Stony Brook University

Genome-wide Analysis of Chromatin Binding Proteins in D. melanogaster and C. elegans

A Dissertation Presented

By

Xin Feng

To

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

In

Biomedical Engineering

Stony Brook University

May 2011

Stony Brook University

The Graduate School

Xin Feng

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

Lincoln Stein, M.D., Ph.D. - Dissertation Advisor
Director and Senior Principal Investigator, Informatics and Biocomputing,
Ontario Institute for Cancer Research, Canada
Associate Professor, Department of Biomedical Engineering, Stony Brook University

Wei Lin, Ph.D. - Chairperson of Defense
Research Assistant Professor, Department of Biomedical Engineering, Stony Brook University

Michael Zhang, Ph.D.
Cecil H. and Ida Green Distinguished Chair of Systems Biology Science
Director, Center for Systems Biology, Department of Molecular and Cell Biology,
The University of Texas at Dallas
Associate Professor, Department of Biomedical Engineering, Stony Brook University

Zhiping Weng, Ph.D.
Director and Professor, Program in Bioinformatics and Integrative Biology,
University of Massachusetts Medical School

This dissertation is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

Genome-wide Analysis of Chromatin Binding Proteins in D. melanogaster and C. elegans

By

Xin Feng

Doctor of Philosophy

In

Biomedical Engineering

Stony Brook University

2011

The mechanisms of regulating the translation of information encoded in DNA into gene expression have been intensively investigated since last century. A large portion of the efforts concentrate on characterizing the proteins that bind to specific chromatin or DNA regions. These proteins play important roles in the regulating hierarchy. Until the beginning of the 21st century, studies probing these chromatin binding proteins are generally conducted at the scale of a single gene or a limited region of the whole genome. The recent advancement in next-generation sequencing has provided a revolutionary method named as ChIP-seq that accurately generates genome-wide profiles of chromatin binding proteins. The modENCODE project has generated genome wide protein binding sites for a large number of chromatin binding proteins of model organisms D.melanogaster and C.elegans. It is thus possible to investigate the spatial distribution of these proteins at the genome-scale. To achieve this goal, an algorithm is needed to find protein binding sites across the genome. Although many existing algorithms suffice the basic need, none of them can resolve binding sites that stay closely to each other and does not sacrifice other desired properties such as specificity of the algorithm.

In this thesis, I present my work in designing a ChIP-seq peak calling algorithm called PeakRanger which addresses the above-mentioned concerns. PeakRanger, along with other accessory computing programs are used to analyze the datasets generated by the modENCODE project. With these tools, genome-wide binding sites of a large selection of chromatin binding proteins are generated for both D.melanogaster and C.elegans. The distributions of D.melanogaster insulator binding proteins were analyzed in details, showing their global correlation with gene expression regulation. The properties of binding sites that stay closely to each other are also characterized, which is the first report of doublet binding sites of D.melanogaster. It is shown that doublet binding sites are preferred regions for histone markers of promoters.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

TSS: Transcription Start Site
IBP: Insulator Binding Proteins
ChIP-Chip: Chromatin immunoprecipitation (ChIP) combined with microarray (Chip)
ChIP-Seq: Chromatin immunoprecipitation (ChIP) combined with sequencing technology (seq)
ENCODE: The ENCyclopedia Of DNA Elements project
modENCODE: The Model organism ENCyclopedia Of DNA Elements project

Chapter 1 Introduction

1.1 The chromatin biology

To transcribe and translate the genetic information properly, a regulatory system is built-in for every living organism. In eukaryotes such as human, fly and worm, a major portion of this regulatory system is consisted of the proteins that interact with DNA. Histones and transcription factors are the two main classes of proteins having this regulatory role. In particular, histones are involved in regulation of the chromatin structure. And transcription initiation requires a suitable chromatin structure. Transcription factors control the initiation of transcription and they decide the levels of transcription. Histones and transcription factors also interact, adding an additional level of regulation.

1.1.1 DNA is packed into chromatin

DNA is usually packed into a much compact structure called chromatin[1]. There are both structural and functional reasons for this tight structure. By folding a long DNA sequence into chromatin, the space required to store the same amount of information is much less. It is estimated that the compression ratio can be 10,000[2]. The tight three-dimensional (3D) arrangement of chromatin also makes it easier for cis-regulatory elements, which are functional DNA sequence segments, to interact with each other.

Nucleosome is the basic building unit for eukaryotes chromatin[3]. A nucleosome is a protein-DNA complex that organizes a DNA strand into a group of smaller pieces of sequences. Each nucleosome wraps about two rounds of DNA sequences with an appropriate length of 146 base pairs[3]. The proteins involved in nucleosome building are generally referred as histones. Histone H2.A, H2.B, H3 and H4 form the central complex of nucleosome. With the help of histone H1 and other auxiliary proteins, nucleosomes are then further packed to form the chromosome.

1.1.2 The two environments of chromatin: euchromatin and heterochromatin

Two distinct types of chromatin exist in the cells: the euchromatin and the heterochromatin [4-6]. Euchromatin is less packed in structure, making it easier to access the genes. Euchromatin is thus associated with active gene transcription and is commonly referred as “open” chromatin[4]. On the contrary, heterochromatin is tightly condensed, making access to genes difficult[5]. Thus heterochromatin is considered “closed” regions. There are two types of heterochromatin: constitutive and facultative[7]. Constitutive heterochromatin remains packed all the time, which is usually found around highly repetitive DNA regions and centromere[8, 9]. Facultative heterochromatin is instead dynamic and can be unpacked as response to regulatory signals[7]. An important feature of heterochromatin is its ability to spread genomic regions and repress genes it encountered, causing sequence-independent repression[10]. A built-in mechanism to control the spreading of heterochromatin is through insulators and the proteins that bind to them, which is discussed later.

1.1.3 Histone modifications establish chromatin environments and regulate transcriptions

The partitioning of chromatin into two different environments is the functional consequence of histone modifications[3, 11]. Histone modification refers the changes to the amino-acid residues of histones[3, 11]. Commonly seen modifications include but are not limited to methylation, acetylation and de-acetylation, phosphorylation and ubiquitination[11]. Among them, methylation and acetylation are now being intensively studied. Histone modifications happen to both canonical histones and histone variants. Increasing evidences show that the modifications to histones regulate the interactions of histones to other proteins and thus regulate the structure of chromatin[3]. These modifications are selectively enriched in both the euchromatin and heterochromatin domains of chromatins and thus regulate the transcription of genes[3, 11]. The ability of histone modification to alter the structure of chromatin includes two aspects: the interruption of the interaction between DNA and histones; And the recruitment of additional proteins which confers further effects on chromatin structure[11]. Acetylation is considered to have the most potential to interrupt the interactions between DNA and histones since it neutralizes the charges of lysines[12]. A number of proteins are found to be recruited by a specific type of modified residues: The polycomb family protein Pc2 of the polycomb-repressive-complex 1 (PRC1) is recruited by tri-methylation of H3K27 (H3K27Me3), and then other subunits of PRC1 will contribute to the ubiquitination of H2A tails to fully repress the genes[13].

Methylation is the process of adding a methyl group into the long tails of histone amino-acid residues[11]. The phenomena of histone methylation was found in mid-20th century and histone methylation has been found to associate with transcription in 1999[14, 15]. Lysines of histones are shown to be frequently marked by methylation[2]. It is also discovered that the 4th, 9th and 27th lysine of core histone H3 and the 20th lysine of H4 are common places for histone methylation[2, 11]. Unlike the acetylation process, a single histone can be mono-methylated, as well as di- and tri-methylated, thus adding an additional regulatory mechanism.[11]

Activation	H3K4Me1	H3K4Me3	H3K9Me1	H3K9Ac
Repression	H3K9Me2	H3K9Me3	H3K27Me3	

Table 1-1 Typical histone modifications and their functions

Different degrees of histone methylation at the 4th lysine have been shown to be concrete markers of euchromatin[3, 16-18]. And the methylation of the 27th lysine proves to be associated with hetrochromatin[3, 16-18]. These methylation marks co-exist with many other histone modification marks and together they mark various cis-regulatory elements. One recent study has determined the correlation between the enhancers and mono- and di-methylation of the 4th lysine of histone H3 (H3K4Me1 and H3K4Me2)[18]. In the reported study, the binding sites of p300/CBP, a family of histone acetyl-transferase (HAT, discussed below), are found to co-localize with H3K4Me1, H3K4Me2 and the acetylation of the 27th lysine of histone H3 (H3K27Ac). Since p300/CBP possesses the role of HAT[19], it is considered that they selectively bind to putative enhancers. The co-existing marks of various histone methylations thus indicate that they might be predictive marks for enhancers and thus actively transcribed genes. The relationship between histone methylation and enhancers is further supported in

another study which uses DNase I hypersensitivity assays to predict enhancers. The study confirms the link between H3K4Me1, H3K4Me2 and enhancers and further argues that the mono-methylation of the 9th lysine of histone H3 (H3K9Me1) is also linked to enhancer [16, 20]. Connection between histone methylation and promoters is also supported[18]. In particular, H3K4Me3 has been shown to associate with active promoters[18].

Acetylation adds an acetyl group into the lysine residue of histone tails [3, 16-18]. Acetylation has been correlated to gene transcription and is among the most studied topic of histone modifications[21]. The major effect of acetylation is believed to be the removal of the electronic charge of the amid-acid tails which results a loose interaction of histones to DNA, thus granting accessibility to the packed DNA sequence.

The process of acetylation is carried out by histone acetyl-transferase (HAT). The evidence of the link between acetylation and gene transcription is given by the HAT property of Gcn5, a determined transcription regulator conserved in both human and yeast[22]. The GNAT family, consisted of variants of Gcn5 and other related transcription factors, has been shown to associate with different stages of transcription[22]. Another HAT family that is much more famous than GNAT is the p300 and CBP family. p300/CBP has been shown to localize with active enhancers in both human and D.melanogaster [18, 23]. The role of p300/CBP as HAT and their links with enhancers support that acetylation and HAT is deeply involved in gene regulation. The acetylation of the 14th and 18th lysine of histone H3 (H3K14Ac and H3K18Ac) is shown to be present at a set of enhancers in $CD4^+$ cell lines[24, 25]. It is also shown that H3K27Ac involves in cell differentiation. A study in human embryonic stem cells (ESC) indicates that H3K27Ac and H3K4Me1 together mark active promoters but inactive promoters are instead marked by H3K27Me3 and H3K4Me1[26, 27]. In the stage of differentiation, some of these inactive genes are turned on through a process in which H3K27Me3 is replaced with H3K27Ac.

In contrary to acetylation and HAT, Deacetylation and histone deacetylases (HDAC) is thought to serve as a layer to repress the activity of transcription[2]. A major mechanism through which the repression is conferred is the formation of heterochromatin after deacetylation of histones[28]. And like HAT, HDAC plays important roles in this process. Although it is known that the formation of condensed chromatin can be disrupted if the activity of HDAC is inhibited[28], it is still not clear how the instructions of deacytlating a segment of chromatin is given.

Similar with HAT, HDAC is a large family of proteins. In D.melanogaster, five HDACs have been identified: HDAC3,HDAC6, HDACX(also known as HDAC11) which has been shown to correlate with promoters, and HDAC1(also known as Rpd3) and HDAC4a, which has been found to associate with heterochromatin, as well as promoters[29].

Other than the methylation, acetylation and deacetylation described earlier, other types of histone modifications also exist. This includes histone phosphorylation[30] and ubiquitination[31]. The knowledge about these histone modifications is relatively limited.

1.2 The role of transcription factors in regulating transcription

Transcription initiation requires an open chromatin so that various cis-regulatory elements such as promoters can be accessed[11]. Histone modification plays key roles in maintaining the open chromatin[2, 11]. The binding of PolII is initiated by binding of other proteins to the promoters of genes[32]. These proteins, along with other ones that bind to specific sequences in promoters are named transcription factors. Promoters can be conceptually divided into two parts:

core promoters and other functional elements[32]. The most examined core promoter is TATA box[33], which is about 30bp upstream of transcription start site (TSS). At the time of transcription initiation, TATA box is bound by TATA box binding protein (TBP), which is part of transcription factor II D (TFIID). Following the binding of TFIID, TFIIB, TFIIE, TFIIF, TFIIH and PolII is recruited to form the pre-initiation complex (PIC)[32]. Binding of PolII to promoters occur ubiquitously across the genome and the basal level of transcription is very low[32]. To facilitate the level of transcription, other transcription factors referred as transcription activators are needed[32]. Proteins bind to proximity of promoters to support the transcription machinery of PolII complex[34]. Proteins also bind to enhancers to further boost the transcription activity of genes[34].

1.2.1 Regulation through binding to enhancers

Enhancers are another class of cis-regulatory elements of the genome[35]. Enhancers regulate their target promoters to control gene expressions[35]. Enhancers contain binding sites for transcription factors which regulate the target genes[35]. Enhancers are distal to promoters. It has been reported that enhancers can even act on promoters not on the same chromosome[36]. There may be two explanations for the distal effect of enhancers. Since DNA usually exists as a compact 3D structure-chromatin, enhancers located in different chromosomes may actually be located near to the target promoters[37]. Enhancers may also help forming loop structures in chromatin so that they can reach promoters hundreds of kilo base pairs away[35, 36].

A major role of enhancer binding proteins is to recruit enzymes and other chromatin remodelers to change the histones and thus the structures of chromatin to make the DNA accessible to other regulators[35]. A recent study conducted in prostate cancer cell lines discovered a set of enhancers that respond to androgen stimulation[17]. The discovered enhancers are initially free of androgen receptor and FOXA1 binding. The sustained stimulation displaced the nucleosome at the enhancer and its location is later bound by FOXA1. It is found that the characteristic feature of these enhancers is the flanking histone modifications H3K4Me2 and histone variants H2A.Z. This feature is then used to predict enhancers with similar properties and generated a comprehensive identification of these enhancers.

A famous protein that binds to enhancer is CBP/p300[18]. CBP/p300 possesses the property of histone acetylation transferase and its presence helps maintain the open structure of chromatin.

It is still not clear how enhancers find the target promoters[37, 38]. Once the connection is established between enhancer and promoter, the two regulatory elements interact through the proteins that bind to them[38].

1.2.2 Insulators and insulator binding proteins

Since enhancers control promoters in a way independent of the distance and orientation, other regulatory mechanism must exist so that gene expression maintains precision. One such control is through insulator, which binds to a class of transcription factors called insulator binding proteins (IBP). A number of IBP has been identified. In D.melanogaster, five IBPs have been identified: CCCTC-binding factor (CTCF), Suppressor of hairy wing (Su(Hw)), CP190, BEAF and Mod(mdg4)[37]. Among these five IBPs, CTCF is the only IBP that has homologue in human [39], which makes it of particular interest. In fact, CTCF has been implied to have multiple roles[40].

Insulator and the proteins binding to them perform two basic functions: barriers to stop the spreading of heterochromatin and enhancer-blockers to prevent irrelevant regulation of enhancers to promoters[37].

1.2.2.1 The barrier role of insulators

This set of insulators was originally identified at the boundaries between euchromatin and heterochromatin[41]. In the β-globin locus of chicken cells, the spread of adjacent hetrochromatin domain is prevented by the HS4 insulator, which is bound by CTCF[41]. In D.melanogaster, CTCF is also found to have the role as a barrier[41].

1.2.2.2 The enhancer-blocker role of insulators

Many enhancer-blockers have been identified in D.melanogaster[37]. The decoy model is proposed to explain the way enhancer-blockers work[37]. Enhancer may interact with the enhancer-blocker which serves as the decoy. This decoy interaction may thus prevent the unwanted interactions between enhancers and promoters. Another model to explain the effects of enhancer-blocker is that insulators collaborate with each other and other structures of the chromatin to form loop structures[37]. Enhancers thus can only interact with promoters within the same loop.

1.3 Comprehensive catalogue of histone modifications and transcription factor binding sites

As discussed earlier, histones and transcription factors are conserved regulators of gene transcriptions. The close collaboration between them, along with other regulatory systems guarantees the perfect amount of proteins is produced with a perfect timing. It is thus expected that malfunctioning of these two important classes of proteins may cause severe human diseases. It has been shown that more than one hundred of transcription factors are directly linked to diseases[42]. Given that around 2600 transcription factors have been found in human cells[43], the number of disease-responsible transcription factors will undoubtedly increase. On the side of histones, it has been recently found histone methylation transferase MLL2 mutations cause Kabuki syndrome[44].

Given their indispensable roles, it is important to figure out how transcription factors and histone modifications choose the binding sites or enriched regions. Until the beginning of 21th century, the studies on transcription factors and histone modifications are generally conducted at the level of genes or a very limited genomic region. These researches prove fruitful. However, many of the previously obtained conclusions are subject to questioning when put into the context of the whole genome. One such example is the determination of the structure of core promoter. Classically, TATA-box is considered the only core element that is capable of transcription initiation, however, recent analysis has indicated that a large portion of promoters in human, D.melanogaster and A.thaliana are actually free of TATA box[45-49].

It has been found that the typical range of the number of binding sites for a given transcription factor is up to ten thousands in the genome[38]. And it is also found that almost every transcription factor binds to the promoter region of genes[38]. An immediate question to ask is how the specificity of transcription regulation is achieved if transcription factors bind essentially to every possible place. Two hypotheses exist for this question: through the

combinatorial binding of a family of transcription factors or through the combinatorial marks of different histone modifications or the mixture of them both.

To test the two hypotheses, the comprehensive map of transcription factor binding and histone modifications are needed. Although project such as ENCODE[50] and modENCODE[51] is generating parts of this global map, individual experiment may need more customized profiles for a set of factors-of-interest. Once the dataset is comprehensive enough, one can ask if any given combination of transcription factors and histone marks can reliably predict the outcome of treatment (gene down-regulation or up-regulation). Recent technology advancement in sequencing greatly helps the identification of the comprehensive catalogue of these genome-wide profiles for transcription factors and histone modifications.

1.3.1 ChIP-Chip and ChIP-Seq

Chromatin immunoprecipitation (ChIP) is a technique that selectively extracts proteins of interest and the sequences bound by them from cells[52]. A typical workflow of ChIP starts with cross-linking proteins to DNA in a cell lysate. The chromatin is then sheared into a cohort of pieces. A specific antibody is then applied to precipitate the proteins of interest. The immunoprecipitated complex of proteins and their cross-linked DNA then undergoes a process to remove the proteins, leaving only pure DNA sequences bound by the proteins of interest. The purified DNA sequences can be finally determined by various techniques such as PCR, microarray hybridization (ChIP-Chip) and next-generation sequencing (ChIP-Seq).

ChIP-Chip is the combination of ChIP and microarray (Chip)[53]. A microarray contains a large number DNA segments which are identical to the reference genome of the species of interest. By hybridizing the ChIPed DNA sequences to the microarray, the binding sites of proteins can be decided.

Instead of hybridizing the purified DNA to a set of fixed reference sequences in chips, these DNA of interest can also be directly sequenced using the high-throughput next-generation sequencing technology (seq)[54, 55]. Compared to ChIP-Chip, ChIP-Seq has much better resolution[54]. ChIP-Chip suffers from the noise during hybridization to the microarray[54]. ChIP-Chip also relies on the tiled sequences of the microarray, limiting its genome coverage[54]. The major dis-advantage of ChIP-Seq is its high cost[54]. However, the cost is decreasing rapidly with higher through-put of newer sequencing platforms and technologies[54]. Besides, the cost can also be tuned by trading-off the sensitivity. With lower sequenced reads, the cost is lower at the cost of lower sensitivity due to lower genome coverage. Another limitation of ChIP-Seq is the availability of quality antibodies to the proteins of interest[54]. In case a low quality antibody is used, the proteins immunoprecipitated could be only fractions of the actual binding proteins.

A typical ChIP-Seq experiment usually consists of two sets of sub-experiment: the sequencing of the ChIPed regions and the sequencing of the genomic background as the control. The utility of the control experiment is to remove the noise generated during the shearing process. Since portions of DNA are packed with histones, open chromatin regions are likely to be fragmented and result higher sequenced reads. Repetitive regions of the genome are also likely to be enriched due to the incomplete number of repeat pieces in the reference genome[54]. It is thus possible that the dataset generated by the pre-immunoprecipitation also contains peaks that resemble a true binding site. By comparing the ChIP-Seq dataset with the control dataset, computer software could better recognize suspicious binding sites and avoid false positives.

ChIP-Seq can be used to identify the binding sites of transcription factors and also the enriched regions for histone modifications. The major difference among these two types of experiments is the antibody used to perform the ChIP process.

Due to the superior performance of ChIP-Seq and its ability to profile the genome-scale localization of transcription binding sites, it is now the standard tool to probe the transcription factor regulatory system. Another important tool that directly measures the levels of transcriptions is RNA-Seq[55]. RNA-Seq can indirectly sequence the sequence of transcripts and thus gives the genome-wide transcription levels.

1.3.2 The typical data processing flow of ChIP-Seq data

The processing flow of ChIP-Seq data can be divided into four parts: base-calling, reads-alignment, peak-calling and downstream integrative analysis. Each part of the processing flow involves a separate set of algorithms and software which differ significantly in design. Data processing before the part of integrative analysis is now relatively routine but the integrative analysis remains flexible with no routines to follow.

1.3.2.1 Raw reads alignment using aligners

The output of base callers is a nucleotide sequence, the raw read. A single ChIP-Seq experiment usually generates millions of raw reads. Raw reads must first be annotated with the exact location and strand orientation relative to the reference genome. This stage is thus referred as raw reads alignment.

The basic idea shared by all aligners is to look up a raw read in the dictionary, which is the reference genome. Based on how the dictionary is represented in these aligners, they can be classified into two generations. The first generation aligners are usually referred as hash-table-based aligners, because they all use hash table to store the information of the reference genome. A hash table is a data structure used to represent the one-to-one or one-to-many relationship. When aligner processes a read, it compares the fed raw reads against the hash table and searches for any match. The raw read is transformed to a key using an aligner-specific hashing function, which takes time to finish. Once the key is generated, the match can be found immediately if such a match exists. Aligners falling into this category includes: Eland which is shipped with the Illumina sequencer, ZOOM[56], MAQ[57], SOAP[58], RMAP[59], and SHRiMP[60].

The second generation aligners use burrows-wheeler transform (BWT) to build the look-up table in more space efficient way. BWT was originally developed for text compression. The major reason that BWT is preferred over hash table is its much lower memory foot-print. A human reference genome needs only 2.7Gb to store when transformed using BWT and compressed accordingly. Among this category of aligners are Bowtie[61], BWA[62] and SOAPV2[63]. Bowtie and BWA use essentially the same BWT indexing algorithms, varying mainly in how the index is searched.

