POPULATION SAMC, CHIP-CHIP DATA ANALYSIS AND BEYOND

A Dissertation

by

MINGQI WU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2010

Major Subject: Statistics

POPULATION SAMC, CHIP-CHIP DATA ANALYSIS AND BEYOND

A Dissertation

by

MINGQI WU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Faming Liang
Committee Members,	Jeffrey D. Hart
	Samiran Sinha
	Yanan Tian
Head of Department,	Simon J. Sheather

December 2010

Major Subject: Statistics

ABSTRACT

Population SAMC, ChIP-chip Data Analysis and Beyond. (December 2010)

Mingqi Wu, B.S., Fudan University, P. R. China;

M.S., Fudan University, P. R. China;

M.S., Rice University;

M.S., Texas A&M University

Chair of Advisory Committee: Dr. Faming Liang

This dissertation research consists of two topics, population stochastics approximation Monte Carlo (Pop-SAMC) for Baysian model selection problems and ChIP-chip data analysis. The following two paragraphs give a brief introduction to each of the two topics, respectively.

Although the reversible jump MCMC (RJMCMC) has the ability to traverse the space of possible models in Bayesian model selection problems, it is prone to becoming trapped into local mode, when the model space is complex. SAMC, proposed by Liang, Liu and Carroll, essentially overcomes the difficulty in dimension-jumping moves, by introducing a self-adjusting mechanism. However, this learning mechanism has not yet reached its maximum efficiency. In this dissertation, we propose a Pop-SAMC algorithm; it works on population chains of SAMC, which can provide a more efficient self-adjusting mechanism and make use of crossover operator from genetic algorithms to further increase its efficiency. Under mild conditions, the convergence of this algorithm is proved. The effectiveness of Pop-SAMC in Bayesian model selection problems is examined through a change-point identification example and a large-p linear regression variable selection example. The numerical results indicate that Pop-SAMC outperforms both the single chain SAMC and RJMCMC significantly.

In the ChIP-chip data analysis study, we developed two methodologies to identify

the transcription factor binding sites: Bayesian latent model and population-based test. The former models the neighboring dependence of probes by introducing a latent indicator vector; The later provides a nonparametric method for evaluation of test scores in a multiple hypothesis test by making use of population information of samples. Both methods are applied to real and simulated datasets. The numerical results indicate the Bayesian latent model can outperform the existing methods, especially when the data contain outliers, and the use of population information can significantly improve the power of multiple hypothesis tests.

To my family and xixi

ACKNOWLEDGMENTS

One does not get to this point alone.

First and foremost, my gratitude goes to my advisor, Dr. Faming Liang, an outstanding statistician in the field of statistics computing. In the past three and half years, Dr. Liang gave me unwavering support and indoctrination about scientific research and attitude. His understanding, encouragement and personal guidance not only have provided a good basis for the present work, but more importantly, have helped me form the habit of continual thinking and have affected my personal philosophy about life; I will cherish these for ever.

I would like to thank the rest of my committee members, Dr. Jeffrey D. Hart, Dr. Samiran Sinha, and Dr. Yanan Tian. I am so grateful for them for taking time from their busy schedule to review my work and provide me with their valuable comments.

Also, my thanks are extended to Dr. Michael Longnecker for his constant support ever since I entered this department. I also appreciate the friendship and help from my classmates. Thank you all.

My parents and brothers deserve special thanks for their continuous encouragement. They are my strong supporters and always love me. Whenever I need them, they are always there. Without their love, this work would not have been possible.

Finally and most importantly, I would like to thank my wife, Zheyi Chen (xixi). I am very lucky to have her in my life. Her support, understanding and love are the biggest power to help me accomplish my Ph.D study. Thanks xixi!

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

TABLE Page

TABLE Page

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

This dissertation research consists of two topics, population stochastics approximation Monte Carlo (Pop-SAMC) for Baysian model selection problems and ChIP-chip data analysis, which belong to statistics computing and bioinformatics category respectively. The work cut across the fields of stochastic approximation Monte Carlo (SAMC), population based MCMC methods, Baysian model selection problems, ChIP-chip data analysis, latent variable models, multiple hypotheses testing, non-parametric methods. This chapter provides the backgrounds for each research topic, which is organized as follows. Section 1.1 describes the basic idea of Bayesian approach to model selcection problems and introduces the representative methods to compute posterior probabilities of the potential models in the literature. Section 1.2 contains the motivation of ChIP-chip technology as well as a short review of the existing methods for ChIP-chip data analysis. Section 1.3 displays the structure of this dissertation.

1.1 Bayesian Model Selection

Model selection is the task of selecting a mathematical model from a set of potential models, given evidence. A Bayesian approach to model selection problems proceeds as follows. Suppose that we have a posterior distribution of models denoted by $P(m|\boldsymbol{y}) \propto P(m)f(\boldsymbol{y}|m)$, where $\boldsymbol{y}$ denotes the data, m is the model index, which belongs to a set of competing models, $m \in M$, and $P(m)$ is the prior probability of model m, $f(\boldsymbol{y}|m)$ is the marginal likelihood, which is obtained by integrating out the

---

This dissertation follows the style of *Biometrics*.

model parameters. By comparing the posterior probability of each potential models, the model with the maximum posterior probability will be selected.

Various computing methods have been developed to estimate the posterior probability of potential models. The criteria to judge each method is based on the accuracy of their estimation. With its ability to traverse over the space of possible models, the reversible jump MCMC (RJMCMC) algorithm (Green, 1995) has been shown to be quite effective for Bayesian model selection problems, especially when the model space is simple, i.e., there are no well-separated multiple modes. However, when the distribution of model probability is complex, RJMCMC is prone to get trapped into local modes. To overcome the local-trap problem, Liang and his coauthors (2007a) have proposed a stochastic approximation Monte Carlo (SAMC) algorithm, which make the dimension-jumping moves freely and thus provide a full exploration for the model space by introducing a self-adjusting mechanism based on the past samples. This mechanism penalizes the over-visited subregions and rewards the under-visited subregions, and thus enables the system to escape from local traps very quickly.

SAMC has been compared with RJMCMC in Bayesian model selection problems (Liang et al. 2007a). The results show that SAMC outperforms RJMCMC when the model space is complex. However, when the model space is simple, e.g., it only contains several models with comparable probability, SAMC may not be better than RJMCMC. Inspired by the success of population-based MCMC algorithms, e.g., adaptive direction sampling (Gilks et al., 1994), conjugate gradient Monte Carlo (Liu, Liang and Wong, 2000), parallel tempering (Geyer, 1991; Hukushima and Nemoto, 1996), and evolutionary Monte Carlo (Liang and Wong, 2000, 2001), in Chapter II, we propose a population SAMC (Pop-SAMC) algorithm to accelerate the convergence of SAMC. The new algorithm works on a population of of SAMC chains, which provides a more efficient self-adjusting mechanism, and consequently improves the

convergence of SAMC. Furthermore, running a population of chains in parallel enables incorporation of *crossover* operators from the genetic algorithm (Holland, 1975) into simulations. With this operator, the distributed information across a population could be shared among chains/population, the efficiency of the new algorithm can be further increased. The effectiveness of Pop-SAMC in Bayesian model selection problems is examed through two typical examples. The results show that Pop-SAMC can make a significant improvement over SAMC and it can also work better than RJMCMC even when the model space is simple.

1.2 ChIP-chip Data Analysis

The chromatin immunoprecipitation (ChIP) coupled with microarray (chip) analysis, provides the researchers an efficient way of mapping protein-DNA interactions across a whole genome. The ChIP-chip technology has been used in a wide range of biomedical studies, such as identification of human transcription factor binding sites (Cawley et al., 2004), investigation of DNA methylation (Zhang et al., 2006), and investigation of histone modifications in animals (Bernstein et al., 2005) and plants (Zhang et al., 2007). Data from ChIP-chip experiments encompass DNA-protein interaction measurements on millions of short oligonucleotides (probes) which often tile one or several chromosomes or even the whole genome. The data analysis consists of two steps: (1) identifying the bound regions where DNA and the protein are cross-linked in the experiments; and (2) identifying the binding sites through sequence analyses of the bound regions. The goal of this work is to develop effective methods for the first step analysis.

Several methodologies has been develeoped to analyse ChIP-chip data in the literature, they can be roughly grouped into three categories, the sliding window

methods (Cawley et al., 2004; Bertone et al., 2004; Ji and Wong, 2005; Keles et al., 2006), the hidden Markov Model (HMM) methods (Li et al., 2005; Ji and Wong, 2005; Munch et al., 2006; Humburg et al., 2008), and the Bayesian methods (Qi et al., 2006; Keles, 2007; Gottardo et al., 2008). A detail review for each category will be given in Chapter III section 3.2.1. Other methods have been suggested, e.g., by Zheng et al. (2007), Huber et al. (2006) and Reiss et al. (2008), but are less common.

Analysis of the ChIP-chip data is very challenging, due to the large amount of probes and the small number of replicates. The existing methods do not perform satisfactorily in various aspects. The power of the sliding window tests is usually low, especially for the tests for the regions where the probe density is low. This is because there will be only very limited neighboring information available for those tests. The HMM methods have the potential to make use of all probe information, where the model parameters are estimated using all available data. However, parameter estimation for these models is typically either heuristic or suboptimal, leading to inconsistencies in their applications. Bayesian methods have also the potential to make use of all probe information. Like the HMM methods, the Bayesian methods estimate the model parameters using all available data. However, these methods usually require multiple replicates or some extra experimental information to parameterize the model, and long CPU time due to involving of MCMC simulations.

In Chapter III, we propose two new methods to analyze ChIP-chip data. One is a Bayesian latent model, the other is a population-based nonparametric testing method. The former models the neighboring dependence of probes by introducing a latent indicator vector; The later provides a nonparametric method for evaluation of test scores in a multiple hypothesis test by making use of population information of samples. Both methods are applied to real and simulated datasets. The numerical results indicate the Bayesian latent model can outperform the existing methods, es-

pecially when the data contain outliers, and the use of population information can significantly improve the power of multiple hypothesis tests.

1.3 Dissertation Structure

Chapter II develops a Pop-SAMC algorithm, its convergence is shown under mild conditions. The effectiveness of this algorithm for Bayesian model selection problems is examed through two typical examples along with the comparisons with SAMC and RJMCMC. Chapter III is independent of Chapter II and is dedicated to new methodology development for ChIP-chip data analysis. Two new methods are proposed: Bayesian latent model and population-based nonparametric testing method. Both methods are applied to real and simulated datasets with comparisons with existing methods. Chapter IV gives a summary of this dissertation and points out some directions for future research.

CHAPTER II

POPULATION SAMC FOR BAYESIAN MODEL SELECTION PROBLEMS

2.1 Introduction

Given data, how to select an optimal model, according to some criteria, from a set of potential models is an important topic for statistician. In the Bayesian framework, the success of choosing the right model relies on how accurately you can estimate the posterior probability of each of the potential models. With its ability to traverse over the space of possible models, the reversible jump MCMC (RJMCMC) algorithm (Green, 1995) has been shown to be quite effective for Bayesian model selection problems, especially when the model space is simple, i.e., there are no well-separated multiple modes. However, when the distribution of model probability is complex, RJMCMC is prone to get trapped into local modes.

To overcome the local-trap problem, Liang and his coauthors (2007a) have proposed a stochastic approximation Monte Carlo (SAMC) algorithm, which make the dimension-jumping moves freely and thus provide a full exploration for the model space by introducing a self-adjusting mechanism based on the past samples. The basic idea of SAMC can be described as follows. Suppose that we are interested in sampling from a distribution,

$$f(x) = c\psi(x), \quad x \in \mathcal{X}, \tag{2.1}$$

where $\mathcal{X}$ is the sample space and c is an unknown constant. Let $E_1, ..., E_k$ denote a partition of $\mathcal{X}$, and let $w_i = \int_{E_i} \psi(x)dx$ for $i = 1, ..., k$. SAMC seeks to draw samples

from the trial distribution

$$f_w(x) \propto \sum_{i=1}^{k} \frac{\pi_i \psi(x)}{w_i} I(x \in E_i) \tag{2.2}$$

where π_i's are pre-specified constants such that $\pi_i > 0$ for all i and $\sum_{i=1}^{k} \pi_i = 1$, which define the desired sampling frequency for each of the subregions. If $w_1, ..., w_k$ can be well estimated, sampling from $f_w(x)$ will result in a "random walk" in the space of subregions (by regarding each subregion as a point) with each subregion being sampled with a frequency proportional to π_i. Hence the local-trap problem can be overcome essentially, provided that the sample space is partitioned appropriately. The way to partition sample space is problem dependent. For examples, if our goal is to minimize the target distribution, then we can partition the sample space according to the target density function; If our goal is model selection, then we can partition the sample space according to the index of models.

As mentioned above, the success of "random walk" in the sample space depends crucially on the estimation of w_i. SAMC provides a systematic way to estimate w_i in an online manner. Let θ_{ti} denote the working estimate of $\log(w_i/\pi_i)$ obtained at iteration t, and let $\theta_t = (\theta_{t1}, ..., \theta_{tk})$. Let $\{\gamma_t\}$ be a positive, nondecreasing sequence satisfying

$$(i) \quad \sum_{t=1}^{\infty} \gamma_t = \infty, \qquad (ii) \quad \sum_{t=1}^{\infty} \gamma_t^{\zeta} < \infty, \tag{2.3}$$

for any $\zeta > 1$. For example, one may set

$$\gamma_t = \frac{T_0}{max(T_0, t)}, \qquad t = 1, 2, ..., \tag{2.4}$$

for some value $T_0 > 1$. Since $f_w(x)$ is invariant to a scale change of $w = (w_1, ..., w_k)$, i.e., $f_{cw}(x) = f_w(x)$ for any $c > 0$, θ_t can be kept in a compact set Θ by adding to or subtracting a constant vector from θ_t, provided that Θ is large enough and

$0 < \int_{\mathcal{X}} \psi(x)dx < \infty$. Under the above setting, one iteration of SAMC can be described as follows:

The SAMC algorithm:

1. **Metropolis-Hastings(MH) sampling**. Simulate a sample x_t by a single MH update with the invariant distribution

$$f_{\theta_t}(x) \propto \sum_{i=1}^{k} \frac{\psi(x)}{e^{\theta_{ti}}} I(x \in E_i) \tag{2.5}$$

2. **Weight updating**. Set

$$\theta^* = \theta_t + \gamma_{t+1}(\boldsymbol{e}_t - \boldsymbol{\pi}), \tag{2.6}$$

where $\boldsymbol{e}_t = (I(x_t \in E_1), ..., I(x_t \in E_k))$ and $I(\cdot)$ is the indicator function. If $\theta^* \in \Theta$, set $\theta_{t+1} = \theta^*$; otherwise, set $\theta_{t+1} = \theta^* + \boldsymbol{c}^*$, where $\boldsymbol{c}^* = (c^*, ..., c^*)$ can be an arbitrary vector which satisfies the condition $\theta^* + \boldsymbol{c}^* \in \Theta$.

A remarkable feature of SAMC is its self-adjusting mechanism, which operates based on past samples. This mechanism penalizes the over-visited subregions and rewards the under-visited subregions, and thus enables the system to escape from local traps very quickly. Mathematically, if a subregion i is visited at time t, $\theta_{t+1,i}$ will be updated to a larger value, $\theta_{t+1,i} \leftarrow \theta_{t,i} + \gamma_{t+1}(1 - \pi_i)$, such that, this subregion has a smaller probability to be visited in the next iteration. On the other hand, for those regions, $j(j \neq i)$, not visited this itertaion, $\theta_{t+1,j}$ will decrease to a smaller value, $\theta_{t+1,j} \leftarrow \theta_{t,j} - \gamma_{t+1}\pi_j$, such that, the chance to visit these regions will increase next iteration.

Although SAMC has been quite effective in exploring the whole sample space, its convergence is usually slow. Because, SAMC runs in a single chain. At each iteration, there is one and only one component of $\boldsymbol{e}_t$ is not equal to zero, and the

information gained for θ_t is minimal and thus the adjustment process is slow. As a result, a large variation of θ_t will be observed even after long iterations, especially when the number of subregions is large. Inspired by the success of population-based MCMC algorithms, e.g., adaptive direction sampling (Gilks et al., 1994), conjugate gradient Monte Carlo (Liu, Liang and Wong, 2000), parallel tempering (Geyer, 1991; Hukushima and Nemoto, 1996), and evolutionary Monte Carlo (Liang and Wong, 2000, 2001), we propose a population SAMC (Pop-SAMC) algorithm to accelerate the convergence of SAMC. The new algorithm works on a population of of SAMC chains. The benefits are two-fold. Firstly, it provides a more efficient self-adjusting mechanism. Intuitively, when we have a population of SAMC chains running in parallel, the information gained for θ_t at each iteration is increased, which leads to a more accurate adjustment of θ_t. Consequently, this improves the convergence of θ_t. Secondly, running a population of chains in parallel enables incorporation of *crossover* operators from the genetic algorithm (Holland, 1975) into simulations. With this operator, the distributed information across a population could be shared among chains/population, the efficiency of this algorithm can be further increased.

The effectiveness of Pop-SAMC in Bayesian model selection problems is examed through a change-point identification example and a large-p linear regression variable selection example. The numerical results show that the new method performs significantly better than both the single chain SAMC and RJMCMC, in estimating probabilities of competing models. A rigorous proof for the convergence of Pop-SAMC is provide in Appendix.

The remainder of this chapter is organized as follows. In Section 2.2, we describe the Pop-SAMC algorithm and study its convergence theory. In Section 2.3, we illustrate the efficience of Pop-SAMC in estimating partition weight through a multimodal example. In Section 2.4, we show the superiority of Pop-SAMC in Bayesian

model selection problems by studying a change-point identification example and a large-p linear regression variable selection example respectively, with comparisons with SAMC and RJMCMC. In Section 2.5, we conclude this chapter with a brief discussion.

2.2 Population SAMC Algorithm

2.2.1 Population SAMC

Consider the distribution defined in (2.1). Suppose the sample space $\mathcal{X}$ has been partitioned into disjoint subregions, denoted by $E_1, ..., E_k$, and the same gain factor sequence setting, $\{\gamma_t\}$ as defined in (2.3), (2.4) for single chain SAMC, will be used in the Pop-SAMC.

As its name suggests, Pop-SAMC works on a population of SAMC chains in parallel. At each iteration, a set of independent samples, called a population, are generated. Let $\boldsymbol{x}^t = (x_1^t, ..., x_N^t)$ represent the population generated at iteration t, and N is the population size. One iteration of Pop-SAMC algorithm consists of the following two steps:

The Pop-SAMC algorithm

1. **MH Sampling**

For each of the population chains, simulate a sample x_i^t, for $i = 1, ..., N$, by a single MH update with the invariant distribution as defined in (2.5). A new population of samples $\boldsymbol{x}^t$ will be obtained.

2. **Weight updating**

Set

$$\theta^* = \theta_t + \gamma_{t+1}(\hat{\boldsymbol{p}}_t - \boldsymbol{\pi}), \tag{2.7}$$

where $\hat{\boldsymbol{p}}_t = (\sum_{i=1}^N I(x_i^t \in E_1)/N, ..., \sum_{i=1}^N I(x_i^t \in E_k)/N)$. If $\theta^* \in \Theta$, set $\theta_{t+1} = \theta^*$;

otherwise, set $\theta_{t+1} = \theta^* + \boldsymbol{c}^*$, where $\boldsymbol{c}^* = (c^*, ..., c^*)$ can be an arbitrary vector which satisfies the condition $\theta^* + \boldsymbol{c}^* \in \Theta$.

Pop-SAMC is a generalized version of SAMC, its generalization is mainly in the weight updating step. Since a population of chains run parallel in the new algorithm, multiple samples will be generated at each iteration, which enables a frequency estimator $\hat{\boldsymbol{p}}_t$ to estimate the probability that a sample is drawn from each subregion at iteration t, i.e. $\boldsymbol{p}_t = (\int_{E_1} f_{\theta_t}(x)dx, ..., \int_{E_k} f_{\theta_t}(x)dx)$. Compared with the indicator vector $\boldsymbol{e}_t$ used by single chain SAMC in updating θ_t, $\hat{\boldsymbol{p}}_t$ not only carries more information of the partition weight, but also is a more accurate estimator of $\boldsymbol{p}_t$. This is the key reason that Pop-SAMC has the potential to outperform SAMC in updating θ_t.

2.2.2 Convergence

Regarding the convergence of Pop-SAMC, we note that for the empty subregions, the corresponding components of θ_t will trivially converge to $-\infty$ as $t \to \infty$. Therefore, without loss of generality, we show in Appendix only the convergence of the algorithm for the case that all subregions are non-empty. Extending the proof to the general case is trivial, since replacing (2.7) by (2.8) (given below) will not change the process of pop-SAMC simulation.

$$\theta' = \theta_t + \gamma_t(\widehat{\boldsymbol{p}}_{t+1} - \boldsymbol{\pi} - \boldsymbol{\nu}), \tag{2.8}$$

where $\boldsymbol{\nu} = (\nu, \ldots, \nu)$ is an m-vector of ν, and $\nu = \sum_{j\in\{i:E_i=\emptyset\}} \pi_j/(k - k_0)$ and k_0 is the number of empty subregions.

In our proof, we assume that Θ is a compact set; that is, there exists a constant vector $\boldsymbol{c}_t$ for each t such that $\boldsymbol{c}_t + \theta_t \in \Theta$. This assumption is made only for the reason of mathematical simplicty. Extension of our results to the case that $\Theta = \mathbb{R}^m$ is trivial

with the technique of varying truncations studied in Chen (2002) and Andrieu et al. (2005). Interested readers can refer to Liang et al. (2010) for the details, where the convergence of SAMC is studied with $\Theta = \mathbb{R}^m$. In the simulations of this work, we set $\Theta = [-10^{100}, 10^{100}]^m$, as a practical matter, this is equivalent to setting $\Theta = \mathbb{R}^m$.

Under the above assumptions, we have the following theorem concerning the convergence of the Pop-SAMC algorithm, whose proof can be found in Appendix.

Theorem 2.2.1 *Let $E_1, \ldots, E_m$ be a partition of a compact sample space $\mathcal{X}$ and $\psi(x)$ be a non-negative function defined on $\mathcal{X}$ with $0 < \int_{E_i} \psi(x)dx < \infty$ for all E_i's. Let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_m)$ be an m-vector with $0 < \pi_i < 1$ and $\sum_{i=1}^m \pi_i = 1$. Let $\{\gamma_t\}$ be a non-increasing, positive sequence satisfying (2.3). If Θ is compact and the drift condition [condition (A_2) given in Appendix] is satisfied, then, as $t \to \infty$, we have almost surely,*

$$\theta_{ti} \to \begin{cases} C + \log\left(\int_{E_i} \psi(x)dx\right) - \log(\pi_i + \nu), & \text{if } E_i \neq \emptyset, \\ -\infty, & \text{if } E_i = \emptyset. \end{cases} \tag{2.9}$$

where C is an unknown constant, $\nu = \sum_{j \in \{i: E_i = \emptyset\}} \pi_j/(k - k_0)$, and k_0 is the number of empty subregions.

The constant C can be determined by imposing a constraint, e.g., $\sum_{i=1}^m e^{\theta_{ti}}$ is equal to a known number.

The drift condition assumption is classical, which implies the existence of the stationary distribution $f_\theta(x)$ for any $\theta \in \Theta$. To have the drift condition satisifed, we assume that $\mathcal{X}$ is compact and $f(x)$ is bounded away from 0 and ∞ on $\mathcal{X}$. This assumption is true for many Bayesian model selection problems, for example, the Bayesian change-point identification and the Bayesian regression variable selection problems considered in this chapter. For both problems, after integrating out the

model parameters, the sample space is reduced to a finite set of models. For continuum systems, one may restrict $\mathcal{X}$ to the region $\{x : \psi(x) \geq \psi_{min}\}$, where ψ_{min} is sufficiently small such that the region $\{x : \psi(x) < \psi_{min}\}$ is not of interest. Otherwise, one may put conditions on the tail behavior of $f(x)$ as prescribed by Andrieu et al. (2005). For the proposal distribution used in the MH sampling, we assume it to satisfy the local positive condition: There exists $\epsilon_1 > 0$ and $\epsilon_2 > 0$ such that $q(x, y) \geq \epsilon_2$ if $|x - y| \leq \epsilon_1$. This condition is quite standrad and has been widely used in the study of MCMC convergence, see, e.g., Roberts and Tweedie (1996).

Theorem 2.2.2 concerns the convergence rate of θ_t, which gives a L^2 upper bound for the mean squared error of θ_t. Its proof can be found in Appendix.

Theorem 2.2.2 *Under the conditions as assumed in Theorem 2.2.1, there exists a constant λ such that for each non-empty subregion*

$$E\|\theta_{ti} - \theta_i^*\|^2 \leq \lambda\gamma_t,$$

where $\theta_i^* = C + \log\left(\int_{E_i} \psi(x)dx\right) - \log(\pi_i + \nu)$.

In Appendix, we linked the upper bound λ to the parameter updating vector $\hat{\boldsymbol{p}}_t - \boldsymbol{\pi}$, and showed that the Pop-SAMC algorithm tends to have a smaller value of λ than the single-chain SAMC algorithm. This implies that Pop-SAMC tends to converge faster than the single-chain SAMC. In what follows, we summarize this result into a corrollary of Theorem 2.2.2, whose proof is given in the appendix.

Theorem 2.2.3 *Pop-SAMC tends to have a smaller L^2 upper bound than the single-chain SAMC algorithm.*

2.2.3 Crossover

Another attractive feature of Pop-SAMC is that, with parallel indepent running chains, information can be exchange or share between different chains to further increase the algorithm's efficiency. Borrow information globally can be realized through *crossover* operator from the genetic algorithm (Holland, 1975). One pioneer work in this direction is the evolutionary Monte Carlo algorithm (EMC) (Liang and Wong, 2000, 2001). Motivated by successes of the EMC, we incorporated crossover into Pop-SAMC and below we only discuss the simplest case for illustration purpose.

Suppose we have a population $\boldsymbol{x} = (x_1, ..., x_N)$ at iteration t, where $x_i = (a_i^1, ..., a_i^d)$ is a d-dimensional vector, called an individual or chromosome in Pop-SAMC. Let p_c denote the crossover rate, and $N_c = N \times p_c$ is the number of the chromosome in the current population that will be crossovered, which should be even. For each iteration, the crossover operator works as follows:

1. **Selection**

Random select N_c chromosomes from the current population $\boldsymbol{x}$, and randomly allocate them to form $N_c/2$ pairs.

2. **Crossover**

For each of the pairs, (x_i, x_j) $(i \neq j)$, an integer crossover point c is first decided by drawing unifromly from $\{1, ..., d\}$, then two new chromosomes (y_i, y_j) are obtained by swapping the components of the two parental chromosomes to the right of the crossover point. As shown below.

$$x_i = (a_i^1, ..., a_i^d) \qquad y_i = (a_i^1, ..., a_i^c, a_j^{c+1}, ..., a_j^d)$$

$$\Rightarrow$$

$$x_j = (a_j^1, ..., a_j^d) \qquad y_j = (a_j^1, ..., a_j^c, a_i^{c+1}, ..., a_i^d)$$

Following MH rules, the new chromosomes are accepted into the new population $\boldsymbol{y}$ with probability equal to $min(1, r_c)$, and

$$r_c = \frac{f_{\theta_t}(y_i) f_{\theta_t}(y_j)}{f_{\theta_t}(x_i) f_{\theta_t}(x_j)} \times \frac{T((x_i, x_j)|(y_i, y_j))}{T((y_i, y_j)|(x_i, x_j))}, \tag{2.10}$$

where $T((y_i, y_j)|(x_i, x_j)) = P((x_i, x_j)|\boldsymbol{x})P((y_i, y_j)|(x_i, x_j))$. $P((x_i, x_j)|\boldsymbol{x})$ is the select probability of (x_i, x_j) from the population $\boldsymbol{x}$ and $P((y_i, y_j)|(x_i, x_j))$ is the generating probability of (y_i, y_j) from the parental chromosomes (x_i, x_j). Since our parental chromosomes are chosen randomly from the population and by the symmetric properties of the crossover operator, it is easy to show that $T((y_i, y_j)|(x_i, x_j)) = T((x_i, x_j)|(y_i, y_j))$. Thus, equation (2.10) will reduce to the likelyhood ratio between the new chromsomes and the old ones.

$$r_c = \frac{f_{\theta_t}(y_i) f_{\theta_t}(y_j)}{f_{\theta_t}(x_i) f_{\theta_t}(x_j)} \tag{2.11}$$

In the Pop-SAMC, the crossover operation can be included in the *MH Sampling* step, N_c chromsomes are updated using crossover operator, and $(N - N_c)$ chromosomes are updated with a single MH step respectively according to the invariant distribution as defined in (2.5).

The rationale behind the effectiveness of crossover operator can be explained as follows. Pop-SAMC works on a population of chains. At each iteration, some samples obtained in one chain may be better than others in terms of likelyhood value. If this chain happens to be selected into the crossover operation, by exchanging parts of its chromosome with other individuals, the overall quality of the whole population will be improved. In short, combining with crossover, Pop-SAMC's self-adjusting mechanism can use the information from two dimensions, vertically learning from past samples, horizontally mergering global informations, which make it super efficient.

2.3 Population SAMC vs SAMC: An Illustration Example

To illustrate the superior performance of Pop-SAMC, compared with SAMC, in estimating sample space partition weight, we study a multimodal example (Liang and Wong, 2001), whose density function is given by

$$f(x) = \frac{1}{2\pi\sigma^2}\sum_{i=1}^{20}\alpha_i exp\Big\{-\frac{1}{2\sigma^2}(\boldsymbol{x}-\boldsymbol{\mu}_i)'(\boldsymbol{x}-\boldsymbol{\mu}_i)\Big\}, \tag{2.12}$$

where each component has an equal variance $\sigma^2 = 0.01$ and is assigned an equal weight $\alpha_1 = ... = \alpha_{20} = 0.05$. See Liang and Wong (2001) for the values of the mean vectors. It is shown that some componets are far from others (more than 30 times of the standard deviation in distance), e.g., the components at the right two corners (refer to Figure 1), which puts great challege on the testing algorithm.

Let $\mathcal{X} = [-10^{100}, 10^{100}]^2$, and let it be partitioned according to $U(x) = -log\{f(x)\}$, (In terms of physics, $U(x)$ is called the energy function of the distribution), with an equal energy bandwidth $\Delta u = 0.5$ into the following subregions: $E_1 = \{x : u(x) < 0\}, E_2 = \{x : 0 \leq u(x) < 0.5\}, ..., E_{50} = \{x : u(x) > 24.0\}$ and the desired sampling distribution to be uniform $\pi_1 = ... = \pi_{50} = \frac{1}{50}$. Both Pop-SAMC and SAMC provide a self-adjusting mechanism to online learning the partition weights $\int_{E_i} f(x)dx/\pi_i$, for $i = 1, ..., 50$. With uniform desired sampling distribution, the partition weight reduce to the probability that a sample is drawn from each subregion i, i.e. $P(E_i) = \int_{E_i} f(x)dx$. Thus, to compare their efficiency to learn the partition weight is equivalent to compare their estimates of $P(E_i)$. The true value of $P(E_i)$ can be calculated with a total of 20×10^8 samples drawn equally from each of the twenty components of $f(x)$. In order to have a fair comparison, we run each algorithm with the same number of energy evaluations, and use the same proposal distribution with the same step size.

