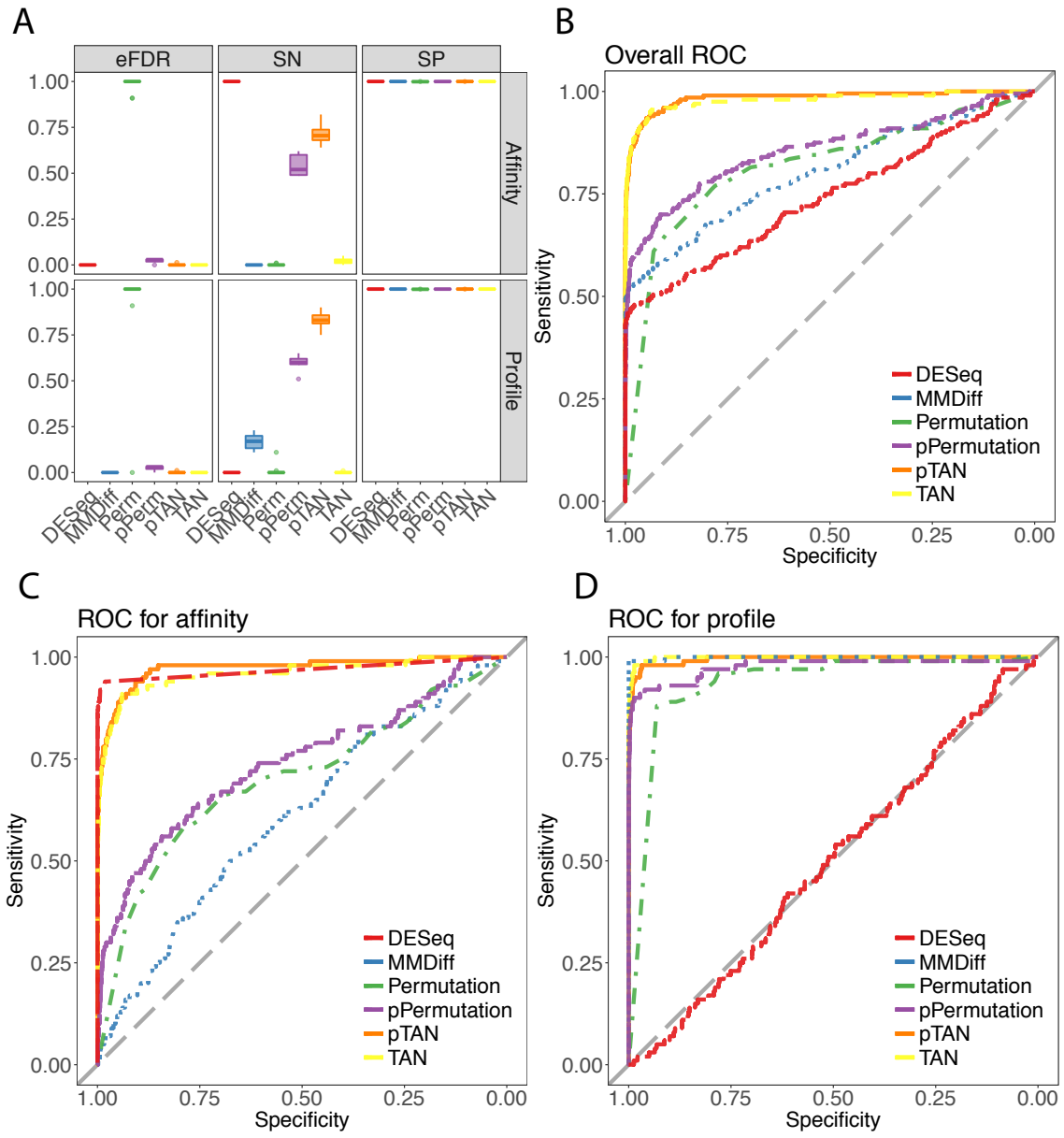


Figure 3.5: (A) Performance summary on 10 simulation replications at FDR level of 0.05. (B-D) ROC curves for various methods under affinity and profile changes. Average power to detect simulated DE regions. Averages are calculated over 10 runs of simulated data sets. TP: true positives, FP: false negatives, TN: true negatives, eFDR: empirical FDR, SN: sensitivity, SP: specificity, auROC: area under the ROC curve.

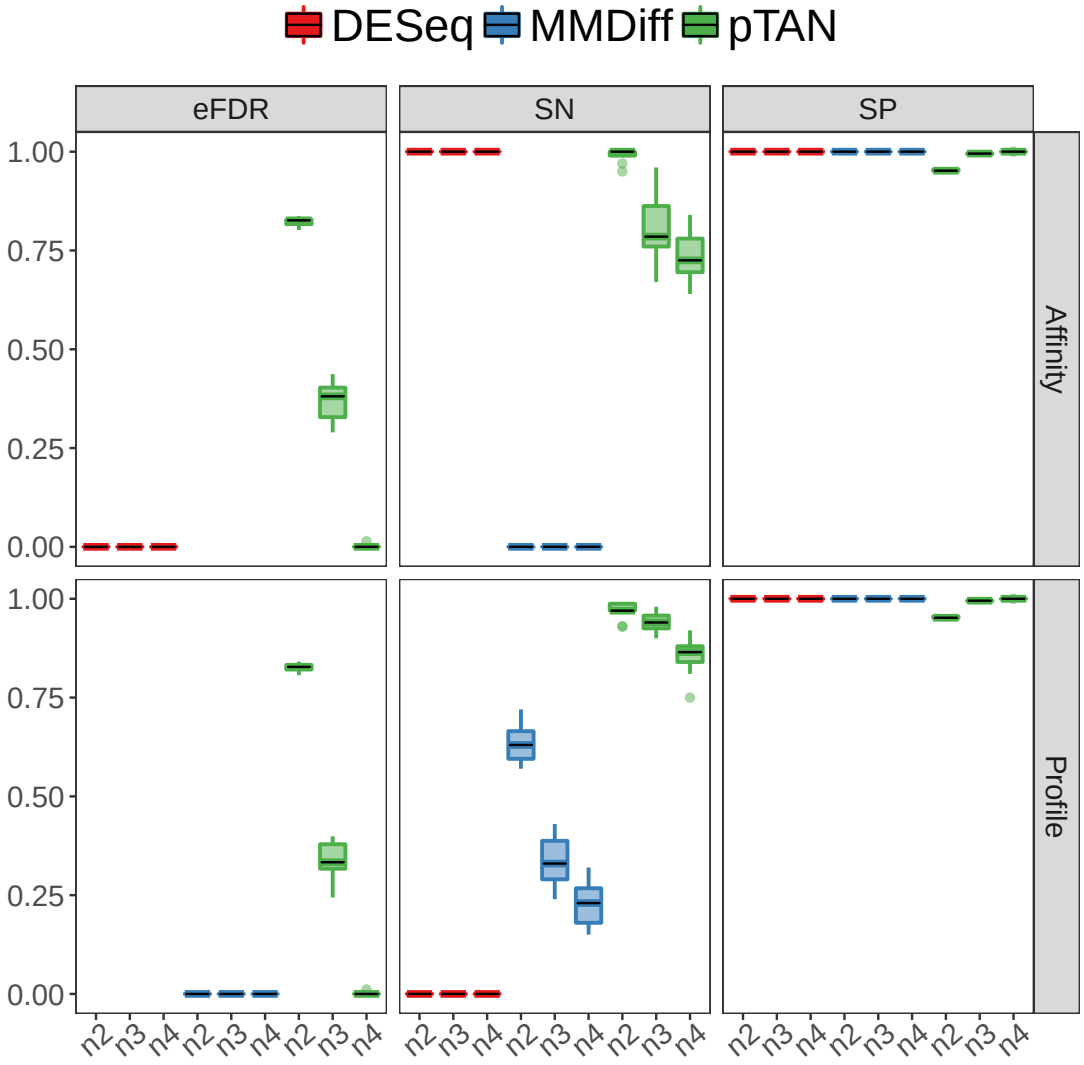


Figure 3.6: Comparative results of methods as a function of the number of replicates per condition. FDR threshold: 0.05; TP: true positives, FP: false negatives, true: TN negatives, eFDR: empirical FDR, SN: sensitivity, SP: specificity, auROC: area under ROC.

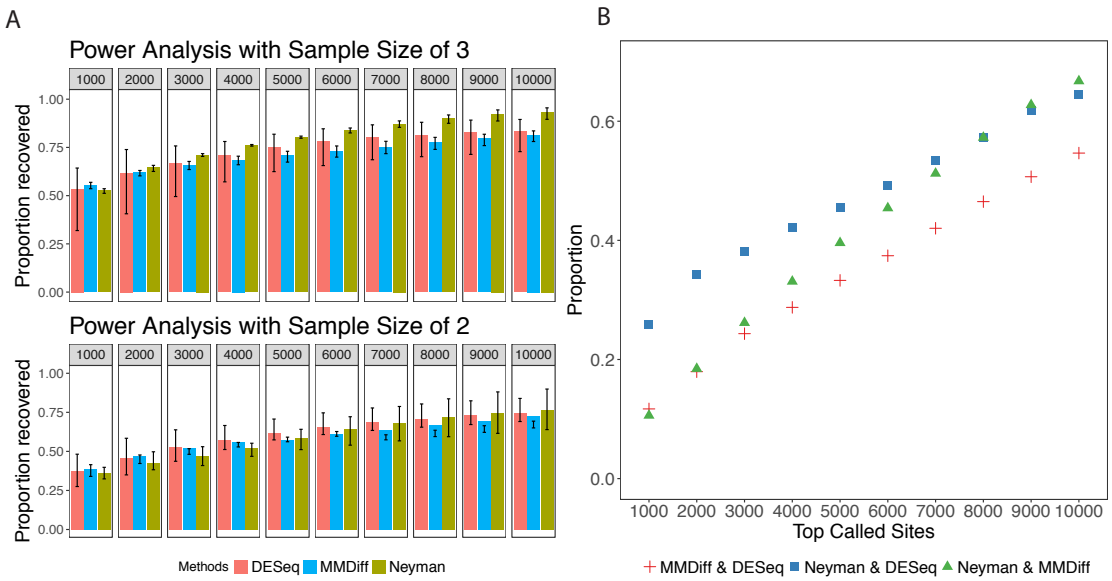


Figure 3.7: (A) Within method consistency in identifying differential enriched peaks across varying sample sizes. (B) Reproducibility of differential enrichment detection across methods.

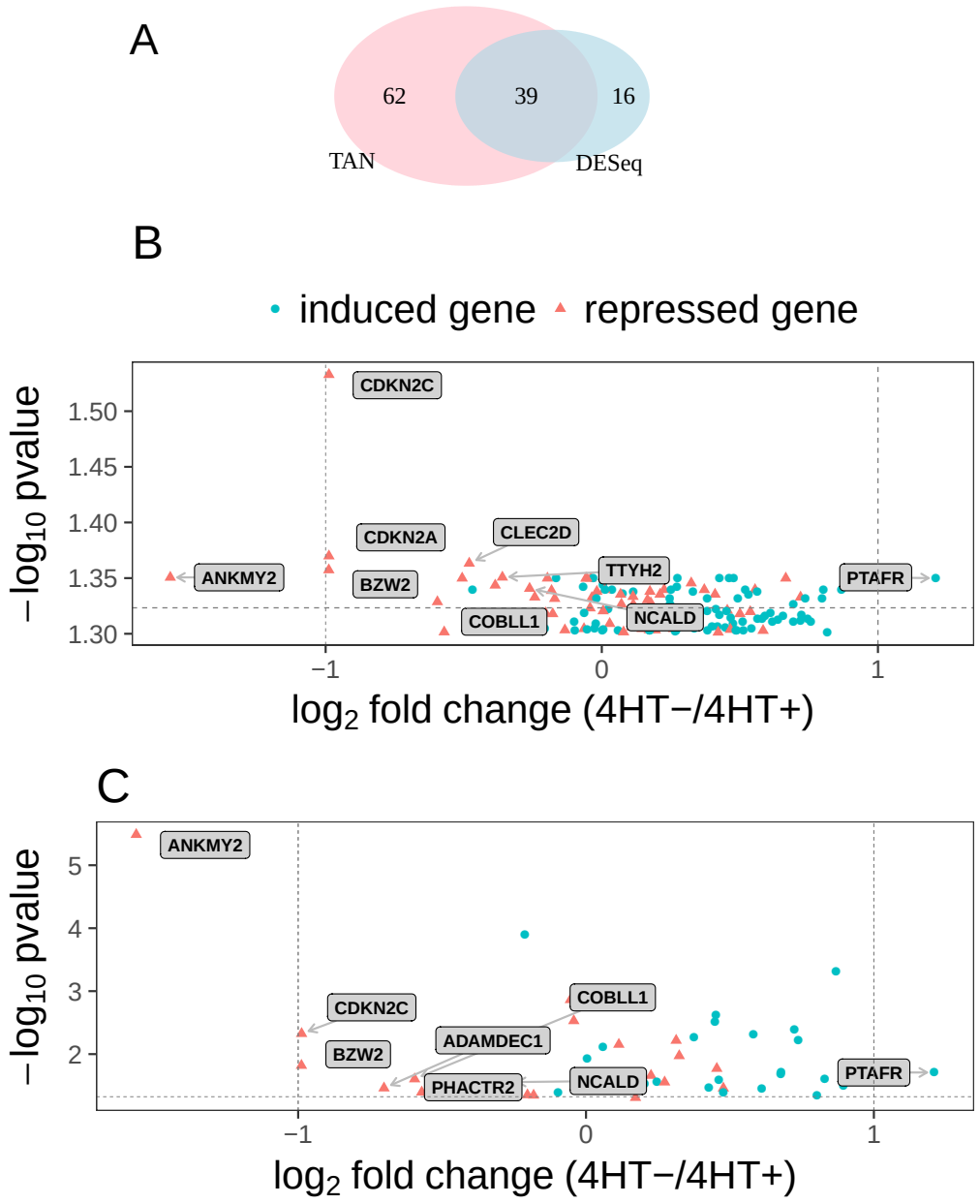


Figure 3.8: (A) Number of differentially expressed genes with TAN and/or DESeq identified differentially modified peaks. (B-C) Genes selected by TAN and DESeq using rankings of their p-values and fold change statistics. The cutoff for p-values is depicted by horizontal lines and the vertical lines mark log fold-change values of $\pm$ 1. Annotated genes in (B) are those selected for ChIP-qPCR validation. Annotated genes in (C) are those selected by DEseq based on ranking of p-values and fold changes and are among the ChIP-qPCR validation set.

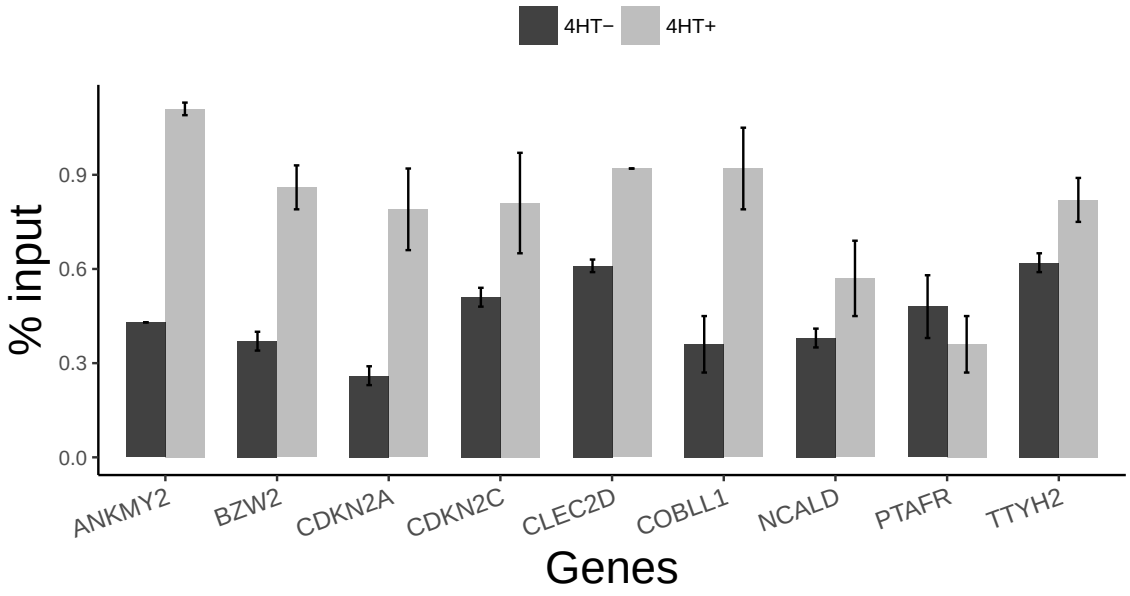


Figure 3.9: H3K27me ChIP-qPCR results for EBNA3C genes.