BWT based aligners are usually preferred to the hash table based aligners in the application of ChIP-Seq. They are generally much faster than their processors. Bowtie is up to hundreds of times faster than SOAP and thirty-five times faster than MAQ[61]. BWT based aligners also have significant advantages in memory consumptions, primarily due to the benefit of using BWT to index the reference genome. Bowtie, however, suffers from loss of sensitivities compared to MAQ. MAQ is thus preferred in the applications of single nucleotide polymorphism (SNP)

finding. For ChIP-Seq, with increased sequencing depth, Bowtie is good for most datasets given its fastest running speed.

1.3.2.2 Finding the enriched regions of ChIP-Seq dataset with peak callers

With aligned reads generated by raw reads aligners, the next step is to find the enriched regions, or peaks of the dataset. The peaks should correspond to the actual binding locations of transcription factors, or the specifically targeted genomic regions by other proteins such as histones or PolII. More and more ChIP-Seq experiments now also include an additional control experiment in which the genomic background is also sequenced. An algorithm should be able to find in the treatment dataset the enriched regions of aligned reads, relative the same region in the control dataset.

Before peak calling, a genome-scale profile of the aligned reads is constructed. Since reads are from both the positive and negative strands, each of the two strands can have an individual profile. The two sets of profiles can then be combined to get a single profile by shifting the profiles of each strand with the length estimated from the dataset[64]. Another way to build the profile is to extend the aligned reads with a fixed length close to the size of the fragment of the library and then combine the reads on both strands to obtain a single profile[65]. The shifting method is expected to be more accurate in terms of its ability to identify the precise binding sites of transcription factors since it utilizes the information on both strands. However, the estimating of the shift size remains a challenging task. The benefits of shifting method will also degrade when the actual enriched regions span over a large segment due to the complexities of topologies of the region. In contrast, the extending method is faster since it does not estimate the shift length but suffers from the relatively less accuracies.

With the genome-wide reads profile, a straight forward way to find the enriched regions is to measure the enrichment ratio of the reads in the treatment over the reads in the control. Both the ratio and the absolute reads should be considered when evaluating the significance of enrichment. A pair of region with 10 and 1 reads respectively is not as significant as the pair of 100 and 10. A realistic way to model the enrichment is through the binomial distribution. In particular, the reads of the candidate region in the treatment dataset can be modeled as the total number of trials and reads in the control as the number of successes. The significance of the enrichment can be measure as the probability of observing equal or less number of control reads. With this binomial distribution model, a paired reads of 10 and 1 obtains the significance of $1e^{-2}$ while the pair of 100 and 10 gets $1e^{-17}$. Other statistical distributions can also be used. A modified Poisson distribution is used to model the regional enrichment in ChIP-Seq dataset by peak caller MACS[66]. Another strategy is to measure the similarity of reads profile between the positive and negative strands. It is observed that the distribution of aligned reads around the binding sites show a bi-modal distribution in the positive and negative strands[64]. By calculating the correlation between the two strands, the binding sites can be found by looking for the position of the local maximum of correlations. Both the enrichment ratio and the correlation maximum method work well in identifying the binding sites of transcription factors. However, the correlation method is not appropriate when the enriched regions are broadly stretched. Histone modification datasets are usually of such character and cannot be processed in this way. In fact, it does not make sense to find the sharp binding sites in these datasets. The similar case also happened to PolII datasets where broad and sharp enriched regions interleave with each other.

An important aspect of a peak caller is the way it determines the significance of the discovered regions. Ideally, a peak caller should be able to assess the likelihood that the discovered enriched regions represent true binding sites or histone modification marks. However, current peak callers only assign a value of statistical significance, usually in the form of p value. There have been relatively few reports on how the two different kinds of measures agree. Fortunately, p values are so far working well. In addition to the assignment of p values, false discovery rate (FDR) serves as an additional control of the quality of called peaks. In statistical distribution based methods, FDR estimation is usually based on the Bonferroni correction, which controls the added errors when multiple statistical tests are considered simultaneously. Due to the uncertainty of connection between biological significance and statistical significance, the performance evaluation of peak callers is now only empirical. Two major benchmarks for peak callers are its ability to discover biologically validated enriched regions and the distance between the called peaks and the occurrence of motifs. qPCR validation can be performed for a set of binding sites and test the percentage of qPCR validated sites out of the total called peaks[54]. For transcription factors with motifs known in prior, one can search the sequences of the binding sites and measure the distance between the center of binding sites and the center of the motifs.

Implementation of peak callers is also of great importance. ChIP-Seq experiments generate datasets at the scale of gigabytes and thus challenge the efficiency of a peak caller. The implementation should be a good compromise of memory consumption and running speed. Given that most workstation computers now have powerful central processing unit (CPU) with multiple processing cores, a peak caller should be able to run in parallel mode to fully utilize the powerful CPUs. Unfortunately, only a few existing peak callers support parallel processing. Another issue related the large size of input files is the efficiency of data loading module of the peak caller. The speed of transferring data from computer hard drives to memories takes much more time than from memories to CPUs. It is estimated that the data loading costs more than 50% of the total running time. To solve this problem, the C or C-plus-plus (C++) language is preferred over scripting languages such as R. C/C++ has proven record of excellent performance over many other programming languages, at the price of more difficulties in maintaining the source codes.

1.3.2.3 Downstream analysis after peak calling

The most challenging part of interpreting the ChIP-Seq results usually happens after the peak calling step. To make sense of the peak calling results, the first step usually involves direct visualization of the called peaks aligned to the reads profile and genome annotations. This step serves as a naive but effective way to assess the overall quality of called peaks. By checking the localization of called peaks with annotated genomic regions such as transcription start sites, the quality of called peaks can be roughly estimated. To further analyze the dataset, multiple extra datasets are needed. It is a common practice to overlay genome-wide profiles of a set of transcription factors and histone modifications and analyze the overlapped regions. An aggregating profile plot can be generated by averaging the reads count of one ChIP-Seq dataset over a set of specific regions in another. For example, the aggregating profile of a transcription factor over annotated promoters will show a peak in the center of promoters if it preferably binds to promoters. Motif-discovery analysis can also be conducted using the sequences of discovered binding sites. A motif is a recurring pattern of DNA sequence that directs the binding of proteins. Established motif-finder such as MEME[67] can be applied to search for a consensus sequence

out of a group of sequences extracted from the called peaks. A motif-scanner such as MAST[68] is then able to scan the whole genome for occurrences of this consensus motif. Another useful analysis is to search for differentially regulated genes. This type of analysis usually requires additional information of genome-wide transcription profiles. A gene with upstream binding sites can be compared with the one in the control for any changes of transcription levels. Other types of analysis are also possible and the key to success is close collaboration between bioinformaticians and wet-lab researchers.

1.4 The modENCODE project

As shown by the discussions presented earlier in this introduction, fully decoding the genome is requires genome-wide knowledge about transcription factors, histones and many other proteins which together regulate the activities of genes by interacting with various functional elements of DNA. This global regulatory map is still missing.

To facilitate the discovery of the regulatory map in human, the National Human Genome Research Institute (NHGRI) launched the Encyclopedia of DNA Elements (ENCODE) project, aiming at determining the regulatory DNA elements in the 1% of human genome[50]. The pilot phase of ENCODE finds that the complexity of human genome is far more than what was expected. In 2007, as an expansion to the original ENCODE project, the model organism ENCODE (modENCODE) project was initiated with a similar aim to annotate the functional elements in D.melanogaster and C.elegans[51]. Both of the two model organisms have been intensively researched before the modENCODE project. A large number of research tools and systems are developed and for the study of biological processes that resemble those in human. Besides, the most immediate benefit is that the size of the genomes of both D.melanogaster and C.elegans is only thirtieth of human, thus reducing costs significantly.

1.4.1 Probing the binding profiles of histones and transcription factors

An immediate goal of the modENCODE project is to generate the genome-scale binding sites for a vast number of transcription factors as well as histone modification marks in both D.melanogaster and C.elegans. The primary technology used for C.elegans is ChIP-Seq and both ChIP-Chip and ChIP-Seq will be used for D.melanogaster[51]. Multiple cell lines or developmental stages will be investigated, adding an extra dimension of the datasets.

1.4.2 Data management and the Data Coordinating Center (DCC)

The data generated by the modENCODE project is beyond the management capacity of any individual laboratories participated in the consortium. To effectively manage these datasets, the Data Coordinating Center (DCC), led by Dr. Lincoln Stein, is established. Both the raw and processed datasets are accessible through the FTP services provided by DCC[51]. Various facilities are also provided to help data visualization and integrative analysis.

The processing and storage of most ChIP-Seq datasets happens at cluster led by Dr. Robert Grossman at University of Illinois at Chicago (UIC).

1.5 Unsolved problems

1.5.1 The genome wide distribution of insulator binding proteins

Although insulators and insulator binding proteins play important roles in the regulation hierarchy, the studies are usually limited to a certain classical loci. The modENCODE project has generated the genome scale binding sites of a number of insulator binding proteins in D.melanogaster, which provides a great opportunity to investigate the spatial distributions of these proteins. It is also interesting to analyze the potential relationship of IBP and genes near to the binding sites. In chapter 2, I present my work in analyzing the insulator binding proteins of D.melanogaster, with focus on their genome-scale distributions and co-localizations with different kinds of chromatin. The relationship between insulator binding proteins and their nearby genes are also analyzed.

1.5.2 Finding the interactions of transcription factor binding sites

The accurate answer of questions regarding the genome-wide distributions of transcription factors or histone marks rely on the accurate identification of enriched regions profiled by ChIP-Seq. The peak caller which finds all the enriched regions thus plays indispensable roles. A most wanted ability of peak caller is to identify interactions of transcription binding sites. Most existing peak callers work well when peaks are spaced far from each other but they fail to distinguish peaks that stay close to each other. In the modENCODE project, the original peak caller could not resolve close peaks. The ideal peak caller should be able to differentiate these close peaks and does not sacrifice other performance index, such as specificities. Due to the lack of the appropriate tools, these closely spaced peaks have not been systematically investigated. In Chapter 3, I present my work in developing the peak caller: PeakRanger. PeakRanger works equally well on punctate and broad sites, can resolve closely-spaced peaks, has excellent performance, and is easily customized. Benchmarks show that PeakRanger is a well-balanced and is particularly suitable for large-scale data processing. Following that, I demonstrate the discovery and characterization of a class of special binding sites: doublet peaks. The profile of doublet peaks is compared to the regular peaks with highest binding signal. The result shows that the identified doublet peaks may be preferred sites for promoters.

1.5.3 The comprehensive binding sites profile for D.melanogaster and C.elegans

The ongoing modENCODE project has generated raw datasets for the genome-wide binding sites and histone modification profiles for a large number of transcription factors. In Chapter 4, my work on the modENCODE project is described. In particular, I show my efforts in building the computational pipelines for the raw datasets. The big picture about the ChIP-Seq datasets processed for both fly and worm is shown.

1.5.4 The facility to support large-scale ChIP-Seq data processing

The datasets generated by the sequencers are of huge volumes. The size of the raw images is at the scale of terabytes and aligned reads at the scale of gigabytes. A typical human whole genome resequencing analysis generates more than 100Gb data. For ChIP-Seq experiments in model organisms such as fly and worm, the generated dataset per run is smaller than whole genome resequencing but is still significantly large. With this huge size, most web-lab laboratories will soon run out of storage spaces.

On the other hand, the huge size of next-generation sequencing dataset is no longer suitable for earlier algorithms that are designed for much smaller datasets. To extract the information from these datasets, multi-steps are required. This hierarchy thus calls for coordination among algorithms involved in analysis. Theoretically good algorithms may not work in real worlds without taking care of careful allocations of computing resources. Computer memory falls in to this category. In order to process 100Gb data with much less memory, bioinformaticians need to apply knowledge of computer system architecture and software engineering to implement algorithms and brutal na ïve codes will not be able to process data efficiently.

Cloud computing may provide a solution to these problems[69, 70]. Cloud computing service provides instant access to computing resources on demand. Commercial cloud computing service provides host thousands of computers and organize them as a whole so that computing resources on each individual computer can be accessed as if they are a single powerful computer. In chapter 3, I demonstrate a peak caller that is capable to utilize the power of cloud computing to process huge amounts of datasets.

The complex structure of algorithms and demanding requirements of software engineering make the processing of next-generation sequencing data less accessible to scientists. Without the aid from bioinformaticians with significant expertise, interpreting datasets is formidable. And this has been a bottleneck of applying next-generation sequencing technology. One effective way to cope with the burden of informatics is to encapsulate algorithms and other details of data processing into pipelines. By providing researchers the interface and hiding the implementations, it allows scientists understand and interpret their datasets much easier without worrying about the underlying complex computations. To address this problem, I show in chapter 3 a library that is able to perform integrative analysis in a much organized fashion.

Chapter 2 Genome-wide analysis of insulator binding proteins of D.melanogaster

2.1 Background

Insulators are a group of classical regulatory elements which have been implicated in many biological functions[71]. Their action is mediated by a group of insulator binding proteins (IBPs) which recognize and bind go the insulator in order to mediate their function. Currently, five D.melanogaster insulator binding proteins are known: CTCF, CP190, Su(Hw), BEAF and Mod(mdg4) [72]. Over the past decade, IBPs have been found to regulate a large number of biological functions. The best characterized of these, CTCF, is thought to act both as a transcriptional activator and repressor[73, 74], to mediate X chromosome inactivation[75], and may be involved in genetic imprinting at the IGF2/H19 locus [76].

Despite emerging insights into the functions of IBP and how they deliver these functions, our knowledge is limited to a small number of specific genomic contexts. Although the enhancer blocker role of CTCF in the classical Bithorax complex (BX-C) region has been well studied with both computational prediction and experimental validation[77], it is still not clear why CTCF also play other roles outside of the BX-C region.

2.2 Identified properties of insulator proteins

2.2.1 CP190 and CTCF binding sites are enriched of actively transcribed regions

My first work was directed at correlating the binding of Drosophila IBPs to chromatin state. To identify the position of IBPs I used ChIP-chip[53] binding site data from Kevin White's group at the University of Chicago. This data set was created by extracting chromatin from S2 cells and from pooled 0-12 hour embryos, cross-linking and shearing it, then immunoprecipitating with antibodies against one of the insulator binding proteins. The immunopreciptated chromatin fragments were then labeled, the cross-links reversed, and the fragments identified by hybridization to Affymetrix tiling arrays, thereby providing a profile of insulator binding sites.

To relate IBP binding to chromatin state, I took advantage of a recent innovation in genome-profiling which uses successive salt-extracted chromatin fractions to distinguish transcriptionally active versus inactive chromatin[78]. Chromatins extracted with 80mM NaCl is thought to represent transcriptionally active regions, while fractions obtained from 600mM can almost quantitatively recover the whole chromatin. The insoluble fractions remaining after 600mM extraction is also rich in actively transcribed regions. Using tiling arrays the Henikoff group has created salt solubility profiles for chromatin isolated from S2 cells, thereby allowing the salt solubility profile to be directly compared to IBP binding profiles. I correlated the salt extraction profiles with three IBPs:CP190, CTCF and Su(Hw). By averaging the values of salt extractions aligned to the center of IBP binding sites, I generated salt extraction profiles for each IBP, at the three different salt levels of 80mM, 150mM-600mM and 80-600mM, respectively (Figure 2-1). The 80mM extraction covers actively transcribed regions, while the 150-600mM and 80-600mM extractions represent transcriptionally inactive regions[78]. For both CP190 and CTCF, the profiles are similar for all salt extractions. The 80mM extraction shows a sharp peak

within +/- 2kb from the binding sites midpoint, indicating that these two IBP preferably bind to regions with active transcription. Correspondingly, the 150-600mM and 80-600 mM extraction profiles, which correspond to inactive chromatin, show no such peak, but are instead flat or even have a shallow valley centered around the binding site.

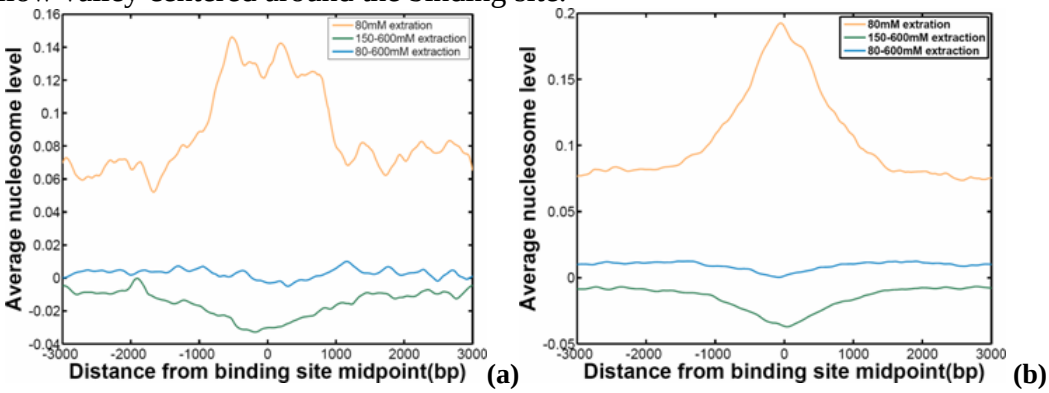

Figure 2-1 The correlation between IBPs and low-salt-soluble regions. (a) Low-salt-soluble chromatins correlate with CTCF binding sites, while the higher salt-soluble regions don't. (b) The same correlation pattern was found for CP190.

To exclude the possibility that the correlation of IBP binding peaks with chromatin solubility had occurred by chance, I generated a set of random binding sites and repeated the same analysis. No correlation with chromatin solubility was identified. Therefore, CP190 and CTCF binding sites colocalize with low-salt extractions and thus overlap genomic regions of active transcription.

2.2.2 CP190 and CTCF binding sites correlate with transcription levels

The previous analysis demonstrates that CP190 and CTCF binding sites are enriched in active chromatin, as defined by salt solubility. This suggests that these regions will be transcriptionally active. To directly test this, I compared the binding sites densities for transcription fragments separated according to their mean expression levels in D.melanogaster embryos. The data was obtained from the public website of modENCODE (www.modencode.org). I aligned transcription fragments by their transcription start sites (TSS), averaged their expression tiling array signal and quantized them into 5 successive 20% quintiles. I then plotted the position of the IBP binding site relative to the TSS against the IBP binding signal across the five quintiles (Figure 2-2). A similar plot is also produced for the end sites of transcription fragments.

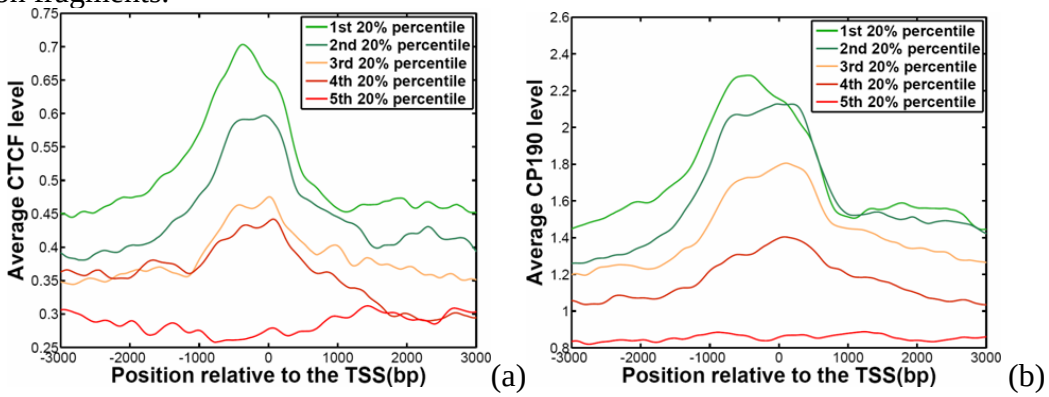

Figure 2-2 IBP correlate with gene transcription levels.(a)The higher the transcription levels, the more saturated the binding of CTCF. (b) The same pattern was found for CP190.

As expected, we observed similar profiles for CP190 and CTCF. Both of the two IBPs have higher binding densities in transcription fragments with a higher mean level of transcriptions. The first four quantile groups possess similar shapes of peaks which differ primarily in amplitude. For these four quantile groups, CP190 and CTCF have binding density peaks near the TSS and transcription end sites. Downstream of the TSS the level of IBP binding diminishes. The lowest expression quintile group shows little or no binding of CTCF or CP190.

2.2.3 Su(Hw) behaves in a reversed style compared with CP190 and CTCF

Surprisingly, IBP Su(Hw) shows a totally inverse pattern with respect to both salt extraction profiles and transcription level profiles (Figure 2-3). With respect to gene expression levels (Figure 2-3, left), Su(Hw) shows no peak of binding near the TSS. Instead, it is strongly depleted in the bodies of active genes, and has relatively high levels of binding in the upstream regions and bodies of inactive genes. With respect to chromatin state, there is no association between Su(Hw) binding sites and salt extractability.

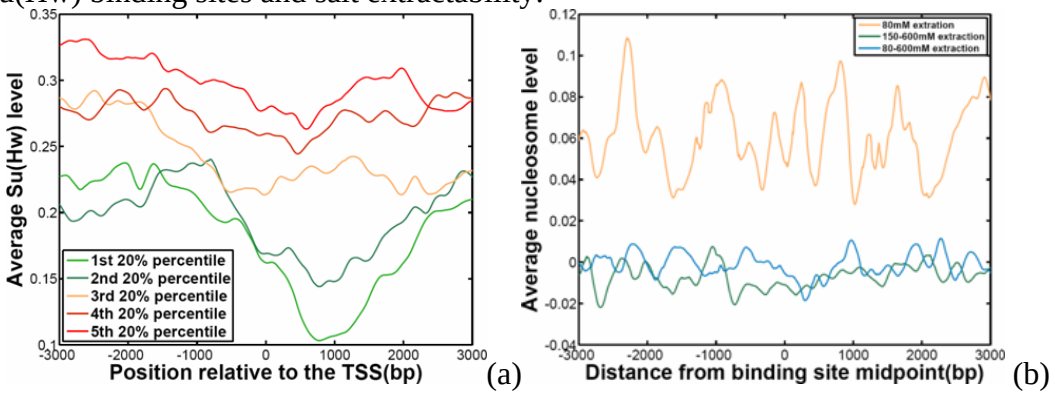

Figure 2-3 Su(Hw) shows a different pattern compared to the other two IBPs.(a)In contrary to CTCF and CP190, the higher the expression level, the lower the saturation of Su(Hw) binding is.(b)The salt solubility profiles shows that Su(Hw) does not correlate with any salt solubility profiles.

To explain these observations, I examined the relative distributions of these three IBPs(Figure 2-4). CP190 shows substantial colocalization with CTCF, but neither CP190 nor CTCF colocalize with Su(Hw) (data not shown). While 67% of CTCF and 76% of CP190 binding sites reside within the 2 kb region around the TSS, only 35% of Su(Hw) sites are found in this range (Figure 2-4). This supports the concept that CTCF and CP190 are both involved in positively regulating the activity of promoters, and possibly interact with each other, while Su(Hw) is involved in other aspects of genomic organization. This analysis is consistent with an earlier study[79].

