INFERRING REGULATORY SIGNAL FROM GENOMIC DATA

VINSENSIUS BERLIAN VEGA S N
(B.Sc. (Hons. 1), M.Sc., NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

NATIONAL UNIVERSITY OF SINGAPORE

2008

ACKNOWLEDGEMENTS

I am greatly indebted to Dr. Sung Wing-Kin for being my supervisor in this project. He has been unyielding in providing me with guidance and inspiration. Our many invaluable discussions helped me significantly to navigate through the research process. I extend my utmost gratitude for his constant encouragement and support.

I am grateful to Dr. Edison Liu Tak-Bun for all the invaluable comments, pointers, and support that he gave. I would also like to thank Dr. Philip M. Long and Dr. Karuturi Radha Krishna Murthy for the many great discussions and collaborations.

Many thanks to my colleagues at the Genome Institute of Singapore for their helpful comments and inputs, especially for the biological insights which I would have not obtained otherwise.

TABLE OF CONTENTS

SUMMARY

The recent rapid growth of biological data opens a whole range of exciting possibilities for and necessitates development of data mining methods tailored towards understanding the complex mechanisms of biological systems. Bioinformatics has gone from providing support, in terms of data management, visualization, and such, to generating new insights and directing future experiments. One key topic in molecular biology is the understanding the regulatory process and mechanism of gene expression.

This project focuses on addressing issues related to gene expression regulation, namely identification of relevant or responsive genes from microarray data and analysis of sequencing-based localization of interaction sites of transcription factor (TF) and DNA.

We began by creating a model for complex system which accounts for intricate relationships between the observable input and output data as well as the potential noise that confound both the input and the output. In the context of gene regulation, the inputs are genomic sequences and genomic signals while the output is gene expression. We then decouple the analysis of input, i.e. distilling genomic signals, and output, i.e. identifying relevant and responsive genes.

On the output front, we focused on analyzing microarray data. The first task was to develop a method that would identify a minimal gene signature cassette, a problem

which we translated as determining robust and non-redundant set of genes for classification. A key modification of the well-known boosting framework was found to satisfy the requirement and also outperform the widely successful support vector machine (SVM). The second task was to better utilize time-course expression data to identify primary response genes caused by an external stimulant. The presence of indirectly influenced genes made the problem difficult. Rather than attempting to rank genes based on their own predictive power or expression pattern, we explored the notion of primary response and indirect response. We devised the Friendly Neighbor framework that exploits the relationship between primary response and other downstream response. Genes were assessed based on their shared expression dynamics, rather than their individual profiles. A pair of genes was said to be "friends" if their expression dynamics are similar. Each gene was then scored based on the number of genes that were "friendly" to it. Genes with higher scores were more likely to be primary responders. Our experiments showed that the shared expression dynamics property indeed helped to propel the performance of unsupervised identification of primary response genes to much closer to the performance of supervised algorithms.

In terms of genomic signals, we researched on models and methods to decipher high-throughput sequencing-based TF-DNA interaction data. In particular, we started by devising a simple formula to assess the sequencing adequacy of a given library. The formula can be used to obtain a relative estimate of the sequencing saturation. Leveraging on the unique characteristic of ChIP-PET, we proposed a new model for ChIP fragment size distribution. This model worked well on all the test libraries and outperformed the earlier model. We developed a model of fragment enrichment that

attempts to parameterize the quality of the dataset and the extent of actual TF-DNA interactions. Genomic regions were analyzed in terms of clusters of overlapping fragments. An analytical model of random fragment accumulation under random uniform distribution was constructed, where the probability of generating a cluster of size *n* by chance alone was $\left(1-e^{-\lambda k}\right)^{(n-1)}$ and the probability of initiating such a cluster was $\frac{e^{-\lambda k}(\lambda k)^{(n-1)}}{(n-1)!}$. This model allowed for more precise computation of *p*-value and thus more efficient and principled identification of TF-DNA interaction regions. A sliding-window based extension was also proposed to mitigate systematic biases in the data arising from aberrant genomic copy number of the underlying biological model system. Experimental results demonstrate the accuracy of our analytical models, for assessing library quality and calculating chance accumulation probability, and the effectiveness of the adaptive method, in reducing false positive identifications of TF-DNA interaction regions.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

The field of bioinformatics has grown rapidly in the recent years, producing a multitude of computational tools and offering new insights. The vast amount and rapid growth of biological data and databases, while remain a major reason for the need of bioinformatics, is no longer its main reason of existence. More computational analysis methods developed went beyond data management, organization, and manipulation (e.g. efficient storing and fast searching) and ventured into hypothesis testing and knowledge discovery, generating new insights leading to novel or refined biological paradigms, for example the Fragile Breakage Model of genome rearrangement proposed by Pevzner and Tesler (2003).

The understanding of how genes are regulated and the knowledge of what set of complexes is affecting which group of genes are paramount in the effort of deciphering and reconstructing the molecular clockwork of cells. While the identification and discovery of the mechanisms and rules of gene regulation are accelerated by technological developments of the measuring apparatus and protocols (e.g. DNA-microarray (Schena *et al.*, 1998; Barret and Kawasaki, 2003), ChIP-chip (Iyer *et al., 2001*; Ren *et al.*, 2000), and next generation sequencing machines), the challenges and complexities are also growing in tandem. The paradigm of promoter-sufficient gene regulation, for example, worked well in lower order organisms like

yeast, but is clearly insufficient to explain the regulatory complexities found in higher order organisms. The growing body of available data related to gene regulation and expression presents an opportunity for novel theoretical inferences and hypotheses building.

1.2 Project Scope and Objectives

Although we are interested in the broad spectrum of computational analysis and prediction of gene expression and regulation, within the context of this project, we limit ourselves by partitioning the problem into two major sub-problems of regulated (or responsive) genes identification and genomic regulatory elements discovery, which are easily reframed in terms of feature selection and classification problems. This project is targeted at developing data mining methods for analyzing microarray and high-throughput genomic sequencing data. Specifically, we aim to:

1. Formulate a unified framework of gene expression and regulation analysis,
2. Design algorithms for identifying minimal and non-redundant set of gene signature from microarray data and for predicting the primary responsive genes upon treatments, and
3. Devise methodologies for analyzing sequencing-based high-throughput genome-wide transcription factor (TF) DNA interaction data.

Parts of this thesis have been published in the Machine Learning (Long and Vega, 2003), IEEE BIBE (Karuturi and Vega, 2004), PLOS Genetics (Lin, Vega, *et al.*, 2007), and the International Conference on Computational Science (Vega, Ruan, and Sung, 2008).

1.3 Report Organization

The reminder of the report is organized as follows. Chapter 2 provides the domain knowledge and outlines the overarching problems and details our proposed paradigm for delving into the problems. Background information, motivation, and problem formulations are further expounded in the chapter. Chapter 3 presents our algorithms for analyzing microarray data to identify gene signature cassettes and primary responsive genes. Chapter 4 delves into the analysis of sequencing-based TF-DNA interaction data. We conclude this report with a summary and cursory exploration of the possible future directions in Chapter 5.

Chapter 2

Models for Understanding Gene Expression and Regulation

2.1 Domain Background

This section serves as a primer on the field of molecular biology, in particular on topics that are relevant to our project. Two things are emphasized here, namely: (i) the definitions and concepts related to gene regulation, and (ii) the relevant technologies used to generate the data.

2.1.1 Gene Expression Regulation and Its Mechanism

Central Dogma of Molecular Biology

Cell is a very complex system. The three key components of living cells are DNA, RNA, and protein. *Central dogma of molecular biology* teaches us that, in all known living organisms, DNA serves as the template or the blueprint for constructing RNAs and in turn proteins (Crick, 1970; Strachan and Read, 1999; Snustad and Simmons, 2000). Proteins and ncRNA (non-coding RNA (Eddy, 1999; Eddy, 2001)), the true workhorses in cells, carry out complex cell functions, mediate molecular signaling, catalyze chemical reactions, provide structural foundation, and a number of other vital processes. DNA, on the other hand, encodes the molecular instructions for building the proteins. As the carrier of molecular instruction, DNA is also the vehicle for propagating hereditary messages during cell replication. For these reasons, many have

described DNA as informational, protein as functional, and RNA as both informational and functional.

Regulations and Expression

For the cell to have a "meaning" or state, the contents of the cell need to be controlled. Since it is impossible to control every action of every single molecule in the cell, what is being controlled is the amount of those molecules that are present within the cell. The synthesis of proteins from their DNA templates comprises *transcription* (i.e. the formation of mRNA from DNA) and *translation* (i.e. the assembly of amino acids sequences from mRNA).

A *DNA sequence* is a string of nucleic acids and is represented as a string from the alphabet set {A,C,G,T} (denoting adenine, cytosine, guanine, and thymine) written in the direction from 5'-end to 3'-end. A *genome* is the complete set of DNA sequences of an organism. At present, a genome is generally associated to a single *species*, unless specified otherwise for particular application. A *gene* is a region of the genome that can be converted into RNA. The word "gene" carries many meanings and has evolved with the development of molecular biology, ranging from the unit of hereditary to protein association (one gene one protein) to unit of transcription. In the context of this study, a gene is tied into a location in the genome and is implicitly assumed to be subject to transcription.

Strictly speaking a gene is said to be *expressed* when its corresponding final functional gene product is produced, proteins for most cases or RNAs for genes that encode functional non-coding RNAs (Eddy, 1999; Eddy, 2001).

Transcription Regulations and Transcription Factors

The process of transcription starts from the beginning of the gene (also known as the *Transcription Start Site* (TSS)). Transcription is initiated only when the RNA-polymerase, assisted by other proteins, bind to the 5'-upstream of the TSS. The binding of this transcription machinery is followed by the unwinding of DNA double helix, initiation of RNA chain, elongation of RNA, and termination of transcription by the release of RNA and RNA-polymerase. Inducement (or inhibition) of such binding leads to the increase (or decrease) in the amount of transcripts in the cell. This is how the cell regulates transcriptions. By controlling when and where the transcription complexes bind, the cell directs which genes to be transcribed and manages the amount of mRNAs present. The cell exercises its regulatory role on transcriptions through a class of proteins known as *transcription factors* (or TF for short) (Strachan and Read, 1999; Snustad and Simmons, 2000), which could both activate or repress (Gaston and Jayaraman, 2003) transcription.

To exert their regulatory roles, transcription factors (TFs) need to bind to specific segments of the DNA, known as the *transcription factor binding sites* (TFBS). The requirement of TF binding to TFBS is important and serves as a means to identify the genes that they can regulate. It would be meaningless if transcription factors could affect genes indiscriminately. The specificity of TF binding is postulated to be largely dependent on the sequence composition of a DNA fragment, which is often termed as the TF *recognition sequence* (or more popularly *binding sequence* or *binding motif*). Stated this way, computationally speaking, the location of TFBS can

be identified by searching the locations in the genome that bear good resemblance to the TF's binding sequence.

DNA binding sites are usually found in the proximal sequences of the genes, dubbed as *cis-regulatory regions*. The *cis*-regulatory region includes sequences 5' upstream and 3' downstream of the gene. Many call the 5'-upstream sequences as the *promoter region* and consider only 5' upstream sequences as the regulatory regions. It has been shown in a number of cases that regulatory sequences exist in 3' downstream of the genes, e.g. Lamb and Rizzino (1998) reported a binding site of Oct4 in the 3'-UTR (UnTranslated Region) of FGF-4 gene, and even in distal sequences.

Besides directly binding to a specific site in the genome, TF might indirectly interact with the DNA by forming a complex with other TFs or DNA-binding proteins which would in turn bind to their associated sites in the genome. Such possibility, coupled with the fact that TFBS are commonly short (and thus ubiquitous), confound sequence analysis efforts in pinning down real functional TFBS. Barraged by these uncertainties, it is the molecular dynamics of protein-DNA interactions and genomic chromatin structure that facilitates the recognition and discrimination of binding sites by their transcription factors.

2.1.2 Measurement Apparatus for High-Throughput Molecular Biology

Measuring Expression

Abundance of RNA in the cell can be quantified in many ways. mRNA microarray (Barrett and Kawasaki, 2003; Shena et. al., 1998) offers a unique advantage in terms of throughput, time, cost, and quality. mRNA microarray (or microarray for short) exploits the property that a single strand DNA hybridizes to its complementary strand to form a (more) physically and chemically stable double strand (Mulligan, 2003). A microarray contains a vast number of single strand oligonucletides (short DNA) pieces. A *probe* is a group of DNA pieces of exactly the same sequence and proximally placed on the array. Each probe is typically constructed based on the sequence of a gene. The level of RNA in the cell is detected by first converting the RNA into DNA (i.e. reverse transcribing RNA to cDNA), followed by labeling the CDNA with certain fluorescent dye, hybridizing them into the microarray, and finally reading the amount of hybridized fragment using a laser scanner. The more fragments coming from a gene, the brighter the probe associated to it will be.

Chromatin-ImmunoPrecipitation

A key technology in the study of transcription factor is the ImmunoPrecipitation (IP) assay. In brief, the IP experiment extract a certain (or certain group of) protein from a given biological sample, based on the prepared antibody. Such extraction brings with it all other compounds that form a complex with the target protein. Since transcription factors are expected to interact (i.e. form complexes) with the DNA, immobilization of such TF-DNA complexes followed by extraction of these complexes using the IP protocol allows researchers to collect DNA

where such complexes have occurred. This procedure is known as *Chromatin-ImmunoPrecipitation* (or ChIP). The ChIP procedure produces DNA fragments that are bound by the transcription factor if interest. These fragments can be further utilized for a number of applications, including: determination of TF binding motif, localization of TFBS, measurement of TF activity. In this project, we are particularly interested in its use for the localization of the TFBS through the coupling of *high-throughput sequencing*. High-throughput sequencing in this context refers to the application of sequencing technology to sequence only a fraction of each fragment in the interest of characterizing larger pool of fragments. With the availability of whole genome sequences, partial sequencing of a fragment is, in principle, sufficient to uniquely locate the source of the fragment in the genome. Additional details are given in Section 2.4 below and in Chapter 4.

2.2 Overall Problem Description and Abstraction

We are interested in the problem of determining a gene's response towards a certain stimulant, given its associated genomic sequences. More precisely, we are interested in learning and predicting the transcriptional activities of a gene (proxied by microarray readouts (Barrett and Kawasaki, 2003; Shena et. al., 1998)), with respect to a certain transcription factor, based on the gene's regulatory sequences (which are typically, but not necessarily, be the genomic DNA sequences surrounding the gene's transcription start site (TSS)).

Problem 2.1 (Predicting transcriptional activities) *Given a Transcription Factor* T*, genes' regulatory regions* $S = \{s_1,...,s_N\}$*, and their corresponding transcript readings* $R = \{r_1,...,r_N\}$ *under the stimulation of* T*, where* $s_i = \{A,C,G,T\}^*$ *and* $r_i \in \Re^n$*, learn the function* M *such that* $M(s_i) = \hat{r} \in \Re^n$ *and* $\forall i : |M(s) - r_i|$ *is minimized. Note that* s_i *here could extend beyond* s_N*, i.e.* M *should generalize well to unseen examples.*

In the above, R could be the actual expression readouts, the normalized expression readouts (e.g. expression ratio to some form of control data), or otherwise. Problem 2.1 lays out the problem in terms of measurable and collectible data, hiding several dimensions about the nature of the system. For one, it subtracts out the fact that the state of the cell, in addition to the input data s_i, plays a key role in influencing the response r_i. Gene expressions (i.e. r_i) is significantly influenced by the current state of the cell. It also folds out the interdependencies between two response readouts, r_i and r_j, and assumes that the genes are completely independent.

Also, nothing is explicitly said about the nature of the input, s_i, which in reality contains superfluous noise unrelated to the response r_i. A gene's regulatory region (s_i) can be expected to contain noise as well as other information that may not be relevant in the current state of the cell. The same is true for the response variable r_i as well. The real interest is in fact the conceptual entities, let's call them the *Control Signal* and the *Outcome*, that respectively govern the generation (or at least reflected by) of s_i and r_i. The relationship between the Control Signal and the Outcome are the actual gold. However, since those are not easily quantifiable, by mining S and R we hope to shed some light about the underlying model. Figure 1 illustrates this situation.

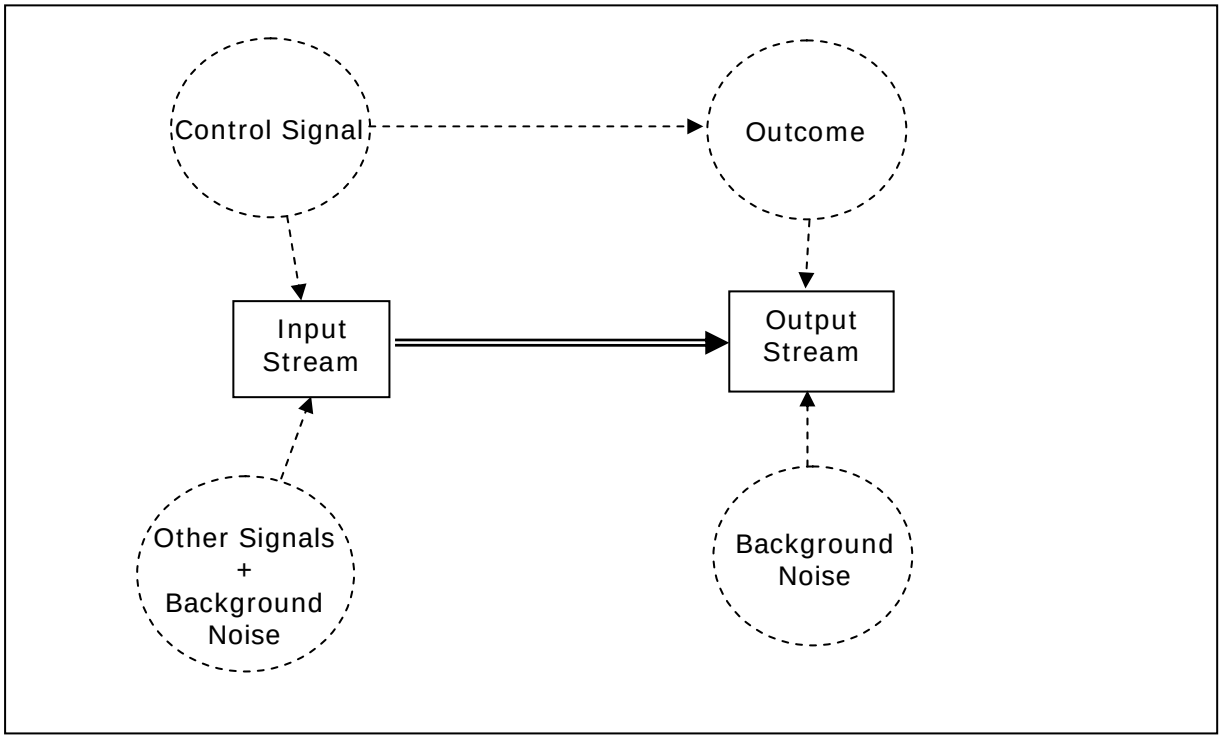


Figure 1. Modeling a complex system. Dashed shapes and arrows represent unobservable information. Solid boxes indicate known or measurable information. Solid double-line arrow indicates a simplifying assumption (that output is directly resulted from input) often taken when analyzing such data.

In the model depicted in Figure 1, only two sets of data are known: the input stream, which reflects or is generated by the Control Signal of interest coupled with other irrelevant signals and/or the background noise, and the output stream, which reflects or is generated by the true Outcome and sprinkled by the background noise. The overall goal is to learn the relationship between the control signal model and the outcome model. The model also highlights the fact that the non-direct relationship between the observed input and output streams[1], which allows for the possibility that two matching inputs, $s_i = s_j$, could yield different responses, i.e. $r_i = r_j$. Having described the intricacies of problem 2.1, we can now shape it into a more generic framework:

Problem 2.2 (Two streams framework) *Let* $S = \{s_1, \ldots, s_N\}$ *be the sequences of observed input stream and* $R = \{r_1, \ldots, r_N\}$ *be the observed sequences of corresponding output stream (or response), where* $s_i \in \Sigma_C^*$ *and* $r_i \in \Sigma_O^*$. Σ_C *and* Σ_O *denote the alphabet sets for input and output respectively. The generation of* S *is governed by an unobservable model* C*, other control signals, and systematic noise.* C *in turn influences an unobservable model* O *which governs the generation of* R*, along with some noise. The task is to learn an algorithm* M*, which given* $s_i \in \Sigma_C^*$ *outputs a prediction of* $\hat{r}_i \in \Sigma_O^*$ *that minimally deviates fro the true response* r_i.

Again, the annunciation of problem 2.2 is motivated by the huge underlying (unmeasured and unknown) complexities present in gene regulation mechanism. Problem 2.2 implies that in building a predictor of gene regulation based on DNA sequence, one should be wary of over-fitting and focus on generalization error. This is quite evident in the current situation where, unlike in other more closed system setup

[1] As a side note, the word ``streams'' is purposely employed to underline the expected complexity and volume of the data

(e.g. spam filtering, handwriting recognition, network routing), the more data produced (e.g. more TF binding sites identified) the further we seem to be getting from being able to conclusively predict gene expression. And that, we are brought into the realization of the need of additional cell-state data (e.g. epigenetics data (Bird ,2007; Reik, 2007)). This formulation of the problem also implies that learning algorithms and models that incorporate, explicitly or implicitly, the underlying relationships could be expected to fare better in the long run. Examples of such tools include Hidden Markov Model and Artificial Neural Network. Note that the declaration of problem 2.2 is intended more to help structure the thought process in viewing the overarching problem addressed by this project as a philosophical framework and less for being directly solved as an explicit mathematical problem statement.

Evidently, this framework also encompasses a range of different problems. Surely, the transcriptional activity prediction based on sequence data fits into this framework. Prediction of stock prices based on newspaper articles also falls under this scheme. Events, C, that influence the behaviour of market players, O, (and thus the stock prices R) are partially captured in noisy newspaper articles S. Another example is automated monitoring software that screens incoming and outgoing traffic from the internet into a large intranet and designed to intercept and thwart possible hacking attempts. Forecasting of the election results from newspaper articles could also be similarly modeled. All of these examples share a common theme that the response variable r_i is not a direct product, or one-to-one mapping, of the input s_i.

Two different strategies are possible in approaching problem 2.2:

1. Trying to directly learn the relationship between S and R. This could be done through classification or regression of vector-valued response variables. Although conceptually simple, in practice such algorithms can be complex and might be intractable.
2. The alternative approach involves abstracting out or simplifying/reducing the complexity of either the input or the response or both. The idea is intuitive, by reducing the response variables or the input vector, applications of existing algorithms become feasible. The challenge lies in devising an algorithm that captures the appropriate features from each stream. In other word, the aim is to develop feature extraction, reduction, and selection algorithms.

Although the goals of problems 2.1 and 2.2 are extremely desirable, the present genomic technologies and experimental limitations prevented us from executing effective research into them. Staying within the scope of the thesis, we concerned our research with gaining more insights into the true nature of the Outcome and the Control Signal, as well as the elements of Background Noise and other signals peppering them. The Output Stream needs to be dissected first, as it could considerably reduce the input space, by identifying the relevant ones, and provide additional domain knowledge. Following which, the Control Signal needs to be distilled from the Input Stream. In summary, we decoupled the main problem into the analysis of the Output Stream, i.e. expression of regulated genes, and the analysis of the Input Stream, i.e. genomic regulatory signal.

2.3 Expression of Regulated Genes

Within the framework outlined in Problem 2.1, the set of transcript readings R encompasses the set of genes within genome, as comprehensive as possible. The larger the set R, the more complex the model M could potentially be, as each gene reading r_i is associated with a regulatory sequence s_i. Assuming that many (or even most) of the measured transcripts are not related to the regulation by transcription factor T, the complexity of the Input Stream, and hence the resultant model M, can be reduced through proper selection of subsets of R.