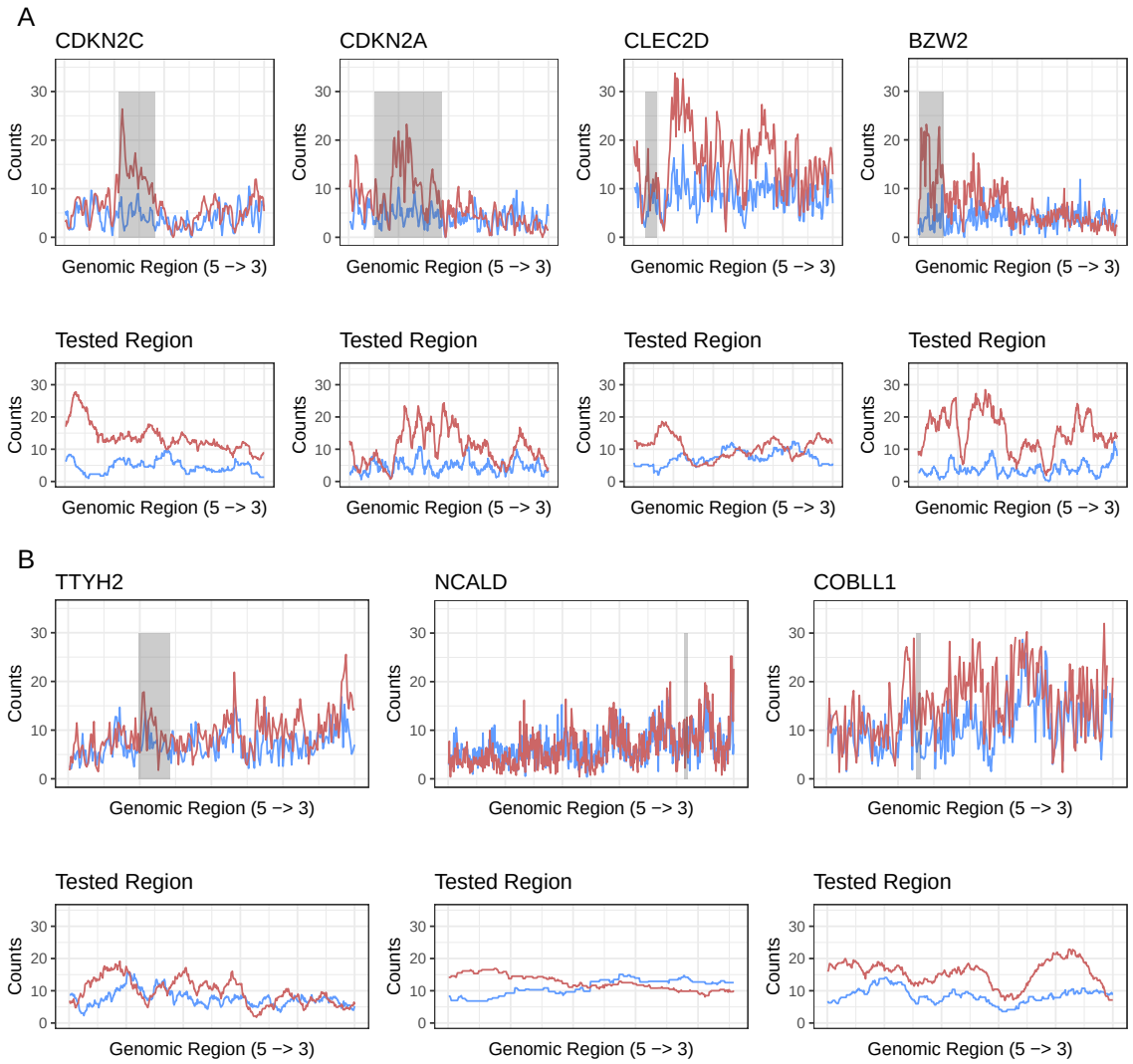


Figure 3.10: Average H3K27me3 signals over gene bodies for 4HT-(blue) and 4HT+(red) for (A) top 4 genes ranked by p-values, (B) 3 genes with small p-values but small fold changes. Shaded regions mark the tested regions whose profile plots are further illustrated below their corresponding gene body plots.

4 SOFTWARE FOR HI-C AND CHIP-SEQ DIFFERENTIAL ANALYSIS

4.1 TreeHiC: Hierarchical testing for differential chromatin interaction analysis

Introduction

This section contains instructions on how to use *TreeHiC* for differential interaction analyses of Hi-C data. Improvements to the chromatin conformation analysis technology enable us to explore the 3D structure of the chromatin at a finer scale. With more data available in the high-resolution Hi-C protocols, one would like to perform comparative analysis between cell types, and experimental conditions.

Here, we present a novel approach, TreeHiC, for rigorous detection differential interactions. TreeHiC replies on hierarchical multiple testing framework introduced by Yekutieli (2008) and is the first of its kind applied to Hi-C data. Our computational experiments and simulations illustrate that TreeHiC is powered for detecting changes while robustly controlling the FDR under a wide range of settings and resolutions. It also is considerably more powerful than existing methods, especially in sparse testing problems where number of hypotheses could be millions with a weak signal-to-noise ratio, giving it a clear advantage over existing differential interaction approaches. Additionally, while the current version of TreeHiC implements methodology pertaining to Hi-C differential analysis, it is easily extendable for other similar data such as ChIA-PET and HiChIP.

Installation

Before installation, the following R package dependencies should be installed:

- data.table
- readr

- Matrix
- HiTC
- doParallel

In addition, for the partition and extremum search phase in our framework, *TreeHiC* requires the Topology ToolKit TTK (Tierny et al., 2017) for persistence-driven segmentation and analysis tasks. User could follow the following link `https://topology-tool-kit.github.io/installation.html` for a detailed instructions for the installation of TTK.

Then run the following code in R to install *TreeHiC* from github.

```
devtools::install_github("duydnguyen/TreeHiC",ref ="devel")
```

Quick start

A typical differential analysis of Hi-C data is described below. For simplicity, assume that we have already had inputs as a set of Hi-C 2D contact matrices for each conditions (or cell lines). The discussion of how to obtain these processed of Hi-C contacts is postponed to section 4.1. Note that *TreeHiC* takes input matrices with two different types: (1) the (usual) *matrix* object in R, and (2) the *dgCMatrix* object in R package *Matrix*. The sparse *dgCMatrix* matrix input is required to perform differential interaction detection for higher-resolution Hi-C studies (e.g., less than 20K). The code itself is split across several steps:

1. Load data and create *treeHiCDataSet* object

```
library(TreeHiC)
hicDb <- new("treeHiCDataSet")
hicDb <- TreeHiC::HiCDataSetFromMatrix(hicDb,
          contactMatrixList = matInputs,
          colData = coldata, path = path)
```

2. Evaluate height function and required file inputs for partition and extremum search

```
hicDb <- TreeHiC::evalDiffMat(hicDb)
write.csv(hicDb@d_height[,'f'], file =
      paste0(hicDb@path,"temp/heights-scalar.csv"),
      row.names = FALSE, quote=FALSE)
```

3. Create persistence graph and persistence grid

```
  TreeHiC::get_persistence_curve(path = hicDb@path)
  hicDb <- selectPLevelGrid(hicDb)
```

4. Perform partition and extremum search

```
TreeHiC::get_partitions(path = hicDb@path,
          pLevelGrid = hicDb@pLevelGrid[["pLevelGrid"]])
```

5. Test for significant differences between groups

   a. Evaluate p-values (e.g., permutation tests or *diffHiC*)

   b. Perform differential interaction analysis

```
  hicDb <- TreeHiC::hic_diff(hicDb,
          mat_pvals = mat_pvals, alpha = 0.05)
```

Loading genomewide chromatic interactions

The *.hic* format is one of the widely adopted chromatin interaction storage format from Aiden's group (`http://www.aidenlab.org/data.html`). For the current format, *TreeHiC* supports matrix inputs. Therefore, in this section, we explain how to extract and make available such inputs for our pipeline. The first option is to follow Aiden's github repository for their data extraction steps from *.hic* format.

For the second option, we explain the same pipeline proposed by R package FIND (Djekidel et al., 2018).

Below is an example showing how to deal with such data. Here, we use K562 and GM12878 cell lines. For illustration purpose, we extract Hi-C contact matrices from chromosome 22.

```
library(FIND)
require(Matrix)
require(HiTC)

## We need to know the chromosomes lengths
require(BSgenome.Hsapiens.UCSC.hg19)
seqlen = seqlengths(BSgenome.Hsapiens.UCSC.hg19)[1:22]

## chromosome to perform DA
chr <- 22

## normalization method:
## accepted values are : "NONE","VC","VC_SQRT" or "KR"
normMethod <- "VC_SQRT"
resolution <- 50*10^3

#### * load all HiC matrices
## We load two K562 replicates and GM12878
hic_mats = list(K562_reps1 =
 paste0(path_to_hic,"K562/HIC071_30.hic"),
 K562_reps2 = paste0(path_to_hic,"K562/HIC074_30.hic"),
 GM12878_reps1 = paste0(path_to_hic,"GM12878/HIC025_30.hic"),
 GM12878_reps2 = paste0(path_to_hic,"GM12878/HIC026_30.hic"))


## We use this function as a proxy to call
```

```
fct = function(chrom, seqlen, hicfile, resolution,
            normMethod = "VC_SQRT") {
   intdata = readJuiceBoxFormat(hicfile, chromosome = chrom,
              normMethod = normMethod,
              resolution = resolution)
    smry <- summary(intdata)
    bins <- tileGenome(seqlen[chrom], tilewidth = resolution,
         cut.last.tile.in.chrom = TRUE)
    bins$name <- 1:length(bins)
    # correct size
    intdata <- sparseMatrix(i = smry$i, j = smry$j,
               x = smry$x,
               dims = c(length(bins), length(bins)))
    HTCexp(intdata, bins, bins)
}

## Read the list of .hic matrices
hic_matrices = list()
for (hic in names(hic_mats)) {
    print(hic)
    # here we are using chromosom 22, just to test
    chroms = names(seqlen)[chr]
    hic_lst = mclapply(chroms, fct,
              hicfile = hic_mats[[hic]],
              resolution = resolution, seqlen = seqlen,
              normMethod = normMethod, mc.cores = mc_cores)
    hic_matrices[[hic]] = HTClist(hic_lst)
}

## save_hic_mats
save(hic_matrices, file = paste0("hic_matrices_chr",
```

```
    chr,"_res", resolution/10^3,"K.RData"))
```

An example of TreeHiC pipeline

Here, we elaborate the steps in section 4.1. As before, we utilize K562 and GM12878 cell lines, and detect differential interactions on chromosome 22. In addition, we use two biological replicates for each cell lines.

load data and create treeHiCDataSet object

```
library(readr)
library(data.table)
library(TreeHiC)

#### * parameters for our analysis
## number of reps in each conditions
nReps <- 2
alpha <- 0.05
## working path to save results
path <- paste0("DeriveData/chr", chr,"/")

system(paste0('mkdir ', path, 'temp'))

hic_matricesDb <- list('c1List' =
  list(hic_matrices[[1]][[1]]@intdata,
  hic_matrices[[2]][[1]]@intdata ),
  'c2List' = list(hic_matrices[[3]][[1]]@intdata,
  hic_matrices[[4]][[1]]@intdata ))
```

Next, we create *colData* and matrix inputs *matInputs*. This step is similar to that of DESeq (Love et al., 2014) in differential gene expression.

```
sample.ids <- c('mat1', 'mat2')
txt_files <- rep(1:2)
condition <- factor(c(rep(c('K562', 'GM12878'), 1 )))
coldata <- data.frame('SampleID' = sample.ids,
          'Condition' = condition, 'files' = txt_files)
matInputs <- list()

m1 <- hic_matricesDb[['c1List']][[1]]
m2 <- hic_matricesDb[['c2List']][[1]]
for (i in 2:length(hic_matricesDb[['c1List']])) {
  m1 <- m1 + hic_matricesDb[['c1List']][[i]]
}
for (i in 2:length(hic_matricesDb[['c2List']])) {
  m2 <- m2 + hic_matricesDb[['c2List']][[i]]
}
m1 <- m1 / length(hic_matricesDb[['c1List']])
m2 <- m2 / length(hic_matricesDb[['c2List']])
matInputs <- list('mat1' = m1, 'mat2' = m2)
```

We now initialize *treeHiCDataSet* object

```
hicDb <- new("treeHiCDataSet")
hicDb <- TreeHiC::HiCDataSetFromMatrix(hicDb,
    contactMatrixList = matInputs,
    colData = coldata, path = path)
```

Evaluate height function

```
hicDb <- TreeHiC::evalDiffMat(hicDb)
write.csv(hicDb@d_height[,'f'], file =
      paste0(hicDb@path,"temp/heights-scalar.csv"),
```

```
    row.names = FALSE, quote=FALSE)
```

Create persistence graph and persistence grid

```
TreeHiC::get_persistence_curve(path = hicDb@path)
hicDb <- selectPLevelGrid(hicDb)
```

Create persistence graph and persistence grid

```
  TreeHiC::get_persistence_curve(path = hicDb@path)
  hicDb <- selectPLevelGrid(hicDb)
```

For visualization purpose, we also provide a function to plot persistence graph as follows:

```
TreeHiC::plot_persistence_curve(hicDb@pLevelGrid,
  path = hicDb@path)
```

Perform partition and extremum search

```
TreeHiC::get_partitions(path = hicDb@path,
pLevelGrid = hicDb@pLevelGrid[["pLevelGrid"]])
```

Test for significant differences between groups

So far, we obtained the *hicDb* object which stores all information required for this last step. As mentioned in the main text, *TreeHiC* is scalable as its procedure is readily accompanied with p-value generations from different models. Specifically, we welcome a square matrix of p-values from two methods: permutation test (Stansfield and Dozmorov, 2017), and diffHiC (Lun and Smyth, 2015). In this example, we show the code to obtain the permutation p-values.

```
mat_pvals <- TreeHiC::eval_perm_pvals(
              mat1 = as.matrix(hicDb@contactMatrixList[[1]]),
              mat2 = as.matrix(hicDb@contactMatrixList[[2]]),
              excluded = hicDb@excluded)
```

Lastly, we arrive at the testing step

```
hicDb <- TreeHiC::hic_diff(hicDb,
        mat_pvals = mat_pvals, alpha = 0.05)
```

For the final list of result which include the called interactions and their corresponding p-values, we use the following code

```
hic_diff_result <- data.frame(hicDb@hic_diff_result)
```

4.2 tan: A differential analysis pipeline for ChIP-seq data

Introduction to the tan package

We present a brief overview of the *tan* package (`https://github.com/duydnguyen/tan`). This package provides a framework for identifying differentially enriched (DE) regions from ChIP-seq data. In this vignette, we utilized the data from a ChIP-seq experiment investigating H3K27me3 to illustrate our pipeline. To load the required packages, we use:

```
library(tan)
library(tanExample)
```

For a typical workflow, *tan* takes a set of genomic regions (or peaks) and their aligned reads from ChIP-seq experements. Its goal is to predict the DE regions between two or multiple conditions. Specifically, the pipeline performs the following steps:

Inputs

A set of pre-define regions for DE pipeline

For its first input, *tan* takes a set of genemic regions in BED format as candidates to perform differential analysis. If such candidate regions are unavailable, we suggest using mosaics (`http://bioconductor.org/packages/release/bioc/html/mosaics.html`) to obtain these regions.

Read coverage data

After a set of candidate regions is available, *tan* also takes read coverages as its additional inputs. There are several ways to load count data from bam files and convert them into counts (e.g. *bamsignals* or *Segvis*). Here, illustrate a way of obtain these coverage data via tan. A typical workflow of obtaining reads requires to firstly

load all reads in R, secondly process them and lastly convert them into counts. *tan* package was efficiently implemented to merge these steps into one single function *bamCoverage()*.

Loading toy data

tan accepts bam file to generate read coverages. First, we load the required packages (which are all required for installing *tan*).

```
library(GenomicRanges)
library(GenomicAlignments)
library(data.table)
```

For demonstration, we used the H3K27me3's region of BZW2's gene body whose genomic coordinates are stored in BZW2.bed. In the following, we will use sorted and index bam files to load reads. The bam files need to be sorted and indexed. Note that tan require the bam index to be named like bam file with ".bai" suffix.

```
files = list.files(system.file("extdata/bzw2",
        package = "tanExample"),full.names = TRUE)
basename(files[c(1,3,6,9,12)])
## [1] "BZW2.bed"                  "BZW2_rep1_minus_sorted.bam"
## [3] "BZW2_rep1_plus_sorted.bam" "BZW2_rep2_minus_sorted.bam"
## [5] "BZW2_rep2_plus_sorted.bam"
bam_files <- files[c(3,6, 9, 12)]
bed_files <- files[1]
# checking if there is an index
file.exists(gsub(".bam$", ".bam.bai", files[c(1,3,6,9,12)]))
## [1] TRUE TRUE TRUE TRUE TRUE
```

We then set up the following parameters

```
binsize <- 150
# smooth parameter
sm <- 1
```

```
mc_cores <- 3
bed_content <- read.table(file = files[1],
   stringsAsFactors = FALSE)
gr <- GRanges(seqnames = bed_content[, 1],
ranges = IRanges(bed_content[,2],
        bed_content[,3]),
        strand = "*")
chromosomes <- c("chr7")
gr
## GRanges object with 3 ranges and 0 metadata columns:
##       seqnames               ranges strand
##          <Rle>            <IRanges>  <Rle>
##   [1]     chr7 [16685756, 16746148]      *
##   [2]     chr7 [16685756, 16690756]      *
##   [3]     chr7 [16690756, 16695756]      *
##   -------
```

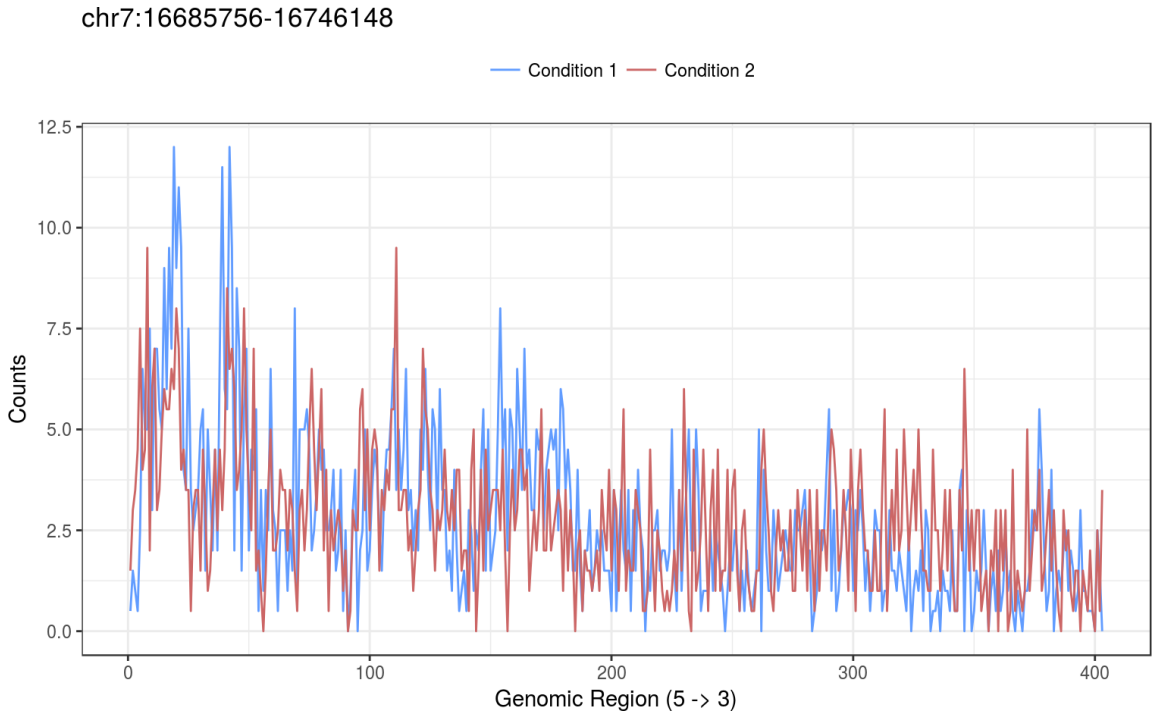
The *binsize* parameter indicated how reads are counted. A value of 1 correponds to single base pairs. Very often it is better to count reads mapping to bins. Bins are small partitions of fixed size tiling a larger region. *bamCoverage()* introduces the *binsize* option to implement this.