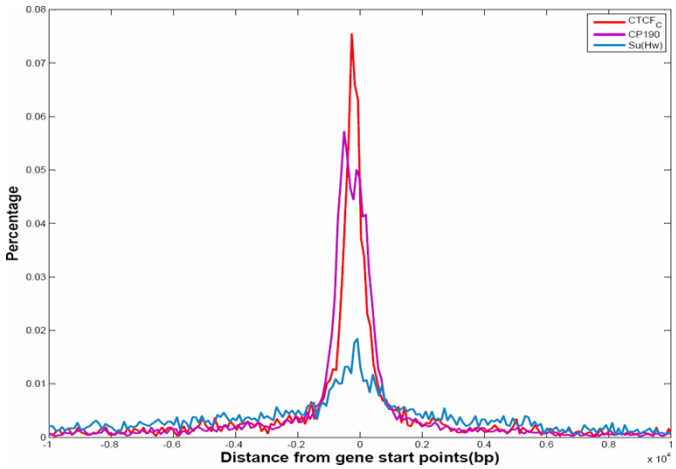

Figure 2-4 The distribution of the distance from insulators to TSS. The majority of CTCF(67%) and CP190(76%) binding sites are located within 2kb of TSS, However, Su(Hw) has only 35% of its sites located within this range.

2.2.4 Genes regulated by insulator binding proteins are functionally distinct

2.2.4.1 Su(Hw) or CTCF bound genes are shared by CP190

To distinguish whether different classes of genes are regulated by the three characterized IBPs, I defined an IBP-bound gene as one with at least one IBP within 200bp upstream of its TSS. I searched for genes using gene annotation files from Flybase [80]. 630, 2466 and 5539 genes were found for Su(Hw),CTCF and CP190 respectively (Figure 2-5). At the first glance, it is surprising to see that genes with Su(Hw) is only one fourth of that with CTCF because Su(Hw) has more enriched regions than CTCF on a genome-wide scale. A reasonable explanation is that a much larger portion of Su(Hw) enriched regions are outside promoters [81].

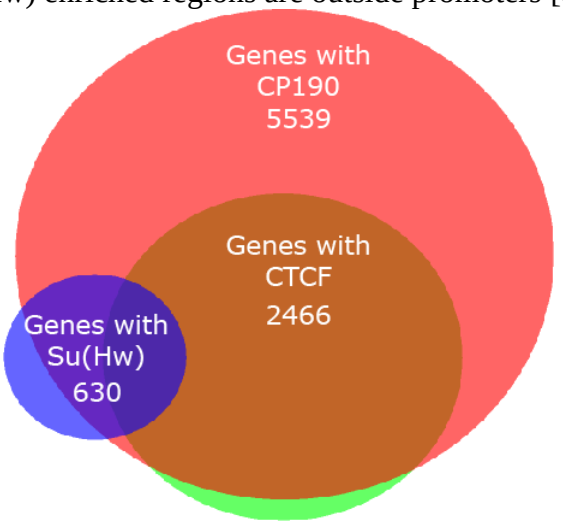

Figure 2-5 The Venn diagram of genes bound by each IBP. The number is the total number of genes bound by each IBP. Diagrams were drawn in proportion to the size of shared groups.

Since IBPs are known to form protein complexes, I expected that the three groups of genes

should show large overlaps between each other. I thus analyzed the overlap between these genes. Indeed, 80% of genes with Su(Hw) overlap with genes bound by CP190, and 96% CTCF bound genes are shared by CP190 bound genes. This supports the proposed role of CP190 as a shared recruiter of other IBPs[81]. Although CTCF and Su(Hw) bound genes are shared by CP190, it is interesting to note that the overlap between Su(Hw) bound genes and CTCF bound genes is much smaller. The overlap of genes between CP190 and the other two IBPs, but not between CTCF and Su(Hw) suggests that CTCF and Su(Hw) may regulate different groups of genes, along with the coworker CP190. Since 96% CTCF genes are shared by CP190, I only analyze genes with overlapping CP190 and CTCF sites in the following sections.

2.2.4.2 Su(Hw) bound genes were inactive compared to those with CTCF

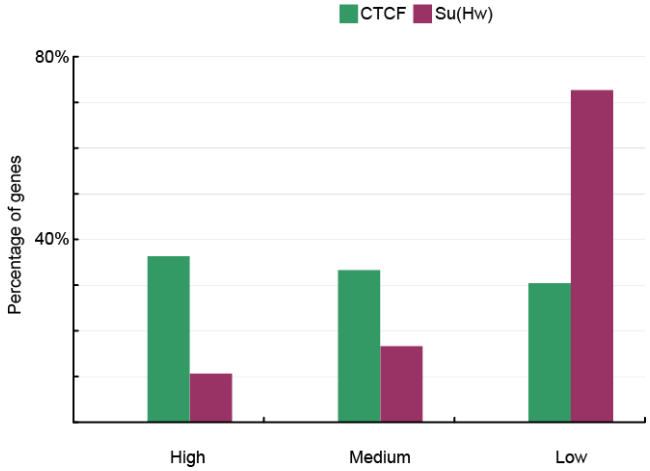

Figure 2-6 The distribution of expression levels of genes bound by each IBP.

To investigate the three groups of genes with each IBP, I first calculated the expression values for each gene and performed statistical tests to see if three groups of genes elicit similar distribution of expression values (Figure 2-6). In coherence with the overlap of genes, the Su(Hw) gene group differs significant from the CTCF group ($p < e\text{-}085$, Wilcoxon rank sum test), even though a lot of genes with Su(Hw) are shared by CP190.

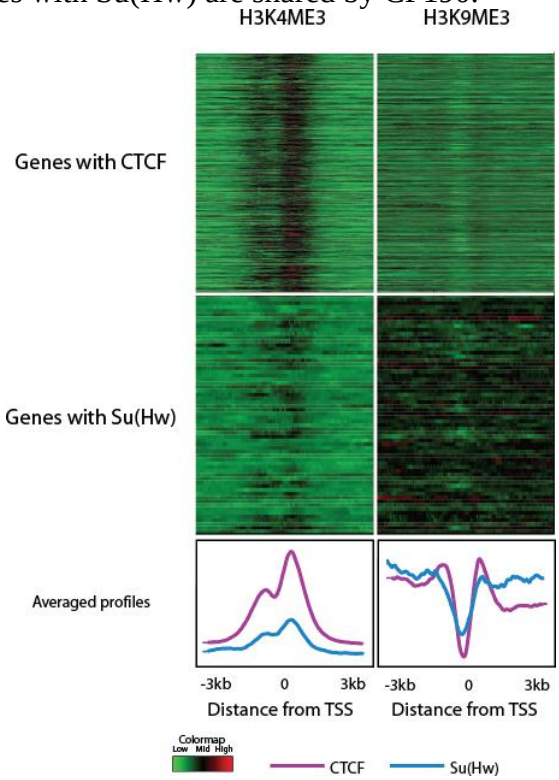

Figure 2-7 The histone profiles of the IBP binding regions of genes.

To further dissect the difference, genes in two groups were ranked and then assigned to three groups representing high, medium and low expression values. In the case of CTCF, the bound genes distributes evenly in each bin. However, nearly 80% Su(Hw)-binding genes are found in the group of low expression levels. These observations raised a hypothesis that genes with CP190 may be functionally heterogeneous, with Su(Hw) and CTCF being the two markers.

To evaluate the hypothesis, I first asked whether the three groups of genes were actively transcribed during the early embryo stages. I expected that genes bound by Su(Hw) should be generally inactive compared to genes bound by CTCF since the majority of them exhibited low transcription levels. I performed this analysis with two categories of data: H3K4Me3 and H3K9Me3. Genes were aligned by TSS and regions within +- 3kb on both sides of TSS were sampled. In the profiles of H3K4me3, CTCF bound genes are characterized by a peak centered at the TSS, which was absent in the gene group with Su(Hw). Since the peak of H3K4me3 at TSS is a positive sign of active genes, this was consistent of the distribution as shown in Figure 2-7. For H3K9Me3, since it is correlated with inactive genomic regions, I expected to see a dip around TSS and much lower levels of it on the downstream of genes with CTCF. Indeed, all IBP bound genes showed a dip centered at TSS and the downstream of genes with CTCF less enriched of H3K9Me3 compared to those with Su(Hw).

2.2.4.3 PCA analysis revealed different characters of bound genes by CTCF and Su(Hw)

I then asked if the genes bound by IBP also elicit characteristic behaviors throughout early embryogenesis. To test his possibility I performed principle component analysis (PCA)[82] on different groups of genes. The data for analysis was downloaded from the publicly available website of modENCODE. The data contains measured mRNA levels on a genome-wide scale of D.melanogaster melanogaster. For every 2 hours, data was collected until embryo 12 hour. The data thus has 6 stages (0-2hours, 2-4 hours, 4-6 hours, 6-8 hours,8-10 hours, and 10-12 hours) and the resulting data matrix is 6x8635 in size.

The analysis reveals that the first principle component accounts for 87.2% variance in the data and the second one contributes 9% (Table 2-1). Thus, the first two components represent more than 96% variance in the date, indicating that the overall shape and trend of the data can be best visualized when plotting against the first and second principle components.

Projection coefficients	**Principle components**					
	1	**2**	**3**	**4**	**5**	**6**
E0-2	-0.5152	-0.78132	-0.3414	0.012211	0.086075	-0.00082
E2-4	-0.28115	-0.13481	0.540978	-0.14156	-0.74258	-0.19664
E4-6	-0.30937	-0.04166	0.691876	0.094313	0.504784	0.400199
E6-8	-0.4374	0.320943	-0.01681	0.581169	0.140493	-0.58983
E8-10	-0.43739	0.391877	-0.31766	0.175621	-0.33959	0.63879
E10-12	-0.42094	0.336247	-0.10444	-0.77609	0.226508	-0.21265
% variance	87.2%	9.0%	2.2%	1.1%	0.3%	0.2%

Table 2-1 PCA results of the development stages data. The 6 stages are treated as variables and gene expression levels as observations. The first two components represent more than 96% variance in the date

The biological meaning of the first and second principle components can be distilled from the projection coefficients. The first principle component can be explained as the negative of overall expression value. Consider a gene that is actively expressed in every stage throughout the embryo 0-12 hours, its projection on the first component is a small negative value. Alternatively, a gene that is repressed should be projected as a large positive value. In this case, I conclude that the first principle component represents a weighted average of all developmental stages. The second principle component represents genes that are expressed at low levels in the early embryo, but become transcriptionally active later. The second principle component is different from the first one in that only the first three projection coefficients are negative and the rest three are positive. Given a gene that is repressed before

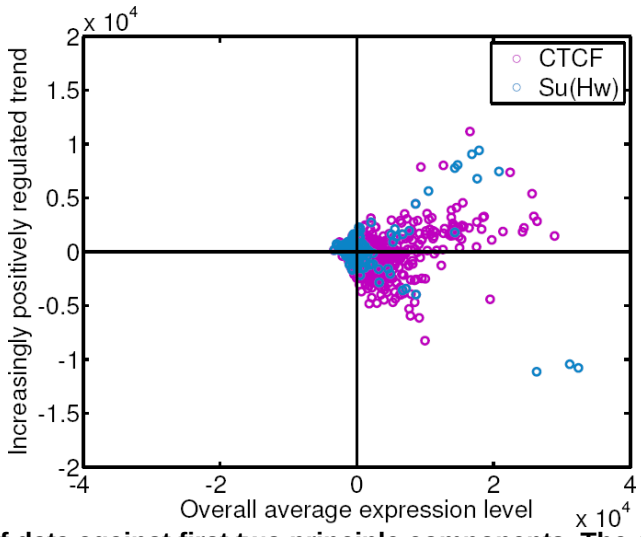

Figure 2-8 The plot of data against first two principle components. The overall average expression level stands for the first principle component while the increasingly positively regulated trend stands for the second one.

The 6th hour and actively transcribed later, it should be mapped to the positive half of the second principle component. And thus the more dramatic the change is, the larger the mapped positive value is.

To continue our comparison of CTCF and Su(Hw)-bound genes, I plotted these groups of genes against their two principle components (Figure 2-8). For convenience, I plotted the negative values of the mapped first principle component against the second principle component for all genes. The understanding of the plot can be obtained by checking the four quadrants. Genes falling into the top right quadrant (positive component1, positive component2) are initially repressed but later up-regulated. Genes falling into the bottom right quadrant (positive component1, negative component 2) instead should undergo the opposite developmental pattern. Genes with a large positive first component should have overall high expression levels, while genes occupying the left two quadrants should have much lower overall expression levels and are relatively constant throughout the stages. I found that most Su(Hw)-bound genes are located in the center of the graph, corresponding to constitutively low expression genes. Only 5% Su(Hw)-bound genes are found outside this central area. In contrary, genes bound by CTCF dominate two right quadrants. A large portion of these genes showed a very high positive value on the first principle component and maintain active transcriptions during all tested embryonic stages. The genes located in the right half of the plot undergo dramatic change of expression levels during embryonic development. Thus, I conclude that CTCF regulates a differential group of genes compared with Su(Hw).

2.2.4.4 Gene ontology (GO) analysis revealed different functional annotations of genes bound by CTCF and Su(Hw)

To further explore the significance of the differential genes bound by the three IBPs, I conducted gene ontology (GO) analysis[83] using these genes. Three major GO categories, including molecular functions, cellular components and biological process were analyzed (Table 2-2).

GO Terms, Molecular functions	Genes with CTCF	Genes with Su(Hw)

binding	●	
translation regulator activity	●	
transcription regulator activity	●	
structural molecule activity	●	
enzyme regulator activity		
catalytic activity		
transporter activity		●
molecular transducer activity		●
GO Terms, Cellular components	**Genes with CTCF**	**Genes with Su(Hw)**
macromolecular complex	●	
organelle part	●	
organelle	●	
membrane-enclosed lumen	●	
cell	●	
cell part	●	
envelope	●	
GO Terms, Biological process	**Genes with CTCF**	**Genes with Su(Hw)**
biological regulation	●	
metabolic process	●	
cellular process	●	
growth	●	
gene expression	●	
localization	●	
reproduction	●	
developmental process	●	●
establishment of localization	●	
multicellular organismal process	●	●
maintenance of localization		

Table 2-2 Gene ontology analysis for genes bound by CTCF and Su(Hw). Only significant GO terms are listed for each category.

I found that although these genes shared some GO terms in the biological process category, they were substantially different in molecular functions and cellular components. The most significant case is the cellular components category in which CTCF bound genes share no GO term with Su(Hw) bound genes. In the molecular function category, CTCF bound genes are enriched in the structural molecule activity. Since CTCF is thought to play major roles in chromatin organization, this observation is as expected. However, Su(Hw)-bound genes are never annotated with this GO term. In the biological process category, I still observed a significant discrepancy between genes with CTCF and genes with Su(Hw) for many GO terms. Together, the GO analysis shows that CTCF may regulate a group of genes which are functionally distinct from those regulated by Su(Hw).

2.3 Discussions

In this chapter, the genome-wide analysis of D.melanogaster insulator binding proteins is shown. We have confirmed that Su(Hw) serves as an indicator for repressed genes, at the genome-scale, which agrees with its roles previously identified in a few loci. The sharing of CTCF peaks with CP190 indicates that they are close collaborators. The much larger number of

CP190 peaks indicates that it may serve as the factor that first binds to insulators, followed by the binding of CTCF. The co-localization of CP190 and CTCF with active chromatins indicates that open chromatin may be a requirement for the effects of IBP. Another interesting to note is the discrepancy of the genome-wide distributions between CP190/CTCF and Su(Hw). CP190/CTCF appears like other putative transcription factors that are primarily enriched at promoters. However, Su(Hw) does not show this trend, although it binds to certain promoters. This indicates that even for the family of insulator binding proteins, there may be functional distinctions.

Chapter 3 Building the computational tools for ChIP-Seq analysis

3.1 Building an ultra-fast and multi-purpose peak caller

3.1.1 Background

The genome-wide characterization of chromatin protein binding sites and the profiling of patterns of histone modification marks are essential for understanding the dynamics of chromatin, unraveling the transcriptional regulatory code and probing epigenetic inheritance. The main technique for performing this characterization is chromatin immunoprecipitation (ChIP), coupled with massively parallel short-read sequencing (seq)[16, 54, 84-86]. Unlike its predecessor ChIP-chip [53, 87], ChIP-seq provides improved dynamic range and spatial resolution[54].

After mapping sequenced ChIP reads to the reference genome, the first critical task of ChIP-seq data analysis is to accurately identify the target binding sites or regions enriched in histone marks [55]. Since downstream analysis relies heavily on the accurate identification of such binding sites or regions, a large number of algorithms have been proposed for peak calling[64-66, 85, 88-100].

Despite the availability of such a large set of peak callers, many of these algorithms have disadvantages in real-world settings. Some algorithms have high sensitivity, but call an excessive number of false positive peaks due to low specificity. Others have the opposite problem. Another limitation of the current generation of peak callers is that many are optimized to detect either narrow punctate features, such as those generated by transcription-factor binding site experiments, or else optimized to detect broad peaks, such as those characterized by regions of modified histones. Hence a ChIP-seq production environment may need to install and maintain two different peak calling software packages. Those algorithms that attempt to handle both type of peak typically do so at the sacrifice of inter-peak and spatial resolution. The former is the ability to distinguish two or more closely-spaced peaks, while the latter is the ability to correctly locate the target binding site or histone modification boundaries. Both types of resolution are essential for understanding the underlying biology of chromatin dynamics. An example of how loss of resolution can affect the interpretation of ChIP-seq data is shown in Figure 3-1.

Software usability is also an issue. Some otherwise excellent peak callers are difficult to use because they require unusual data file formats, run slowly on real-world data sets, or do not take advantage of cluster computing. Poor usability can also impede the ability of a researcher to integrate the software with other tools in an analytic pipeline.

Here we present our efforts to address these concerns by creating PeakRanger, a novel peak caller that is both accurate and usable. Across a series of six accuracy benchmarks and three software usability benchmarks, it compares favorably to 10 other peak callers selected from the recent literature. In addition, PeakRanger supports MapReduce[101] based parallel computing in a cloud environment, allowing it to scale well to large data sets in high-volume applications.

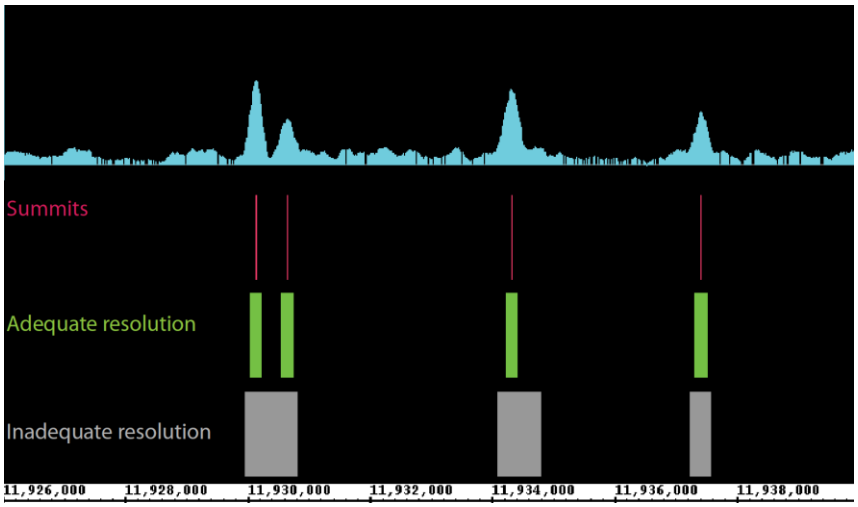

Figure 3-1 The strategy of calling broad regions and resolutive power of peak callers. Some peak callers are designed to call surrounding enriched regions instead of summits. This will degrade the accuracies of estimating the locations of binding events(summits) and also significantly reduce the resolutions.

3.1.2 Algorithm design and implementation

3.1.2.1 Reads profile building

The first step of peak calling is to build a read coverage profile using aligned raw reads. A key step in ChIP-Seq is to shear the immunoprecipitated chromatin into fragments of 200-500 bp prior to extracting the DNA and sequencing it. Because the shear size is much larger than the small reads produced by early next-generation sequencing machines, many peak calling algorithms make use of the “shift” distance between coverage peaks defined by plus and minus strand read alignments, but this has become less useful as the read length produced by next-generation sequencers approaches the ChIP-Seq DNA shear size. PeakRanger uses the same “blind-extension” strategy as PeakSeq[65] in which the shear size is provided by the user and not estimated from aligned raw reads. This choice significantly simplifies the software design and improves performance.

3.1.2.2 Peak calling and summit identification

We identify broad regions of signal enrichment using the same algorithm as PeakSeq, which detects contiguous enrichment regions by thresholding. To localize summits within these regions of enrichment we use a "summit-valley-alternator" algorithm. This algorithm starts by searching for the first summit within the region, where a summit is defined as the location that has the maximum signal value before subsequent locations drop below a pre-defined cutoff value. The value is calculated by multiplying the current maximum signal value with delta, a tuning factor that should be chosen based on the needs of users. Delta is in the range (0, 1); an optional dynamic delta algorithm is also provided. Since the reads signal of broad regions are usually noisy, we perform additional signal processing before calling summits. See methods for details.

3.1.2.3 Software Engineering

PeakRanger is written in C++, and can be compiled on Linux, MacOS and Windows Platforms. It runs as a command-line program.

3.1.3 Algorithm evaluation

3.1.3.1 Sensitivity

In order to evaluate the sensitivities of the 11 algorithms, we evaluated them using two independent ChIP-Seq datasets whose binding sites had been validated by qPCR[85, 96].Peaks called by each peak caller were ranked by their confidence scores and then compared to the list of validated sites. As measured by the average recovered proportion of validated sites, PeakRanger ranks within the top group, all of which have very similar sensitivities(Figure 3-2). The highest ranking peak caller in this set was F-Seq, but it performed poorly in the specificity test as described below.