Table 1: Comparison of the estimated partition weight $P(E_i)$ for the multimodal example. The number in the parentheses is the standard error. CPU: the CPU time (in seconds) cost by a single run of the corresponding algorithm on a Intel Core 2 Duo 3.0 GHz computer.

Estimates	True Prob.(%)	Pop-SAMC	SAMC
$P(E_2)$	23.87	23.85(0.05)	23.65(0.85)
$P(E_3)$	30.27	30.25(0.06)	30.31(0.92)
$P(E_4)$	18.56	18.59(0.04)	18.13(0.46)
$P(E_5)$	11.24	11.21(0.02)	11.30(0.47)
$P(E_6)$	6.63	6.64(0.02)	6.27(0.12)
$P(E_7)$	3.84	3.85(0.01)	3.63(0.07)
$P(E_8)$	2.26	2.26(0.01)	2.15(0.04)
$P(E_9)$	1.34	1.34(0.00)	1.27(0.02)
CPU (s)	—	1.81	2.36

Pop-SAMC was run for this example 100 times independently with the setting: $N = 10$, $T_0 = 50$, $k = 50$, Iterations=10^5 and SAMC was also applied for this example 100 times independently with the same setting except the following parameters: $T_0 = 100$, Iterations=10^6. The computational results along with the true value of $P(E_i)$ for $i = 2, ..., 9$ are summarized in Table 1, other subregions with zero or tiny probability are not listed. The results show that Pop-SAMC has made a significant improvement in accuracy over SAMC in estimating the partition weight. On average, the standard error of the Pop-SAMC estimeates is only about 1/10 of that of the SAMC estimates. These results are achieved under the same number of engergy evaluations for each of the algorithm, which made the compairson fair. In the table, we also reported

the CPU times cost by a single run of each method to further clarify the fairness of our comparison. Pop-SAMC even cost less CPU time than SAMC in this example. In addition, we examed the sample path of Pop-SAMC. 100 samples were collected for each of the 10 chains, at equally spaced time points since the beginning of the simulation. The evolving path of the 100×10 samples has been shown in Figure 1. It clearly shows that Pop-SAMC can have a thorough exploratin of the sample space in a very short time. In summary, Pop-SAMC converges much faster than SAMC in estimating the sample space partition weight, while maintaining the suprior ability of SAMC in sample space exploration.

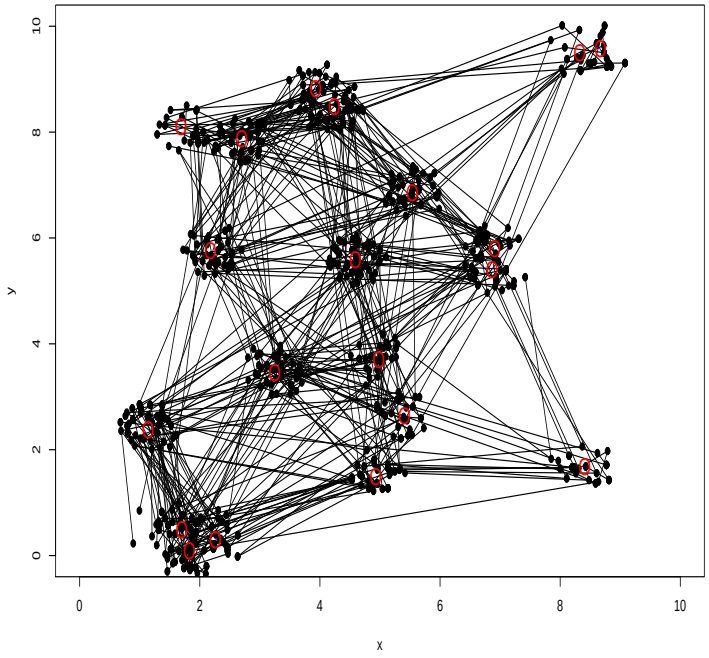


Figure 1. Sample path of population SAMC.

2.4 Bayesian Model Selection Problems

2.4.1 A Change-point Identification Example

The change-point identification problem can be described as follows. Suppose we have a sequence of independent observations $\boldsymbol{y} = (y_1, y_2, ..., y_n)$ and they can be partitioned into blocks, such that the sequence follows the same distribution within blocks. Our goal is to identify the unknown number and the locations of the boundary, called change-point, between blocks. For simplicity, we assume that the observations within each block are drawn independently from a normal distribution $N(\mu_b, \sigma_b^2)$, where b is the index of blocks. After a change-point, both the mean and variance may shift.

In the literature, this problem has been studied by several authors using simulation-based methods, e.g., the Gibbs sampler (Barry and Hartigan, 1993), reversible jump MCMC (Green, 1995), jump diffusion (Philips and Smith, 1996), and evolutionary Monte Carlo (Liang and Wong, 2000). In this paper, we follow Liang and Wong (2000)'s approach, a latent vecotor is introduced to indicate the change-point position. Let $\boldsymbol{z} = (z_1, ..., z_{n-1})$ be a latent binary vector associated with the observations index except the last one, indicating the potential change-point, where $z_i = 1$ indicates a change-point, and 0 otherwise. Let $\boldsymbol{z}^{(k)}$ correspond to a model with k change-point, with unknown positions of the change-points be denoted by $c_1, ..., c_k$. For convenience, we let $c_0 = 0$ and $c_{k+1} = n$ and they follow the order $c_0 < c_1 < c_2 < ... < c_k < c_{k+1}$. Under the above setting, we have

$$y_i \sim N(\mu_b, \sigma_b^2), \quad c_{b-1} < i \leq c_b, \tag{2.13}$$

for $b = 1, 2, ..., k+1$ and $i = 1, ..., n$. For model $\boldsymbol{z}^{(k)}$, the parameter vector is $\boldsymbol{\theta}^{(k)} = (\boldsymbol{z}^{(k)}, \mu_1, \sigma_1^2, ..., \mu_{k+1}, \sigma_{k+1}^2)$. Let $\mathcal{X}_k$ denote the model space with k change-points,

$\boldsymbol{z}^{(k)} \in \mathcal{X}_k$, and $\mathcal{X} = \bigcup_{k=0}^{n-1} \mathcal{X}_k$. The log-likelihood function of model $\boldsymbol{\theta}^{(k)}$ is then

$$L\big(\boldsymbol{y}|\boldsymbol{\theta}^{(k)}\big) = -\sum_{i=1}^{k+1}\Big\{\frac{c_i - c_{i-1}}{2} log\sigma_i^2 + \frac{1}{2\sigma_i^2}\sum_{j=c_{i-1}+1}^{c_i}(y_j - \mu_i)^2\Big\}. \tag{2.14}$$

To conduct a Bayesian analysis for the model, we specify the following prior distribution for the model parameters:

$$\sigma_i^2 \sim IG(\alpha, \beta), \quad P(\mu_i) \propto 1, \tag{2.15}$$

where $IG(\cdot, \cdot)$ denotes an inverse Gamma distribution with hyperparameters α, β, and an improper uniform prior is put on each μ_i. In addition, we assume that the latent vector $\boldsymbol{z}^{(k)}$ follows a truncated Poisson distribution,

$$P(\boldsymbol{z}^{(k)}) \propto \frac{\lambda^k}{\sum_{j=0}^{n-1}\frac{\lambda^j}{j!}}\frac{(n-1-k)!}{(n-1)!}, \quad k = 0, 1, ..., n-1, \tag{2.16}$$

where λ is a hyperparameter; $(n-1)$ is the largest number of change-points allowed by this model. Conditioning on the number of change-points k, we put an equal prior probability on all possible configurations of $\boldsymbol{z}^{(k)}$. By assuming that all the priors are independent, the log-prior density is

$$P\big(\boldsymbol{\theta}^{(k)}\big) = a_k - \sum_{i=1}^{k+1}\Big\{(\alpha+1) log\sigma_i^2 + \frac{\beta}{\sigma_i^2}\Big\}, \tag{2.17}$$

where $a_k = (k+1)\{\alpha log\beta - log\Gamma(\alpha)\} + log(n-1-k)! + klog\lambda$. Combining the likelihood (2.14) and prior distributions (2.17), integrating out μ_i and σ_i^2 for $i = 1, ..., k+1$ and taking the logarithm, we get the following log-posterior density function

$$\begin{aligned}\log P\big(\boldsymbol{z}^{(k)}|\boldsymbol{y}\big) &= a_k + \frac{k+1}{2}\log 2\pi - \sum_{i=1}^{k+1}\Big\{\frac{1}{2}\log(c_i - c_{i-1}) - \log\Gamma\Big(\frac{c_i - c_{i-1} - 1}{2} + \alpha\Big) \\ &+ \Big(\frac{c_i - c_{i-1} - 1}{2} + \alpha\Big)\log\Big[\beta + \frac{1}{2}\sum_{j=c_{i-1}+1}^{c_i} y_j^2 - \frac{\big(\sum_{j=c_{i-1}+1}^{c_i} y_j\big)^2}{2(c_i - c_{i-1})}\Big]\Big\}\end{aligned} \tag{2.18}$$

Samples generated from the above posterior distibution can be used to estimate $P(\mathcal{X}_k|\boldsymbol{y})$. For Pop-SAMC, if we let $E_k = \mathcal{X}_k$ and $\psi(\cdot) \propto P\big(\boldsymbol{z}^{(k)}|\boldsymbol{y}\big)$, it follows from (2.9) that $\hat{w_i}^{(t)}/\hat{w_j}^{(t)} = e^{\theta_{ti}-\theta_{tj}}$ forms a consistent estimator for the Bayes factor $P(\mathcal{X}_i|\boldsymbol{y})/P(\mathcal{X}_j|\boldsymbol{y})$. Without loss of generality, we restrict our consideration to the models with $k_{min} \le i, j \le k_{max}$, where k_{min} and k_{max} can be determined easily with a short pilot run of the above algorithm, the probability of those models outside this range is zero. For the change-point identification problem, the details of the sampling step of Pop-SAMC are designed similarly to those described in Liang (2009), except the weight updating step, which follows (2.9).

In this example, the simulated dataset consists of 1000 observations with $y_1, ..., y_{120} \sim N(-0.5, 1)$, $y_{121}, ..., y_{210} \sim N(0.5, 0.5)$, $y_{211}, ..., y_{460} \sim N(0, 1.5)$, $y_{461}, ..., y_{530} \sim N(-1, 1)$, $y_{531}, ..., y_{615} \sim N(0.5, 2)$, $y_{616}, ..., y_{710} \sim N(1, 1)$, $y_{711}, ..., y_{800} \sim N(0, 1)$, $y_{801}, ..., y_{950} \sim N(0.5, 0.5)$, and $y_{951}, ..., y_{1000} \sim N(1, 1)$. The time plot is shown in Figure 2. For this example, we set the hyperparameters $\alpha = \beta = 0.5$, which forms a vague prior for σ_i^2; and set $\lambda = 1$. After a short pilot run, we set $k_{min} = 7$ and $k_{max} = 14$.

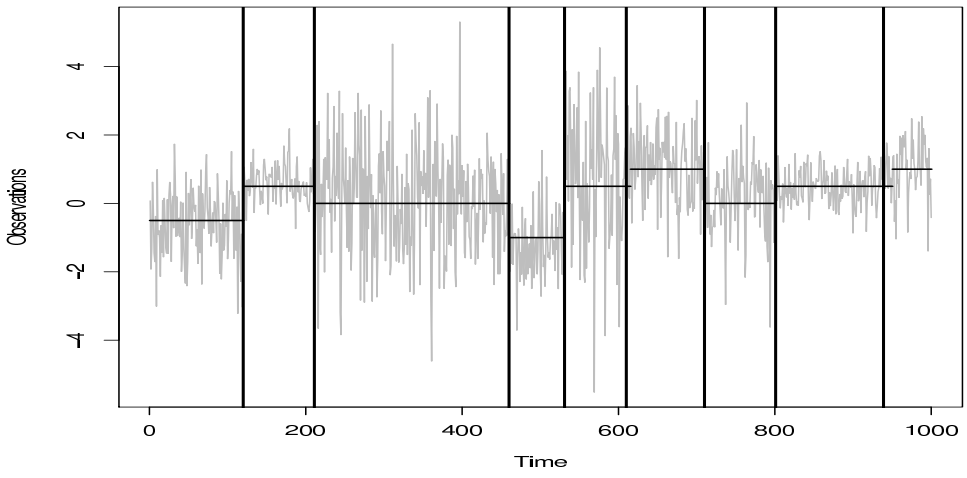


Figure 2. A comparison of the true change-point position (horizontal line) and the MAP estimates (verticla line).

First, Pop-SAMC was run for this example 50 times independently with the following setting: $N = 20$, $T_0 = 10$, Iterations=5×10^4 and $\pi_1 = ... = \pi_8 = \frac{1}{8}$. The results are summarized in Figure 2 and Table 2. Figure 2 shows the comparison between the eight true change-point pattern with its MAP (maximum a *posteriori*) estimate, which are $(120, 210, 460, 530, 615, 710, 800, 950)$ and $(120, 211, 460, 531, 610, 710, 801, 939)$ respectively. The two patterns match very well except the last point. A detailed exploration of the simulated dataset gives a strong support to the MAP estimate. The last ten observations of the second last block have a larger mean value than the expected and thus, they have been grouped into the last block. The MAP estimates also achieves larger log-posterior probability than that of the true pattern, which is $5305.57 > 5300.24$.

Second, for comparison, SAMC and RJMCMC were also applied to this example. Each algorithm was run 50 times independently. The results are summarized in Table 2. SAMC employes the same setting as Pop-SAMC except two parameters, $T_0 = 100$, Iterations=10^6. RJMCMC employes the same transition proposals as those used by Pop-SAMC and SAMC and performs 10^6 iterations in each run. Under these settings, for a single run, each of the three algorithms performs exactly the same number of energy evaluations with the same transition proposals. Therefore, the comparison made in Table 2 are fair to each of the algorithm. This is evidenced by the CPU times cost by a single run of each method reported in the Table.

The comparison shows that Pop-SAMC works best among these three methods, with smallest standard error achieved in estimating the posterior probability $P(\mathcal{X}_k|\boldsymbol{y})$. As expected, RJMCMC works better than SAMC in this example. Because for this example, the model space is quite simple, it only contains one mode with comparable probabilities. As point out earlier in this section, under such situation, SAMC may not be better than RJMCMC. However, Pop-SAMC does. Although, Pop-SAMC

Table 2: Comparison of the estimated posterior distribution $P(\mathcal{X}_k|\boldsymbol{y})$ for the change–point identification example and its standard deviation (SD). CPU: the CPU time (in seconds) cost by a single run of the corresponding algorithm on a Intel Core 2 Duo 3.0 GHz computer.

	Pop-SAMC 10%Cr		Pop-SAMC		SAMC		RJMCMC	
k	prob(%)	SD	prob(%)	SD	prob(%)	SD	prob(%)	SD
7	0.1029	0.0014	0.1009	0.0018	0.0949	0.0026	0.0998	0.0052
8	55.5077	0.2272	55.6082	0.2698	54.5699	0.6833	55.0832	0.3261
9	33.3677	0.1364	33.2264	0.1693	33.5970	0.4432	33.5365	0.1794
10	9.2642	0.0873	9.3098	0.1010	9.8146	0.2910	9.4942	0.1548
11	1.5646	0.0253	1.5633	0.0233	1.7117	0.0778	1.5884	0.0547
12	0.1767	0.0037	0.1756	0.0031	0.1943	0.0113	0.1813	0.0108
13	0.0150	0.0004	0.0149	0.0003	0.0165	0.0011	0.0153	0.0013
14	0.0010	0.0000	0.0010	0.0000	0.0012	0.0001	0.0012	0.0002
CPU(s)	16.1		16.2		16.2		15.2	

is essentially an important sampling method as SAMC, its improved self-adjusting mechanism makes it much more efficient than SAMC. Amazingly, this improvement in its ability to learn from past samples enables Pop-SAMC to even conquer RJMCMC for those problems in which RJMCMC succeeds.

It is worth pointing out that, both Pop-SAMC and SAMC beat RJMCMC in the low probability model spaces, e.g. $k = 7, 13, 14$, even though SAMC is worse than RJMCMC overall. The reason is the following. Essentially, RJMCMC does not have self-adjusting ability, it samples each model in a frequency proportional to its probability. In contrast, due to their self-adjusting mechanism, Pop-SAMC and SAMC sample equally from each model space, they work well for the low probability model space as well as for the high probability part.

Finally, in order to further increase Pop-SAMC's efficiency and fully use the information among the population, we incorporate crossover operator into the com-

putation with 10% crossover rate and keep all other settings intact. The algorithm was run 50 times independently, and the results are also included in Table 2 for comparison. Unsurprisingly, by using information two dimensionally, Pop-SAMC works more efficiently.

2.4.2 A Large-p Regression Model Selection Example

To have a further assessment of the performance of the Pop-SAMC in Bayesian model selection problems, we consider a linear regression variable selection example, in which the number of observations n is much less than the number of potential predictors p.

The linear regression model with a fix number of potential predictors $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_p\}$ usually takes the form

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N_n(\mathbf{0}, \mathbf{I_n}/\tau) \tag{2.19}$$

where $\boldsymbol{y} = (y_1, y_2, ..., y_n)'$ is the response vector, $\boldsymbol{X} = [\mathbf{1}, \boldsymbol{x}_1, ..., \boldsymbol{x}_p]$ is an $n \times (p+1)$ design matrix, and $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_p)$ is a $(p+1)$-vector of regression coefficients. The problem of interest is to find a subset model M_k of the form

$$\boldsymbol{y} = \boldsymbol{X}_k\boldsymbol{\beta}_k + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N_n(\mathbf{0}, \mathbf{I_n}/\tau) \tag{2.20}$$

which is "best" under some criterion, where $0 \leq k \leq p$, $\boldsymbol{X}_k = [\mathbf{1}, \boldsymbol{x}_1^*, ..., \boldsymbol{x}_k^*]$, $\boldsymbol{x}_1^*, ..., \boldsymbol{x}_k^*$ are the selected predictors, and $\boldsymbol{\beta}_k = (\beta_0^*, \beta_1^*, ..., \beta_k^*)$ is the vector of regression coefficients of the subset model. For model M_k, the likelihood function is

$$L_k(\boldsymbol{y}|\boldsymbol{X}_k, \boldsymbol{\beta}_k, \tau, M_k) = \Big(\frac{\tau}{2\pi}\Big)^{n/2} exp\Big\{ -\frac{\tau}{2}\big(\boldsymbol{y} - \boldsymbol{X}_k\boldsymbol{\beta}_k\big)'\big(\boldsymbol{y} - \boldsymbol{X}_k\boldsymbol{\beta}_k\big)\Big\}. \tag{2.21}$$

The prior distributions for each parameters are assigned as follows. We first

assume τ and $\boldsymbol{\beta}_k$ are subject to the g priors (Zellner,1986),

$$P(\tau) \propto \frac{1}{\tau}, \quad \boldsymbol{\beta}_k|\tau, M_k \sim N\Big(\mathbf{0}, \frac{g}{\tau}(\boldsymbol{X}'_k\boldsymbol{X}_k)^{-1}\Big), \tag{2.22}$$

where g is a hyperparameter. We further assume that all the p predictors are linearly independent, and each has the same prior probability q to be included in the model. Therefore, the prior probability imposed on the mode M_k is

$$P(M_k) = q^k(1-q)^{p-k}, \tag{2.23}$$

with q being subject to the uniform distribution $Unif[0,1]$.

Collecting the likelihood and prior distributions, we get the posterior distribution,

$$P(M_k, \tau, \boldsymbol{\beta}_k, q|\boldsymbol{y}) \propto L_k(\boldsymbol{y}|\boldsymbol{X}_k, \boldsymbol{\beta}_k, \tau, M_k)P(\tau)P(\boldsymbol{\beta}_k|\boldsymbol{X}_k, \tau, g)P(M_k|q)P(q) \tag{2.24}$$

Integrating out τ, $\boldsymbol{\beta}_k$ and q from (2.24) and taking the logarithm, we get the log-posterior of model M_k (up to an additive constant),

$$\begin{aligned} \log P(M_k|\boldsymbol{y}) &= \log\Gamma(k+1) + \log\Gamma(p-k+1) - \frac{k}{2}log(1+g) \\ &\quad -\frac{n}{2}\log\Big[\boldsymbol{y}'\boldsymbol{y} - \frac{g}{1+g}\boldsymbol{y}'\boldsymbol{X}_k(\boldsymbol{X}'_k\boldsymbol{X}_k)^{-1}\boldsymbol{X}'_k\boldsymbol{y}\Big] \end{aligned} \tag{2.25}$$

where g is specified by the user, which reflects their prior knowledge on the model space. Typically, large g concentrates the prior on parsimonious models with a few large coefficients, and small g tends to concentrate the prior on saturated models with small coefficients (George and Foster 2000). The evaluation of the posterior distribution involves inverting a $(k+1)\times(k+1)$ matrix, which can be calculated in a recursive manner using the matrix inversion in block form, and this will save the computation cost tremendously.

A. Simulation Study

The small n large p example is modified from some examples studied in Fernández et al. (2001) and Cai et al. (2009). The dataset is generated as follows.

Let $\boldsymbol{z}_i \sim N_{150}(\mathbf{0}, \mathbf{I})$ for $i = 1, ..., 600$ and define

$$\begin{aligned}
\boldsymbol{x}_i &= \boldsymbol{z}_i, \quad i = 1, ..., 30; \\
\boldsymbol{x}_i &= \boldsymbol{z}_i + 0.3\boldsymbol{z}_{i-30} + 0.5\boldsymbol{z}_{i-29} - 0.7\boldsymbol{z}_{i-28} + 0.9\boldsymbol{z}_{i-27} + 1.1\boldsymbol{z}_{i-26} + 0.2\boldsymbol{z}_{i+80} \\
&\quad -0.4\boldsymbol{z}_{i+180} + 0.6\boldsymbol{z}_{i+280} - 0.8\boldsymbol{z}_{i+380} + \boldsymbol{z}_{i+480}, \quad i = 31, ..., 40; \\
\boldsymbol{x}_i &= \boldsymbol{z}_i + 0.3\boldsymbol{z}_{i-30} + 0.5\boldsymbol{z}_{i-29} + 0.7\boldsymbol{z}_{i-28} - 0.9\boldsymbol{z}_{i-27} + 1.1\boldsymbol{z}_{i-26} + 0.2\boldsymbol{z}_{i+80} \\
&\quad +0.4\boldsymbol{z}_{i+180} - 0.6\boldsymbol{z}_{i+280} + 0.8\boldsymbol{z}_{i+380} - \boldsymbol{z}_{i+480}, \quad i = 41, ..., 50; \\
\boldsymbol{x}_i &= \boldsymbol{z}_i + 0.3\boldsymbol{z}_{i-30} - 0.5\boldsymbol{z}_{i-29} + 0.7\boldsymbol{z}_{i-28} + 0.9\boldsymbol{z}_{i-27} + 1.1\boldsymbol{z}_{i-26} - 0.2\boldsymbol{z}_{i+80} \\
&\quad +0.4\boldsymbol{z}_{i+180} + 0.6\boldsymbol{z}_{i+280} + 0.8\boldsymbol{z}_{i+380} + \boldsymbol{z}_{i+480}, \quad i = 51, ..., 60; \\
\boldsymbol{x}_i &= \boldsymbol{z}_i, \quad i = 61, ..., 600;
\end{aligned} \tag{2.26}$$

The response variable is defined as

$$\boldsymbol{y} = \mathbf{1} + \sum_{i=31}^{60} \boldsymbol{x}_i + \boldsymbol{\epsilon}, \tag{2.27}$$

where $\boldsymbol{\epsilon} \sim N_{150}(\mathbf{0}, 4\mathbf{I})$ and is independent of other predictor variables.

For this example, we set the hyperparameter $g = max(n, p^2)$, the so called *benchmark prior* recommended by Fernández et al. (2001). Given the full posterior distribution, in applying Pop-SAMC to this example, we follow the same fashion as in the change-point identification example. We first partition the sample space according to the model index k. Let $E_k = \mathcal{X}_k$, the model space with k selected variables, $M_k \in \mathcal{X}_k$, and $\psi(\cdot) \propto P(M_k|\boldsymbol{y}, \eta_\tau)$. It follows from (2.9) that $\hat{w_i}^{(t)}/\hat{w_j}^{(t)} = e^{\theta_{ti} - \theta_{tj}}$ forms a consistent estimator for the Bayes factor $P(\mathcal{X}_i|\boldsymbol{y})/P(\mathcal{X}_j|\boldsymbol{y})$. We restrict the model space

to be $k_{min} \leq i, j \leq k_{max}$. After a pilot run, we set $k_{min} = 10$ and $k_{max} = 40$.

Table 3: Comparison of the estimated posterior distribution $P(\mathcal{X}_k|\boldsymbol{y})$ for the simulated large p linear regression example and its standard deviation (SD).

	Pop-SAMC 10%Cr		Pop-SAMC		SAMC		RJMCMC	
k	prob(%)	SD	prob(%)	SD	prob(%)	SD	prob(%)	SD
10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.2125	0.3689
11	0.0000	0.0000	0.0000	0.0000	0.0012	0.0035	0.1345	0.2223
12	0.0000	0.0000	0.0000	0.0000	0.0052	0.0135	0.1515	0.1636
13	0.0001	0.0002	0.0001	0.0002	2.0299	6.5153	45.6895	8.4369
14	0.0001	0.0002	0.0001	0.0001	1.4000	4.4203	38.0850	6.8908
15	0.0000	0.0000	0.0000	0.0001	0.2382	0.7520	7.3470	1.2674
16	0.0000	0.0000	0.0000	0.0000	0.0245	0.0712	1.0135	0.2736
17	0.0000	0.0000	0.0000	0.0001	0.0082	0.0128	1.5905	5.0314
18	0.0000	0.0000	0.0000	0.0001	0.0135	0.0260	1.1135	1.5058
19	0.0000	0.0000	0.0000	0.0002	0.0059	0.0099	0.9490	1.0440
20	0.0000	0.0000	0.0000	0.0001	0.0021	0.0040	0.3740	0.5632
21	0.0000	0.0000	0.0000	0.0000	0.0005	0.0011	0.0930	0.1583
22	0.0000	0.0000	0.0000	0.0000	0.0001	0.0003	0.0190	0.0361
23	0.0000	0.0000	0.0000	0.0000	0.0001	0.0001	0.0035	0.0114
24	0.0001	0.0002	0.0001	0.0003	0.0080	0.0177	0.0005	0.0022
25	0.0000	0.0001	0.0001	0.0001	0.0029	0.0068	0.0005	0.0022
26	0.0001	0.0001	0.0001	0.0000	0.0008	0.0016	0.0000	0.0000
27	83.5210	1.0919	84.1339	1.4828	81.0277	10.0109	2.6935	12.0457
28	14.9215	0.9560	14.3358	1.2640	13.7389	2.2726	0.4760	2.1287
29	1.4415	0.1639	1.4189	0.2248	1.3769	0.2886	0.0460	0.2057
30	0.1080	0.0211	0.1037	0.0188	0.1077	0.0272	0.0040	0.0179
31	0.0071	0.0018	0.0066	0.0014	0.0071	0.0022	0.0000	0.0000
32	0.0004	0.0002	0.0004	0.0001	0.0004	0.0002	0.0000	0.0000

Again, we applied four algorithms to this example and compare their efficiency. Each algorithm was run for this example 20 times independently. First, Pop-SAMC was run for this example with the following setting: $N = 20$, $T_0 = 400$, Iterations=3×10^5 and $\pi_1 = ... = \pi_{31} = \frac{1}{31}$. Then, we include crossover operator into Pop-SAMC. The modified algorithm was run for this example with 10% crossover rate while keeping all other settings intact. Finally SAMC and RJMCMC were applied to this

example respectively. SAMC employes the same setting as Pop-SAMC except one parameter, Iterations=6×10^6. RJMCMC also performs 6×10^6 iterations in each run, with the first 50000 iterations as warming. For all the four algorithms, the same transition proposal is used, and for a single run, each of them performs exactly the same number of energy evaluations. Therefore, the comparisons are fair to each of the algorithm. The computational results are summarized in Table 3, those subregions with zero probability for all the algorithms are not listed.

The comparison shows that the order of the four algorithm in terms of efficiency for this example is Pop-SAMC with 10% crossover > Pop-SAMC > SAMC > RJMCMC. This order is consistent with that in the change-point identification example, except SAMC beats RJMCMC this time. In fact, RJMCMC failed in this example, it chose the model M_{13} instead of the true model M_{27}. This is because there are two modes in the model space, which are well separated. For RJMCMC, it does not have the self-adjust ability, which makes it easilly get trapped into a local mode. However, all the other three algorithms have a self-adjusting mechanishm, which enables them to get out of local trap and explore the whole sample space quickly. The efficiency improvement for Pop-SAMC based algorithms over SAMC is also significant, especially at the true mode and low probability model space, e.g. $k = 11 \sim 26$.

We further check the estimates of the marginal inclusion probabilities produced by Pop-SAMC and RJMCMC in a single run, which is shown in Figure 3. From this plot, we may tell that all the 27 variables selected by Pop-SAMC are in the true variables rang from 31 to 60. Due to the correlation among the variable set, variable 32, 33 and 34 were not selected. On the other hand, the variables selected by RJMCMC are also belong to the true variable set, but it only found 13 out of the 27.

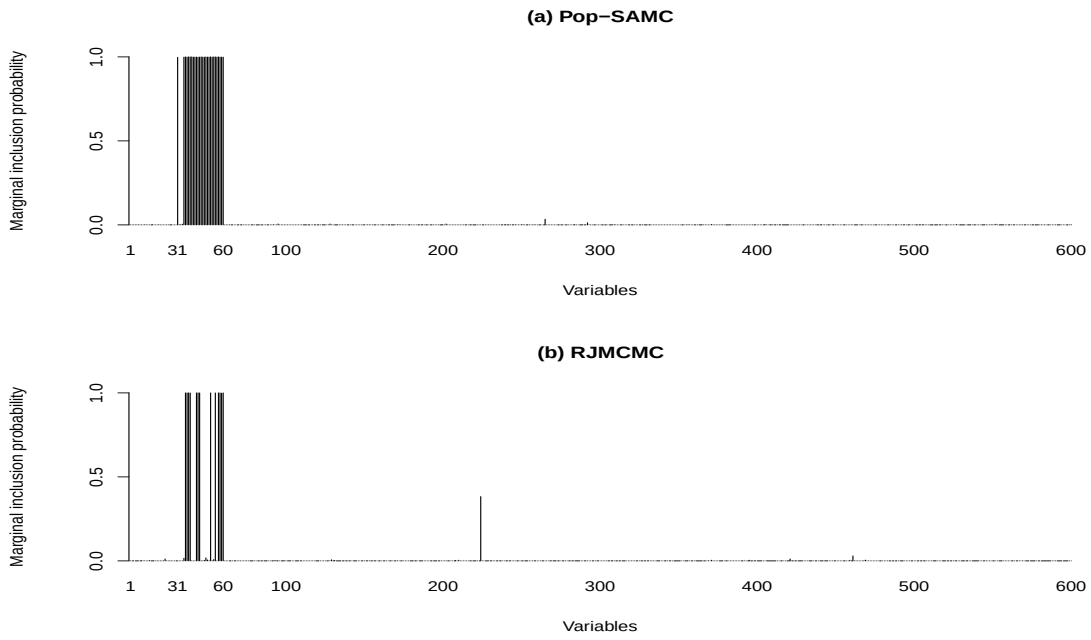


Figure 3. Estimates of marginal inclusion probabilities produced by (a)Pop-SAMC and (b)RJMCMC in a single run.