Next, let's count how many reads map to the regions given in the bed file. Using the *bamCoverage()*, this is straightforward.

```
  coverage <- tan::bamCoverage(bam_files = bam_files,
              bed_files = bed_files, mc_cores = mc_cores,
              sm = sm, binsize = binsize)
class(coverage)
## [1] "list"
```

bamCoverage() returns a coverage list whose length equals to the number of regions in the given bed file. Each element of coverage list is a matrix storing values of reads. Let's summarize this with a plot

```
tan::plotCoverage(coverage[[1]], title = toString(gr[1]))
```



Differential analysis

The *tan*'s differential analysis workflow consists of two main phases:

1. Generating empirical null distributions,
2. Testing phase and predicting differential regions.

Here, we present an example of 8000 genomic regions sampled from our H3K27me3 data. We included the following inputs: (1) a set of genomic regions, and (2) their corresponding read coverages

```
files = list.files(system.file("extdata",
        package = "tanExample"),full.names = TRUE)
```

```
basename(files[2:3])
## [1] "coverage_vignette.RData" "gr_sitesSelect.RData"
load(files[3])
gr_sitesSelect
## GRanges object with 8000 ranges and 0 metadata columns:
##          seqnames                 ranges strand
##             <Rle>              <IRanges>  <Rle>
##      [1]     chr5 [113552335, 113552599]      *
##      [2]     chr5 [135821134, 135822027]      *
##      [3]     chr5 [155717398, 155717666]      *
##      [4]     chr6 [ 34889368,  34889621]      *
##      [5]     chr6 [ 47486289,  47486568]      *
##      ...      ...                   ...    ...
##   [7996]     chr4 [ 45632355,  45632606]      *
##   [7997]    chr19 [ 57437518,  57437799]      *
##   [7998]     chr1 [244590200, 244590597]      *
##   [7999]     chr4 [  4418526,   4418799]      *
##   [8000]    chr17 [ 26869200,  26869520]      *
##   -------
```

Creating a tanDb object

We first load the coverage, and construct a **tanDb** object as follows.

```
load(files[2])
tanDb <- new("tanDb", coverage = coverage)
```

Sampling design

The *tan* power lies in accomodating regions (or peaks) with different lengths by using the adaptive Neyman statistics. Its "adaptive" property not only allows a wide range of peak lengths, but also improve the power of detecting DE regions.

However, for cases of regions with large genomic intervals, we advise using a sampling design across these intervals. This procedure mimics a Latin hypercube sampling in computer experiments or for Monte-Carlo integration. Such designs have a space-filling property which helps capture the spatial and shape structures of histone profile coverage. As a result, it improves the power of detecting DE regions. For instance, using the *createDesigns()*, we could create a grid of evenly spaced points for each region.

```
tanDb <- createDesigns(tanDb, s.size = 500, LHD = TRUE,
          Uniform = FALSE )
```

Empirical null distributions

In phase 1, *tan* evaluates the (within) Neyman statistics between pairs of replicates within the same conditions. Then, it clusters these peaks into bins based on their total counts. These bins are simply partitions of read counts across the pre-defined genomic regions. After that, *tan* constructs the corresponding null distributions based on the statistics available from each bin (or partition) of counts.

To obtain this goal, we first generate total counts (or area under coverage profiles). This step is required for peak clustering in later steps. We proceed as follows:

```
tanDb <- calculateTotalCounts(tanDb, nSamples = 3,
          bNormWidth = FALSE,
          bSampleMean = FALSE)
head(tanDb@Ns)
##       ab       ac       bc       AB        AC       BC
## [1,] 2194.469 1953.895 1666.230 3338.210 1847.0019 3403.150
## [2,] 4416.688 2780.517 2845.768 2370.757  667.7475 3038.504
## [3,] 3637.914 4082.268 2264.053 5021.198 2836.7046 5729.104
## [4,] 1583.548 1134.583 1206.026 6644.338 5718.9592 2217.909
## [5,] 4346.282 3389.183 3600.858 4141.331 3229.9142 3761.427
## [6,] 2821.856 3104.555 3339.286 6400.928 6340.7494 1988.829
```

Next, we cluster peaks into bins based on their total counts obtained by the previous step.

```
# quantile vector for binning
quantprobs <- seq(0, 1, 0.05)
tanDb <- calculateWithinSites(tanDb, quantprobs = quantprobs)
```

We then calculate the variance for each grid point in our sampling design obtained from previous step. Based on the clustering, we also pool variance for sites within the same clusters.

```
Global_lower <- 100
## pooled quantile at each genomic position:
poolQuant <- 0.5
# number of points for moving average
movAve <- 20
tanDb <- calculateVariance(tanDb, minus_condition = TRUE,
                           Global_lower = Global_lower,
                           poolQuant = poolQuant,
                           movAve = movAve )
tanDb <- calculateVariance(tanDb, minus_condition = FALSE,
                           Global_lower = Global_lower,
                           poolQuant = poolQuant,
                           movAve = movAve )
```

Finally, (within) adaptive Neyman statistics for are obtained as follows:

```
tanDb <- generateWithinTan(tanDb, minus_condition = TRUE)
tanDb <- generateWithinTan(tanDb, minus_condition = FALSE)
```

This completes the phase 1 of our pipeline.

Testing phase

For testing phase, *tan* constructs the (between) Neyman statistics by testing replicates from different conditions. It then maps the regions being tested to their corresponding empirical null distributions based on regionsâŁ™ total read counts. Consequently, TAN provides the p-values for each testing regions, leading to the prediction of differential regions after multiple testing correction. The code is as follows:

```
tanDb <- computePvalues(tanDb, quant = 0.5,
    poolQuant = poolQuant,
    movAve = movAve, Global_lower = Global_lower,
    ignore_sitesUnused = TRUE, na_impute = FALSE)
```

The raw and adjusted p-values could be obtained by extracting slot *tanDb@PvalList*. This completes the testing phase.

Furthermore, the quantile parameter quant for combined p-values could be adjusted without rerunning *computePvalues*. The following function performs this task:

```
pvals <- evalPvals(P = tanDb@PvalList,
          total = nrow(P['pval']),
          quant = 0.25, nSamples = tanDb@nSamples,
          BH = FALSE, na.rm = TRUE)
pvals.a <- p.adjust(pvals, method = 'BH')
```

The adjusted p-values could be obtained directly by changing the parameter BH = FALSE to BH = TRUE.

```
pvals.a <- evalPvals(P = tanDb@PvalList,
            total = nrow(P['pval']),
            quant = 0.25, nSamples = tanDb@nSamples,
            BH = TRUE, na.rm = TRUE)
```

For visualizing the results of diffentially/non-differentially enriched sites declared by our method, we could use the *plotCoverage* function as described in previous sections.

```
de_sites <- which(pvals.a <= 0.05)
tan::plotCoverage(tanDb@coverage[[de_sites[1]]],
      title = "coverage_plot_of_DE_site")
```

5 CONCLUSIONS

In this thesis, I present two novel computation methods for large-scale inference in two important techniques in high-throughput sequencing technologies. The first one is chromatin conformation capture with high-throughput sequencing (Hi-C), and chromatin immunoprecipitation coupled with high-throughput next generation sequencing (ChIP-seq). Hi-C data provides key insights into the 3D structures of the human genome, while ChIP-seq has been successfully used for genome-wide profiling of transcription factor binding sites, histone modifications, and nucleosome occupancy in many organisms and humans.

In Chapter 2, I present the *TreeHiC* framework to detect differential interactions in Hi-C data. Prior to this work, very few tools to identify differential interactions between experimental conditions from Hi-C data have been developed in recent years. In general, these tools vastly differ in term of usability and in range of applicability, falling short of fully exploiting the power of Hi-C data. Such shortcomings become apparent as none of the methods formally address the rate of false discovery and their performance for reported findings, especially in high-resolution Hi-C studies which involved in testing a large number of hypotheses. To overcome these limitations, we proposed TreeHiC, a hierarchical testing procedure for quantitative comparison applied to Hi-C data. Different from all the currently available methods, TreeHiC formally addresses and resolves three critical issues in large-scale Hi-C studies: (i) the existence of sparsity and weak signal-to-noise ratios in high-resolution Hi-C differential analysis, (ii) the FDR control and performance for reported findings, and (iii) lack of biological experiments. Additionally, TreeHiC is practically scalable, as its procedure is readily accompanied with different models. Lastly, while the current version of TreeHiC implements methodology pertaining to Hi-C differential analysis, it is easily extendable for other similar data such as ChIA-PET and HiChIP.

Here, we summarize our contributions to Hi-C differential interaction analysis, and additional remarks that have not been mentioned in the main text. To start with,

as for any research problem, it is not a trivial task to establish objectives that existing methods have not been addressed. This is especially the case in this work since Hi-C differential analysis is not a well-studied problem due to the relative new Hi-C technology. In addition, when putting what we want to achieve in our new method, we would like to ensure that the proposed objectives are practical and realistic, not just solving "small" or theoretically interesting cases. Furthermore, Hi-C data has several limitations, making it very challenging to utilize complex statistical models. Among these limitations, having no biological replicate experiments is one of the most drawbacks. In addition, with the increasing availability in high-resolution Hi-C data, the proposed method must accommodate and perform the analysis in a reasonable amount of time. For instance, with a resolution of 5Kbp, Hi-C contact map between chromosome 1 with itself would produce a contact matrix of dimension approximately $50K \times 50K$. With these limitations in mind, we propose the following objectives for our new tool, *TreeHiC*. The method aims to

- Work well with a wide range of HiC data resolutions/bin sizes.
- Work well in the case of sample size $n = 1$ (e.g., biological replicates).
- Is practically scalable by avoid constraining within a rigid statistical model.
- Attain more powerful than existing methods, and control the False Discovery Rate (FDR).

As discussed in details with simulation and experimental studies (Chapter 2, Section 2.3), we showed that our method could potentially resolve these proposed objectives, making it an important choice for Hi-C differential analysis. Here, we especially emphasize our contribution in the case of sample size $n = 1$, as it has never been addressed and resolved in previously published works.

Now, we further discuss other important contributions of *TreeHiC* that have not been addressed directly in the main text. First, we did not invent the partition and extremum search algorithm (Morse-Smale Complex, Appendix A.1), nor the hierarchical testing procedure (Yekutieli, 2008). They are two separate works in their own

right. Note that Morse-Smale complex is one of the important branches/methods in the field of Topological Data Analysis (TDA). Though TDA has been widely applied in other fields such as machine learning and engineering (Edelsbrunner et al., 2001), its application in genomics is limited. We believe that this is the first time these two methods were applied in Hi-C studies. One of our first contributions is to unify these two frameworks in *TreeHiC*. This makes possible by considering the space of minima and maxima as our main features when detecting differential interactions. Additionally, the hierarchical/nested structures of our partition algorithm make it feasible to accompany with the tree testing in Yekutieli (2008). The connection between these two methods proves to be considerably more powerful in complex large-scale Hi-C studies.

Second, we mentioned in Appendix A.1 that partition algorithm in (Gerber et al., 2012) is not fitting for large scale Hi-C studies due to its speed. Note that its integration to our pipeline is straightforward since its pipeline is implemented in the R package *msr*. Alternatively, for the partition and extremum search, we reply on the newly published software, the Topology ToolKit TTK (Tierny et al., 2017). Here, the contribution is to integrate the TTK implementation into our R package *TreeHiC*. Here, we emphasize this since our contribution to *TreeHiC* software has not been address in the main text. Though TTK is efficient due to its parallellization and C++ implementation, its integration is challenging, partly due to its input, The Visualization Toolkit (VTK) file format (`https://www.vtk.org/`). We successfully implemented a pipeline from Hi-C contact matrices to VTK output, as it is required for TTK's parallellization in the partition steps. Lastly, we performed more comprehensive evaluations as discussed in the main text. This contribution is critical since there are no guidelines of how to evaluate these new called interactions; each methods have their own evaluations and conclusions (e.g. Lun and Smyth (2015) and Djekidel et al. (2018)).

In Chapter 3, I discuss a unified statistical framework, called *tan*, to address the problem of differential enrichment analysis from histone modification ChIP-seq data. *tan* relies on a new test based on the adaptive Neyman test introduced by (Fan and Lin, 1998) and is applicable when the study design includes two or more

biological replicates across two or more biological conditions. Prior to this work, many different tools to identify differential ChIP enrichment between experimental conditions have been developed. However, they vastly differ in terms of usability and in range of applicability, reflecting the complex structure of ChIP-seq data. As a result, the choice of method impacts the quantity and characteristics of the identified differential regions. To overcome these limitations, we proposed a nonparametric method, TAN, to identify differentially enriched regions based on the ChIP-seq data. Different from all the currently available methods, *tan* models the spatial histone enrichment profiles, rather than simply considering the total read counts in a given region. Together with its ANOVA-based testing setting suitable for multiple conditions, TAN is widely applicable to diverse types of ChIP-seq data, avoiding pitfalls of making an improper choice of tool.

A APPENDIX A

A.1 The Morse-Smale Complex

Let $\mathfrak{h} : \mathcal{M} \rightarrow [0, 1]$ where $M \subset \mathbb{R}$ or $\mathbb{R}^2$ be a scalar function. These representations present the topological features of the function and form a baseline for an exploratory data analysis tool. Here, we are interested in splitting the domain $\mathcal{M}$ recursively into smaller partitions. In particular, the MS complex provides a tool to examine $\mathcal{M}$ based on the critical points of function $\mathfrak{h}$. Informally, the interior of each partition is a *monotonic* region which contains a single local minimum or maximum. Thus, the MS complex of function $\mathfrak{h}$ decomposes the domain $\mathcal{M}$ into partitions where $\mathfrak{h}$ is increasing or decreasing. When applied to log fold-change function $\mathfrak{h}$ as discussed in Chapter 2, Section 2.2, it provides a powerful way to represent, visualize, and compare the extrema of $\mathfrak{h}$.

We discuss Morse-Smale complex (MS) decomposition (Gerber et al., 2012) to form 2-D representations of $\mathfrak{h}$. Gerber and Potter (2012) implements MSR R package for approximating Morse-Smale complexes on k-nearest neighbor graphs of high-dimensional data proposed in Gerber et al. (2012). More recent implementations focus on variations of this algorithm. Among them, the Topology ToolKit TTK (Tierny et al., 2017) targets low dimensional (2D or 3D) domains for applications in scientific data analysis and visualization. It focused on a variation of the algorithm by Shivashankar and Natarajan (2012) to provide an efficient parallellization. During the initial stage of this work, we utilized the MSR package (Gerber and Potter, 2012) to explore and validate our initial results. We enjoyed the simplicity of the algorithm, and the level of integration to our Hi-C analysis since it was written in the R statistical environment (R Core Team, 2018). Although efficient in small Hi-C data and applicable in high-dimensional data, this tool is considerably slow when dealing with large-scale Hi-C studies. This constitutes a serious limiting factor for our Hi-C analysis. Since we only work with a low dimensional domain (e.g., 2D in Hi-C data), we adapted the TTK software platform (Tierny et al., 2017) to perform the partition and extreme search in our pipeline due to its speed and efficiency.

Here, since we mainly focus on the expository aspect and basic understanding of the MS decomposition, we briefly introduces the main concepts and highlights properties relevant for the proposed partition and extremum search proposed in Gerber et al. (2012).

Preliminary

The Morse-Smale complex (MS) relates the number and connectivity of critical points (i.e., maxima, minima, and saddle points) of the function $h : \mathcal{M} \to [0, 1]$ where $M \subset \mathbb{R}$ or $\mathbb{R}^2$ is a smooth, compact manifold. A smooth function $h : \mathcal{M} \to [0, 1]$ is *Morse* if for all critical points x of h the Hessian matrix $Hf(x)$ is not singular. An *integral line*, $\lambda : \mathbb{R} \to \mathcal{M}$ is a curve in $\mathcal{M}$ with $\frac{d\lambda}{ds}(s) = \bigtriangledown h(\lambda(s))$. Define $src(\lambda) = \lim_{s\to-\infty} \lambda(s)$ and $sink(\lambda) = \lim_{s\to\infty} \lambda(s)$ as source and sink of the integral line, respectively. We emphasize that source and sink are both, by definition, critical points of h. We next define the ascending and descending manifolds of a critical points x as

$$A(x) = \{\lambda : src(\lambda) = x\},$$

$$D(x) = \{\lambda : sink(\lambda) = x\}.$$

A Morse function h is Morse-Smale (MS) if the ascending and descending manifolds intersect transversally only. In other words, the MS complex is the set of intersections $A(x_i) \cap D(x_j)$ over all critical points x_i, x_j.