2.3.1 Minimal Set of Gene Signature

In situations whereby stimulation of transcription factor T is not possible or that such data is not readily available, activity of transcription factors is sometimes investigated through comparison of different cell types where the transcription factors of interest are known to exhibit distinct behaviors. For example, the transcription factor PPARγ is known to be expressed in adipocytes but not in pre-adipocytes (Fu *et al.*, 2005). Genes regulated by PPARγ could therefore be identified by comparing expression profiles of adipocytes and pre-adipocytes. In such setup, genes that can be used as markers for the different cell type are potentially regulated by the transcription factor of interest. Stated this way, the problem is now rendered into the familiar problem of feature selection for classification. Our interest, however, was more specific. We wanted to not only attain a robust set for microarray classification, but to do so using as few genes as possible.

Problem 2.3 (Minimal Gene Set for Class Discovery) *Let* $Y = \{y_1,...,y_B\}$ *be the labels of* B *samples and* $X = \{H_1,...,H_B\}$ *be their expression profiles, where* $H_i = [x_{1,i},\ldots,x_{N,i}]$ *represent a vector of* N *genes' expressions. Let* C_A *be a classification algorithm that utilizes expression values of gene subset* $A \subseteq \{1,\ldots,N\}$ *to predict the sample labels* Y*. Determine the subset* A*, minimizing its size while maintaining a good generalized performance of* C_A*.*

Why did we aim to compile as few and as non-redundant genes as possible? Although the differentially expressed genes in this setup are likely to be truly regulated by the transcription factor *T*, the regulation may be indirect. It is more likely that the transcription factor *T* regulates a core set of primary targets, which in turn influence the regulatory network. The non-redundant criterion functions as a filter for direct target, while minimizing the set of selected genes reduces the overall noise. Moreover, the formulation of Problem 2.3 in fact appeals to a number of other applications, for example in gene marker discovery where the goal is to identify a set of genes whose protein level, typically measured by ELISA (Parker, 1990) or such, can be used as a predictive variable for certain cell state/disease. There, it is essential to obtain a small (due to resource constraint) and redundant (for robustness purposes) set of features.

2.3.2 Dominant Set of Expression Pattern

When the activity of transcription factor *T* can be subjected by external stimulation or perturbation, more ideal experiments for finding genes directly regulated by *T* could be performed. Typically the experimental setup consists of perturbing the biological system with external stimulant and monitoring the expression levels across several

timepoints. Timecourse expression data of non-perturbed system is also generated as the corresponding control data.

We shall now construct a general model for the problem by treating it as a system. Let *Z* be a system and $H = [x_1, \ldots, x_N]$ be a vector of *N* sensor readouts (or features $x_i \in \Re$) taken on the system, describing the state of the system. Let's also assume that the system can be subjected to an arbitrary factor *T* and that $H^{T,j}$ captures the state of the system at time *j*, under the influence of factor *T*. Unless stated otherwise, let $H^{0,j}$ denotes the state of the system at time *j* given no external factors. Note that for a given system *Z* and an external factor *T*, the features *H* can either be directly affected (primary response), indirectly affected, or unaffected by *T*. Our goal is to identify features that are directly influenced by *T*.

We can now define $N \times B$ matrix *X* as the net effect of factor *T* over *B* consecutive time points as:

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,B} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,B} \end{bmatrix}$$

We additionally define:

$$G_i = [x_{i,1}, \ldots, x_{i,B}] \text{ , and } H_j = [x_{1,j}, \ldots, x_{N,j}]^T$$

Note that the above formulation is in line with the response variables of the framework outlined in Problem 2.2. G_i is in fact $r_i \in \Sigma_O^B$, where $\Sigma_O = \Re$. In the context of gene expression data, *H* represents a single microarray reading that simultaneously probes *N* transcripts, while G_i is the expression level of gene (or transcript) *i* across *B* microarrays.

We shall now try to model the direct and indirect responses, for each timepoint. Let $d_i \in \{0,1\}$ be a binary variable denoting the primary response indicator to *T*, i.e. feature *i* is a primary response of *T* if and only if $d_i = 1$. We can define $E = [e_1, \ldots, e_N]$ as the 'basal' response of *T* such that $\forall i : d_i = 1 \Rightarrow (x_i^T - x_i^0) = e_i$. Then, for all indirect response feature *i*, the observed effect is proportional to the wighted sum of the effect to direct responses, i.e. $\forall i : d_i = 0 \Rightarrow (x_i^T - x_i^0) = \sum_{j;d_j=1} f_{i,j} \times e_j$ with $f_{i,j} \in \Re$. Altogether:

$$\forall i \in [1..N] : (x_i^T - x_i^0) = \begin{cases} d_i = 1 \Rightarrow b_i \\ d_i = 0 \Rightarrow \sum_{j;d_j=1} f_{i,j} \times e_j \end{cases}$$

or more generally:

$H = F(DE^T)$, where $F = \begin{bmatrix} f_{1,1} & \cdots & f_{1,N} \\ \vdots & \ddots & \vdots \\ f_{N,1} & \cdots & f_{N,N} \end{bmatrix}$, $D = diag(d_1, \ldots, d_N)$, and if $d_i = 1$ then $f_{i,i} = 1$ and $\forall j : f_{i,j} = 0$.

It is clear from the above that our goal is to solve matrix *D*, since primary responsive feature *i* has $d_i = 1$. Note also that the formulation captures the states and configuration (matrices *F*, *D*, *B*) for a particular given observation, and they may change with time. Thus, for each time point *j*, $H_j = F_j(D_j E_j^T)$. Nevertheless, to simplify, we assume that D_j is constant, i.e. $D_j = D$.

Problem 2.4 (Direct response features) *Given a time series data* X *consisting the observed changes of* N *features due to presence of external factor* T *across* B *consecutive timepoints as described above, find the features that were directly influenced by* T*, i.e. find* i *such that* $d_i = 1$.

Note also that the primary response features, i.e. features with $d_i = 1$, are in fact dominating the response landscape, since the indirect responses were propagated from primary responses, as modeled through matrix *F*. If matrix F is sufficiently sparse, then the overall patterns of response *X* would be dominated by the patterns exhibited by primary responses. As such, Problem 2.4 can be viewed as finding the dominant pattern.

2.4 Genomic Regulatory Signal

For the purpose of our study, we define *Genomic Regulatory Signals* as the information contained in DNA sequences that are relevant to the gene regulatory activity of transcription factors. Discussions on genomic regulatory signal typically bring into mind a host of computational and algorithmic challenges, such as motif discovery, sequence alignment, evolutionary analyses, and phylogenetic tree construction. During the course of our research, however, the landscape of data mining of regulatory signals has been transformed from medium throughput (for example analysis of promoter sequences or other set of sequences, arranged based expression profiles or other biologically meaningful categorization) into high-throughput genome-wide analyses.

The trend of high-throughput genome-wide analysis was initiated circa late 2000, employing a technique known as Chromatin-Immunoprecipitation on chip, or ChIP-on-chip (Ren *et al.*, 2000), where ChIP fragments are quantified by hybridizing them into a DNA microarray. A major technological advancement was the introduction of sequencing-based Chromatin-Immunoprecipitation (ChIP), spurred by the rapid development of the so-called next generation sequencing machines. One clear advantage of sequencing-based approach is that it is less biased compared to hybridization-based, which introduce a heavy bias during the probe selection stage. Various variants have since been introduced, including ChIP-SACO (Impey *et al.*, 2004), ChIP-PET (Wei *et al.*, 2006), ChIP-STAGE (Bhinge *et al.*, 2007), and the most recent ChIP-Seq (Johnson *et al.*, 2007).

In the context of high-throughput sequencing of ChIP fragments (or ***htsChIP***), due to the vast number of unspecific fragments sequenced along with the ChIP-enriched ones, the challenge is to identify locations in the genome where the observed fragment enrichment can be confidently ascribed to TF-DNA interaction. This project focused on data generated through the ChIP-PET protocol. In particular, five questions were addressed:

1. How can we quickly assess whether a given ChIP-PET library has been adequately sequenced?
2. What is the best model of ChIP fragment length distribution?
3. How can we assess a given ChIP-PET library in terms of its quality and total number of bound regions?
4. Can we distinguish (at finer resolution) regions that are bound by TF from those that were fragment-enriched by chance?
5. Without the presence of a control library, how can we reduce a systematic genome bias originating from fluctuations of genomic copy number (which is common among model systems based cell-lines)?

The exact problem formulations will be discussed in chapter 4.

Chapter 3

Inferring Patterns of Gene Expression

3.1 Overview

In this chapter, we detail our approaches for solving the problem of inferring relevant genes from microarray data, focusing on two specific challenges: the identification of minimal set of signature genes (Section 3.2) and the identification of treatment responsive genes based on time-course microarray studies (Section 3.3).

3.2 Modifying Boosting for Class Prediction in Microarray Data

Identification of minimal set of signature genes is pertinent in the context of microarray-based tissue type prediction. While creating a good-performing microarray-based tissue type predictor is somewhat straightforward (e.g. approaches based on k-NN, SVM, and other generic machine learning models), the challenge of discovering a minimal yet robust set of genes is still relevant. Biologically, such minimal gene set might represent a key cellular regulator important for a specific tissue type (e.g. cancer) and could potentially be regulated by a similar mechanism (e.g. similar set of transcription factors). When the different tissue type is in fact derived from treatment of ligands that interact or activate certain transcription factor or that the tissue types were substantially related to activity of a specific transcription

factor, such list of signature genes reflect the representative set (or the core set) of genes' response to the treatment, which could mean that the genes are more likely to be direct targets of the activated transcription factor (see Section 2.3.1).

3.2.1 Problem Description

Following the definition stated in Problem 2.3, we model the problem as follows: let $X = \{x_{i,j} \in \Re \mid 1 \le i \le N, 1 \le j \le B\}$ be the set of expression array arranged as an $N \times B$ matrix, where $x_{i,j}$ is the expression level of *i*-th gene in the *j*-th sample. $Y = [y_1, \ldots, y_B]$ is the sample labels, where y_j denotes the label of the *j*-th sample. For ease of notation, let $G_i = [x_{i,1}, \ldots, x_{i,B}]$ represents the expressions of *i*-th gene across all samples and $H_j = [x_{1,j}, \ldots, x_{N,j}]$ denotes the expression profile of *j*-th sample. Our goal is to develop a learning algorithm $M(X, Y, k)$, that takes as input the expression data *X*, the associated labels *Y*, and the maximum number of genes *k* that the classifier is allowed to use, and outputs a classifier $C_A(H')$. Given a vector H' of gene expression data of a biological sample, the classifier $C_A(H')$ predicts the label of H' based on the gene subset $A \subseteq \{1, \ldots, N\}$. This gene subset *A* should be examinable from the output classifier $C_A(H')$.

3.2.2 Support Vector Machine Algorithms

Prior to our investigation, there have been a couple of papers describing the application of Support Vector Machines (SVM) for class prediction in the context of microarray data. As part of our experiment, we employed several variants that were more in line with the specific goal of identifying a minimal gene subset for classification.

Wilcoxon/SVM

Mann-Whitney Wilcoxon Rank-Sum test (Mann and Whitney, 1947; Wilcoxon, 1949) has proved to be useful in multiple contexts of microarray data analysis, especially for discovering differentially expressed genes. In conjunction with SVM, the test can be used to select genes for building a classifier. Specifically, this algorithm:

- Chooses the *k* genes identified as differentially expressed between the two types of tissues according to the Wilcoxon-Mann-Whitney test with the highest confidence (using the training data provided), and
- Applies SVM with a linear kernel and soft margin with the cost parameter C.

In our experiments, the parameter C is chosen to minimize the five-fold cross-validation error on the training set of the entire inductive process including feature selection. The optimization was done using a simple successive refinement algorithm.

SVM-RFE

Another version is our implementation of SVM with Recursive Feature Elimination (Guyon *et al.*, 2002). It has a parameter *k*, the number of genes used. The data is first rescaled and translated so that each attribute has mean 0 and variance 1 over the training data (the parameters are chosen using the training data, and any test data is rescaled and translated in the same way). Training proceeds in a number of iterations.

In each iteration:

- A separating hyperplane is trained using SVM with a linear kernel and the default value of C from SVMlight (Joachims, 1998) (some cross-validation experiments suggested that this performed better than the value C = 100 used in Guyon et al. (2002),
- the features (in this case genes) are ranked by the absolute magnitude of their corresponding weights in this hyperplane, and
- the bottom ranking half are deleted.

When the last step would reduce the number of genes to less than *k*, then instead genes are removed from the bottom of the list until *k* genes remain. This is the less computation-intensive of the algorithms proposed by Guyon *et al.* (2002). It appeared impractical to evaluate the more computation-intensive algorithm in a similar way. It also appeared impractical to choose C using cross-validation on the training set.

3.2.3 Practical Variants of AdaBoost for Expression Data

In this section, we describe several boosting algorithms customized for expression data. Recall that, for comparison, pseudo-code for AdaBoost is given in Fig. 2.

Given $\left\{1 \le i \le m \mid (x_i, y_i) \in \Re^n \times \{-1,1\}\right\}$:

- For each index *i* of an example, initialize $D_i(i) = 1/m$
- For each round *t* from 1 to *T*:
  - Choose a decision stump h_t to minimize the weighted error on the training data with respect to D_t, i.e. to minimize $\sum_{i:h_t(x_i)\neq y_i} D_i(i)$
  - Calculate the error $\varepsilon_t = \sum_{i:h_t(x_i)\neq y_i} D_i(i)$,
  - Set the update factor $\beta_t = \varepsilon_t / (1-\varepsilon_t)$
  - Update the distribution:
    - For each *i*, set $D'_{t+1}(i) = \begin{cases} \beta_t D_t(i) & \text{if } h_t(x_i) = y_i \\ D_t(i) & \text{otherwise} \end{cases}$
    - Normalize D'_{t+1} to get D_{t+1}, i.e. for each *i*, set $D_{t+1}(i) = \frac{D'_{t+1}(i)}{\sum_j D'_{t+1}(j)}$,
  - Set the weight $\alpha_t = \ln \frac{1}{\beta_t}$ with which decision stump *t* votes (if $\beta_t = 0$, then $\alpha_t = 0$ and the algorithm can halt)
- Return the final classification rule: $h(x) = \begin{cases} 1 & \text{if } \sum_{t:h_t(x)=1} \alpha_t > \sum_{t:h_t(x)=-1} \alpha_t \\ -1 & \text{otherwise} \end{cases}$

Figure 2. Pseudo-code for AdaBoost applied with decision stumps (adapted from Freund & Schapire (1996)).

AdaBoost-VC

We view AdaBoost-VC as the most theoretically principled variant of AdaBoost that we propose. Our design of AdaBoost-VC is guided by the following commonly adopted point of view (Vapnik & Chervonenkis, 1971; Vapnik, 1982, 1989, 1995, 1998; Valiant, 1984; Haussler, 1992). We assume that a probability distribution over instance/class pairs is used to generate the training data. We further assume that after the algorithm comes up with the classification rule, the instances on which it must be

applied, together with their correct classifications, are also generated according to the same distribution. In the below discussion, it will be useful to consider a collection of random variables, one for each decision stump s, that indicate whether, for a random instance/class pair (x, y), it is the case that $s(x) \neq y$. We will refer to each such random variable as an *error random variable*, or an *error* for short. Due to the reweighting of the examples, the classification rules returned by different invocations of the base learner tend to have negatively associated errors, say in the sense of (Dubhashi & Ranjan, 1998). Negative association formalizes the idea that a collection of random variables tend to behave differently. Boosting promotes this property in the error random variables by weighting the examples so that examples on which previous decision stumps were incorrect are more important, and thus tend not to be errors for future decision stumps.

When the errors of the decision stumps output by boosting are negatively associated, all else being equal, adding more voters improves the accuracy of the aggregate classifier by reducing the variance of the fraction of voters that correctly classify a random instance, making the correct fraction less likely to dip below 1/2 (this is for a similar reason that adding more independent coin flips reduces the variance of the fraction coming up heads - negative association accentuates this effect (Dubhashi & Ranjan, 1998)). However, when the errors of the individual voting classification rules are unequal, there is a balance to be struck, informally, between the diversity of opinion and its quality. In the case in which the errors are exactly independent, one can work out how optimally to strike this balance (Duda & Hart, 1973): it involves assigning weights to the voters as a function of their accuracy, and taking a weighted vote. To a first approximation, the weighting of the voters

computed by AdaBoost might be viewed as akin to this, but taking some account of what dependence there is among the errors.

Intuitively, one would like the errors of the voting classification rules to be negatively associated with respect to the underlying distribution generating the test data. However, some theory (Schapire & Singer, 1999; Kivinen &Warmuth, 1999) suggests that the tendency of the voters in the output of AdaBoost to have negatively associated errors is a byproduct of the more direct effect that the voting classification rules tend to have negatively associated errors with respect to the distribution that assigns equal weight to each of the training examples.

The above viewpoint, that AdaBoost approximates finding a set of classification rules with negatively associated errors and then weighting them optimally, also suggests that the weights assigned to the voters should be a function of their accuracy with respect to the underlying distribution. A special case of this is the observation mentioned in the introduction that a voter that is perfect on the training data should not vote with infinitely large weight, as is done in the standard AdaBoost.

In AdaBoost, the weight assigned to a voting classifier, and the reweighting of the examples after it is chosen, is based on the (weighted) error of the voter on the training data. We propose to instead use an estimate of the error with respect to a probability distribution over the entire domain. The probability distribution can be obtained by (i) starting with the original underlying distribution, (ii) reweighting every possible instance/class pair according to the number of previously chosen voters that got it wrong in the analogous way as is done by AdaBoost on the training data,

and (iii) normalizing the result so that it is a probability distribution (i.e., the distribution used in "boosting-by-filtering" (Freund, 1995)).

How to obtain such an estimate? For an individual voter, the weighted error on the training data can be viewed as an estimate of the error according to the reweighted underlying distribution. However, the estimate is biased by the fact that the voterwas chosen to minimize this weighted error. Vapnik (1982) proposed to counteract biases like this with a penalty term obtained though a theoretical analysis (Vapnik & Chervonenkis, 1971; Vapnik, 1982). Informally, in this case, this analysis provides bounds on the difference between the observed error rate of the best decision stump and the true error rate with respect to the underlying distribution that hold with high probability for any distribution on the instance/class pairs; Vapnik proposed to adjust the estimate by adding this bound. Kearns *et al.* (1997) proposed a variant based on a guess of what the result of the tightest possible analysis would be. In our context, if *m* is the number of examples, *n* is the number of genes, and ε^{emp} is the (weighted) training error, the estimate obtained is

$$\varepsilon^{emp} + \frac{\ln n}{m}\left(1 + \sqrt{1 + \frac{m\varepsilon^{emp}}{\ln n}}\right) \qquad \textbf{(3.2.1)}$$

(The fact that the estimate is based on a weighted sample weakens the link between their recommendation and this application; if the weight is concentrated in a few examples, the effective number of examples is less than m. Coping with this in a principled way is a potential topic for future research.) The following expression matches theory a little more closely (Vapnik, 1982; Haussler, Littlestone, &Warmuth, 1994; Talagrand, 1994; Li, Long,

& Srinivasan, 2001)

$$\varepsilon^{emp} + \frac{\ln n}{m}\left(\ln m + \sqrt{1 + \frac{m\varepsilon^{emp}}{\ln n}}\right) \qquad \textbf{(3.2.2)}$$

(In short, it has been shown that the $\ln m$ term is necessary in the theoretical bounds on how accurate the best decision stump can be.) Another issue must be confronted: what to do if a classifier returned by the base learner correctly classifies all of the data. Even if Eq 3.2.1 or Eq. 3.2.2 is used, since no errors are made, none of the weights of any of the examples will change, and the base learner will return the same classification rule again the next time it is called, and so on for the remaining number of rounds. We get around this by requiring that a given gene can be used in only one decision stump.

When we began experimentation with an algorithm that used Eq. 3.2.2 together with only allowing each gene to appear once, it became immediately obvious that the penalty term in Eq. 3.2.2 was too severe: the estimates were immediately far above 1/2. However, Eq. 3.2.2 is based on an analysis concerning a worst-case probability distribution. In practice, the "effective" number of genes will be much less. In microarray data, this could be because many genes (i) have expression profiles similar to other genes, or (ii) are completely unassociated with the class label, and therefore present substantially less of a threat to be in decision stumps that fit the data well by chance. One could imagine estimating the effective number of genes, for example by clustering genes based on their expression profiles and counting the number of clusters with members that correlate significantly with the class label. Instead of incurring the resulting expense in system complexity and computation time, we use the following expression

$$\varepsilon^{emp} + \frac{d}{m}\left(\ln m + \sqrt{1 + \frac{m\varepsilon^{emp}}{\ln n}}\right) \qquad \textbf{(3.2.3)}$$

with d as an adjustable parameter. In our experiments, we chose d from among $\{0,\ldots,3\}$ to minimize five-fold cross-validation error on the training set. In case of a tie, the geometric mean of the values of d attaining the minimum was used. Pseudo-code for AdaBoost-VC is in Fig. 3.

Given $\left\{1 \le i \le m \mid (x_i, y_i) \in \Re^n \times \{-1,1\}\right\}$:

- For each index *i* of an example, initialize $D_i(i) = 1/m$, and the set A of available attributes to $\{1,\ldots,n\}$.
- For each round *t* from 1 to *T*:
  - Choose a decision stump h_t to minimize the weighted error on the training data with respect to D_t, i.e. to minimize $\sum_{i:h_t(x_i)\neq y_i} D_i(i)$
  - Calculate the weighted empirical error $\varepsilon_t^{emp} = \sum_{i:h_t(x_i)\neq y_i} D_i(i)$,
  - Set $\varepsilon_t = \varepsilon_t^{emp} + \frac{d}{m}\left(\ln m + \sqrt{1 + \frac{m\varepsilon_t^{emp}}{d}}\right)$
  - Set the update factor $\beta_t = \varepsilon_t/(1-\varepsilon_t)$
  - Update the distribution:
    - For each *i*, set $D'_{t+1}(i) = \begin{cases} \beta_t D_t(i) & \text{if } h_t(x_i) = y_i \\ D_t(i) & \text{otherwise} \end{cases}$
    - Normalize D'_{t+1} to get D_{t+1}, i.e. for each *i*, set $D_{t+1}(i) = \frac{D'_{t+1}(i)}{\sum_j D'_{t+1}(j)}$,
  - Set the weight $\alpha_t = \ln\frac{1}{\beta_t}$ with which decision stump *t* votes (if $\beta_t = 0$, then $\alpha_t = 0$ and the algorithm can halt)
- Return the final classification rule: $h(x) = \begin{cases} 1 & \text{if } \sum_{t:h_t(x)=1}\alpha_t > \sum_{t:h_t(x)=-1}\alpha_t \\ -1 & \text{otherwise} \end{cases}$

Figure 3. Pseudo-code for AdaBoost-VC.

AdaBoost-NR ("no repeat")

This algorithm is like AdaBoost, with two changes. First, as in AdaBoost-VC, each gene is constrained to be in at most one decision stump. Second, if a decision stump correctly classifies all of the training data, its weight is set as if its weighted error on the training data was $0.1/m$, where *m* is the number of samples. This is instead of the infinite weight given to such a stump by AdaBoost. The choice of $0.1/m$ is intended to have the effect, in most cases, of ensuring that the decision stump has the largest weight of those chosen. We evaluated this algorithm to gain insight into the share of the improvement seen by AdaBoost-VC that could be attributed to using each gene at most once. However, it appears to be a useful algorithm in its own right.

AdaBoost-PL ("piecewise linear")

This algorithm is an instantiation of AdaBoost with "confidence-rated" predictions (Schapire and Singer, 1999). The classes are designated by 1 and −1, and the base classifiers are functions from expression profiles to the continuous interval [−1, 1]. When a base classifier h is applied to an expression profile *x*, the sign of *h(x)* is interpreted as its class prediction, and the magnitude of *h(x)* is interpreted as its confidence in that prediction.