B. A Real Data Analysis

The dataset we studied here was generated by Lan et al. (2006). As described in Zhang et al. (2009), the experiment concerns the genetic basis for differences between two inbred mouse populations (B6 and BTBR). Based on their inhouse selective phenotyping algorithm, 60 (B6×BTBR) F_2-*ob*/*ob* mice (29 males and 31 females) were selected. A total of 60 arrays were used to monitor the expression levels of $22,690$ genes. Some physiological phenotypes were also measured by quantitative realtime RT-PCR, e.g. the numbers of stearoyl-CoA desaturase 1 (SCD1). The raw data are available in GEO http://www.ncbi.nlm.nih.gov/geo; (accession number GSE3330).

We treat the phenotypic value (SCD1) as the dependent variable, and the expression levels of genes as predictors. The value of SCD1 was first adjusted to remove

the possible gender effects, then its correlation to each gene is calculated. We ordered the genes according to the correlation from high to low, and took the first 1000 genes as the potential predictors.

Three algorithms were applied to this dataset, Pop-SAMC, SAMC and RJMCMC. Each algorithm was run for this example 20 times independently. Follow the procedure in the simulation study, the sample space was partitioned according to the model index, and after a few pilot runs, we restricted the model space to be $1 \leq k \leq 10$. In the pilot runs, we also found, with such a small number of observations, $n = 60$, setting $g = p^2$ made the penalty to complex models so strong such that the mode was pushed around 1. In order to consider more potential models, we relaxed the penalty in priors and set $g = p = 1000$. The parameters for each algorithm were set as follows. For Pop-SAMC, $N = 10$, $T_0 = 20$, Iterations=6×10^4, and $\pi_1 = ... = \pi_{10} = \frac{1}{10}$; SAMC employes the same setting as Pop-SAMC except $T_0 = 50$ and Iteration=6×10^5; RJMCMC also performs 6×10^5 iterations with the first 20000 iterations as warming. As in the simulation study, the same transition proposal is used for all the algorithm, and for a single run, each of them performs exactly the same number of energy evaluations. Therefor, the comparison is fair. The computational results are summarized in Table 4.

The results is clear, Pop-SAMC works best among the three methods with the smallest standard error achieved in estimating the posterior probability of potential models. Since there is only one mode in the model space for this dataset, as an important sampling algorithm, SAMC can not beat RJMCMC overall, it only won the battle in the low probability model space. However, with the improved self-adjusting mechanism, Pop-SAMC have conquered RJMCMC in the whole model space.

Table 4: Comparison of the estimated posterior distribution $P(\mathcal{X}_k|\boldsymbol{y})$ for the real large p linear regression example and its standard deviation (SD). CPU: the CPU time (in seconds) cost by a single run of the corresponding algorithm on a Intel Core 2 Duo 3.0 GHz computer.

	Pop-SAMC		SAMC		RJMCMC	
k	prob(%)	SD	prob(%)	SD	prob(%)	SD
1	0.2792	0.1049	0.1991	0.0948	0.1810	0.0907
2	16.0970	1.9719	14.6317	4.8154	13.6790	2.5625
3	20.2100	3.2758	18.2059	4.2074	18.0265	3.3172
4	35.5714	4.0382	36.0330	8.9530	39.0120	5.0084
5	16.9414	1.1698	17.0727	2.1761	17.8005	1.7256
6	6.7641	0.8092	7.0985	1.9672	6.5390	0.9450
7	2.5384	0.4956	3.3053	1.9966	2.5420	1.4040
8	0.9915	0.2542	1.9386	2.0487	1.3045	2.6034
9	0.4478	0.1443	1.1381	1.4715	0.7115	2.1479
10	0.1591	0.0853	0.3770	0.4761	0.2045	0.7313

2.5 Discussion

In this work we have proposed a population SAMC algorithm and show that, in both theory and numerical examples, it can be more efficient than the single chain SAMC and RJMCMC algorithm.

Our theory on the convergence of population SAMC algorithms can also be extended to the multiple SAMC algorithm (Younes 1999; Liang 2009) for which a single chain is run but with multiple samples being generated at each iteration. Our

theory is consistent with the numerical results reported in Liang (2009): The multiple SAMC algorithm can be more efficient than the single chain SAMC algorithm. This should be of interest for practical applications.

The population SAMC algorithm works on a population of inhomogeneous Markov chains. Its population setting provides a basis for including more global advanced MCMC operators other than crossover from genetic algorithm, such as the snooker operator (Gilks et al., 1994), and the gradient operator (Liu et al., 2000), into simulations. This can potentially improve further the convergence of the algorithm.

Lastly, we want to point out the population size should be balanced with the choice of the total number of iterations, as the convergence of the algorithm only occurs as $\gamma_t \to 0$. In our experience, $5 \sim 50$ may be a good range for the population size.

CHAPTER III

CHIP-CHIP DATA ANALYSIS AND BEYOND

3.1 Introduction

3.1.1 Biological Background

A. Protein, DNA and RNA

Proteins, from the Greek proteios, meaning first, are a class of organic compounds which are present in and vital to every living cell.

In structure, proteins are linear polymers built from 20 different amino acids connected by peptide bond. A typical protein contains 200-300 amino acids but some are much smaller (the smallest are often called peptides) and some are much larger (the largest to date is titin, a protein found in skeletal and cardiac muscle; it contains 26,926 amino acids in a single chain!). The linear polymers chain is not a straight line, instead, it forms 3-D structure for different functions. According to their functions, proteins can be divided into different categories, such as, transport proteins, transcription factors, antibodies etc.

Proteins are made from DNA, which serves as the templates for protein synthesis. DNA is the abbreviation for deoxyribonucleic acid, which is the genetic material present in the cells of humans and almost all other living organisms. The main role of DNA molecules is the long-term storage of information. DNA is often compared to a set of blueprints, since it contains the instructions needed to construct other components of cells, such as proteins and RNA molecules. This functionality makes it as the fundamental building block for an individual's entire genetic makeup. Nearly every cell in a persons body has the same DNA. Most DNA is located in the cell nucleus (where it is called nuclear DNA), while a small amount of DNA can also be

found in the mitochondria (where it is called mitochondrial DNA or mtDNA).

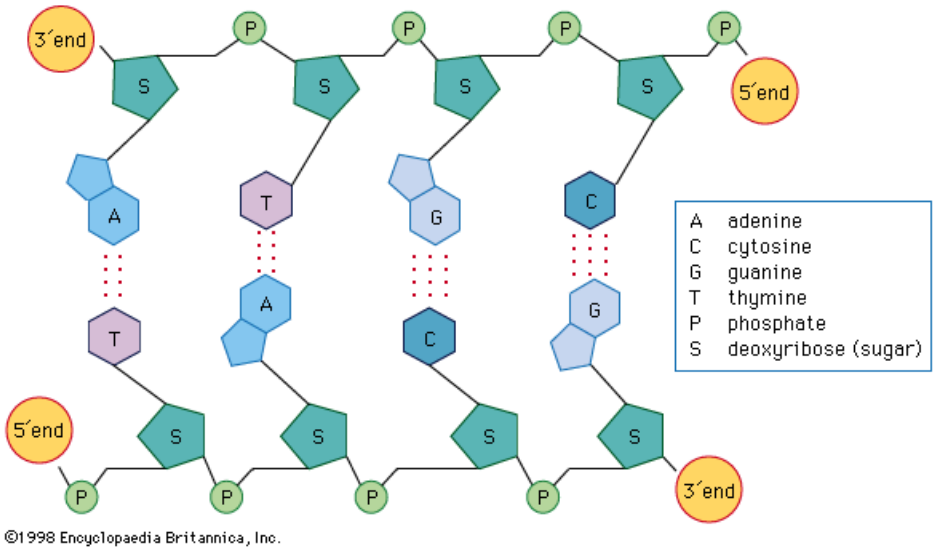


Figure 4. DNA structure, showing the nucleotide bases Adenine(A), Guanine(G), Cytosine(C) and Thymine(T) linked to a backbone of alternating phosphate (P) and deoxyribose sugar (S) groups. Two sugar-phosphate chains are paired through hydrogen bonds between A and T and between G and C, thus forming the twin-stranded double helix of the DNA molecule. (Encyclopaedia Britannica, Inc. 1998)

As shown in Figure 4, chemically, DNA is a linear polymer of simple units called *nucleotides*, with a backbone made of sugars (S) and phosphate (P) groups joined by covalent bonds. Attached to each sugar is one of four types of molecules called *bases*: Adenine(A), Guanine(G), Cytosine(C) and Thymine(T). It is the order (sequence) of these four bases along the backbone that determines each person's genetic characteristics.

DNA does not usually exist as a single molecule, but instead bases pair up with each other, A with T and C with G, to form units called *base pairs*. Each base is also attached to a sugar molecule and a phosphate molecule. Together, a base, sugar,

and phosphate formed a *nucleotide*. Nucleotides are arranged in two long strands that form a spiral called a double helix, which was proposed by James D. Waston and Francis H.C. Crick in 1953. The structure of the double helix is somewhat like a ladder, with the base pairs forming the ladders rungs and the sugar and phosphate molecules forming the vertical sidepieces of the ladder. The total length of all DNA

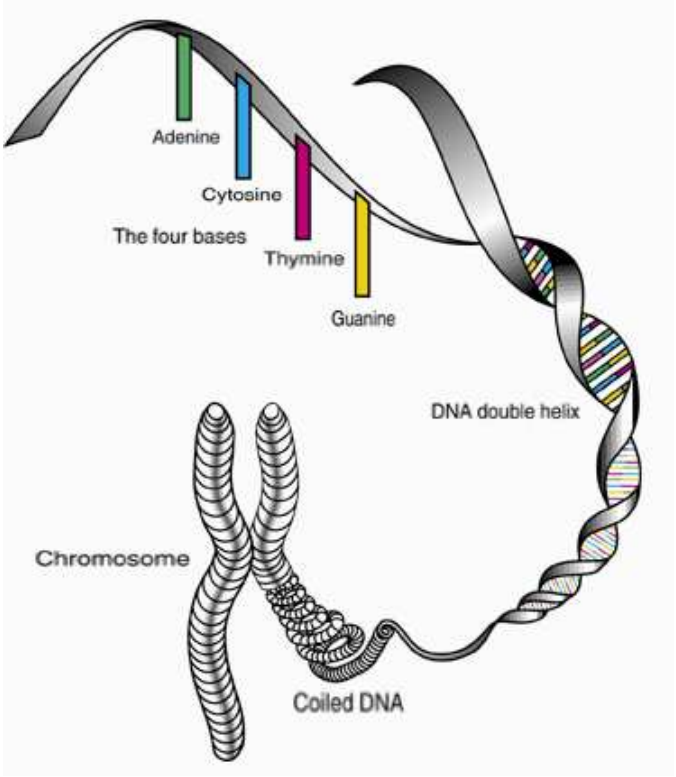


Figure 5. DNA-Chromosome structure (University of California Lawrence Livermore National Laboratory and the Department of Energy)

molecules in one human cell is about 2 meters. But they need to be fitted to the nucleus of the cell, which is of size around 2 micrometers (2×10^{-6} meter). Therefore, the double-helix shaped DNA should be further packed together to form *chromosome* (see Figure 5). All chromosomes in one cell of a certain species compose the genome of this species. All genetic information used to code various proteins is contained in

the DNA molecules in the genome, and this is why DNA is so important for living organisms. But to make protein encoded by DNA, a transient product, RNA, is needed.

RNA is very similar to DNA, but differs in a few important structural details: in the cell RNA is usually single stranded, while DNA is usually double stranded. RNA nucleotides contain ribose, while DNA contains deoxyribose (a type of ribose that lacks one oxygen atom), and in RNA the nucleotide Uracil(U) substitutes for Thymine(T), which is present in DNA.

Three major types of RNA are crucial to protein synthesis: messenger RNA (mRNA), transport RNA (tRNA), and ribosomal RNA (rRNA). Simply speaking, mRNA copy genetic information from DNA and provides blueprint, tRNA works to bring amino acids together, and rRNA connects them to proteins. The relationship

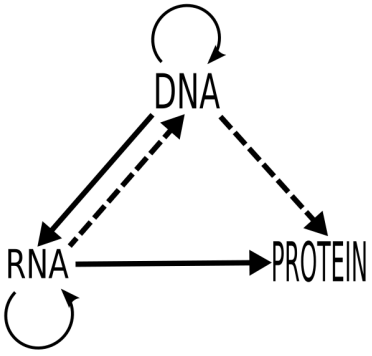


Figure 6. Central dogma of molecular biology. Solid arrows represent probable transfers, dotted arrows possible transfers.

among DNA, RNA and protein is well explained by the so called *central dogma* of molecular biology (see Figure 6). DNA contains the complete genetic information that defines the structure and function of an organism. Proteins are formed using the genetic code of the DNA.

B. Genes and Transcription Factor

Although DNA contains the complete genetic information, not every DNA segment in the genome is used to code proteins and RNAs, only some regions of genomic sequence corresponding to such information. These regions are called *gene*.

Genes are locatable regions of DNA sequences that are essential for the synthesis of functional proteins or RNAs, they are the working subunits of DNA. A gene generally consist of two parts: *regulatory regions* that control the transcription of the gene and *transcribed regions* which store the genetic information. The regulatory regions play important roles in control gene expression and thus protein expression. According to different functions, regulatory regions can be divided into three major categories (see Figure 7): promoters, which can activate transcription when bound the pre-initiation complex; enhancers, which can accelerate the transcription speed when bound by co-activator complex; repressors, which can block the transcription when bound by co-repressor complex.

Figure 7. Regulatory regions in a DNA sequence: promoters (red regions), enhancers (green regions) and repressors (blue regions).

As state in central dogma of molecular biology, in the transcription step, a DNA segment that constitutes a gene is read and transcribed into a single stranded sequence of RNA. This process is regulated by some special proteins, named *transcription factors*. To initiate transcription, at the first step, transcription factors recognize the regulatory regions and bind to them, then they controls the transfer of genetic information from DNA to RNA. Without transcription factors, the creation of new RNA from DNA cannot occur. Only after certain transcription factors are bound

to regulatory regions, does the RNA polymerase bind to it. Thus, the interaction between transcription factors and their DNA binding sites are the key to determining where the DNA chain becomes "unzipped", creating a single strand to which RNA can be bound while it's being built. However, most of the DNA binding sites of transcription factors along human chromosomes are unknown.

Identify the transcription factors binding sites on DNA sequences is the first step to understand the genomes encodes information that specifies when and where a gene will be expressed. This comes to the motivation of this project.

3.1.2 Microarray Technology

Microarray analysis permits scientists to detect and measure thousands of genes in a small sample simultaneously under different experiment conditions.

As described in central dogma of molecular biology, A gene (a DNA segment) is first read and transcribed into mRNA before being translated into a protein. So far, microarray analysis focus on the transcription level. What it measures is the absolute/relative abundance of the mRNA transcribed from different genes. The microarray is simply a glass slide (shown in Figure 8), which composed of millions of spots, each of which corresponds to one gene. Within one spot, there are thousands of identical probes. Each probe is just a single-strand DNA segment rooted to the slide and used to detect existence of certain DNA segments. The enormous number of probes within each spot is to increase hybridization probability and possibilities.

DNA microarrays work on the principal of *base-pairing*. Base-pairing allows probes to hybridize to targets on the microarray (see Figure 9). Generally, there are two kinds of microarray: *cDNA microarry* and *oligonucleotide microarray*. In cDNA microarray, DNA segments are reverse-transcribed from mRNA that has been yielded from DNA coding sequences. Such DNA segments are called complemen-

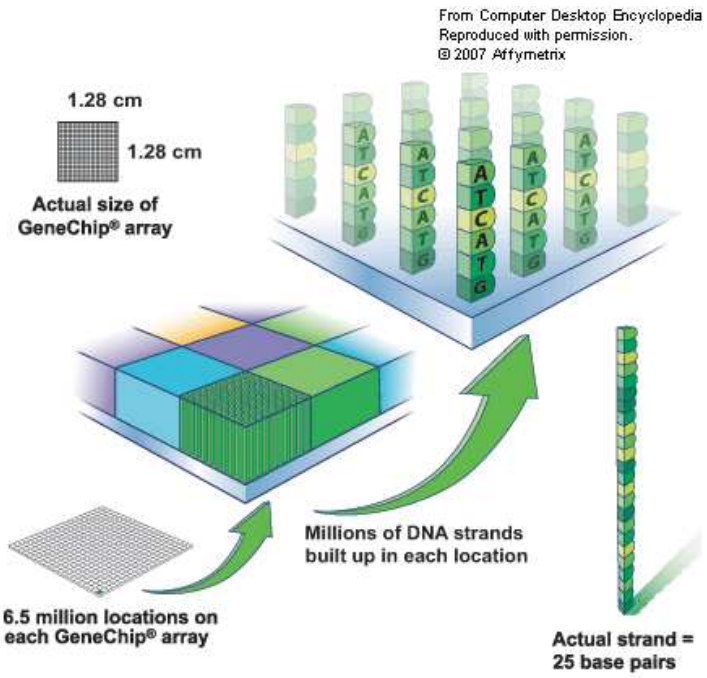


Figure 8. Microarry chip (Image courtesy of Affymetrix).

tary DNA (cDNA), and DNA segments complementary to these cDNAs are used as probes. In oligonucleotide microarray, produced by Affymetrix, instead of using DNA complementary to the cDNA, which are made from the whole mRNA, shorter DNA segments (probes) are used, about 25 nucleotide long. The probes come in pair, perfect-match (PM) and mismatch (MM). For MM the DNA segments are idential to PM probes except that the central (13th position) nucleotide is changed. The mismatch probe measures the degree of cross hybridization, or how much lower the detection signals for noise are.

The ultimate goal of microarray analysis is to determine which genes are turned on and which are turned off in a given cell. To achieve this goal, the messenger RNA molecules present in that cell must be collected first, then these mRNAs are reversed-transcribed to cDNAs and labels by attaching a fluorescent dye. Next, the labeled

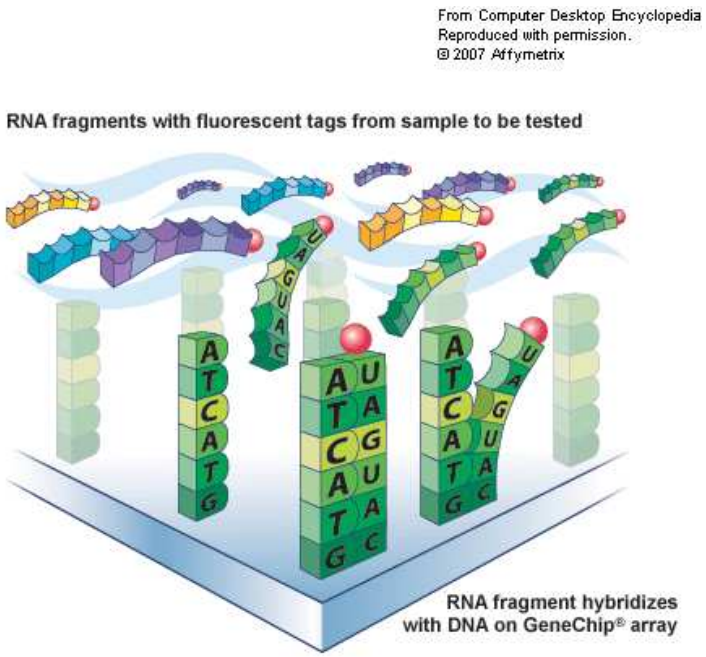


Figure 9. Hybridization between mRNA and probes (Image courtesy of Affymetrix).

cDNAs are hybridized to the probes on the microarray slide. After hybridization, cDNA not bound to probes are washed away, leaving bound ones with their fluorescent tag (see Figure 10). Finally, the slide is scanned twice with red and green laser respectively to measure the signal intensity under different experiment conditions, two images are obtained for further analysis.

If a particular gene is very active, it produces many molecules of mRNA, thus, corresponding cDNA, which hybridize to the DNA on the microarry and generate a very bright fluorescent area. Genes that are somewhat active produce fewer mRNAs, which results in dimmer fluorescent spots. If there is no fluorescence, none of the corresponding cDNA of the mRNA have hybridized to the DNA, indicating that the gene is inactive. By this technique, to exam the activity of various genes is possible.

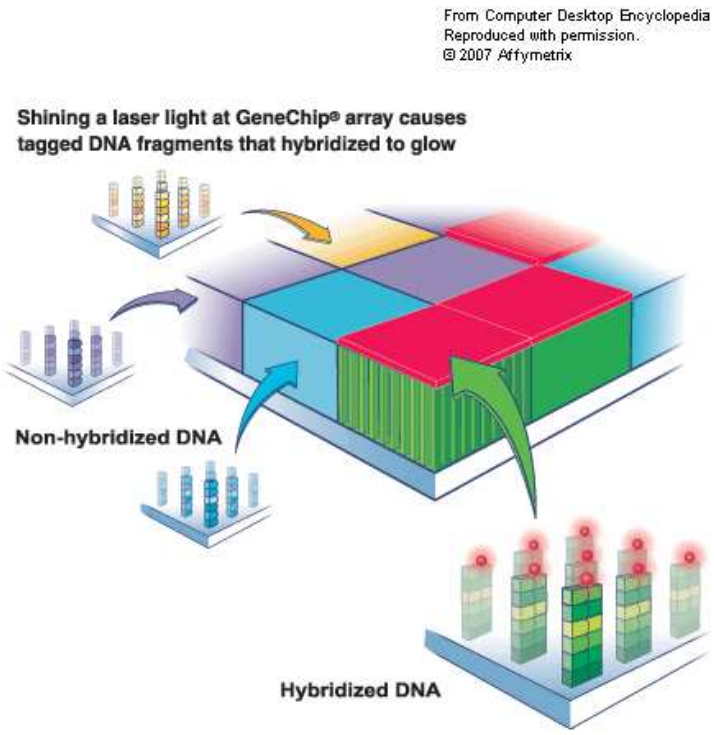


Figure 10. Hybridized DNA (Image courtesy of Affymetrix).

3.1.3 ChIP-chip Technology

ChIP-on-chip (also known as ChIP-chip) is a technique that combines chromatin immunoprecipitation (ChIP) with microarray technology (chip). "ChIP" is a method for isolating DNA fragments that are bound by specific proteins (for example, transcription factors). "chip" refers to DNA microarray technology, which are used for measuring the concentration of these DNA fragments. Since the DNA microarray probes can tile the whole genome, the analysis to determine the location of such protein-DNA binding sites is possible on the genome wide scale.

The ChIP-chip experimental procedure can be described as following (see Figure 11):

1. Cross-linking: proteins bind to DNA, for example, transcription factors bound

to their target DNA sequences. DNA and proteins are cross-linked in vivo with formaldehyde, they are linked together by covalent bonds.

2. Sonication: in this step, the DNA sequences is chopped into small fragments. i.e. DNA bound by proteins is sheared by sonication to small fragments, some of which are bound by proteins, and the rest are not.

3. Immunoprecipitation: the purpose of this step is to isolate the DNA fragments bound by proteins. Protein-DNA complexes are immunoprecipitated using an antibody that targets the protein of interest. Antibodies recognize and bind to the target proteins and cause the complex (cross linked DNA-protein) to precipitate.

4. Amplification: Cross-linking between DNA and protein is reversed and bound DNA is released. However, The amount of DNA obtained from immunoprecipitation is very low, Amplification step is necessary. With standard biochemical technique, copies of the existing DNA fragments are made. After amplification, DNA fragments are labeled with a fluorescent tag such as Cy5 (red), meanwhile, a sample of DNA which is not enriched by the above immunoprecipitation step is also amplified and labelled with another fluorescent dye, such as Cy3 (green).

5. Hybridisation: the obtained DNA fragments from amplification are poured over the surface of the DNA microarray. Whenever a labeled fragment "finds" a complementary fragment on the array, they will hybridize and form again a double-stranded DNA fragment. For Affymetrix developed high-density oligonucleotide tiling array technology, the DNA pools from different group (treatment/control) are hybridized to separate microarrays. PM and MM values are output for each group, which allow comparison among different experiment conditions. By com-

paring the hybridization signals generated by an immunoprecipitated sample versus a non-specific antibody control or input DNA control, the regions of chromatin-protein interaction can be identified.

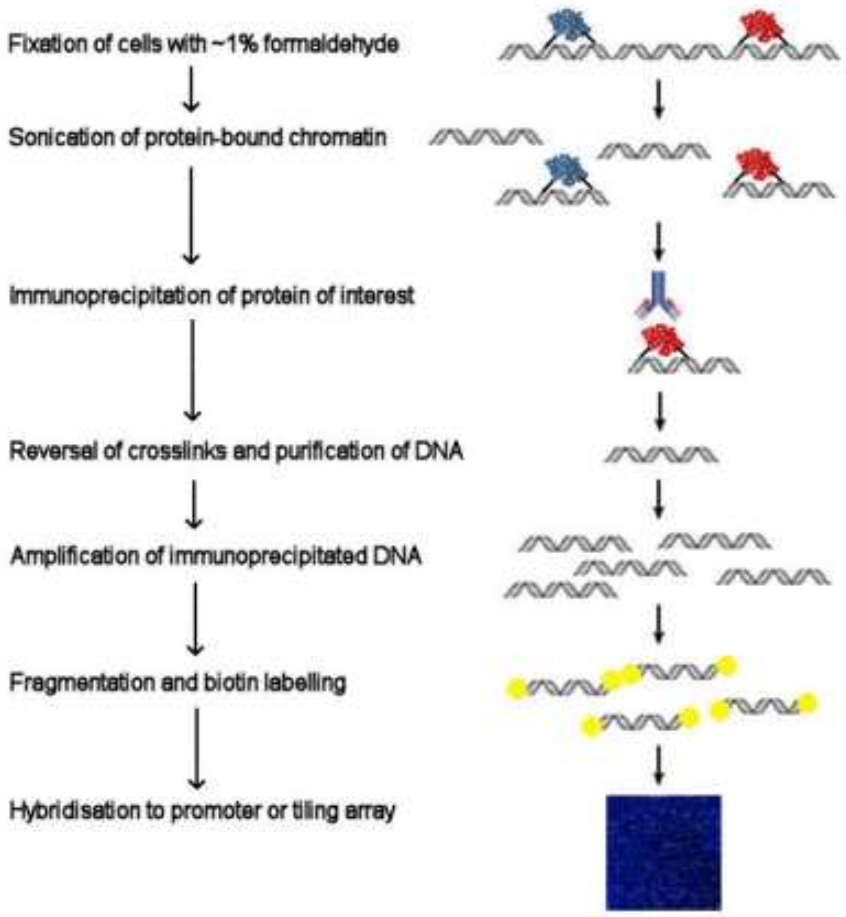


Figure 11. Principle of a ChIP on chip experiment.

Data from ChIP-chip experiments encompass DNA-protein interaction measurements on millions of short oligonucleotides (probes) which often tile one or several chromosomes or even the whole genome. Along the genome, the ChIP-chip data can be viewed in the form of a one-dimension signals, among which a peak generally corresponds to a protein binding site. Therefore, to locate the proteins binding sites is

equivalent to detect the statistical significant peaks in the signals. Figure 12 shows an example of the global appearance of the raw data of PM intensity under treatment condition. The x-axis denotes the genomic position of each probe and the y-axis shows the signal intensity. The data analysis consists of two steps: (1) identifying the bound regions where DNA and the protein are cross-linked in the experiments; and (2) identifying the binding sites through sequence analyses of the bound regions. The goal of this chapter is to develop effective methods for the first step analysis.

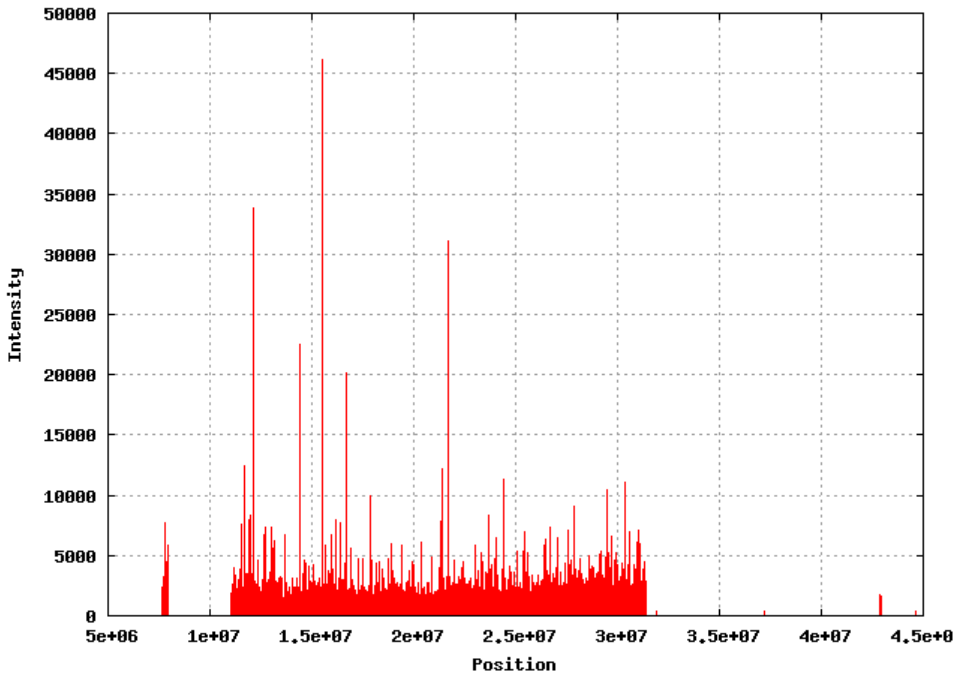


Figure 12. An overview of the PM raw data under treatment condition along the genomic position.

3.2 Bayesian Latent Model *

3.2.1 Literature Review

Analysis of the ChIP-chip data is very challenging, due to the large amount of probes and the small number of replicates. The existing methods in the literature can be roughly grouped into three categories, the sliding window methods (Cawley et al., 2006; Bertone et al., 2004; Ji and Wong, 2005; Keles et al., 2006), the hidden Markov Model(HMM) methods (Li et al., 2005; Ji and Wong, 2005; Munch et al., 2006; Humburg et al., 2008), and the Bayesian methods (Qi et al., 2006; Keles, 2007; Gottardo et al., 2008). Other methods have been suggested, e.g., by Zheng et al. (2007), Huber et al. (2006) and Reiss et al. (2008), but are less common.