Persistence Simplification and Hierarchical Partitions

The MS complex introduces a measure of strength of each extremal point, called *persistence*. We aim to define this measure formally. Here, persistence describes the significance of an extremal points in geometric terms, and not in the statistical sense of hypothesis testing. Let x_i be the critical points of h. Define $s(x_i)$ as the set of critical points that have a direct integral line connecting to x_i.. Let $n(x_i) = \arg\min_{x_j \in s(x_i)} \|h(x_i) - h(x_j)\|$, the persistence of a critical point x_i is defined as $p(x_i) = \|h(x_i) - h(n(x_i))\|$. Roughly speaking, persistence is amount to the change of

h in L_∞-norm such that the critical point pair $(x_i, n(x_i))$ is either canceled or merged into a single critical point. Note that we emphasized in Chapter 2, Section 2.2 that removing the critical points recursively with increasing persistence levels leads to a nested (hierarchical) series of MS complices, also called a filtration (Edelsbrunner and Harer, 2009). At each level, some of the partitions induced by the MS complex are merged into a single partition until the MS partitioning consists of only a single partition (i.e., the entire input domain).

Computation of the Morse-Smale Complex

As previously mentioned, the MS complex is defined in terms of ascending and descending manifolds. The definition itself leads to a direct algorithm. Here, we present an algorithm to compute MS complex partitions in Gerber et al. (2012). Let our data be $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}$ or $\mathbb{R}^2$ and associated scalar function values $Y = \{h(x_1), \ldots, h(x_n)\}$. By following the gradient at x_i which needs to be estimated in advance, we could determine the source and sink at for each data point x_i.This section describes an algorithm to compute the source and sink for each point x_i by approximating the domain via a k nearest neighbor graph (algorithm 1). The algorithm relies paths of steepest ascent and descent based on the graph connectivity

(Gerber et al., 2012).

Result: $\mathtt{partition}(.)$, data structure with partition assignments
Data set with observation x_i and scalar function value $h(x_i)$, $\{(x_i, h(x_i))\}_{i=1}^{n}$
Adjacencies of k-nearest neighbor graph, $\mathtt{adj}(x_i) = \{x_j : x_i \in knn(x_j), x_j \in knn(x_i)\}$
Direction of steepest ascent, $p_a(x_i) = \arg\max_{x_j \in adj(x_i)} h(x_j) - h(x_i)$
Direction of steepest descent, $p_d(x_i) = \arg\max_{x_j \in adj(x_i)} h(x_i) - h(x_j)$
(Approximate integral lines for each data point)
for $i \leftarrow 1$ **to** n **do**
  $x_a = x_i$
  (Find source for x_i)
  while $p_a(x_a) \neq x_a$ **do**
    $x_a = p_a(x_a)$
  end
  $x_d = x_i$
  (Find sink for x_i)
  while $p_d(x_d) \neq x_d$ **do**
    $x_d = p_d(x_d)$
  end
  Assign x_i to partition with maximum x_a and minimum x_d
  $\mathrm{partition}(x_i) = (x_a, x_d)$
end

Algorithm 1: Compute Morse-Smale complex partitions

Based on Algorithm 1's pseudocodes, each point x_i is assigned to a partition of the MS complex. As a result, we obtain a set of l partitions $\mathcal{C} = \{C_1, \ldots, C_l\}$ such that $\cup_i C_i = \{x_i\}_{i=1}^{n}$ and $C_j \cap C_i = \emptyset, \forall i \neq j$. For the next steps, an approximation of saddle point values between neighboring partitions is necessary to obtain the persistence-based hierarchy of the MS complex. This approximation is attained by verifying the points of the edges in nearest neighbor graph that cross partition boundaries These observations lead to Algorithm 2 (computing the persistence of

an extremal point), and Algorithm 3 (merge partitions/an extrema pair).

Result: p, persistence value of an extremal point x
Set of partitions that contain extrema x, $C_i, i = 1, \ldots, l_x$
Persistence of extrema x, $p_x = \infty$.
for $i \leftarrow 1$ **to** l_x **do**
  for $x_1 \in C_i$ **do**
    $t_x = 0$
    for $x_2 \in C_j$ **do**
      (Is (x_1, x_2) an edge?)
      if $x_1 \in adj(x_2)$ *or* $x_2 \in adj(x_1)$ **then**
        (Is edge crossing partitions?)
        if *partition*$(x_1) \neq$ *partition*(x_2) **then**
          $\Delta = \max(|h(x) - h(x_1)|, |h(x) - h(x_2)|)$
          (Update closest saddle?)
          if $t_x < \Delta$ **then**
            $t_x = \Delta$
          end
        end
      end
    end
  end
end
if $t_x < p_x$ **then**
  $p_x = t_x$
  (Store extrema pair and its persistence)
  $p = (extrema(C_i), extrema(C_j), p_x)$
end

Algorithm 2: Compute persistence of an extremal point x

In these algorithms, the choice k of the number of nearest neighbors affects the MS complex approximation. For instance, a large k could act a a smoothing of h and remove some of the effects of noise. However, large k increases the potential shortcuts that results in the merge of two valid partitions. On the other hand,

Result: $\mathtt{partition}(.)$
Extrema pair to merge ordered such that $h(x_1) < h(x_2) : x_1 = \mathtt{extrema}(C_i)$, $x_2 = \mathtt{extrema}(C_j)$
(Update partitions containing extrema x_1, x_2)
for $i \leftarrow 1$ **to** n **do**
  (Maximum and minimum of the partition of x_i)
  $(x_a, x_d) = \mathtt{partition}(x_i)$
  if x_1, x_2 *are maxima* **then**
    (Update partition?)
    if $x_1 == x_a$ **then**
      $\mathtt{partition}(x_i) = (x_2, x_d)$
    end
  else
    (Update partition?)
    if $x_2 == x_d$ **then**
      $\mathtt{partition}(x_i) = (x_a, x_1)$
    end
  end
end

Algorithm 3: Merge partitions

small k increases the potential of introducing *artificial* extrema caused by noise or connecting to a set of nearest neighbors that are not representative of the directional derivatives of h. The discussions in Gerber et al. (2010, 2012) proposed the number of nearest neighbors in the MS computation is $k = 5d$, where $d = 2$ in the case of Hi-C data. They presented that with low dimension (e.g., $d = 1, 2$, or 3), this choice of k performs well in their simulated studies. However, its performance degrades quickly when applying to large dimension d. Fortunately, for the scope of Hi-C data, only the case $d = 2$ is required.

A.2 Additional Materials for Simulation Studies

We further present detailed illustrations of our simulation configurations and generation of Hi-C contacts under the two proposed models from the literature.

Simulation Model 1

Here, we discussed the generation process of Hi-C contact maps described in Stansfield and Dozmorov (2017). Interaction frequencies (IFs) at each distance $d = |x-y|$ can be modeled by the following components, $IFs \sim \hat{IF}_d + spead_d + sparsity_d$. $\hat{IF}_d$ is the expected IF at distance d, $spread_d$ is the distribution of IFs at that distance. For the first component, $\hat{IF}_d$ was estimated by fitting the power-law distribution $IF_d = C * d^{-\alpha}$ by maximum likelihood estimation. Here, α is from 1.8 to 2.2 on GM12878 cell, at resolution from 1Mb to 50kb, on chromosome 1. For the second component, $spread_d$ is estimated using a normal distribution $N(0, SD)$, where $SD \in (1.6, 3.2)$ is the standard deviation of IF_d. This parameter is set to 1.9 in the current simulations. Lastly, since real Hi-C contact maps contains many zeros, the proportion of zeros was modeled as $P(IF = 0) = \gamma * distance$ where $\gamma = 0.001$ by default. To add known differences, model 1 introduces fold changes to one of the matrices. Specifically, the IFs at a given interaction differences were altered as $IF_{x,y,\theta} = \theta^{\nu} * IF_{x,y}$.

Simulation Model 2

In this second simulation model, FIND (Djekidel et al., 2018) used the K562 Hi-C heat map as a reference. To induce pairs of interaction (x, y), they used a negative-bionomial distribution with dispersion of 10^4. Nondifferential pairs are sampled from a negative binomial with a mean from the correponding interactions in K562 cell. Furthermore, the differential pairs are sampled from a negative binomial with a mean equal to the fold change of their corresponding pairwise interaction in K562 contact map. The sparsity of interaction pairs is approximately 1%. Since FIND also accounts for local spatial dependency between interacting loci, they applied a Gaussion smoother to simulate the effect of correlations in the vicinity of each differential interaction loci.

Simulation Settings

We conducted simulations to evaluate model performance under two main settings: (i) the fold-change values to account for signal-to-noise ratios, and (ii) resolution parameter to account for sparsity in Hi-C studies. As a result, the simulation study included 6 signal configurations for each proposed models: 3 levels of signal-to-noise ratios times 2 values of resolutions. Each configuration was run 10 times. For each run, we recorded the observed proportions of false discoveries for FDR thresholding at 0.05.

For our simulation to be similar to real data, we use a MA plot between K562 and GM12878 cells (Rao et al., 2014). This MA plot mimics that of gene expression analysis where M-value is the $\log_2$ fold change between libraries. A-value presents the average log-count-per-million (CPM), i.e., the average abundance across all libraries. Such MA plots could be plotted using the R package *edgeR* (Robinson et al., 2009). Here, we present a sample of MA plots between K562 and GM12878 cells for resolution 50K (Figure A.1), and resolution 10K (Figure A.2). Based on these observations, for the fold change effects, we set $\log_2$ fold change values of $1.5, 2, [4, 6]$ as small, medium, and large effects, respectively.

For the sparsity parameter, we applied dimension of Hi-C matrices $n = 300, 1000$ for dense and sparse settings, respectively. Furthermore, for the dense setting, we randomly select approximately 1000 pairwise contacts. On the other hand, for the dense setting, we tried to make the simulated differential interactions as sparsely distributed as possible by selecting $\leqslant 500$ pairwise contacts.

A.3 Experimental Hi-C Data: Processing and Normalization

Hi-C assay and its variants generate hundreds of millions of short paired-end reads which typically range from 40bp lines to 101 bps. Datasets utilized for illustration are lymphoblastoid cell lines (GM12878 and K562) with accession number GSE63525 obtained from NCBI Gene Expression Omnibus (Barrett et al., 2010).

Hi-C Data Processing Pipeline

Data were generated using *in situ* Hi-C (developed by Rao et al. (2014)). Here, we summarize the Hi-C data processing pipeline described in Rao et al. (2014).

Sequence Alignment

All Hi-C data was generated using Illumina paired-end sequencing. Most reads were 101bp paired ends. The Illumina sequencer produces two fastq files, one for each read end. The pipeline begins by splitting each of the two fastq files into chunks containing 1.5 million single end reads, with roughly 200 chunks for one lane of data. Each chunk is mapped to b37 (for human) using the Burrows-Wheeler single end aligner, *bwa-sq* (Li and Durbin, 2010), with default parameters. After alignment, each fastq file chink has a corresponding SAM file. Next, the two sorted SAM files for each chunk (corresponding to the first and second read) are merged into single, paired-end SAM file. We also provide sequencing depths for each studied cell lines. The sequencing depths for K562's four samples are 47.2, 48.1, 46.4, and 44.9 where values are $\times 10^6$. For GM12878, number of reads are 197.0, 202.4, 146.1, and 59.7 (values $\times 10^6$).

Filtering of Abnormal Alignments

About 75% of the time, each read in a read pair will align to a single site in the genome. We call such read pairs "normal." Another 20% of read pairs are "chimeric." This means that at least one of the two reads comprises multiple subsequences, each of which align to different parts of the genomes. Theses read pairs are classified as "unambiguous" or "ambiguous." In an "unambiguous" chimeric read pair, one read maps chimerically to both locus A and locus B, and the other read maps to either locus A or locus B, but not to both. These "unambiguous" chimeric read pairs comprise roughly 15% of all read pairs and are included in our maps as ligation junctions between locus A and locus B. All other chimeric read pairs are "ambiguous" and are not included in our Hi-C maps.

Filtering of Low-quality Alignments

Read pairs with low alignment quality were thrown out by applying a threshold. One of two thresholds is applied: $MAPQ > 0$ (i.e., a unique "best" alignment exists) or $MAPQ \geqslant 30$ (i.e., the chances that a alignment is erroneous is at most 1 in 1000).

Construction of Contact Matrices

Contact matrices were generated using varying locus sizes. For instance, to calculate the contact matrix with a 1 Mb locus size (i.e., "1 Mb matrix resolution"), the linear genome is divided into 1 Mb bins and count the number of contacts observed between each pair of bins.

Contact Matrix Normalization

Let the number of observed contacts between locus i and locus j be denoted M_{ij}. Due to biases in the Hi-C experiment, chromatin accessibility, nucleosome occupancy, alignability and restriction site density at a locus can affect the contact count. In our analysis of K562 and GM12878, we an approach proposed by Lieberman-Aiden et al. (2009), namely *Square root vanilla coverage normalization*. In this approach, a row-specific normalization term R_i was calculated by summing the counts in a row (the L_1 norm) and taking the reciprocal. A column-specific normalization term C_j was calculated similarly. For intrachromosomal matrices, $C_i = R_i$. Next, the normalized matrix entry M^*_{ij} is therefore $R_i M_{ij} C_j$. Here, this normalization procedure is called "vanilla coverage normalization" (or "VC normalization".)

One issue with VC normalization is that it tends to overcorrect. A simple fix toward reducing this effect is to use the square root of the VC vector. The square root can be motivated very briefly by observing that such a correction makes the entries of M^*_{ij} dimensionless, by converting units of [reads] to units of $[reads]/[reads^{0.5}][reads^{0.5}]$.

A.4 Other Discussions

Discussions on diffHiC results

In this section, we discuss operating characteristics of *diffHiC* in details. In particular, we seek to examine the following: (i) when does diffHiC fail?, and (ii) simulation scenario where diffHiC works?

First, we address the cases when diffHiC's performance degrades in our simulation settings. As mentioned in the main text and Appendix A.2: Additional materials for simulations, model 1 is utilized to quantify methods in the case of no biological replications (e.g., $n = 1$). This is an important case as $n = 1$ is a typical sample size in Hi-C studies. Prior to this work, there are no available tools that present a framework in this setting. In their main model description (Lun and Smyth, 2015), it is required that $n \geqslant 2$ to estimate the model's parameters. However, in their implementation in the R package *diffHiC*, they also included an option to perform differential detection when $n = 1$, with a warning that their dispersion parameter estimation is not reliable in this case. Here, we only include diffHiC for the comparison purpose with our tree methods. We acknowledge that diffHiC model's requirement is not satisfied in this scenario. As expected, in model 1's simulation studies, diffHiC failed in both two resolution configurations, namely low resolution (dense) in Figure 2.5, and high resolution (sparse) in Figure 2.6. Furthermore, according to model 1's generation steps in Appendix A.2, the Negative Bionomial does not hold. This is in contrast with model 2 where interaction differences were generated by Negative Bionomial models, which we discuss in the next paragraph. This factor of model robustness also contributes to diffHiC's overall performance.

On the other hand, under model 2, diffHiC's performance improves markedly. In particular, its performance is comparable with those of tree methods in the case of large fold change (e.g., $\log_2(\text{fold change}) \geqslant 4$), and low resolution in Figure 2.7. Such improvement comes from two factors. First, model 2's generation utilizes Negative Binomial model, which is in agreement with diffHiC's model assumption.

Second, dispersion parameters in diffHiC can be estimated due to sample size $n = 2$. However, when moving the high resolution configuration (dense), diffHiC's performance was deteriorated. In summary, in our simulation studies, diffHiC fails when: (1) its Negative Binomial is not satisfied, (2) there are no biological replicates (i.e., $n = 1$), and (3) Hi-C resolution is high.

Discussions on Theoretical Guarantees of TreeHiC's Hierarchical Framework

We next sought to examine the theoretical guarantees of our hierarchical testing framework based on Yekutieli (2008). Specifically, we discuss a universal bound for the FDR of the discoveries in TreeHiC.

Let q (e.g., 0.05) be the FDR target level for our differential testing. We aim at finding a universal bound for our (tree) FDR if we follow the testing procedure proposed in the main text. Based on theoretical results derived in Yekutieli (2008), a bound for the FDR is as follows:

$$\text{FDR} \leqslant q \times \delta^* \times \text{fdr-multiplier}$$

where

- δ^* is a multiplicative factor where its maximum is at 1.44. In most scenarios, $\delta^* \approx 1$ is sufficient (Yekutieli, 2008) based on their simulation studies.
- For fdr-multiplier, Yekutieli (2008) proposed the universal bound fdr-multiplier $<$ 2.

In a practical case of Hi-C differential analysis, the number of discoveries (i.e., the called differential interactions) greatly exceed the number of families tested, especially in high-resolution case. If this holds, Yekutieli (2008) proposed another bound on fdr-multiplier, namely:

$$\text{fdr-multiplier} \leqslant \frac{(\text{observed no. of discoveries} + \text{observed no. of families tested})}{(\text{observed no. of discoveries} + 1)}$$

Although the theoretical properties of the above fdr-multiplier are not clear, we could estimate it for different applications. Specifically, for Hi-C applications where the number of called interactions is much greater than the number of families tested, we approximate fdr-multiplier ≈ 1. In summary, FDR could be bounded the following: (i) $2 \times 1.44 \times q$ (most conservative), (ii) $1.44 \times q$, and (iii) q. Based on our simulation and experimental studies, we found that $\text{FDR} \leqq q$ is sufficient. For our current pipeline, it is our default setting. However, user could use different bounds by adjusting the FDR target level q accordingly.

In summary, hierarchical FDR methodology can be used to control the FDR in complex large-scale studies. It also is considerably more powerful than the BH Benjamini and Hochberg (1995) procedure in sparse setting problems, providing that the data have a hierarchical structure. Throughout our procedure, the assumption is made that the p-values are independently distributed. Specifically, it implies that dependence between a parent's p value and any of its children should not be allowed. On the other hand, dependence across the tree can be allowed.