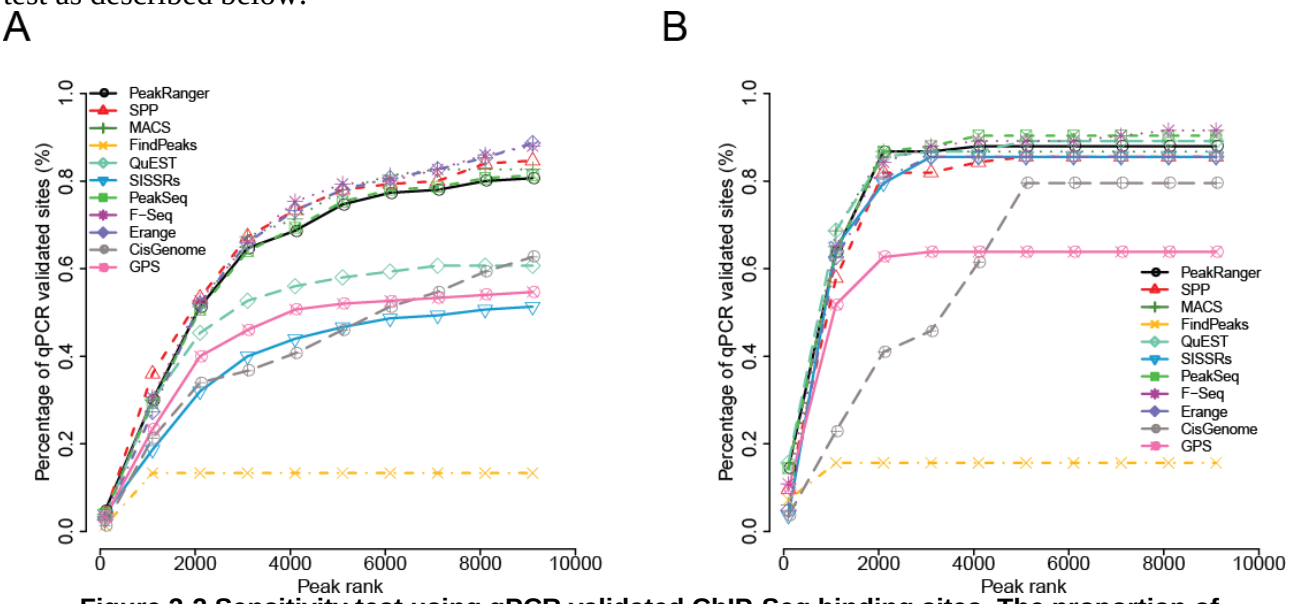

Figure 3-2 Sensitivity test using qPCR validated ChIP-Seq binding sites. The proportion of recovered qPCR validated binding sites is shown as a function of the ranked peaks called by each peak caller. Peaks are ranked based on significance values reported. A) Test results on the GABP dataset. B) Test results on the NRSF dataset.

3.1.3.2 Specificity

It is more difficult to evaluate the specificity of peak calling than sensitivity because there is no golden standard of true-negative binding sites of sufficient size to confidently evaluate specificity. To partially address this issue, we performed a specificity analysis using a previously-published synthetic dataset [97]. This data set was generated from a real-world control (no antibody) experiment that contains no binding events, which was then spiked with simulated binding site peaks. Since all peaks were generated by the author, the locations of all simulated binding sites are known and false positive peaks can thus be defined.

Figure 3-3 graphs the true positive rate against (1-the false positive rate) for each of the peak callers at a fixed FDR rate of 0.01, as shown in Figure 3, in the top group, PeakRanger, PeakSeq, GPS and MACS have nearly the same good specificity and sensitivity. SPP is close to

the top group. While SISSRs has higher sensitivity, it suffers from higher false positives. In contrast, although CisGenome called only a few false positive peaks, it recovered fewer peaks than the top group. F-Seq, Erange and FindPeaks all had unusually high false positive rates in this test.

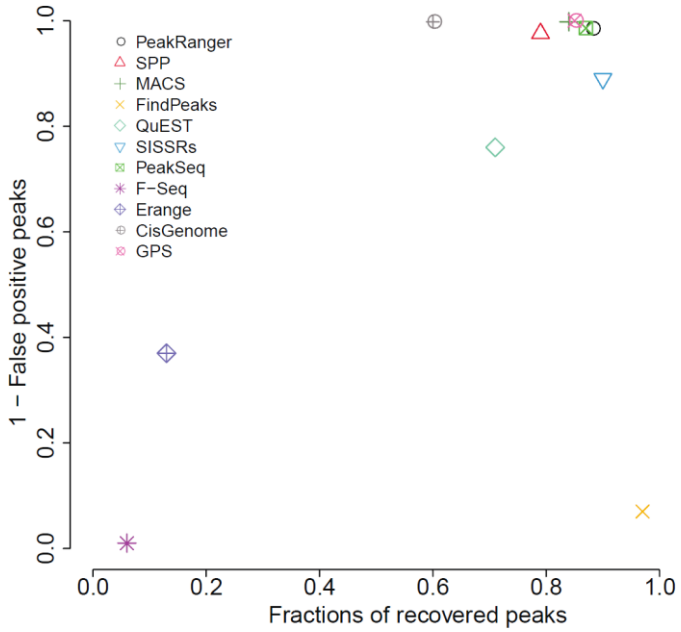

Figure 3-3 The specificity test. Peak calls of all peak callers on a synthetic dataset are shown. All peak callers were configured to have a FDR cut off of 0.01.

3.1.3.3 Spatial accuracy

Spatial accuracy measures the ability of the peak caller to correctly identify the biological binding site underlying punctate peaks. To evaluate spatial accuracy, we again used the ChIP-Seq data sets for the GABP and NSRF transcription factor targets. To identify the most likely biological binding sites, we used MAST[68] and the canonical target binding site motif and corresponding position specific scoring matrices (PSSMs) to find all matches in the 200bp surrounding regions.

We ran each of the peak callers on the data sets, and measured the distance between the binding site motifs and the centers of the closest overlapping peak call. As shown in Figure 3-4, algorithms that report peaks as single bp coordinates are significantly better than those that report broader regions. In particular, SPP, FindPeaks, GPS and QuEST were all tied for first place, closely followed by PeakRanger. However, the difference in spatial accuracy among the top-ranked peak callers is small.

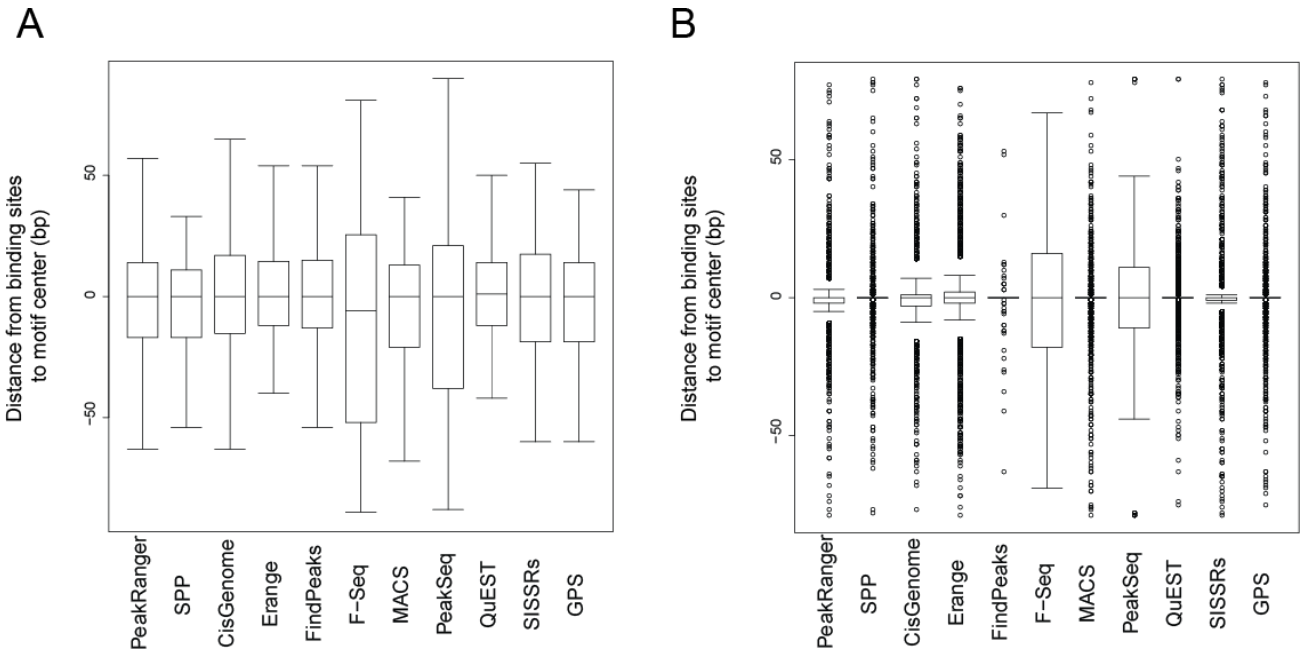

Figure 3-4 The spatial accuracies of peak callers. The distance from binding sites to motif center is measured for A) GABP and B) NRSF.

3.1.3.4 Intra-peak resolution

This benchmark measures the ability of peak callers to distinguish between two closely-spaced peaks. This is a particularly difficult task for region-reporter algorithms, which tend to merge close peaks, potentially missing biologically-significant duplets. PeakRanger identifies closely-spaced summits within an enriched region by identifying local maxima within a smoothed model of coverage.

There are no real-world gold standard data sets for evaluating inter-peak resolution, so we adapted the semi-synthetic data set used previously for the specificity benchmarks. We created a series of derivative data sets to simulate closely spaced binding sites by generating a peak adjacent to each synthetic binding site. The inter-peak spacing varied from 200 to 500 bp in each of 13 derived data sets. To compensate for changes in coverage introduced by this modification, we added the same number of reads to the control. Some peak callers, including PeakRanger, provide a "resolution mode" that seeks to discover all summits within an enriched region. For this benchmark, we set each algorithm to use resolution mode or equivalent when available, or the default settings when not.

As shown in Figure 3-5A, no peak caller is able to resolve closely-spaced peaks in this data set when the peak separation is less than 250 bp. In the range of 250-350 bp, FindPeaks and PeakRanger lead the group in sensitivity, but FindPeaks produces an excessive number of false positives, as shown in Figure 3-5B. The other algorithms have lower sensitivities across this range, and some exhibit very high false positive rates as well. MACS crashed on the 200bp, 400bp and 500bp data sets, and so these data points are missing.

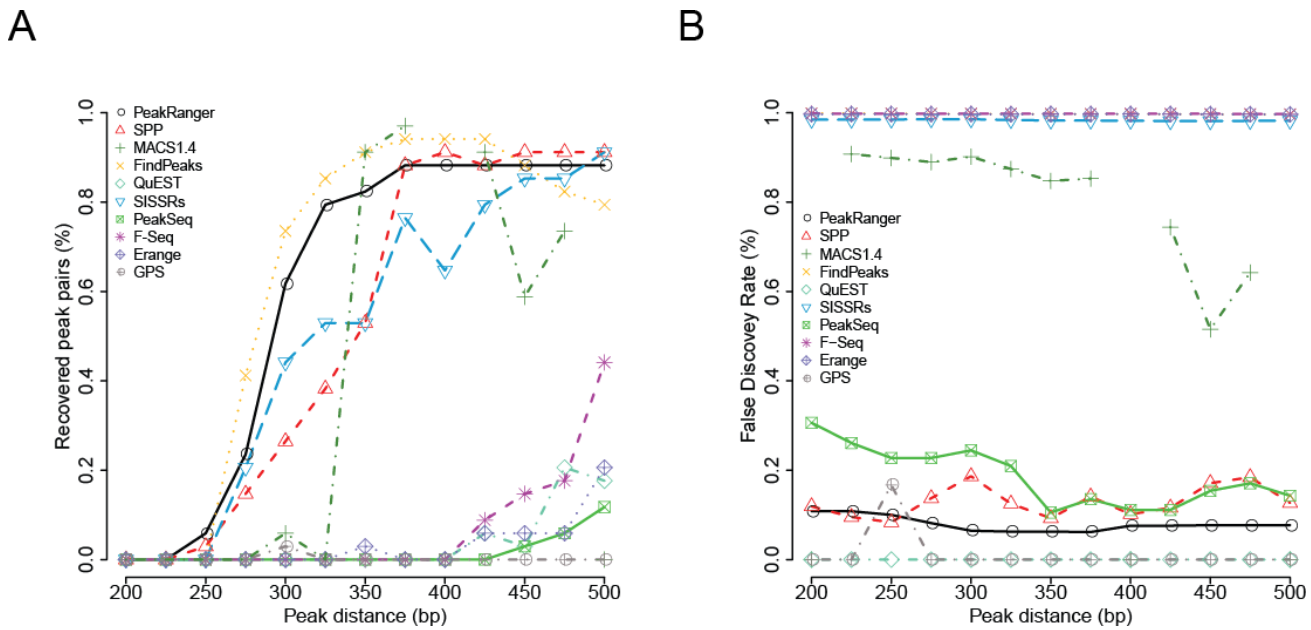

Figure 3-5 Resolution test. We called peaks on a series of semi-synthetic datasets consisting of paired peaks of increasing inter-peak separation. A) The percentage of close peaks recovered as the function of increasing inter-peak distance. B) The percentage of false positive peaks called. MACS crashed on the 200bp, 400bp and 500bp datasets, and so these data points are not plotted.

3.1.4 Usability design and performance tuning

Published algorithms are sometimes released in the research prototype stage, and do not have the software engineering necessary to work in a high volume, high availability setting. Ideally, a number of software engineering issues should be addressed. First, the software should be as fast as possible. Our experience in large projects such as the modENCODE project[51] supports the notion that a faster peak caller will significantly reduce the time to analyze and interpret ChIP-Seq data, because all the downstream analyses rely on accurate peak calls and there is often a cycle in which the results of downstream analyses inform additional rounds of peak calling using different parameter sets. Second, the software should support multiple common data formats. Transforming file formats requires extra time, computing resources, and introduces a step in which programming errors can creep in. Third, the software should be easy to use and requires less computing expertise from users. Finally, the software should be able to handle very large ChIP-Seq data sets, given the rapid increase in next generation sequencing capacity.

We implemented PeakRanger in the compiled C++ programming language to optimize performance. We avoided performance losses from disk I/O by keeping all working data in memory rather than in temporary files; this has the effect of trading a larger memory footprint for increased execution speed. To take advantage of modern multi-core processors, we also designed PeakRanger to use parallel processing.

To benchmark the performance of PeakRanger against other peak callers, we recorded the running time of them required to process a typical data set. As shown in Figure 3-6, PeakRanger is more than twice as fast as the next fastest peak caller tested, while consuming an acceptable amount of memory.

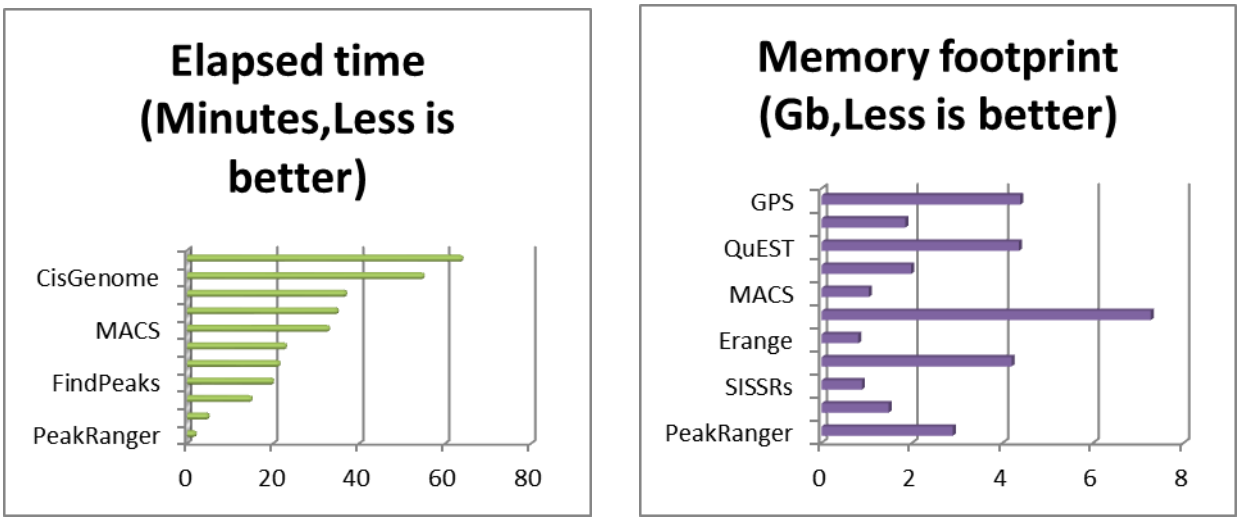

Figure 3-6 The performance of peak callers. Running time and memory footprint was recorded for peak callers using the GABP dataset.

To enable the support of multiple input data formats, we adopted designs shared by SPP and MACS which separate data loading from data processing. We wrote individual modules for specific data formats and let users to choose the one they need. PeakRanger currently supports Bowtie[61], Eland, SAM[102] and BAM[102] formats. Other file formats can be added by writing additional importation modules. PeakRanger is also capable of exporting its results in formats suitable for data visualization, including both compressed and uncompressed versions of the UCSC Genome Browser "wiggle" format.

To support multiple species, peak calling packages need basic genome build information such as the names and sizes of chromosomes. For users' convenience, PeakRanger can either derive this information directly from the input files, or can be given pre-computed genome tables. Although the former mode is convenient, it does add a small amount of overhead to the execution time.

Although hard to quantify, we noted considerably variation in the difficulty of installing and configuring the various peak caller packages during our benchmarking tests. For example, some packages require the user to make changes to the source code in order to change the location of hard-coded file paths and run-time parameters. PeakRanger makes all its run-time configuration parameters available as command-line options, and also provides a reasonable set of presets for common analysis tasks. For example, PeakRanger provides "resolution mode" and "region mode", which are presets suitable for analyzing transcription factor binding sites and other punctate data on the one hand, and broad regions such as histone modifications on the other. All run-time parameters can be read from external configuration files as well, allowing parameter sets to be managed by source code control, versioned, and shared among laboratories.

PeakRanger does not provide a graphical user interface (GUI) such as those provided by CisGenome , USeq and Sole-Search[89]. While GUIs are convenient for casual users, they make it difficult to integrate the software into the automatic workflows needed by high-throughput laboratories, which are the target audience for PeakRanger.

	GUI	Command line support	Data format	Customizable input	Automatic format detection	Species	Reusable configuration file	Wiggle file generation	No preprocessing	Parallel processing	Cloud parallel computing
PeakRanger		Yes	Eland, Bowtie, SAM/BAM, BED	Yes		All	Yes	Yes	Yes	Yes	Yes
MACS		Yes	Eland, Bowtie, SAM/BAM, BED		Yes	All		Yes	Yes		
FindPeaks	Yes	Yes	Eland, Bowtie, BED, GFF			All		Yes			
SPP			Eland, Bowtie, MAQ, Arachne			All		Yes	Yes	Yes	
QuEST		Yes	Eland, Bowtie, Solexa, MAQ			All		Yes		Yes	
GPS		Yes	Eland, Bowtie, SAM, NovoAlign, BED			All			Yes		
Erange		Yes	Eland, Bowtie, Blat, BED			All		Yes			
CisGenome	Yes	Yes	Eland, BED			All		Yes			
F-Seq		Yes	BED			All		Yes	Yes		
SISSRs		Yes	BED			All			Yes		
PeakSeq			Eland			Human		Yes			

Table 3-1 Usability summary of peak callers. This table summarizes commonly supported software features by existing peak callers.

3.1.5 Real world usage of PeakRanger

It is common for studies of histone modifications to identify broad regions enriched in the modification of interest and then to correlate these broad regions with other biological annotations such as genes. Although this type of analysis is straightforward, it ignores the detailed internal structure of the enriched profiles, which can contain summits and valleys relating to quantitative differences in modification efficiency and/or heterogeneity within the sample.

Recently there have been several publications reporting biologically significant phenomena based on the internal structures of the enriched histone modification regions [17, 18, 103]. Therefore it is desirable that a peak caller be able to retrieve both broad enriched regions while simultaneously identifying the detailed summits within these regions. Here we demonstrate such an example using PeakRanger.

In the paper recently published by He et al[17], the authors found that after exposures to 5-α-dihydrotestosterone (DHT) the central nucleosome was depleted from a subpopulation of androgen receptor (AR) binding sites, leaving a pair of flanking nucleosomes. Without knowing the region structure in advance, it is difficult to identify the paired nucleosomes from the read coverage signal alone, and He et al built additional models to identify and quantify the paired binding sites.

We applied PeakRanger directly to the He data set, using a preset that allowed it to find both broad enriched regions and summits within the regions. We then compared the number of summits in each enriched region before and after DHT exposure to directly identify the subpopulation of AR binding sites that have depleted central nucleosomes. In order to accomplish this objective, we configured PeakRanger to detect summits with comparable heights. As shown in Figure 3-7A, the profile plot strongly resembled that reported in the original publication, and had an average twin-peak separation of 360 bp, close to the publication estimate of 370 bp. As a comparison, we repeated the same procedure using QuEST. The resulting estimated peak distance was 240 bp and the profile plot departed from the original one.

For other peak callers, since no information is available for the number of summits of an enriched region, we could not perform the same analysis.

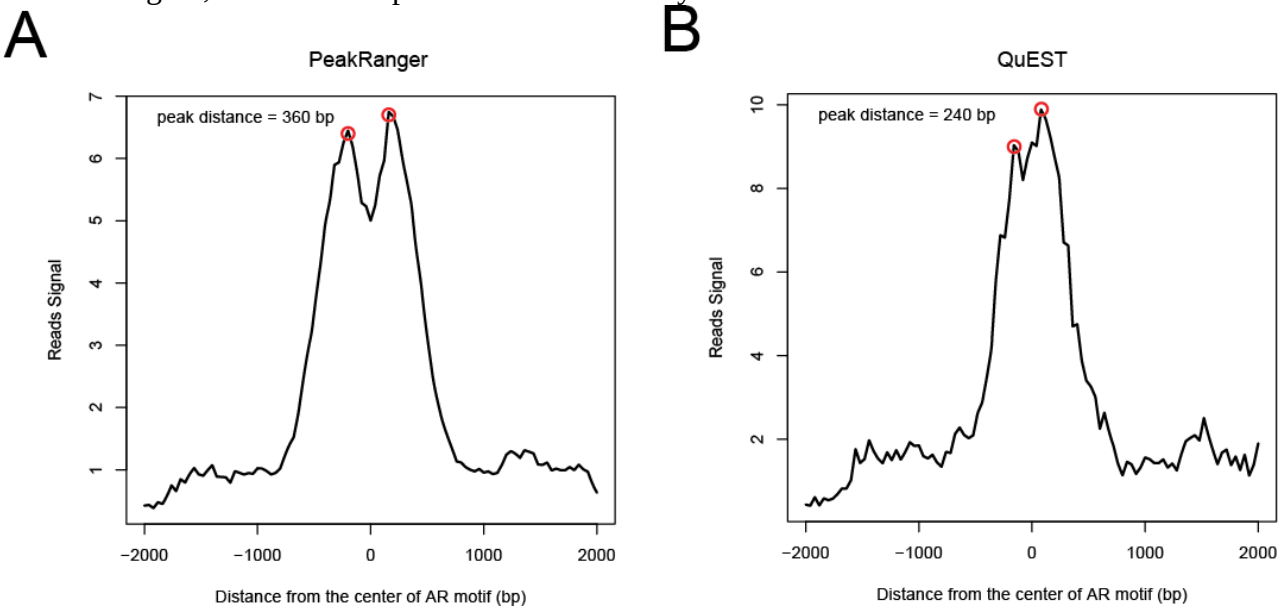

Figure 3-7 Estimating the peak distance from DHT sensitive subgroups. The analysis conducted by He et al[17] is repeated by using just peak calls generated by A) PeakRanger and B) QuEST. PeakRanger gave a much closer estimate of the twin-peak distance than QuEST.