The sliding window methods are to test a hypothesis for each probe using the information from the probes within a certain genomic distance sliding window, and then try to correct for the multiple hypothesis tests. The test statistics used are varied. Cawley et al. (2004) used Wilcoxon's rank sum test, Keles et al. (2006) used a scan statistic which is the average of t-statistics within the sliding window, and Ji and Wong (2005) used a scan statistic which is the average of empirical Bayesian t-statistics within the sliding window. Since each test uses information from neighboring probes, the tests are not independent, rendering a difficult adjustment in the multiple hypothesis testing step. We note that the power of the sliding window tests is usually low, especially for the tests for the regions where the probe density is low. This is because there will be only very limited neighboring information available for those tests. Since, in the ChIP-chip experiments, the DNA samples hybridized to the

* Part of this section is reprinted with permission from "Bayesian modeling of ChIP-chip data using latent variables" by Mingqi Wu, Faming Liang and Yanan Tian, 2009. *BMC Bioinformatics* **10**:352.

microarrays are prepared by PCR, which is known to perform independently of the form of DNA, the far probes should have similar intensity patterns as long as they are of similar positions to their nearest bound regions. This provides a basis for us to devise powerful methods that make use of information from all probes.

The HMM methods have the potential to make use of all probe information, where the model parameters are estimated using all available data. However, in most of the existing implementations of HMMs, the model parameters are estimated in an *ad hoc* way. For example, Li et al. (2005) estimated the model parameters using previous results on Affymetrix SNPs arrays. An exception is tileHMM (Humburg et al., 2008), where the model parameters are estimated using the Baum-Welch and Viterbi training algorithms (Rabiner, 1989). However, it is known that these algorithms are prone to get trapped in local optimal solutions, rendering the estimates suboptimal to the problem.

Bayesian methods have also the potential to make use of all probe information. Like the HMM methods, the Bayesian methods estimate the model parameters using all available data. However, these methods usually require multiple replicates or some extra experimental information to parameterize the model. For example, the joint binding deconvolution model (Qi et al., 2006) requires one to know the DNA fragment lengths, measured separately for each sample via extrophoretic analysis; and the hierarchical gamma mixture model(HGMM) (Keles, 2007) requires one to first divide the data into genomic regions containing at most one bound region, but such information is, in general, unavailable. Using BAC (Gottardo et al., 2008) does not need extra experimental information, but it is extremely slow, roughly 10 hours for a dataset with 300,000 probes on a personal computer. One reason for the slow speed is the use of MCMC simulations.

3.2.2 Bayesian Latent Model

In this section, we propose a Bayesian latent variable model for tiling array data. Our method differs from the existing Bayesian methods, such as the joint binding deconvolution model (Qi et al., 2006), the HGMM (Keles, 2007), and the Bayesian hierarchical model (Gottardo, et al., 2008) in several respects. Firstly, it works on the difference between the averaged treatment and control samples. This enables the use of a simple model for the data, which avoids the probe-specific effect and the sample (control/treatment) effect. As a consequence, this enables an efficient MCMC simulation of the posterior distribution of the model, and also makes the model rather robust to the outliers. Secondly, it models the neighboring dependence of probes by introducing a latent indicator vector. Thirdly, it does not require multiple replicates or extra experimental information. As described below, it can work on a single intensity measurement for the probes. The Bayesian latent model has been successfully applied to several real and ten simulated datasets, with comparisons with some of the existing Bayesian methods, hidden Markov model methods, and sliding window methods. The numerical results indicate that the Bayesian latent model can outperform the others, especially when the data contain outliers. Our method is also computationally efficient; it takes about 30 minutes for a dataset with 300,000 probes on a personal computer.

A. Method and Model

Consider a ChIP-chip experiment with two conditions, treatment and control. Let X_1 and X_2 denote, respectively, the samples measured under the treatment and control conditions. Each sample has m_l, $l = 1, 2$, replicates providing measurements for n genomic locations along a chromosome or the genome. Suppose that these samples have been normalized and log-transformed. In this paper, we summarize the measurements

for each probe by

$$Y_i = \bar{X}_{1i} - \bar{X}_{2i}, \tag{3.1}$$

where $\bar{X}_{li}$ is the intensity measurement of probe i averaged over m_l replicates.

The underlying assumption for the summary statistic in (3.1) is that the intensity measurements for each probes has a variance independent of its genomic position. The rationale is that the DNA samples used in the experiments are prepared by PCR, which is known to perform independently of the form of DNA, and that the amount of the DNA samples provides the main sources for the variation of probe intensities. We note that a similar assumption has also been made in other Bayesian software, e.g., tileHMM. Otherwise, Y_i can be adjusted by its standard error to a shrinkage t-statistic (Opgen-Rhein and Strimmer, 2007) or an empirical Bayes t-statistic (Ji and Wong, 2005), depending on the estimate of the standard error. Note that both the adjustments are toward the constant variance of probes. Even with the adjustments, the Bayesian latent model developed in this paper can still work reasonably well, as the normality assumption approximately holds for the modified t-statistics.

Suppose that the data consists of a total of K bound regions, and that region k consists of n_k $(k = 1, \ldots, K)$ consecutive probes. For convenience, we call all the non-bound regions by region 0 and denote by n_0, the total number of probes contained in all the non-bound regions, although the probes in which may be non-consecutive. Thus, we have $\sum_{k=0}^{K} n_k = n$. Let $\boldsymbol{z} = (z_1, \ldots, z_n)$ be a latent binary vector associated with the probes, where $z_i = 1$ indicates that probe i belongs to a bound region and 0 otherwise. Given $\boldsymbol{z}$, we can re-index $(y_1, \ldots, y_n)$, a realization of $(Y_1, \ldots, Y_n)$, by y_{kj}, $k = 0, \ldots, K$, $j = 1, \ldots, n_k$. Then y_{kj} can be modeled as follows,

$$y_{kj} = \mu_0 + \nu_k + \epsilon_{kj}, \tag{3.2}$$

where μ_0 is the overall mean, which models the difference of sample effects (between the treatment samples and the control samples); $\nu_0 = 0$ and $\nu_k > 0, k = 1, \ldots, K$ accounts for the difference of probe intensities in different bound regions; ϵ_{kj}s are random errors independently and identically distributed as $N(0, \sigma^2)$. In words, model (3.2) assumes that, conditioning on the latent vector $\boldsymbol{z}$, y_{kj}s are mutually independent and also identically distributed within the same bound region. We are aware that for the tiling array data, the probe intensities tend to form a peak around the true binding site. Since, given $\boldsymbol{z}$, the order of probes is meaningless to us, the model (3.2) is appropriate if ignoring the order of the probes. We note that a similar assumption has also been used in the HGMM and HMM methods. Conditioning on $\boldsymbol{z}$, the likelihood of the model can be written as

$$\begin{aligned} f(\boldsymbol{y}|\boldsymbol{z}, \mu_0, \nu_1, \ldots, \nu_K, \sigma^2) &= \prod_{j=1}^{n_0} \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(y_{0j}-\mu_0)^2} \right) \\ &\quad \times \prod_{k=1}^{K} \prod_{j=1}^{n_k} \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(y_{kj}-\mu_0-\nu_k)^2} \right). \end{aligned} \tag{3.3}$$

To conduct a Bayesian analysis for the model, we specify the following prior distributions for the model parameters:

$$\sigma^2 \sim IG(\alpha, \beta), \quad f(\mu_0) \propto 1, \quad \nu_k \sim U(\nu_{min}, \nu_{max}) \tag{3.4}$$

where $IG(\cdot, \cdot)$ denotes an inverse Gamma distribution, $U(\cdot, \cdot)$ denotes a uniform distribution, and $\alpha, \beta, \nu_{min}, \nu_{max}$ are hyperparameters. In this paper, we set $\alpha = \beta = 0.05$, which form a vague prior for σ^2; and set $\nu_{min} = 2s_y$ and $\nu_{max} = \max_i y_i$, where s_y is the sample standard error of y_i. Different values of ν_{min}, e.g., s_y and $1.5s_y$, have also been tried in our simulations, and the results are similar. The sensitivity issue of the Bayesian latent model to the hyperparameters will be further discussed in Section 3. In addition, we assume that the latent vector $\boldsymbol{z}$ follows a truncated Poisson

distribution,

$$f(\boldsymbol{z}|\lambda) = \frac{1}{C}\frac{\lambda^K e^{-\lambda}}{K!}, \quad K \in \{0, 1, \ldots, K_{\max}\}, \tag{3.5}$$

where K, denoting the total number of bound regions specified by $\boldsymbol{z}$, and is thus a function of $\boldsymbol{z}$; λ is a hyperparameter; $K_{\max}$ is the largest number of bounded regions allowed by the model; and

$$C = \sum_{K=0}^{K_{\max}} \frac{\lambda^K e^{-\lambda}}{K!}, \tag{3.6}$$

which makes the prior (3.5) a proper distribution. The rationale behind this prior can be explained as follows. Since the length of each bound region is very short comparing to the chromosome or the whole genome, it is reasonable to view each bound region as a single point, and thus, following the standard theory of Poisson process, the total number of bound regions can be modeled as a Poisson random variable. Conditioning on the total number of bound regions, as implied by (3.5), we put an equal prior probability on all possible configurations of $\boldsymbol{z}$, i.e., assuming a non-informative prior for $\boldsymbol{z}$. The prior (3.5) penalizes a large value of K, where the parameter λ represents the strength of penalty. We do not recommend to use a large value of λ, as the number of true bound regions is usually small and a large value of λ will lead to discovery of too many false bound regions. Our experience shows that a value of λ around 0.01 usually works well for the ChIP-chip data. In this paper, we set $\lambda = 0.01$ in all simulations. The parameter $K_{\max}$ is usually set to a large number. We set $K_{\max} = 5000$ in all simulations of this paper. As long as the value of $K_{\max}$ has been reasonably large, increasing it further would have a negligible effect on simulations. Finally, we would like to point out that the bound region identification problem can also be viewed as a change-point identification problem that has been widely studied in statistics. For the change-point identification problem, the same

truncated Poisson prior has been used for modeling the total number of change-points by many authors, see, e.g., Phillips and Smith (1996), Liang and Wong (2000).

If $\nu_1, \ldots, \nu_K \in (\nu_{min}, \nu_{max})$, combining the likelihood and prior distributions, integrating out σ^2, and taking the logarithm, we get the following log-posterior density function

$$\begin{aligned} \log f(\boldsymbol{z}, \mu_0, \nu_1, \ldots, \nu_K | \boldsymbol{y}) &= Constant - (\frac{n}{2} + \alpha) \log \Big(\frac{1}{2} \sum_{j=1}^{n_0} (y_{0j} - \mu_0)^2 \\ &\quad + \frac{1}{2} \sum_{k=1}^{K} \sum_{j=1}^{n_k} (y_{kj} - \mu_0 - \nu_k)^2 + \beta \Big) - \log(K!) \\ &\quad + K \Big(\log(\lambda) - \log(\nu_{max} - \nu_{min}) \Big), \end{aligned} \tag{3.7}$$

otherwise, the posterior is equal to 0.

Due to the design of ChIP-chip experiments, it is obvious that the intensity measurements of the neighboring probes are positively dependent. To model this dependence, we use a latent indicator vector $\boldsymbol{z}$. This makes our model different from the existing models, such as the joint binding deconvolution model (Qi et al., 2006), the HGMM (Keles, 2007), and the Bayesian hierarchical model used in BAC (Gottardo et al., 2008). Both the joint binding deconvolution model and the Bayesian hierarchical model model the mean of probe intensities through the Gaussian random field (GMF), although their formulations may not be in the standard form of the GMF. Like the Bayesian latent model, the HGMM models the mean of probe intensities by a piece-wise constant function. The difference is that the HGMM requires one to first divide the data into genomic regions containing at most one bound regions, and thus it allows different non-bound regions to have different means. Considering the physical property of PCR, which performs independently of the form of DNA, allowing different non-bound regions to have different mean values may not be necessary.

B. MCMC Simulation

To simulate from the posterior distribution (3.7), we used the Metropolis-within-Gibbs sampler (Müller, 1991); Note that when a component of $\boldsymbol{z}$ is updated, the sum of square terms in the posterior density can be calculated in a recursive manner, and this simplifies the computation of the posterior density greatly. The detail scheme for simulating samples from the posterior distribution can be described as follows:

(a) Conditioned on $\boldsymbol{z}^{(t)}$, updating $\mu_0^{(t)}, \nu_1^{(t)}, \ldots, \nu_K^{(t)}$ using the Metropolis-Hastings (MH) algorithm, where t indexes the number of iteration cycles.

(b) Conditioned on $\mu_0^{(t)}, \nu_1^{(t)}, \ldots, \nu_K^{(t)}$, updating each component of $\boldsymbol{z}^{(t)}$ according to the following rule:

Given $z_i^{(t)}$: change $z_i^{(t)}$ to $z_i^{(t+1)} = 1 - z_i^{(t)}$ using the MH algorithm.

When a component of $\boldsymbol{z}$ is updated in step (b), the sum of square terms in the posterior density function can be calculated in a recursive manner, i.e., only the terms related to z_i need to be re-calculated.

C. Inference of Bound Regions

Let $p_i = P(z_i = 1|\boldsymbol{y})$ be the marginal posterior probability that probe i belongs to a bound region. Since the bound regions are expected to consist of several consecutive probes with positive IP-enrichment effects, the regions which consists of several consecutive probes with high marginal posterior probabilities are likely to be bound regions. To identify such regions, we follow Gottardo et al. (2008) to consider the joint posterior probability

$$\rho_i(w, m|\boldsymbol{y}) = P\Big(\sum_{j=i-w}^{i+w} z_j \geq m|\boldsymbol{y} \Big), \tag{3.8}$$

where i is the index of the probes, w is a pre-specified half-window size, and m is the minimum number of probes belonging to the bound region. As explained in Gottardo et al. (2008), the purpose of introducing the joint posterior probability is to remove the false bound regions, which usually consists of only few isolated probes with large enrichment effects. We found that the choice $w = 5$ and $m = 5$ works well in practice. This choice of w is consistent with the moving window size used in other work, such as Ji and Wong (2005), Keles (2007), and Gottardo et al. (2008). The choice of m is chosen for robustness to false bound regions. It also reflects our belief that a bound region should consist of at least five consecutive probes with large enrichment effects.

Note that estimation of ρ_i is trivial based on the samples simulated from the posterior distribution. The value of ρ_i depends on a lot of parameters, such as w, m and the hyperparameters of the model. However, we found that the orders of ρ_i are rather robust to these parameters. This suggests us to treat ρ_i as a conventional testing p-value, and to control the false discovery rate (FDR) of the bound regions using a FDR control method, e.g., the empirical Bayes method (Efron, 2004) or the stochastic approximation-based empirical Bayes method (Liang and Zhang, 2008) Both the methods allow for the dependence between testing statistics and an empirical determination of the density of the testing statistics.

Although a strict control of FDR is important to the detection of bound regions, it is not the focus of this section. In this section, we will follow other Bayesian methods, such as BAC, to simply set a cutoff value of ρ_i. We classify probe i as a probe in bound regions if $\rho_i \geq 0.5$, and classify probe i as a probe in nonbound region otherwise. As we will see in the numerical examples, the joint posterior probability can lead to a good detection of true bound regions.

3.2.3 Results

A. Real Data Analysis

a. The Estrogen Receptor data (ER data)

The estrogen receptor (ER) data were generated by Carroll et al. (2005), which mapped the association of the estrogen receptor on chromosomes 21 and 22. Here we just used a subset of the data to illustrate how the Bayesian latent model works. The subset we used is available from the BAC software at www.bioconductor.org/packages/2.2/bioc. It consists of intensity measurements for 30001 probes under the treatment and control conditions with three replicates each. The same subset has been used by BAC for a demonstration purpose.

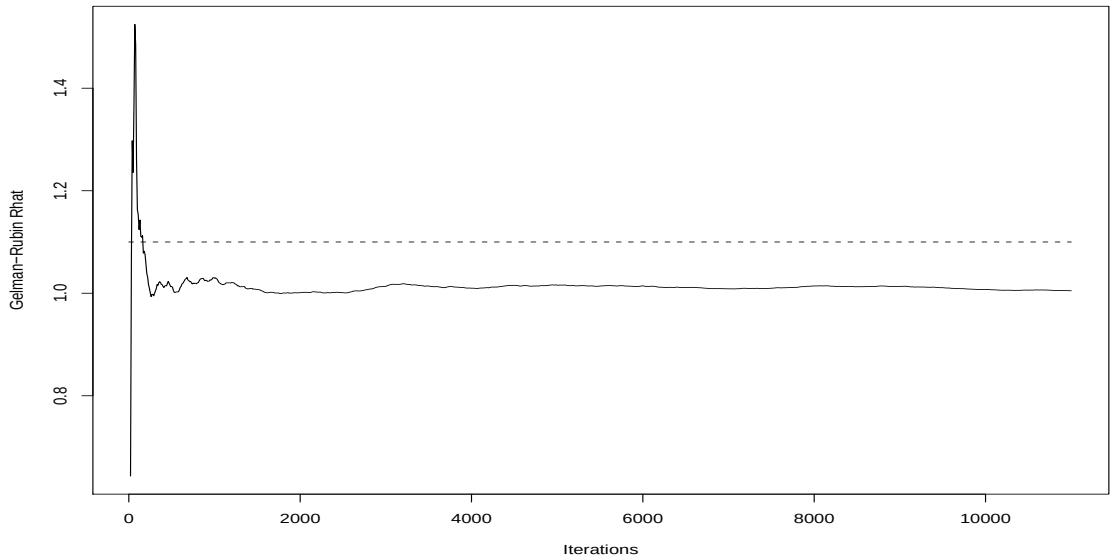


Figure 13. Convergence diagnostic of the Bayesian latent method for the ER example

.

The Bayesian latent model was first applied to the dataset. The algorithm was run 5 times. Each run consisted of 11000 iterations, and cost about 4.4 minutes CPU time on a personal computer (Intel Xeon 2.80 GHz, 1G memory, Linux operating system). All computations of this paper were done on this computer. Figure 13 provides a diagnostic plot for the convergence of the runs, where the statistic Gelman-Rubin $\hat{R}$ (Gelman and Rubin 1992) was plotted versus iterations. The simulations are usually considered to be converged when the statistic Gelman-Rubin $\hat{R}$ falls below the horizontal line 1.1. Figure 13 indicates that for this example, the simulations converged very fast, usually within two hundreds of iterations. Therefore, we discarded the first 1000 iterations for the burn-in process, and used the remaining 10,000 iterations for further inference. Figure 14(b) shows the estimates of the joint posterior probabilities resulted from one run.

For comparison, BAC and tileHMM (available at cran.r-project.org/web/packages) were also applied to this dataset. Both BAC and tileHMM produced a probability measure for each probe, similar to ρ_i, on how likely it belongs to a bound region. The results were shown in Figure 14(c) and (d), respectively. The comparison shows that all the three methods produced very similar results for this dataset. However, the results produced by the Bayesian latent model are neater; the joint posterior probabilities produced by it tend to be dichotomized, either close to 1 or close to 0. This gives the user a clear classification for the bound and non-bound regions. To provide some numerical evidence for this statement, we calculated the ratio $\#\{i : P_i > 0.5\}/\#\{i : P_i > 0.05\}$, where $\#\{i : P_i > a\}$ denotes the number of probes with P_i greater than a. Here P_i refers to the joint posterior probability for the Bayesian latent model and BAC, and the conditional probability for tileHMM. The ratios resultant from the Bayesian latent model, BAC and tileHMM are 0.816, 0.615 and 0.674, respectively.

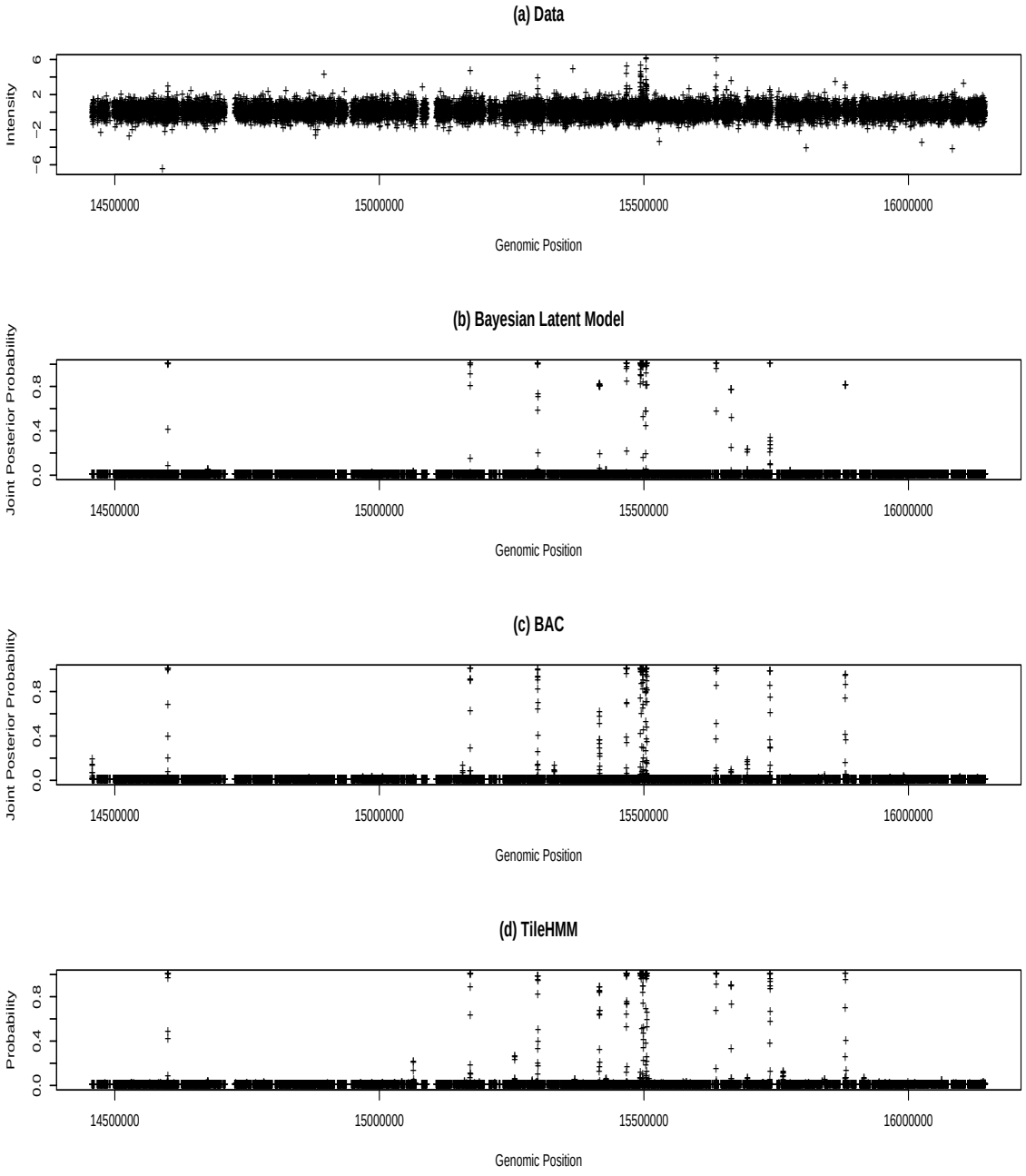


Figure 14. Comparison results for the ER data: (a) original data; (b) the joint posterior probability produced by the Bayesian latent model; (c) the joint posterior probability produced by BAC; and (d) the posterior probability produced by tileHMM.

Later, we assessed the sensitivity of the Bayesian latent method to the values of the hyperparameters $\nu_{\min}$ and λ with other parameters fixed, $\alpha = \beta = 0.05$ and $\nu_{\max} = \max_i y_i$. The cross settings $\{0.5s_y, 1.0s_y, 1.5s_y, 2s_y, 2.5s_y, 3s_y\} \times [0.0001, 0.1]$ for $(\nu_{\min}, \lambda)$ were tried for this dataset. For each setting, the algorithm was run 5 times, and each run consisted of 11,000 iterations. To measure the similarity of the bound regions resultant from different settings of the hyperparameters, we propose to use the adjusted Rand index (Rand, 1971; Hubert and Arabie, 1985). The adjusted Rand index is usually used in the literature of clustering, which measures the degree of agreement between two partitions of the same set of observations even when the comparing partitions having different numbers of clusters. It is obvious that the problem of bound region identification can also be viewed as a clustering problem; where the genome was partitioned into a series of segments, non-bound or bound regions, and each of the segments forms a cluster.

The adjusted Rand index is defined as follows. Let Ω denote a set of n observations, let $C = \{c_1, \ldots, c_s\}$ and $C' = \{c'_1, \ldots, c'_t\}$ represent two partitions of Ω, let n_{ij} be the number of observations that are in both cluster c_i and cluster c'_j, let $n_{i\cdot}$ be the number of observations in cluster c_i, and let $n_{\cdot j}$ be the number of observations in cluster c'_j. The adjusted Rand index is

$$r = \frac{\sum_{i,j} \binom{n_{ij}}{2} - \left[\sum_i \binom{n_{i\cdot}}{2} \sum_j \binom{n_{\cdot j}}{2}\right] \Big/ \binom{n}{2}}{\left[\sum_i \binom{n_{i\cdot}}{2} + \sum_j \binom{n_{\cdot j}}{2}\right] \Big/ 2 - \left[\sum_i \binom{n_{i\cdot}}{2} \sum_j \binom{n_{\cdot j}}{2}\right] \Big/ \binom{n}{2}}. \tag{3.9}$$

A higher value of r means a higher correspondence between the two partitions. When the two partitions are identical, r is 1. When a partition is random, the expectation of r is 0. Under the generalized hypergeometric model, it can be shown (Hubert and

Arabie, 1985) that

$$E\left[\sum_{i,j}\binom{n_{ij}}{2}\right] = \left[\sum_{i}\binom{n_{i\cdot}}{2}\sum_{j}\binom{n_{\cdot j}}{2}\right]\Big/\binom{n}{2}. \tag{3.10}$$

Refer to Hubert and Arabie (1985) for the theoretical justification of r.

In calculations of the adjusted Rand indices for the sensitivity experiments, we used the result shown in Figure 14(b) as the standard; that is, if a partition is identical to that partition, r will be 1. The results are summarized in Figure 15 where the adjusted Rand index is plotted as a function of $\log(\lambda)$ for different setting of $\nu_{\min}$. Figure 15 shows that, for each value of $\nu_{\min}$, the adjusted Rand index varies between 0.9 and 1.0 as λ runs from 0.0001 to 0.1. This indicates that the performance of the Bayesian latent model is rather robust to the choices of $\nu_{\min}$ and λ. Finally, we

Table 5: Robustness test of the Bayesian latent model on different choice of w and m: for each setting, the algorithm was run 5 times, and the average of adjusted Rand indices and its standard error (in the parentheses) are reported.

Adjusted Rand Index		m			
		3	5	7	10
w	2	0.996(0.001)	*	-	-
	5	0.995(0.001)	0.998(0.001)	0.908(0.013)	0.663(0.031)
	7	0.994(0.001)	0.998(0.001)	0.907(0.013)	0.742(0.017)
	10	0.972 (0.022)	0.997(0.001)	0.907(0.012)	0.729(0.017)

examined the robustness of the Bayesian latent model to different choice of w and m with other parameters fixed at $\alpha = \beta = 0.05$, $\lambda = 0.01$, and $\nu_{min} = 2s_y$. The

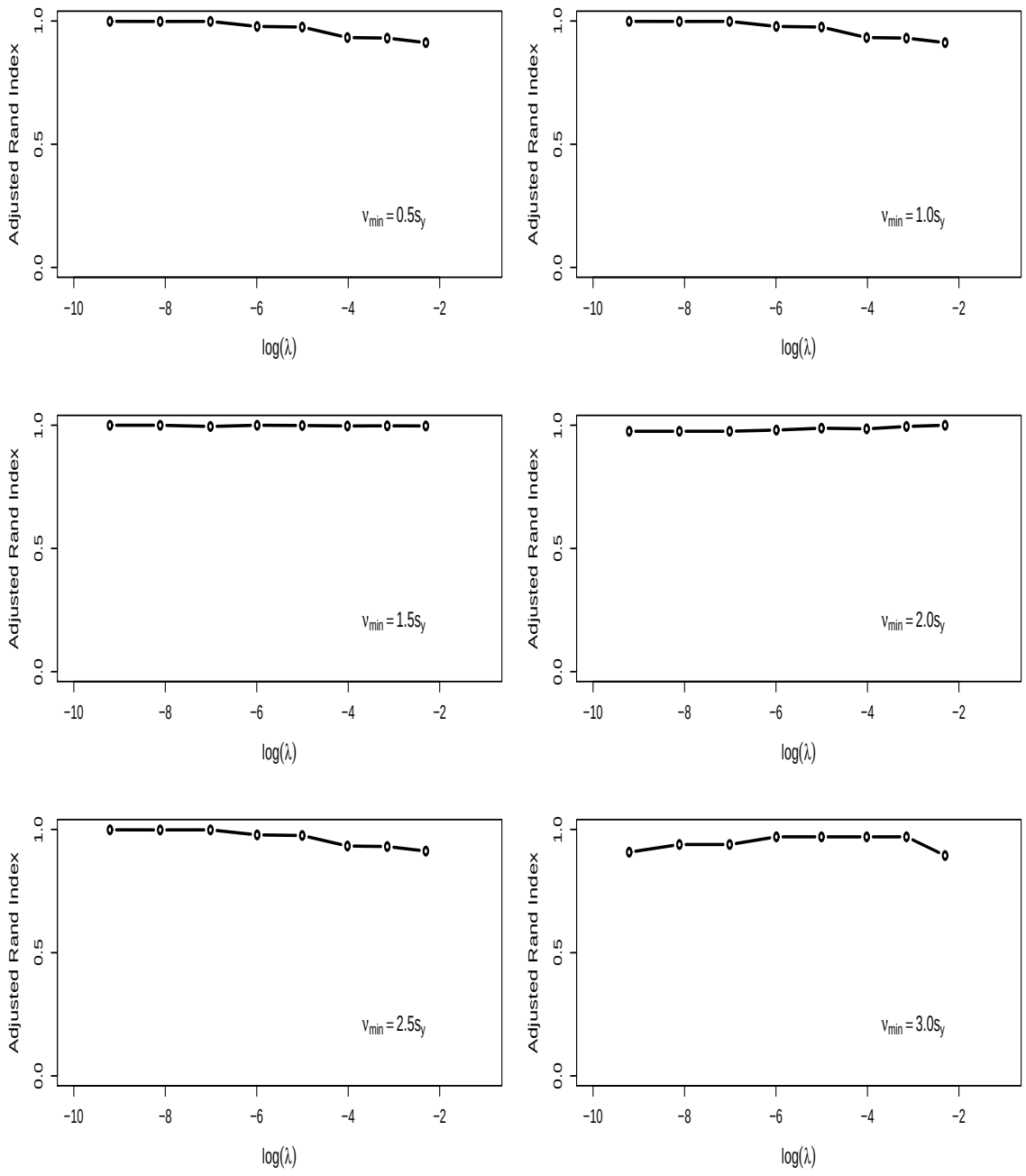


Figure 15. Sensitivity analysis for the hyperparameters.

cross settings $\{2, 5, 7, 10\} \times \{3, 5, 7\}$ for (w, m) were tried for this dataset. Again, the adjusted Rand index is used as the similarity criterion and the result shown in Figure 14(b) as the standard. The results were summarized in Table 5, which indicates, for this dataset, the Bayesian latent model is quite robust to the choices of w and m. In practice, to achieve robustness to outlying probes, we suggest to avoid choosing a small m. In all the following simulations, we set $m = 5$.