Notes on Transformations of Log Fold Change Function

We are interested in two transformations of the $\log_2$ fold change function $h(x, y) = \log_2\left(\frac{\bar{G}(x,y)}{\bar{F}(x,y)}\right)$:

1. Apply an inverse transformation: $h^*(x, y) = \log_2\left(\frac{\bar{F}(x,y)}{\bar{G}(x,y)}\right)$:
2. Apply a positive number: $h^*(x, y) = \log_2\left(\frac{\bar{G}(x,y)}{\bar{F}(x,y)+\epsilon}\right)$:

For the second transformation, we are interested on the results of our analysis when applying different values of $\epsilon > 0$. Here, we seek to explain that these two transformations do not change our analysis, namely our (x, y)-loci of extrema.

Without loss of generality, we only explain our arguments for the first transformation. There are several ways to explain this conclusion. For the first explanation, though these transformation do change the value of $h(x, y)$, it does not change the coordinates (x, y) where h achieves (local) maxima or minima. We could see this by taking the first derivatives of h and h^*. By setting the first derivatives to zeros, these two normal equations have the same roots. Therefore, the sets of loci where extrema of h or h^* are achieved are the same. As a result, this transformation does not change the result of our analysis. This is because for our partition and extremum search, we only require the set of (x, y)-loci where extrema occur. For the second explanation, we arrived at the same conclusion by running a simulation with and without the inverse transformation.

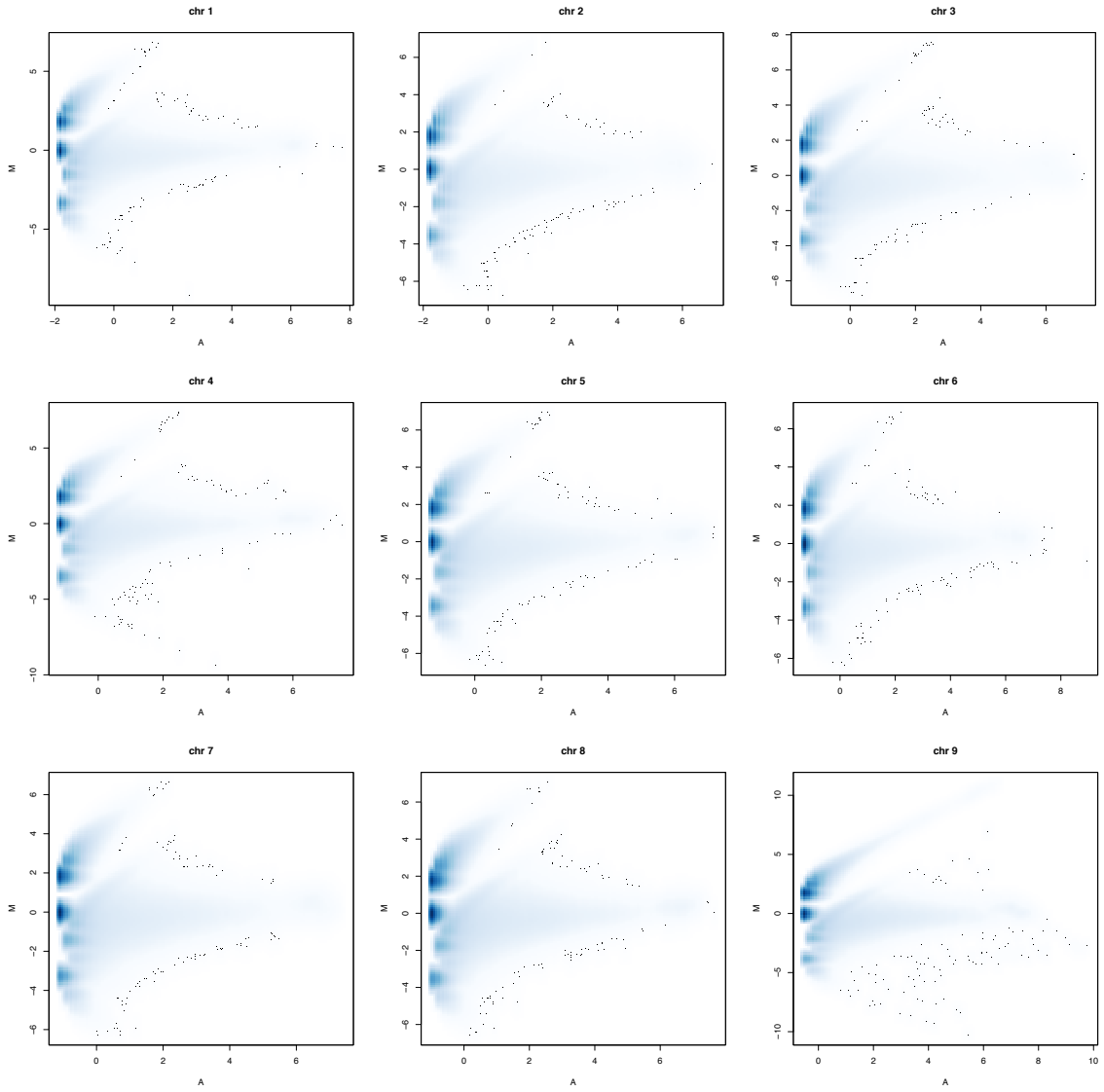


Figure A.1: MA plots for chromosomes 1-9 for resolution 50K: Log fold change with respect to the average abundance for readl Hi-C data. Each point represents a 50Kb bin pair. The M-value is defined as the library sized-adjusted $\log_2$-fold change between replicates for K562 and GM12878 cells. x-axis shows the M-values. y-axis shows the A-values.

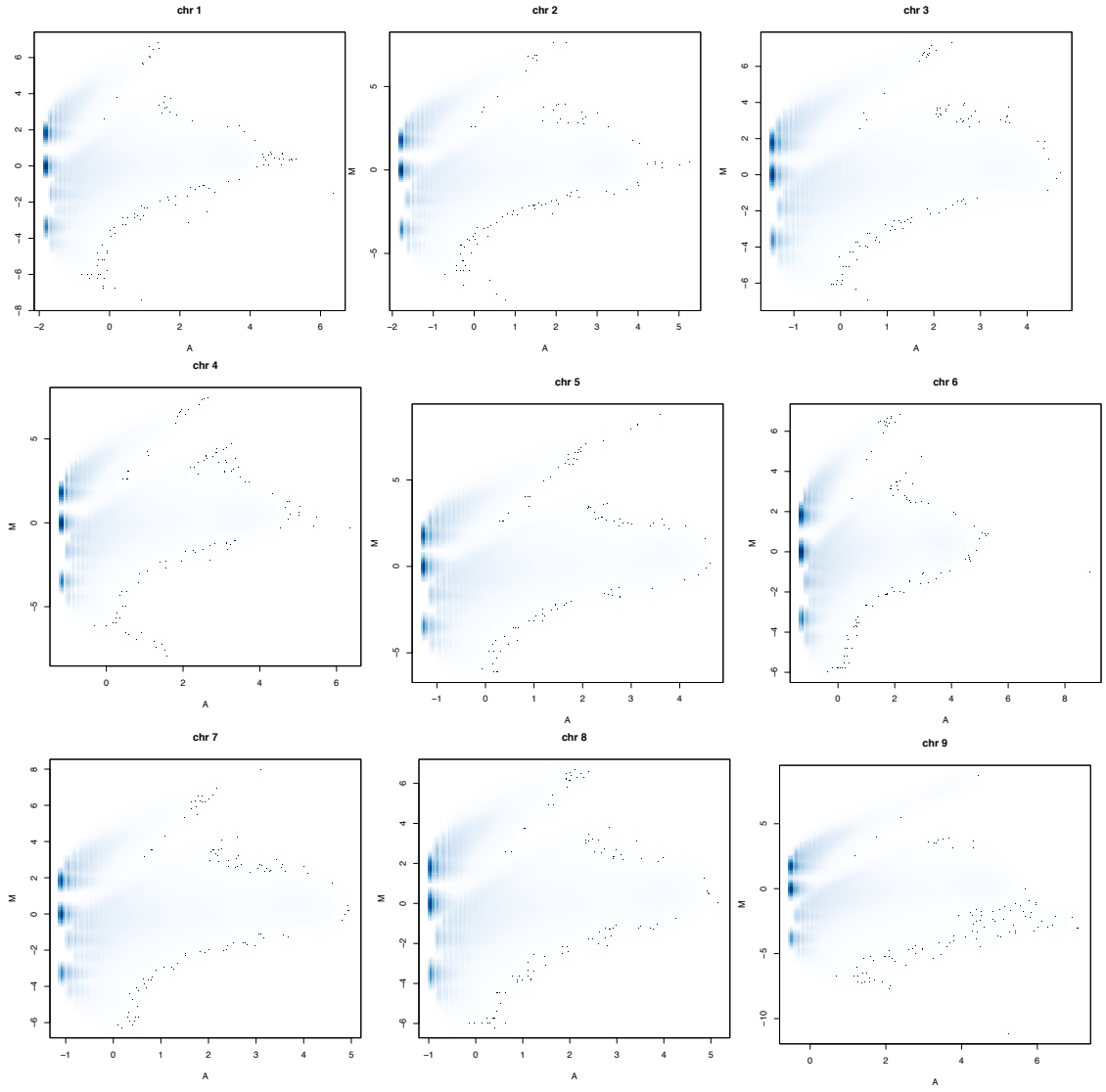


Figure A.2: MA plots for chromosomes 1-9 for resolution 10K: Log fold change with respect to the average abundance for readl Hi-C data. Each point represents a 10Kb bin pair. The M-value is defined as the library sized-adjusted $\log_2$-fold change between replicates for K562 and GM12878 cells. x-axis shows the *M*-values. y-axis shows the *A*-values.

B APPENDIX B

B.1 An Extension to Multiple Conditions

We discuss an adaptive high-dimensional ANOVA, called HANOVA from Fan and Lin (1998). To ease the notation, we only present the framework for a given genomic region. The collected data are of form $\{X_{kj}(t)\}$ where $\mathbb{E}[X_{kj}(t)] = f_k(t)$ and $Var[X_{kj}(t)] = \sigma_k^2(t)$. Here K denotes the index of conditions ($1 \leqslant k \leqslant K$), j represents the membership of each condition ($1 \leqslant j \leqslant n_k$), and t is the index for genomic positions ($1 \leqslant t \leqslant T$). Of interest is to test

$$H_0 : f_k(t) = f(t) \text{ for } t = 1, \ldots, T \text{ and } k = 1, \ldots, K.$$

Let $\bar{X}_k(t)$ be the average curve of the kth condition, namely

$$\bar{X}_k(t) = n_k^{-1} \sum_{j=1}^{n_k} X_{kj}(t),$$

and let $\hat{\sigma}_k^2(t)$ be the standard deviation curve

$$\hat{\sigma}_k^2(t) = (n_k - 1)^{-1} \sum_{j=1}^{n_k} \{X_{kj}(t) - \bar{X}_k(t)\}.$$

Building upon the framework of the adaptive test for the two-sample comparison, the HANOVA test statistics can be evaluated as follows

$$F^* = \max_{1 \leqslant m \leqslant T} \frac{1}{\sqrt{2(K-1)m}} \left\{ \sum_{t=1}^{m} \sum_{k=1}^{K} n_k \hat{\sigma}_k(t)^{-2} \{\bar{X}_k(t) - \bar{X}(t)\}^2 - (K-1)m \right\},$$

where $\bar{X}_k(t)$ and $\hat{\sigma}_k^2(t)$ were defined above and

$$\bar{X}(t) = \sum_{k=1}^{K} n_k \hat{\sigma}_k(t)^{-2} \bar{X}_k(t) / \sum_{k=1}^{K} n_k \hat{\sigma}_k(t)^{-2}.$$

Finally, one can normalize the test statistics as in the main paper, leading to the HANOVA test statistics

$$F_{\text{HANOVA}} = \sqrt{2\text{loglog}T}F^* - 2\text{loglog}T + .5\text{logloglog}T - .5\log(4\pi).$$

One can use our testing procedure in the main paper to find p values.

B.2 Procedure of Differential Analysis for Specific Sample Sizes

Procedure of differential analysis for sample sizes $n_1, n_2 = 4$

We describe our method of generating estimates of p-values in the case $n_1, n_2 = 4$. We introduce the following notations. Let $\{a_i, b_i, c_i, d_i\}$ and $\{A_i, B_i, C_i, D_i\}$ be labels for replicates from conditions 1 and 2 from peak i, respectively. The following describes our method for generating empirical null distribution of adaptive Neyman test. For condition 1, let $T_1^{min}, T_2^{min}, T_3^{min}$ be the adaptive Neyman tests from samples $(a_i, b_i$ vs. $c_i, d_i)$, $(a_i, c_i$ vs. $b_i, d_i)$, $(a_i, d_i$ vs. $b_i, c_i)$. The adaptive Neyman tests $T_1^{plus}, T_2^{plus}, T_3^{plus}$ are computed in a similar way with samples from condition 2. Under the null hypothesis of no differential enrichment, we could consider that T^{min}'s and T^{plus}'s are sampled from the unknown null distribution.

A naive way of computing the p-value is

$$\text{p-value} = \frac{|t \in \{T^{min}\text{'s}, T^{plus}\text{'s}\} : t \geqslant T|}{|\{T^{min}\text{'s}, T^{plus}\text{'s}\}|},$$

where the adaptive Neyman test T is computed from the samples $\{a_i, b_i, c_i, d_i\}$ vs. $\{A_i, B_i, C_i, D_i\}$. This is not appropriate since T and $\{T^{min}\text{'s}, T^{plus}\text{'s}\}$ have different degrees of freedom. To remedy this, we use the following procedure to generate adaptive Neyman tests for "between" conditions.

Let $S_- = \{(a_i, b_i), (a_i, c_i), (a_i, d_i), (c_i, d_i), (b_i, d_i), (b_i, c_i)\}$ and $S_+ = \{(A_i, B_i), (A_i, C_i), (A_i, D_i), (C$ $(B_i, D_i), (B_i, C_i)\}$. The "between" condition adaptive Neyman tests $T_j^{bt}, j = 1, \ldots, 36$

result from testing one pair of sample in S_- versus than in S_+. The corresponding p-values for T_j^{bt} is

$$p_j^{bt}\text{-value} = \frac{|t \in \{T^{min}\text{'s}, T^{plus}\text{'s}\} : t \geqslant T_j^{bt}|}{|\{T^{min}\text{'s}, T^{plus}\text{'s}\}|}.$$

Since there are large biological variability, the estimations of p-values are expected to be highly variable. To obviate this problem, we pool peaks with similar total counts to generate robust estimates of p-values. Specifically, peaks are binned into pre-specified quantiles determined on the averaged counts per peak. To obtain empirical p-values, we compute the probability of observing an adaptive Neyman test between biological replicates in the given bin, which is at least as large as the one observed for a given peak in the comparison between conditions. This leads to the following estimation:

$$p_j^{bt}\text{-value} = \frac{|t \in \cup_{\text{peak p} \in \text{Bin(test j)}} \{T_{\text{peak p}}^{min}\text{'s}, T_{\text{peak p}}^{plus}\text{'s}\} : t \geqslant T_j^{bt}|}{|\cup_{\text{peak p} \in \text{Bin(test j)}} \{T_{\text{peak p}}^{min}\text{'s}, T_{\text{peak p}}^{plus}\text{'s}\}|},$$

where $\mathtt{Bin}(\text{ test j})$ contains peaks with similar mean counts as the replicates are used to compute T_j^{bt}.

Raw p−values are subsequently corrected for multiple testing using the method of Benjamini and Hochberg. The final p-value for testing the hypothesis of differential enrichment at peak i is taken as the median of $p_j^{bt}, j = 1, \ldots, 36$. Taking the median is our current method of obtaining the final p-value of testing DE for peak i. In our simulation studies, this procedure controls FDR well and has high sensitivity. We emphasize that the procedure of generating p_j^{bt} is appropriate since all the adaptive Neyman tests have the same degrees of freedom. Further, by considering a pair of samples from one condition versus the others, we take into account of biological variability within conditions as opposed to getting a single test from all the samples in each conditions.

Procedure of differential analysis for sample sizes $n_1, n_2 = 2$

In the case of sample size $n_1, n_2 = 2$, it is not possible to generate the "within"

adaptive Neyman tests. The above procedure therefore cannot be extended in this case. We adapt an idea from permutation tests which under the null distribution and exchangeability assumption, the distribution of the statistics is the same after relabeling. Let T_1^i and T_2^i be adaptive tests result from testing pairwise samples (m_1^i, p_1^i) vs. (m_2^i, p_2^i), and (m_1^i, p_2^i) vs. (m_2^i, p_1^i), respectively. The "between" condition adaptive Neyman test T_{bt}^i is computed as a result of testing samples (m_1^i, m_2^i) vs. (p_1^i, p_2^i), as usual. The corresponding p-values for T_{bt}^i is

$$p^{bt}\text{-value} = \frac{|t \in \{T_1^i, T_2^i\} : t \geqslant T_{bt}^i|}{|\{T_1^i, T_2^i\}|}.$$

The pooled version is readily extended as follows:

$$p^{bt}\text{-value} = \frac{|t \in \cup_{\text{peak } p \in \text{Bin}(i)} \{T_1^p, T_2^p\} : t \geqslant T_{bt}^i|}{|\cup_{\text{peak } p \in \text{Bin}(i)} \{T_1^p, T_2^p\}|}.$$

B.3 Additional Materials For Simulation Studies

Precision-Recall Curves for Simulation Studies

We performed simulation studies to evaluate the precision-recall (PR) curves of TAN. The simulations were conducted by the same settings in the main text. We evaluated the PR curves for each method (Figure SB.2) by varying the threshold for the corresponding test statistics or p-values.

Next, by means of PR curves, we examined the ability of TAN to identify the set of non-null peaks at various sample sizes, and compared it to DESeq. The resulting PR curves for different sample sizes are shown in Figure SB.3 (overall), Figure SB.4 (for affinity case), and Figure SB.5 (for profile case).