3.2 Enabling cloud computing for peak calling

3.2.1 Background

Chromatin immunoprecipitation sequencing (ChIP-Seq) is establishing itself as an essential tool for probing the dynamics of chromatin and protein interactions. After mapping the raw output reads to reference genomes, obtaining the bound regions of the proteins is crucial for understanding the underlying cis-regulations. This "peak calling" step thus has attracted lots of attentions of researchers.

With sequencing industry's rapidly increasing capacity to generate more and longer sequencing reads, peak calling algorithms face similar challenging demand for computational resources as other next generation sequencing data processing algorithms. Instead of purchasing and maintaining new hardware and over-pay for the new hardware, cloud computing offers an alternative way to satisfy the demand for computing resources. Researchers are able to rent computing facilities from cloud computing service provides and only pay for the computing hours actually used. Researchers also have great flexibility to add or remove computing units at will and need not to worry about maintaining and upgrading of these rented computing units. Beyond these benefits, current cloud computing infrastructures also offer a computational model called MapReduce[101] which was originally designed by Google to process huge datasets. Unlike previous parallel programming frameworks, MapReduce hide the details of coordination of cluster computers so that users only need to work on their own algorithms and thus greatly reduces the difficulties of designing parallel applications. There has been a lot of implementations of Map Reduce and among them Hadoop is a free open source implementation.

After years Hadoop[104] is now the industry standard and has been deployed by many large companies like Amazon and Yahoo.

Researchers of next-generation sequencing projects have investigated a lot on how to utilize the power of cloud computing with the help of Hadoop. CloudBurst[105] is among the earliest that use Hadoop to speed up aligning raw sequenced reads. Soon after that, CrossBow[106] and Myrna[107] illustrated processing RNA-seq datasets in hours that would otherwise months if using regular methods. There are also Hadoop projects in other fields other that next-generation sequencing. They show the MapReduce framework help spread the demand for computing sources to individual nodes in the cloud and thus greatly reduces the demand for a single computer.

Even though Hadoop has made great success, the reports on integrating Hadoop with ChIP-Seq algorithms are rare. The reason partly roots in the nature of peak calling algorithms. Hadoop requires the input of an algorithm can be splitted to random smaller portions which can then be processed independently of each other. Further, the core algorithm should also be splittable to independent sub-jobs. This diagram fits very well to the above mentioned next generation sequencing reads aligner projects. For example, since each read can be mapped independently of other reads, CrossBow launches a large number of workers for individual reads. Unlike these read aligners, ChIP-Seq peak callers usually need to compile information of the whole chromosome before calling individual peaks. These characteristics significantly limit the potentials of applying Hadoop to ChIP-Seq peak calling algorithms. It is ideal that these peak callers can be rewritten from bottom up to satisfy the requirements for independencies, however, the process can be significantly time-consuming. Further, not every computing language can be supported out of box by Hadoop.

Another issue to be handled is the security and privacy of the data. In order to use the rented computing units, users have to first transfer the data to the remote computers. No matter how users send the data to cloud service providers, the chance still exists for information leak. Besides, given the current network transfer rates, it will be time-consuming to upload and download huge datasets. In this case, for laboratories already having private local clusters, it will be nice if we can transform them into private clouds and perform Hadoop computing within them. However, up to today, most Hadoop projects are conducted on Amazon AWS and other off-site cloud service providers.

In order to address these concerns, we first investigated the potentials of integrating cloud computing with ChIP-Seq peak calling algorithms without total rewriting original algorithms. We proposed a design pattern that may help applying Hadoop to peak calling algorithms. We then modified PeakRanger based on the proposed pattern. Subsequent benchmarks show that the cloud-PeakRanger is up to 10 times faster in a 64 node cluster, while maintaining the same sensitivity and specificity of the original algorithm. After that, we evaluated the efforts needed to deploy cloud-PeakRanger in our own private cloud. With a set of customized scripts, we found the configuration of private cloud and Hadoop cluster doable.

3.2.2 Algorithm design

3.2.2.1 Chromosome-level-independency (CLI) model

Existing peak calling algorithms generally follow a "read-and-call" scheme. They have to first load sufficient reads into memories and then build models for sample and control datasets in

order to call peaks. The process of finding enriched regions needs not just the information of the candidate region but also data of its neighborhood or even the chromosome. For example, PeakRanger divides a chromosome into a series of windows. After that, it calls candidate regions within each window using the reads within the whole window. It also has a post-peak-calling process in which it conducts the correction for multi statistical tests to further remove called peaks that are likely false positives. This multi-test-correction step also depends on information of other chromosomes because it needs the number of called raw peaks of other chromosomes. Another similar example is the QuEST algorithm which first estimates the peak shift distance based on a series of regions scattered in the whole chromosome. These characteristics are not favored by Hadoop. However, we find that usually peak calling on one chromosome is independent of other chromosomes. In another word, peak callers usually will repeat the same set of calculations on each chromosome and processing one chromosome does not require the input from another chromosome. The above mentioned PeakRanger and QuEST all follow this pattern. This finding inspires us to propose the straight-forward chromosome-level-independency (CLI) model.

In the CLI model, we make the following rules:

- Users should split the input to peak callers by chromosomes. Further, the splitting process must be independent of each other. For algorithms that also require control datasets, the control datasets must be labeled so that the algorithm can differentiate it from sample datasets.
- Users should modify the peak caller so that it is able to process a stream of mixed sample and control reads. The peak caller should not hold any assumption on when and where it will get a sample or control read.
- Algorithms should defer post-calling-process to another dedicated application that runs after all sub-programs finish.

In general, the CLI model detaches the data-loading and preprocessing step from the core of the algorithms to enable the parallel preprocessing. It also postpones the post-processing step so that the core algorithm can run independently from each other.

The major benefits of the CLI model is that users do not have to completely re-coding the algorithm, instead, they can keep the efforts minimal by just modifying segmentations that deal with data input and output. The detached preprocessing step of the CLI model will also scale very well with increasing nodes. Since the splitting process is independent of each other, it does not really matter where the reads are actually processed and thus we should initiate a large cluster to maximize the parallelization of preprocessing and reduce the time for data loading. This feature is particularly important for peak callers. Currently, the time for loading the datasets is usually the major portion of time cost for many concurrent peak callers. For example, we tested MACS on a dataset with around 200 million reads and found it spent more than half an hour to load the dataset but only less than 5 minutes to finish model building and peak calling.

3.2.2.2 Adaptation of CLI to the Hadoop framework

We then have to adapt the CLI model to the Hadoop framework. Within the Hadoop framework, a job can be expressed as a series of "map-then-reduce" sub-jobs(). In a typical MapReduce job, Hadoop first starts a certain number of mappers to map the input datasets to set of keys. Then a Hadoop partitioner assigns keys to a set of reducers. Each individual reducer then fetches the data according to the keys it receives and processes these data. In the context of

our proposed CLI model, "map-then-reduce" is analogue to "split-then-call-peaks" and chromosomes are used as keys. That is, we delegate the data loading/preprocessing to mappers and peak calling to reducers. After mappers finishes splitting data by chromosomes, the partitioner assigns jobs based on the number of available reducers and reducers will do the actual peak calling.

Hadoop is written in Java and supports Java applications out of the box. Unfortunately, most existing ChIP-Seq peak calling algorithms are written in non-Java languages, with only a few exceptions such as USeq and FindPeaks. Although Hadoop is now adding support to other languages such as C/C++ and Python, the efforts remain in infancy. It is doable that we recode the algorithms written in supported languages, but since our goal is to minimize the efforts for recoding, we instead go with an alternative mechanism provided by Hadoop: the Hadoop Streaming system. Hadoop Streaming provides an extra level of job abstraction so that mappers and reducers can be coded using non-Java languages. This mechanism allows us only to tailor the interfaces of the algorithm and leave the rest untouched so that re-programming the whole algorithm is avoided. We thus consider Hadoop Stream a perfect match with our goal.

3.2.3 Implementation of the Chromosome-level-independency with PeakRanger

PeakRanger consists of two parts: the preprocessing and peak calling, which is ideal to test the CLI model. After the modification, the preprocessing part runs as the mapper, an individual application. Each mapper will read a small portion of the datasets and emit a series of records to the Hadoop partitioner. The format for emitted records by mappers is straight forward:

chromosome read_coordianate orientation sequence

As required by the CLI model, we enabled mappers to differentiate sample reads from control reads. If the read is from control datasets, it will emit records with negative genomic coordinates, in contrary, reads from sample datasets will all be emitted with positive coordinates. The orientation field is either "+" for reads of positive strand or "-" for reads of negative strand. A typical emitted record is:

chrX 154577574 - ATGCAAGAAAGCGATTTTAAA

Since PeakRanger ignores the read qualities, the quality scores are not coded in the emit format. In case other peak calling algorithms want more information such as quality scores, these extra information can just be added to the end of the line, as long as the chromosome goes first. For reducers, we added support for the emit format mentioned above. If supports for other formats are required, users can enhance mappers to correctly parse the input datasets and emit the same records. The downstream reducers thus will not be affected.

The core of PeakRanger algorithm was not modified except we cancelled its routines for doing the post-call-processing, as required by the CLI model. We instead wrote a script to do the multiple-testing-correction after the peak calling results.

Finally, as required by the Hadoop Streaming system, we programmed both the mappers and reducers so that they both can read a special data file called "STDIN" and write to "STDOUT".

3.2.4 Configuration and deployment of Hadoop-PeakRanger

We currently run our own cluster with capable number of computers. Since it is also our goal to evaluate the potentials of using local computing facilities, we configured our cluster to make it run as a private mini-cloud. Similar to Amazon AWS which runs a much larger number of cluster nodes, our private cloud has a central cloud controller that process user requests and allocate cloud units. We chose Eucalyptus as the cloud controller. Although Eucalyptus is commercial software, it does provide a nicely maintained free open source version suitable for most academic projects. To save repeated work, we wrote our own scripts that can deploy Hadoop to a large set of allocated cloud nodes. As long as the network address list for all cloud nodes is available, the script is able to copy Hadoop binaries to each machine and then updates all necessary configurations. We also wrote utility scripts for starting, stopping and checking the Hadoop server. This Hadoop install script along with these utility scripts build a convenient package that provides most functions needed for our projects.

The optimal performance of Hadoop depends on the data and the hardware. Hadoop thus provides a large number of tunable parameters. Some major parameters such as the maximum number of parallel Hadoop tasks per node have great impacts on the system performance. Discovering the favorable set of parameters requires a lot of field tests and cooperation of IT experts. We thus just left most parameters at default values and only configured required ones such as server network address. By doing this we hope to try the best to leave our test results independent of computer hardware and any tricky system configuration. For the list of configured parameters, please refer to the supplementary lists.

Characteristic to most cloud applications, we started with building our own virtual system image. The raw image is based on the Linux distribution Debian. We then configured the raw image to run the standard Java system with Hadoop binaries installed. After that, a number of nodes were initiated with the configured image loaded. We then use our in-house made scripts to initiate the Hadoop system based on the nodes we started. The Hadoop system takes a while to go online. During the system initialization, the master will register nodes within the network. In our study, a 64-node Hadoop system took about 15 minutes to get ready.

3.2.5 Performance evaluation

After the system was online, we did two tests: 1) test with fixed number of nodes and datasets with increasing sizes; 2) test with increasing number of nodes and datasets with fixed sizes. All nodes used in this test are single-core computers with 3G RAM memories except for the master node which is a quad-core computer.

For the first test, we used a semi-synthetic datasets as described in the USeq paper. To increase the dataset volume, we just added to the original datasets additional copies of all the original reads. We chose node size 64 for this test. The number of mapper tasks was automatically determined by Hadoop based on the size of input datasets. The number of reducers was configured as 64 to make sure that partitioner assign reduce tasks evenly to each node. The mapper and reducer executables were copied by Hadoop to each node.

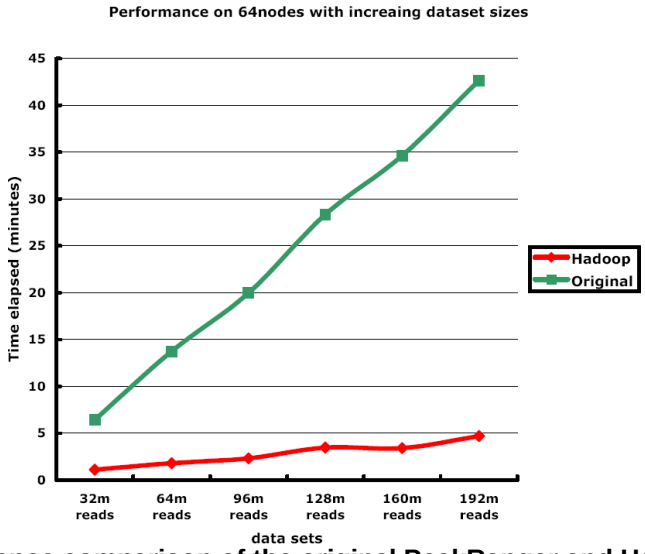

Figure 3-8 Performance comparison of the original PeakRanger and Hadoop-PeakRanger: increasing dataset size.

As shown in Figure 3-8, in general, the running time for Hadoop-PeakRanger increased much slower than the original PeakRanger. Initially, with only 32M total reads, the Hadoop-PeakRanger didn't outperform the original significantly. This was actually expected due to the overhead of Hadoop system. Because during system initialization, Hadoop has to split the input datasets and transmits them to mappers on each node. After mappers finish, Hadoop shuffles and sorts the output of mappers, which also brings overhead. However, the situation soon changed when the size of datasets got larger. With larger datasets, the overhead is diluted by the power of parallelization. As shown in the figure, the Hadoop-PeakRanger finished processing a 14G dataset with 192 million reads in less than 5 minutes, which was more than 10 times faster than the original PeakRanger.

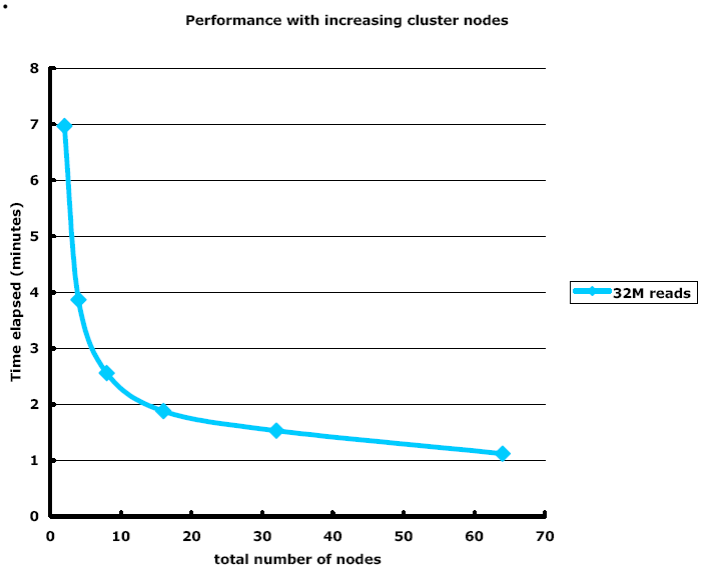

Figure 3-9 Performance comparison of the original PeakRanger and Hadoop-PeakRanger: increasing node numbers.

In the second test, we tested how fast the running time would reduce with increasing number of nodes (Figure 3-9). We used the dataset as described earlier without any duplication. The test sequence of nodes number is 2N, where N is from 1 to 6. Since the number of reducers

will significantly affect the performance, we used the same number of reducers as computing nodes to make the performance comparable among the set of tests. The number of mappers was the same for each test since the size of the input was fixed. The benchmarks shows that the running time reduced dramatically while the total number of nodes were less than 25-the number of total chromosomes of the test datasets. Adding more than 25 nodes did not significantly reduce running time because the system can run at most 25 simultaneous reducer tasks. The lag is actually in accordance to the Amdahl's law which states that the marginal benefits of adding more parallel processors falls rapidly, as long as the non-parallel portion of the algorithm is not zero. To ensure that the modified PeakRanger is algorithmically correct as the original version, we also tested the specificity and sensitivity of Hadoop-PeakRanger and found the results generally agree on each other.

3.3 A library designed for integrative analysis

3.3.1 Background

The datasets generated by next-generation sequencing technology are usually a bunch of raw reads. These raw datasets then go through another set of algorithms to produce meaningful processed datasets. Although the algorithms used to process raw datasets may differ in a lot of aspects, the results of these downstream algorithms are usually expressed in terms of genomic regions. A copy number variation (CNV) detector, for example, reports each CNV as: chromosome, start of CNV region, end of CNV region; RNA-Seq algorithms report regions with significant transcriptions; ChIP-Seq peak detectors such as PeakRanger report each peak in a similar way. Analyzing these next-generation sequencing datasets thus relies on manipulating these genomic regions. For example, a popular way to analyze next-generation sequencing datasets is to integrate multiple genomic features, including transcription factor binding sites, gene expression values from RNA-seq datasets and other genomic annotations. This type of integrative analysis usually involves calculating overlap of regions between two sets of datasets and also many other regions-based calculations.

To deal with the requirements of handling genomic regions, a couple of tools have been proposed. BEDTools[108] and CEAS[109] are two nice software suites that can perform region analyses. The biggest issue for these tools, however, is the lack of flexibility. Users may not be able to perform customized region analysis. For example, CEAS by default only provides results against promoters. And CEAS does not support the use of only a subset of promoters in a specific chromosome. These libraries are also not easy to expand in order to handle new input types. To address these issues, a library that models genomic regions and can also be easily expanded and customized is necessary. By design, the proposed library should be able to provide a set of interfaces so that users can build their own analysis based on these interfaces; The library should also be able to handle different types of input and output formats without re-writing the core of the library. In this chapter, I propose xBED, a library written in Java that models genomic regions and address these concerns. I first show the framework of the library and demonstrate its flexibility; A couple of analysis based on the proposed framework is then shown to illustrate the effectiveness of the library.

3.3.2 Design of the library

3.3.2.1 Overview

xBED models genomic regions at three levels: the region, the chromosome and the dataset as a whole (Figure 3-10). A dataset is modeled as a set of sub-dataset organized by chromosomes. Each sub-dataset is then modeled as a cohort of genomic regions.

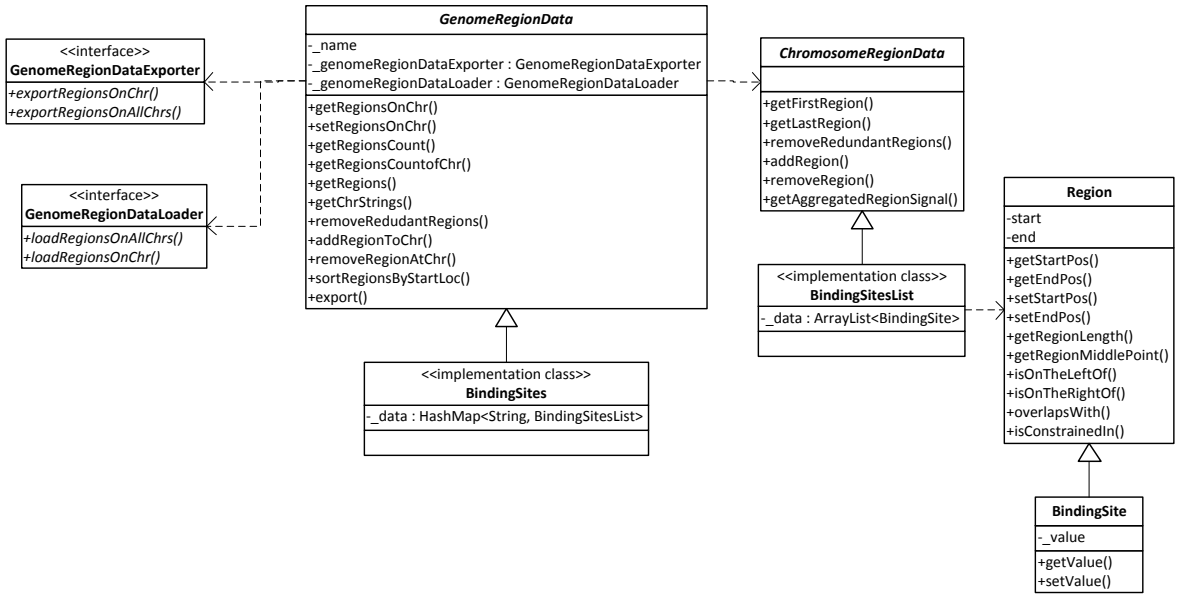

Figure 3-10 The overview of the xBED library.

The main class of this library is GenomeRegionData and ChromosomeRegionData, which encapsulate the whole dataset and the chromosome level data, respectively. The GenomeRegionData class has two interface classes: GenomeRegionDataLoader and GenomeRegionDataExporter. As implied by their names, these two interface classes specifiy the requirements for loading and exporting the datasets as a whole. Every GenomeRegionData is consisted of a set of ChromosomeRegionData, which represents the regions on each chromosome. The Region class provides abstraction for the actual individual genomic region. Thee GenomeRegionData, ChromosomeRegionData and Region classess are all abstract or partially abstract so they only specify the framework of the whole library. To actually use the framework, three concrete classes are provided:BindingSites implements GenomeRegionData; BindingSitesList implements ChromosomeRegionData; And BindingSite implements the Region class.