The robustness of the results with respect to changes of α, β and ν_{max} are not studied in the paper. The reason is that $\nu_{\max}$ is completely determined by the data, and the values of α and β we used form a vague prior for the variance σ^2.

b. $p53$ Data

In a ChIP-chip experiment, Cawley et al. (2004) mapped the binding sites of four human transcription factors Sp1, cMyc, p53-FL, and p53-DO1 on chromosomes 21 and 22. The experiment consisted of 6 treatment and 6 input control arrays, and the chromosomes spanned over three chips A, B and C. Refer to Cawley et al. (2004) for the details of the experiment. For the testing purpose, p53-FL data on chips A, B and C were used in this paper, which contains 14 quantitative PCR verified regions. As in Cawley et al. (2004), the data were pre-processed by filtering out the local repeats, quantile-normalized (Bolstad et al., 2003), and all were scaled to have a median feature internsity of 1000 for the purpose of adjusting batch effect, then log-transformed. Since the normalization is not the focus of this paper, we skipped the details.

The Bayesian latent method was first applied to the $p53$ data. The data on chip A, chip B, and chip C were analyzed separately. Each run consisted of 11,000 iterations. Diagnostic plot for the convergence of these runs indicates that they can converge within several hundreds of iterations, even the data on each chip consists

of more than 300,000 probes. Accordingly, the first 1000 iterations were discarded for the burn-in process, and the samples from other iterations are used for further analysis. For comparison, BAC and tileHMM were also applied to this example.

Table 6: Computational results for the $p53$-FL data with a cutoff of 0.5. Both the total number of regions and quantitative PCR verified(V) ones detected by each method are reported. Best results in terms of detection of all the validated regions are highlighted in bold.

	Chip A		Chip B		Chip C		p53	
Method	V(2)	Total	V(3)	Total	V(9)	Total	V(14)	Total
Bayesian latent	**2**	19	**3**	44	8	44	13	107
BAC	**2**	38	1	29	**9**	33	12	100
tileHMM	**2**	29708	**3**	1944	**9**	2144	**14**	33796

Given the posterior probabilities, a cutoff of 0.5 was used for all methods to detect bound regions. All resultant bound regions having less than 3 probes or 100 bps were considered to be spurious and removed, and those regions separated by 500 bps or less were merged together to form a predicted bound regions following the approach taken by Cawley (2004). The results were summarized in Table 6. Although tileHMM detected all the 14 validated regions, it essentially fails for this example. It identified a total of 33796 bound regions, which should contain too many false bound regions. We suspect that the failure of tileHMM for this example is due to its training algorithm; it is very likely that tileHMM converged to a local maximum of the likelihood function. This have been noted by Humburg et al. (2008), tileHMM may converge to a local maximum of the likelihood function with either the Baum-Welch algorithm or the

Viterbi training algorithm, rendering an ineffective inference for the model.

Both the Bayesian latent method and BAC work well for this example. At a cutoff of 0.5, BAC identified 100 bound regions, which cover 12 out of 14 experimentally validated bound regions. The Bayesian latent method works even better. At the same cutoff, it only identified 70 bound regions, but which also cover 12 out of 14 experimentally validated bound regions. For further comparison of the Bayesian latent method and BAC, we relaxed the cutoff value and counted the total number of regions needed to cover all experimentally validated regions. We found that the Bayesian latent method only needs to increase the total number of regions to 127, while BAC needs to increase to 1864 regions. Note that the BAC and tileHMM's results reported here may be a little different from those reported by other authors, due to the difference of the normalization methods.

B. Simulation Study

To have a careful assessment of the performance of the Bayesian latent model, we simulated 10 datasets based on the Sp1 data of Cawley et al.'s experiment (2004). Each dataset consists of 200,000 probes, two conditions (control and IP-enriched), and six replicates under each condition. The probe genomic coordinates we used in simulations were the first 200,000 genomic positions used in the Sp1 data. Each dataset consisted of 996 bound probes, forming 50 bound regions. As in Gottardo et al. (2008), the bound regions were assumed to describe a peak with the intensity function given by $A\exp\{-4(g_i - C)^2/B^2\}$, where A is the amplitude of the peak, B controls the width of the peak, C represents the center of the peak, and g_i is the genomic position of probe i. We also followed Gottard *et al.* (2008) to generate the centers of the bound regions randomly across the set of possible coordinates while imposing a separation of at least 3000 bps between peaks; and to generate the values

of parameter B uniformly between 600 and 1000 bps. The values of parameter A were generated uniformly between 3 and 5. The variance of the probe intensity was estimated from the Sp1 data.

Firstly, we compare the performance of different models when there is outlier probes. For the simulated data, Figure 16 (a) shows the intensity values of the first 20000 probes, as processed in (3.1). The data seems a little noisy, but still contains enough information for identification of the true bound regions. To see this, we first smoothed each replicate of the data using a moving window approach with a window size of 1000bp, and then calculated the smoothed intensity values as in (3.11).

$$\widetilde{Y}_i = \widetilde{X}_{1i} - \widetilde{X}_{2i}, \tag{3.11}$$

where $\widetilde{X}_{li}$, $l = 1, 2$, denotes the average of the smoothed data under condition l. Figure 16 (b) plots the smoothed data, where the five bars, centered at 681, 5188, 8293, 13122 and 16145, correspond to the five true bound regions, respectively. To make the problem more challenging, we tested our algorithm on the non-smoothed data. Our algorithm was run for 11000 iterations, for which the first 1000 iterations were discarded for the burn-in process, and the remaining 10000 iterations were used for inference. Figure 16(c) showed the estimates of the joint posterior probabilities. It indicate that the five bound regions have been identified by our algorithm accurately. For comparison, BAC and tileHMM were also applied to the same dataset, with the results being shown in Figures 16(d)&(e), respectively. The comparison shows that BAC totally fails for this datasets; and tileHMM works for the dataset, but the bound regions identified by it tend to be falsely prolonged. This example suggests that the Bayesian latent model is more robust to outlier probes than BAC and tileHMM.

Secondly, the performance of different models is assessed using the area under the receiving operating characteristic (ROC) curve and the error rate. The former

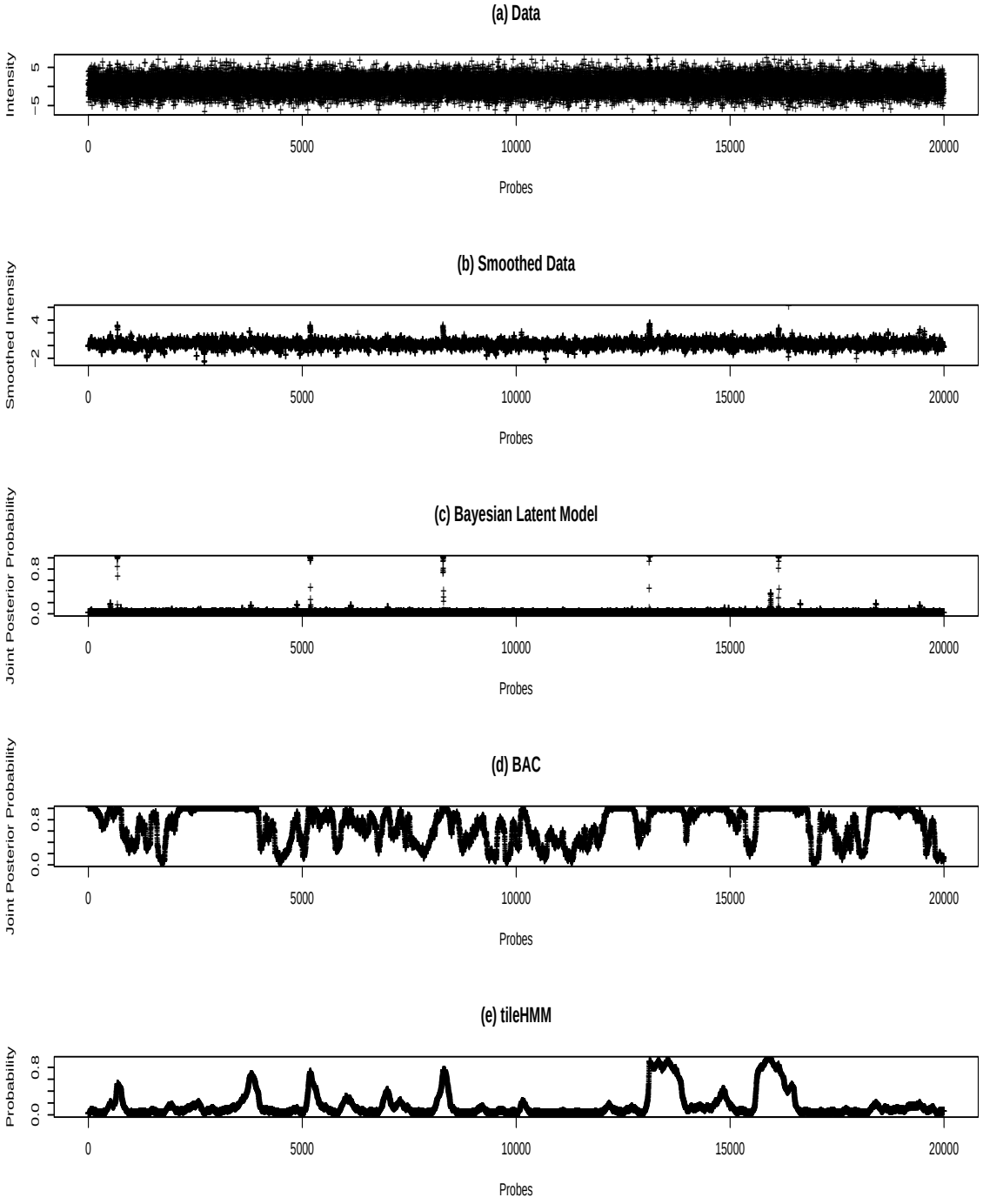
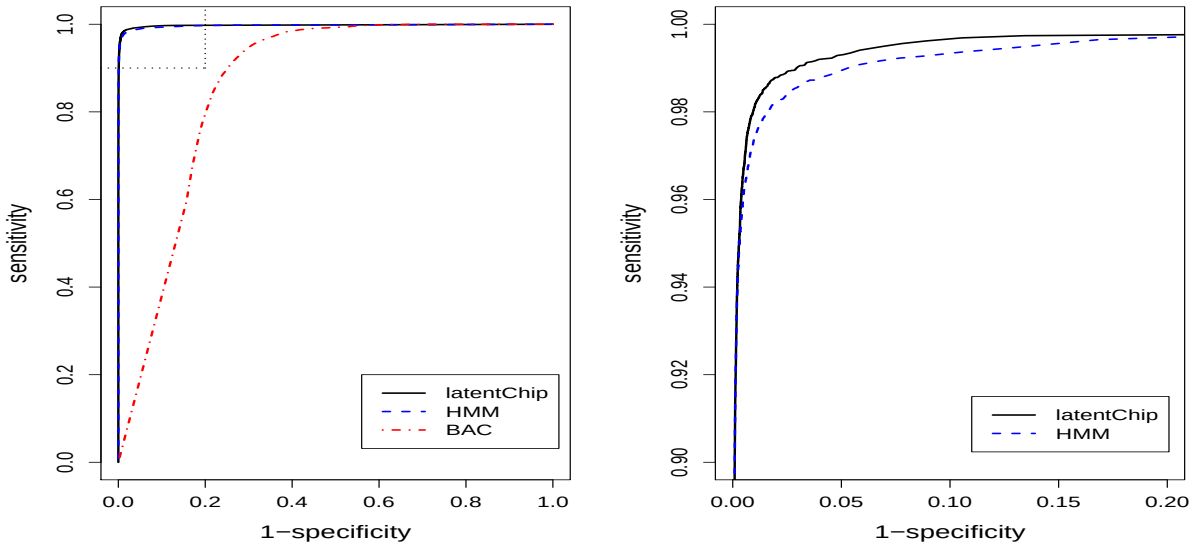
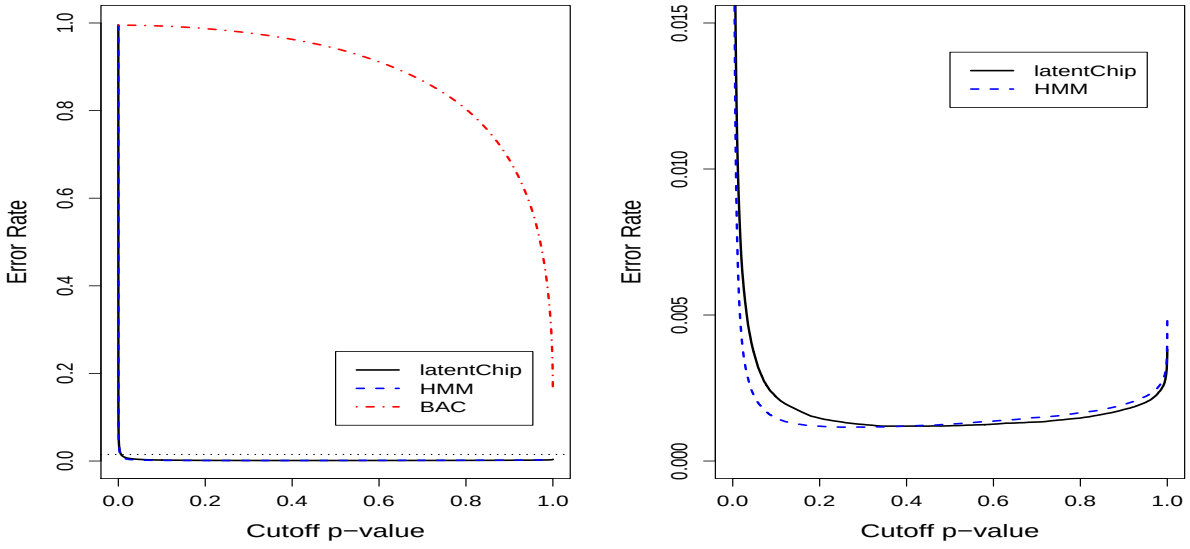


Figure 16. Comparison results for the simulated data: (a) non-smoothed data; (b) smoothed data; (c) output of the Bayesian latent model; (d) output of BAC; and (e) output of tileHMM;



(a) ROC Curves



(b) Error Rate

Figure 17. Averaged ROC curves and error rate for different models on simulated datasets: (a) ROC curves; (b) error rate. All the plots were obtained by averaging over the 10 datasets. The plots on the right provide a closer view of the area enclosed by the dotted line and axies on the left.

is a standard measure for the performance of a multiple hypothesis testing method, which shows the true positive discovery rate ($sensitivity$) against the false positive discovery rate ($1-specificity$) at probe level. The later is a standard measure for the performance of a classification method, which shows the proportion of totally incorrect probe calls, including both false positives and false negatives, against different cutoff values. All the three methods, Bayesian latent method, BAC and tileHMM, were applied to the 10 full datasets. The averaged ROC curve and error rate across a range of cutoffs are obtained and plotted in Figure 17. As indicated by Figure 17(a), the Bayesian latent method and tileHMM have very similar performances on these datasets, and both are much better than BAC. By further examining the plot on the right, which provides a closer view of the area enclosed by the dotted line and axis on the left, it is easy to see that the Bayesian latent method is better than tileHMM for this example. Next, we checked the error rate for each model. The results were shown in Figure 3(b). Again, the Bayesian latent method and tileHMM perform very well and both are much better than BAC. From the right plot of Figure 17(b), we can see that the optimal cutoff for tileHMM is close to 0.3, while it is close to 0.5 for the Bayesian latent method. Figure 17(b) also suggest that both the Bayesian latent method and tileHMM are robust to the choice of cutoff values, ranging from 0.2 to 0.8, while BAC is not.

Later, based on the true bound regions which are known for these 10 simulate datasets, we use the adjusted Rand index r to assess the quality of the results produced by the above three algorithms. In addition, we calculated p-values of the two-sample t-tests, H_0: $r_{BL} = r_O$ *vs* H_1: $r_{BL} > r_O$, where r_{BL} denotes the r-value produced by the Bayesian latent model, and r_O denotes the r-value produced by the other method. The results were summarized in Table 7. The tests indicate that the Bayesian latent model can lead to more accurate identifications of true bound regions than BAC and

Table 7: Computational results for the simulated datasets, where "Total" denotes the average number of bound regions identified for each of the 10 datasets, ND denotes the number of true bound regions that are not discovered by the algorithm, FD denotes the number of false bound regions discovered by the algorithm, r is the adjusted Rand index, the number in the parentheses is the standard error, and "EB t-scan" refers to the empirical Bayesian t-scan method proposed by Ji and Wong (2005).

Method	Total	ND	FD	r	p-value
Bayesian Latent	50.5 (0.58)	2.3 (0.33)	2.8 (0.57)	0.9545 (0.0080)	—
tileHMM	48 (0.77)	4.2 (0.57)	2.2 (0.55)	0.9250 (0.0107)	0.02
BAC	2934.7 (6.60)	0 (0)	2884.7 (6.6)	0.0609 (0.0003)	0.00
Wilcox	56.1 (0.95)	3.9 (0.48)	6.4 (0.62)	0.9221 (0.0088)	0.007
t-scan	78.9 (2.11)	3.1 (0.31)	27.6 (1.71)	0.9047 (0.0089)	0.0003
EB t-Scan	71.5 (1.52)	3.0 (0.39)	20.9 (1.38)	0.9176 (0.0068)	0.001

tileHMM.

For a thorough comparison, we also applied the sliding window methods, including the Wilcoxon rank sum test method (Cawley et al., 2004), t-scan statistic (Keles et al., 2006) and empirical Bayesian t-scan statistic (Ji and Wong, 2005), to the 10 datasets. For the testing purpose, we identified the most significant 996 probes, which is the same as the true number of bound probes, as the bound probes for each of the datasets and each of the sliding window methods. We note that this cutoff number should be determined by a multiple hypothesis test in practice, and this choice makes the comparison a little favorly biased toward the sliding window methods. The results were summarized in the lower panel of Table 7, which indicates that the Bayesian

latent model also outperforms the sliding window methods.

3.2.4 Discussion

We have proposed a Bayesian latent model for the ChIP-chip experiments. The new model mainly differs from the existing Bayesian models, such as the joint deconvolution model, the hierarchical gamma mixture model, and the Bayesian hierarchical model, in two respects. Firstly, it works on the difference between the averaged treatment and control samples. This enables the use of a simple model for the data, which avoids the probe-specific effect and the sample (control/treatment) effect. As a consequence, this enables an efficient MCMC simulation of the posterior distribution, and also makes the model fairly robust to the outliers. Secondly, it models the neighboring dependence of probes by introducing a latent indicator vector. A truncated Poisson prior distribution is assumed for the latent indicator variable, with the rationale being justified at length.

The Bayesian latent model has been successfully applied to the ER, $p53$, and some simulated datasets, with comparisons with BAC, tileHMM, and some sliding window methods. The numerical results indicate that the Bayesian latent model can outperform others, especially when the dataset contains outlying probes.

The Bayesian latent model can be generalized in a few ways. Firstly, it can be generalized to allow different bound regions to have different variances. This generalization has been implemented by us. The numerical results are very similar to those reported in the paper.

Secondly, it can be generalized to work on the multiple replicates directly. This can be simply done by modifying (3.2) to multivariate normals. This generalization will certainly slow down the simulations, but the results may not be improved significantly. The reason is that under the assumption of constant variances for probe

intensities, the statistic (3.1) is sufficient for the mean intensity of probes, while the latter has been designed in the experiment as the main measure for differentiating bound and non-bound regions.

The reason why the Bayesian latent method outperforms tileHMM and BAC can be explained as follows, through a detailed comparison of the models used by them. TileHMM implemented a standard two-state hidden Markov model, with the emission distribution of state S_i, $i = 1, 2$, being modeled as a t-distribution. TileHMM and the Bayesian latent model are mainly different in two respects.

- TileHMM is a non-Bayesian method, where maximum likelihood estimates are used for all model parameters and inference for the bound regions are based on the conditional probability of the hidden states. TileHMM is trained using the Baum-Welch algorithm and the Viterbi algorithm. It is known that the Baum-Welch algorithm is an EM algorithm implemented in the context of HMM, and that it tends to converge to a local maximum of the likelihood function. The Viterbi algorithm provides a fast alternative to the Baum-Welch algorithm, but may not converge to a local maximum. The Bayesian latent method is a Bayesian method, where inference for bound regions is based on the posterior distribution of the latent variable. The posterior distribution is simulated using the Metropolis-within-Gibbs sampler, which is known to converge to its target distribution when the number of iterations becomes large.

- TileHMM models all bound regions to have the same mean value, while the Bayesian latent model allows different bound regions to have different mean values. Our model fits the real data better.

The mixed performance of tileHMM on the simulated and real datasets indicates that the inferiority of tileHMM is mainly due to its training algorithm. In addition,

as indicated by our simulated examples, tileHMM tends to misidentify the bound regions with relatively low probe intensities, because it models all bound regions to have the same mean value.

BAC models the probe intensity using a mixed-effect model:

$$y_{cpr} = \mu_p + \gamma_{cp} + \epsilon_{cpr}, \quad c = 1, 2, \tag{3.12}$$

where $c = 1$ denotes the control sample, $c = 2$ denotes the treatment sample, r is the index of replicates; μ_p is a random probe effect distributed as $N(0, \sigma_\mu^2)$; γ_{cp} is the probe enrichment effect with $\gamma_{1p} = 0$; and ϵ_{cpr} is the random error distributed as $N(0, \sigma_{cp}^2)$. The authors further modeled the probe enrichment effect by a mixture of a point mass at zero and a truncated Gaussian distribution, i.e.,

$$\gamma_{2p} \sim (1 - w_p)\delta_0 + w_p TN_+(\xi, \sigma_\gamma^2), \tag{3.13}$$

where $TN_+(\xi, \sigma_\gamma^2)$ denotes a truncated Gaussian distribution truncated at zero, and w_p is the *a priori* proportion of probes belonging to nonbound regions. The *a priori* proportion depends on a latent Markov random field prior $\boldsymbol{\theta} = \{\theta_p, 1 \leq p \leq P\}$, through a logistic transformation

$$w_p = \frac{e^{\theta_p}}{1 + e^{\theta_p}}, \tag{3.14}$$

and a Gaussian intrinsic autoregressive model (Besag and Kooperberg, 1995) for $\boldsymbol{\theta}$,

$$\theta_p | \theta_{\partial p} \sim N\left(\frac{\sum_{p' \in \partial p} \theta_{p'}}{n_p}, \frac{n}{n_p \kappa}\right), \tag{3.15}$$

where ∂p corresponds to the probes p' immediately adjacent to p, n_p is the cardinality of ∂p, κ is a smoothing parameter, and n is the number of neighboring probes used. The model is trained using a MCMC algorithm.

The main difference between the BAC and the Bayesian latent methods is that

BAC models the control and treatment samples jointly, while the Bayesian latent method models the difference between the averaged treatment and control samples. Since BAC models the treatment and control samples jointly, it has to include the probe-specific effect in the model and assume a complicated structure for the random error, assuming the variance depends on both the probe and the type of samples (control or treatment). By working on the difference between the averaged treatment and control samples, the Bayesian latent method eliminates the probe effect in the model and the dependence of the random error on the probe and the type of samples. This simplifies the model greatly and enables an efficient MCMC simulation from the the posterior distribution. In addition, due to the complicated structure of the model, BAC includes too many parameters, and this makes the model potentially overfitted, especially when the number of replicates is small. This explains why BAC always tends to identify too more bound regions than does the Bayesian latent model. On the other hand, the simplicity of the Bayesian latent model makes it rather robust to outlying probes. As indicated by our examples, it work well for all examples studied in this paper.

3.2.5 Software Package

An **R** software package called **LatentChIP**, which implements the Bayesian latent model under linux operating system has been developped, and is available upon request.

3.3 Testing Multiple Hypotheses Using Population Information of Samples *

3.3.1 Background

In biomedical study, many problems involve simultaneous tests of thousands, or even millions, of null hypotheses. For example, Gottardo et al. (2006) considered the problem of detection of differentially expressed genes under HIV-infected and non-infected conditions using microarrays, where 7680 hypotheses (genes) were tested simultaneously; and Cawley et al. (2004) considered the problem of identification of human transcription factor binding sites via ChIP-chip experiments, where more than $300,000$ hypotheses were evaluated simultaneously. How to effectively use the vast data in multiple hypothesis tests poses a great challenge for statisticians. The conventional multiple hypothesis testing procedure consists of the following typical steps:

- *Sample collection.* Let $X_1, \ldots, X_{r_1}$ denote the samples collected under the control condition, and let $Y_1, \ldots, Y_{r_2}$ denote the samples collected under the treatment condition. In a microarray experiment, for example, $X_i = (x_{1i}, \ldots, x_{ni})'$ is a vector of gene expression levels measured on array i and n is the number of genes involved in the experiment. This is the same for Y. Henceforth, X's and Y's are called the control and treatment samples, respectively; and $(x_{k,1}, \ldots, x_{k,r_1})$ and $(y_{k,1}, \ldots, y_{k,r_2})$ are called the control and treatment samples of subject k, respectively.

---

* Part of this section is reprinted with permission from "Testing multiple hypotheses using population information of samples" by Mingqi Wu and Faming Liang, 2010. *JP Journal of Biostatistics* **4**(2), 181-201.

- *Test score or p-value evaluation.* This is hypothesis dependent. For example, to test the mean difference between the control and treatment samples, the two-sample Welch t-statistic (Welch, 1938)

$$t_k = \frac{\sum_{j=1}^{r_1} x_{k,j}/r_1 - \sum_{j=1}^{r_2} y_{k,j}/r_2}{\sqrt{s_{x,k}^2/r_1 + s_{y,k}^2/r_2}}, \tag{3.16}$$

with the degree of freedom calculated as

$$\nu = \frac{(s_{x,k}^2/r_1 + s_{y,k}^2/r_2)^2}{(s_{x,k}^2/r_1)^2/(r_1-1) + (s_{y,k}^2/r_2)^2/(r_2-1)}, \tag{3.17}$$

is often used under the assumption that the experimental samples of each subject are mutually independent and normally distributed, where $s_{x,k}^2$ denotes the sample variance of $x_{k,1}, \ldots, x_{k,r_1}$ and $s_{y,k}^2$ denotes the sample variance of $y_{k,1}, \ldots, y_{k,r_2}$. The p-value of subject k can be calculated as $p_k = P(T_\nu > t_k)$, where T_ν denotes a student t random variable with degree of freedom ν. The test score of subject k can be calculated as $z_k = \Phi^{-1}(1-p_k)$, where $\Phi(\cdot)$ denotes the cumulative distribution function (CDF) of the standard normal distribution.

- *Significant subject identification.* This can be done with various criteria, e.g., the per-comparison error rate, the family-wise error rate (Dudoit et al., 2003), and the false discovery rate (FDR) (Benjamini and Hochberg, 1995; Efron, 2004).

Although the above procedure has succeeded in many applications, a drawback of the procedure is that the power of each individual test is low. This is because the sample replicates r_1 and r_2 are usually small and each individual test only makes use of sample information from the subject that it is testing. Given the vast data involved in a multiple hypothesis test, it is natural to think about how to make effective use of population information of samples to improve the power of the test for each individual

subject and thus to improve the power of the multiple hypothesis test.

In this project, we propose a nonparametric method for evaluation of test scores for each individual subject. The method consists of two key steps, smoothing over neighboring subjects and density estimation over control samples, both of which allow for the use of population information of the subjects. The new method is tested on both the gene expression data and the ChIP-chip data. The numerical results indicate that use of population information can significantly improve the power of multiple hypothesis tests. In other words, the proposed method can significantly reduce the number of duplicates of the routine microarray and ChIP-chip experiments and thus the experimental cost, while maintaining the same level of statistical power in the analysis.

3.3.2 Population-based Multiple Hypothesis Test

In this section, we first describe a nonparametric method for evaluation of test scores for each individual subject under the assumption that the control samples are homogeneous, and then describe a procedure on how to prepare homogeneous control samples. The control samples are said homogeneous if the samples are identically distributed over all subjects. Finally, we describe how to identify significant subjects using a stochastic approximation FDR method (Liang and Zhang, 2008).

A. A Test With Homogeneous Control Samples

Suppose that r_1 control samples, $X_1, \ldots, X_{r_1}$, and r_2 treatment samples, $Y_1, \ldots, Y_{r_2}$, have been collected in the experiment, respectively; and that the control samples are homogeneous. Furthermore, suppose that we are interested in testing simultaneously the mean difference of the control and treatment samples of n subjects; that is, to test the hypotheses $H_{k0} : \mu_{x,k} = \mu_{y,k}$ versus $H_{k1} : \mu_{x,k} < \mu_{y,k}$ for $k = 1, \ldots, n$, where

$\mu_{x,k}$ and $\mu_{y,k}$ denote, respectively, the means of the control and treatment samples of subject k. To make use of population information in the test, we propose the following procedure:

- *Density estimation.* Fit a base density for $X_i = (x_{1i}, x_{2i} \ldots, x_{ni})'$ using a non-parametric density estimation method, e.g., estimating f with the kernel estimator of the form

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{j=1}^{n} K(\frac{x - x_{ji}}{h}) \tag{3.18}$$

  where K is the kernel, and h is the bandwidth which can be selected using an empirical plug-in rule. As shown by Hall, Lahiri and Truong (1996), this estimator is consistent even for long-range dependent data. Its asymptotic expansion for the mean integrated squared error (MISE) agrees to the second order with that of independent data. Denote the CDF of the fitted density by F_{X_i}. Totally, r_1 base densities are obtained. The kernel CDF, F_X, can be represented as

$$F_X = \frac{1}{r_1} \sum_{i=1}^{r_1} F_{X_i}, \tag{3.19}$$

  by averaging over all base CDFs.

- *Test score evaluation.* Evaluate the p-value for each treatment sample of subject k using the kernel CDF F_X by

$$p_{k,j} = 1 - F_X(y_{k,j}), \quad j = 1, 2, \ldots, r_2, \quad k = 1, 2, \ldots, n, \tag{3.20}$$

  and then evaluate the test score of subject k by

$$Z_k = \frac{1}{r_2} \sum_{j=1}^{r_2} \Phi^{-1}(1 - p_{k,j}), \quad k = 1, 2, \ldots, n, \tag{3.21}$$

  which averages the test scores over all replicates.