The base classifiers used in our implementation of AdaBoost-PL are piecewise-linear generalizations of decision stumps. Note that a decision stump that predicts 1 exactly when $x_i \geq \theta$ can be written as outputting $sign(x_i - \theta_i)$. This is replaced with $\pi\left(\frac{x_i - \theta_i}{c\sigma_i}\right)$, where:

- π is defined by

$$\pi(u) = \begin{cases} 1 & \text{if } u \geq 1 \\ -1 & \text{if } u \leq -1 \\ u & \text{otherwise} \end{cases}$$

- σ_i is the standard deviation of feature x_i on the training data, and
- c is an adjustable parameter, chosen to minimize five-fold cross-validation error on the training set (the values in $\{0.05,0.1,0.2,0.5,1.0,2.0\}$ were tried, and the geometric mean of the values resulting in the minimum error was used)

Similarly, $sign(-x_i - \theta_i)$ is replaced by $\pi\left(\frac{-x_i - \theta_i}{c\sigma_i}\right)$. The base classifier h_t of round t is chosen in minimize $\sum_i | h_t(x_i) - y_i | D_t(i)$, where the weights $D_t(i)$ of the examples are updated as in Schapire & Singer (1999).

Arc-x4-RW ("re-weight") and Arc-x4-RW-NR

Since the main problem with AdaBoost on expression data appears to be concentrating too much weight on the predictions of decision stumps that do well on the training data, an anonymous referee asked whether an algorithm like Arc-x4 (Breiman, 1998) might be well-suited to such data. Arc-x4-RW is like boosting, except: (i) all base classifiers in the final class prediction rule vote with equal weight, and (ii) the weight of example i in round t is proportional to $1 + c_{i,t}^4$, where $c_{i,t}$ is the number of base classifiers prior to round t that classified example i incorrectly. The difference between Arc-x4-RWand Arc-x4 is that, instead of minimizing the weighted training error as in Arc-x4-RW, Arc-x4 resamples from the training set m times with probabilities proportional to the weights, and minimizes the error on the result. Arc-

x4-RW-NR, is like Arc-x4-RW, except with the added constraint that each gene appears in at most one decision stump.

3.2.4 Evaluation

Dataset

Seven datasets were used in our experiments. Six were part of the published version of this work:

- In the well-known ALL-AML dataset (Golub *et al.*, 1999), the task is to determine whether a given gene expression profile belongs to an Acute Lymphoblastic Leukemia (ALL) tissue or an Acute Myeloid Leukemia (AML) tissue. It contains 72 samples (47 ALL, 25 AML), each with expression profiles concerning 7129 genes.
- Liver cancer (HCC) dataset (Neo *et al.*, 2004) with an additional inclusion of a matched tumor-normal pair, totaling 76 samples (38 tumor and 38 normal) with expression profiles concerning 9050 genes measured with a cDNA microarray. Ratios against a universal human reference containing a mixture of tissues types were measured, a log transform was applied, and the data was normalized so that the average log ratio for each array was 0.
- Another dataset concerns colon cancer (Alon *et al.*, 1999): again, it contains expression profiles for tumor and normal samples.
- The next two datasets analyze expression profiles of breast cancer samples (West *et al.*, 2001) with classes defined by (i) whether the gene responsible for estrogen response is being expressed (ER), and (ii) whether the tumor has spread to the lymph nodes (LN).

- Another dataset (Pomeroy *et al.*, 2002) involves predicting whether a patient with a brain tumor survives after treatment.
- The final dataset (Kuriakose *et al.*, 2004) requires us to predict whether a sample is generated from human head and neck (HNC) normal mucosa or cancer tissue.

Aside from the HCC dataset, on which we applied standard preprocessing steps, we used all datasets exactly as we found them.

We evaluated all of the algorithms with two constraints on the number of genes (k) they used, 10 or 100. For the boosting-based algorithms, this was achieved by limiting the number of rounds of boosting to k. The use of k in the algorithms used by SVM was described in Section 3.2.2. For each algorithm and each dataset, we performed the following steps 100 times and averaged the results: (a) randomly split into a training set with 2/3 of the examples and a test set with 1/3 of the examples, (b) apply the algorithm on the training set, (c) calculate the error rate on the test set. This is similar to what was done by Dudoit, Fridlyand, and Speed (2002); they argued persuasively that this is preferable to more standard techniques like k-fold cross-validation and leave-one-out cross-validation when the goal is to compare the performance of different algorithms, since it reduces the variance of the estimates of the generalization error rates. We subjected all of the algorithms to the same training/test splits, eliminating one source of variance in the estimates of the differences between their average training set errors.

It is worth emphasizing that feature selection was redone using only the training data after each training-test split. Doing cross-validation after feature selection can optimistically bias the resulting error estimates dramatically (Ambroise & McLachlan, 2002; Miller *et al.*, 2002). Also, whenever an algorithm had parameters to set, these were chosen separately for each training-test split, by doing cross-validation on the training set only.

Our results are summarized in Table 1. The first observation is that, on the ALL-AML and HCC datasets, where there is a strong association between expression profiles and class designations, AdaBoost-VC, AdaBoost-NR, and Arc-x4-RW-NR all substantially improved on the performance of raw AdaBoost. These algorithms also compare well with the two algorithms using SVM on the ALL-AML and HCC datasets, and to a lesser extent on the ER dataset, especially when only 10 genes are used.

Generally, it appears that as the association between expression profiles and class designations grows weaker, the relative performance of the algorithms using SVM improves. Arc-x4-RW-NR appears to substantially improve on Arc-x4-RW overall. The additional inductive bias in favor of weighting genes equally appears to be being rewarded. Note that while AdaBoost-VC reduces the weight associated with stumps that perform well on the training data, which has the effect of evening out the weights among the stumps, it also reduces the weights of stumps that perform moderately well on the training data, in some cases reducing them to nearly zero. Thus, overall, the effect of AdaBoost-VC is not necessarily to even out the weights among the voters. Arc-x4-RW-NR appears to perform the best overall, though its

performance on the ALL-AML and HCC datasets is nearly indistinguishable from the performance of AdaBoost-VC and AdaBoost-NR. The similarity in performance was also recapitulated in HNC (in particular those based on 100 genes). Taken together, these results supported our intuition that a key modification in the application of boosting for expression data involves reduction of reliance to the classification performance of the individual decision stump / weak classifier. It is conceivable that this rule also apply to other datasets with small number of samples and significantly larger number of features.

Algorithm	Gene limit	ALL-AML	HCC	ER	Colon	LN	Brain	HNC
Adaboost	10	6.2	7.8	19.9	25.3	40.4	42.3	16.8
Adaboost-VC	10	3.9	5.6	18.1	24.4	43.8	41.1	11
Adaboost-NR	10	3.5	6	19.5	25.1	42.7	41.2	11.5
Adaboost-PL	10	7	7.2	20.6	23.4	36.5	41.9	8
Arc-x4-RW	10	6.5	8.2	19.8	25	39.1	41.4	11.1
Arc-x4-RW-NR	10	3.3	5.5	17.8	24.7	42.1	40.7	9.7
SVM-RFE	10	13.4	8.6	20.9	19.2	48.4	39.2	15.6
Wilcoxon/SVM	10	6.4	6.7	23.2	24.3	35.4	39.3	8.2
Adaboost	100	5.2	6.9	16.1	23.4	35.4	38.2	16.6
Adaboost-VC	100	2.8	4.8	13.8	22.6	42.8	38.2	10.4
Adaboost-NR	100	2.7	4.9	13.2	21.9	40.6	36.5	9.9
Adaboost-PL	100	5	5.4	17.2	23.2	36.2	38.6	9.4
Arc-x4-RW	100	5.4	7.4	16.6	23.7	36.9	38	12
Arc-x4-RW-NR	100	2.6	4.8	12.8	21.6	41.1	36.1	10
SVM-RFE	100	6.5	6.7	12.6	20.7	48.1	35.7	11.8
Wilcoxon/SVM	100	3.3	4.1	17.5	23.6	40.4	37.8	7

Table 1. Performance of algorithms for microarray classification. Comparison of cross-validation estimates of generalization error percentage of eight algorithms on seven microarray datasets.

3.3 Friendly Neighbour Method for Identification of Treatment Responsive Cassettes

As mentioned earlier in Section 2.3.2, when the activity of a transcription factor *T* could be influenced by external stimulation or perturbation, a more ideal experiment to identify direct target genes of the transcription factor would be a timecourse experiment, measuring the expression of genes in samples of subjected to the external stimulation and contrasting it to those from untreated samples. A handful of techniques tailored to exploit temporal information embedded within time-course data have been proposed prior to our study. (Park *et al.*, 2003) developed a statistical test that extends ANOVA and coupled it with permutation test to arrive at an empirical *p*-value for each gene. The CAGED algorithm (Ramoni *et al.*, 2002) models each gene's time-course readings using autoregressive models and progressively merge models, two at a time, into single model as long as the resultant model has a higher marginal probability. Kasturi *et al.* (2003) viewed each gene's time-course profile as a probability distribution over time and employed Kullback-Leibler (KL) divergence to quantify dissimilarity of the shape of the expression profiles between a pair of genes. The utility of constructing and fitting biologically-motivated mathematical models for the discovery of important genes in time-course data is illustrated in (Xu *et al.*, 2002), where they built a statistical model for a gene's expression level at each time-point, estimated its parameters using the empirical data, and performed significance tests on the fitted parameters. Note that many of the time-course specific methods mentioned here include a preprocessing step of gene filtering. Again, most employed threshold-based filtering.

Threshold-based filtering assumes that noisy gene profiles in the subjects of interest exhibit low expression values or low expression deviations from the control. There also exists a different kind of noise in microarray data. If we are to define noisy genes as irrelevant genes to the study, then randomly oscillating genes, regardless of their absolute or relative expression levels, are in fact noise. Such genes might not be weeded out by thresholding. Wilcoxon-Mann-Whitney test does a good job in removing such genes for supervised analysis of single time-point multiple-array studies.

If randomly expressed genes are basically noise, what then is non-random expression pattern that constitutes non-noisy genes? With regard to the data, we can define non-random expression patterns as those shared by large groups of genes. In time-course data, this means that a gene is significant (or rather, non-noisy) if its expression profile across time is shared with a number of other genes. Its significance is proportional to the number of genes that share its profile.

3.3.1 Problem Description

Reformulating the generalized model outlined in the Section 2.3.2, the input expression ratio data of N genes/transcripts measured over B time-points is modeled in $N \times B$ matrix:

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,B} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,B} \end{bmatrix}$$

And correspondingly:

$$G_i = [x_{i,1}, \ldots, x_{i,B}] \text{ , and } H_j = [x_{1,j}, \ldots, x_{N,j}]^T$$

Where $x_{i,j} \in \Re$ is the expression ratio of the i^{th} gene at the j^{th} timepoint. G_i can be viewed as the expression ratio profile of gene *i* across the measured time-points and H_j is the expression ratios within timepoint *j*. Similar to the goal outlined in Problem 2.4, the goal here is also to determine the genes that are directly regulated by *T*. Likewise, the direct target attribute is encoded in the matrix $D = diag(d_1, \ldots, d_N)$, where gene *i* is a direct target if and only if $d_i = 1$. Recall that *D* should actually be defined for each time point *j*. For simplicity, we maintain the assumption that *D* is constant across all timepoints, i.e. $\forall j : D_j = D$. Recall also that we model the net effect observed at each timepoint as a mixture of basal signals *E* dependent on mixing matrix *F* and the direct response indicator *D*, i.e. $H_j = F_j(D_j E_j^{\ T})$.

Challenges and Observations

Given that none of the matrices *F*, *D*, and *E* are not known, one might estimate them by trying to fit these parameters with sufficient replicates of *X*. Such a luxury, we believe, would be rare for the present moment. In most settings, chances are that there is inadequate amount of data for directly solving the matrices *F*, *D*, and *E*. Finding proxies for detecting *i* where $d_i = 1$ is more feasible.

In a natural system, it's not inconceivable to expect matrix *F* to be sparse, for instance we expect that each gene should only be affected by a handful of other genes. To certain extent, we also expect it to be stable (i.e. $F_j \sim F_{j+1}$). By stable, here we mean that non-zero components in F_j would most likely be non-zero and having the same sign in F_{j+1}, and vice-versa. For example, if a feature *j* is truly affected by (and

let's say positively propotional to) *i* at one time point, we expect *j* to be similarly influenced by (and positively proportional to) *i* at other time points. In the following discussion, we will also assume that only a single replicate of *X* is available.

3.3.2 Unsupervised Algorithms

Problem 2.4 calls for ranking algorithms that require no training examples. We list here potential unsupervised approaches that could be employed to detect the direct responders.

Statistical ranking

By making some reasonable hypotheses or expectations, one can easily compute a statistics and use it to rank the features based on their likelihood of being direct responders.

Such methods include:

- **Deviations of the means**. Recall that *X* gives the net effect, due to factor *T*, measured on the system. For the unresponsive features *u*, it's not unreasonable to expect their net effects to be around zero or, in other words, the mean of $G_u \approx 0$. Further, since we assume that D_j is constant and $F_j \sim F_{j+1}$, direct responders *i* can be expected to yield a mean that deviates substantially from zero. Statistical tests that assess whether the mean of a given set of values is zero, such as *t*-test and wilcoxon rank sum test (Mann and Whitney, 1947; Wilcoxon, 1949), are clearly applicable.
- **Dynamics of the net effect**. Still assuming that only the non-responsive features have near-zero values, we can exploit the observed dynamics of the net effect values. Among them would be to base the ranking of the features on

the maximum magnitude of the response (i.e. $\max_j (|x_{i,j}|)$), the variance, or the range of the values (i.e. $\max_j (|x_{i,j}|) - \min_j (|x_{i,j}|)$). Each of these carries the expectation that significantly deviating genes are the responsive ones.

Clustering based

Clustering algorithm, a powerful tool for data mining and explorations, might also be used to generate putative ranking of features. As is, clustering outputs are meant for investigating relationship between examples, with respect to the underlying similarity measure. The resultant clusters are not directly translatable to ranking or ordering of the clustered items, unless certain assumptions are made. For this problem, responding features can be reasonably assumed to form tight (i.e. having a good similarity) and sizeable clusters. Hence, given a hierarchical clustering of the features, a putative order of response can be generated by giving a higher ranking to features that fall in a tighter and larger cluster.

3.3.3 Supervised Algorithms

Although problem 2.4 is naturally unsupervised, the identity of some direct responders might have been uncovered from other means. This is useful for both (i) ranking of other features that yet to have their nature determined and (ii) evaluating the putative ranking generated by unsupervised approaches. Listed below are a couple of potentially useful supervised algorithms for identifying direct responders.

SVM based

The widely successful and generic classification algorithm Support Vector Machine (Vapnik, 1995) treats examples as vectors and classifies (or predicts the label of) a

new example based on the sign of its distance (see Eq. 3.3.1) to the separating hyperplane, which was learned from the training examples.

$$y(G) = sign\left(b + \sum_{i \in SuppVector} \lambda_i y_i K(G, G_i) \right) \qquad \textbf{(3.3.1)}$$

For the purpose of ranking, we can base the ranking on the raw distance to the hyperplane. Assuming that the direct responders are assigned positive labels, *G* can be ranked on descending $y'(G)$, where

$$y'(G) = b + \sum_{i \in SuppVector} \lambda_i y_i K(G, G_i) \qquad \textbf{(3.3.2)}$$

k-*NN based*

The application of the *k*-Nearest Neighbour (*k*NN) algorithm for ranking is also straightforward. Given a previously unseen example, instead of predicting its label based on the dominant labels of its *k*-nearest neighbours, we can order the unseen examples, *G*, based on the number of positive examples among the *k*-nearest known examples of each unseen example.

3.3.4 Friendly Neighbour Approach

Motivation

When only direct responses are assumed to be present in the system, the matrices *F* and *D* are both reduced to identity matrices, making $H_j = E_j$. The problem can be then easily solved by identifying non-zero e_i in E_i, while controlling for noise

and/or minimizing fitting error[1]. Presence of indirect responses, although confounding, can be exploited to help the identification of direct responses.

The constraint described in section 2.3.2 states that direct responders are only influenced by themselves (if $d_i = 1$ then $f_{i,j} = 1 \Leftrightarrow i = j$). Unless otherwise stated, for simplicity, we also assume that the direct targets influence indirect target in the same direction, i.e. $\forall i, j : f_{i,j} \geq 0$.

The expectation that the matrix F is relatively sparse (see Section 3.3.1) means that an indirect responder j has only a handful of direct responders, $A_j = \{i \mid f_{i,j} \neq 0 \wedge d_i = 1\}$, affecting it, while the stability hypothesis implies that G_j and G_i, where $i \in A_j$, should be tangibly similar. Clearly, most (if not all) direct responders would then possess a sizeable number of other features that are similar.

Main algorithm

To exploit the interaction between the direct and indirect responders, we introduce the notion of *friendly*. Two features i and j are called to be friendly, under a given similarity function $sim(X, Y)$, if $sim(G_i, G_j)$ is above a certain threshold θ. For each feature i, its Friendly Neighbor score can then be defined as the total number of features that are friendly to it, or:

$$FN_{score}(i) = \left|\{ j \mid sim(G_i, G_j) > \theta \}\right| \qquad \textbf{(3.3.3)}$$

[1] Recall that matrix X gives the changes in the measured values due to external factor T. Hence, ideally, all non-zeros are caused by T. If we then assume that only direct responders are present, all non-zeros are then direct responders.

To identify the direct responders, the features can be ranked based on decreasing order of their FN_{score}. In the settings of gene expression, the *FNscore(i)* measure the number of genes that are similar to gene G_i. The higher the score the more probable that the feature *i* responses directly to *T*.

Similarity measures

The calculation of FN_{score} relies on the underlying similarity function $sim(X, Y)$, where $X = [x_1, \ldots, x_n]$ and $Y = [y_1, \ldots, y_n]$. An appropriate similarity function should exploit and leverage on the underlying nature of the data being investigated. Several useful similarity measures (including those described in (Karuturi and Vega, 2004) are:

Sign Match (SM) The *sign match* similarity function, $SM(X, Y)$, counts the number of features corresponding elements of vectors *X* and *Y* whose signs agree. Let's first define a step function:

$$\sigma(a) = \begin{cases} 1 & \text{if } a > 0 \\ 0 & \text{if } a \leq 0 \end{cases}$$

Hence, $SM(X, Y) = \sum_i \sigma(x_i \times y_i)$. For uniformity, the similarity score is normalized to be a real value between 0 and 1. The refined sign match similarity function is thus:

$$SM(X, Y) = \frac{1}{n} \sum_i \sigma(x_i \times y_i)$$

Longest Consecutive Sign Match (LCSM) The above simple sign match similarity assumes that elements of the vectors are completely independent of each

other. If the elements of the vectors are ordered in some meaningful manner (e.g. in temporal order, just like the settings for problem 2.4) and suppose that consecutive consistent behaviour is desirable, we might opt for a stricter measure that prefer consistency or continuity across consecutive elements. The *longest consecutive sign match* intends to capture the most persitent sign agreement between vectors *X* and *Y*. It considers the longest stretch of sign agreements as the representative "consistent" similarity between two vectors. $LCSM\ (X,Y)$ can be calculated as:

$$LCSM\ (X,Y) = \max_i\ w_i \text{, where}$$

$$\forall i \in [1,n] : w_i = \sigma(x_i \times y_i)(1 + w_{i-1}) \text{ and } w_0 = 0$$

This similarity score is also normalized such that $0 \le LCSM\ (X,Y) \le 1$ by using the alternative formula $LCSM\ (X,Y) = \frac{1}{n}\max_i\ w_i$.

Weighted Consecutive Sign Match (WCSM) The sign agreement based similarity can be further generalized into what we call the *weighted consecutive sign match*. In this framework, consecutive matches are given bonuses. The bonus is proportional to a constant Δ, while mismatches reduce accumulated the bonus score. The similarity score can then be formulated as:

$$WCSM\ (X,Y) = \sum_i \sigma(x_i \times y_i)(1 + w_i) \text{, where}$$

$$\forall i \in [2,n] : w_i = \max\left(w_{i-1} + \Delta(2\sigma(x_{i-1} \times y_{i-1}) - 1), 0\right) \text{ and } w_1 = 0$$

The normalized score can also be calculated by dividing the raw score by the maximum possible score:

$$WCSM\ (X,Y) = \frac{1}{n + \frac{n-1}{2} n\Delta} \sum_i \sigma(x_i \times y_i)(1 + w_i)$$

Pearson Correlation (PC) *Pearson Correlation* is one of the widely used similarity function. It characterizes the linear relationship between the two vectors. The score ranges from +1 to -1, representing perfect positive linear correlation to perfect negative linear correlation. It is computed by:

$$PC(X,Y) = \frac{\sum_i x_i y_i - \frac{1}{n} \sum_i x_i \sum_i y_i}{\sqrt{\left(\sum_i x_i^2 - \frac{1}{n} \left(\sum_i x_i \right)^2 \right) \left(\sum_i y_i^2 - \frac{1}{n} \left(\sum_i y_i \right)^2 \right)}}$$

Normalization of the correlation, into $PC(X,Y) \in [0,1]$, is also straighforward. We might choose not to normalize it, so as not to lose the intuitive interpretation of the score.

3.3.5 Evaluation

Dataset

For evaluation purposes, we shall use the microarray data from (Lin, Vega, *et al.*, 2004), which was generated as part of the effort to identify genes regulated by *estrogen receptor*. Estrogen receptor (ER) is a nuclear hormone receptor, and a transcription factor, that gets activated by estrogen or E2, an estrogen agonist. The data were obtained by hybridizing human cell lines that have received and have not received (to act as the control or baseline) E2 treatment into microarrays containing ~18,000 genes, i.e. $N \approx 18,000$. This was done across 16 timepoints (i.e. B=16), namely from 1 to 8 hours after treatment with an hour intervals and from 10 to 24 hours with two hours intervals. Computational analysis of this set is hoped to produce a list of genes that are directly responsive towards estrogen. Biological

experimentations are no doubt still required to confidently ascertain the response of the genes.

After the initial publication of our work, a list of ~370 experimentally determined ER responsive genes was subsequently published as in the Estrogen Responsive Genes Database (ERGDB; Tang *et al.*, 2004). Although timecourse data is somewhat abundant, such accompanying list of direct targets is quite rare. In our experiments, this list acts as the list of positives (i.e. genes that are directly regulated by estrogen). Note that absence from the list does not immediately translate into a gene being a real negative. We can only say that those genes not in the list are putatively negative. For simplicity, however, we assume that they are negatives.

Experimental setup

The matrix X was obtained by taking the log of the expression ratios between the treated versus untreated cell lines. From this $\sim 18,000 \times 16$ matrix, our task is to identify the genes that are directly regulated by estrogen. During our preliminary analysis, we observed that timepoints 4 and 5 exhibit unusual behaviours, including significant presence unsually high log-expression ratio values. Depending on the nature of the algorithm, excluding them might be beneficial. In our experiment runs, we tested both with and without these timepoints.

For performance measure, we opt to plot the ROC curves obtained from each of the methods. The area under the ROC curves (AUC of ROC curves) provides a quick and useful measure for comparing different ordering of genes. The (normalized) AUC value ranges from 0 to 1, where 1 signifies the perfect ranking and 0.5 indicates ranking that could be due to chance alone.

Unsupervised approaches

For this microarray expression data we experimented with the friendly neighbor method, statistical ranking approaches, and a clustering based ranking procedure as discussed above. In applying the friendly neighbour approach, three different similarity measures were employed: sign match, longest consecutive sign match, and Pearson correlation. The ranking based on hierarchical clustering output is done as described in Section 3.3.2, using Pearson correlation, as the similarity function, and average linkage clustering. Note that the output of a hierarchical clustering is, to some extent, depending on the order of the dataset. To remove this bias, we scrambled the ordering of the genes in the input before clustering them. This was done five times, and the average of the performance is reported.