3.3.2.2 Flexible Data input and output

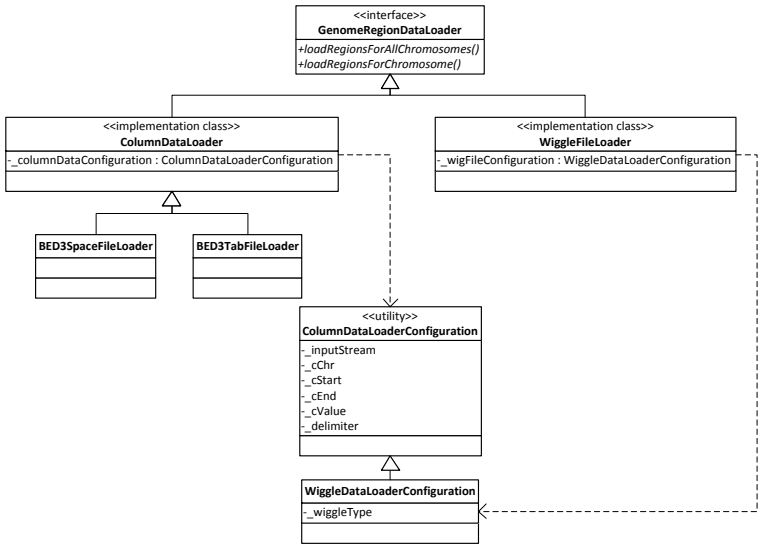

Figure 3-11 The overview of the data loading module.

```
importRegions(){
…….. //Actual data export
}
```

```
import(GenomeRegionDataImporter exporter){
  importer.importRegions();//Delegate the export job to the exporter
}
```

```
GenomeRegionData data = new GenomeRegionData();
…….. //Data initialization
GenomeRegionDataImporter importer = new GenomeRegionDataImporter();
…….. //Exporter initialization
data.import(importer); //export the data now
```

Figure 3-12 The sample data importing codes.

The data loading and exporting part are both extracted as two interfaces(Figure 3-11,Figure 3-12,Figure 3-13,Figure 3-14). By doing this, xBED can support future data I/O format and requirements. At the same time, xBED provides two loaders that process most tabulated files and wiggle files. Many popular data formats follow the table fashion: a file is consisted of a number of lines with each line a number of columns delimited by either space or the tab-character. Currently, xBED implements I/O facilities to read BED/GFF as well as other tabulated files or input streams. Reading of BED/GFF files can be abstracted as a process of extracting information from some or all columns of a tabulated file. For BED file, usually columns 1-4 contain essential information to build a genomic region: chromosome name, start location, end location and region value. For GFF file, the case is similar with only different columns. Thus ColumnDataLoader is designed to implement this abstract behavior. It allows users to specify, via the ColumnDataLoader class, how to process the tabular text file. For the convenience of users, the library also provides wrapper classes for BED files: BED3SpaceFileLoader and BEDTabFileLoader. Both of these two loaders are based on ColumnDataLoader, with a specific

ColumnDataLoaderConfiguration associated with each other. Users do not have to specify the loader configuration if they use these two wrapper classes.

Wiggle file loading is implemented as WiggleFileLoader. Parsing wiggle files is a different type of work compared to parsing tabular files and thus it is implemented separately.

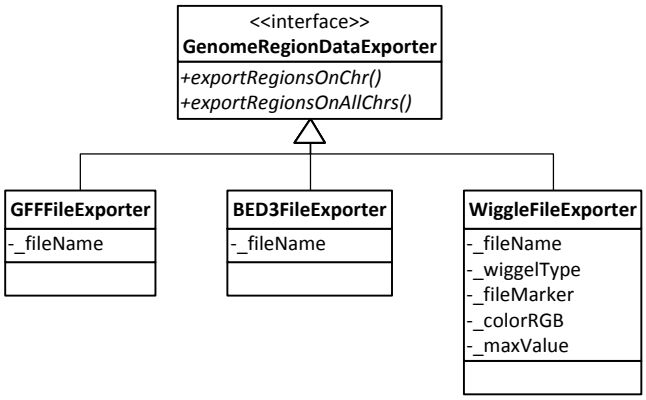

Figure 3-13 The overview of the data exporting module

```
exportRegions(){
…….. //Actual data export
}
```

```
export(GenomeRegionDataExporter exporter){
  exporter.exportRegions();//Delegate the export job to the exporter
}
```

```
GenomeRegionData data = new GenomeRegionData();
…….. //Data initialization
GenomeRegionDataExporter exporter = new GenomeRegionDataExporter();
…….. //Exporter initialization
data.export(exporter); //export the data now
```

Figure 3-14 The overview of the data exporting module

The exporters share a similar design as loaders. xBED supports exporting genome datasets as BED, GFF and Wiggle files. The support for new file formats can be added very easily by implementing the GenomeRegionDataExporter accordingly. The GenomeRegionData class does not need to modify its exporting codes since it can just call the two abstract functions. A sample segment for exporting is as following:

3.3.3 Performance of the xBED library

The performance of libraries serving similar purposes as xBED is rarely mentioned. This is at least partly due to the fact that the whole bioinformatics community has not finished transition into the stage where performance of software plays a key role. The implementation of algorithms in xBED always put the performance on top of the priority list. A core algorithm that lies under a lot of analysis is the one that finds regions in a dataset that overlap with a specified region. Given the fact that each element in the dataset is a region that is sorted based on the start index, the binary search algorithm is applied to ensure a O(nlogn) complexity. The binary search algorithm

is slightly modified so that all overlapped regions around the first found region are also found. A benchmark was performed to evaluate the performance of the aggregating profile algorithm which relies heavily on this algorithm. In the benchmark, a test dataset with binding sites and a wiggle file were fed to the XBED aggregating profiler and the one from the CEAS suite. The running time for the two profilers was recorded. As shown in Figure 3-15, the profiler from xBED is much more efficient than the one from CEAS.

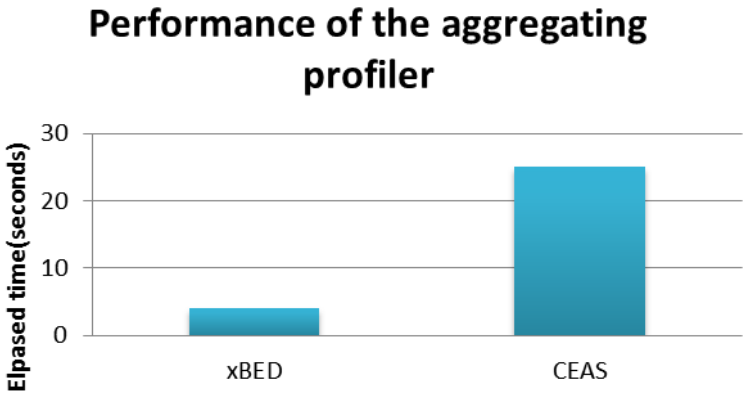

Figure 3-15 Performance comparison of the xBED library and CEAS.

3.3.4 Using xBED to implement various genomic analysis

3.3.4.1 Introduction

A large number of genomic analyses can be implemented using xBED since nearly all of these analyses are just combinations of various operations on the genomic regions on each chromosome. On top of that, xBED provides additional abstraction of the chromosomes and the whole dataset, which facilitates the cross-datasets integrative analysis. In the following I will show how to use the API included in xBED to implement some most popular analysis of next generation sequencing datasets.

3.3.4.2 Use xBED to implement aggregating profile plot

One of the most useful plots generated from sequencing datasets is the aggregating profile plot. Under the framework of xBED, it is straight forward to implement this feature with the following algorithm:

```
AggregatePlot(GenomeRegionData target, GenomeRegionData ref){
        result = InitiateResult();

        for (Each Chromsome at target.getChrStrings()) {
                for (Each BindingSite of target.getRegionsOnChr(Chromosome) ) {
                        profile = getAggregatedRegionSignal(BindingSite);
                        if (profile.IsOfLowResolution()) {
                                pad_profile(profile)
                        }
                        AddProfileToResult(profile, result);
                }
        }

        GetMeanProfile(result);
        return result;
}
```

Figure 3-16 The sample codes for aggregating plot.

The ChromosomeRegionData class has the getAggregatedRegionSignal() method which will provide the profile of overall signals of a specific region. By adding all such profiles together, we can get the aggregating profile plot. For datasets with very low resolutions, a pad_profile() method is provided to do basic intrapolations to compensate the low quality signals. Currently, xBED provides a set of implementation of variants of the aggregate profile plotting, as shown below:

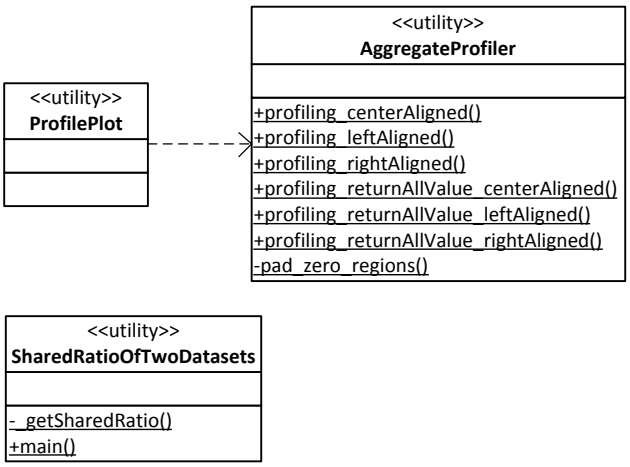

Figure 3-17 Aggregating utilities provided by the xBED library.

3.3.4.3 Use xBED to find the overlap regions between two datasets

Another useful function is to check the percentages of shared regions between two datasets. The key of this function is to get the overlapped regions and it can be implemented as shown in :

```
overlappedData(BindingSites targetData, BindingSites inputData) {
  result = new BindingSites();
  for (Each Chromsome at target.getChrStrings()) {
  result.setRegionsOnChr(Chromosome, getOverlappedRegions(targetData, inputData))
  }
// Print out the overlap ratio
  Printout( result.getRegionsCount() / inputData.getRegionsCount())
// Export the overlapped dataset as BED file
  Result.export(new BED3FileExporter());
}
```

Figure 3-18 Sample codes for region overlap calculation

This algorithm serves as the basis for many useful analysis. Once such analysis is to find out genes that overlap with any transcription factor binding sites. To do this, the program starts by loading the gene annotations and binding sites regions. Following that, the program can just copy the algorithm described in Figure 3-18.

3.4 Discussions and Conclusions

Figure 3-19 summarizes the accuracy and software engineering benchmarks discussed above, where each of the 11 peak callers examined is ranked from 0 (worst) to 10 (best) for a particular benchmark. The last column of the table is a simple sum of the ranks. No single peak caller ranks as the best on all benchmarks; in particular, algorithms with high sensitivity often have low specificity. However, PeakRanger manages a good compromise among all the performance benchmarks and ranks first in the aggregate ranking.

	Resolution: Recovery rate	Resolution: False Positive rate	Specificity	Sensitivity (gabp)	Sensitivity (nrsf)	Spatial accuracy(gabp)	Spatial accuracy (nrsf)	Speed	Memory	Usability	Overall
PeakRanger	9	8	10	6	7	6	8	10	4	10	78
SPP	8	7	5	8	4	7	10	3	5	8	65
MACS	6	5	7	7	6	3	10	4	8	9	65
QuEST	4	10	3	4	8	10	10	2	2	8	61
Erange	3	3	1	9	5	9	7	6	10	7	60
PeakSeq	2	6	9	6	10	1	6	9	7	3	59
SISSRs	7	4	6	1	3	2	9	8	9	4	53
FindPeaks	10	2	2	0	0	8	10	7	3	8	50
GPS	0	9	8	3	1	4	10	0	1	8	44
F-Seq	5	1	0	10	9	0	5	5	0	5	40
CisGenome	0	0	4	2	2	5	7	1	6	6	33

Figure 3-19 Summary of benchmarks performed in this study. For each benchmark item, peak callers are ranked and scored (see methods). The score has a range of 0 to 10 and 10 is the best score. The overall rank is based on the sum of all scores in all benchmarks.

While the PeakRanger algorithm shares many design characteristics with QuEST and FindPeaks, it makes improvements on them both. One such improvement is the algorithm to find local maxima and minima within enriched regions. QuEST searches for local maxima one at a time. While this design makes sense at the first glance, we found it improvable after a lot of tests using real datasets. In this regard, we instead designed PeakRanger to find all local maxima at one time. This "one-stop and all" ideology was originally inspired by SPP. To compensate the

unique properties of each enriched region, we also designed a naive adaptive algorithm which will generate an optimal threshold used in detection of local maxima for each enriched region. The tests presented in this study demonstrate the effectiveness of these refinements.

Although PeakRanger represents a successful compromise among multiple measures of accuracy, researchers should consider one of the other peak calling algorithms if a particular performance characteristic is of the top priority. For example, if identifying the precise center of the peak is critical to an experiment, then researchers should consider GPS, QuEST, MACS, SPP or FindPeaks, all of which have better spatial accuracy than PeakRanger.

The current design for the cloud version is based on chromosome-level-independence (CLI), which limits the practical level of parallelization to the number of chromosomes in the genome. This concept can be generalized to region-level-independence (RLI) by breaking the genomes into a set of arbitrary regions and call peaks in each individual region. However, this is dependent on the peak calls for each region being independent of each other, a criterion that is not satisfied when an enriched region crosses the region boundary. Additional manipulation of the regions to allow for overlap between them, and adjustments for the changes in coverage in overlapped regions will be necessary to implement this, and is deferred to future work. However, even with the current design we are able to archive an order-of-magnitude increase in speed, which is sufficient for most practical applications.

In this study, we only applied the CLI model to PeakRanger but we believe that the same model should also work on other peak callers as long as they do peak calling by chromosomes. We do realize that there are exceptions. One such example is MACS. Unlike QuEST which estimates the peak shift distance based on a single chromosome, MACS selects candidate regions across the genome to estimate the shift distance. In this case, if peaks across the whole genome have similar characteristics, the results should not be affected and MACS could also adapt to the CLI model. In the extreme case, if the quality of peak calling degrades significantly without information from other chromosomes, users can still parallelize the data loading part and leave the rest peak calling serial.

We also believe that similar performance gain is expected to other peak callers. The reason is that many peak callers spend the most time on the "read" step of the "read-and-call". The CLI model guarantees that the "read" step is completely parallel and scales well with the capacity of service providers. And since our design ensures that only one MapReduce job will be launched. This will save additional time by avoiding overheads of initiating multiple MapReduce jobs.

The Chromosome-level-independency (CLI) model can be further generalized as Region-level-independency (RLI). In concept, we can break the genomes into a set of regions and do the peak calling in each individual region, as long as the processes of peak calling in these regions are independent of each other. According to the RLI model, we can scale Hadoop based applications with the capacity of cloud service providers to fully explore the power of cloud computing. It is expected that the RLI model may require a lot of efforts of redesigning existing peak callers. But it is worth of trying if users want further performance gains.

Compared to Hadoop projects that archived hundreds of folds of performance gains, the largest speed up we observed in the project is about 10. While this is much less than Crossbow, our Hadoop-PeakRanger can process about 14G data in less than 5 minutes. The throughput per hour is still comparable to Crossbow. Besides, the author of Crossbow did a lot of application-specific optimizations to further improve the performance gains. Advanced techniques such as shared memories and memories mapping are not applicable to every peak caller. Adding these features require above-average programming expertise and much more time. In contrary, we

discussed a method that may be applicable to a large number of applications. What's more, the performance gains are obtained with much less efforts-no advanced system configuration and no advanced time-consuming re-programming.

Working with private cloud instead of manipulating datasets remotely is much more convenient and safer. In the private cloud we have more detailed control of the cluster configuration and can make adjustments whenever necessary. It is also a nice discovery that home-made scripts for Hadoop setup are practical and suffice most research purposes.

The framework and the design of xBED has been shown and two sample algorithms were given to show the utility of the proposed framework. The performance of the library is also demonstrated by the aggreagating profiler of the library. The presented library has been used in the modENCODE project and all the aggregating profile plots used in this thesis were produced by the library.

3.5 Methods

Read coverage profile building and peak calling

The procedures for building reads profiles and peak calling are based on those used by PeakSeq with the following modifications: Application of a mappability map is removed to enable support for multiple species. A fixed number of windows are used. Candidate regions are required to have a positive excess value in order to reduce false positives.

Coverage profile enhancement and summit detection

The read coverage profile is padded prior to summit detection. The original profile is scanned and locations with zero read counts are detected. These locations are padded with the average value of the two nearest non-zero coverage regions. The padded profiles are then scanned for summits. The algorithm starts by searching for the coordinate with reads maxima in the region. Then, all the remaining coordinates that have above-threshold reads are selected as summits. The threshold is obtained by multiplying the region-maximum value with a tuning factor (Delta) in the range (0, 1). Smaller Delta results in more summits and vice. versa. An optional dynamic delta algorithm is also provided. If users enable the dynamic delta option, PeakRanger will try its best to identify all valid summits in enriched regions while at the same time ignoring noises.

Algorithm implementation

PeakRanger is implemented using C++ and is open source. It compiles and runs on any operating systems that support the GNU G++ development environment. PeakRanger includes source files from PeakSeq, Bowtie and Bamtools [42]. Valgrind [47] tests for possible memory leaks are done for all tests described in this manuscript. Additional Valgrind tests were done using private datasets. The support for cloud computing relies on the Hadoop library[104].

Selection and configuration of peak callers

We based our selection of peak callers on two recent reviews [55, 110] to represent the algorithm diversity and popularity. We also added recently-published algorithms which had not been included in the reviews. This resulted in an initial set of 17 candidate peak callers(Table 3-2), which we then screened to exclude callers that could not be compiled, required additional

data files that we could not provide, or failed to produce peak calls in an initial test set. After screening, we finally included 10 peak callers in remained.

Algorithms	Reference	Version	Initial Screening	Notes
ERANGE		3.2.1	PASSED	
FindPeaks		4	PASSED	
F-Seq		1.84	PASSED	
GLITR		literature version	FAILED	Requires too many control tags, More than 4X the total number of treatment tags are required
MACS		1.3.7.1 and 1.4.0beta	PASSED	1.4.0beta was used for the resolution test
PeakSeq		1.01	PASSED	The package programmed using the C programming
QuEST		2.4	PASSED	
SICER		1.03	FAILED	Program requires significant modifications before
SISSRs		1.4	PASSED	
SPP		1.8	PASSED	
Useq		7.0	FAILED	Obtained zero peaks from test datasets.
Minimal ChipSeq Peak Finder		literature version	FAILED	Release webpage is missing
CisGenome		1.2	PASSED	Only Linux core programs were used
HPeak		2.1	FAILED	Program reported missing file: summary.pl while
Sole-Search		1.0	FAILED	Non-web version is not available
CSDeconv		literature version	FAILED	Computationally infeasible
GPS		0.10.1	PASSED	

Table 3-2 The compilation and selection of peak callers.

All programs were run with their default/recommended settings. Tests were done in a generic desktop with the following specs: CPU: Intel Q6600, RAM: 12G, Harddisk: 2TB 7200 rpm.

Sensitivity test

The GABP dataset and NRSF dataset were downloaded from the website of QuEST (http://mendel.stanford.edu/SidowLab/downloads/quest/). The qPCR validation list was downloaded from the [110] paper. Peaks were ranked based on the metrics provided by each peak caller. For F-Seq, which identified too many peaks, only the top 10,000 ranked peaks were used.

Specificity test

The original dataset used in the resolution test was from the website of USeq (http://sourceforge.net/projects/useq/). Peak callers were configured to have FDR 0.01 when calling peaks.

Spatial accuracy test

The GABP dataset and NRSF dataset were downloaded from the website of QuEST (http://mendel.stanford.edu/SidowLab/downloads/quest/). PSSMs were obtained from TRANSFAC[111]. The MAST program from the MEME software suite was used to detect motif occurrences[67]. Boxplots were generated with R[112].

Resolution test

The original dataset used in the resolution test was from the website of USeq (http://sourceforge.net/projects/useq/). Peaks were systematically shifted and reintroduced into

the dataset to produce a series of synthetic peak pair datasets (Figure 3-20). We excluded CisGenome from the test because it failed to complete the benchmark. MACS version 1.4.0 beta was used in this test instead of MACS 1.3.7.1 since the latter does not have the ability to call multiple summits within a region. For the PeakRanger benchmark, we used a delta value of 0.2 to enable the ability to call multiple summits. For QuEST, we used a dip_fraction of 0.8 because QuEST uses a threshold value of (1-dip_fraction) X (maxima reads). For FindPeaks, we used a -subpeaks option of 0.2 for the "-subpeaks" option. We calculated recovery rate and false discovery rate using custom Java programs.

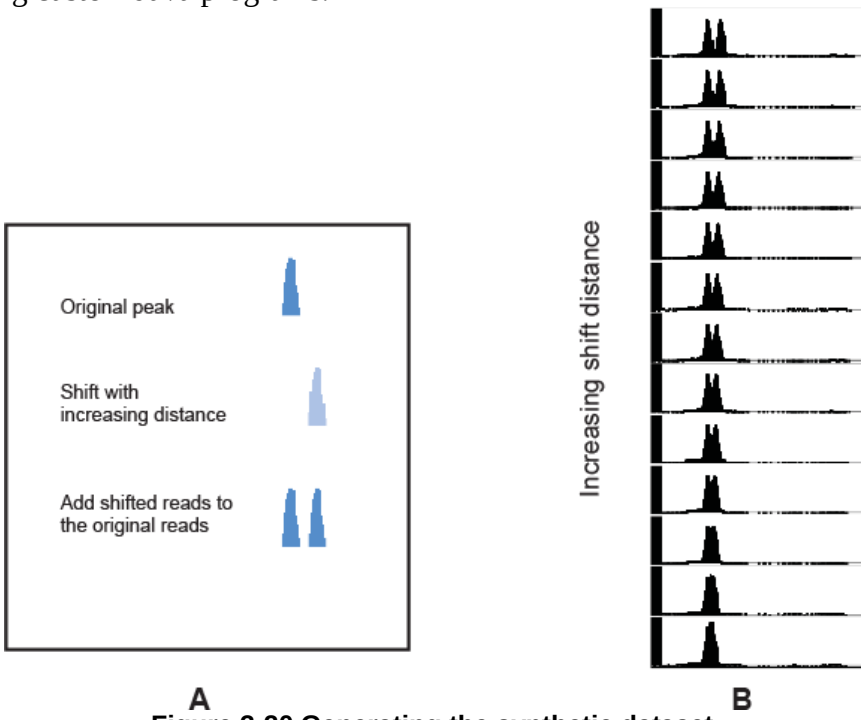

Figure 3-20 Generating the synthetic dataset

Histone modification usage example

The dataset was downloaded from GEO using the accession ID: GSE20042. We used a delta value of 0.4 for PeakRanger, and a dip_fraction of 0.6 for QuEST.

Speed and memory footprint test

We used the GABP dataset. SPP gave us an error message when we attempted to run it with parallel support, so it was run in regular mode. We ran PeakRanger with the "-t 4" option to enable parallel processing. QuEST automatically launched multiple processing sub-programs.

Plots and data visualizing

Signal tracks are drawn using the IGB browser[113].

Preparation of Summary Table

For the Figure 3-19, we ranked each peak caller based on its relative performance in each benchmark. For the resolution:recovery test, we ranked average recovery rate. For the resolution:false discovery rate, we ranked average false discovery rate. For the specificity test,

we ranked recovery rate minus false discovery rate. For the spatial accuracy test, we ranked the absolute distance between the higher and lower hinge of the distance distribution. For the sensitivity test, we ranked the average recovery rate. For the speed test, we ranked elapsed clock time. For the memory test, we ranked the peak memory footprint consumed during execution. For the usability test, we ranked the sum of the features listed in Table 3-1.