Hereafter, this test will be called the population-based test. Comparing to the two-sample t-test, the population-based test has several significant advantages. Firstly, it incorporates the population information of the control samples into the individual tests by basing the p-value evaluation on the fitted kernel distribution of the control samples. Secondly, it allows for the use of a single pair of control-treatment samples in multiple hypothesis tests; that is, both r_1 and r_2 can be as small as 1. Thirdly, it is a nonparametric method which avoids the normality assumption for the samples. In our experience, the normality assumption is often violated by real biomedical data.

B. A General Procedure for Preparing Homogeneous Control Samples

The key assumption of the population-based test is that the control samples are homogeneous, otherwise, the density estimation step is not sound. However, the raw control samples collected in biomedical study are usually not homogeneous. For example, this can be caused by the subject-specific effect. Here, we propose a general procedure, which will transform the raw control samples to be homogeneous or approximately homogeneous. Note that our underlying assumption for the transformation is that the experimental samples follows a distribution in the location-scale family.

Let $\boldsymbol{a}_k = (x_{k,1}, \ldots, x_{k,r_1}, y_{k,1}, \ldots, y_{k,r_2})'$ represent the samples of subject k, $k = 1, \ldots, n$, where the part $(x_{k,1}, \ldots, x_{k,r_1})$ denotes the control samples, and the part $(y_{k,1}, \ldots, y_{k,r_2})$ denotes the treatment samples. The transformation procedure can be described as follows.

For $k = 1, 2, \ldots, n$, do the following:

- *Neighboring subject identification.* Calculate the distance between subject k and

all other subjects using

$$d(\boldsymbol{a}_k, \boldsymbol{a}_s) = \|\boldsymbol{a}_k - \boldsymbol{a}_s\|, \quad \text{for } s = 1, \ldots, n, \tag{3.22}$$

where $\|\boldsymbol{z}\|$ denotes the Euclidean norm of the vector $\boldsymbol{z}$. Identify l nearest subjects in terms of distance $d(\cdot, \cdot)$. The l subjects are called the neighboring subjects of subject k. The l is a predetermined number, depending on the problem under study. How to choose l will be discussed later; its effect will be measured in our simulated microarray data example.

- *Smoothing.* Smooth the samples of subject k by weightedly averaging the samples of the neighboring subjects. The method of weight assignment is also problem dependent. Generally speaking, the magnitude of the weight assigned to a neighboring subject should be reversely correlated to its distance to subject k.

- *Standardization.* Let $\boldsymbol{a}_k^* = (x_{k,1}^*, \ldots, x_{k,r_1}^*, y_{k,1}^*, \ldots, y_{k,r_2}^*)'$, $k = 1, 2, \ldots, n$, denote the smoothed samples of subject k. Let $v_{x^*,k}$ and $v_{y^*,k}$ denote the James-Stein shrinkage variance (Opgen-Rhein and Strimmer, 2007) of the smoothed control and smoothed treatment samples, respectively. Thus,

$$v_{x^*,k} = \lambda s_{x^*,median}^2 + (1 - \lambda) s_{x^*,k}^2, \tag{3.23}$$

where $s_{x^*,k}^2$ is the sample variance of $(x_{k,1}^*, \ldots, x_{k,r_1}^*)$, $s_{x^*,median}^2$ is the median of $s_{x^*,k}^2$'s, and λ is the pooling parameter defined by

$$\lambda = \min\left(1, \frac{\sum_{k=1}^n \widehat{Var}(s_{x^*,k}^2)}{\sum_{k=1}^n (s_{x^*,k}^2 - s_{x^*,median}^2)^2}\right), \tag{3.24}$$

where $\widehat{Var}(s_{x^*,k}^2) = \frac{r_1}{(r_1-1)^3} \sum_{i=1}^{r_1} (w_{ki} - \bar{w}_k)^2$, $w_{ki} = (x_{k,i}^* - \bar{x}_k^*)^2$, $\bar{w}_k = \frac{1}{r_1} \sum_{i=1}^{r_1} w_{ki}$, and $\bar{x}_k^* = \frac{1}{r_1} \sum_{i=1}^{r_1} x_{k,i}^*$. The $v_{y^*,k}$ can be defined similarly. Given $v_{x^*,k}$ and $v_{y^*,k}$,

we estimated the pooled variance of the samples of subject k by

$$\widehat{\sigma}_k^2 = \frac{(r_1 - 1)v_{x^*,k} + (r_2 - 1)v_{y^*,k}}{r_1 + r_2 - 2}. \tag{3.25}$$

Note that setting $\lambda = 0$, $\widehat{\sigma}_k^2$ is reduced to the conventional pooled variance estimator for two samples.

Then, under the null hypothesis $H_{k0} : \mu_{x,k} = \mu_{y,k}$, we standardize the control and treatment samples of subject k by

$$\tilde{x}_{k,i} = \frac{x^*_{k,i} - \bar{x}^*_k}{\widehat{\sigma}_k}, \qquad \tilde{y}_{k,j} = \frac{y^*_{k,j} - \bar{x}^*_k}{\widehat{\sigma}_k}, \tag{3.26}$$

for $i = 1, \ldots, r_1$ and $j = 1, \ldots, r_2$.

It is clear that the samples $\tilde{x}_{k,i}$s are identically distributed under the mild assumption that the original samples $x_{k,i}$ follows a distribution in the location-scale family. Thus, the transformed control samples are homogeneous, and the density estimation followed by the test score evaluation method described in §2.1 is applicable. This approach is similar to Song and Hart's cluster-based density estimate (2009). Note that the transformation procedure has been designed to incorporate information from other subjects. This reflects in two steps, smoothing over neighboring subjects and calculation of James-Stein shrinkage variance. As argued at the end of the paper, smoothing over neighboring subjects reduces effectively the variation of the experimental samples, while causing only negligible bias to the mean of the samples as long as l, the size of neighboring subject set, is reasonable.

In this section, we only outline the idea how to prepare homogeneous control samples by using population information of the samples. In practice, many detailed steps, such as determination of neighboring subjects and smoothing weight assignment, will depend on the problem under study. In Section 3.3.3 and 3.3.4, we will

give details on how the idea works for the ChIP-chip data and the gene expression data.

C. FDR Control

Given the test scores, a multiple hypothesis testing procedure is still needed for identification of significant subjects. Here, we adopted the stochastic approximation-based FDR control method developed by Liang and Zhang (2008), which, hereafter, will be abbreviated as the SA-FDR method. The SA-FDR method falls into the class of empirical Bayes methods (Efron, 2004). Like other methods in this class, it works by fitting the test scores with a two-component mixture model

$$f(z) = \pi_0 f_0(z) + (1 - \pi_0) f_1(z), \tag{3.27}$$

where π_0 is the prior probability that a null hypothesis is true, f_0 is the empirical null distribution, f_1 is the alternative distribution, and f_0 is stochastically smaller than f_1. Given the estimators of π_0 and f_0, the positive FDR (Storey et al., 2004) of a rejection rule $\Lambda = \{Z_i \geq z_0\}$ can be estimated by

$$\widehat{\mathrm{Fdr}}(\Lambda) = \frac{N\widehat{\pi}_0[1 - \widehat{F}_0(z_0)]}{\#\{z_i : z_i \geq z_0\}}, \tag{3.28}$$

where $\#\{z_i : z_i \geq z_0\}$ denotes the number of subjects with test scores greater than z_0, $\widehat{\pi}_0$ denotes the estimator of π_0, and $\widehat{F}_0$ denotes the CDF estimator of f_0. Note that $\widehat{\mathrm{Fdr}}(\Lambda)$ can be intuitively interpreted as the expected proportion of null subjects, i.e., the subjects with the null hypotheses being true, among those with the test score greater than z_0. Following the suggestion by Storey (2002), the q-value defined below

$$q(z) \equiv \inf_{\{\Lambda : z \in \Lambda\}} \mathrm{Fdr}(\Lambda), \tag{3.29}$$

is used in this paper as a reference quantity for the decision of multiple hypothesis testing.

In Liang and Zhang (2008), π_0 and f_0 are estimated using a two-step procedure:

- Fit the distribution of the test scores with a mixture of exponential power distributions using the stochastic approximation method (Robbins and Monro, 1951; Benveniste et al., 1990).

- Clustering the components of the mixture exponential power distributions into two clusters, which correspond to f_0 and f_1 of the mixture (3.27) respectively, according to the mutual distance between the components.

Liang and Zhang (2008) showed theoretically that the method is valid under general dependence between test scores. We note that for the population-based test proposed in this paper, the use of the SA-FDR method is not essential. Any other multiple comparison methods, e.g., the methods developed by Benjamini and Yekutieli (2001), Storey et al. (2004), and Efron (2004), can be equally used here. To use the method proposed by Benjamini and Yekutieli (2001) and Storey et al. (2004), one may need to transform the test scores to p-values via the transformation $P = 1 - \Phi^{-1}(Z)$.

3.3.3 ChIP-chip Data Analysis

In this section, we applied the population-based test to ChIP-chip data for the purpose of identification of transcription factors binding sites (TFBS). The performance of our method is first assessed on a real dataset, and then assessed on some simulated datasets.

A. $p53$ Data

The dataset we studied here was generated by Cawley et al. (2004), whose experiment mapped the binding sites of four human transcription factors Sp1, cMyc, $p53$-FL, and $p53$-DO1 on chromosomes 21 and 22. The chromosomes were spanned over three chips A, B and C. All experiments were done under three conditions: IP, control GST and control input. For each transcription factor, under each experiment condition, 6

samples (2 biological replicates $\times$ 3 technical replicates) were obtained. For the testing purpose, $p53$-FL data on chips A, B and C, under IP and control input conditions, were analyzed in this paper. The raw data is available at http://transcriptome.affymetrix.com/publication/tfbs.

For comparison, the raw data were pre-processed as in Cawley et al. (2004). We first filtered out the local repeats, and then normalized the data using the quantile-normalization method (Bolstad et al., 2003). After normalization, the data were rescaled to have a median feature intensity of 1000, log-transformed, and then processed as prescribed in section 3.3.2 B. For the ChIP-chip data, the neighbor identification step can be skipped, because, by the nature of the data, the probes have been self-clustered into bound and non-bound regions. For the smoothing step, the Gaussian weighted moving average method was applied as in Zheng et al. (2007). A window with size $1000bp(\pm 500bp)$ was moving along the genome. The intensity of the probe in the center of the window is updated by

$$\boldsymbol{a}_k^* = \sum_{i \in window} w_i \boldsymbol{a}_i / \sum_{i \in window} w_i, \qquad w_i = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{d_{k,i}^2}{2\sigma^2}). \tag{3.30}$$

where $\boldsymbol{a}_i$ denotes the intensity values of probe i measured in the experiment, $d_{k,i}$ is the genomic distance between the central positions of probe i and probe k, and the standard deviation is set to be one fourth of the window size, $\sigma = 250bp$. Note that the probe-specific effect has been removed by sample centralization in the standardization step (3.26). Hence, the control samples are, at least approximately, homogeneous after pre-processing.

Following Keles et al. (2006), we define a scan-statistic, which is a moving average of the test scores resultant from the population-based method, i.e.,

$$Z_k^s = \frac{1}{2w+1} \sum_{i=k-w}^{k+w} Z_i, \tag{3.31}$$

where $w = 5$ is the half moving window size, which is the same as that used in Ji and Wong (2005), Keles et al. (2006), and Gottardo et al. (2008). By doing so, information from neighboring probes were further borrowed for identification of bound regions.

Table 8: Computational results for the $p53$-FL data. V: the number of bound regions that have been experimentally validated and identified by the method; a: the cutoff number specified by Cawley et al. (2004); b: the number of bound regions that have been experimentally validated on the chip; and τ^*: the number of "significant" probes needed to cover all experimentally validated bound regions.

	Chip A		Chip B		Chip C	
Method	V(36^a,2^b)	τ^*	V(353^a,2^b)	τ^*	V(423^a,10^b)	τ^*
Wilcoxon	2	29	1	862	6	6401
Population-based	2	34	2	71	8	1136

In Cawley et al. (2004), a cutoff of 10^{-5} was used for the p-values resultant from the Wilcoxon rank sum test, and this led to 36, 353 and 423 probes being identified as "significant" probes on chip A, B and C, respectively. For comparison, we set the cutoff numbers to 36, 353 and 423 for the test scores on chips A, B and C, respectively. Following the approach taken by Cawley et al. (2004), the regions having less than 3 probes or 100 bps were considered to be spurious and removed, and the regions separated by 500 bps or less were merged together to form a predicted bound region. The results were summarized in Table 8. The Wilcoxon rank sum test identified 9 out of 14 experimentally validated bound regions, while the population-based test identified 12 out of the 14 validated bound regions. For

a further comparison, we relaxed the cutoff number and counted the total number of "significant" probes needed to cover all the 14 validated bound regions. For the Wilcoxon rank sum test, it needs to increase the total number of "significant" probes to 7292; while for the population-based method, it only requires 1241 "significant" ones. The population-based test outperforms significantly the Wilcoxon rank sum method for this example.

B. Simulated Data

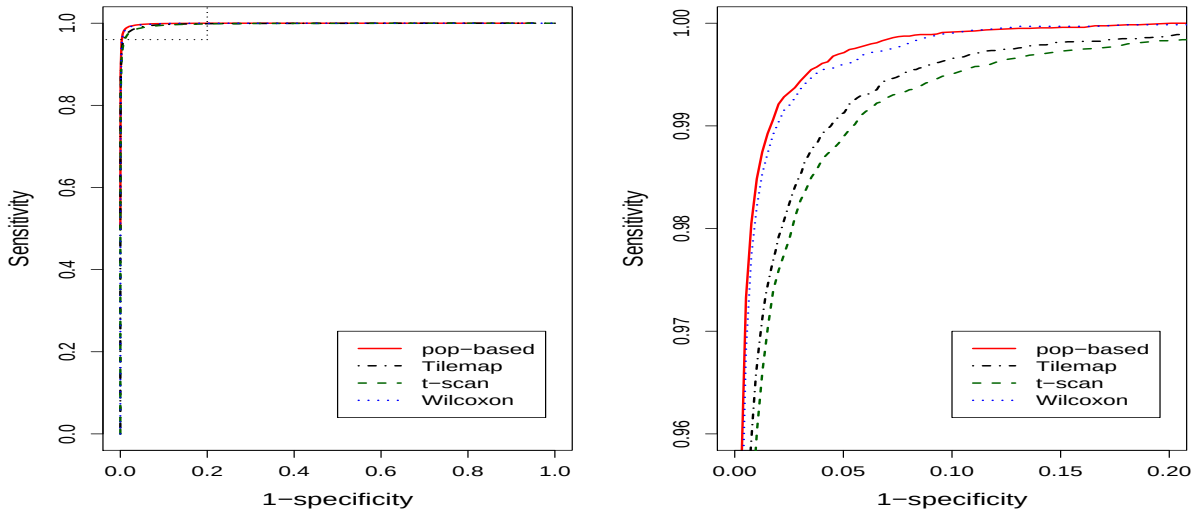
To have a careful assessment of the performance of the population-based test on ChIP-chip data, we simulated 20 datasets based on the Sp1 data generated by Cawley et al. (2004). Each dataset consists of 200,000 probes, two conditions (IP and control input), and six replicates under each condition. We extract the first 200,000 genomic positions of the Sp1 data as the probe genomic coordinates in the simulations. Each dataset consisted of 996 bound probes, which form 50 bound regions. As in Gottardo et al. (2008), the bound regions were assumed to describe a peak with the intensity function given by $A\exp\{-4(g_i - C)^2/B^2\}$, where A is the amplitude of the peak, B controls the width of the peak, C represents the center of the peak, and g_i is the genomic position of probe i. We also followed Gottardo et al. (2008) to generate the centers of the bound regions randomly across the set of possible coordinates while imposing a separation of at least 3000 *bps* between peaks; and to generate the values of parameter B uniformly between 600 and 1000 *bps*. The values of parameter A were generated uniformly between 3 and 5. The variance of the probe intensity was estimated from the Sp1 data. For comparison, four different methods were applied to the 20 simulated datasets, including the Wilcoxon rank sum test (Cawley et al., 2004), t-scan test (Keles et al., 2006), Tilemap (Ji and Wong, 2005), and population-based test. The results were summarized in table 9. For testing purpose, we tried

Table 9: Computational results for the simulated datasets. At each cutoff number τ, the number of false negative bound regions (missed regions), the number of false positive bound regions (extra regions), and the number of true positive probes (matched probes) with their standard deviations (the numbers in the parentheses) were calculated by averaging over the 20 datasets.

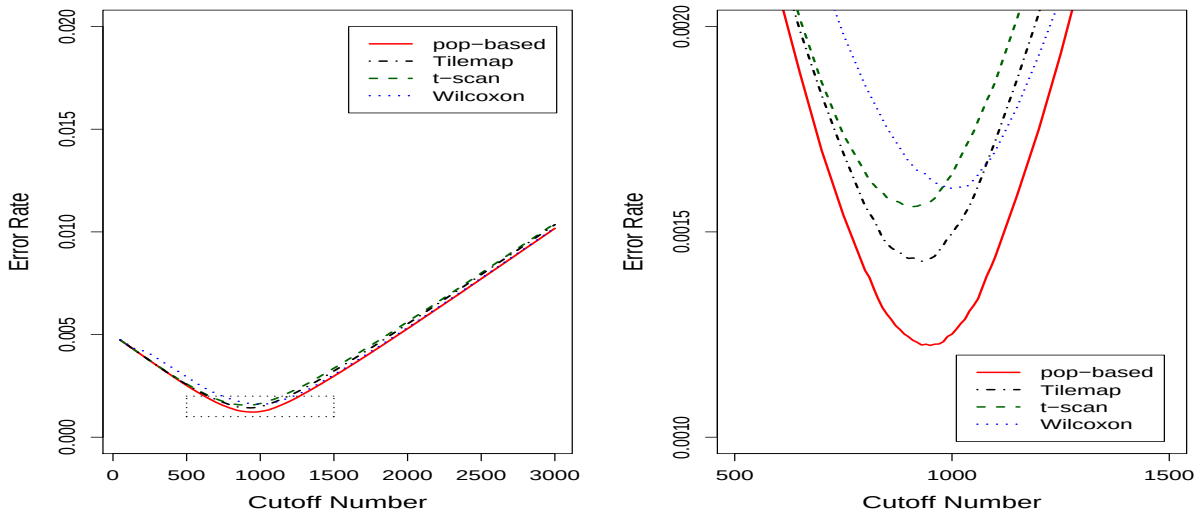
		Methods			
τ	Criteria	Wilcoxon	t-scan	Tilemap	pop-based
	Missed regions	8.1(.37)	4.9(.35)	4.5(.37)	**4.2(.29)**
800	Extra regions	1.2(.29)	5.7(.69)	2.7(.31)	**0.9(.18)**
	Matched probes	712.3(2.62)	733.5(2.54)	741.8(1.85)	**757.4(2.15)**
	Missed regions	5.6(.34)	3.3(.29)	3.3(.28)	**2.5(.21)**
900	Extra regions	3.2(.52)	13.9(1.36)	7.9(.63)	**2.5(.29)**
	Matched probes	780.9(3.26)	791.9(3.63)	804.5(2.94)	**823.6(2.42)**
	Missed regions	3.2(.40)	2.5(.24)	2.6(.23)	**1.5(.15)**
1000	Extra regions	6.1(.66)	26.6(1.30)	19.8(1.16)	**6.4(.37)**
	Matched probes	837.3(3.82)	834.0(4.30)	848.5(3.32)	**872.9(2.79)**
	Missed regions	1.95(.36)	2.1(.23)	1.7(.21)	**1.0(.15)**
1100	Extra regions	15.6(.89)	44.7(1.92)	37.7(1.55)	**12.3(.60)**
	Matched probes	878.3(4.71)	861.3(4.07)	875.8(3.95)	**904.0(2.58)**
	Missed regions	1.5(.26)	1.6(.22)	1.4(.17)	**0.9(.17)**
1200	Extra regions	27.9(1.14)	67.8(2.48)	58.9(1.59)	**21.1(.75)**
	Matched probes	905.0(4.47)	879.4(4.24)	894.8(3.49)	**923.0(2.55)**
	Missed regions	0.7(.16)	1.2(.17)	1.1(.20)	**0.5(.11)**
1500	Extra regions	84.6(2.27)	144.1(3.14)	141.6(2.69)	**48.9(0.99)**
	Matched probes	943.7(4.22)	912.0(4.00)	925.6(3.24)	**950.5(2.26)**

different cutoff numbers for the probes. At each of the cutoff numbers, the methods were compared in three criteria, the number of false negative bound regions (i.e., the number of bound regions not identified by the method), the number of false positive bound regions (i.e., the number of falsely identified bound regions), and the number of true positive probes (i.e., the number of correctly identified bound probes). The numerical results show that for this example, Tilemap works better than the t-scan test in terms of all three criteria, and the Wilcoxon method finds less false positive bound regions than Tilemap and t-scan. While, the population-based method outperforms other three methods in all three criteria.

Later, we compared the receiving operating characteristic (ROC) curves (Bradley, 1997) and the error rate curves for the four methods. The ROC curve shows the true positive discovery rate ($sensitivity$) against the false positive discovery rate ($1 - specificity$) at the probe level, and the area under the curve (AUC) has been used as a summary measure of accuracy for multiple hypothesis tests. The error rate curve shows the proportion of incorrect probe calls, including both false positives and false negatives, against different cutoff values. The error rate has been used as a summary measure for the performance of a clustering method. The averaged ROC curve and error rate curve (over 20 datasets) were shown in Figure 18. In terms of AUCs, the four tests are ranked as the population-based test, Wilcoxon, Tilemap and t-scan, from the best to the worst. This is a little different from our impression obtained from Table 9, where it seems that Tilemap and t-scan outperform the Wilcoxon method. An interpretation for the difference is that AUC emphasizes more on the false discovery rate. It is indeed that the Wilcoxon method consistently produced smaller numbers of false positive bound regions than do the Tilemap and t-scan methods for this example. Next, we examined the error rates of the four methods. Figure 18(b) indicates that all the four methods have an optimal cutoff number around 996, which



(a) ROC Curves



(b) Error Rate

Figure 18. Averaged ROC curves and error rate curves (over 20 datasets) for the population-based, Tilemap, t-scan and Wilcoxon methods. (a) the ROC curve; (b) the error rate curve. The right panel plot provides a closer view for the area enclosed by the dotted line and the axes in the left panel plot.

is the number of true bound probes. It is remarkable that, among the four tests, the population-based test has consistently the lowest error rate at various cutoff values.

3.3.4 Microarray Data Analysis

In this section, we considered the application of the population-based test to microarray data for identification of differentially expressed genes. First, we tested the new method on a simulated example, which was modified from some examples used in the literature. Next, we applied the new method to a real dataset which is typical in this area.

A. A Simulated Example

This example is modified from examples of Qiu et al. (2005) and Liang et al. (2007b). It consists of multiple simulated datasets. Let n denote the number of genes included in each dataset, and let m denote the number of differentially expressed genes. The datasets were generated in the following way. First, generate an $n \times 8$ matrix and denote this matrix by $X = (x_{ij})$, $i = 1, \ldots, n$ and $j = 1, \ldots, 8$. The elements of this matrix are set as

$$x_{ij} = \begin{cases} \mu_i + \sigma_i z_{ij}, & \text{if } i = 1, \\ \mu_i + \rho \frac{\sigma_i}{\sigma_{i-1}}(x_{i-1,j} - \mu_{i-1}) + \sigma_i \sqrt{1-\rho^2} z_{ij}, & \text{if } i = 2,\ldots,\text{n}, \end{cases} \tag{3.32}$$

where μ_i, σ_i and z_{ij} are drawn independently from the distributions

$$\mu_i \sim U(-0.5, 0.5), \sigma_i \sim U(0.5, 1.5), z_{ij} \sim N(0, 1). \tag{3.33}$$

It is not difficult to show that $\text{Corr}(x_{i,j}, x_{i+1,j}) = \rho$ for $i = 1, \ldots, n-1$ and any j. In other words, there is constant correlation between the expression levels of adjacent genes.

Next, define

$$y_{ij} = \begin{cases} x_{ij} + \mu, & \text{for } i = 1, \ldots, m, \quad j = 5, \ldots, 8, \\ x_{ij}, & \text{otherwise,} \end{cases} \tag{3.34}$$

where μ is a constant representing the mean expression level difference of the differentially expressed genes and nondifferentially expressed genes. For each dataset $Y = (y_{ij})$, $i = 1, \ldots, n$ and $j = 1, \ldots, 8$, generated in the above procedure, the first m rows model the differentially expressed genes, the first 4 columns represent the control samples, and the last 4 columns represent the treatment samples.

For comparison, we calculated the test scores using the following three methods:

- Score A: the population-based test with the first three control and the first three treatment samples. The fourth control and the fourth treatment samples were discarded for illustration of superiority of this test. To prepare homogeneous control samples, the data are smoothed as follows,

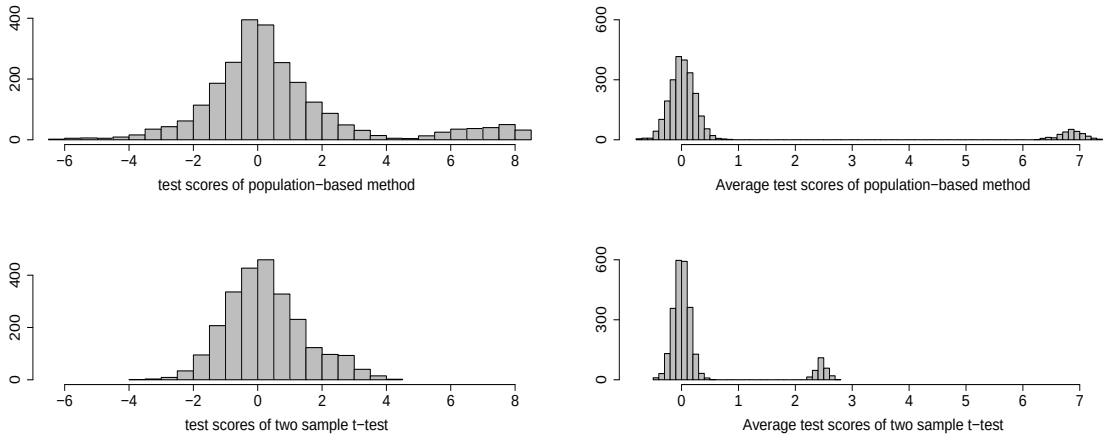
$$\boldsymbol{a}_k^* = \sum_{i \in C_k} w_i \boldsymbol{a}_i, \quad w_i = \frac{\widehat{\rho}_{i,k}^2}{\sum_{i \in C_k} \widehat{\rho}_{i,k}^2}, \tag{3.35}$$

  where $\boldsymbol{a}_i = (x_{i1}, x_{i2}, x_{i3}, y_{i1}, y_{i2}, y_{i3})'$, C_k denotes the set of neighboring genes of gene k, and $\widehat{\rho}_{i,k}$ denotes the Pearson correlation coefficient between the expression levels of gene k and gene i. For this example, we set $|C_k|$, the size of C_k, to be 10 based on the belief that there are about 10 genes co-expressed with each gene in the dataset. As indicated by our numerical results presented below, the population-based test is rather robust to the size of C_k.

- Score B: two-sample t-test with the first three control and the first three treatment samples.

- Score C: two-sample t-test with four control and four treatment samples.

Please note that the t-tests used in scores B and C are exact, as the data are generated from normal distributions. From this point of view, the comparison is a little unfair to score A. In the following, we first compare score A and score B, and then compare score A and score C, by looking at the power of the resulting multiple hypothesis tests.



(a) (b)

Figure 19. Histograms of test score for the simulated data. Left panel: histograms of test scores of one dataset. Right Panel: histograms of the average test scores of 50 simulated datasets.

a. Score A *versus* Score B

In this comparison, we fix $n = 2500$, $m = 250$, $\mu = 3$, $\rho = 0.3$, and $|C_k| = 10$, generated 50 different datasets in the above procedure and calculated scores A and B. Then the true FDRs (tFDRs) were calculated at different cutoff numbers. A cutoff number, τ, defines a classification criterion, classifying the τ genes with the highest test scores as "differentially" expressed genes. The computational results are summarized in Table 10. It shows that the tFDRs resultant from score A is lower than

those from score B for all the chosen cutoff numbers. This implies that the multiple hypothesis test based on Score A can have a higher power than that based on Score B. In addition, we compare the histograms of score A and Score B. Figure 19 (a) shows the histograms of the test scores for one of the 50 datasets, which indicates that the distribution of the test scores produced by the population-based method has relatively heavier and longer tail in the significant region than that produced by the two-sample t-test. Hence, the differentially and non-differentially expressed genes can be better separated by score A than by score B. Furthermore, we examined the histograms of the averaged test scores, where the average is taken for each gene over the 50 datasets. As shown in Figure 19 (b), the distance between differentially expressed genes and non-differentially ones is almost three times longer for the population-based method than for the two-sample t-test method. This supports again our claim that the differentially and non-differentially expressed genes can be better separated by score A for this example.

Table 10: Computational results for the datasets generated with $n = 2500$, $m = 250$, $\mu = 3$, $\rho = 0.3$ and $|C_k| = 10$. The value of tFDR and its standard deviation (the number in the parentheses) were calculated by averaging over 50 datasets.

	Cutoff number τ					
Method	50	100	200	300	400	500
Score A	.006(.002)	.005(.001)	.022(.002)	.201(.002)	.387(.001)	.507(.001)
Score B	.051(.005)	.078(.004)	.165(.004)	.294(.003)	.422(.002)	.520(.001)

b. Score A *versus* Score C

We compared Score A and Score C with various choices of the parameters n, ρ, μ, and $|C_k|$ in terms of specificity and sensitivity of multiple hypothesis tests. The specificity is defined as the proportion of correctly identified non-significant subjects, and the sensitivity is defined as the proportion of correctly identified significant subjects. In notations of Table 11, they are defined as

$$\text{specificity} = \frac{U}{U+V}, \quad \text{sensitivity} = \frac{S}{T+S}. \tag{3.36}$$

The specificity and sensitivity working together provide a good measure for the quality of multiple hypothesis tests. The average of sensitivity values over multiple datasets provides a natural estimate for the average power (Dudoit et al., 2003) of the multiple hypothesis test. The average power has been used to assess the quality of multiple hypothesis tests by more and more authors, e.g., Storey (2005), Wasserman and Roeder (2006), and Rubin et al. (2006).

Table 11: Notations for multiple hypotheses testing.

	Accept H_i	Reject H_i	Total
Genes for which H_i is true:	U	V	m'
Genes for which H_i is false:	T	S	m
Total	W	R	n

Effect of n In this comparison, we assessed the effect of n, the number of genes included in the dataset, on the effectiveness of score A. For this purpose, we fix $\mu = 3$, $\rho = 0.3$, $|C_k| = 10$, and considered three different pairs of (n, m): (1250,125), (2500, 250) and (5000, 250). For each pair of (n, m), we generated 50 different datasets,

Table 12: Computational results for the data generated with $\mu = 3$, $\rho = 0.3$, $|C_k| = 10$, and different choices of (n, m). The values of Spec. (specificity), Sens. (sensitivity/power) and their standard deviations (the numbers in the parentheses) were calculated by averaging over 50 datasets. $\Lambda(q)$ denotes a rejection region with the nominal value of FDR being q.