Supervised approaches

The list of known and verified estrogen responsive genes allows us to experiment with supervised algorithms, serving as both a comparative "ideal performance" for unsupervised algorithms as well as to gain insight on the nature of the data. For this SVM-based and *k*NN-based rankings were employed. Both SVM and kNN has been shown to work reasonably well on microarray data. We use the SVMlight (Joachims, 1998) implementation of SVM, and -unless stated otherwise- the default settings of the parameters were kept. Several kernels were applied, namely linear kernel, polynomial kernel, and the radial basis function (RBF) kernel. Pearson correlation is used as the similarity measure for *k*-NN.

In our evaluation, we performed the following steps 100 times and the average performance was reported:

a) split the dataset into a training set, with 2/3 of the examples, and a test set, containing 1/3 of the examples,

b) use the training set to train the classifier, and

c) measure its performance on the test set.

Unsupervised Algorithm	E2 Complete	E2 Excluding time 4 & 5
FN (SM, θ=0.75)	0.701	0.722
FN (SM, θ=0.80)	0.704	0.723
FN (SM, θ=0.85)	0.710	0.723
FN (SM, θ=0.90)	0.701	0.719
FN (SM, θ=0.95)	0.669	0.700
FN (SM, θ=1.00)	0.668	0.699
FN (LCSM, θ=0.65)	0.638	0.667
FN (LCSM, θ=0.75)	0.623	0.696
FN (LCSM, θ=0.85)	0.673	0.698
FN (LCSM, θ=0.95)	0.665	0.701
FN (LCSM, θ=1.00)	0.675	0.703
FN (PC, θ=0.25)	0.645	0.723
FN (PC, θ=0.50)	0.638	0.727
FN (PC, θ=0.75)	0.679	0.725
FN (PC, θ=0.95)	0.705	0.721
FN (PC, θ=1.00)	0.494	0.485
T-test	0.489	0.464
Wilcoxon rank-sum test	0.424	0.409
Maximum of absolute	0.533	0.586
Variance	0.539	0.601
Dynamic range	0.528	0.587
Hierarchical Clustering-based	0.578	0.411

Table 2. The performance of unsupervised algorithms for detecting estrogen responsive genes, measured by calculating the area under the ROC curves. The Friendly Neighbour (FN) approach employed normalized sign match (SM), normalized longest consecutive sign match (LCSM), and Pearson Correlation (PC). The thresholds were varied to observe their effect to the performance. For comparison, statistics-based and clustering-based ranking were performed. The hierarchical clustering used pearson correlation and average linkage.

Results

Unsupervised algorithms

Table 2 gives the performance results for each method. Evidently, the Friendly Neighbour methods consistently showed a good performance. Note that the Pearson correlation measure used here is not normalized (i.e. it ranges from -1 to 1). Hence a threshold of 0.5 in PC roughly corresponds to a threshold of 0.75 under a normalized similarity function. Ignoring the fourth and fifth timepoints benefit algorithms that are based on FN and that make use of the dynamics of the expression ratios. This indicates that timepoint 4 and 5 are somewhat erroneous.

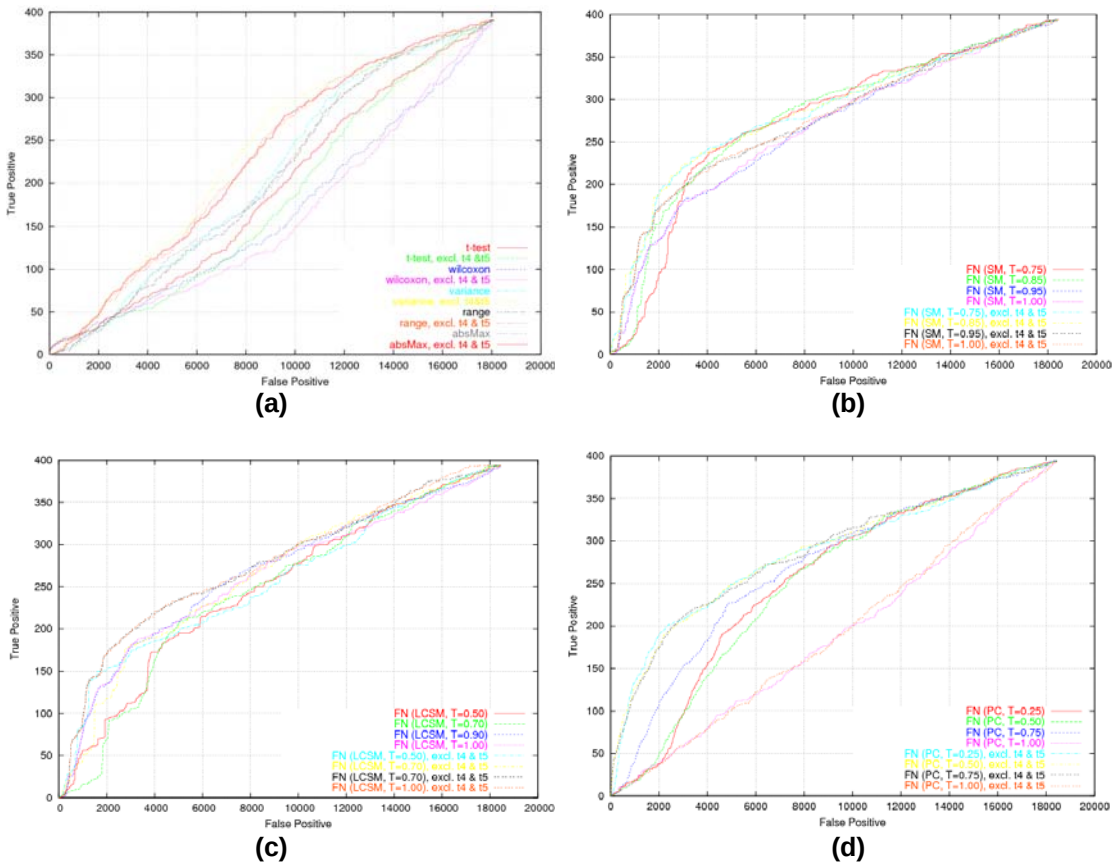


Figure 4. ROC curves for (a) non-FN unsupervised algorithms, (b) FN with sign match, (c) FN with longest consecutive sign match, and (d) FN with Pearson Correlation. Among the unsupervised methods, FN with SM/LCSM consistently showed good performance. FN with PC is somewhat sensitive to the threshold, which can be observed more clearly in Fig. 5.

Figures 4a to 4d show the actual ROC curves for the different unsupervised methods and FN-based methods using different similarity measures, under various thresholds. Overall, the FN-based rankings offer the best and stable performance. Care must be taken when using FN with Pearson correlation, as it seems that the performance is affected rapidly as the threshold is nearing 1 (see Fig. 5).

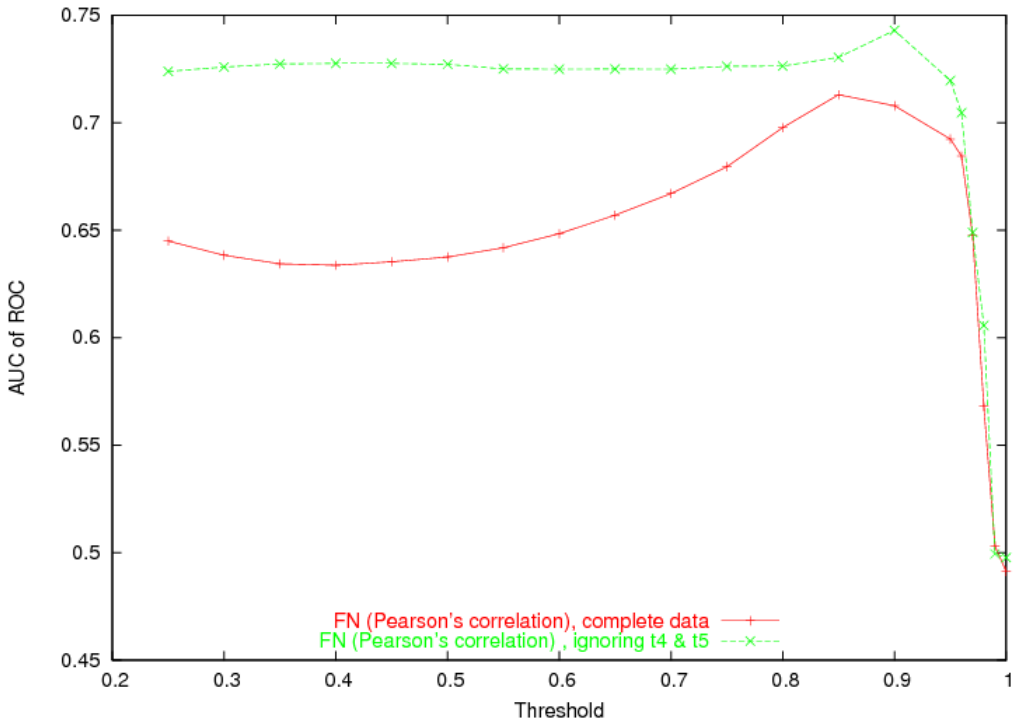


Figure 5. Area under the ROC curves for different threshold settings for Friendly Neighbour with Pearson correlation as the similarity measure.

Supervised algorithms

The results of the two classification algorithms are about the same (see Table 3). Both the *k*-NN and SVM (under various settings) reported a performance of around 0.75 (AUC of ROC). Under SVM, a cost factor ratio (between making errors on positive examples to making errors on negative examples) of 60 seems to work well. This is in line with the actual fact that negative examples are roughly 60 times more than the positive ones. Inclusion or exclusion of the two noise timepoints (the fourth and fifth) appear to have non-significant and non-consistent effect to the performance of the two

classification algorithms. The steady results made under various *k* for k-NN hinted that the positive examples are somewhat proximal to each other.

Supervised Algorithm	E2 Complete	E2 Excluding time 4 & 5
SVM (linear, j=1)	0.706	0.681
SVM (linear, j=30)	0.764	0.763
SVM (linear, j=60)	0.765	0.765
SVM (linear, j=90)	0.756	0.759
SVM (linear, j=100)	0.750	0.755
SVM (linear, j=150)	0.649	0.638
SVM (RBF, γ=0.25, j=1)	0.746	0.746
SVM (RBF, γ=0.25, j=60)	0.754	0.775
SVM (RBF, γ=0.5, j=60)	0.746	0.739
SVM (RBF, γ=1, j=60)	0.739	0.741
SVM (RBF, γ=2, j=60)	0.739	0.749
SVM (RBF, γ=4, j=60)	0.738	0.741
SVM (Poly d=2, j=1)	0.746	0.745
SVM (Poly d=2, j=60)	0.756	0.756
SVM (Poly d=3, j=60)	0.763	0.747
SVM (Poly d=4, j=60)	0.764	0.753
SVM (Poly d=5, j=60)	0.766	0.756
kNN (k=1)	0.715	0.705
kNN (k=3)	0.721	0.729
kNN (k=5)	0.734	0.734
kNN (k=7)	0.74	0.733
kNN (k=9)	0.741	0.742
kNN (k=15)	0.748	0.752
kNN (k=21)	0.746	0.745

Table 3. Performance of the supervised algorithms, under various settings. Three types of kernel were used. To compensate for the lack of positive examples (only ~370 of ~18,000 genes are known to be responsive), their importance is elevated (through parameter j). Overall, the performance of supervised algorithms is good, about 0.75 on the average.

As expected, supervised algorithms outperformed unsupervised algorithms. It is worth to note, nevertheless, that the friendly neighbour methods' performance tops those among other unsupervised approaches and is still comparable to the supervised ones. Performance increase for unsupervised algorithms might be attainable if we combine multiple approaches. Additionally, we have also showed that the FN framework can be applied for the detection of cell-cycle regulated genes (Karuturi and Vega, 2004).

Chapter 4

Inferring Regulatory Signals in Genomic Sequences

4.1 Overview

As described in Section 2.4, with regard to deciphering regulatory signals in the genome, we focused on the recent development of high-throughput sequencing-based localization of TF-DNA interaction sites, in particular towards a comprehensive analysis of data generated using the Chromatin-ImmunoPrecipitation (ChIP) Paired-End diTagging (PET) approach (Wei et al., 2006) developed within the Genome Institute of Singapore. Briefly, the ChIP-PET protocol couples enrichment of DNA fragments involved in TF-DNA interactions (through a ChIP assay) with efficient sequencing of the fragments' ends.

The Chromatin Immuoprecipitation (ChIP) assay (see also Section 2.1.2), a powerful approach to study in vivo protein-DNA interactions, consists of five major steps: (i) cross-link the DNA binding proteins to the DNA in vivo, (ii) shear the chromatin fibers (using sonication or otherwise) to a certain range of fragment size, (iii) immunoprecipitate the chromatin fragments using specific antibody against given protein targets, (iv) reverse the cross-linking of protein-bound DNA, and (v) analyze the ChIP enriched DNA fragments. These DNA fragments can then be profiled using low throughput methods, e.g. real-time qPCR, as well as high throughput approaches, such as hybridization-based ChIP-chip analysis (Iyer *et al., 2001*; Ren *et al.*, 2000;

Horak *et al.*, 2002; Weinmann *et al.*, 2002) or direct DNA sequencing, as mentioned in Section 2.4.

The sequencing approaches have their advantages over the hybridization-based approaches by elucidating the exact nucleotide content of target DNA sequences. In a ChIP-PET experiment, 5' (18bp) and 3' (18bp) signatures for each of the ChIP enriched DNA fragments were extracted and joined to form the paired end tag structure (PET or ditag) that were then concatenated for efficient sequencing analysis. The PET sequences were then mapped to the reference genome to infer the full content of each of the ChIP DNA fragments. As such, the paired-end sequencing has the benefit being able to determine the genomic source of a fragment without sequencing the fragment in its entirety. Thus allowing much more fragments to be sequenced and inspected. Figure 6 illustrates the typical processes in the construction of a ChIP-PET library.

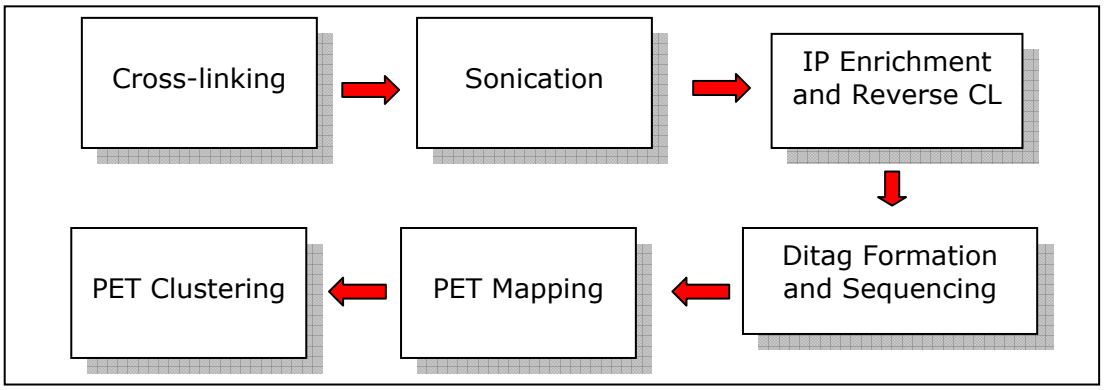


Figure 6. A schematic of typical stages in the construction of a ChIP-PET library. Cross-linking "freezes" the chromatin configuration, including TF interaction with DNA. Sonication cuts the DNA into much manageable fragments. The immunoprecipitation (IP) stage captures fragments cross-linked with the desired TF. Reverse cross-linking frees the DNA fragments, which are then sequenced at their two ends. The sequenced ends are then mapped into the reference genome. The mapped ditags (or PETs) are then clustered.

We addressed five issues in our study: (i) conducting preliminary assessment on the quality of a given library, (ii) constructing a better model of ChIP fragment lengths, (iii) modeling of ChIP fragment distribution in the whole-genome, (iv) identifying the true transcription factor binding regions, and (v) minimizing the effect of aberrant genome. All these were carried out in the context of ChIP-PET data, although the techniques and approaches were definitely general enough to be applied for data generated using other platforms.

4.2 Initial Assessments of ChIP-PET Library

4.2.1 Sequencing Saturation Analysis

The appeal of ChIP-PET (or other htsChIP protocols) comes from the potential of being able to map transcription factor binding sites in an unbiased manner across the whole genome. Prior to analyzing any given ChIP-PET library in depth, the first question to ask is whether we have collected enough fragments to be confidently say that we have a complete genome-wide coverage or, at the very least, to know the caveats and limitations of the given library when pursuing further analyses. We want to know the fragment sampling has reached a certain *saturation level* (given the experimental and technological limitations). That is to say, we want to assess how much information would extra sequencing add to the current library. If the library is fully saturated, extra sequencing should only replicate the already known *useable information*. In this analysis, the usable information is uniquely mapped PET fragment. Figure 7 reviews the processing stages involved in the ChIP-PET mapping pipeline. The uniquely mapped PET fragment is obtained at the end of this pipeline.

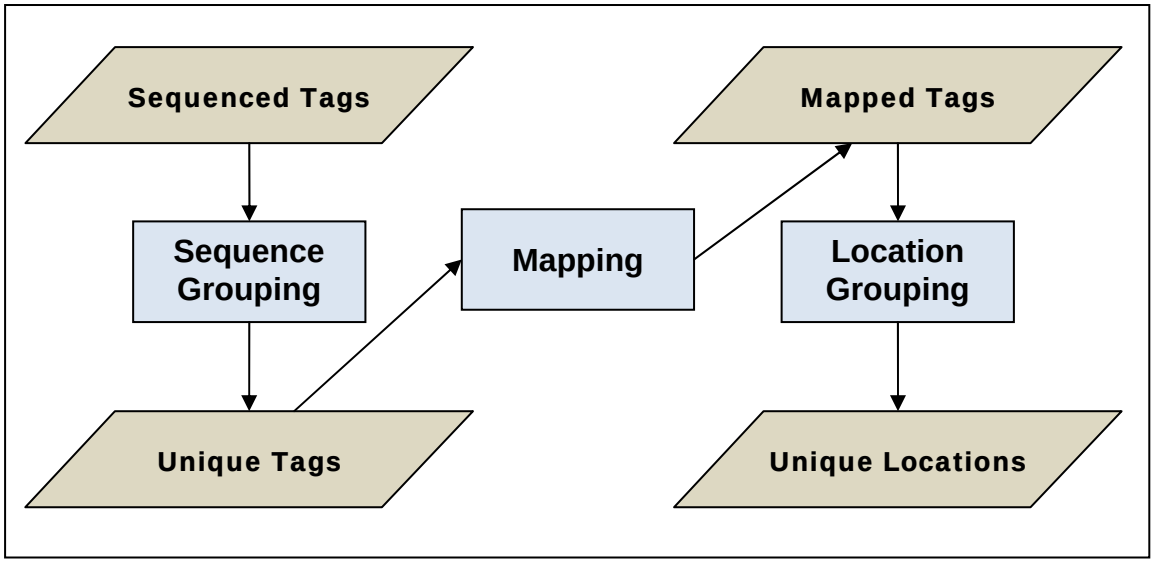


Figure 7. Four stages in PET mapping. Partially adapted from (Chiu *et al.*, 2006). Sequenced ditags are first group into unique tags, based on sequence similarity. These unique tags are then mapped to the genome and further grouped based on location.

We used the Hill function (Hill, 1910) to model the growth of usable information (i.e. uniquely mapped fragment) as a function of total sequences produced. The Hill function has been shown to be useful in modeling dynamics of gene expression level (Alon, 2006; Kuznetsov *et al.*, 2002). The exact formula for Hill function is:

$$f(x) = \frac{ax^b}{c^b + x^b} \qquad \textbf{(4.2.1)}$$

Where x is the total number of PETs sequenced (i.e. the size of "Sequenced Tags" input in Fig. 7), $f(x)$ is the number of unique locations recovered (i.e. the size of "Unique Locations" output in Fig. 7), a is the maximum number of recoverable unique location in the library, and b and c are positive constants. To estimate the saturation level of a given library, we perform the following steps:

1. If chronological sequencing data is available, generate an empirical curve of the number of total unique location obtained (y-axis) as a function of total number of PETs sequenced (x-axis). If not, progressively sample the library (without replacement) to construct the empirical curve.
2. Fit the Hill function to empirical curve. In our implementation, we make use of the nonlinear least-squares Marquardt-Levenberg algorithm (Bates and Watts, 1988) to perform the fitting.
3. Report the fraction of total unique location observed divided by the estimated maximum (a) as the saturation level of the library. Estimation done without chronological sequencing data is estimated as the average of multiple runs (typically 100 runs). Note that in practice, the fitting sometimes required manual intervention (in terms of adjusting the initial values), for example when local minima were reported and visual inspection showed erroneous fitting.

Figure 8 shows an example of such Hill function fitting to assess the saturation level of the ER ChIP-PET library (Lin *et al.*, 2007).

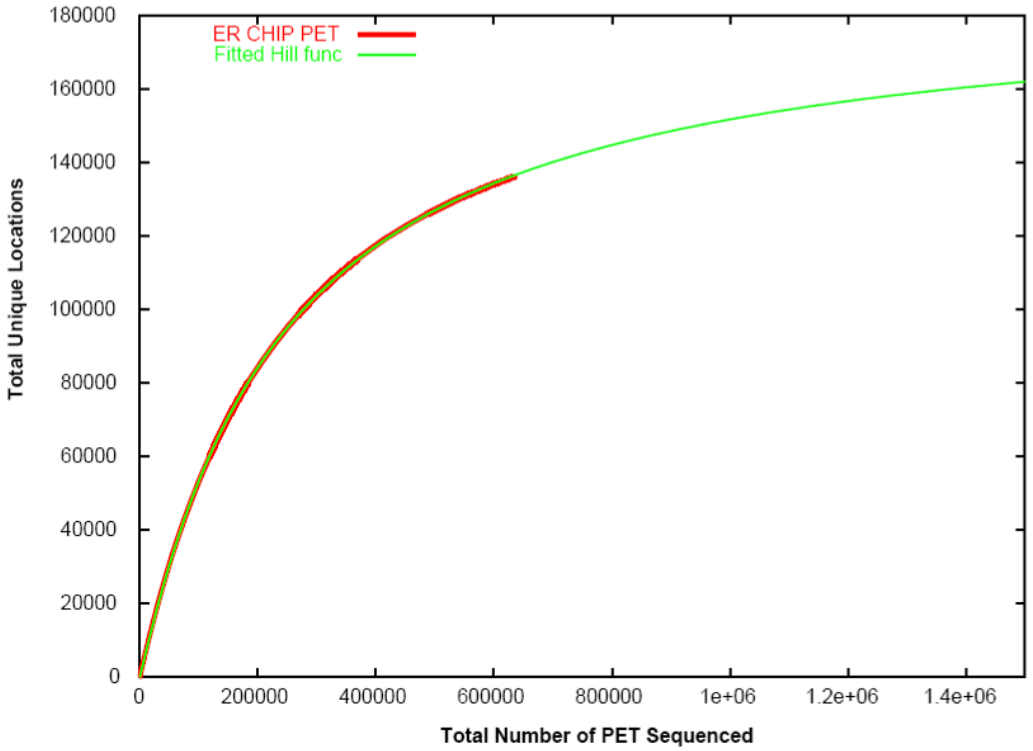


Figure 8. Saturation analysis of the ER ChIP-PET library. Fitting of Hill function (green curves) to the empirical chronological sequencing data (red curve) showed that the ER ChIP-PET library reached 73.23% of the saturated level.