Software licensing and availability

PeakRanger can be downloaded from: http://www.modencode.org/software/ranger. We currently provide the full source code, as well as binaries for Linux 64-bit systems. Binaries for other operating system and an Amazon EC2 image will be available during the first quarter of 2011.

Chapter 4 Application of PeakRanger in the modENCODE project

4.1 Comprehensive identification of transcription binding sites in D.melanogaster and C.elegans

PeakRanger was used by modENCODE as the standard ChIP-Seq peak caller for 29 ChIP-Seq experiments for involving 23 C. elegans transcription factors across various developmental stages. It was used by modENCODE project to process all worm transcription factors' ChIP-Seq datasets. For the ChIP-Seq datasets from D.melanogaster, PeakRanger was used in the process both datasets from transcription factors and histone modifications. PeakRanger was able to process the entire C.elegans datasets in less than 2 hours running on a regular workstation with 8G ram and a quad core CPU. This illustrates PeakRanger's ability to integrate into a high-throughput environment. Ultra-high through-put enabled great collaborated analysis among different labs. A couple of internal analysis shows that peaks produced by PeakRanger were of high quality (Data not shown).

4.1.1 PeakRanger effectively recognize close summits

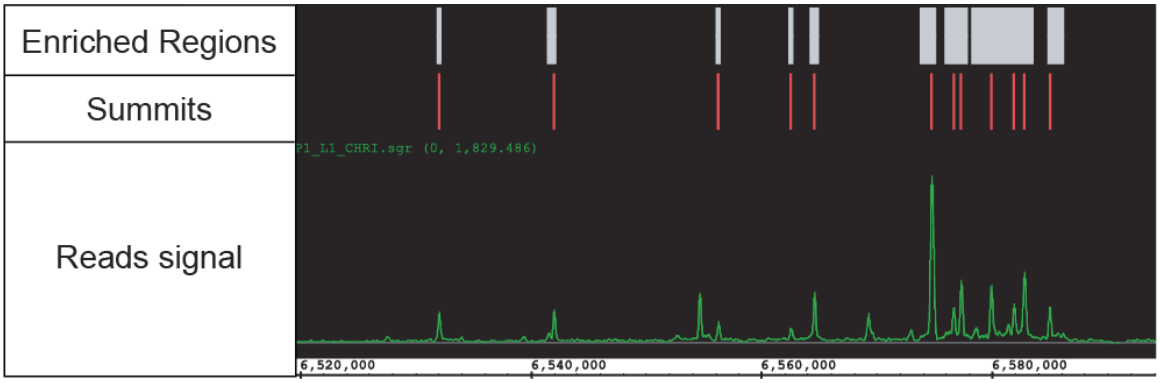

Figure 4-1 PeakRanger result for region 6520000-6600000 of chromosome I of dataset for C.elegans transcription factor BLMP-1. PeakRanger successfully identified most obvious summits within enriched regions without introducing any false positives.

After processing all ChIP-Seq datasets for C.elegans, I checked the results for a couple of transcription factors to confirm that PeakRanger was able to correctly distinguish closely clustered binding sites without introducing false positives. Figure 4-1shows a regional snapshot of the region starting at 6520000 for C.elegans transcription factor BLMP-1. As shown in the picture, especially in the region around 6580000, PeakRanger correctly marks clustered peaks with zero false positive peaks called. At the same region of 6580000, two tiny peaks were not reported by PeakRanger. The reason for this is that compared to other called summits within this regions, these two peaks do not possess enough confidences as peaks instead of noises.

4.1.2 IDR analysis and quality control of binding sites

For experiments with replicates, an additional quality control process is applied. The primary procedure involved is the irreproducibility discovery rate (IDR) analysis [114]. The IDR

analysis evaluates the reads coverage signal as well as the reproducibility of the signals in each of the replicates, thus giving a score to measure the reliability of the peaks. The IDR score is designed to resemble FDR value and a smaller IDR value represents higher quality threshold.

Dataset	# peaks in Rep 1	# peaks in Rep 2	% Reproducible in intersection	Corr in reproducible component	# peaks at 1% IDR	# peaks at 5% IDR	# peaks at 25% IDR
caudal_1	54	780	59%	81%	0	1	24
caudal_2	127	1536	64%	60%	0	1	51
ctcf500	2137	2149	76%	99%	1278	1505	1964

Table 4-1 Sample IDR analysis results for CTCF and Caudal.

For all datasets in D.melanogaster and C.elegans, IDR analysis was performed for those with replicated datasets. In Table 4-1, two sample IDR analysis results are shown. The CTCF500 dataset is of much higher quality than both caudal_1 and caudal_2, as indicated by the 99% correlation value in reproducible component as well as the 70% peaks that passed the IDR 5% threshold. The reason for low portions of IDR-passing peaks varies. For the caudal datasets shown here, the major reason is that the number of called peaks in the replicates departs from each other. It is possible the quality of replicate 1 of the caudal dataset is questionable so that it produced only a few peaks. For the datasets like caudal that didn't pass the IDR analysis, they were further examined by the data production group of modENCODE and were corrected later.

4.1.3 Data organization and processing

All datasets were processed using the computational cluster provided by Dr. Robert Grossman from the University of Illinois at Chicago (UIC). The results are also stored in the same cluster. The modENCODE DCC also has a dedicated website to provide access to these results: http://www.modencode.org/ .

Various processing pipelines were assembled to automate the processing procedure. In particular, these pipelines were fed with datasets. Aligned reads were then generated by Bowtie, the second generation Burrows-Wheeler Transform (BWT) based reads aligner. The configuration of Bowtie used default values except for the treatment for reads that map to multiple locations. Only reads that were uniquely mapped to the reference genome were kept for downstream analysis. PeakRanger was then used to call peaks using aligned reads. If needed, PeakRanger was applied to replicates. IDR analysis was then carried out for replicates. Multiple instances of these pipelines were usually running simultaneously in the cluster, based on the private cloud environment in the UIC cluster.

4.2 Characterization of doublet peaks

Since closely cluster peaks have been observed during the data processing, these peaks, which were named as doublet peaks, were further analyzed.

4.2.1 Background

There have been few papers investigating the properties of these closely spaced binding sites. A primary reason for this is the lack of computational tools effective in identifying these special peaks. Now with PeakRanger, researchers can easily discover these doublet peaks by

configuring PeakRanger with a high-resolution setup. I have systematically identified doublet peaks using modENCODE ChIP-Seq datasets of D.melanogaster. In this section, we are going to report the efforts in characterizing these doublet peaks. In particular, we claim that these doublet peaks are not likely due to data artifacts and they may play certain roles different from non-doublet peaks.

4.2.2 Calling doublet peaks for fly transcription factors

We used datasets of CTCF, CBP, ORC and MCM. The selection of these factors is based on the availability of the datasets.

To identify doublet peaks, we first used PeakRanger to call peaks in a set of ChIP-Seq datasets of fly transcription factors. The parameters were tuned so that PeakRanger can identify peaks with high resolution. The number of peaks identified for each factor is comparable and is in the range 1500 ~ 2000. Of these identified peaks, we found significant portions of doublet peaks. To verify that the identified doublet peaks indeed contain more than 1 summit, we plotted the average profile around these doublet peaks using their own read coverage profiles. We also plotted the average profile for the same number of regular single summit peaks from the top group of all peaks. As shown in Figure 4-2, compared to top regular peaks, the identified doublet peaks all demonstrate a clear twin-summit pattern. The average height of these doublet peaks are around half of that of the top regular peaks. These doublet peaks also appear to span a wider region than regular peaks, up to a couple of hundred base pairs. CTCF's doublet peaks does not show a strong pattern as others, although after zooming-in the region we confirm that it indeed shows a twin-summit pattern for doublet peaks.

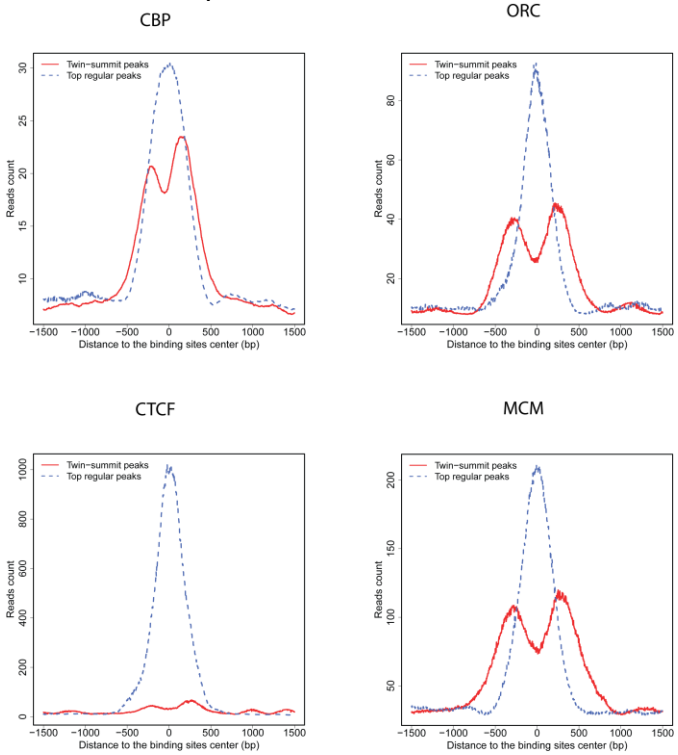

Figure 4-2 The reads profile of doublet peaks compared to regular peaks.

4.2.3 Doublet peaks are prevalent among fly transcription factors

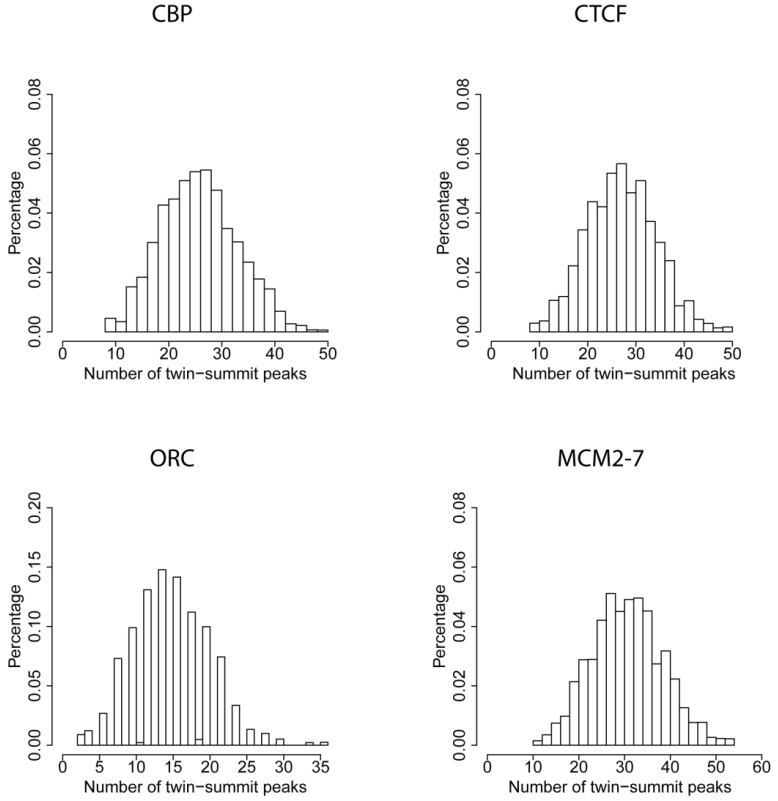

Figure 4-3 The distribution of randomly generated doublet peaks.

Transcription factor	Count of doublet peaks	Percentage of doublet peaks	P value
CBP	348	17%	< 1e-5
CTCF	190	8%	<1e-5
ORC	232	15%	<1e-5
MCM2-7	314	15%	<1e-5

Table 4-2 Statistics of the simulation results.

We found that for all tested transcription factors, up to 17% of called peaks are doublet peaks(Figure 4-3, Table 4-2). To estimate the statistical significance of these peaks, we did a simulation to count the odds of having such kind of doublet peaks. The simulation is based on genome annotations of worm and fly. In such a simulation, a fixed number of regions are randomly dispensed to upstream regions of genes and the occurrence of closely spaced peaks is counted for each such simulation. For each of transcription factors, we did such simulation and it turned out that the maximum expected portion of doublet peaks were usually below 5%, far less than the observed percentages. We thus believe that these doublet peaks are not likely data artifacts.

4.2.4 Doublet peaks are marked by PolII

To further characterize doublet peaks, we generated the genome-wide PolII profile plots of doublet peaks. The result shows that PolII demonstrates a coherent doublet peak in these transcription factor binding sites. As a comparison, we extracted the same number of peaks from

top non-doublet binding sites with highest significance. We then plotted the same profile for these non-doublet top regular peaks (Figure 4-4). As expected, only a single PolII peak is observed. This discovery supports that these doublet binding sites are likely to be biologically functional since the enrichment of PolII indicates the possibility of active gene transcriptions. CTCF is an exception to the PolII-Doublet peaks correlation, although there is a single-summit PolII in CTCF binding sites profile.

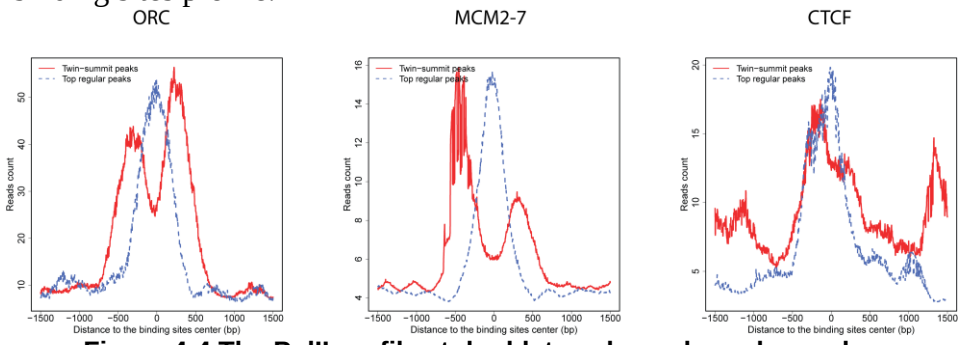

Figure 4-4 The PolII profile at doublet peaks and regular peaks.

4.2.5 Doublet peaks regions are more enriched with histones marking active promoters

We then obtained similar profile plots of various histone marks. The plot was generated for both doublet peaks and top regular peaks. As shown in Figure 4-5, the doublet peaks tend to be synergetic to cluster in much active genomic regions compared to regular singleton peaks.

All four factors show that their doublet peaks, compared to top regular peaks, are more enriched with H3K4ME3. For H3K4ME1, MCM and ORC‘s doublet peaks show a stronger valley pattern but CTCF and CBP are on the contrary. Since the peak pattern of H3K4ME3 and the valley pattern of H3K4ME1 resembles previously published prediction of promoters[18, 115], our results indicate that double binding of these transcription factors may more likely to mark promoters than top regular peaks. It is also possible that promoters marked by doublet peaks are more likely to be active than those marked by top regular peaks. To test the likelihood of the promoter-enrichment hypothesis, we measured the percentage of binding sites in each group that overlap with an annotated promoter. However, the result shows that doublet peaks of MCM and ORC do not exhibit a significantly higher overlap with promoters than regular peaks.

For CTCF and CBP, we did not observe the similar pattern. The reason may be that CTCF possess a complex combination of roles and it is not solely a marker for promoters. And since CBP is believed to be a close collaborator of CTCF, it is not surprise that its profile is similar to that of CTCF.

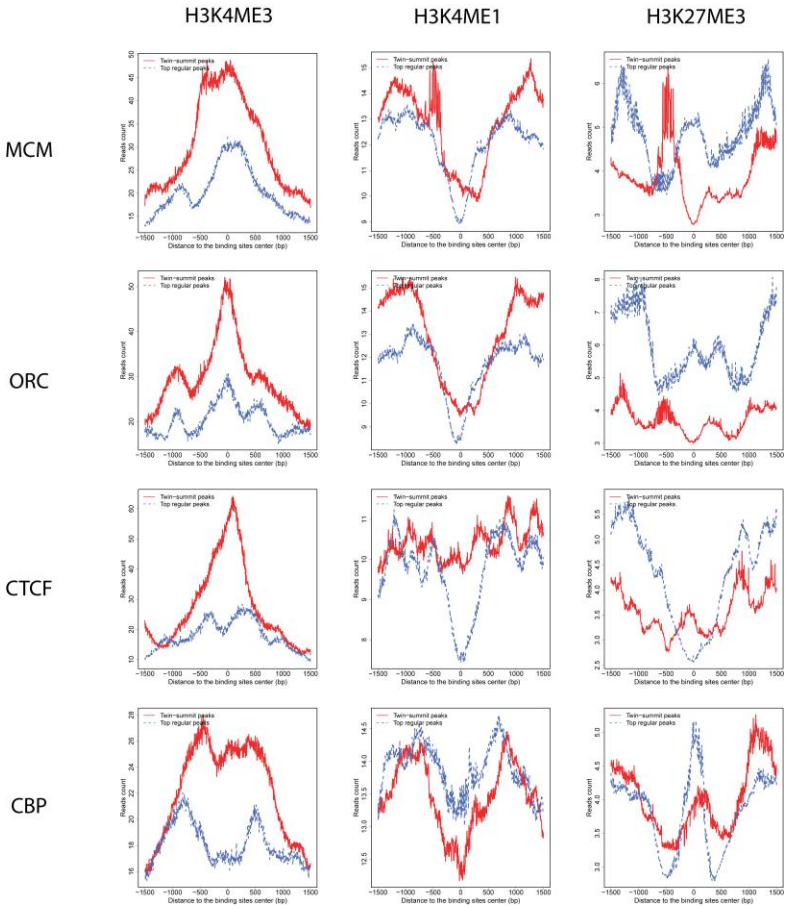

Figure 4-5 Various histone modification profiles at doublet peaks and regular peaks.

4.3 Conclusion

In this chapter, the peak calling of ChIP-Seq datasets in the modENCODE datasets is shown. In particular, the steps involved and the quality control of these steps are demonstrated. Followed by that, the properties of the doublets peaks were analyzed at the genome-scale, which should be the first report in the field. The authenticity of these doublets peaks are confirmed by their correlation with multiple other biological marks, especially the correlation with PolII. A statistical modeling of these doublet peaks shows that they are not like to be noises in terms of statistical significance. However, due to the limited datasets, the concrete biological roles of doublet peaks could not be fully identified, although there are clues that these doublet peaks may serve as synergetic marks of promoters. All these results were generated by PeakRanger and the xBED library.

Chapter 5 Conclusions

5.1 The genome-wide properties of insulator binding proteins and doublet peaks

In this thesis, I have demonstrated the analysis of the genome-scale distribution of insulator binding proteins (IBP) and doublet peaks. The analysis is based on the datasets generated by the modENCODE project and proves fruitful. In particular, the relationship between IBP and active chromatin and genes is analyzed. The results confirmed many previous conclusions obtained in local analysis but also reveals that the functions of IBP at the global scale are more complex than it is thought previously. On the other hand, the identification of doublet peaks for some D.melanogaster transcription factors demonstrates again the utility of genome-wide analysis. This type of clustered peaks has not yet been analyzed at the genome-scale before due to the lack of appropriate tools. The exciting discovery that PolII and H3K4Me3 are coherently enriched at the regions of doublet peaks indicates that these special peak clusters may confer synergetic effects on transcriptions.

5.2 PeakRanger greatly contributes to the identification of global transcription factor binding profiles

The development of PeakRanger and its application to the modENCODE project has proved a great success. PeakRanger demonstrated its ultra-high resolution in calling peaks from ChIP-Seq datasets, which brings almost zero false positive peaks. The work of PeakRanger has been included in two Science papers [116, 117].

5.3 Future work

In despite of the problems I have addressed in this thesis, many other problems remain and it is expected that more problems will emerge with the increasing popularity of genome-wide analysis of transcription factors and histone modifications. On the other hand, the characterized IBP and doublet peaks still have more to discover. I present a few such problems below.

5.3.1 The classification of CTCF binding sites

It has been shown that CTCF cannot be clearly classified as a specific category of transcription factors. Based on the result shown in this thesis, CTCF binding sites may enrich in promoters and active chromatins; CTCF sometimes also bind to inactive regions. CTCF is also present in binding sites where binding of CP190 is absent. The flexibility of CTCF binding sites indicate that CTCF possess a complex combination of biological roles in regulating gene transcriptions and chromatin structures. To further classify subsets of CTCF binding sites based on their distinct biological functions, we can integrate CTCF binding sites with its close collaborators, not limited to other insulator binding proteins. The increasing completeness of co-binding patterns of CTCF and other transcription factors will help identify the mechanisms of CTCF's functions.

5.3.2 Assignment of target genes of transcription factors

The current practice to determine the target genes of transcription factors is simply to find the genes that have upstream or downstream binding sites within a distance limit. Although this

method sounds reasonable given the fact that the regulation of genes involves binding of promoters, it does not fully capture the regulatory systems well. A particular example is the effect of binding to enhancers, which can exhibit effects even on genes located in different chromosomes. It is also possible that a group of distant binding sites collaborate in the regulation of a common gene. This task may require the genome-wide structural datasets of chromatin, such as the chromatin conformation capture [118] datasets.