Setting	Method	Measure	$\Lambda(0.2)$	$\Lambda(0.1)$	$\Lambda(0.05)$
	Score A	Spec.	.975(.001)	.987(.001)	.993(.001)
n=1250		Sens.	**.958(.004)**	**.931(.006)**	**.889(.019)**
m=125	Score C	Spec.	.975(.002)	.990(.001)	.996(.000)
		Sens.	**.885(.008)**	**.772(.011)**	**.625(.017)**
	Score A	Spec.	.978(.001)	.989(.001)	.994(.000)
n=2500		Sens.	**.950(.003)**	**.922(.005)**	**.882(.007)**
m=250	Score C	Spec.	.976(.001)	.990(.001)	.996(.000)
		Sens.	**.889(.005)**	**.774(.009)**	**.634(.013)**
	Score A	Spec.	.990(.001)	.995(.000)	.997(.000)
n=5000		Sens.	**.911(.005)**	**.867(.006)**	**.787(.018)**
m=250	Score C	Spec.	.990(.001)	.997(.000)	.999(.000)
		Sens.	**.781(.007)**	**.618(.010)**	**.445(.013)**

calculated scores A and C, and applied the SA-FDR method to each of the datasets. The computational results were summarized in Table 12. The results show that Score A outperforms Score C consistently, as long as n is reasonably large. It is remarkable that, even with only three-fourths of control and treatment samples, the tests based on Score A still have higher sensitivity (power) than those based on Score C. The tests based on these two types of scores have about the same specificity values.

Table 13: Computational results for the datasets generated with $n = 2500$, $m = 250$, $\mu = 3$, $\rho = 0.3$, and different values of $|C_k|$. Refer to Table 12 for the notations used in this table.

Method	Setting	Measure	$\Lambda(0.2)$	$\Lambda(0.1)$	$\Lambda(0.05)$
	$\|C_k\| = 5$	Spec.	.976(.001)	.988(.001)	.993(.001)
		Sens.	**.939(.003)**	**.895(.005)**	**.835(.008)**
Score A	$\|C_k\| = 10$	Spec.	.978(.001)	.989(.001)	.994(.000)
		Sens.	**.950(.003)**	**.922(.005)**	**.882(.007)**
	$\|C_k\| = 15$	Spec.	.979(.001)	.989(.001)	.994(.000)
		Sens.	**.955(.003)**	**.931(.004)**	**.896(.007)**
Score C		Spec.	.976(.001)	.990(.001)	.996(.000)
		Sens.	**.889(.005)**	**.774(.009)**	**.634(.013)**

Effect of $|C_k|$ In this comparison, we assessed the effect of $|C_k|$ on the effectiveness of score A. For this purpose, we fix $n = 2500$, $m = 250$, $\mu = 3$, $\rho = 0.3$, and considered two different values of $|C_k|$, 5 and 15. For each value of $|C_k|$, we generated 50 different datasets, calculated scores A and C, and applied the SA-FDR method to each of the datasets. The results were summarized in Table 13. Combining with the results

presented in Table 12 for the case $|C_k| = 10$, we can see that at all different values of $|C_k|$, the tests based on Score A have a higher power than those based on Score C. In addition, the power of the tests based on Score A increases as $|C_k|$ increases. The latter observation implies that smoothing over neighboring subjects is indeed a useful idea for improving the power of multiple hypothesis tests.

Table 14: Computational results for the datasets generated with $n = 2500$, $m = 250$, $\mu = 3$, $|C_k| = 10$, and different values of ρ. Refer to Table 12 for the notations used in this table.

Setting	Method	Measure	$\Lambda(0.2)$	$\Lambda(0.1)$	$\Lambda(0.05)$
$\rho = 0.0$	Score A	Spec.	.978(.001)	.988(.001)	.994(.000)
		Sens.	**.949(.002)**	**.918(.004)**	**.873(.006)**
	Score C	Spec.	.974(.001)	.990(.001)	.996(.000)
		Sens.	**.894(.005)**	**.775(.009)**	**.619(.015)**
$\rho = 0.6$	Score A	Spec.	.979(.001)	.990(.001)	.994(.001)
		Sens.	**.948(.003)**	**.913(.005)**	**.859(.008)**
	Score C	Spec.	.976(.001)	.990(.001)	.996(.000)
		Sens.	**.881(.006)**	**.767(.010)**	**.612(.015)**

Effect of ρ In this comparison, we assessed the effect of gene dependency on the effectiveness of score A. For this purpose, we fix $n = 2500$, $m = 250$, $\mu = 3$, and $|C_k| = 10$, and considered two different values of ρ, 0 and 0.6. For each value of ρ, we generated 50 different datasets, calculated scores A and C, and applied the SA-FDR method to each of the datasets. The results were summarized in Table 14. Combining with the results presented in Table 12 for the case $\rho = 0.3$, it can be seen

that the performance of Score A is almost independently of the value of ρ. At all different values of ρ, the tests based on Score A have higher powers than those based on Score C.

Table 15: Computational results for the datasets generated with $n = 2500$, $m = 250$, $\rho = 0.3$, $|C_k| = 10$, and different values of μ. Refer to Table 12 for the notations used in this table.

Setting	Method	Measure	$\Lambda(0.2)$	$\Lambda(0.1)$	$\Lambda(0.05)$
$\mu = 2$	Score A	Spec.	.972(.002)	.985(.001)	.993(.001)
		Sens.	**.738(.012)**	**.620(.023)**	**.445(.036)**
	Score C	Spec.	.984(.001)	.995(.001)	.999(.000)
		Sens.	**.578(.014)**	**.359(.016)**	**.181(.017)**
$\mu = 4$	Score A	Spec.	.982(.001)	.992(.001)	.996(.000)
		Sens.	**.993(.001)**	**.988(.001)**	**.975(.003)**
	Score C	Spec.	.974(.001)	.989(.001)	.995(.000)
		Sens.	**.978(.002)**	**.931(.004)**	**.851(.007)**

Effect of μ In this comparison, we assessed the effect of the expression levels of differentially expressed genes on the effectiveness of score A. For this purpose, we fix $n = 2500$, $m = 250$, $\rho = 0.3$, and $|C_k| = 10$, and considered two different values of μ, 2 and 4. For each value of μ, we generated 50 different datasets, calculated scores A and C, and applied the SA-FDR method to each of the datasets. The computational results were summarized in Table 15. Combining with the results presented in Table 12 for the case $\mu = 3$, it can be seen that score A outperforms score C irrespective of the value of μ. The improvement is especially significant when

the value of μ is small.

In summary, the population-based test can outperform the two-sample t-test in almost all scenarios for this example. In addition, for this example, to achieve the same or even higher testing power while maintaining the same level of specificity, the population-based test requires less than 3/4 of the control and treatment samples than does the two-sample t-test. This implies that use of the population-based test can potentially lead to a great saving of experiment cost.

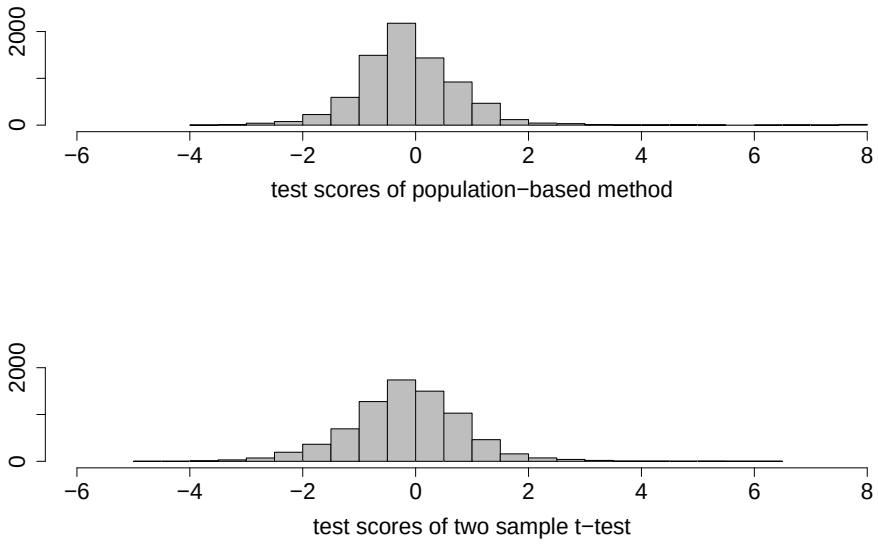


Figure 20. Test score of HIV dataset for population-based method and two-sample t-test.

B. HIV Data

This dataset includes $n = 7680$ genes. It concerns the difference of gene expression levels of uninfected cells and HIV-infected cells (Wout et al., 2003). As described by Gottardo et al. (2006), the experiment was carried out on four different slides

under the same RNA preparation. Each slide reported on the same set of 7680 genes. Among them, 12 known differentially expressed HIV-1 genes are included as positive controls. Dye-swapped hybridizations technique was used to compensate dye bias in this experiment. Two of the four slides were hybridized with the green dye (Cy3) for the control(uninfected) samples and the red dye (Cy5) for the treatment(HIV-infected) samples, the dyes were reversed on the other two slides. Totally, 4 control and 4 treatment samples were included in this dataset. The raw data was downloaded from http://wwwstatubcca/~raph/PublicFiles/.

The raw data were pre-processed as in Gottardo et al. (2006). They were first quantile-normalized (Bolstad et al., 2003), log-transformed, and then the mean of the log expression values was adjusted to zero for each chip. Afterwards, the two-sample t-test was applied to the pre-processed data. On the other hand, for the population-based test, the data were further processed as prescribed in section 3.3.2 B. As for the simulated example, we set the size of C_k, the number of neighboring genes, to be 10, and smoothed the gene expression levels using (3.35).

Figure 20 displays the histograms of the test scores produced by the two methods. It is easy to see that the histogram produced by the population-based test has a relatively shorter left-tail and longer right-tail than that produced by the two-sample t-test. This difference implies that the population-based test can have a higher power than the two-sample t-test. The SA-FDR methods were applied to the test scores resultant from the two methods. By controlling the nominal FDR at 10%, only 16 differentially expressed genes were detected using the test scores produced by the two-sample t-test. This is too conservative, comparing to 33, 86 and 81 genes found by the software BRIDGE (Gottardo et al., 2006), the empirical Bayes gamma-gamma model (Newton et al., 2001) and the empirical Bayes lognormal-normal model (Kendziorski et al., 2003), respectively. Using the test scores resultant from the population-based

test, 50 genes were identified as being differentially expressed. It is remarkable that, not only the built-in 12 positive control genes are covered by the 50 significant genes, but also their test scores are ranked among the top 13 test scores. This does not happen for the two-sample t-test. All the above evidences indicate the effectiveness of the population-based test for detecting differential expressed genes with microarray experiments.

3.3.5 Discussion

We have proposed a population-based method for evaluation of test scores for each individual subject involved in a multiple hypothesis test. The method consists of two key steps, smoothing over neighboring subjects and density estimation over control samples, both of which allow for the use of population information of the subjects. The new method is tested on both the gene expression data and the ChIP-chip data. The numerical results indicate that use of population information can significantly improve the power of multiple hypothesis tests. In other words, the proposed method can significantly reduce the number of duplicates of the routine microarray and ChIP-chip experiments and thus the experimental cost, while maintaining the same level of statistical power in the analysis.

The strength of the new method comes from two sources, smoothing over neighboring subjects and density estimation over control samples. Smoothing over neighboring subjects effectively reduces variation of the experimental samples, while causing only negligible bias to the mean of the samples as long as the size of neighboring subjects set is reasonable. As shown by our numerical examples, smoothing over neighboring subjects samples does improve the power of multiple hypothesis tests.

Nonparametric density estimation over control samples provides us a robust way of test scores evaluation, which relaxes the distribution assumption for the experi-

mental samples from normality to a location-scale family. Moreover, it automatically accounts for the extremeness of a large size sample with its built-in mechanism. This is beyond the ability of the two-sample t-test and other tests based on the individual subject samples.

At last, we would like to mention that the idea of using population information of samples to improve the power of multiple hypothesis tests is not brand new. In the literature, a variety of estimators of variance, which borrow information across subjects, have been proposed, although the idea of using population information was not stated there explicitly. These estimators can be classified into two categories, the empirical Bayes approach-based estimators and the James-Stein shrinkage approach-based estimators. The work falling into the first category are L*ö*nnstedt and Speed (2002), Wright and Simon (2003), Smyth (2004), and Ji and Wong (2005). The work belonging to the second category include Cui et al. (2005) and Opgen-Rhein and Strimmer (2007). In these work, the variance of each individual subject samples was estimated by combining information of all subjects samples. Although the population information of samples is used very limitedly in these approaches, their numerical results do show that use of population information can improve the power of multiple hypothesis tests.

CHAPTER IV

SUMMARY AND FUTURE RESEARCH

4.1 Summary

In this dissertation, we have proposed a population SAMC algorithm. Compared with the single chain SAMC, the new algorithm provides a more efficient self-adjusting mechanism, and since it works on a population of chains, more global, advanced MCMC operators, such as the crossover operator of the genetic algorithm can be included into simulations, which can further improve the convergence of the algorithm. Under mild conditions, the convergence of Pop-SAMC has been proved. A theory indicating Pop-SAMC converges faster than single chain SAMC has been established. The theoretical results are illustrated by a multimodal example. Finally, the effectiveness of the new algorithm for Bayesian model selection problems is examined through a change-point identification problem and a large-p linear regression variable selection problem. The numerical results suggest that the Pop-SAMC algorithm can outperforms both the single chain SAMC and RJMCMC no matter the model space is complex or simple.

The second part of this dissertation focuses on new methodologies development of ChIP-chip data analysis. Two new methods have been proposed, the Bayesian latent model and a population based nonparametric testing method. Both methods are applied to real and simulated dataset along with the comparison with the existing methods. The numerical results show that, the Bayesian latent model can outperforms most of the existing methods, especially when the data contain outliers, and the use of population information of samples can significantly improve the power of multiple hypothesis test.

4.2 Future Research

With the development of next generation massively parallel sequencing technologies, a new technology named ChIP-seq, which combines chromatin immunoprecipitation with massively pareallel shot-read sequencing, has been invented. Compared with the ChIP-chip, ChIP-seq can provide higher resolution mapping and stronger protein binding sites signals. therefore, it has the potential to replace ChIP-chip method in analyzing protein-DNA interactions.

A typical ChIP-seq data set consist of tens of millions of sequence reads, which are generated from the ends of DNA fragments. Currently, most existing methodologies to analyze ChIP-seq data is based on Poisson model. Such as, MACS (Zhang et al., 2008) and CisGenome (Ji et al., 2008). In CisGenom, a marginal version of Poisson model, the negative binomial background model is developed to analyze one sample case, and a conditional version of Poisson model, the binomial model is developed for two sample case. They use the small reads count from nonbinding regions to estimate the model parameters, and based on the estimate null model, they may calculate FDR for each level of counts.

While these model based methods have been proved their effectiveness for certain ChIP-seq data analysis, these methods might be fail in the situation when the data is not consitent with the underling models. Considering the large dataset generated from ChIP-seq experiments, it is desirable to develop some nonparametric approaches, which do not have model assumption for the data. An interesting research direction is to find a nonparametric method to construct the null distribution, and caculate FDR accordingly.

REFERENCES

Andrieu, C., Moulines, *É*. and Priouret, P. (2005). Stability of stochastic approximation under verifiable conditions. *SIAM Journal on Control and Optimization* **44**, 283-312.

Barry, D. and Hartigan, J. A. (1993). A Bayesian analysis for change point problems. *Journal of the American Statistical Association* **88**, 309-319.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society B* **57**, 289-300.

Benjamini, Y. and Yekutieli, D. (2001). On the control of false discovery rate in multiple testing under dependency. *Annals of Statistics* **29**, 1165-1188.

Benveniste, A., Métivier, M., and Priouret, P. (1990). *Adaptive Algorithms and Stochastic Approximations*. New York: Springer-Verlag.

Bernstein, B. E., Kamal, M., Lindblad-Toh, K., Bekiranov, S., Bailey, D. K., Huebert, D. J., McMahon, S., Karlsson, E. K., Kulbokas E. J. 3rd, Gingeras, T. R., Schreiber, S. L., Lander, E. S. (2005). Genomic maps and comparative analysis of histone modifications in human and mouse. *Cell* **120**, 169-181.

Bertone, P., Stolc, V., Royce, T. E., Rozowsky, J. S., Urban, A. E., Zhu, X., Rinn, J. L., Tongprasit, W., Samanta, M., Weissman, S., Gerstein, M., Snyder, M. (2004). Global identification of human transcribed sequences with genome tiling arrays. *Science* **306**(5705), 2242-2246.

Besag, J. and Kooperberg, C. (1995). On conditional and intrinsic autoregrssions. *Biometrika* **82**, 733-746.

Bolstad, B. M., Irizarry, R. A., Astrand, M., Speed, T. P. (2003). A comparison of normalization methods for high density ologonucleotide array data based on variance and bias. *Bioinformatics* **19**(2), 185-193.

Bradley, A. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition* **30**, 1145-1159.

Cai, A., Tsay, R. S., and Chen, R. (2009). Variable selection in linear regression with many predictors. *Journal of Computational and Graphical Statistics* **18**(3), 573-591.

Carroll, J. S., Liu, X. S., Brodsky, A. S., et al. (2005). Chromosome-wide mapping of estrogen receptor binding reveals long-range regulation requiring the forkhead protein foxal. *Cell* **122**, 33-43.

Cawley, S., Bekiranov, S., Ng, H. H., Kapranov, P., Sekinger, E. A., Kampa, D., Piccolboni, A., Sementchenko, V., Cheng, J., Williams, A. J., Wheeler, R., Wong, B., Drenkow, J., Yamanaka, M., Patel, S., Brubaker, S., Tammana, H., Helt, G., Struhl, K., Gingeras, T. R. (2004). Unbiased mapping of transcription factor binding sites along human chromosomes 21 and 22 points to widespread regulation of noncoding RNAs. *Cell* **116**, 499-509.

Chen, H.-F. (2002) *Stochastic Approximation and Its Applications*, Boston: Kluwer Academic Publishers.

Cui, X., Hwang, J.T.G., Qiu, J., Blades, N.J., Churchill, G.A. (2005). Improved statistical tests for differential gene expression by shrinking variance components

estimates. *Biostatistics* **6**, 1, 59-75.

Dudoit, S., Shaffer, J.P., and Boldrick, J.C. (2003). Multiple hypothesis testing in microarray experiments. *Statistical Science* **18**, 71-103.

Efron, B. (2004). Large-scale simultaneous hypothesis testing: the choice of a null hypothesis. *Journal of the American Statistical Association* **99**, 96-104.

Fernández, C., Ley, E. Steel, M. F.J. (2001). Benchmark priors for Bayesian model averaging. *Journal of Econometrics* **100**, 381-427.

Gelman, A., Rubin, D. B.(1992). Inference from iterative simulation using multiple sequences (with discussion). *Statistical Science* **7**, 457-511.

George, E. I. and Foster, D. P. (2000). Calibration and empirical Bayes variable selection. *Biometrika* **87**, 731-747.

Geyer, C. J. (1991). Markov chain Monte Carlo maximum likelihood. *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface (pp.153-163)*, Seattle, WA: Interface Foundation.

Gilks, W. R. (1994). Adaptive direction sampling. *The Statistician* **43**, 179-189.

Gottardo, R., Li, W., Johnson, W. E., Liu, X. S. (2008). A flexible and powerful Bayesian hierarchical model for ChIP-chip experiments. *Biometrics* **64**, 468-478.

Gottardo, R., Raftery, A.E., Yeung, K.Y., and Bumgarner, R.E. (2006). Bayesian robust inference for differential gene expression in microarrays with multiple samples. *Biometrics* **62**, 10-18.

Green, P. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* **82**, 711-732.

Hall, P., Lahiri S.N. and Truong Y.K. (1995). On bandwidth choice for density estimation with dependent data. *The Annals of Statistics* **23**, 2241-2263.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.

Huber, W., Toedling, J., Steinmetz, L. M. (2006). Transcript mapping with high-density oligonucleotide tiling arrays. *Bioinformatics* **22**(16), 1963-1970.

Hubert, L., Arabie, P. (1985). Comparing partitions. *Journal of Classification* **2**, 193-218.

Hukushima, K. and Nemoto, K. (1996). Exchange Monte Carlo method and application to spin glass simulations. *Journal of the Physical Society of Japan* **65**, 1604-1608.

Humburg, P., Bulger, D., Stone, G. (2008). Parameter estimation for robust HMM analysis of ChIP-chip data. *BMC Bioinformatics* **9**, 343.

Ji, H., Jiang H., Ma, W., Johnson, D. S., Myers, R. M. and Wong, W. H. (2008). An integrated software system for analyzing ChIP-chip and ChIP-seq data. *Nature Biotechnology* **26**, 1293 - 1300.

Ji, H., Wong, W. H. (2005). TileMap: create chromosomal map of tiling array hybridizations. *Bioinformatics* **21**(18), 3629-3636.

Keles, S. (2007). Mixture modeling for genome-wide localization of transcription factors. *Biometrics* **63**, 10-21.

Keles, S., Van Der Laan, M. J., Dudoit, S., Cawley, S. E. (2006). Multiple testing methods for ChIP-chip high density oligonucleotide array data. *Journal of Computational Biology* **13**(3), 579-613.

Kendziorski, C., Newton, M., Lan, H., and Gould, M.N. (2003). On parameter empirical Bayes methods for comparing multiple groups using replicated gene expression profiles. *Statistics in Medicine* **22**, 3899-3914.

Lan, H., Chen, M., Flowers, J. B., Yandell, B. S., Stapleton, D. S., Mata, C. M., Mui, E. T., Flowers, M. T., Schueler, K. L., Manly, K. F., Williams, R. W., Kendziorski, K. and Attie, A. D. (2006). Combined expression trait correlations and expression quantitative trait locus mapping. *PLoS Genetics* **2**, e6.

Liang, F. (2009). Improving SAMC using smoothing methods: theory and applications to Bayesian model selection problems. *The Annals of Statistics* **37**, 2626-2654.

Liang, F., Liu, C. and Carroll, R. J. (2007a). Stochastic approximation in Monte Carlo computation. *Journal of the American Statistical Association* **102**, 305-320.

Liang, F., Liu, C. and Carroll, R. J. (2010). *Advanced Markov Chain Monte Carlo: Learning from Past Samples*. Chichester, West Sussex, UK: John Wiley and Sons Ltd.

Liang, F., Liu, C., and Wang, N. (2007b). A robust sequential Bayesian method for identification of differentially expressed genes. *Statistica Sinica* **17**, 571-597.

Liang, F. and Wong, W. H. (2000). Evolutionary Monte Carlo sampling: applications to C_p model sampling and change-point problem. *Statistica Sinica* **10**, 317-342.

Liang, F. and Wong, W. H. (2001). Real-parameter evolutionary sampling with applications in Bayesian mixture models. *Journal of the American Statistical Association* **96**, 653-666.

Liang, F., Zhang, J. (2008). Estimation the false discovery rate using the stochastic approximation algorithm. *Biometrika* **95**(4), 961-977.

Liu, J. S., Liang, F. and Wong, W. H. (2000). The use of multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association* **95**, 121-134.

Li, W., Meyer, C. A., Liu, X. S. (2005). A hidden Markov model for analayzing ChIP-chip experiments on genome tiling arrays and its application to p53 binding sequences. *Bioinformatics* **21**(Suppl), i274-i282.

Lönnstedt, I. and Speed, T. (2002). Replicated microarray data. *Statistica Sinica* **12**, 31-46

Munch, K., Gardner, P. P., Arctander, P., Krogh, A. (2006). A hidden Markov model approach for determining expression from genomic tiling microarrays. *BMC Bioinformatics* **7**, 239.

Müller, P. (1991). A generic approach to posterior integration and gibbs sampling. Technical Report, Department of Statistics, Purdue University, West Lafayette, IN.

Newton, M.C., Kendziorski, C.M., Richmond, C.S., Balattner, F.R., and Tsui, K.W. (2001). On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data. *Journal of Computational Biology* **8**, 37-52.

Opgen-Rhein, R., Strimmer, K. (2007). Accurate ranking of differentially expressed genes by a distribution-free shrinkage approach. *Statistical Applications in Genetics and Molecular Biology* **6**, Article 9.

Phillips, D. B. and Smith, A. F.M. (1996). Bayesian model comparison via jump diffusions. *Markov Chain Monte Carlo in Practice* (W.R. Gilks, S. Richardson and D.J. Spiegelhalter, eds.), 215-239, London: Chapman & Hall.

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., Gifford, D. K. (2006). High-resolution computational models of genome binding events. *Nature Biotechnology* **24**(8), 963-970.

Qiu, X., Klebanov, L., and Yakovlev, A. (2005). Correlation between gene expression levels and limitations of the empirical Bayes methodology for finding differentially expressed genes. *Statistical Applications in Genetics and Molecular Biology* **4**(1), Article 34.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* **77**, 257-286.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* **66**, 846-850.

Reiss, D. J., Facciotti, M. T., Baliga, N. S. (2008). Model-based deconvolution of genome-wide DNA binding. *Bioinformatics* **24**(3), 396-403.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics* **22**, 400-407.

Roberts, G. O. and Tweedie, R. L. (1996). Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. *Biometrika* **83**, 95-110.

Rubin, D., Dudoit, S. and van der Laan, M.J. (2006). A method to increase the power of multiple testing procedures through sample splitting.
Available at http://www.bepress.com/ucbbiostat/paper171.

Smyth, G.K. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology* **3**(1), Article 3.

Song, J. and Hart, J. D. (2009). Bootstrapping in a high dimensional but very low-sample size problem. *Journal of Statistical Computation and Simulation*, to appear.

Storey, J. D. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society B* **64**, 479-498.

Storey, J. D. (2005). The optimal discovery procedure: A new approach to simultaneous significance testing. *UW Biostatistics Working Paper Series*, Working Paper 259.

Storey, J. D., Taylot, J. E. and Siegmund, D. (2004). Strong control, conservative point estimation, and simultaneous conservative consistency of false discovery rates: A unified approach. *Journal of the Royal Statistical Society B* **66**, 187-205.

Wasserman, L. and Roeder, K. (2006). Weighted hypothesis testing. Technical report, Department of Statistics, Carnegie Mellon University, Pittsburgh, PA.

Welch, B. L. (1938). The significance of the difference between two means when the population variances are unequal. *Biometrika* **29**, 350-362.

Wout, A. B., Lehrma, G. K., Mikheeva, S. A., O'Keeffe, G. C., Katze, M. G., Bumgarner, R. E., Geiss, G. K., and Mullins, J. I. (2003). Cellular gene expression upon

human immunodeficiency virus type 1 infection of $CD4^{+}$-*T-cell* lines. *Journal of Virology* **77**, 1392-1402.

Wright, G. W. AND Simon, R. M. (2003). A random variance model for detection of differential gene expression in small microarray experiments. *Bioinformatics 19*, 2448-2455.

Younes, L. (1999). On the convergence of Markovian stochastic algorithms with rapidly decreasing ergodicity rates. *Stochastics and Stochastics Reports* **65**, 177-228.

Zellner, A. (1986). On assessing prior distributions and Bayesian regression analysis with g-prior distributions. in *Bayesian Inference and Decision Techniques: Essays in Honor of Bruno de Finetti*, eds. P. K. Goel and A. Zellner, Amsterdam: North-Holland/Elsevier, pp. 233-243.

Zhang, D., Lin, Y. and Zhang, M. (2009). Penalized orthogonal-components regression for large p small n data. *Electronic Journal of Statistics* **3**, 781-796.

Zhang, X., Clarenz, O., Cokus, S., Bernatavichute, Y. V., Goodrich, J., Jacobsen, S. E. (2007). Whole-genome analysis of histone h3 lysine 27 trimethylation in arabidopsis. *PLoS Biol* **5**(5), e129.

Zhang, X., Yazakij, J., Sundaresan, A., Cokus, S., Chan, S. W. L., Chen, H., Henderson, I. R., Shinn, P., Pellegrini, M., Jacobsen, S. E., Ecker, J. R. (2006). Genome-wide high-resolution mapping and functional analysis of DNA methylation in arabidopsis. *Cell* **126**, 1189-1201.

Zhang, Y., Liu, T., Meyer, C.A., Eeckhoute, J., Johnson, D. S., Bernstein, B. E.,

Nussbaum, C., Myers, R.M., Brown, M., Li, W. and Liu, X. S. (2008). Model-based analysis of ChIPseq(MACS). *Genome Biology* **9**(9), R137.

Zheng, M., Barrera, L. O., Ren, B., Wu, Y. N. (2007). ChIP-chip: Data, model, and analysis. *Biometrics* **63**(3), 787-796.

APPENDIX A

PROOFS IN CHAPTER II

The appendix is organized as follows. In Section A.1, we consider a general stochastic approximation Monte Carlo algorithm, give a theorem for its convergence which has been proved in the literature, and prove a theorem for its convergence rate. In Section A.2, we proved Theorem 2.2.1, Theorem 2.2.2 and Corollary 2.2.3.

A.1 Convergence of a General Stochastic Approximation Monte Carlo Algorithm

Let $x_t = (x_t^{(1)}, \ldots, x_t^{(\kappa)})$ be the collection of the samples generated by a MH kernel at iteration t, $\mathbf{f}_{\theta_t}(x)$ be the invariant distribution of the MH kernel, $h(\theta) = \int_{\mathcal{X}^k} H(\theta, x)\mathbf{f}_\theta(dx)$, and $\xi_{t+1} = H(\theta_t, x_{t+1}) - h(\theta_t)$. The SAMC algorithm can then be expressed in a more general form by replacing (2.7) by (A.1),

$$\theta^* = \theta_t + \gamma_{t+1}h(\theta_t) + \gamma_{t+1}\xi_{t+1}. \tag{A.1}$$

The convergence of the general stochastic approximation Monte Carlo algorithm is analyzed by Liang (2009) under the following conditions.

Conditions on the step-sizes

(A_1) The sequence $\{\gamma_t\}_{t=0}^{\infty}$ is non-increasing, positive and satisfies the condition (2.3).

Drift conditions on the transition kernel For a function $g : \mathcal{X} \to \mathbb{R}^d$, define the norm $\|g\|_V = \sup_{x\in\mathcal{X}} \frac{|g(x)|}{V(x)}$, and define the set $\mathcal{L}_V = \{g : \mathcal{X} \to \mathbb{R}^d, \|g\|_V < \infty\}$.

Let P_θ be the joint transition kernel for generating the samples x at each iteration by ignoring the subscript t. Let $\mathcal{X}$ be the sample space, let A be a measurable set belong to $\mathcal{B}_\mathcal{X}$, and $\mathcal{B}_\mathcal{X}$ is the σ-algebra generated by all subsets of $\mathcal{X}$.