Ideally, such saturation analysis should be embedded into the automated pipeline of ChIP-PET library construction. This would allow feedback into the system should the saturation is not sufficient. We noted two weaknesses of the current saturation estimation procedure that inhibit its incorporation into the automated pipeline, namely: (i) the need of manual intervention during the fitting process, and (ii) the time taken for running multiple fittings should chronological data be missing. Even with presence of chronological data, a considerable manual manipulation of the data was still needed, due to file formats and other issues. Observing that saturation is essentially a measurement of multiplicity, i.e. the number of sequenced PETs that identify a unique location, we developed the *Multiplicity Index* to roughly gauge the relative saturation level across different libraries. Multiplicity is created when two or more PETs are merged or grouped into one. Such merging happens twice in the mapping pipeline (see Fig. 7): (i) grouping of Sequenced Tags into Unique Tags, and

(ii) merging of Mapped Tags into Unique Locations. We define Multiplicity Index (MI) as:

$$MI = \sqrt{A \times B} \qquad \textbf{(4.2.2)}$$

$$A = stag / utag$$
$$B = mtag / uloc$$

Where *stag*, *utag*, *mtag*, and *uloc* are the number sequenced tags, unique tags, mapped tags, and unique locations respectively. The ratio between *stag* and *utag*, i.e. *A*, can be viewed as the multiplicity factor obtained during sequence clustering. The ratio between *mtag* and *uloc*, i.e. *B*, can be viewed as the multiplicity factor achieved after PET mapping. The MI is then the geometric average of the two multiplicity factors.

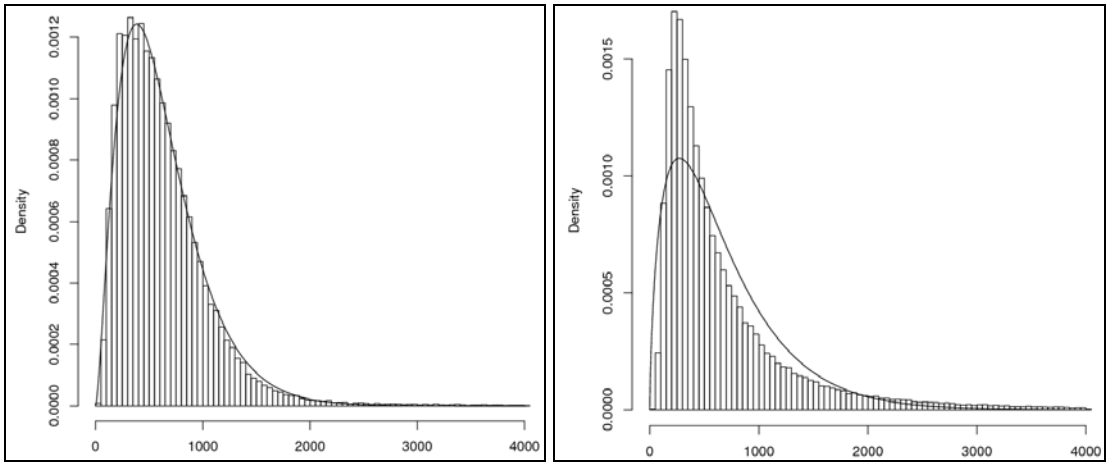
Using seven ChIP-PET libraries (p53 ChIP-PET (Wei *et al*., 2006), ER ChIP-PET (Lin et al., 2007), Oct4 ChIP-PET (Loh *et al.*, 2006), Nanog ChIP-PET (Loh *et al.*, 2006), Sox2 ChIP-PET (data unpublished), PPARγ ChIP-PET (Hamza *et al.*, under review), RXR ChIP-PET (Hamza *et al.*, under review)), we estimated their saturation levels as described earlier and computed their Multiplicity Indices (see Table 4). We observed that the two values were significantly correlated (Pearson's r = 0.9516; p-value 9.64e-4). This correlation means that we can use the Multiplicity Index to give a rough indication of the saturation level of the library. Note however that the Multiplicity Index is a relative indicator which could not be directly translated into saturation level.

Library	Saturation	Multiplicity Index
p53	79.466%	2.40141
ER	73.233%	1.82667
PPARγ	62.684%	1.78874
RXR	65.204%	1.77775
Oct4	27.964%	1.18124
Sox2	27.541%	1.16744
Nanog	19.641%	1.12613

Table 4. Comparison of estimated saturation level and Multiplicity Index (MI). Multiplicity Index correlates well with the estimated saturation. However, their direct mathematical relationship is not apparent.

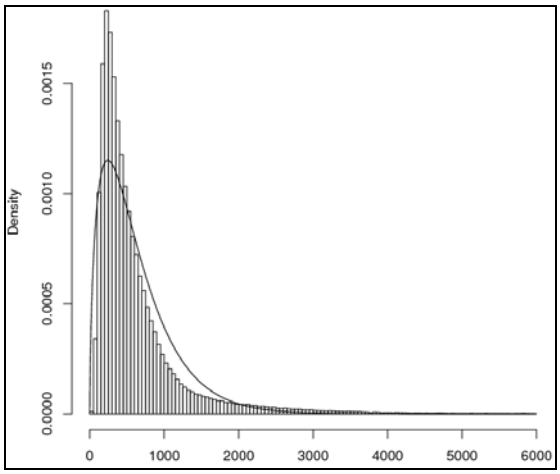
4.2.2 Modeling ChIP-PET Fragment Length

The characterization of both ends in the ChIP-PET protocol offers an additional advantage of being able to precisely model the distribution of ChIP fragment. ChIP fragment length is an important parameter in analyzing genome-wide ChIP library (see Sections 4.3 and 4.4 below, and (Qi *et al.*, 2006)). Qi *et al.* (2006), who used the fragment length in construction the "influence function" that models the spread of signals from a given binding site to its surrounding, suggested modeling the fragment length as a Gamma distribution. Using ChIP-PET libraries, we can assess the accuracy of this model. For a given ChIP-PET library, we fitted the Gamma fragment length model by first constructing a frequency histogram of ChIP-PET lengths based 50bp bins and fitting the Gamma distribution to the empirical distribution using the nonlinear least-squares Marquardt-Levenberg algorithm. Manual intervention in terms of adjusting the initial values was done whenever necessary. Figure 9 shows the best Gamma fitting for six ChIP-PET libraries (p53, ER, Oct4, Sox2, Nanog, and NF-κB (Lim *et al.*, 2007)).

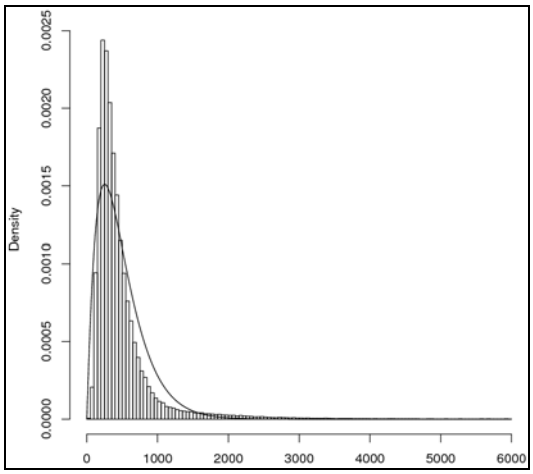


(**a**) p53

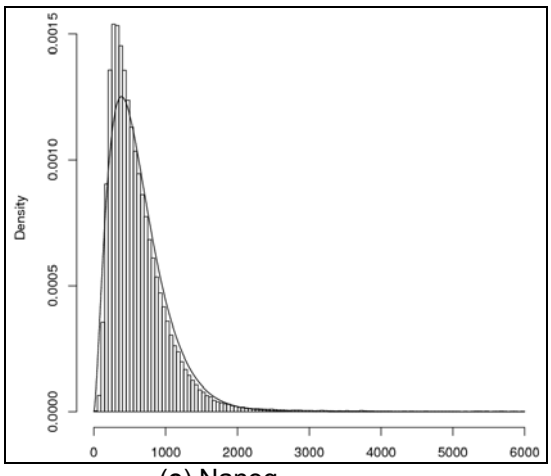
(**b**) ER



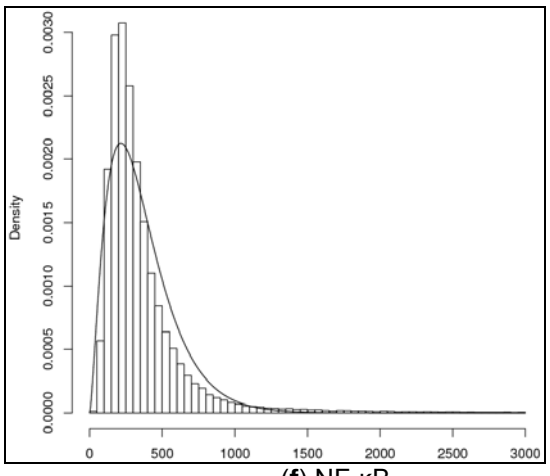
(**c**) Oct4



(**d**) Sox2



(**e**) Nanog



(**f**) NF-κB

Figure 9. Fitting Gamma distribution to ChIP fragment length. The *x*-axis and *y*-axis represent the fragment length and the fraction of fragments having certain length. Although the fragment distribution of p53 ChIP-PET library (**a**) was reasonably good, Gamma distribution could not fit the other five libraries: (**b**) ER, (**c**) Oct4, (**d**) Sox2, (**e**) Nanog, and (**f**) NF-κB.

Gamma distribution appeared to fit the fragment size distribution from the p53 ChIP-PET library reasonably well. However, when fitted on the other five libraries' fragment lengths, Gamma distribution could not model them well, even after manual intervention attempts.

Normal-Exponential Convolution

We observed that the Gamma distribution underestimated the amount of short length fragments (100-300bp) while overestimated the proportion of medium length fragments (600-1500bp). It seemed that there were intense accumulations of short fragments. If the genome was truly sheared randomly through the sonication process and that all nucleotides in the genome were equally likely to be shear, then in fact the length distribution is expected to follow an exponential distribution. Gamma distribution, $G(s,c)$, allows an additional flexibility of not having all points equally probable to serve as the shearing point, but it still imposes a uniform mean distance (characterized by the scale parameter *c*) between shearing points and/or muted-shearing points and expects a fixed number of muted-shearing points between shearing points (reflected by the shape parameter *s*).

Plots in Fig. 9 suggest that there is a kind of minimum fragment length where the probability of obtaining fragments shorter than that is significantly and rapidly decreasing. This notion was also reflected in the EMSA gel-shift images produced from the ChIP fragment (data not shown; obtained from colleagues at the Genome Institute of Singapore). The images showed a kind of thick band around the shorter end of fragment lengths. We postulated that in addition to the random shearing points, there are "unbreakable regions" or "atomic sizes" of fragments that prevent the

fragments from being sheared below certain lengths. The in-between regions, on the other hand, are sheared randomly. Figure 10 illustrates our proposed model.

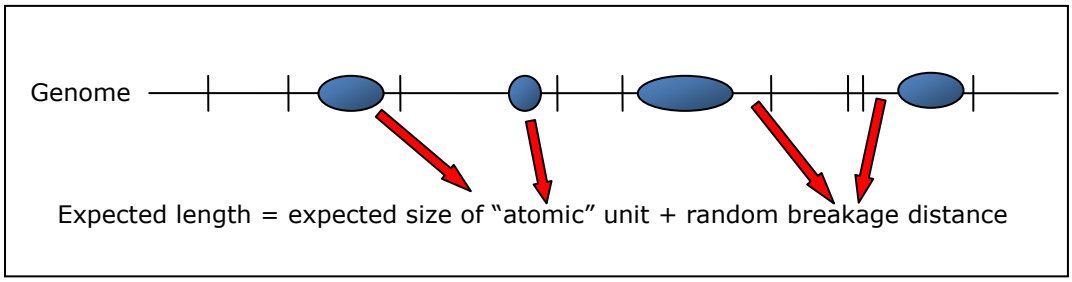


Figure 10. DNA shearing model with "atomic" units. This model takes into account the observed increase proportions of fragments with certain length.

While it is hard to ascertain the true origin of such atomic units, several sources are possible. This "atomic units" could be caused by the underlying biological constructs and structure, for example: the region could be "protected" by some protein complexes (e.g. nucleosomes or the transcription factors complexes). It could also be that the pseudo atomic length was an artifact of the limit of the shearing technology.

Under the new model, the length of a ChIP fragment is the sum of the atomic unit plus the distances between random shearing points. Since the shearing points are now assumed to be completely random, i.e. on the non-"atomic" region, the distance distribution should follow the exponential distribution (parameterized by the rate λ). Further, it is reasonable to assume that the size of these atomic units follows the normal distribution (with mean μ and standard deviation σ). The probability of a ChIP fragment having a length *x*-bp is $f(x;\mu,\sigma,\lambda)$ where it is a convolution of the normal and exponential distributions, as follow:

$$f(x;\mu,\sigma,\lambda) = N(x;\mu,\sigma) * Exp(x;\lambda)$$

Expanding further:

$$f(x;\mu,\sigma,\lambda)=\int_{-\infty}^{\infty}Exp(y;\lambda)\times N(x-y;\mu,\sigma)dy=\int_{0}^{\infty}Exp(y;\lambda)\times N(x-y;\mu,\sigma)dy$$

$$=\int_{0}^{\infty}\lambda e^{-\lambda y}\times\frac{e^{-(x-y-\mu)^2/2\sigma^2}}{\sigma\sqrt{2\pi}}dy=\frac{\lambda}{\sigma\sqrt{2\pi}}\int_{0}^{\infty}e^{-\lambda y}\times e^{-(x-y-\mu)^2/2\sigma^2}dy$$

$$=\frac{\lambda}{\sigma\sqrt{2\pi}}\int_{0}^{\infty}e^{-(x-y-\mu)^2/2\sigma^2-\lambda y}dy=\frac{\lambda}{\sigma\sqrt{2\pi}}\int_{0}^{\infty}e^{-\frac{(x-y-\mu)^2+2\sigma^2\lambda y}{2\sigma^2}}dy$$

$$=\frac{\lambda}{\sigma\sqrt{2\pi}}\int_{0}^{\infty}e^{-\frac{x^2-2xy-2\mu x+y^2+2\mu y+\mu^2+2\sigma^2\lambda y}{2\sigma^2}}dy=\frac{\lambda}{\sigma\sqrt{2\pi}}\int_{0}^{\infty}e^{-\frac{(-x+y+\mu+\lambda\sigma^2)^2+2\lambda x\sigma^2-2\lambda\mu\sigma^2-\lambda^2\sigma^4}{2\sigma^2}}dy$$

$$=\frac{\lambda}{\sigma\sqrt{2\pi}}\int_{0}^{\infty}e^{-\frac{(-x+y+\mu+\lambda\sigma^2)^2}{2\sigma^2}+\frac{-2\lambda x\sigma^2+2\lambda\mu\sigma^2+\lambda^2\sigma^4}{2\sigma^2}}dy$$

$$=\frac{\lambda}{\sigma\sqrt{2\pi}}e^{\frac{-2\lambda x\sigma^2+2\lambda\mu\sigma^2+\lambda^2\sigma^4}{2\sigma^2}}\int_{0}^{\infty}e^{-\frac{(-x+y+\mu+\lambda\sigma^2)^2}{2\sigma^2}}dy=\frac{\lambda}{\sigma\sqrt{2\pi}}e^{\lambda(\mu-x)+\frac{\lambda^2\sigma^2}{2}}\int_{0}^{\infty}e^{-\left(\frac{y-x+\mu+\lambda\sigma^2}{\sigma\sqrt{2}}\right)^2}dy$$

$$=\frac{\lambda e^{\lambda(\mu-x)+\frac{\lambda^2\sigma^2}{2}}}{\sigma\sqrt{2\pi}}\int_{0}^{\infty}e^{-\left(\frac{y-x+\mu+\lambda\sigma^2}{\sigma\sqrt{2}}\right)^2}dy=\frac{\lambda e^{\lambda(\mu-x)+\frac{\lambda^2\sigma^2}{2}}}{\sigma\sqrt{2\pi}}\int_{0}^{\infty}e^{-\left(\frac{y-x+\mu+\lambda\sigma^2}{\sigma\sqrt{2}}\right)^2}dy$$

Let $z=t(x)=\frac{y-x+\mu+\lambda\sigma^2}{\sigma\sqrt{2}}$, and as such $dz=dt(x)=\frac{1}{\sigma\sqrt{2}}dy$

The above formulation for *f(x)* can be rewritten as:

$$f(x;\mu,\sigma,\lambda)=\frac{\lambda e^{\lambda(\mu-x)+\frac{\lambda^2\sigma^2}{2}}}{\sigma\sqrt{2\pi}}\int_{0}^{\infty}e^{-\left(\frac{y-x+\mu+\lambda\sigma^2}{\sigma\sqrt{2}}\right)^2}dy=\frac{\lambda e^{\lambda(\mu-x)+\frac{\lambda^2\sigma^2}{2}}}{\sigma\sqrt{2\pi}}\int_{t(0)}^{t(\infty)}e^{-z^2}\sigma\sqrt{2}dz$$

$$=\frac{\lambda e^{\lambda(\mu-x)+\frac{\lambda^2\sigma^2}{2}}}{\sigma\sqrt{2\pi}}\times\sigma\sqrt{2}\int_{t(0)}^{t(\infty)}e^{-z^2}dz=\frac{1}{2}\lambda e^{\lambda(\mu-x)+\frac{\lambda^2\sigma^2}{2}}\times\frac{2}{\sqrt{\pi}}\int_{t(0)}^{t(\infty)}e^{-z^2}dz$$

$$=\frac{1}{2}\lambda e^{\lambda(\mu-x)+\frac{\lambda^2\sigma^2}{2}}\times\frac{2}{\sqrt{\pi}}\left(\int_{0}^{t(\infty)}e^{-z^2}dz-\int_{0}^{t(0)}e^{-z^2}dz\right)=\frac{1}{2}\lambda e^{\lambda(\mu-x)+\frac{\lambda^2\sigma^2}{2}}\times\left(erf\left(t(\infty)\right)-erf\left(t(0)\right)\right)$$

$$=\frac{1}{2}\lambda e^{\lambda(\mu-x)+\frac{\lambda^2\sigma^2}{2}}\times\left(erf(\infty)-erf\left(\frac{\mu-x+\lambda\sigma^2}{\sigma\sqrt{2}}\right)\right)$$

$$=\frac{1}{2}\lambda e^{\lambda(\mu-x)+\frac{\lambda^2\sigma^2}{2}}\times\left(1-erf\left(\frac{\mu-x+\lambda\sigma^2}{\sigma\sqrt{2}}\right)\right)$$

Hence, the probability density function for the ChIP fragment length under the new model is:

$$f(x;\mu,\sigma,\lambda) = \frac{1}{2}\lambda e^{\lambda(\mu-x)+\frac{\lambda^2\sigma^2}{2}} \times \left(1 - erf\left(\frac{\mu - x + \lambda\sigma^2}{\sigma\sqrt{2}}\right)\right) \qquad \textbf{(4.2.3)}$$

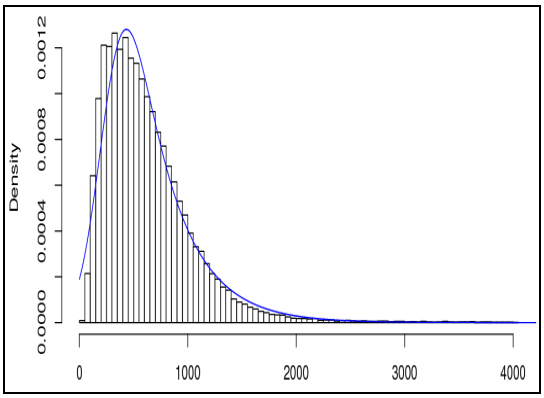
Where $erf(x)$ is the error function.

Evaluation of the ChIP Fragment Length Model

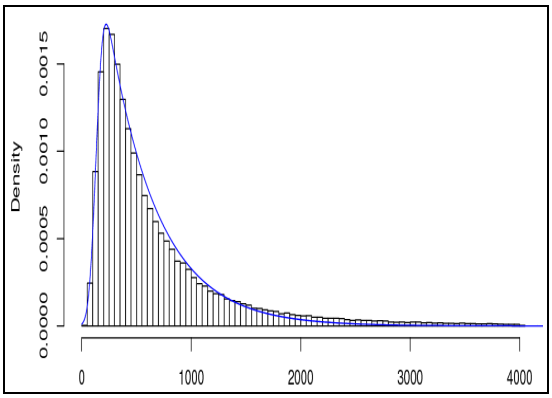
To evaluate our proposed model, we took the same six ChIP-PET libraries and similarly fitted the Normal*Exponential distribution to 50bp binned histogram of ChIP-PET lengths using the nonlinear least-squares Marquardt-Levenberg algorithm. The fitted parameters are tabulated in Table 5 and fitted curves are shown in Fig. 11. The proposed Normal*Exponential distribution were able to model the ChIP-PET fragment lengths of the six libraries very well and generally much better than the Gamma distribution (Fig. 9). Interestingly, we observed in the fitted parameters for the atomic unit that the mean (μ) was around one nucleosome (~146bp) and the overall size of the atomic unit is around one or two nucleosomes, supporting the hypothesis that nucleosome structure might play a part in protecting a region from being sheared.

Library	μ	σ	1/λ
p53	197.3	136.01	437.8284
ER	133.9	51.74	452.9234
Oct4	131.4	55.14	408.4967
Sox2	159.2	57.86	262.3102
Nanog	191.8	73.25	440.7616
NF-κB	132.5	50.91	192.3232

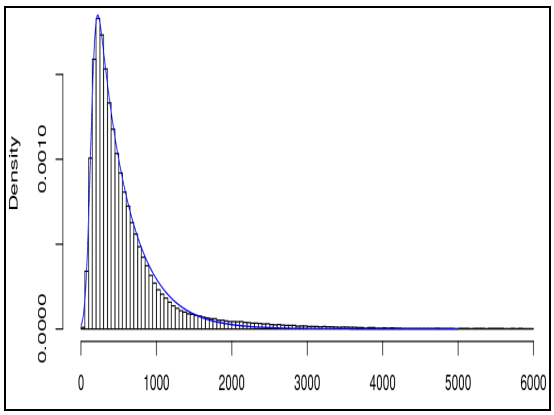
Table 5. Parameters of Normal*Exponential distribution fitted to PET fragment length. The mean (μ) and standard deviation (σ) of the atomic unit seemed to fluctuate around the size of one to two nucleosomes. 1/λ was tabulated for the exponential part to give a sense of the average distances between random shearing points.



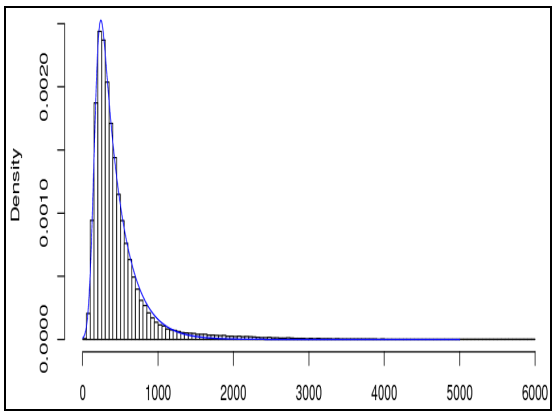
(**a**) p53



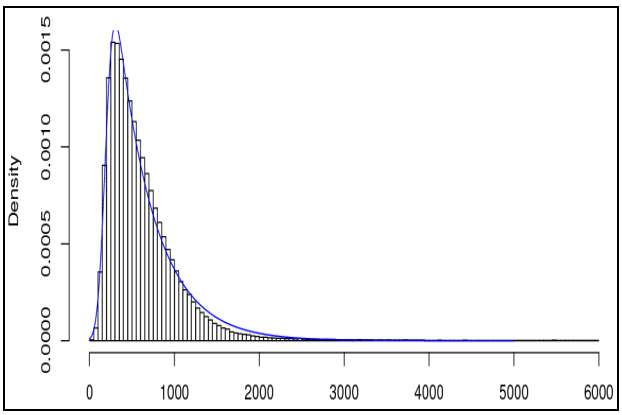
(**b**) ER



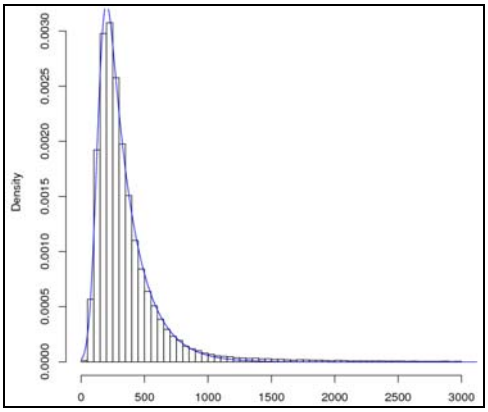
(**c**) Oct4



(**d**) Sox2



(**e**) Nanog



(**f**) NF-κB

Figure 11. Curves of fitted Normal*Exponential distribution to ChIP fragment length. The *x*-axis and *y*-axis represent the fragment length and the fraction of fragments having certain length. Six libraries were used: (**a**) p53, (**b**) ER, (**c**) Oct4, (**d**) Sox2, (**e**) Nanog, and (**f**) NF-κB. The Normal*Exponential distribution had better fit better than the Gamma distribution (see Fig. 9).