Performance of TAN under different sample sizes and quantiles of obtaining final p-values

We performed simulation studies to evaluate operating characteristics of TAN under different sample sizes and various quantiles of obtaining our final p-values.

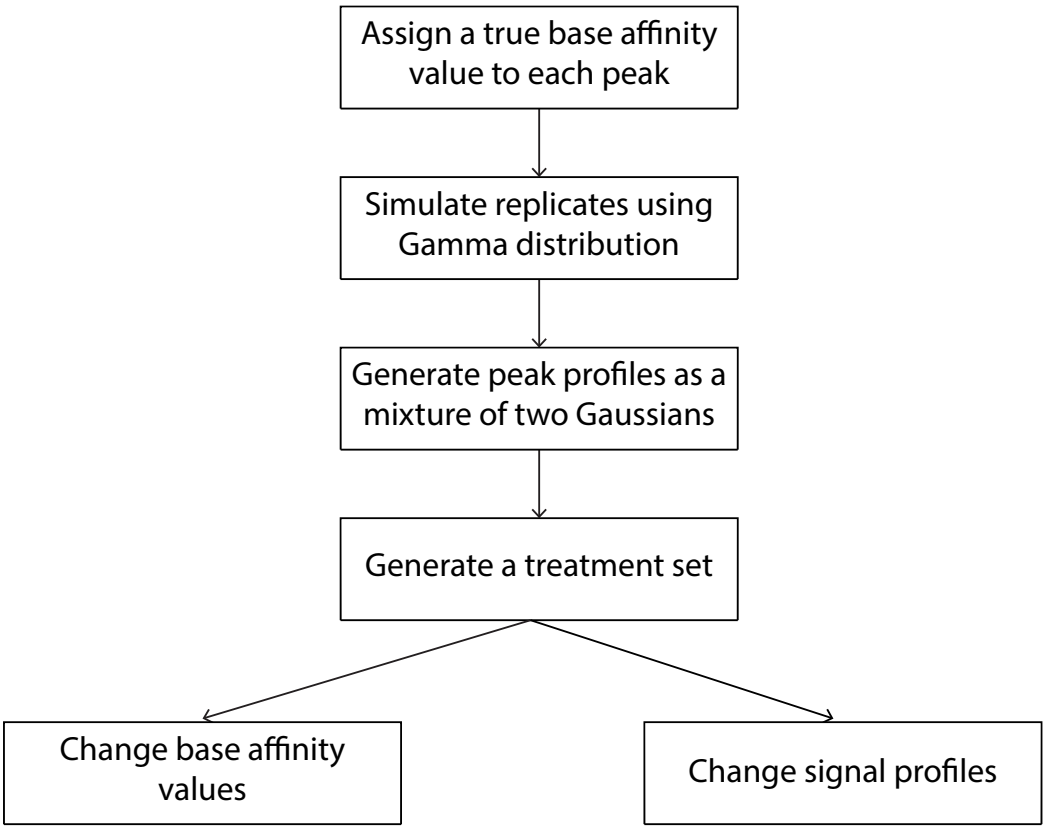


Figure B.1: Simulation Description

The simulations were conducted by the same settings in the main text. The results on the affinity and profile changes are shown in tables SB.1 and SB.2, respectively.

B.4 Additional H3K27me3 signals over gene body regions

To further support our findings in the main text, UCSC genome browser views of differential enriched regions at three annotated genes CDKN2A, COBLL1, and BZW2 are provided in Figures SB.6, SB.7, and SB.8, respectively. For simplicity, we only present the first replicate for each conditions 4HT+ and 4HT-. Furthermore, in

	TP	FP	FN	TN	eFDR(%)	SN(%)	SP(%
pTAN ($n = 4$, quant= 0.25)	74.5 + 0.0	1.0 + 0.0	25.5 + 0.0	9799.0 + 0.0	1.40 + 0.0	74.5 + 0.0	100.0 + 0.0
pTAN ($n = 4$, quant= 0.5)	72.5 + 6.4	0 + 0.3	27.5 + 6.4	9800 + 0.3	0.0 + 0.5	72.5 + 6.4	100 + 0.0
pTAN ($n = 4$, quant= 1.0)	69.0 + 0.1	0 +0	31 + 0.1	9800 + 0	0 + 0.0	69 + 0.1	100 + 0
pTAN ($n = 3$, quant= 0.25)	79.0 + 0.1	207.0 + 0.7	21.0 + 0.1	9593 + 0.3	71.4 + 0.0	79 + 0.1	97.88 + 0.0
pTAN ($n = 3$, quant= 0.5)	78.5 + 0.1	47.5 + 0.1	21.5 + 0.1	9752.0 + 0.1	38.1 + 0.1	78.5 + 0.1	99.5 + 0.0
pTAN ($n = 3$, quant= 1.0)	77.0 + 0.1	1.0 + 0.0	23.0 + 0.1	9799.0 + 0.0	1.41 + 0.0	77 + 0.1	100.0 + 0.0
pTAN ($n = 2$)	100 + 0.0	471 + 0.3	0.0 + 0.0	9329.0 + 0.3	82.6 + 0.0	100.00 + 0.0	95.2 + 0.0
DESeq ($n = 4$)	100 + 0.0	0.0 + 0.0	0.0 + 0.0	9800.0 + 0.0	0.0	100.0 + 0.0	100.0 + 0.0
DESeq ($n = 3$)	100 + 0.0	0.0 + 0.0	0.0 + 0.0	9800.0 + 0.0	0.0	100.0 + 0.0	100.0 + 0.0
DESeq ($n = 2$)	100 + 0.0	0.0 + 0.0	0.0 + 0.0	9800.0 + 0.0	0.0	100.0 + 0.0	100.0 + 0.0

Table B.1: Performance summary on 10 simulation replications for affinity changes under different sample sizes and various quants of obtaining our final p-values. Average power to detect simulated DE regions. Averages are calculated over 10 runs of simulated data sets. FDR level: 0.05, TP: true positives, FP: false negatives, FN: false negatives, TN: true negatives, eFDR: empirical FDR, SN: sensitivity, SP: specificity.

these gene regions, we clearly marked peaks from the union peak set and declared DE by TAN.

The coverage plots of H3K27me3 signals at genes ANKMY2 and PTAFR (Figure SB.9) are also provided to complete our analysis in the main text.

B.5 H3K27me3 data set: pre-processing and quality metrics

In this section, we append our analysis in the main paper with the pre-processing steps of the H3K27me3 data. Data generation process is described in detail in the main text. Here, we provide information about read mapping, and basic quality control.

1. Read mapping

   Sequence reads were mapped to the human genome hg19 (GRCh37) using BOWTIE (`http://bowtie-bio.sourceforge.net/`) with parameters `q -S -p`

	TP	FP	FN	TN	eFDR(%)	SN(%)	SP(%)
pTAN ($n = 4$, quant=0.25)	91.5 + 0.0	1.0 + 0.0	8.5 + 0.0	9799.0 + 0.0	1.1 + 0.0	91.5 + 0.0	100.0 + 0.0
pTAN ($n = 4$, quant=0.5)	86.5 + 0.0	0 + 0	13.5 + 0.0	9800 + 0.0	0.0 + 0.0	86.5 + 0.0	100 + 0.0
pTAN ($n = 4$, quant=1)	25.5 + 0.1	0 + 0.0	74.5 + 0.1	9800 + 0	0.0 + 0.0	25.5 + 0.1	100 + 0
pTAN ($n = 3$, quant=0.25)	95.0 + 0.0	207 + 0.3	5 + 0.0	9593 + 0.3	68.2 + 0.0	95 + 0.1	97.89 + 0.0
pTAN ($n = 3$, quant=0.5)	94 + 0.0	47.5 + 0.1	6.0 + 0.0	9752.5 + 0.1	33.33 + 0.1	94.9 + 0.0	99.5 + 0.0
pTAN ($n = 3$, quant=1.0)	84.0 + 0.0	1.0 + 0.0	16.0 + 0.0	9799.0 + 0.0	1.26 + 0.0	84.0 + 0.0	99.99 + 0.0
pTAN ($n = 2$)	97.0 + 0.0	471.0 + 0.3	3 + 0.0	9329.0 + 0.3	82.8 + 0.0	97.0 + 0.0	95.2 + 0.0
DESeq ($n = 4$)	0 + 0.0	0.0 + 0.0	77.0 + 0.1	9800.0 + 0.0	NaN	0.0 + 0.0	100.0 + 0.0
DESeq ($n = 3$)	0 + 0.0	0.0 + 0.0	100 + 0.0	9800.0 + 0.0	NaN	0.0 + 0.0	100.0 + 0.0
DESeq ($n = 2$)	0 + 0.0	0.0 + 0.0	100 + 0.0	9800.0 + 0.0	NaN	0.0 + 0.0	100.0 + 0.0

Table B.2: Performance summary on 10 simulation replications for profile changes under different sample sizes and various quants of obtaining our final p-values. Average power to detect simulated DE regions. Averages are calculated over 10 runs of simulated data sets. FDR level: 0.05, TP: true positives, FP: false negatives, FN: false negatives, TN: true negatives, eFDR: empirical FDR, SN: sensitivity, SP: specificity.

`10 -v 2 -m 4 --best --strata`. All the samples have mapping rates at least 70%. The resulting data sets are summarized in Table SB.3.

2. Data quality control

   We utilize the ENCODE consortium's quality metrics for analyzing the quality of the H3K27me3 data. Here, we computed the following quality control metrics for ChIP-seq: (1) PCR bottleneck coefficient (PBC), normalized strand cross-correlation coefficient (NSC), and relative strand cross-correlation coefficient (RSC). As illustrated in Table SB.4, PBC values for replicates 1,2, and 3 range from moderate to mild bottlenecking (PBC from 0.5-0.9), while PBC for replicate 4 is severe bottenecking (PBC from 0-0.5). According to ENCODE guidelines, 89% of Histone ChIP of ENCODE datasets have no or mild bottlenecking. Overall, all metrics show data of good quality.

Table B.3: Summary of total mapped read for each samples. The values are $\times 10^6$.

Samples	conditions	number of mapped reads
replicate 1, input	4HT-	63.5
	4HT+	55.0
replicate 1, ChIP	4HT-	87.6
	4HT+	90.3
replicate 2, input	4HT-	81.5
	4HT+	45.9
replicate 2, ChIP	4HT-	63.1
	4HT+	83.8
replicate 3, input	4HT-	36.3
	4HT+	33.0
replicate 3, ChIP	4HT-	58.8
	4HT+	56.4
replicate 4, input	4HT-	68.9
	4HT+	61.1
replicate 4, ChIP	4HT-	74.4
	4HT+	74.0

B.6 Other Discussions

The Epstein-Barr virus (EBV) nuclear protein EBNA3C datasets

The H3K27me3 datasets presented consist of ChIP-seq measurements from LCLs conditional for EBNA3C activity cultured in the presence (4HT+) or absence (4HT-) of 4-hydroxytamoxifen for two weeks, with 4 biological replicates per condition (GEO accession # GSE109221). Similarly, the EBNA3C (E3C) RNA-seq experiment was conducted under the same two conditions with eight replicates per condition. ChIP-seq data was aligned with Bowtie (Langmead et al., 2009) (version `0.12.7`) and RNA-seq data was processed with RSEM (Li and Dewey, 2011) (version `1.2.31`) using GENCODE version v19 for human genome version GRCh37.

Table B.4: Quality metric table for each samples. PBC: PCR bottleneck coefficient, NSC: normalized strand cross-correlation coefficient, RSC: relative strand cross-correlation coefficient.

Samples	conditions	PBC	NSC	RSC
replicate 1	4HT-	0.6858204	2.925859	1
	4HT+	0.6538893	2.604561	1
replicate 2	4HT-	0.532745	3.364339	1
	4HT+	0.738736	2.673006	1
replicate 3	4HT-	0.9278612	3.519682	1
	4HT+	0.9079123	2.913435	1
replicate 4	4HT-	0.01178538	3.926008	1
	4HT+	0.2446608	3.821252	1

Differential analysis of H3K27me3 in 4HT+ and 4HT- samples

R package DESeq, the implementation of the DESeq method, provides three approaches for shrinking the dispersion parameter in their NB model. Users can choose between local and parametric regression to estimate the dispersion. However, the parametric regression in the package is prone to failure and leads to poor point estimation test performance as discussed in Landau and Liu (2013). In our analysis, DESeq failed to estimate the dispersion parameter in all three ways proposed in their R package. An improved version, DESeq2, was proposed by Love et al. (2014) to remedy these drawbacks. Here, we utilized the DESeq2 R package for our differential peak calling. As expected, the parametric regression in DESeq2 failed to capture the dispersion trend. Hence, only the local regression results are presented. As mentioned in Landau and Liu (2013), this method is conservative, allowing overestimation of the dispersion.

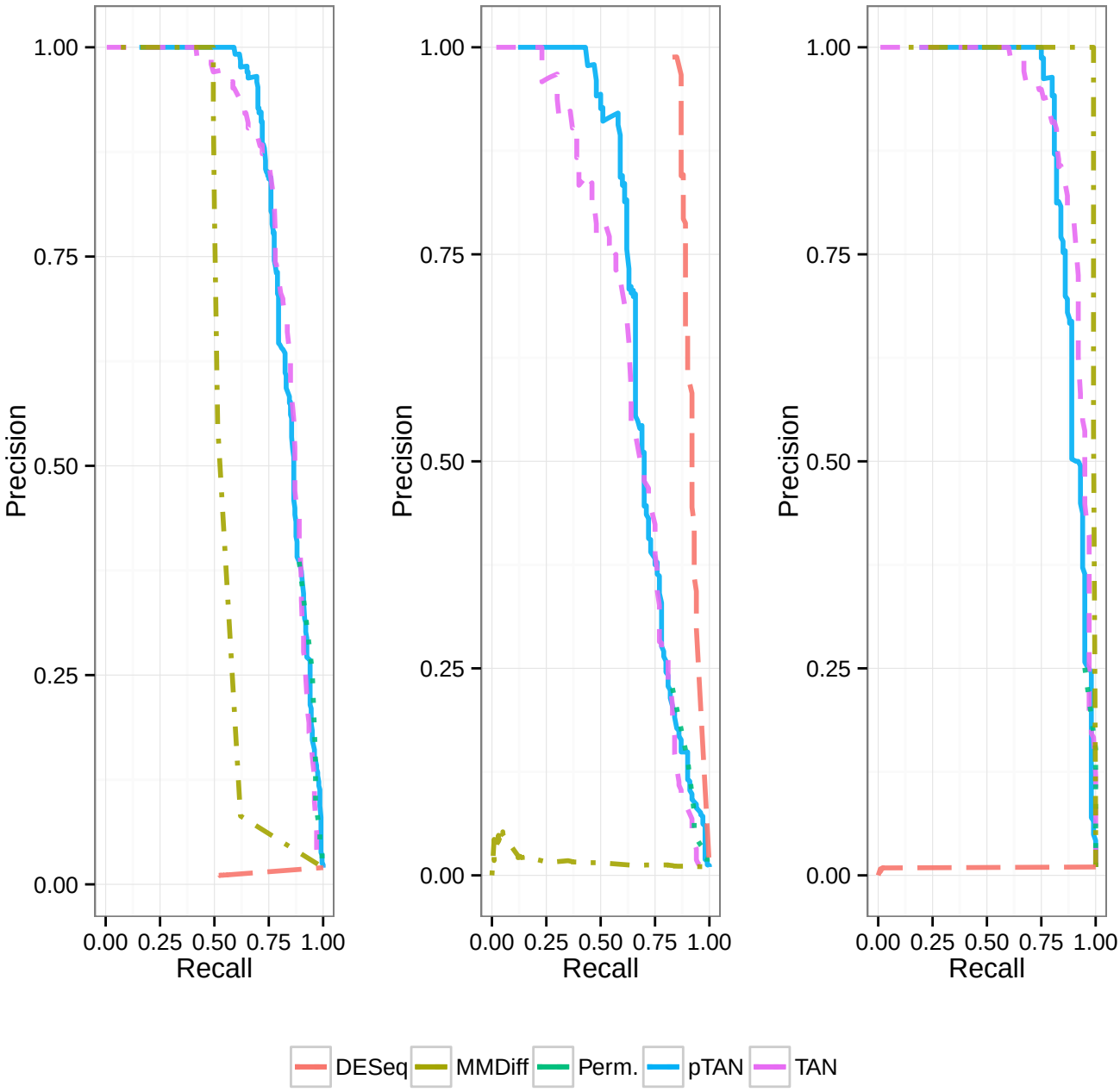


Figure B.2: PR curves for various methods under affinity and profile changes. Averages are calculated over 10 runs of simulated data sets. (Left) plot shows the overall PR curves, (Middle) plot shows PR curves for affinity case, and (Right) plot shows PR curves for profile case.