References

1. Widom J: **STRUCTURE, DYNAMICS, AND FUNCTION OF CHROMATIN IN VITRO**. *Annual Review of Biophysics and Biomolecular Structure* 1998, **27**(1):285-327.
2. Grant P: **A tale of histone modifications**. *Genome Biology* 2001, **2**(4):reviews0003.0001 - reviews0003.0006.
3. Kornberg RD, Lorch Y: **Twenty-Five Years of the Nucleosome, Fundamental Particle of the Eukaryote Chromosome**. *Cell* 1999, **98**(3):285-294.
4. Huisinga K, Brower-Toland B, Elgin S: **The contradictory definitions of heterochromatin: transcription and silencing**. *Chromosoma* 2006, **115**(2):110-122.
5. Grewal SIS, Jia S: **Heterochromatin revisited**. *Nat Rev Genet* 2007, **8**(1):35-46.
6. Ward WS: **The structure of the sleeping genome: implications of sperm DNA organization for somatic cells**. *J Cell Biochem* 1994, **55**(1):77-82.
7. Craig JM: **Heterochromatin—many flavours, common themes**. *BioEssays* 2005, **27**(1):17-28.
8. Lohe AR, Hilliker AJ, Roberts PA: **Mapping Simple Repeated DNA Sequences in Heterochromatin of Drosophila melanogaster**. *Genetics* 1993, **134**(4):1149-1174.
9. Pidoux AL, Allshire RC: **The role of heterochromatin in centromere function**. *Philosophical Transactions of the Royal Society B: Biological Sciences* 2005, **360**(1455):569-579.
10. Talbert PB, Henikoff S: **Spreading of silent chromatin: inaction at a distance**. *Nat Rev Genet* 2006, **7**(10):793-803.
11. Kouzarides T: **Chromatin Modifications and Their Function**. *Cell* 2007, **128**(4):693-705.
12. Clayton AL, Hazzalin CA, Mahadevan LC: **Enhanced Histone Acetylation and Transcription: A Dynamic Perspective**. *Molecular Cell* 2006, **23**(3):289-296.
13. Schuettengruber B, Chourrout D, Vervoort M, Leblanc B, Cavalli G: **Genome Regulation by Polycomb and Trithorax Proteins**. *Cell* 2007, **128**(4):735-745.
14. Murray K: **The occurence of epsilon-N-methyl lysine in histones**. *Biochemistry* 1964, **3**:10 - 15.
15. Chen D, Ma H, Hong H, Koh S, Huang S, Schurter B, Aswad D, Stallcup M: **Regulation of transcription by a protein methyltransferase**. *Science* 1999, **284**:2174 - 2177.
16. Barski A, Cuddapah S, Cui K, Roh TY, Schones DE, Wang Z, Wei G, Chepelev I, Zhao K: **High-resolution profiling of histone methylations in the human genome**. *Cell* 2007, **129**(4):823-837.
17. He HH, Meyer CA, Shin H, Bailey ST, Wei G, Wang Q, Zhang Y, Xu K, Ni M, Lupien M *et al*: **Nucleosome dynamics define transcriptional enhancers**. *Nat Genet* 2010, **42**(4):343-347.

18. Heintzman ND, Hon GC, Hawkins RD, Kheradpour P, Stark A, Harp LF, Ye Z, Lee LK, Stuart RK, Ching CW *et al*: **Histone modifications at human enhancers reflect global cell-type-specific gene expression**. *Nature* 2009, **459**(7243):108-112.
19. Shiama N: **The p300/CBP family: integrating signals with transcription factors and chromatin**. *Trends in Cell Biology* 1997, **7**(6):230-236.
20. Wang Z, Zang C, Rosenfeld JA, Schones DE, Barski A, Cuddapah S, Cui K, Roh T-Y, Peng W, Zhang MQ *et al*: **Combinatorial patterns of histone acetylations and methylations in the human genome**. *Nat Genet* 2008, **40**(7):897-903.
21. Shahbazian MD, Grunstein M: **Functions of Site-Specific Histone Acetylation and Deacetylation**. *Annual Review of Biochemistry* 2007, **76**(1):75-100.
22. Brownell JE, Zhou J, Ranalli T, Kobayashi R, Edmondson DG, Roth SY, Allis CD: **Tetrahymena Histone Acetyltransferase A: A Homolog to Yeast Gcn5p Linking Histone Acetylation to Gene Activation**. *Cell* 1996, **84**(6):843-851.
23. Das C, Lucia MS, Hansen KC, Tyler JK: **CBP/p300-mediated acetylation of histone H3 on lysine 56**. *Nature* 2009, **459**(7243):113-117.
24. Roh TY, Cuddapah S, Zhao K: **Active chromatin domains are defined by acetylation islands revealed by genome-wide mapping**. *Genes Dev* 2005, **19**:542-552.
25. Roh TY, Wei G, Farrell CM, Zhao K: **Genome-wide prediction of conserved and nonconserved enhancers by histone acetylation patterns**. *Genome Res* 2007, **17**:74-81.
26. Rada-Iglesias A: **A unique chromatin signature uncovers early developmental enhancers in humans**. *Nature* 2011, **470**:279-283.
27. Creyghton MP: **Histone H3K27ac separates active from poised enhancers and predicts developmental state**. *Proc Natl Acad Sci USA* 2010, **107**:21931-21936.
28. Ekwall K, Olsson T, Turner B, Cranston G, Allshire R: **Transient inhibition of histone deacetylation alters the structural and functional imprint at fission yeast centromeres**. *Cell* 1997, **91**:1021 - 1032.
29. Negre N, Brown CD, Ma L, Bristow CA, Miller SW, Wagner U, Kheradpour P, Eaton ML, Loriaux P, Sealfon R *et al*: **A cis-regulatory map of the Drosophila genome**. *Nature* 2011, **471**(7339):527-531.
30. Nowak SJ, Corces VG: **Phosphorylation of histone H3: a balancing act between chromosome condensation and transcriptional activation**. *Trends in Genetics* 2004, **20**(4):214-220.
31. Shilatifard A: **Chromatin Modifications by Methylation and Ubiquitination: Implications in the Regulation of Gene Expression**. *Annual Review of Biochemistry* 2006, **75**(1):243-269.
32. Patikoglou G, Burley SK: **EUKARYOTIC TRANSCRIPTION FACTOR-DNA COMPLEXES**. *Annual Review of Biophysics and Biomolecular Structure* 1997, **26**(1):289-325.
33. Lifton RP, Goldberg ML, Karp RW, Hogness DS: **The organization of the histone genes in Drosophila melanogaster: functional and evolutionary implications**. *Cold Spring Harb Symp Quant Biol* 1978, **42 Pt 2**:1047-1051.
34. Sandelin A, Carninci P, Lenhard B, Ponjavic J, Hayashizaki Y, Hume DA: **Mammalian RNA polymerase II core promoters: insights from genome-wide studies**. *Nat Rev Genet* 2007, **8**(6):424-436.
35. Ong C-T, Corces VG: **Enhancer function: new insights into the regulation of tissue-specific gene expression**. *Nat Rev Genet* 2011, **12**(4):283-293.

36. Blackwood EM, Kadonaga JT: **Going the Distance: A Current View of Enhancer Action**. *Science* 1998, **281**(5373):60-63.
37. Raab JR, Kamakaka RT: **Insulators and promoters: closer than we think**. *Nat Rev Genet* 2010, **11**(6):439-446.
38. Farnham PJ: **Insights from genomic profiling of transcription factors**. *Nat Rev Genet* 2009, **10**(9):605-616.
39. Moon H, Filippova G, Loukinov D, Pugacheva E, Chen Q, Smith ST, Munhall A, Grewe B, Bartkuhn M, Arnold R *et al*: **CTCF is conserved from Drosophila to humans and confers enhancer blocking of the Fab-8 insulator**. *EMBO Rep* 2005, **6**(2):165-170.
40. Phillips JE, Corces VG: **CTCF: Master Weaver of the Genome**. *Cell* 2009, **137**(7):1194-1211.
41. Valenzuela L, Kamakaka RT: **Chromatin Insulators**. *Annual Review of Genetics* 2006, **40**(1):107-138.
42. Vaquerizas JM, Kummerfeld SK, Teichmann SA, Luscombe NM: **A census of human transcription factors: function, expression and evolution**. *Nat Rev Genet* 2009, **10**(4):252-263.
43. Babu MM, Luscombe NM, Aravind L, Gerstein M, Teichmann SA: **Structure and evolution of transcriptional regulatory networks**. *Current Opinion in Structural Biology* 2004, **14**(3):283-291.
44. Ng SB, Bigham AW, Buckingham KJ, Hannibal MC, McMillin MJ, Gildersleeve HI, Beck AE, Tabor HK, Cooper GM, Mefford HC *et al*: **Exome sequencing identifies MLL2 mutations as a cause of Kabuki syndrome**. *Nat Genet* 2010, **42**(9):790-793.
45. Cooper SJ, Trinklein ND, Anton ED, Nguyen L, Myers RM: **Comprehensive analysis of transcriptional promoter structure and function in 1% of the human genome**. *Genome Res* 2006, **16**:1-10.
46. Gershenzon NI, Ioshikhes IP: **Synergy of human Pol II core promoter elements revealed by statistical sequence analysis**. *Bioinformatics* 2005, **21**:1295-1300.
47. Ohler U: **Identification of core promoter modules in Drosophila and their application in accurate transcription start site prediction**. *Nucleic Acids Res* 2006, **34**:5943-5950.
48. Ohler U, Liao GC, Niemann H, Rubin GM: **Computational analysis of core promoters in the Drosophila genome**. *Genome Biol* 2002, **3**.
49. Molina C, Grotewold E: **Genome wide analysis of Arabidopsis core promoters**. *BMC Genomics* 2005, **6**:25.
50. **Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project**. *Nature* 2007, **447**:799-816.
51. Celniker SE: **Unlocking the secrets of the genome**. *Nature* 2009, **459**:927-930.
52. Collas P: **The Current State of Chromatin Immunoprecipitation**. *Molecular Biotechnology* 2010, **45**(1):87-100.
53. Ren B: **Genome-wide location and function of DNA binding proteins**. *Science* 2000, **290**:2306-2309.
54. Park PJ: **ChIP-seq: advantages and challenges of a maturing technology**. *Nat Rev Genet* 2009, **10**(10):669-680.
55. Pepke S, Wold B, Mortazavi A: **Computation for ChIP-seq and RNA-seq studies**. *Nat Meth* 2009, **6**(11s):S22-S32.

56. Lin H, Zhang Z, Zhang M, Ma B, Li M: **ZOOM! Zillions Of Oligos Mapped**. *Bioinformatics* 2008, **24**:2431 - 2437.
57. Li H, Ruan J, Durbin R: **Mapping short DNA sequencing reads and calling variants using mapping quality scores**. *Genome Res* 2008, **18**:1851 - 1858.
58. Li R, Li Y, Kristiansen K, Wang J: **SOAP: short oligonucleotide alignment program**. *Bioinformatics* 2008, **24**:713 - 714.
59. Smith A, Xuan Z, Zhang M: **Using quality scores and longer reads improves accuracy of Solexa read mapping**. *BMC Bioinformatics* 2008, **9**:128.
60. Rumble SM: **SHRiMP: accurate mapping of short color-space reads**. *PLoS Comput Biol* 2009, **5**:e1000386.
61. Langmead B, Trapnell C, Pop M, Salzberg S: **Ultrafast and memory-efficient alignment of short DNA sequences to the human genome**. *Genome Biology* 2009, **10**(3):R25.
62. Li H, Durbin R: **Fast and accurate short read alignment with Burrows–Wheeler transform**. *Bioinformatics* 2009, **25**(14):1754-1760.
63. Li R, Yu C, Li Y, Lam T-W, Yiu S-M, Kristiansen K, Wang J: **SOAP2: an improved ultrafast tool for short read alignment**. *Bioinformatics* 2009, **25**(15):1966-1967.
64. Kharchenko PV, Tolstorukov MY, Park PJ: **Design and analysis of ChIP-seq experiments for DNA-binding proteins**. *Nature Biotech* 2008, **26**:1351-1359.
65. Rozowsky J: **PeakSeq enables systematic scoring of ChIP-seq experiments relative to controls**. *Nature Biotech* 2009, **27**:66-75.
66. Zhang Y: **Model-based analysis of ChIP-seq (MACS)**. *Genome Biol* 2008, **9**:R137.
67. Bailey TL, Elkan C: **Fitting a mixture model by expectation maximization to discover motifs in biopolymers**. *Proc Int Conf Intell Syst Mol Biol* 1994, **2**:28-36.
68. Bailey TL, Gribskov M: **Combining evidence using p-values: application to sequence homology searches**. *Bioinformatics* 1998, **14**(1):48-54.
69. Stein L: **The case for cloud computing in genome informatics**. *Genome Biology* 2010, **11**(5):207.
70. Armbrust M, Fox A, Griffith R, Joseph AD, Katz RH, Konwinski A, Lee G, Patterson DA, Rabkin A, Stoica I *et al*: **Above the Clouds: A Berkeley View of Cloud Computing**. In.: EECS Department, University of California, Berkeley; 2009.
71. Gaszner M, Felsenfeld G: **Insulators: exploiting transcriptional and epigenetic mechanisms**. *Nat Rev Genet* 2006, **7**(9):703-713.
72. Cuddapah S, Jothi R, Schones DE, Roh T-Y, Cui K, Zhao K: **Global analysis of the insulator binding protein CTCF in chromatin barrier regions reveals demarcation of active and repressive domains**. *Genome Research* 2009, **19**(1):24-32.
73. Lobanenkov VV, Nicolas RH, Adler VV, Paterson H, Klenova EM, Polotskaja AV, Goodwin GH: **A novel sequence-specific DNA binding protein which interacts with three regularly spaced direct repeats of the CCCTC-motif in the 5'-flanking sequence of the chicken c-myc gene**. *Oncogene* 1990, **5**(12):1743-1753.
74. Vostrov AA, Quitschke WW: **The Zinc Finger Protein CTCF Binds to the APBbeta Domain of the Amyloid beta -Protein Precursor Promoter. EVIDENCE FOR A ROLE IN TRANSCRIPTIONAL ACTIVATION**. *J Biol Chem* 1997, **272**(52):33353-33359.
75. Filippova GN, Cheng MK, Moore JM, Truong J-P, Hu YJ, Di Kim N, Tsuchiya KD, Disteche CM: **Boundaries between Chromosomal Domains of X Inactivation and**

Escape Bind CTCF and Lack CpG Methylation during Early Development. *Developmental Cell* 2005, **8**(1):31-42.

76. Kanduri C, Pant V, Loukinov D, Pugacheva E, Qi C-F, Wolffe A, Ohlsson R, Lobanenkov VV: **Functional association of CTCF with the insulator upstream of the H19 gene is parent of origin-specific and methylation-sensitive**. *Current Biology* 2000, **10**(14):853-856.
77. Holohan EE, Kwong C, Adryan B, Bartkuhn M, Herold M, Renkawitz R, Russell S, White R: **CTCF Genomic Binding Sites in *Drosophila* and the Organisation of the Bithorax Complex**. *PLoS Genet* 2007, **3**(7):e112.
78. Henikoff S, Henikoff JG, Sakai A, Loeb GB, Ahmad K: **Genome-wide profiling of salt fractions maps physical properties of chromatin**. *Genome Research* 2009, **19**(3):460-469.
79. Smith ST, Wickramasinghe P, Olson A, Loukinov D, Lin L, Deng J, Xiong Y, Rux J, Sachidanandam R, Sun H *et al*: **Genome wide ChIP-chip analyses reveal important roles for CTCF in Drosophila genome organization**. *Dev Biol* 2009, **328**(2):518-528.
80. Tweedie S, Ashburner M, Falls K, Leyland P, McQuilton P, Marygold S, Millburn G, Osumi-Sutherland D, Schroeder A, Seal R *et al*: **FlyBase: enhancing Drosophila Gene Ontology annotations**. *Nucl Acids Res* 2008:gkn788.
81. Bushey AM, Ramos E, Corces VG: **Three subclasses of a Drosophila insulator show distinct and cell type-specific genomic distributions**. *Genes & Development* 2009, **23**(11):1338-1350.
82. Raychaudhuri S, Stuart JM, Altman RB: **Principal components analysis to summarize microarray experiments: application to sporulation time series**. *Pac Symp Biocomput* 2000:455-466.
83. Michael Ashburner CAB, Judith A. Blake , David Botstein , Heather Butler , J. Michael Cherry , Allan P. Davis , Kara Dolinski , Selina S. Dwight , Janan T. Eppig , Midori A. Harris , David P. Hill , Laurie Issel-Tarver , Andrew Kasarskis , Suzanna Lewis , John C. Matese , Joel E. Richardson , Martin Ringwald , Gerald M. Rubin & Gavin Sherlock **Gene Ontology: tool for the unification of biology**. *Nature Genet* 2000, **25**:25-29.
84. Mikkelsen TS, Ku M, Jaffe DB, Issac B, Lieberman E, Giannoukos G, Alvarez P, Brockman W, Kim TK, Koche RP *et al*: **Genome-wide maps of chromatin state in pluripotent and lineage-committed cells**. *Nature* 2007, **448**(7153):553-560.
85. Johnson DS, Mortazavi A, Myers RM, Wold B: **Genome-wide mapping of in vivo protein-DNA interactions**. *Science* 2007, **316**(5830):1497-1502.
86. Robertson G, Hirst M, Bainbridge M, Bilenky M, Zhao Y, Zeng T, Euskirchen G, Bernier B, Varhol R, Delaney A: **Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing**. *Nat Methods* 2007, **4**:651 - 657.
87. Iyer VR, Horak CE, Scafe CS, Botstein D, Snyder M, Brown PO: **Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF**. *Nature* 2001, **409**(6819):533-538.
88. Lun D, Sherrid A, Weiner B, Sherman D, Galagan J: **A blind deconvolution approach to high-resolution mapping of transcription factor binding sites from ChIP-seq data**. In., vol. 10; 2009: R142.

89. Blahnik KR, Dou L, O'Geen H, McPhillips T, Xu X, Cao AR, Iyengar S, Nicolet CM, Lud 盲 scher B, Korf I *et al*: **Sole-Search: an integrated analysis program for peak detection and functional annotation using ChIP-seq data**. In., vol. 38; 2009: e13.
90. Ji H, Jiang H, Ma W, Johnson D, Myers R, Wong W: **An integrated software system for analyzing ChIP-chip and ChIP-seq data**. *Nat Biotechnol* 2008, **26**:1293 - 1300.
91. Jothi R, Cuddapah S, Barski A, Cui K, Zhao K: **Genome-wide identification of in vivo protein-DNA binding sites from ChIP-Seq data**. *Nucleic Acids Res* 2008, **36**:5221 - 5231.
92. Zang CZ, Schones DE, Zeng C, Cui KR, Zhao KJ, Peng WQ: **A clustering approach for identification of enriched domains from histone modification ChIP-Seq data**. *Bioinformatics* 2009, **25**(15):1952-1958.
93. Fejes A, Robertson G, Bilenky M, Varhol R, Bainbridge M, Jones S: **FindPeaks 3.1: a tool for identifying areas of enrichment from massively parallel short-read sequencing technology**. *Bioinformatics* 2008, **24**:1729 - 1730.
94. Boyle AP, Guinney J, Crawford GE, Furey TS: **F-Seq: a feature density estimator for high-throughput sequence tags**. *Bioinformatics* 2008, **24**(21):2537-2538.
95. Tuteja G, White P, Schug J, Kaestner KH: **Extracting transcription factor targets from ChIP-Seq data**. *Nucleic Acids Res* 2009, **37**(17).
96. Valouev A, Johnson D, Sundquist A, Medina C, Anton E, Batzoglou S, Myers R, Sidow A: **Genome-wide analysis of transcription factor binding sites based on ChIP-Seq data**. *Nat Methods* 2008, **5**:829 - 834.
97. Nix D, Courdy S, Boucher K: **Empirical methods for controlling false positives and estimating confidence in ChIP-Seq peaks**. *BMC Bioinformatics* 2008, **9**:523.
98. Guo Y, Papachristoudis G, Altshuler RC, Gerber GK, Jaakkola TS, Gifford DK, Mahony S: **Discovering homotypic binding events at high spatial resolution**. *Bioinformatics* 2010.
99. Mortazavi A, Williams BA, McCue K, Schaeffer L, Wold B: **Mapping and quantifying mammalian transcriptomes by RNA-Seq**. *Nat Meth* 2008, **5**(7):621-628.
100. Qin Z, Yu J, Shen J, Maher C, Hu M, Kalyana-Sundaram S, Yu J, Chinnaiyan A: **HPeak: an HMM-based algorithm for defining read-enriched regions in ChIP-Seq data**. *BMC Bioinformatics* 2010, **11**(1):369.
101. Jeffrey Dean SG: **MapReduce: Simplified Data Processing on Large Clusters** In: *OSDI'04: Sixth Symposium on Operating System Design and Implementation.* San Francisco, CA; 2004.
102. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, Genome Project Data Processing Subgroup: **The Sequence Alignment/Map format and SAMtools**. *Bioinformatics* 2009, **25**(16):2078-2079.
103. Ramsey SA, Knijnenburg TA, Kennedy KA, Zak DE, Gilchrist M, Gold ES, Johnson CD, Lampano AE, Litvak V, Navarro G *et al*: **Genome-wide histone acetylation data improve prediction of mammalian transcription factor binding sites**. *Bioinformatics* 2010:btq405.
104. **Hadoop** [http://hadoop.apache.org/]
105. Schatz MC: **CloudBurst: highly sensitive read mapping with MapReduce**. *Bioinformatics* 2009, **25**(11):1363-1369.

106. Langmead B, Schatz M, Lin J, Pop M, Salzberg S: **Searching for SNPs with cloud computing**. *Genome Biology* 2009, **10**(11):R134.
107. Langmead B, Hansen K, Leek J: **Cloud-scale RNA-sequencing differential expression analysis with Myrna**. *Genome Biology* 2010, **11**(8):R83.
108. Quinlan AR, Hall IM: **BEDTools: a flexible suite of utilities for comparing genomic features**. *Bioinformatics* 2010, **26**(6):841-842.
109. Shin H, Liu T, Manrai AK, Liu XS: **CEAS: cis-regulatory element annotation system**. *Bioinformatics* 2009, **25**(19):2605-2606.
110. Wilbanks EG, Facciotti MT: **Evaluation of Algorithm Performance in ChIP-Seq Peak Detection**. *PLoS ONE* 2010, **5**(7):e11471.
111. Wingender E, Dietze P, Karas H, Knüppel R: **TRANSFAC: A Database on Transcription Factors and Their DNA Binding Sites**. In., vol. 24; 1996: 238-241.
112. Team RDC: **R: A Language and Environment for Statistical Computing**; 2008.
113. Nicol JW, Helt GA, Blanchard SG, Raja A, Loraine AE: **The Integrated Genome Browser: free software for distribution and exploration of genome-scale datasets**. In., vol. 25; 2009: 2730-2731.
114. Qunhua Li JBB, Haiyan Huang and Peter J. Bickel: **MEASURING REPRODUCIBILITY OF HIGH-THROUGHPUT EXPERIMENTS**. *Annals of Applied Statistics* 2011.
115. Visel A: **ChIP-seq accurately predicts tissue-specific activity of enhancers**. *Nature* 2009, **457**:854-858.
116. Gerstein MB, Lu ZJ, Van Nostrand EL, Cheng C, Arshinoff BI, Liu T, Yip KY, Robilotto R, Rechtsteiner A, Ikegami K *et al*: **Integrative Analysis of the Caenorhabditis elegans Genome by the modENCODE Project**. *Science* 2010, **330**(6012):1775-1787.
117. Consortium Tm, Roy S, Ernst J, Kharchenko PV, Kheradpour P, Negre N, Eaton ML, Landolin JM, Bristow CA, Ma L *et al*: **Identification of Functional Elements and Regulatory Circuits by Drosophila modENCODE**. *Science* 2010, **330**(6012):1787-1797.
118. Dekker J, Rippe K, Dekker M, Kleckner N: **Capturing Chromosome Conformation**. *Science* 2002, **295**(5558):1306-1311.