4.3 Modeling Genome-Wide Distribution of ChIP Fragments

4.3.1 Problem Description

The ChIP experiment involves numerous factors that influence the quality and properties of the resultant libraries. The factors include: (i) number of actual bound regions, (ii) number of fragments sequenced, (iii) the quality of ChIP assay, (iv) size of the genome, and (v) fragment lengths. Note that a number of these factors are typically not directly measured in the context of htsChIP experiment. We asked ourselves whether we could provide some quantification on some of the unmeasured factors based on the available information, in particular the total number of TF-bound regions and a sense of ChIP enrichment strength.

Problem 4.1 (Parameterizing ChIP-PET Library) *Given a ChIP-PET library of* T *ditags mapped to a* G*-bp long reference genome, estimate the total number of binding regions and the signal strength of the underlying ChIP assay, in terms of ChIP enrichment over control*.

4.3.2 A Mathematical Model of ChIP-PET Library

Let T be the number of ChIP fragment sequenced and uniquely mapped to the reference genome of length L basepairs. Assume as well that the fragments are around k-bp in length. Let's suppose that we bin the genome into B bins of equal lengths (say, v-bp), and that the T PETs are distributed across these B bins. If the T fragments are completely random and their distribution is completely unbiased, then the distribution of number of PETs per bins (=X) should follow the Poisson distribution:

$$\Pr_{backg}(X \mid B, T) = \Pr_{pois}(X \mid \lambda = \tfrac{T}{B}) \qquad (\mathbf{4.3.1})$$

Now, let $\xi \in [0,1]$ be the fraction of B bins that contain binding sites and $\alpha \in [0,1]$ be the fraction of ChIP fragments that were bound by the relevant transcription factor. Among the $(\xi * B)$ bins, the PET accumulation rate is influenced by both the randomly distributed $(1-\alpha)T$ fragments distributed across B bins and by αT fragments distributed exclusively among $(\xi * B)$ bins as well. Thus:

$$\Pr_{bound}(X = x \mid B, \xi, T, \alpha) = \sum_{i=0}^{x} \Pr_{pois}(i \mid \lambda = \tfrac{\alpha T}{\xi B}) * \Pr_{pois}(x - i \mid \lambda = \tfrac{(1-\alpha)T}{B})$$

(**4.3.2**)

and

$$\Pr_{nonbound}(X \mid B, T, \alpha) = \Pr_{pois}(X \mid \lambda = \tfrac{(1-\alpha)T}{B}) \quad \text{(4.3.3)}$$

taken together

$$\Pr(X \mid B, \xi, T, \alpha) = \xi \Pr_{bound}(X \mid B, \beta, T, \alpha) + (1-\xi) \Pr_{nonbound}(X \mid B, T, \alpha)$$

(**4.3.4**)

Modeling non-uniform IP enrichment

The probability function above (Eq. 4.3.4) for computing the number TF-bound PETs sampled from a binding-site-containing bin assumes a fixed and constant binding affinity. In general, we can restate the formulation as:

$$\Pr_{bound}(X = x \mid B, \xi, T, \alpha) = \sum_{i=0}^{x} \Pr_{TF-bound}(i \mid B, \xi, T, \alpha) * \Pr_{pois}(x - i \mid \lambda = \tfrac{(1-\alpha)T}{B})$$

Let w_i be the relative binding signal strength of binding-site-containing bin *i*, such that

$$\sum_{i=1}^{\xi B} w_i = 1$$

Thus,

$$\Pr_{TF-bound}(X = i \mid B, \xi, T, \alpha) = \frac{1}{\xi B}\sum_{i=1}^{\xi B}\Pr_{pois}(i \mid \lambda = w_i \alpha T)$$

Assuming fixed and constant binding affinity across all bins, i.e. $\forall i : w_i = w_0 = \frac{1}{\xi B}$, the formulation above is simplified into

$$\Pr_{TF-bound}(X = i \mid B, \xi, T, \alpha) = \frac{1}{\xi B}\sum_{i=1}^{\xi B}\Pr_{pois}(i \mid \lambda = w_0 \alpha T) = \Pr_{pois}(i \mid \lambda = \frac{\alpha T}{\xi B})$$

which is equivalent to Eq. 4.3.2.

If the binding signal intensity follows Gamma distribution, i.e. $W \propto G(s,c)$, we need to model w_i carefully. Let us choose $G(s,c)$, where s is shape and c is scale, such that $E[G(s,c)] = 1$. Since $E[G(s,c)] = s * c$, then we can choose $c = 1/s$. Thus, to achieve $\sum_{i=1}^{\xi B} w_i = 1$, we can set $w_i \sim \frac{G(s,1/s)}{\xi B}$. Following this, the probability for a binding-site-containing bin to have i TF-bound PETs sampled follows the following distribution:

$$\begin{aligned}\Pr_{TF-bound}(i \mid B, \xi, T, \alpha) &= \frac{1}{\xi B}\sum_{j=1}^{\xi B}\Pr_{pois}(i \mid \lambda = w_j \alpha T) \\ &= \frac{1}{\xi B}\sum_{j=1}^{\xi B}\Pr_{pois}(i \mid \lambda = \frac{G(s, s^{-1})}{\xi B}\alpha T) \\ &= \Pr_{pois}(i \mid \lambda = \frac{G(s, s^{-1})}{\xi B}\alpha T)\end{aligned}$$

$$\Pr_{TF-bound}(i \mid B,\xi,T,\alpha) = \Pr_{pois}(i \mid \lambda = \frac{G(s,s^{-1})}{\xi B}\alpha T)$$

$$= \int \Pr_{pois}(i \mid \lambda = \frac{x}{\xi B}\alpha T) * \Pr_{\gamma}(x \mid s, s^{-1})\partial x$$

$$= \int \frac{e^{-\frac{x\alpha T}{\xi B}}\left(\frac{x\alpha T}{\xi B}\right)^{i}}{i!} * x^{s-1}\frac{e^{-\frac{x}{s^{-1}}}}{\Gamma(s)(s^{-1})^{s}}\partial x$$

$$= \int \frac{e^{-\frac{x\alpha T}{\xi B}}\left(\frac{x\alpha T}{\xi B}\right)^{i}}{i!} * x^{s-1}\frac{e^{-xs}s^{s}}{\Gamma(s)}\partial x$$

Thus,

$$\Pr_{TF-bound}(i \mid B,\xi,T,\alpha) =$$

$$= \int \frac{x^{s-1}e^{-\frac{x\alpha T}{\xi B}-xs}\left(\frac{x\alpha T}{\xi B}\right)^{i}s^{s}}{i!\Gamma(s)}\partial x$$

$$= \frac{s^{s}}{i!\Gamma(s)}\int x^{s-1}e^{-\frac{x\alpha T}{\xi B}-xs}\left(\frac{x\alpha T}{\xi B}\right)^{i}\partial x$$

$$= \left(\frac{\alpha T}{\xi B}\right)^{i}\frac{s^{s}}{i!\Gamma(s)}\int x^{s-1}e^{-\frac{x\alpha T}{\xi B}-xs}x^{i}\partial x$$

$$= \left(\frac{\alpha T}{\xi B}\right)^{i}\frac{s^{s}}{i!\Gamma(s)}\int x^{s+i-1}e^{-x\left(\frac{\alpha T}{\xi B}+s\right)}\partial x$$

$$= \left(\frac{\alpha T}{\xi B}\right)^{i}\frac{s^{s}}{i!\Gamma(s)}\left[-x^{s+i}E_{-i-s+1}((\frac{\alpha T}{\xi B}+s)x)\right]_{0}^{\infty}$$

$$= \left(\frac{\alpha T}{\xi B}\right)^{i}\frac{s^{s}}{i!\Gamma(s)}\left[-x^{s+i}((\frac{\alpha T}{\xi B}+s)x)^{-i-s}\Gamma(1+i+s-1,(\frac{\alpha T}{\xi B}+s)x)\right]_{0}^{\infty}$$

Further,

$$\Pr_{TF-bound}(i \mid B,\xi,T,\alpha) =$$

$$= \left(\frac{\alpha T}{\xi B}\right)^{i} \frac{s^{s}}{i!\Gamma(s)} \left[\frac{-x^{s+i}}{((\frac{\alpha T}{\xi B}+s)x)^{s+i}} \Gamma(s+i,(\frac{\alpha T}{\xi B}+s)x) \right]_{0}^{\infty}$$

$$= \left(\frac{\alpha T}{\xi B}\right)^{i} \frac{s^{s}}{i!\Gamma(s)\left(\frac{\alpha T}{\xi B}+s\right)^{s+i}} \left[-\Gamma(s+i,(\frac{\alpha T}{\xi B}+s)x) \right]_{0}^{\infty}$$

$$= \frac{\left(\frac{\alpha T}{\xi B}\right)^{i} s^{s}}{i!\Gamma(s)\left(\frac{\alpha T}{\xi B}+s\right)^{s+i}} \left[-\Gamma(s+i,\infty) - (-\Gamma(s+i,0)) \right]$$

Or more simply:

$$\Pr_{TF-bound}(i \mid B,\xi,T,\alpha) = \frac{\Gamma(s+i)}{i!\Gamma(s)} \left(\frac{\alpha T}{\xi B}\right)^{i} s^{s} \left(\frac{\alpha T}{\xi B}+s\right)^{-s-i} \qquad \textbf{(4.3.5)}$$

Estimating relative signal strength

Now that the model for genome-wide distribution of ChIP fragments has been developed, we are in the position to generate an estimation of the relative signal strength. We define *relative signal strength*, $z \geq 0$, as the average multiplicative factor of the number of fragments found in bins with binding sites compared bins with no binding site (i.e. no significant binding of TF). As such, $z = 2$ is interpreted that bins with binding sites has twice as many ChIP fragments as bins with no binding sites. Similarly, $z = 1$ means that there is no enrichment of fragments in bins with binding sites. This definition is in line with the typical quantification quoted for ChIP-

qPCR measurement, which is the current golden standard for detection of TF-DNA interaction.

Note that z directly influence α, the fraction of fragments that are bound by the TF. α can also be thought of as the probability of sampling a fragment that is bound by the TF from the total pool of fragments in the sample. If we assume that the sampling probability is roughly proportional to the size of the respective fragments and regions, then α can be estimated as:

$$\alpha \approx \frac{\xi \times B \times k \times z}{((1-\xi) \times L) + (\xi \times B \times k \times z)} \quad \textbf{(4.3.6)}$$

Recall that k is the expected length of fragments and L is the length of the genome. The first term of the denominator is the sampling weight of non TF-bound region, while the second term is the sampling weight of TF-bound region. Equation 4.3.6 can be further rewritten as:

$$\alpha \approx \frac{\xi \times B \times k \times z}{((1-\xi) \times L) + (\xi \times B \times k \times z)} = \frac{\xi \times k \times z}{((1-\xi) \times L/B) + (\xi \times k \times z)}$$

Since $v = L/B$ is the size of bin, then

$$\alpha \approx \frac{\xi \times k \times z}{((1-\xi) \times v) + (\xi \times k \times z)}$$

Rearranging the terms, we have:

$$\alpha((1-\xi) \times v) + \alpha(\xi \times k \times z) = \xi \times k \times z$$
$$\alpha((1-\xi) \times v) = (1-\alpha)(\xi \times k \times z)$$

Thus,

$$z = \frac{\alpha \times (1-\xi) \times v}{(1-\alpha) \times \xi \times k} \quad \textbf{(4.3.7)}$$

4.3.3 Evaluation

Dataset

To evaluate our model, we chose ChIP-PET libraries which we had access to the ChIP-qPCR validation data. This was critical since there was virtually no way for us to experimentally measure the true α and ξ, and instead we relied on evaluating the predicted z (which in turn is tightly dependent on the former two variables) against the actual readout from ChIP-qPCR.. The four datasets used in our evaluation were:

- The p53 ChIP-PET library published in (Wei *et al., 2006*). This library had 65,714 PETs uniquely mapped. The reference genome for this library was human genome build hg17 (~3.1 Gbp).
- The Oct4 and Nanog ChIP-PET libraries from (Loh *et al.*, 2006). In these libraries, a total of 366,639 PETs (Oct4) and 265,676 PETs (Nanog) were uniquely mapped to the mouse genome (UCSC mm5; ~2.6Gbp).
- The NF-κB ChIP-PET library reported in (Lim *et al.*, 2007). This library contained 177,437 PETs mapped to the human genome (UCSC hg17).

In addition to the real datasets, we generated a number of simulated datasets to explore the potential limitations of the current model as well as the limitations of the fitting procedure. For a given set of parameters (genome size, fragment size, total number of "binding sites", and enrichment ratio), a probability distribution spanning the specified genome length was constructed, taking into account the number of sites. The fragments were then "sampled" from this artificial genome based on the probability distribution. Note that α and ξ were computed from the above parameters.

Experimental setup

Given a ChIP-PET library, we first transformed the mapped tag data into a frequency table by grouping them into fixed bins of equal length, which in our experiments was fixed to 5kbp. The model (Eq. 4.3.4) was then fitted to the observed data by searching the α and ξ that minimized the sum of squared error (SSE) between the cumulative distribution function (CDF) of the observed data and the model. The choice of using the CDF, rather than probability density function (PDF), as the cost function is to counter the frequently detected noise of spurious spikes of tag densities in certain areas of the genome due to mapping or other issues. A grid-search algorithm was implemented for this fitting. Unless specified otherwise, we use an increment of 0.005 in estimating both α and ξ. In addition to finding the best α and ξ, we ran a series of bootstrapping iterations to estimate the stability of the estimates. The bootstrapping was done upon the bins, to account for systematic noise that might be present among the bins. One hundred bootstrapping iterations were done for each fitting. To assess the accuracy of the estimates of real datasetd, we compared the predicted relative signal strength (z) to the ChIP-qPCR output. Since relative signal strength is defined as enrichment of bound sites over non-bound sites, we contrasted the predicted z with summary statistics of ChIP-qPCR readings of bound sites (defined as enrichment greater than 2-fold). Note that qPCR experiment reports the multiplicative factor of DNA abundance at a given region between two distinct samples.

Experimental results

The results from parameter fitting on the four real libraries are tabulated in Table 6. The table shows estimates from a single run (using the complete observed data) and the bootstrapped runs. The estimates appeared to be quite stable, with the

bootstrapped runs producing a small variance. Based on the known and estimated parameters, we computed the relative signal strength (z) using Eq. 4.3.7. Across all libraries, the predicted z values were similar to the mean of the ChIP-qPCR fold enrichments of the binding sites and showed similar trend to that of the mean and median enrichment (see Table 7).

Library	Genome size (non-gap)	No. of PETs	Average PET length	Single Estimate			Bootstrapped (100 runs; bin-sampling)			
				Alpha	Xi	Relative Strength (z)	Alpha (mean)	Alpha (StDev)	Xi (mean)	Xi (StDev)
p53	~3.1Gbp	65,714	625	0.065	0.005	110.7	0.0709	0.0097	0.0066	0.0023
Oct4	~2.6Gbp	366,639	627	0.32	0.18	17.1	0.31645	0.0065	0.1764	0.0067
Nanog	~2.6Gbp	265,676	623	0.27	0.065	42.7	0.2726	0.0048	0.0662	0.0026
NF-kB	~3.1Gbp	177,437	361	0.165	0.06	42.9	0.16475	0.0129	0.0598	0.0099

Table 6. Alpha and Xi estimates for the four real libraries. The results from 100 bootstrapping iterations showed that the estimates were quite stable.

Library	ChIP-qPCR fold enrichment			
	Min	Median	Mean	Max
p53	11.56	95.51	160.7	900.6
Oct4	2.02	12.22	16.3	97.57
Nanog	2.56	15.53	32.95	201.2
NF-kB	2.6	25.3	33.93	183.3

Table 7. Summary statistics of ChIP qPCR validation for the real libraries.

Experiments using artificial datasets (see Table 8). In general the estimation managed to recapitulate the original parameters used to generate the artificial datasets. The estimates were also shown to be very stable under bootstrapping experiments. The only outliers (poor and unstable) of performance were observed when the dataset itself was too noise (see the two last rows of Table 8), i.e. alpha is very close or equal to 0. In such dataset, there is almost no distinction between calling no bin to be binding (i.e. alpha=0) or all bins to be bound (alpha=1), mathematically speaking.

Simulation setup	No. of PETs	Single Estimate		Bootstrapped (100 runs; bin-sampling)			
		Alpha	Xi	Alpha (mean)	Alpha (StDev)	Xi (mean)	Xi (StDev)
alpha=0.368, xi=0.00833, genome=3Gbp	50k	0.37	0.01	0.37	0	0.01	4.00E-10
	100k	0.37	0.01	0.37	0	0.01	4.00E-10
	150k	0.37	0.01	0.37	0	0.01	4.00E-10
alpha=0.47, xi=0.025, genome=3Gbp	50k	0.48	0.03	0.48	1.90E-08	0.03	0
	100k	0.48	0.03	0.4795	0.0021	0.03	0
	150k	0.48	0.03	0.4775	0.0043	0.03	0
alpha=0.84, xi=0.15, genome=3Gbp	10k	0.79	0.12	0.804	0.0147	0.1248	0.005
	50k	0.83	0.14	0.83	2.90E-08	0.14	1.16E-08
	100k	0.83	0.14	0.83	2.90E-08	0.14	1.16E-08
	150k	0.83	0.14	0.83	2.90E-08	0.14	1.16E-08
	200k	0.83	0.14	0.83	2.90E-08	0.14	1.16E-08
alpha=0.055, xi=0.016, genome=3Gbp	10k	0.06	0.02	0.0708	0.0223	0.0296	0
	50k	0.06	0.02	0.06	0	0.02	0
	100k	0.06	0.02	0.06	0	0.02	0
	150k	0.06	0.02	0.06	0	0.02	0
	200k	0.06	0.02	0.06	0	0.02	0
alpha=0.0207818, xi=0.15, genome=3Gbp	50k	0.01	0.07	0.252	0.2975	0.5745	0.4321
	100k	0.08	0.78	0.1993	0.2009	0.5339	0.4603
	150k	0.01	0.04	0.1508	0.1856	0.4274	0.4475
alpha=0, xi=0, genome=3Gbp	10k	0.81	1	0.8305	0.1366	0.9994	0.0024
	50k	0.54	1	0.5312	0.199	1	0
	100k	0.01	0.08	0.0796	0.1755	0.2966	0.3684
	150k	0.03	1	0.4581	0.4116	0.894	0.2651
	200k	0.16	0.99	0.2406	0.2299	0.7852	0.3668

Table 8. Alpha and Xi estimates for the artificial libraries under various settings.

4.4 Modeling Localized Enrichment of ChIP Fragments

4.4.1 Problem Description

The problem addressed in this section pertains to how ChIP fragments are enriched in finer resolution regions. Going beyond just distinguishing large regions, like in the previous Section 4.3, that are bound (i.e. binding regions) and not bound, we are mostly interested in determining the precise locations of the TF-DNA interactions (i.e. binding sites). We set ourselves to model the accumulation of ChIP fragments around binding site and around non binding site, in order to better identify the binding sites as well as to reduce false positive in our binding site calling.

Problem 4.2 (PETs Accumulation in Local Region) *Given a ChIP-PET library of* T *ditags mapped to a* G*-bp long reference genome, develop a model for fragment accumulation around binding site and non binding site. Additionally, compute the probability of chance accumulation for assessing the likelihood of a region being bound or not bound*.

4.4.2 Fragment Clustering

The protein-DNA interaction regions enriched by ChIP procedure will have more DNA fragments representing the bound regions than the non-bound regions. Therefore, with sufficient sequence sampling in the DNA pool of a ChIP experiment, multiple DNA fragments originated from the bound regions will be encountered, while the non-bound regions will contribute no or minimal number of fragments (which can be constitutively categorized as background nois). As such, assuming that bound fragments should cover the actual binding sites, clustering of fragments would give us an indication of the precise location of actual binding sites.

The primary ChIP-PET data is the locations and lengths of the ChIP-PET fragments. The tuple <*s*, *l*> represents an *l*-bp long PET fragment mapped into location *s*. Two PET fragments $<s_1,l_1>$ and $<s_2,l_2>$, where $s_1 \le s_2$, are said to be overlapping if $s_1 + l_1 \ge s_2$. A ChIP-PET cluster is defined as the largest set of cascading overlapping PET fragments. Figure 12 shows an abstraction of ChIP-PET library, after the clustering stage is performed. Further assuming that binding site can be located anywhere in a bound fragment, the precise location of the binding site is expected to be approximately located at the center of such accumulation. It has been validated that the clustering of overlapping PET fragments is an effective readout to distinguish true signals of protein-DNA interactions from background noises (Wei *et al.*, 2006; Loh *et al.*, 2006).

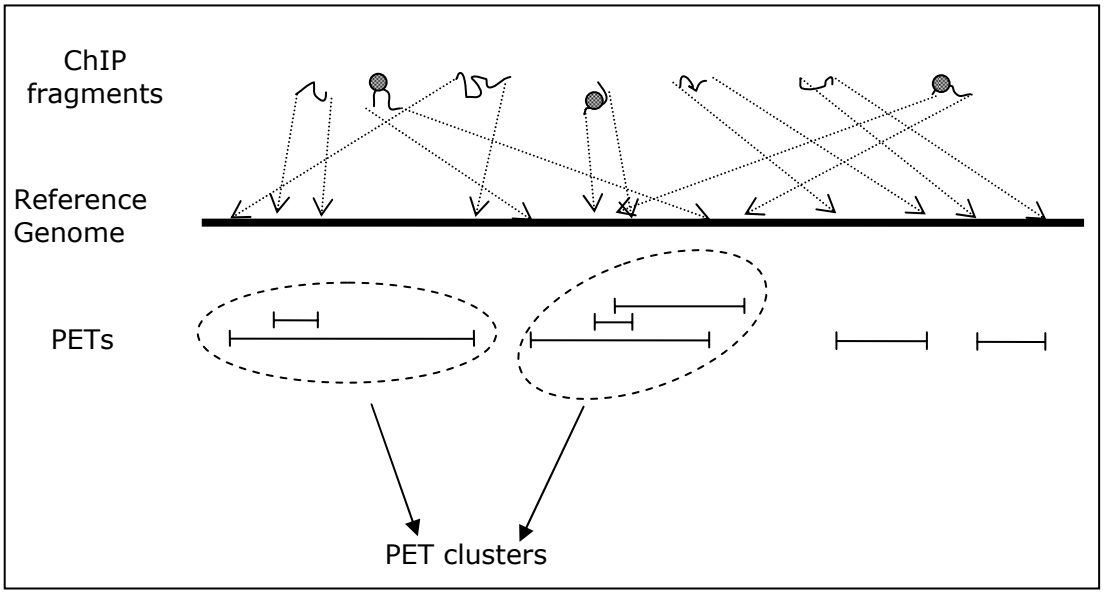


Figure 12. Relationship between ChIP fragments, PETs, and ChIP-PET clusters. ChIP fragments might be TF-bound (shaded circles) or simply noise. Mapped ChIP fragments are called PETs. Overlapping PETs are grouped into ChIP-PET clusters.