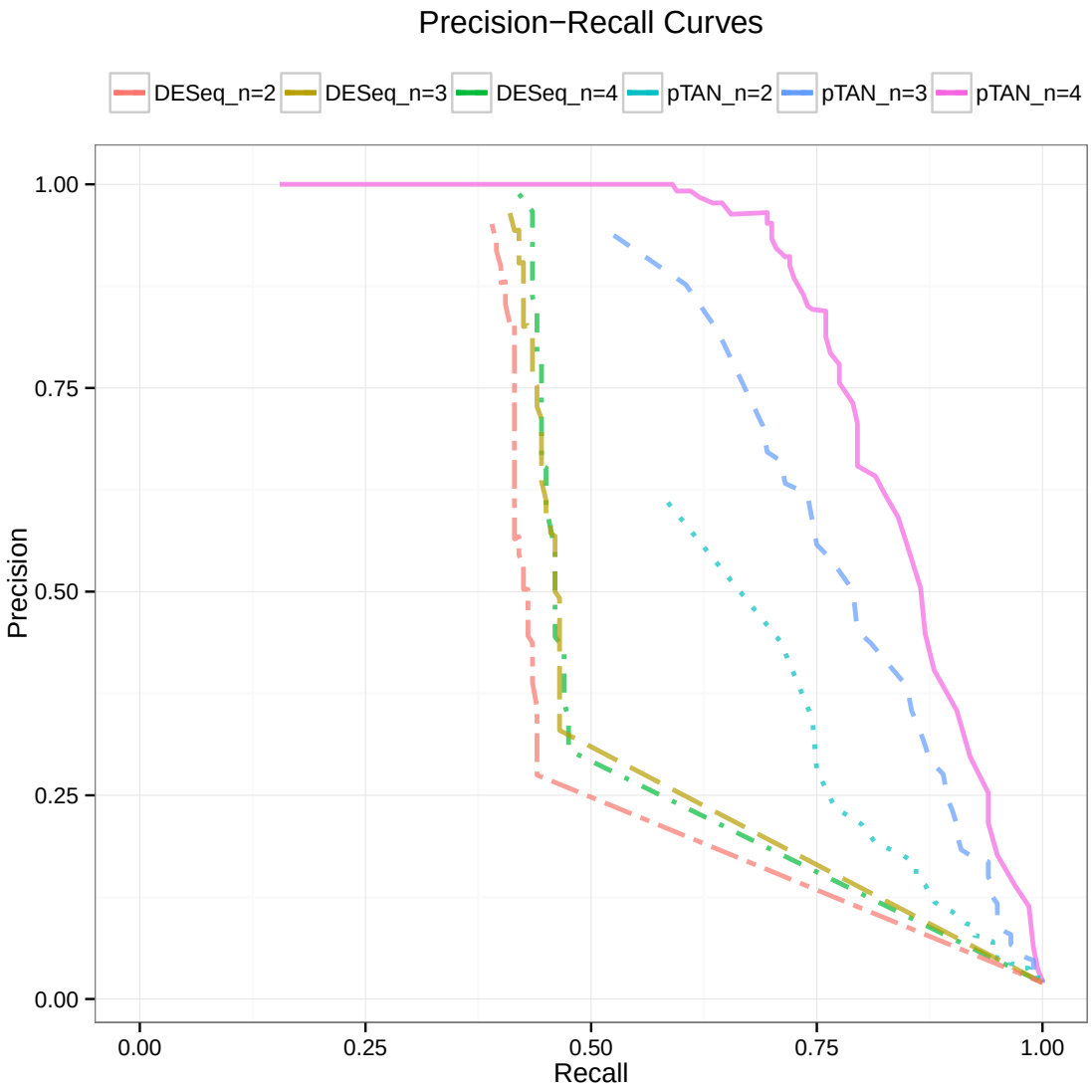


Figure B.3: The overall PR curves for various methods as a function of the number of replicates per condition. Averages are calculated over 10 runs of simulated data sets.

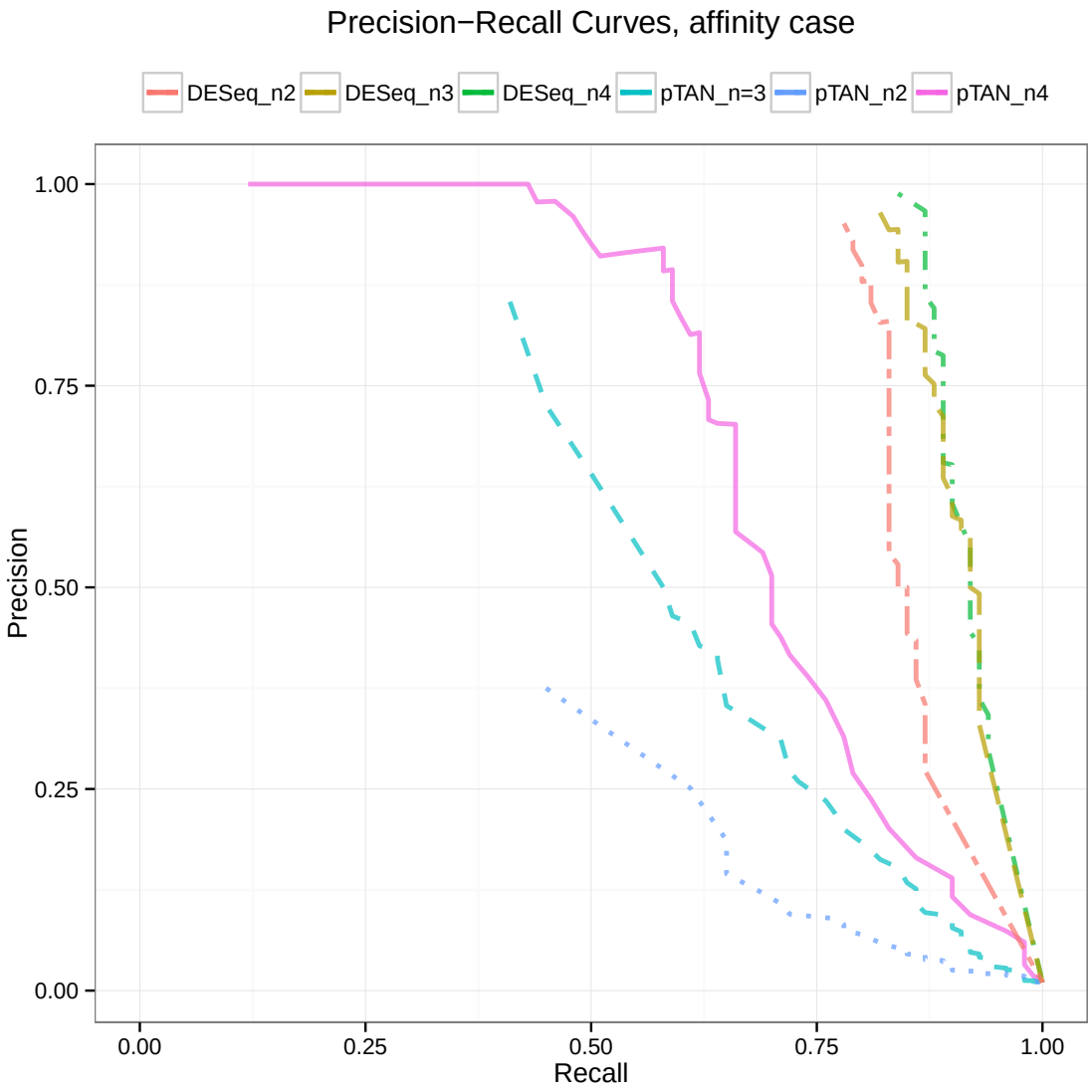


Figure B.4: The PR curves for affinity case as a function of the number of replicates per condition. Averages are calculated over 10 runs of simulated data sets.

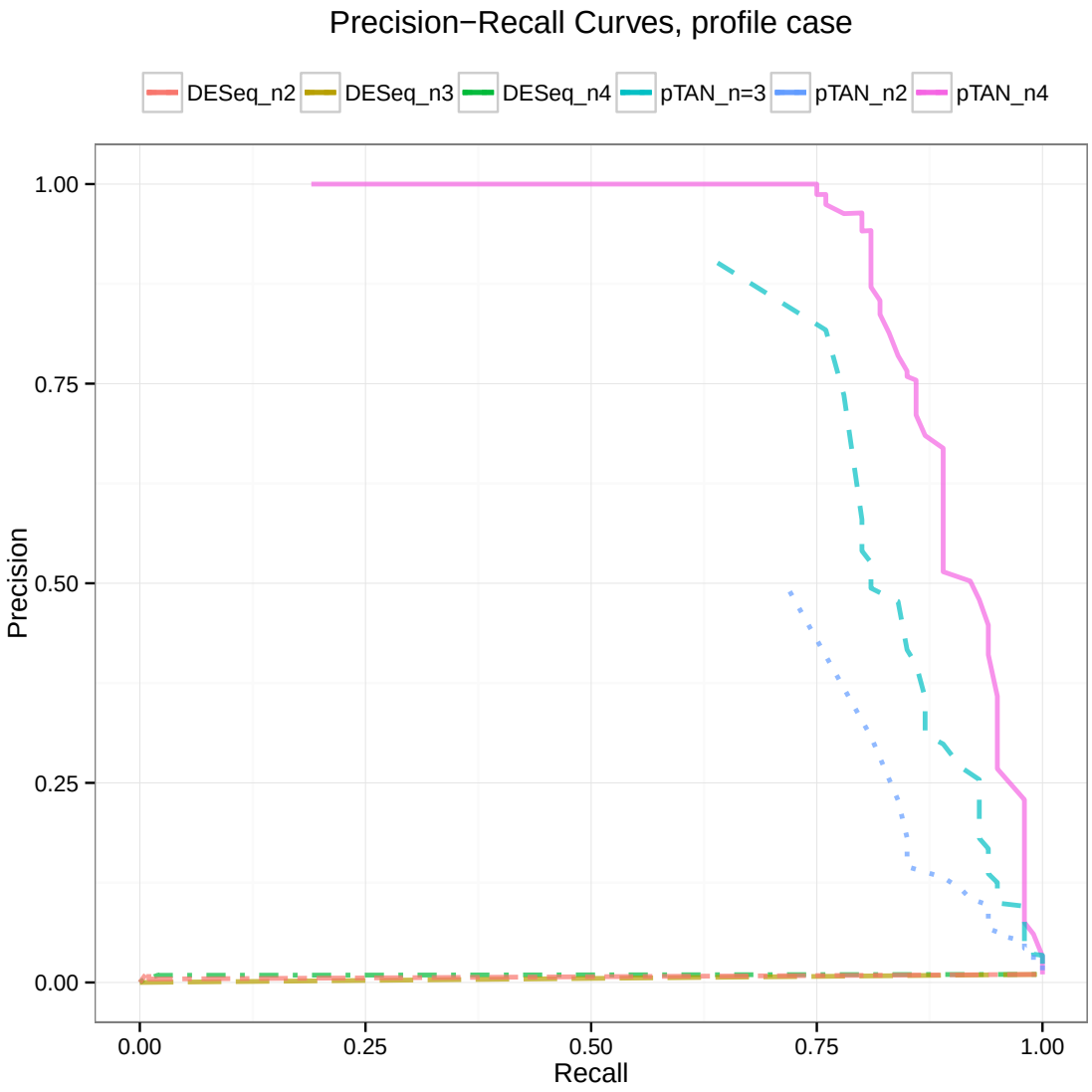


Figure B.5: The PR curves for profile case as a function of the number of replicates per condition. Averages are calculated over 10 runs of simulated data sets.

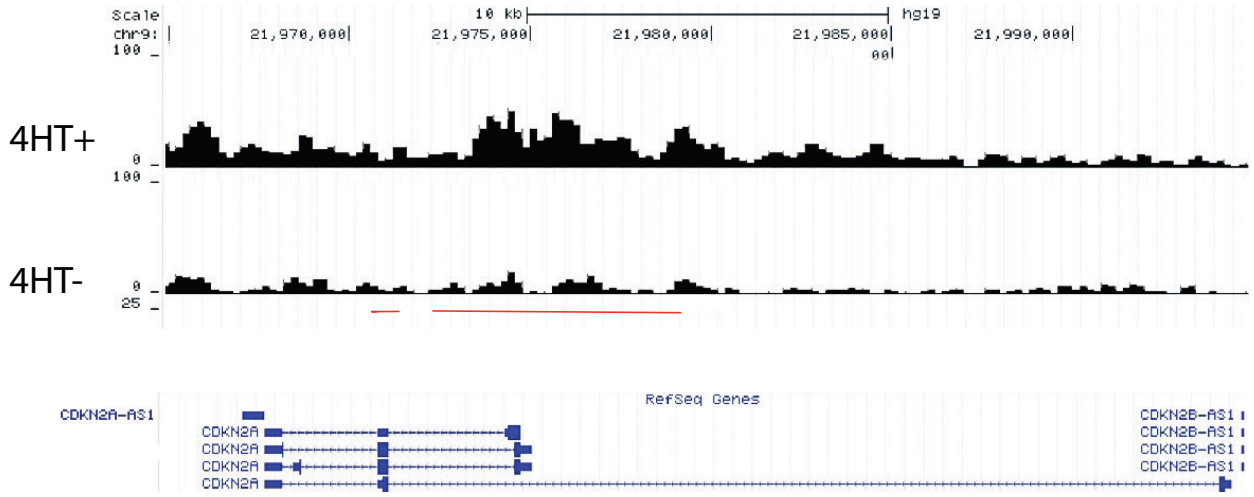


Figure B.6: Genome browser view of CDKN2A for replication 1. Red lines present the ChIP regions declared DE by TAN.

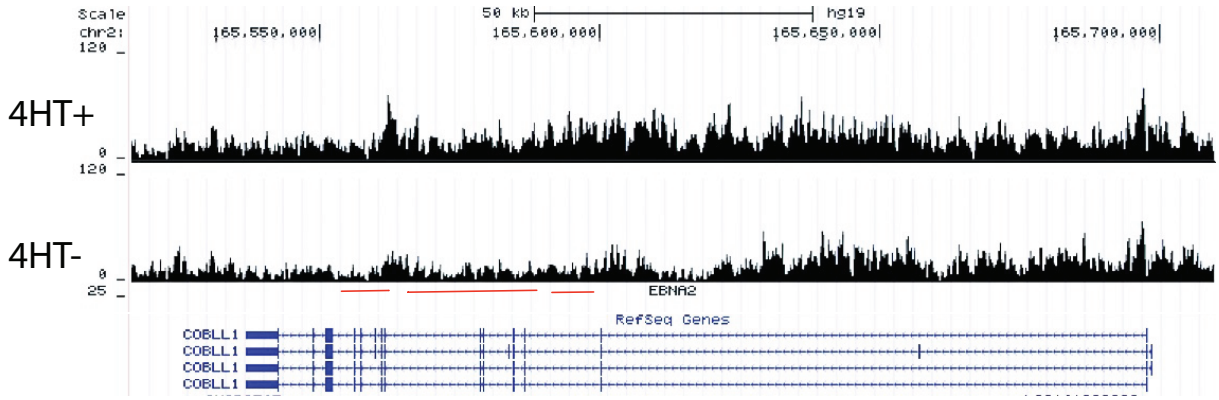


Figure B.7: Genome browser view of COBLL1 for replication 1. Red lines present the ChIP regions declared DE by TAN.

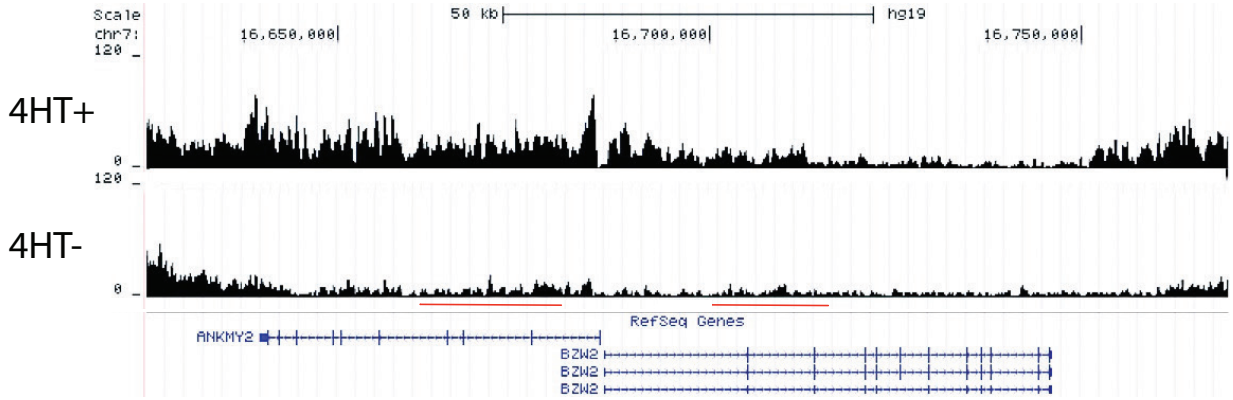


Figure B.8: Genome browser view of BZW2 for replication 1. Red lines present the ChIP regions declared DE by TAN.

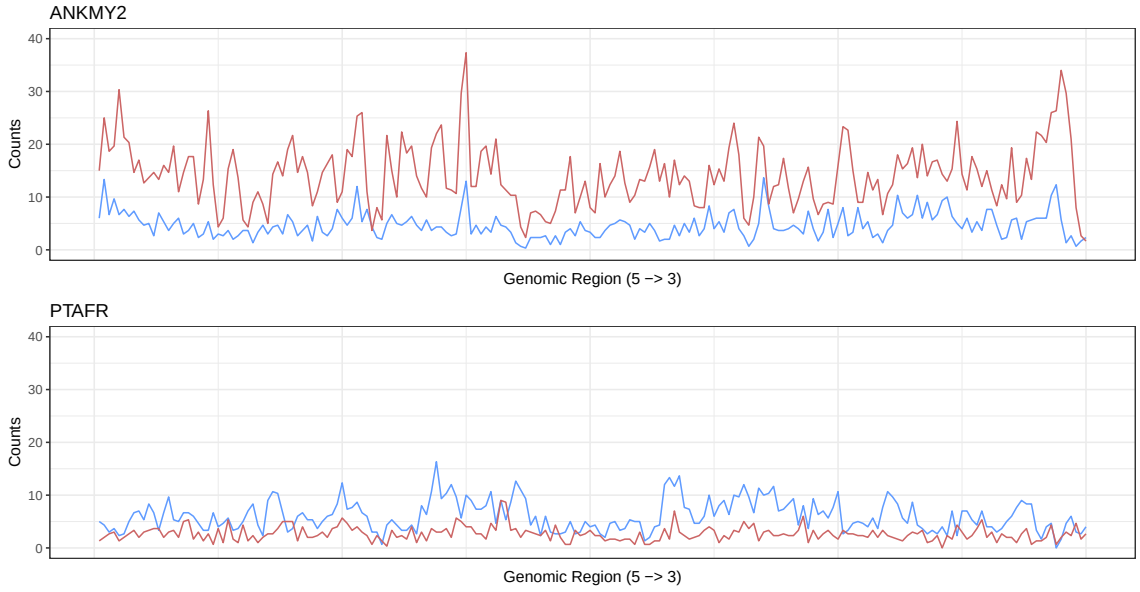


Figure B.9: Observed H3K27me3 signals over gene bodies for 4HT-(blue) and 4HT+(red) for ANKMY2 and PTAFR.