4.4.3 Fragment Accumulation around Non-Bound Sites

Cluster size as a predictive variable

Presence of PET clusters is clearly an initial indication of genomic loci enriched for ChIP PET fragments, most likely due to ChIP pull down of TF-bound fragments. Ideally clusters are generated only by real enrichment due to TF-DNA interactions, i.e. active binding regions. The more PETs that a cluster has, the more probable the TF bounds to the region. There is, however, a possibility that some of the clusters occurred simply by chance alone, resulted from clustering of noisy PETs. We can set a minimum cut-off criterion, say *h*, and classify clusters with at least *h* PETs (i.e. PET*h*+ clusters) to be the highly probable clusters with TF binding. To appropriately determine this threshold, a Monte Carlo approach could be employed. We have shown that this approach was considerably effective (Wei *et al.*, 2006).

More analytically, if we assume that the noisy PETs are randomly and uniformly distributed along the genome, then the distance, *d*, between any two consecutive random PETs is expected to follow the exponential distribution with rate $\lambda = T/G$, where T is the total number of PETs and G is the genome length. By definition, two PETs can be clustered if they overlap by at least one base pair. Suppose *k* is the expected length of a PET. The probability of two PETs overlapping (i.e. the distance between them is less than or equal the (expected) PET length) by chance alone is $\Pr_{\exp}(X \le k;\lambda)$ where $\Pr_{\exp}$ is the cumulative exponential distribution function whose rate is λ. The exact formula for the cumulative function is: $\Pr_{\exp}(X \le k;\lambda) = 1 - e^{-\lambda k}$. Note that two overlapping PETs can be found in a PET2 cluster and beyond. Thus, the probability $\Pr_{\exp}(X \le k;\lambda)$ is the probability of a PET2+ cluster to happen simply by chance alone. Obviously, successive overlaps of

PETs form a higher PET*n* cluster. Hence, more generally, the probability of the occurrence of a PET*n*+ cluster by random is:

$$\Pr_{PET}(Y \geq n; k, \lambda) \approx \left(\Pr_{\exp}(X \leq k; \lambda)\right)^{(n-1)} = \left(1 - e^{-\lambda k}\right)^{(n-1)} \qquad \textbf{(4.4.1)}$$

In place of the Monte Carlo simulations, one can readily compute the p-value of random PET*n*+ clusters using the above equation to determine the appropriate threshold for a given ChIP PET library.

Using maximum support to identify binding regions

While number of PETs forming a cluster indeed provides useful information for assessing whether the cluster is more likely to be true signal, clusters with seemingly good number of PETs can still be generated by random noise. It is not uncommon to find big clusters whose overlapping regions are not well concentrated, going against the intuition that real binding sites should produce crisp and well defined core, an indication that they were formed simply by chance.

Figure 13 shows a snapshot of two clusters from real libraries as visualized by the T2G browser (a GIS in-house visualization suite based on the UCSC genome browser), contrasting a typical good cluster (left part of the figure), having well defined core, to a configuration with scattered overlap region (right part of the figure) most likely formed by random PETs. Note that both clusters are PET5 clusters, but the left cluster contains a clear and strong core region of 5 overlapping PETs, while the right cluster has four contiguous sub-regions with two PET overlap each. We call a PET cluster as a moPET*n* (*maximum overlap* PET *n*) cluster if all of its sub-region is

supported by at most n PETs. Similar to the previous definition, moPETn+ clusters represent the set of moPETm clusters where $m \geq n$. The left PET5 cluster in Fig. 13 is of moPET5, while the right PET5 cluster is of moPET2.

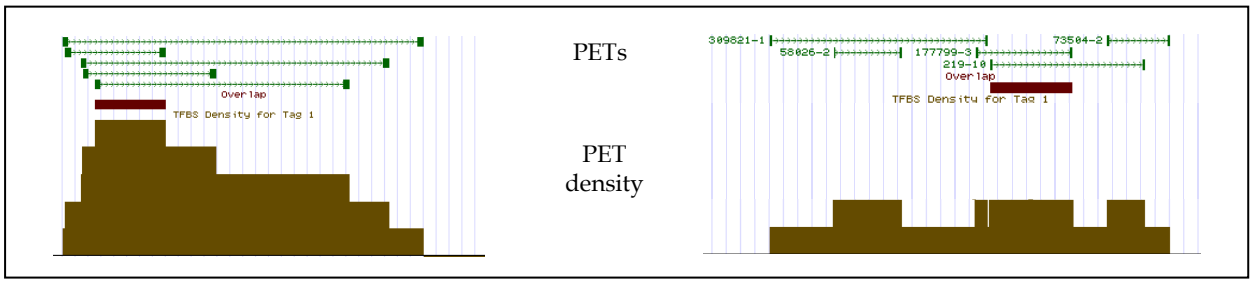


Figure 13. Contrasting high fidelity cluster and noisy cluster. Shown here are two clusters from a real library, visualized using the T2G browser, a GIS in-house visualization tool based on the UCSC genome browser. Good clusters are generally well-defined (left cluster), containing a strong overlapping region. Dispersed ChIP PET segments (right cluster) hint the possibility of cluster formation purely at random and by chance alone.

The probability of a moPETn to be initiated by an arbitrary PET <s,l> can be estimated by the probability of observing additional *(n-1)* PET starting sites at most *l*-bp away from *s*. Under the assumption of random uniform distribution of PET start sites, this probability follows that of Poisson distribution for observing *(n-1)* events whose rate is λ within the interval k (=expected PET length). More formally, the probability of an arbitrary PET to initiate a moPETn cluster:

$$\Pr_{moPET}(Y = n; k, \lambda) \approx \left(\Pr_{poisson}(X = (n-1); \lambda k)\right) = \frac{e^{-\lambda k}(\lambda k)^{(n-1)}}{(n-1)!} \quad \textbf{(4.4.2)}$$

Using $\Pr_{moPET}(Y = n; k, \lambda)$ and given the acceptable p-value level, we can determine the appropriate cut-off of moPETn for identifying true TF-binding regions.

4.4.4 Adaptive Approach for Biased Genomes

The estimation of rate λ, i.e. the expected number of PETs per nucleotide, plays a critical role in Eqs. 4.4.1 and 4.4.2. This rate signifies the expected noise level of the dataset. So far, we have only talked about a single global rate λ, reflecting the assumption that the noisy PETs are randomly uniformly distributed across the genome. Although the genome-wide uniform noise assumption maybe acceptable in general, in cases where *apriori* knowledge about the presence of biasing factors in the genome is available, it should be exploited accordingly. The prevalence of significant genome rearrangements in tumor cells and cancer cell lines, for example, calls for a fine tuning of the generic method described earlier. For instance, the MCF-7 cell line, which has been a platform for Estrogen related studies, contains at least 21 regions of high-level copy number alterations (Shadeo and Lam, 2006). Such biases affect the ChIP-PET data. Regions with significant deletions will contain less than expected PETs and their true binding loci will be much weaker. Amplified regions will have higher PET counts than the overall genome, making their purely random clusters bear stronger signal than those of normal regions. Using single global λ would result in higher false positive rates in amplified regions and higher false negative rates in deleted regions.

We devised a two-phase adaptive approach that takes into account of local biases (see Fig. 14) in predicting the most probable source (true binding vs. noise) of each PET cluster. Given a cluster *c*, the first phase considers the local window of some predefined size *L* centered on the cluster *c*, and, estimates the total number of noise PETs. The second phase computes the local λ and calculates a local moPET (or

PET) cut-off T_{cut}. Clusters *c* is considered to be a binding region if its moPET (or PET) count is greater than T_{cut}.

GoodCluster(*c,p,L*) :

1. Let *D* be the set of clusters that are located at most $\frac{L}{2}$ basepairs away (upstream or downstream) of *c*
2. Let $G = \{\}$
3. For each $d \in D$, if $mo(d) \le mo(c)$ then $G = G \cup \{pet(d)\}$
4. Let *g* be the geometric mean of G, i.e. $g = geomean(G)$
5. Estimate the total local noisy PET $S = \sum_{d \in D} \min(size(d), g)$

Noise Estimation

6. Set $\lambda = \frac{S}{L}$
7. Determine T_{cut} such that:
   $T_{cut} = \min(\{n \mid \Pr_{moPET}(Y \ge n; \lambda k) \le p\})$
8. If $T_{cut} < mo(c)$ then return **TRUE** else return **FALSE**

Threshold Determination

Figure 14. Pseudocode of the adaptive thresholding algorithm. GoodCluster() takes as input the cluster *c*, the *p*-value cutoff *p*, and window size *L*. It will return **TRUE** if cluster *c* meets the significance requirement. The algorithm consists of two main steps: (i) local noise estimation and (ii) local threshold determination. Function geomean(X) computes the geometric mean of set *X*. Functions mo(d) and pet(d) return the moPET and PET count of cluster *d*. In line 7, $\Pr_{moPET}()$ can be replaced with $\Pr_{PET}()$. Estimation of T_{cut} can also be done through Monte Carlo simulations.

The noise estimation step (first phase) counts the number of potentially noisy PETs within the window. This needs to be performed carefully, since there is no actual labeling of which clusters within the current window are real. Overestimation of noise would increase false negatives, while underestimation would add false positives. We adhere to two heuristics, namely: (i) the current cluster should not be assumed as real and (ii) other clusters within the windows that seem to be real clusters should, as much as possible, not be counted as noise. The first rule is stemming from the fact that most of the clusters (especially PET1 clusters) are noise. Observations that binding sites are sometimes located proximal to each other motivated the second rule. The choice of window size *L* also influences the noise estimation accuracy. In

our analysis we set L to be at least twice of the expected distance between two PETs (i.e. λ^{-1}).

In our implementation, the noise estimation starts by identifying the probable noisy clusters. Using the moPET count and based on the assumption that the current cluster c is noisy, clusters with higher moPET counts than the current cluster c are contextually considered non-noise (see line 3 in Fig. 14). Next, we want to know what the expected typical PET count is for a noisy cluster. The expected PET count g of a noisy PET cluster is calculated by taking the geometric mean of the PET counts of the noisy clusters identified earlier. Geometric mean was employed since the PET counts can be considered as the rate of noise per cluster (McAlister, 1879; Fleming and Wallace, 1986). The final sum of noisy PETs, S, is calculated by adding the noisy PET counts of all the clusters within the current window. If a cluster's PET count is less than or equal to g, the entire cluster is considered noisy and its PET count added to the final sum. If a cluster's PET count is greater than g, then it should only contribute an estimated noisy count (i.e. g) towards the final sum. This is done to avoid noise overestimation in windows with multiple real clusters.

The second step is quite straightforward through the application of the Eqs. 4.4.1 or 4.4.2 (using the local rate λ $(= S / L)$ and considering the window length L) or performing sufficient iterations of Monte Carlo simulations, using S as the total number of fragment within the L-bp region.

4.4.5 Evaluation

Dataset

In our evaluation, we made use of both artificial and real datasets. The artificial datasets were generated to assess the preciseness of our analytical formulations (Eqs. 4.4.1 and 4.4.2) in modeling the chance accumulation of ChIP fragments around non-bound regions. Three real datasets were: the p53 ChIP-PET (Wei *et al.*, 2006), the Oct4 ChIP-PET (Loh *et al.*, 2006}, and the Estrogen Receptor (ER) ChIP-PET (Lin *et al.*, 2007}. For each dataset, a set of PET-clusters most likely to represent TF-binding regions were selected based on our proposed algorithms. The selected clusters were then evaluated indirectly by enrichment of putative relevant binding motifs and (whenever available) directly using ChIP qPCR validation data.

The p53 library was the first and the smallest dataset, which contains 65,714 PETs (average length 625bp) and was constructed using the human HCT116 cancer cell lines. The ER ChIP PET library comprised 136,152 PETs, whose average length is 672bp, was assayed on human MCF-7 breast cancer cell lines. The largest library among the three, the Oct4 ChIP PET, was based on mouse E14 cell lines and consists of 366,639 PETs of 627bp on average. The non-gapped genome lengths for human and mouse are estimated at ~2.8Gbp (UCSC hg17) and ~2.5Gbp (UCSC mm5) respectively.

Experimental setup

Evaluation of the analytical models was done using artificial libraries. To generate an artificial random PET library, we preformed a Monte Carlo simulation while taking into account the overall genome length (G), the total number of PETs

(T), and the desired PETs' lengths (minimum and maximum lengths; l_{min} to l_{max}). In each Monte Carlo simulation, T points were randomly picked along the G-bp genome, mimicking the generation of a PET library containing completely random fragments. For each picked point, a random length was sampled from a uniform distribution within the given minimum and maximum bounds. Overlapping PETs are clustered, similar to what would have been done for real PET libraries. Statistics of PETn+ and moPETn clusters were collected and averaged over a sufficient number of Monte Carlo iterations. These are then compared to numerical results from application of Eqs. 4.4.1 and 4.4.2 on the same parameters. In our study we generally ran 100,000 Monte Carlo iterations. The five setups that we tested are listed in Table 9. For the analysis of real libraries, we used a cut-off of p-value < 1e-3 in selecting good clusters. We tested cluster selection based on both PET and moPET counts and using global threshold as well as adaptive threshold.

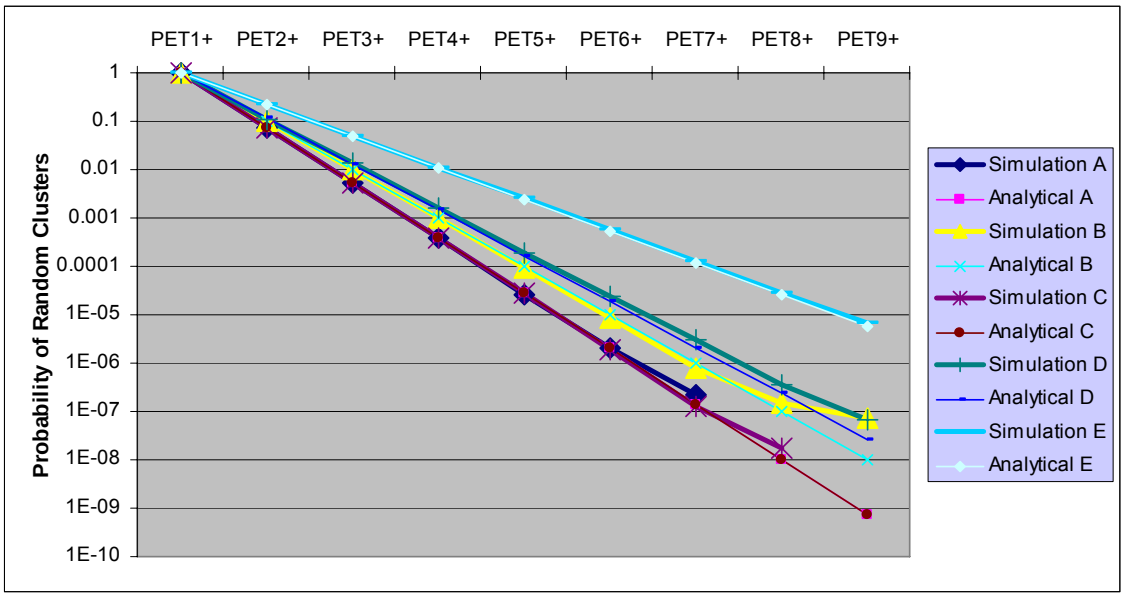
Simulation Set	A	B	C	D	E
Genome Length	2 Mbp	3 Mbp	20 Mbp	10 Mbp	10 Mbp
No. of PETs	300	300	3000	2000	5000
Min. PET length	500 bp	700 bp	500 bp	200 bp	300 bp
Max. PET length	500 bp	700 bp	500 bp	1000 bp	700 bp

Table 9. Simulation setups for artificial ChIP-PET libraries. FiveMonte Carlo simulation sets run to assess the analytical model of random PETn+ and moPETn clusters formations.

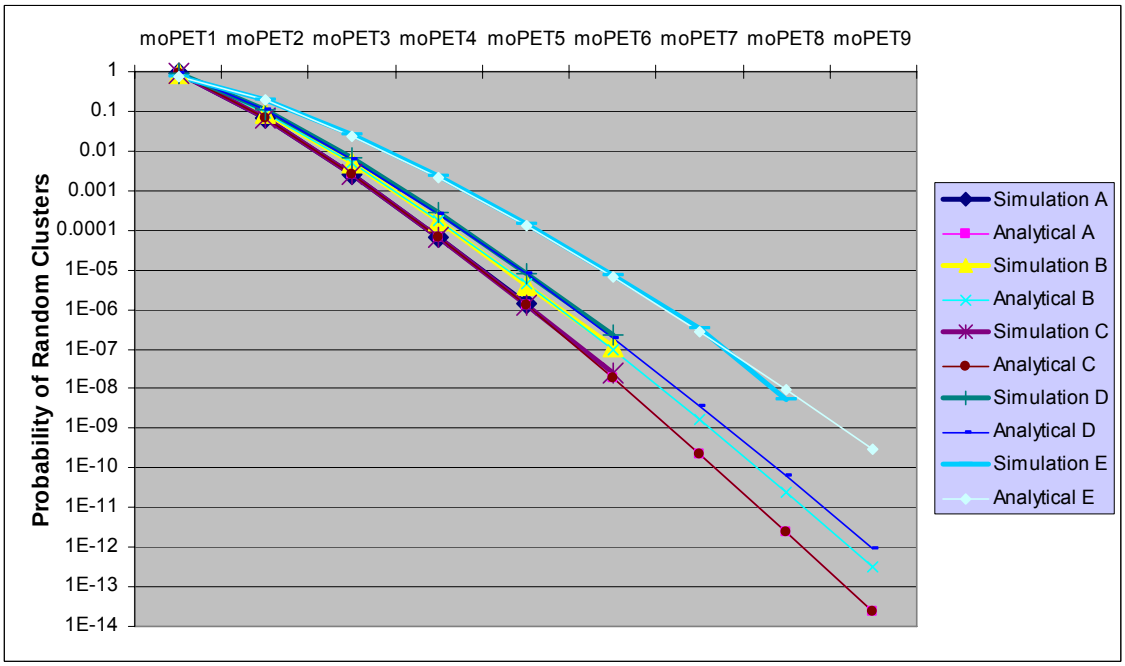
Results

Using the artificial random data were generated through a series of Monte Carlo simulations as described above, we compared the analytical estimations of PETn+ / moPETn clusters distributions to the empirical ones. The collected statistics were used to construct empirical distributions which were then compared with the proposed analytical framework. In each simulation set of 100,000 Monte Carlo runs, we calculated the probability (or the fraction) of PETn+ and moPETn clusters observed in the simulated library. Figure 15a contrasts the empirical probability of PETn+

occurrence (thick lines) against the analytical estimations (thin lines). A similar plot for moPET*n* analysis is shown in Fig. 15b. The analytical curves track the empirical curves very well, reconfirming the validity of the analytical distributions.



(a)



(b)

Figure 15. Comparison of analytical computation and empirical simulation. Probability of (**a**) a random PET*n*+ cluster or (**b**) a random moPET*n* cluster being generated simply by chance alone across different library setups, computed empirically through Monte Carlo simulations (thick lines) and analytically (thin lines) based on $\Pr_{PET}(X)$ of Eq. 4.4.1 or $\Pr_{moPET}(X)$ of Eq. 4.4.2. The analytical curves match the empirical curves well.

Based on the moPET framework and the *p*-value cutoff of < 1e-3, the selected (good) clusters for p53 is moPET3+, for ER is moPET3+ and for Oct4 is moPET4+. With the similar cut-off of *p*-value < 1e-3 and employing the PET size criteria, the selected set of clusters for p53 is PET3+, for ER is PET4+, and for Oct4 is PET4+.

Table 10 gives the validations of each PET cluster group in each library, based on motifs prevalence and additional ChIP qPCR assays on samples of the PET cluster group. We can observe sharp motif enrichment at the selected cut-offs in all libraries, i.e. moPET3+, moPET4+, moPET3+ for p53, Oct4 and ER respectively, especially when compared to the PET2/moPET2 group which is expected to contain many noisy (i.e. random) clusters. Note, however, that PET2/moPET2 clusters are not all noise. They still contain TF-bound regions. Completely random genomic regions have lower motif occurrence rate.

Table 10 also shows how many clusters were further subjected to ChIP-qPCR validations and their validation success rate. The p53 library undoubtedly had the highest validation rate with 100% of the tested sites showing enrichment of p53 binding. The high ChIP-qPCR success rate (>95%) for the selected Oct4 moPET4+ clusters also increased our confidence of the validity of the cluster selection approach.

Cluster Group	Total clusters	% with motifs	ChIP-qPCR tested	% success
PET2	1453	15.97%	0	N/A
PET3	161	59.63%	0	N/A
PET4	66	80.30%	5	100.00%
PET5	38	65.79%	4	100.00%
PET6	29	89.66%	8	100.00%
PET7	13	84.62%	5	100.00%
PET8+	29	82.76%	18	100.00%
moPET2	1489	16.25%	0	N/A
moPET3	140	67.14%	1	100.00%
moPET4	69	81.16%	6	100.00%
moPET5	30	70.00%	4	100.00%
moPET6	26	88.46%	9	100.00%
moPET7+	35	88.57%	20	100.00%

(**A**) p53 ChIP-PET clusters

Cluster Group	Total clusters	% with motifs	ChIP-qPCR tested	% success
PET2	29453	16.74%	10	10.00%
PET3	5556	24.62%	31	9.68%
PET4	1540	34.35%	17	88.24%
PET5	550	42.36%	21	90.48%
PET6	223	52.47%	11	100.00%
PET7	102	49.02%	5	100.00%
PET8+	201	45.77%	20	95.00%
moPET2	32739	17.57%	10	10.00%
moPET3	3734	27.64%	34	8.82%
moPET4	724	41.57%	40	95.00%
moPET5	189	54.50%	14	100.00%
moPET6	93	70.97%	8	100.00%
moPET7+	146	43.15%	9	100.00%

(**B**) Oct4 ChIP-PET clusters

Cluster Group	Total clusters	% with motifs
PET2	5704	40.06%
PET3	930	57.31%
PET4	341	65.69%
PET5	181	70.72%
PET6	124	76.61%
PET7	78	78.21%
PET8+	216	83.33%
moPET2	6100	41.02%
moPET3	756	61.90%
moPET4	281	64.77%
moPET5	134	76.12%
moPET6	95	78.95%
moPET7+	208	85.10%

(**C**) ER ChIP-PET clusters

Table 10. Validation rate and motif enrichments of clusters selected by global thresholding. Evaluation of the various groups of ChIP-PET clusters for the (A) p53, (B) Oct4, and (C) ER ChIP PET libraries. Note that the 'good' PET clusters for the p53, Oct4, and ER libraries are PET3+, PET4+, and PET4+ respectively, or moPET3+, moPET4+, and moPET3+ respectively. The lower PET/moPET groups (e.g. PET2 or moPET2) are presented as a comparison. The top half of each table shows the ChIP PET clusters' enrichment for each corresponding binding site motif, which serves as a good proxy of how likely the clusters are to be true clusters.Whenever possible, results from ChIP qPCR validations on random subsets of ChIP PET clusters within each group are presented in the bottom half of the tables.

Prior to running the ChIP-qPCR validation for the ER library, we noticed unusual concentrations of PETs in some regions. These regions correlated well with the regions previously reported to be amplified in the underlying MCF-7 cell lines (Shadeo and Lam, 2006), for example: some parts of chromosomes 17 and 20. Under the global moPET analysis, the good clusters of ER ChIP PET library are the moPET3+ clusters, totaling 1,474 clusters. The top two good-clusters-containing chromosomes are chromosomes 20 and 17, with about 10% and 9.5% of the selected clusters. Note that both chromosomes 20 and 17 were reported to be highly amplified in MCF-7 (Shadeo and Lam, 2006). This prompted us to employ the adaptive moPET thresholding algorithm to "normalize" the amplified regions. We also applied the adaptive approach on the other two datasets, to see its effect on other libraries from relatively normal cell lines (i.e. the p53 and Oct4 libraries). The result is summarized in Table 11.

Cluster Group	Total clusters	% with motifs	ChIP-qPCR tested	% success
PET2	0	N/A	N/A	N/A
PET3	125	68.80%	0	N/A
PET4	66	80.30%	5	100.00%
PET5	38	65.79%	4	100.00%
PET6	29	89.66%	8	100.00%
PET7	13	84.62%	5	100.00%
PET8+	29	82.76%	18	100.00%
moPET2	0	N/A	N/A	N/A
moPET3	140	67.14%	1	100.00%
moPET4	69	81.16%	6	100.00%
moPET5	30	70.00%	4	100.00%
moPET6	26	88.46%	9	100.00%
moPET7+	35	88.57%	20	100.00%

(**A**) p53 ChIP-PET clusters

Cluster Group	Total clusters	% with motifs	ChIP-qPCR tested	% success
PET2	0	N/A	N/A	N/A
PET3	404	34.16%	6	16.70%
PET4	510	41.18%	16	93.80%
PET5	305	47.54%	19	100.00%
PET6	167	58.08%	11	100.00%
PET7	88	52.27%	5	100.00%
PET8+	195	45.64%	20	95.00%
moPET2	0	N/A	N/A	N/A
moPET3	524	36.83%	6	16.70%
moPET4	717	41.84%	40	95.00%
moPET5	189	54.50%	14	100.00%
moPET6	93	70.97%	8	100.00%
moPET7+	146	43.15%	9	100.00%

(**B**) Oct4 ChIP-PET clusters

Cluster Group	Total clusters	% with motifs	ChIP-qPCR tested	% success
PET2	0	N/A	N/A	N/A
PET3	453	64.24%	18	72.20%
PET4	253	68.77%	8	75.00%
PET5	144	72.92%	5	100.00%
PET6	107	78.50%	4	100.00%
PET7	69	84.06%	1	100.00%
PET8+	208	82.69%	1	100.00%
moPET2	0	N/A	N/A	N/A
moPET3	552	65.58%	20	70.00%
moPET4	245	68.57%	6	83.30%
moPET5	134	76.12%	7	100.00%
moPET6	95	78.95%	2	100.00%
moPET7+	208	85.10%	2	100.00%

(**C**) ER ChIP-PET clusters

Table 11. Validation rate and motif enrichments of clusters selected by adaptive thresholding. Validation results on the (A) p53, (B) Oct4, and (C) ER ChIP-PET libraries on clusters selected by adaptive thresholding, where the top half of each table shows the motif enrichment and the bottom half lists the ChIP-qPCR outcomes. All of the breakdowns shown are based on clusters selected through the adaptive algorithm. The ChIP qPCR for p53 and Oct4 presented here is a subset of what was reported earlier in Table 10. ChIP qPCR for ER was done by taking random clusters from the clusters selected by the adaptive approach.

Note that the application of adaptive thresholding might both exclude clusters selected under the global thresholding and re-include clusters which would otherwise be excluded because they were below the global threshold. Application of global and adaptive moPET thresholding on the p53 library produced the same results (compare Table 10a and 11a). Interestingly, application of adaptive thresholding on the Oct4 library re-included some of the moPET3 clusters, with a higher proportion of motif-containing clusters compared to the entire moPET3 clusters. Only a tiny fraction of the moPET4 was rejected, without any significant impact on the motif enrichment. The ChIP qPCR success rates for the adaptive-selected clusters were higher than before. For the ER ChIP PET library, a sizeable portion of the moPET3+ was no longer considered to be TF-bound. The overall increase in the proportion of motif-containing clusters indicated that the selected clusters were likely to be real. Additional ChIP-qPCR assays on random samples of the selected clusters confirmed that further. The highly amplified chromosomes 17 and 20 no longer had the most number of selected clusters. Chromosomes 1 and 2 contained the selected clusters the most, which was expected since they are the two longest chromosomes (see (Lin *et al.*, 2007)).

Chapter 5

Conclusion

5.1 Summary

Our research was motivated by the recent phenomenal growth and growing complexity of biological data. In particular we were interested in developing computational approaches to help understand the regulatory mechanisms of genes and identify (from relevant datasets) the regulatory targets and genomic regulatory signals. We started off by constructing a paradigm that models and encompasses complex system containing indirect relationship between the observable input and the measurable outputs. We then focused on expression data generated using mRNA microarray and genomic data of TF-DNA interactions obtained from the sequencing-based ChIP-PET protocol. To give more details:

- In Chapter 2, we construct a paradigm that models a complex system, where the relationship between the input and the output might be indirect and is confounded with presence of background noise. For our research, we decided to decouple the analysis of the input and output. The subsequent sections describe in more depth the set of problems that we were investigating.

- Chapter 3 focuses on Microarray data as the primary source data for the output stream in the gene regulation system. We identified and researched on two issues: (i) determination of minimal gene signature cassette, and (ii)

identifying primary response genes from time-course microarray data. Our results showed that AdaBoost can be adequately modified to tackle the first task. An important modification was imposing an additional restriction that each feature could only be used once in building the classifier. This restriction is not typically enforced in AdaBoost. We found that this restriction was critical due to the high-dimensionality of microarray data and actually rendered the AdaBoost to identify the minimal gene set as originally desired. For the second issue, we develop the Friendly Neighbour approach to exploit the intuition that primary response genes are responsible for (or at least very influential to) the expression regulation of other genes. Rather than ranking based on the genes ability to separate treatment labels, genes are appraised based on the number of other genes that share its expression pattern. Our results showed that this method well outperformed other non-supervised methods and was quite close to the performance of supervised methods.

- Chapter 4 opens with a description of the ChIP-PET protocol. Our interest in this subject was fivefold: (i) to provide a quick assessment criteria for library sequencing adequacy, (ii) to model ChIP fragment size more accurately, (iii) to model the distribution of ChIP fragments detected for inferring the overall signal strength, (iv) to model fragment accumulation at true TF-DNA interaction sites, and (v) to develop an algorithm that automatically normalized the effect of aberrant genome. We developed the Multiplicity Index for a quick assessment of sequencing saturation. The Multiplicity Index was shown to correlate significantly to the more rigorous saturation analysis. For ChIP fragment size, we devised the Normal*Exponential model that

incorporates the possible presence of unbreakable region. This model outperformed the previously proposed Gamma distribution. We proposed a model of fragment distribution that factored in the proportion of bound fragments and the bound regions. Fitting the model to the data allowed us to estimate the property of the library. The estimated relative signal strength agreed with the experimental ChIP-qPCR readings. An analytical model was explored for calculating the probability of fragment accumulation around non-bound sites. It was further used to distinguish fragment enrichment of bound regions from random enrichments. Expanding the analysis further, we developed a sliding-window based algorithm that estimates the local noise level and then applying local threshold for selecting binding regions. Our results demonstrated that this approach improves the quality of the selected regions, both in aberrant genome and in (expectedly) normal genome.

5.2 Future Directions

Several interesting research questions emerged during the course of our research. Among them are:

- **Optimizing the similarity measure for FN**. The similarity measure in the FN has an implicit assumption to the relationship of the genes. It is conceivable then to actually construct similarity measures that reflect or favor certain properties (e.g. gene activation rather than repression) and use the FN approach to identify “primary regulators” in an arbitrary dataset

- **Modeling the binding affinity distribution**. In our formulation of a model for ChIP fragment distribution, we have made the provision that the binding regions could yield different binding affinities (and thus enrichment factor). It has not, however, been properly and thoroughly assessed. A comprehensive evaluation would necessitate additional experimental wet-lab data, though.

- **Accounting for Fragment Length Distribution**. Our analytical formulae to compute probability of random fragment enrichment assumes a fixed fragment length. Monte Carlo simulations procedure has the benefit of faithfully incorporate the empirical fragment distribution, when estimating the *p*-value. We have also shown that Normal*Exponential distribution seemed to model the fragment length well. Needless to say, an open task is to incorporate the fragment length distribution into the analytical formulae.

References

Alon, U. An Introduction to Systems Biology: Design Principles of Biological Circuits. CRC Press, 2006.

Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., & Levine, A. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon cancer tissues probed by oligonucleotide arrays. Cell Biology, 96, 6745–6750, 1999.

Ambroise, C., & McLachlan, G. J. Selection bias in gene extraction on the basis of microarray gene expression data. Proc. Natl. Acad. Sci. USA, 99:10, 6562–6566, 2002.

Barrett, J.C. and Kawasaki, E.S. Microarrays: the use of Oligonucleotides and cDNA for the Analysis of Gene Expression. Drug Discovery, 8: 134-141, 2003.

Bates, D. M. and Watts, D. G. Nonlinear Regression and Its Applications. New York: Wiley, 1988.

Bhinge, A.A., Kim, J., Euskirchen, G.M., Snyder, M., Iyer, V.R. Mapping the chromosomal targets of STAT1 by Sequence Tag Analysis of Genomic Enrichment (STAGE). Genome Res. 17(6):910-6, 2007.

Bird, A. Perceptions of Epigenetics. Nature 447: 396-398, 2007.

Breiman, L. Arcing classifiers. The Annals of Statistics, 1998.

Chiu, K.P., Wong, C.H., Chen, Q., Ariyaratne, P., Ooi, H.S., Wei, C.L., Sung, W.K., and Ruan, Y. PET-Tool: a software suite for comprehensive processing and managing of Paired-End diTag (PET) sequence data. BMC Bioinformatics. 7:390, 2006.

Crick, F. Central Dogma of Molecular Biology. Nature, 227: 561-563, 1970.

Dubhashi, D., & Ranjan, D. Balls and bins: A study in negative dependence. Random Structures and Algorithms, 13:2, 99–124, 1998.

Duda, R. O., & Hart, P. E. Pattern Classification and Scene Analysis. Wiley, 1973.

Dudoit, S., Fridlyand, J., and Speed, T. P. Comparison of discrimination methods for the classification of tumors using gene expression data. Journal of the American Statistical Association, 97:457, 77–87, 2002.

Eddy, S.R. Noncoding RNA Genes. Current Opinion in Genetics & Development, 9(6):695-699, 1999.

Eddy, S.R. Non-coding RNA Genes and the Modern RNA World. Nature Reviews Genetics, 2(12):919-929, 2001.

Freund,Y. Boosting a weak learning algorithm by majority. Information and Computation, 121:2, 256–285, 1995.

Freund, Y., & Schapire, R. Experiments with a new boosting algorithm. In Proceedings of the Thirteenth International Conference on Machine Learning, 1996.

Fu, M., Sun, T., Bookout, A. L., Downes, M., Yu, R. T., Evans, R. M., and Mangelsdorf, D. J. A Nuclear Receptor Atlas: 3T3-L1 Adipogenesis. Molecular Endocrinology 19 (10): 2437-2450, 2005.

Gaston, K. and Jayaraman, P.-S. Transcriptional Repression in Eukaryotes: Repressors and Repression Mechanisms. Cellular and Molecular Life Sciences, 60(4):721-741, 2003.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., & Lander, E. S. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science, 286, 531–537, 1999.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. Gene selection for cancer classification using support vector machines. Machine Learning, 46:1–3, 389–422, 2002.

Hamza, M.S, Pott, S., Vega, V.B, Thomsen, J.S, Kandhadayar, G.S, Ng, P.W.N, Chiu, K.P, Pettersson, S., Wei, C.L., Ruan, Y., and Liu, E.T. De-novo identification of PPARγ/RXR binding sites and direct targets during Adipogenesis. (Manuscript under review).

Haussler, D. Decision theoretic generalizations of the PAC model for neural net and other learning applications. Information and Computation, 100:1, 78–150, 1992.

Haussler, D., Littlestone, N., & Warmuth, M. K. Predicting {0, 1}-functions on randomly drawn points. Information and Computation, 115:2, 129–161, 1994.

Hill, A. V. The possible effects of the aggregation of the molecules of haemoglobin on its oxygen dissociation curve. J Physiol (Lond) 40: 4-7, 1910.

Horak CE, Mahajan MC, Luscombe NM, Gerstein M, Weissman SM, and Snyder M. GATA-1 binding sites mapped in the beta-globin locus by using mammalian chIp-chip analysis. Proc Natl Acad Sci USA. 99(5):2924-9, 2002.

Impey, S., McCorkle, S.R., Cha-Molstad, H., Dwyer, J.M., Yochum, G.S., Boss, J.M., McWeeney, S., Dunn, J.J., Mandel, G., Goodman, R.H. Defining the CREB regulon: a genome-wide analysis of transcription factor regulatory regions. Cell. 119(7):1041-54, 2004.

Iyer, V.R., Horak, C.E., Scafe, C.S., Botstein, D., Snyder, M., and Brown, P.O. Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF. Nature 409(6819): 533-8, 2001.

Joachims, T. Making Large-scale Support Vector Machines Learning Practical. Advances in Kernel Methods: Support Vector Machines, pp 169-184, 1998.

Johnson, D.S., Mortazavi, A., Myers, R.M., and Wold, B. Genome-wide mapping of in vivo protein-DNA interactions. Science. 316(5830):1497-502, 2007.

Karuturi, R. K. M, and Vega, V. B. Friendly Neighbors Method for Unsupervised Determination of Gene Significance in Time-course Microarray Data. In Proceedings of the 4th IEEE Symposium on Bioinformatics and Bioengineering, 2004.

Kasturi, J., Acharya, R., and Ramanathan, M. An Information Theoretical Approach for Analyzing Temporal Patterns of Gene Expression. Bioinformatics, 19, 449-458, 2003.

Kearns, M., Mansour, Y., Ng, A. Y., & Ron, D. An experimental and theoretical comparison of model selection methods. Machine Learning, 27, 7–50, 1997.

Kivinen, J., and Warmuth, M. Boosting as entropy projection. In Proc. COLT'99, 1999.

Kuriakose, M.A., Chen, W.T., He, Z.M., Sikora, A.G., Zhang, P., Zhang, Z.Y., Qiu, W.L., Hsu, D.F., McMunn-Coffran, C., Brown, S.M., Elango, E.M., Delacure, M.D., and Chen, F.A. Selection and validation of differentially expressed genes in head and neck cancer. Cell Mol Life Sci. 61(11):1372-83, 2004.

Kuznetsov, V.A., Knott, G.D., and Bonner, R.F. General statistics of stochastic process of gene expression in eukaryotic cells. Genetics 161, 1321-1322, 2002.

Lamb, K.A. and Rizzino, A.. Effects of Differentiation on the Transcriptional Regulation of the FGF-4 Gene: Critical Roles Played by a Distal Enhancer. Molecular Reproduction and Development, 51:218-224, 1998.

Li, Y., Long, P. M., & Srinivasan, A. Improved bounds on the sample complexity of learning. Journal of Computer and System Sciences, 62:3, 516–527, 2001.

Leung, H.C.M and Chin, F.Y.L.. Generalized Planted (*l*,*d*)-Motif Problem with Negative Set. WABI 2005, LNBI 3692, pp. 264–275, 2005.

Lim, C.A., Yao, F., Wong, J.J., George, J., Xu, H., Chiu, K.P., Sung, W.K., Lipovich, L., Vega, V.B., Chen, J., Shahab, A., Zhao, X.D., Hibberd, M., Wei, C.L., Lim, B., Ng, H.H., Ruan, Y., Chin, K.C. Genome-wide mapping of RELA(p65) binding identifies E2F1 as a transcriptional activator recruited by NF-kappaB upon TLR4 activation. Mol Cell. 27(4):622-35, 2007.

Lin, C.Y., Ström, A., Vega, V.B. (co-first author), Kong, S.L., Yeo, A.L., Thomsen, J.S., Chan, W.C., Doray B., Bangarusamy, D.K., Ramasamy, A., Vergara, L.A., Tang, S., Chong, A., Bajic, V.B., Miller, L.D., Gustafsson, J.A., Liu, E.T. Discovery of estrogen receptor α target genes and response elements in breast tumor cells. Genome Biology, 5(9):R66, 2004.

Lin, C.Y., Vega, V.B., Thomsen, J.S., Zhang, T., Kong, S.L., Xie, M., Chiu, K.P., Lipovich, L., Barnett, D.H., Stossi, F., Yeo, A., George, J., Kuznetsov, V.A., Lee, Y.K., Charn, T.H., Palanisamy, N., Miller, L.D., Cheung, E., Katzenellenbogen, B.S., Ruan, Y., Bourque, G., Wei, C.L., and Liu, E.T. Whole-genome cartography of estrogen receptor alpha binding sites. PLoS Genet. 3(6):e87, 2007.

Loh, Y.H., Wu, Q., Chew, J.L., Vega, V.B., Zhang, W., Chen, X., Bourque, G., George, J., Leong, B., Liu, J., Wong, K.Y., Sung, K.W., Lee, C.W., Zhao, X.D., Chiu, K.P., Lipovich, L., Kuznetsov, V.A., Robson, P., Stanton, L.W., Wei, C.L., Ruan, Y., Lim, B., and Ng, H.H. The Oct4 and Nanog transcription network regulates pluripotency in mouse embryonic stem cells. Nat Genet., 38(4):431-40, 2006.

Long, P. M. and Vega, V. B. Boosting and microarray data. Machine Learning, 52(1):31-44, 2003.

Miller, L. D., Long, P. M.,Wong, L., Mukherjee, S., McShane, L. M., & Liu, E. T. Optimal gene expression analysis by microarrays. Cancer Cell, 2:5, 353–361, 2002.

Mann, H. and Whitney, D. On a Test of Whether One of Two Random Variables is Stochastically Larger Than the Other. Annals of Mathematical Statistics, 18: 50-60, 1947.

McAlister, D. The Law of the Geometric Mean. Proceedings of the Royal Society of London 29: 367-376, 1879.

Mulligan, M.E. The physical and chemical properties of nucleic acids. A part of Lecture notes for Biochemistry 3107 taught in the Memorial University of Newfoundland, Canada, 2003.
URL: http://www.mun.ca/biochem/courses/3107/Topics/DNA_properties.html

Fleming, J.P. and Wallace, J.J. How not to lie with statistics: the correct way to summarize benchmark results. Communications of the ACM. 29: 218-221, 1986.

Neo, S.Y., Leow, C.K., Vega, V.B., Long, P.M., Islam, A.F., Lai, P.B., Liu, E.T., and Ren, E.C. Identification of discriminators of hepatoma by gene expression profiling using a minimal dataset approach. Hepatology, 39(4):944-53, 2004.

Park, T., Yi, S.G., Lee, S., Lee, S.Y., Yoo, D.H., Ahn, J.I., and Lee, Y.S. Statistical Tests for Identifying Differentially Expressed Genes in Time-Course Microarray Experiments. Bioinformatics, 19, 694-703, 2003.

Parker, C.W. Immunoassays. In: M. P. Deutscher (ed.): Guide to Protein Purification, Academic Press, 1990.

Pevzner, P.A., Tesler, G. Human and mouse genomic sequences reveal extensive breakpoint reuse in mammalian evolution. Proc Natl Acad Sci U S A 100: 7672–7677, 2003.

Pomeroy, S. L., Tamayo, P., Gaasenbeek, M., Sturla, L. M., Angelo, M., McLaughlin, M. E., Kim, J. Y., Goumnerova, L. C., Black, P. M., Lau, C., Allen, J. C., Zagzag, D., Olson, J. M., Curran, T., Wetmore, C., Biegel, J. A., Poggio, T., Mukherjee, S., Rifkin, R., Califano, A., Stolovitzky, G., Louis, D. N., Mesirov, J. P., Lander, E. S., and Golub, T. R. Prediction of central nervous system embryonal tumour outcome based on gene expression. Nature, 415, 436–442, 2002.

Qi, Y., Rolfe, A., MacIsaac, K.D., Gerber, G.K., Pokholok, D., Zeitlinger, J., Danford, T., Dowell, R.D., Fraenkel, E., Jaakkola, T.S., Young, R.A., and Gifford, D.K. High-resolution computational models of genome binding events. Nature Biotechnology 24(8):963-70, 2006.

Ramoni, M.F., Sebastiani, P., and Kohane, I.S. Cluster Analysis of Gene Expression Dynamics. Proceedings of the National Academy of Sciences, 99, 9121-9126, 2002.

Reik, W. Stability and flexibility of epigenetic gene regulation in mammalian development. Nature 447: 425-432, 2007.

Ren B, Robert F, Wyrick JJ, Aparicio O, Jennings EG, Simon I, Zeitlinger J, Schreiber J, Hannett N, Kanin E, Volkert TL, Wilson CJ, Bell SP, Young RA. Genome-wide location and function of DNA binding proteins. Science. 290: 2306-9, 2000.

Schapire, R., and Singer, Y. Improved boosting algorithms using confidence-rated predictions. Machine Learning, 37:3, 297–336, 1999.

Schena, M. and Heller, R.A. and Theriault, T.P. and Konrad, K. and Lachenmeier, E. and Davis, R.W. Microarrays: biotechnology's discovery platform for functional genomics. Trends in Biotechnology, 16, 301-306, 1998.

Shadeo, A. and Lam, W.L. Comprehensive copy number profiles of breast cancer cell model genomes. Breast Cancer Res. 8(1): R9, 2006.

Snustad, D.P. and Simmons, M.K. Principles of Genetics. John Wiley & Sons, Inc, 2nd edition, 2000.

Strachan, T. and Read, A.P. Human Molecular Genetics. John Wiley & Sons, 2nd edition, 1999.

Talagrand, M. Sharper bounds for Gaussian and empirical processes. Annals of Probability, 22, 28–76, 1994.

Tang, S., Han, H., and Bajic, V.B. ERGDB: Estrogen Responsive Genes Database. Nucleic Acids Research, 32: D533-D563, 2004.

Valiant, L. G. A theory of the learnable. Communications of the ACM, 27:11, 1134–1142, 1984.

Vapnik, V. Statistical Learning Theory. New York, 1998.

Vapnik, V. N. Estimation of Dependencies based on Empirical Data. Springer Verlag, 1982.

Vapnik, V. N. Inductive principles of the search for empirical dependences (methods based on weak convergence of probability measures). In Proceedings of the 1989 Workshop on Computational Learning Theory, 1989.

Vapnik, V. N. The Nature of Statistical Learning Theory. Springer, 1995.

Vapnik, V. N., & Chervonenkis, A. Y. On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability and its Applications, 16:2, 264–280, 1971.

Vega, V.B, Ruan, Y., and Sung, W.-K. A Streamlined and Generalized Analysis of Chromatin ImmunoPrecipitation Paired-End diTag Data. LNCS 5103 Springer, Proceedings of the Eighth International Conference on Computational Science, 2008

Wei CL, Wu Q, Vega VB, Chiu KP, Ng P, Zhang T, Shahab A, Yong HC, Fu Y, Weng Z, Liu J, Zhao XD, Chew JL, Lee YL, Kuznetsov VA, Sung WK, Miller LD, Lim B, Liu ET, Yu Q, Ng HH, Ruan Y. A global map of p53 transcription-factor binding sites in the human genome. Cell. 124:207-19, 2006.

Weinmann AS, Yan PS, Oberley MJ, Huang TH, and Farnham PJ. Isolating human transcription factor targets by coupling chromatin immunoprecipitation and CpG island microarray analysis. Genes Dev. 16(2):235-44, 2002.

West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., J. A. O., Jr., Marks, J. R., and Nevins, J. R. Predicting the clinical status of human breast cancer by using gene expression profiles. Proc. Natl. Acad. Sci. USA, 98:20, 11462–11467, 2001.

Wilcoxon, F. Some Rapid Approximate Statistical Procedures. Stamford, CT: Stamford Research Laboratories, American Cyanamid Corporation, 1949.

Xu, X.L., Olson, J.M., and Zhao, L.P. (2002) A Regression-based Method to Identify Differentially Expressed Genes in Microarray Time Course Studies and Its Application in an Inducible Huntington's Disease Transgenic Model. Hum Mol Genet. 11(17):1977-85, 2002