REFERENCES

Anders, Simon, and Wolfgang Huber. 2010. Differential expression analysis for sequence count data. *Genome Biology* 11(10):R106.

Ay, F., T. L. Bailey, and W. S. Noble. 2014a. Statistical confidence estimation for hi-c data reveals regulatory chromatin contacts. *Genome Research* 24(6):999–1011.

Ay, F., E. M. Bunnik, N. Varoquaux, S. M. Bol, J. Prudhomme, J.-P. Vert, W. S. Noble, and K. G. Le Roch. 2014b. Three-dimensional modeling of the p. falciparum genome during the erythrocytic cycle reveals a strong connection between genome architecture and gene expression. *Genome Research* 24(6):974–988.

Barrett, T., D. B. Troup, S. E. Wilhite, P. Ledoux, C. Evangelista, I. F. Kim, M. Tomashevsky, K. A. Marshall, K. H. Phillippy, P. M. Sherman, R. N. Muertter, M. Holko, O. Ayanbule, A. Yefanov, and A. Soboleva. 2010. NCBI GEO: archive for functional genomics data sets–10 years on. *Nucleic Acids Research* 39(Database):D1005–D1010.

Barski, Artem, Suresh Cuddapah, Kairong Cui, Tae-Young Roh, Dustin E. Schones, Zhibin Wang, Gang Wei, Iouri Chepelev, and Keji Zhao. 2007. High-resolution profiling of histone methylations in the human genome. *Cell* 129(4):823–837.

Benjamini, Yoav, and Yosef Hochberg. 1995. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)* 57(1):289–300.

Bieberstein, Nicole I., Fernando Carrillo Oesterreich, Korinna Straube, and Karla M. Neugebauer. 2012. First exon length controls active chromatin signatures and transcription. *Cell Reports* 2(1):62–68.

Chen, Li, Chi Wang, Zhaohui S. Qin, and Hao Wu. 2015. A novel statistical method for quantitative comparison of multiple ChIP-seq datasets. *Bioinformatics* 31(12): 1889–1896.

Chikina, Maria D., and Olga G. Troyanskaya. 2012. An effective statistical evaluation of ChIPseq dataset similarity. *Bioinformatics* 28(5):607–613.

Consortium, ENCODE Project, et al. 2012. An integrated encyclopedia of dna elements in the human genome. *Nature* 489(7414):57–74.

Cremona, Marzia A., Laura M. Sangalli, Simone Vantini, Gaetano I. Dellino, Pier Giuseppe Pelicci, Piercesare Secchi, and Laura Riva. 2015. Peak shape clustering reveals biological insights. *BMC Bioinformatics* 16(1).

Dekker, J. 2002. Capturing chromosome conformation. *Science* 295(5558):1306–1311.

Dixon, Jesse R., Inkyung Jung, Siddarth Selvaraj, Yin Shen, Jessica E. Antosiewicz-Bourget, Ah Young Lee, Zhen Ye, Audrey Kim, Nisha Rajagopal, Wei Xie, Yarui Diao, Jing Liang, Huimin Zhao, Victor V. Lobanenkov, Joseph R. Ecker, James A. Thomson, and Bing Ren. 2015. Chromatin architecture reorganization during stem cell differentiation. *Nature* 518(7539):331–336.

Djekidel, Mohamed Nadhir, Yang Chen, and Michael Q. Zhang. 2018. FIND: difFerential chromatin INteractions detection using a spatial poisson process. *Genome Research* 28(3):412–422.

Edelsbrunner, Herbert, and John Harer. 2009. *Computational topology*. American Mathematical Society.

Edelsbrunner, Herbert, John Harer, and Afra Zomorodian. 2001. Hierarchical morse complexes for piecewise linear 2-manifolds. In *Proceedings of the seventeenth annual symposium on computational geometry - SCG '01*. ACM Press.

Fan, Jianqing, and Sheng-Kuei Lin. 1998. Test of significance when data are curves. *Journal of the American Statistical Association* 93(443):1007.

Forcato, Mattia, Chiara Nicoletti, Koustav Pal, Carmen Maria Livi, Francesco Ferrari, and Silvio Bicciato. 2017. Comparison of computational methods for hi-c data analysis. *Nature Methods*.

Gerber, S, P Bremer, V Pascucci, and R Whitaker. 2010. Visual exploration of high dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics* 16(6):1271–1280.

Gerber, Samuel, and Kristin Potter. 2012. Data analysis with the morse-smale complex: ThemsrPackage forR. *Journal of Statistical Software* 50(2).

Gerber, Samuel, Oliver RÃ¼bel, Peer-Timo Bremer, Valerio Pascucci, and Ross T. Whitaker. 2012. Morse–smale regression. *Journal of Computational and Graphical Statistics* 22(1):193–214.

Good, Phillip I. 2004. *Permutation, parametric, and bootstrap tests of hypotheses (springer series in statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Heinz, Sven, Christopher Benner, Nathanael Spann, Eric Bertolino, Yin C. Lin, Peter Laslo, Jason X. Cheng, Cornelis Murre, Harinder Singh, and Christopher K. Glass. 2010. Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and b cell identities. *Molecular Cell* 38(4):576–589.

Henle, W., V. Diehl, G. Kohn, H. zur Hausen, and G. Henle. 1967. Herpes-type virus and chromosome marker in normal leukocytes after growth with irradiated burkitt cells. *Science* 157(3792):1064–1065.

Hower, Valerie, Steven N Evans, and Lior Pachter. 2011. Shape-based peak identification for ChIP-seq. *BMC Bioinformatics* 12(1):15.

Hsieh, Tsung-Han S., Assaf Weiner, Bryan Lajoie, Job Dekker, Nir Friedman, and Oliver J. Rando. 2015. Mapping nucleosome resolution chromosome folding in yeast by micro-c. *Cell* 162(1):108–119.

Imakaev, Maxim, Geoffrey Fudenberg, Rachel Patton McCord, Natalia Naumova, Anton Goloborodko, Bryan R Lajoie, Job Dekker, and Leonid A Mirny. 2012. Iterative correction of hi-c data reveals hallmarks of chromosome organization. *Nature Methods* 9(10):999–1003.

Kalchschmidt, Jens S., Adam C. T. Gillman, Kostas Paschos, Quentin Bazot, Bettina Kempkes, and Martin J. Allday. 2016. EBNA3c directs recruitment of RBPJ (CBF1) to chromatin during the process of gene repression in EBV infected b cells. *PLOS Pathogens* 12(1):e1005383.

Kharchenko, Peter V, Michael Y Tolstorukov, and Peter J Park. 2008. Design and analysis of ChIP-seq experiments for DNA-binding proteins. *Nature Biotechnology* 26(12):1351–1359.

Kieff, E., and A. B. Rickinson. 2007. Epstein-barr virus and its replication. *Fields Virology* 2603–2654.

Kuan, Pei Fen, Dongjun Chung, Guangjin Pan, James A. Thomson, Ron Stewart, and Sündüz Keleş. 2011. A statistical framework for the analysis of ChIP-seq data. *Journal of the American Statistical Association* 106(495):891–903.

Landau, William Michael, and Peng Liu. 2013. Dispersion estimation and its effect on test performance in RNA-seq data analysis: A simulation-based comparison of methods. *PLoS ONE* 8(12):e81415.

Langmead, B., C. Trapnell, M. Pop, and S. Salzberg. 2009. Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biology* 10(3):R25.

Leng, N., J. A. Dawson, J. A. Thomson, V. Ruotti, A. I. Rissman, B. M. G. Smits, J. D. Haag, M. N. Gould, R. M. Stewart, and C. Kendziorski. 2013. EBSeq: an empirical bayes hierarchical model for inference in RNA-seq experiments. *Bioinformatics* 29(16):2073–2073.

Li, Bo, and Colin N. Dewey. 2011. RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics* 12(1):323.

Li, Heng, and Richard Durbin. 2010. Fast and accurate long-read alignment with burrows–wheeler transform. *Bioinformatics* 26(5):589–595.

Lieberman-Aiden, E., N. L. van Berkum, L. Williams, M. Imakaev, T. Ragoczy, A. Telling, I. Amit, B. R. Lajoie, P. J. Sabo, M. O. Dorschner, R. Sandstrom, B. Bernstein, M. A. Bender, M. Groudine, A. Gnirke, J. Stamatoyannopoulos, L. A. Mirny, E. S. Lander, and J. Dekker. 2009. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* 326(5950): 289–293.

Liu, Bin, Jimmy Yi, Aishwarya SV, Xun Lan, Yilin Ma, Tim HM Huang, Gustavo Leone, and Victor X Jin. 2013. QChIPat: a quantitative method to identify distinct binding patterns for two biological ChIP-seq samples in different experimental conditions. *BMC Genomics* 14(Suppl 8):S3.

Love, Michael I, Wolfgang Huber, and Simon Anders. 2014. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology* 15(12).

Lun, Aaron T.L., and Gordon K. Smyth. 2015. diffHic: a bioconductor package to detect differential genomic interactions in hi-c data. *BMC Bioinformatics* 16(1).

Mahony, Shaun, Matthew D. Edwards, Esteban O. Mazzoni, Richard I. Sherwood, Akshay Kakumanu, Carolyn A. Morrison, Hynek Wichterle, and David K. Gifford. 2014. An integrated model of multiple-condition ChIP-seq data reveals predeterminants of cdx2 binding. *PLoS Computational Biology* 10(3):e1003501.

Maruo, Seiji, Bo Zhao, Eric Johannsen, Elliott Kieff, James Zou, and Kenzo Takada. 2011. Epstein-barr virus nuclear antigens 3c and 3a maintain lymphoblastoid cell growth by repressing p16ink4aand p14arfexpression. *Proceedings of the National Academy of Sciences* 108(5):1919–1924.

Mendoza-Parra, Marco-Antonio, Malgorzata Nowicka, Wouter Van Gool, and Hinrich Gronemeyer. 2013. Characterising ChIP-seq binding patterns by model-based peak shape deconvolution. *BMC Genomics* 14(1):834.

Milnor, John. 1963. *Morse theory. (AM-51)*. Princeton University Press.

Nostrand, E. L. Van, and S. K. Kim. 2013. Integrative analysis of c. elegans modENCODE ChIP-seq data sets to infer gene regulatory interactions. *Genome Research* 23(6):941–953.

O'Geen, Henriette, Lorigail Echipare, and Peggy J. Farnham. 2011. Using ChIP-seq technology to generate high-resolution profiles of histone modifications. In *Methods in molecular biology*, 265–286. Humana Press.

Ohashi, Makoto, Amy M. Holthaus, Michael A. Calderwood, Chiou-Yan Lai, Bryan Krastins, David Sarracino, and Eric Johannsen. 2015. The EBNA3 family of epstein-barr virus nuclear proteins associates with the USP46/USP12 deubiquitination complexes to regulate lymphoblastoid cell line growth. *PLOS Pathogens* 11(4): e1004822.

Paulsen, Jonas, Geir Kjetil Sandve, Sveinung Gundersen, Tonje G. Lien, Kai Trengereid, and Eivind Hovig. 2014. HiBrowse: multi-purpose statistical analysis of genome-wide chromatin 3d organization. *Bioinformatics* 30(11):1620–1622.

Pepke, Shirley, Barbara Wold, and Ali Mortazavi. 2009. Computation for ChIP-seq and RNA-seq studies. *Nature Methods* 6(11s):S22–S32.

Pope, J. H., M. K. Horne, and W. Scott. 1968. Transformation of foetal human leukocytesin vitro by filtrates of a human leukaemic cell line containing herpes-like virus. *International Journal of Cancer* 3(6):857–866.

R Core Team. 2018. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rao, Suhas S.P., Miriam H. Huntley, Neva C. Durand, Elena K. Stamenova, Ivan D. Bochkov, James T. Robinson, Adrian L. Sanborn, Ido Machol, Arina D. Omer, Eric S. Lander, and Erez Lieberman Aiden. 2014. A 3d map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell* 159(7):1665–1680.

Robertson, Gordon, Martin Hirst, Matthew Bainbridge, Misha Bilenky, Yongjun Zhao, Thomas Zeng, Ghia Euskirchen, Bridget Bernier, Richard Varhol, Allen

Delaney, Nina Thiessen, Obi L Griffith, Ann He, Marco Marra, Michael Snyder, and Steven Jones. 2007. Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. *Nature Methods* 4(8):651–657.

Robinson, M. D., D. J. McCarthy, and G. K. Smyth. 2009. edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 26(1):139–140.

Ross-Innes, Caryn S., Rory Stark, Andrew E. Teschendorff, Kelly A. Holmes, H. Raza Ali, Mark J. Dunning, Gordon D. Brown, Ondrej Gojis, Ian O. Ellis, Andrew R. Green, Simak Ali, Suet-Feung Chin, Carlo Palmieri, Carlos Caldas, and Jason S. Carroll. 2012. Differential oestrogen receptor binding is associated with clinical outcome in breast cancer. *Nature*.

Schmidt, D., M. D. Wilson, B. Ballester, P. C. Schwalie, G. D. Brown, A. Marshall, C. Kutter, S. Watt, C. P. Martinez-Jimenez, S. Mackay, I. Talianidis, P. Flicek, and D. T. Odom. 2010. Five-vertebrate ChIP-seq reveals the evolutionary dynamics of transcription factor binding. *Science* 328(5981):1036–1040.

Schweikert, Gabriele, Botond Cseke, Thomas Clouaire, Adrian Bird, and Guido Sanguinetti. 2013. MMDiff: quantitative testing for shape changes in ChIP-seq data sets. *BMC Genomics* 14(1):826.

Servant, Nicolas, Nelle Varoquaux, Bryan R. Lajoie, Eric Viara, Chong-Jian Chen, Jean-Philippe Vert, Edith Heard, Job Dekker, and Emmanuel Barillot. 2015. HiC-pro: an optimized and flexible pipeline for hi-c data processing. *Genome Biology* 16(1).

Shao, Zhen, Yijing Zhang, Guo-Cheng Yuan, Stuart H Orkin, and David J Waxman. 2012. MAnorm: a robust model for quantitative comparison of ChIP-seq data sets. *Genome Biology* 13(3):R16.

Shen, Li, Ning-Yi Shao, Xiaochuan Liu, Ian Maze, Jian Feng, and Eric J. Nestler. 2013. diffReps: Detecting differential chromatin modification sites from ChIP-seq data with biological replicates. *PLoS ONE* 8(6):e65598.

Shivashankar, Nithin, and Vijay Natarajan. 2012. Parallel computation of 3d morse-smale complexes. *Computer Graphics Forum* 31(3pt1):965–974.

Skalska, Lenka, Robert E. White, Gillian A. Parker, Alison J. Sinclair, Kostas Paschos, and Martin J. Allday. 2013. Induction of p16ink4a is the major barrier to proliferation when epstein-barr virus (EBV) transforms primary b cells into lymphoblastoid cell lines. *PLoS Pathogens* 9(2):e1003187.

Stansfield, John, and Mikhail G. Dozmorov. 2017. Hicdiff: A method for joint normalization of hi-c datasets and differential chromatin interaction detection. *bioRxiv*. `http://www.biorxiv.org/content/early/2017/06/08/147850.full.pdf`.

Start, R., and 0 G. Brown. 2011. DiffBind: differential binding analysis of ChIP-Seq peak data. Bioconductor package: `http://bioconductor.org/packages/release/bioc/html/DiffBind.html`.

Steinhauser, Sebastian, Nils Kurzawa, Roland Eils, and Carl Herrmann. 2016. A comprehensive comparison of tools for differential ChIP-seq analysis. *Briefings in Bioinformatics* bbv110.

Tierny, Julien, Guillaume Favelier, Joshua A. Levine, Charles Gueunet, and Michael Michaux. 2017. The Topology ToolKit. *IEEE Transactions on Visualization and Computer Graphics (Proc. of IEEE VIS)*. `https://topology-tool-kit.github.io/`.

Wang, Junbai, Xun Lan, Pei-Yin Hsu, Hang-Kai Hsu, Kun Huang, Jeffrey Parvin, Tim H-M Huang, and Victor X Jin. 2013. Genome-wide analysis uncovers high frequency, strong differential chromosomal interactions and their associated epigenetic patterns in e2-mediated gene regulation. *BMC Genomics* 14(1):70.

Wang, Zhibin, Chongzhi Zang, Jeffrey A Rosenfeld, Dustin E Schones, Artem Barski, Suresh Cuddapah, Kairong Cui, Tae-Young Roh, Weiqun Peng, Michael Q

Zhang, and Keji Zhao. 2008. Combinatorial patterns of histone acetylations and methylations in the human genome. *Nature Genetics* 40(7):897–903.

Wu, Hao, and Hongkai Ji. 2014. PolyaPeak: Detecting transcription factor binding sites from ChIP-seq using peak shape information. *PLoS ONE* 9(3):e89694.

Xu, Zheng, Guosheng Zhang, Fulai Jin, Mengjie Chen, Terrence S. Furey, Patrick F. Sullivan, Zhaohui Qin, Ming Hu, and Yun Li. 2015. A hidden markov random field-based bayesian method for the detection of long-range chromosomal interactions in hi-c data. *Bioinformatics* 32(5):650–656.

Yekutieli, Daniel. 2008. Hierarchical false discovery rate–controlling methodology. *Journal of the American Statistical Association* 103(481):309–316.

Zhang, Yong, Tao Liu, Clifford A Meyer, Jérôme Eeckhoute, David S Johnson, Bradley E Bernstein, Chad Nussbaum, Richard M Myers, Myles Brown, Wei Li, and X Shirley Liu. 2008. Model-based analysis of ChIP-seq (MACS). *Genome Biology* 9(9):R137.