(A_2) The transition kernel P_θ is irreducible and aperiodic for any $\theta \in \Theta$. There exist a function $V : \mathcal{X} \to [1, \infty)$ and constants $\alpha \geq 2$ and $\beta \in (0, 1]$ such that,

(i) For any $\theta \in \Theta$, there exist a set $C \subset \mathcal{X}$, an integer l, constants $0 < \lambda < 1$, b,ς, $\delta > 0$ and a probability measure ν such that

- $P_\theta^l V^\alpha(x) \leq \lambda V^\alpha(x) + bI(x \in C), \quad \forall x \in \mathcal{X}.$ (A.2)
- $P_\theta V^\alpha(x) \leq \varsigma V^\alpha(x), \quad \forall x \in \mathcal{X}.$ (A.3)
- $P_\theta^l(x, A) \geq \delta\nu(A), \quad \forall x \in C, \ \forall A \in \mathcal{B}_\mathcal{X}.$ (A.4)

(ii) There exists a constant c_1 such that for all $x \in \mathcal{X}$ and $\theta, \theta' \in \Theta$,

- $\|H(\theta, x)\| \leq c_1 V(x).$ (A.5)
- $\|H(\theta, x) - H(\theta', x)\| \leq c_1 V(x)\|\theta - \theta'\|^\beta.$ (A.6)

(iii) There exists a constant c_2 such that for all $\theta, \theta' \in \Theta$,

- $\|P_\theta g - P_{\theta'} g\|_V \leq c_2 \|g\|_V |\theta - \theta'|^\beta, \quad \forall g \in \mathcal{L}_V.$ (A.7)
- $\|P_\theta g - P_{\theta'} g\|_{V^\alpha} \leq c_2 \|g\|_{V^\alpha} |\theta - \theta'|^\beta, \quad \forall g \in \mathcal{L}_{V^\alpha}.$ (A.8)

Lyapunov condition on $h(\theta)$ Let $\mathcal{L} = \{\theta \in \Theta : h(\theta) = \mathbf{0}\}$.

(A_3) The function $h : \Theta \to \mathbb{R}^d$ is continuous, and there exists a continuously differentiable function $v : \Theta \to [0, \infty)$ such that $\dot{v}(\theta) = \nabla^T v(\theta) h(\theta) < 0$, $\forall \theta \in \mathcal{L}^c$ and $\sup_{\theta \in Q} \dot{v}(\theta) < 0$ for any compact set $Q \subset \mathcal{L}^c$.

Convergence of the General SAMC Algorithm Let $\mathcal{P}_{x_0,\theta_0}$ denote the probability measure of the Markov chain $\{(x_t, \theta_t)\}$, started in (x_0, θ_0), and implicitly defined by the sequences $\{\gamma_t\}$. Also define $D(z, A) = \inf_{z' \in A} \|z - \boldsymbol{z}'\|$.

Theorem A.1 (Liang, 2009) *Assume the conditions* (A_1), (A_2) *and* (A_3) *hold, and* $\sup_{x\in\mathcal{X}} V(x) < \infty$. *Let the sequence* $\{\theta_t\}$ *be defined as in (A.1). Then for all* $(x_0, \theta_0) \in \mathcal{X} \times \Theta$,

$$\lim_{t\to\infty} D(\theta_t, \mathcal{L}) = 0, \qquad \mathcal{P}_{x_0,\theta_0} - a.e.$$

Convergence Rate of the General SAMC Algorithm To assess the convergence rate of the algorithm, we need the following additional condition:

(A_4) The mean field function $h(\theta)$ is measurable and locally bounded. There exist a constant $\delta > 0$ and θ_* such that for all $\theta \in \Theta$,

$$(\theta - \theta_*)^T h(\theta) \le -\delta\|\theta - \theta_*\|^2. \tag{A.9}$$

Lemma A.1 is a restatement of Propositions 6.1 of Andrieu, Moulines and Priouret (2005).

Lemma A.1 *(Andrieu et al., 2005) Assume* Θ *is compact and the drift condition* (A_2) *holds. Then for any* $\theta \in \Theta$, *the Poisson equation* $u(\theta, x) - P_\theta u(\theta, x) = H(\theta, x) - h(\theta)$ *has a solution, where* $P_\theta u(\theta, x) = \int_{\mathcal{X}} u(\theta, x')P_\theta(x, x')dx'$. *In addition, there exists a constant* C *such that for all* $\theta, \theta' \in \Theta$,

$$\begin{aligned} &(i) \quad \|P_\theta u(\theta, x)\| \le CV(x), \\ &(ii) \quad \|P_\theta u(\theta, x) - P_{\theta'} u(\theta', x)\| \le C\|\theta - \theta'\|V(x). \end{aligned} \tag{A.10}$$

The following theorem gives a L^2 upper bound for the approximation error of θ_t. A similar result for stochastic approximation MCMC algorithms is given in Benveniste et al. (1990, §1.10.2), but under different conditions and the proofs are different.

Theorem A.2 *Assume* Θ *is compact, the conditions* (A_1), (A_2), (A_3), *and* (A_4) *hold,* $\sup_{x\in\mathcal{X}} V(x) < \infty$, *and the gain factor sequence is chosen in the form*

$$\gamma_t = \frac{T_0}{\max\{T_0, t^\xi\}}, \tag{A.11}$$

where $1/2 < \xi \leq 1$*, and* T_0 *is a constant. Let the sequence* $\{\theta_n\}$ *be defined by (A.1). There exists a constant* λ *such that*

$$E\|\theta_t - \theta_*\|^2 \leq \lambda\gamma_t.$$

PROOF: Writing $\epsilon_t = \theta_t - \theta_*$, and following from the Poisson equation

$$H(\theta, x) = h(\theta) + u(\theta, x) - P_\theta u(\theta, x),$$

we have

$$\|\epsilon_{t+1}\|^2 = \|\epsilon_t\|^2 + 2\gamma_{t+1}\epsilon_t^T h(\theta_t) + 2\gamma_{t+1}\epsilon_t^T\big[u(\theta_t, x_{t+1}) - P_{\theta_t}u(\theta_t, x_{t+1})\big] + \gamma_{t+1}^2\|H(\theta_t, x_{t+1})\|^2. \tag{A.12}$$

Then, decomposing $u(\theta_t, x_{t+1}) - P_{\theta_t}u(\theta_t, x_{t+1})$ as follows:

$$\begin{aligned} u(\theta_t, x_{t+1}) - P_{\theta_t}u(\theta_t, x_{t+1}) = {} & u(\theta_t, x_{t+1}) - P_{\theta_t}u(\theta_t, x_t) + P_{\theta_{t-1}}u(\theta_{t-1}, x_t) \\ & - P_{\theta_t}u(\theta_t, x_{t+1}) + P_{\theta_t}u(\theta_t, x_t) - P_{\theta_{t-1}}u(\theta_{t-1}, x_t). \end{aligned}$$

Note that

$$E\left\{\gamma_{t+1}\epsilon_t\big[u(\theta_t, x_{t+1}) - P_{\theta_t}u(\theta_t, x_t)\big]\right\} = 0, \tag{A.13}$$

and that, by $(A.10)$ in Lemma A.1, there exists a constant c_1 such that

$$\epsilon_t\|P_{\theta_t}u(\theta_t, x_t) - P_{\theta_{t-1}}u(\theta_{t-1}, x_t)\| \leq c_1\gamma_{t+1}, \tag{A.14}$$

and

$$\epsilon_t\big[P_{\theta_{t-1}}u(\theta_{t-1}, x_t) - P_{\theta_t}u(\theta_t, x_{t+1})\big] = z_t - z_{t+1} + (\epsilon_{t+1} - \epsilon_t)P_{\theta_t}u(\theta_t, x_{t+1}), \tag{A.15}$$

where $z_t = \epsilon_t P_{\theta_{t-1}}u(\theta_{t-1}, x_t)$, and

$$\|(\epsilon_{t+1} - \epsilon_t)P_{\theta_t}u(\theta_t, x_{t+1})\| \leq c_2\gamma_{t+1}, \tag{A.16}$$

for some constant c_2. Now we decompose $z_t - z_{t+1}$ as

$$z_t - z_{t+1} = \epsilon_t P_{\theta_{t-1}} u(\theta_{t-1}, x_t) - \epsilon_t P_{\theta_t} u(\theta_t, x_t) + \epsilon_t P_{\theta_t} u(\theta_t, x_t) - \epsilon_{t+1} P_{\theta_t} u(\theta_t, x_t)$$
$$+ \epsilon_{t+1} P_{\theta_t} u(\theta_t, x_t) - \epsilon_{t+1} P_{\theta_t} u(\theta_t, x_{t+1}).$$

As in (A.14), we have

$$\|\epsilon_t P_{\theta_{t-1}} u(\theta_{t-1}, x_t) - \epsilon_t P_{\theta_t} u(\theta_t, x_t)\| \leq c_3 \gamma_{t+1}, \tag{A.17}$$

for some constant c_3. By Lemma A.1, we have

$$\|\epsilon_t P_{\theta_t} u(\theta_t, x_t) - \epsilon_{t+1} P_{\theta_t} u(\theta_t, x_t)\| \leq c_4 \gamma_{t+1}, \tag{A.18}$$

for some constant c_4. In addition, we have

$$E\left(\epsilon_{t+1} P_{\theta_t} u(\theta_t, x_t) - \epsilon_{t+1} P_{\theta_t} u(\theta_t, x_{t+1})\right) = 0. \tag{A.19}$$

Thus, from (A.13)–(A.16) and (A.9), we deduce that

$$E\|\epsilon_{t+1}\|^2 \leq (1 - 2\delta\gamma_{t+1}) E\|\epsilon_t\|^2 + c_5 \gamma_{t+1}^2 + 2\gamma_{t+1} E(z_t - z_{t+1}), \tag{A.20}$$

for some constant c_5. Furthermore, from (A.17)–(A.19), we deduce that

$$E\|\epsilon_{t+1}\|^2 \leq (1 - 2\delta\gamma_{t+1}) E\|\epsilon_t\|^2 + C\gamma_{t+1}^2, \tag{A.21}$$

for some constant C.

Lemma A.2 *Suppose t_0 is such that $1 - 2\delta\gamma_{t+1} \geq 0$ for all $t \geq t_0$. and*

$$\frac{1}{\gamma_{t+1}} - \frac{1}{\gamma_t} < 2\delta. \tag{A.22}$$

Let $\{u_t\}_{t \geq t_0}$ be a sequence of real numbers such that for all $t \geq t_0$

$$u_{t+1} \geq u_t (1 - 2\delta\gamma_{t+1}) + C\gamma_{t+1}^2 \tag{A.23}$$

with additionally

$$E\|\epsilon_{t_0}\|^2 \leq u_{t_0}. \tag{A.24}$$

Then for all $t > t_0$,

$$E\|\epsilon_t\|^2 \leq u_t. \tag{A.25}$$

PROOF: If (A.25) is true, then following (A.21) and (A.23),

$$E\|\epsilon_{t+1}\|^2 \leq (1 - 2\delta\gamma_{t+1})u_t + C\gamma_{t+1}^2 \leq u_{t+1},$$

which completes the proof of the lemma by induction. □

Proof of Theorem A.2 (continued) Take $t_* > t_0$, where t_0 is as defined in Lemma A.2, and choose λ such that

$$E\|\epsilon_{t_*}\|^2 \leq \lambda\gamma_{t_*}. \tag{A.26}$$

Following (A.25), for any sequence $\{u_t\}_{t\geq t_*}$ satisfying (A.23) and (A.24), we have

$$E\|\epsilon_t\|^2 \leq u_t, \tag{A.27}$$

We note that the sequence $u_t = \lambda\gamma_t$ with γ_t being specified in (A.11) satisfies the conditions (A.22) and (A.23) when t becomes large. This completes the proof of this theorem. □

A.2 Proof of Theorem 2.2.1, Theorem 2.2.2 and Theorem 2.2.3

Proof of Theorem 2.2.1. It follows from Theorem A.1, Theorem 2.2.1 can be proved by verifying that Pop-SAMC satisfies the conditions (A_1) to (A_3).

(A_1) It is obvious that this condition is satisfied by the sequence as specified in (2.4).

(A_2) Let $\boldsymbol{x}_{t+1} = (x_{t+1}^{(1)}, \ldots, x_{t+1}^{(\kappa)})$, which can be regarded as a sample produced by κ independent Markov chains on the product space $\mathbb{X} = \mathcal{X} \times \cdots \times \mathcal{X}$ with the transition kernel

$$\boldsymbol{P}_{\theta_t}(\boldsymbol{x}, \boldsymbol{y}) = P_{\theta_t}(x^{(1)}, y^{(1)}) P_{\theta_t}(x^{(2)}, y^{(2)}) \cdots P_{\theta_t}(x^{(\kappa)}, y^{(\kappa)}),$$

where $P_{\theta_t}(x, y)$ denotes a one-step MH kernel at a given value of θ_t. Under the assumptions that both Θ and $\mathcal{X}$ are compact and the proposal distribution is local positive, it has been shown in Liang et al. (2007a) that $P_\theta(x, y)$ satisfies the drift condition (A_2). In what follows, we will show that $\boldsymbol{P}_\theta(\boldsymbol{x}, \boldsymbol{y})$ also satisfies (A_2). To simplify notations, in what follows we will drop the subscript t, denoting x_t by x and $\theta_t = (\theta_{t1}, \ldots, \theta_{tm})$ by $\theta = (\theta_1, \ldots, \theta_m)$.

Roberts and Tweedie (1996) (Theorem 2.2) showed that if the target distribution is bounded away from 0 and ∞ on every compact set of its support $\mathcal{X}$, then the MH chain with a proposal distribution satisfying the local positive condition is irreducible and aperiodic, and every nonempty compact set is small. It follows from this result that $P_\theta(x, y)$ is irreducible and aperiodic, and thus $\boldsymbol{P}_\theta(x, y)$ is also irreducible and aperiodic.

Since $\mathcal{X}$ is compact, Roberts and Tweedie's result implies that $\mathcal{X}$ is a small set and the minorisation condition holds on $\mathcal{X}$ for the kernel $P_\theta(x, y)$; i.e., there exists an integer l, a constant δ, and a probability measure $\nu'(\cdot)$ such that

$$P_\theta^l(x, A) \geq \delta' \nu'(A), \quad \forall x \in \mathcal{X},\ \forall A \in \mathcal{B}_{\mathcal{X}}.$$

Therefore,

$$\boldsymbol{P}_\theta^l(\boldsymbol{x}, \boldsymbol{A}) \geq \delta \nu(\boldsymbol{A}), \quad \forall \boldsymbol{x} \in \mathbb{X},\ \forall \boldsymbol{A} \in \mathcal{B}_{\mathbb{X}},$$

where $\boldsymbol{A} = A_1 \times A_2 \times \ldots \times A_\kappa$, $\delta = (\delta')^\kappa$, and $\nu(\boldsymbol{A}) = \nu'(A_1) \times \nu'(A_2) \times \ldots \times \nu'(A_\kappa)$.

This verifies condition (A.4) by setting $\boldsymbol{C} = \mathbb{X}$. Thus, for any $\theta \in \Theta$ the following conditions hold

$$\begin{aligned} \boldsymbol{P}_\theta^l V^\alpha(\boldsymbol{x}) &\leq \lambda V^\alpha(\boldsymbol{x}) + bI(\boldsymbol{x} \in \boldsymbol{C}), \quad \forall \boldsymbol{x} \in \mathbb{X}, \\ \boldsymbol{P}_\theta V^\alpha(\boldsymbol{x}) &\leq \varsigma V^\alpha(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathbb{X}, \end{aligned} \tag{A.28}$$

by choosing $V(\boldsymbol{x}) = 1$, $0 < \lambda < 1$, $b = 1 - \lambda$, $\varsigma > 1$, and $\alpha \geq 2$. These conclude that (A_2-i) is satisfied.

For Pop-SAMC, we have $H(\theta, \boldsymbol{x}) = \sum_{i=1}^{\kappa} \boldsymbol{e}_{x^{(i)}}/\kappa - \boldsymbol{\pi}$, where $\boldsymbol{e}_{x^{(i)}}$ is an indicator vector of the subregion that $x^{(i)}$ belongs to. Since each component of $H(\theta, \boldsymbol{x})$ takes a value between 0 and 1, there exists a constant $c_1 = \sqrt{m}$ such that for any $\theta \in \Theta$ and all $x \in \mathcal{X}$,

$$\|H(\theta, \boldsymbol{x})\| \leq c_1. \tag{A.29}$$

Also, $H(\theta, x)$ does not depend on θ for a given sample x. Hence, $H(\theta, x) - H(\theta', x) = 0$ for all $(\theta, \theta') \in \Theta \times \Theta$, and the following condition holds,

$$\|H(\theta, \boldsymbol{x}) - H(\theta', \boldsymbol{x})\| \leq c_1 \|\theta - \theta'\|, \tag{A.30}$$

for all $(\theta, \theta') \in \Theta \times \Theta$. Equations (A.29) and (A.30) imply that (A_2-ii) is satisfied by choosing $\beta = 1$ and $V(x) = 1$.

In Liang et al. (2007a), it has shown for the single chain MH kernel that there exists a constant c_2 such that

$$|P_\theta(x, A) - P_{\theta'}(x, A)| \leq c_2 \|\theta - \theta'\|, \tag{A.31}$$

for any measurable set $A \subset \mathcal{X}$. Therefore, there exists a constant c_3 such that

$$
\begin{aligned}
&|\boldsymbol{P}_\theta(\boldsymbol{x}, \boldsymbol{A}) - \boldsymbol{P}_{\theta'}(\boldsymbol{x}, \boldsymbol{A})| = \Big| \int_{A_1} \cdots \int_{A_\kappa} \big[P_\theta(x^{(1)}, y^{(1)}) \cdots P_\theta(x^{(\kappa)}, y^{(\kappa)}) \\
&- P_{\theta'}(x^{(1)}, y^{(1)}) \cdots P_{\theta'}(x^{(\kappa)}, y^{(\kappa)})\big] dy^{(1)} \cdots dy^{(\kappa)} \Big| \\
&\le \int_{A_1} \int_{\mathcal{X}} \cdots \int_{\mathcal{X}} \left|P_\theta(x^{(1)}, y^{(1)}) - P_{\theta'}(x^{(1)}, y^{(1)})\right| P_\theta(x^{(2)}, y^{(2)}) \cdots P_\theta(x^{(\kappa)}, y^{(\kappa)}) \\
&dy^{(1)} \cdots dy^{(\kappa)} \\
&+ \int_{\mathcal{X}} \int_{A_2} \int_{\mathcal{X}} \cdots \int_{\mathcal{X}} P_{\theta'}(x^{(1)}, y^{(1)}) \left|P_\theta(x^{(2)}, y^{(2)}) - P_{\theta'}(x^{(2)}, y^{(2)})\right| P_\theta(x^{(3)}, y^{(3)}) \cdots \\
&P_\theta(x^{(\kappa)}, y^{(\kappa)}) \;\; dy^{(1)} \cdots dy^{(\kappa)} + \cdots \\
&+ \int_{\mathcal{X}} \cdots \int_{\mathcal{X}} \int_{A_\kappa} P_{\theta'}(x^{(1)}, y^{(1)}) \cdots P_{\theta'}(x^{(\kappa-1)}, y^{(\kappa-1)}) \left|P_\theta(x^{(\kappa)}, y^{(\kappa)}) - P_{\theta'}(x^{(\kappa)}, y^{(\kappa)})\right| \\
&dy^{(1)} \cdots dy^{(\kappa)} \le c_3 \|\theta - \theta'\|,
\end{aligned}
$$

which implies that (A.7) is satisfied. Condition (A_2-iii) is then satisfied by choosing $V(\boldsymbol{x}) = 1$ and $\beta = 1$.

(A_3) This condition can be verified as in Liang (2009). Since a part of the proof will be used in proving Theorem (2.2.2), we re-produce the proof below. Since the invariant distribution of the kernel $\boldsymbol{P}_\theta(\boldsymbol{x}, \boldsymbol{y})$ is $f_\theta(\boldsymbol{x})$, we have for any fixed θ,

$$
E(H^{(i)}(\theta, \boldsymbol{x})) = \frac{\int_{E_i} \psi(x) dx / e^{\theta_i}}{\sum_{k=1}^m [\int_{E_k} \psi(x) dx / e^{\theta_k}]} - \pi_i = \frac{S_i}{S} - \pi_i, \quad i = 1, \ldots, m, \quad \text{(A.32)}
$$

where $H^{(i)}(\theta, \boldsymbol{x})$ denotes the ith component of $H(\theta, \boldsymbol{x})$, $S_i = \int_{E_i} \psi(x) dx / e^{\theta_i}$ and $S = \sum_{k=1}^m S_k$. Thus, we have

$$
h(\theta) = \int_{\mathcal{X}} H(\theta, \boldsymbol{x}) f(d\boldsymbol{x}) = (\frac{S_1}{S} - \pi_1, \ldots, \frac{S_m}{S} - \pi_m)'.
$$

It follows from (A.32) that $h(\theta)$ is a continuous function of θ. Let $v(\theta) = \frac{1}{2} \sum_{k=1}^m (\frac{S_k}{S} - \pi_k)^2$. As shown below, $v(\theta)$ has continuous partial derivatives of the first order.

Solving the system of equations formed by (A.32), we have

$$\mathcal{L} = \{(\theta_1, \ldots, \theta_m) : \theta_i = \text{Const} + \log(\int_{E_i} \psi(x)dx) - \log(\pi_i), i = 1, \ldots, m; \theta \in \Theta\},$$

where Const $= \log(S)$ can be determined by imposing a constraint on S. For example, setting $S = 1$ leads to that $c = 0$. It is obvious that $\mathcal{L}$ is nonempty and $v(\theta) = 0$ for every $\theta \in \mathcal{L}$.

To verify the conditions related to $\dot{v}(\theta)$, we have the following calculations.

$$\begin{aligned} \frac{\partial S}{\partial \theta_i} = \frac{\partial S_i}{\partial \theta_i} = -S_i, \qquad & \frac{\partial S_i}{\partial \theta_j} = \frac{\partial S_j}{\partial \theta_i} = 0, \\ \frac{\partial(\frac{S_i}{S})}{\partial \theta_i} = -\frac{S_i}{S}(1 - \frac{S_i}{S}), \qquad & \frac{\partial(\frac{S_i}{S})}{\partial \theta_j} = \frac{\partial(\frac{S_j}{S})}{\partial \theta_i} = \frac{S_i S_j}{S^2}, \end{aligned} \tag{A.33}$$

for $i, j = 1, \ldots, m$ and $i \neq j$.

$$\begin{aligned} \frac{\partial v(\theta)}{\partial \theta_i} &= \frac{1}{2} \sum_{k=1}^{m} \frac{\partial(\frac{S_k}{S} - \pi_k)^2}{\partial \theta_i} \\ &= \sum_{j=1}^{m} (\frac{S_j}{S} - \pi_j) \frac{S_i S_j}{S^2} - (\frac{S_i}{S} - \pi_i) \frac{S_i}{S} \\ &= \mu_{\eta^*} \frac{S_i}{S} - (\frac{S_i}{S} - \pi_i) \frac{S_i}{S}, \end{aligned} \tag{A.34}$$

for $i = 1, \ldots, m$, where $\mu_{\eta^*} = \sum_{j=1}^{m} (\frac{S_j}{S} - \pi_j) \frac{S_j}{S}$. Thus, we have

$$\begin{aligned} \dot{v}(\theta) &= \mu_{\eta^*} \sum_{i=1}^{m} (\frac{S_i}{S} - \pi_i) \frac{S_i}{S} - \sum_{i=1}^{m} (\frac{S_i}{S} - \pi_i)^2 \frac{S_i}{S} \\ &= -\{ \sum_{i=1}^{m} (\frac{S_i}{S} - \pi_i)^2 \frac{S_i}{S} - \mu_{\eta^*}^2 \} \\ &= -\sigma_{\eta^*}^2 \leq 0, \end{aligned} \tag{A.35}$$

where $\sigma_{\eta^*}^2$ denotes the variance of the discrete distribution defined in the following table,

State (η^*)	$\frac{S_1}{S} - \pi_1$	$\cdots$	$\frac{S_m}{S} - \pi_m$
Prob.	$\frac{S_1}{S}$	$\cdots$	$\frac{S_m}{S}$

If $\theta \in \mathcal{L}$, $\dot{v}(\theta) = 0$; otherwise, $\dot{v}(\theta) < 0$. Therefore, $\sup_{\theta \in Q} \dot{v}(\theta) < 0$ for any compact set $Q \subset \mathcal{L}^c$. The proof is completed.

Proof of Theorem 2.2.2. It follows from Theorem A.2 and Theorem 2.2.1, this theorem can be proved by verifying the condition (A_4). To verify (A_4), we first show that $h(\theta)$ has bounded second derivatives. Continuing the calculation in (A.33), we have

$$\frac{\partial^2(\frac{S_i}{S})}{\partial(\theta^{(i)})^2} = \frac{S_i}{S}(1 - \frac{S_i}{S})(1 - \frac{2S_i}{S}), \quad \frac{\partial^2(\frac{S_i}{S})}{\partial\theta^{(j)}\partial\theta^{(i)}} = -\frac{S_i S_j}{S^2}(1 - \frac{2S_i}{S}), \tag{A.36}$$

where S and S_i are as defined in A.32. This implies that the second derivative of $h(\theta)$ is uniformly bounded by noting the inequality $0 < \frac{S_i}{S} < 1$.

Let $F = \partial h(\theta)/\partial\theta$. From (A.33) and (A.36), we have

$$F = \begin{pmatrix} -\frac{S_1}{S}(1 - \frac{S_1}{S}) & \frac{S_1 S_2}{S^2} & \cdots & \frac{S_1 S_m}{S^2} \\ \frac{S_2 S_1}{S^2} & -\frac{S_2}{S}(1 - \frac{S_2}{S}) & \cdots & \frac{S_2 S_m}{S^2} \\ \vdots & \ddots & \vdots & \vdots \\ \frac{S_m S_1}{S^2} & \cdots & \cdots & -\frac{S_m}{S}(1 - \frac{S_m}{S}) \end{pmatrix}.$$

Thus, for any nonzero vector $\boldsymbol{z} = (z_1, \ldots, z_m)^T$,

$$\boldsymbol{z}^T F \boldsymbol{z} = -\Big[\sum_{i=1}^{m} z_i^2 \frac{S_i}{S} - \left(\sum_{i=1}^{m-1} z_i \frac{S_i}{S} \right)^2 \Big] = -\mathrm{Var}(Z) < 0, \tag{A.37}$$

where $\mathrm{Var}(Z)$ denotes the variance of the discrete distribution defined by the following table:

State (Z)	z_1	$\cdots$	z_m
Prob.	$\frac{S_1}{S}$	$\cdots$	$\frac{S_m}{S}$

Thus, the matrix F is negative definite. Applying Taylor expansion to $h(\theta)$ at a point θ_*, we have

$$(\theta - \theta_*)^T h(\theta) \leq -\delta \|\theta - \theta_*\|^2,$$

for some value $\delta > 0$. Therefore, (A_4) is satisfied by SAMC.

Proof of Theorem 2.2.3. To prove Theorem 2.2.3, we first introduce the following two lemmas.

Lemma A.3 *Let $\boldsymbol{x} = (x_1, \ldots, x_\kappa)$ and $\boldsymbol{y} = (y_1, \ldots, y_\kappa)$. Let $P(x, y)$ denote a Markov transition kernel which admits $\pi(x)$ as its stationary distribution, and let $\boldsymbol{P}(\boldsymbol{x}, \boldsymbol{y}) = P(x_1, y_1) \times P(x_2, y_2) \times \ldots \times P(x_\kappa, y_\kappa)$ denote a product kernel. Let $\boldsymbol{g}(\boldsymbol{x}) = \sum_{i=1}^{\kappa} g(x_i)/\kappa$.*

(i) For any $k > 0$,

$$E\|\boldsymbol{P}^k \boldsymbol{g} - \pi(\boldsymbol{g})\| \leq E\|P^k g - \pi(g)\|,$$

where $\pi(g) = \int g(x)\pi(x)dx$.

(ii) $E\|\boldsymbol{g}(\boldsymbol{x})\|^2 \leq E\|g(x)\|^2$, where the inequality holds if and only if $g(x_i) = \boldsymbol{c}$ (a constant vector) for all $x \in \mathcal{X}$.

Proof:

$$\|\boldsymbol{P}^k \boldsymbol{g}(\boldsymbol{x}) - \pi(\boldsymbol{g}(\boldsymbol{x}))\| = \|\frac{1}{\kappa}\sum_{i=1}^{\kappa}(P^k g(x_i) - \pi(g(x_i)))\| \leq \frac{1}{\kappa}\sum_{i=1}^{k}\|P^k g(x_i) - \pi(g(x_i))\|.$$

Taking expectations on both sides of this inequality, we conclude the proof of part (i). Part (ii) follows directly from the inequality

$$\|\boldsymbol{g}(\boldsymbol{x})\|^2 \leq \frac{1}{\kappa}\sum_{i=1}^{\kappa}\|g_i(x)\|^2.$$

□

Lemma A.4 *Let the Markov transition kernels $P(x, y)$ and $\boldsymbol{P}(\boldsymbol{x}, \boldsymbol{y})$, and the functions $\boldsymbol{g}(\boldsymbol{x})$ and $g(x)$ be defined as in Lemma A.3. Let $\boldsymbol{u} = \sum_{n\geq 0}(\boldsymbol{P}^n\boldsymbol{g} - \pi(\boldsymbol{g}))$ be a solution of Poisson equation $\boldsymbol{u} - \boldsymbol{P}\boldsymbol{u} = \boldsymbol{g} - \pi(\boldsymbol{g})$, and let $u = \sum_{n\geq 0}(\tilde{P}^n g - \pi(g))$ be a solution of Poisson equation $u - Pu = g - \pi(g)$. Then*

(i) $E\|\boldsymbol{u}(\boldsymbol{x})\| \leq E\|u(x)\|$.

(ii) $E\|\boldsymbol{P}\boldsymbol{u}(\boldsymbol{x})\| \leq E\|Pu(x)\|$.

(iii) $E\|\boldsymbol{P}u(\boldsymbol{x}) - \boldsymbol{P}'u(\boldsymbol{x})\| \leq E\|Pu(x) - P'u(x)\|$, *where $\boldsymbol{P}'$ denotes a Markov transition product kernel different from $\boldsymbol{P}$, and $\boldsymbol{P}'(\boldsymbol{x}, \boldsymbol{y}) = P'(x_1, y_1) \times P'(x_2, y_2) \times \ldots \times P'(x_\kappa, y_\kappa)$.*

PROOF: By the Poisson equation, we have

$$\begin{aligned}\|\boldsymbol{u}(\boldsymbol{x})\| = \|\sum_{n\geq 0}\boldsymbol{P}^n\boldsymbol{g} - \pi(\boldsymbol{g})\| &= \|\frac{1}{\kappa}\sum_{i=1}^{\kappa}\big[\sum_{n\geq 0}\sum_{n\geq 0}P^n g(x_i) - \pi(g(x_i))\big]\| \\ &\leq \frac{1}{\kappa}\|\sum_{n\geq 0}\sum_{n\geq 0}P^n g(x_i) - \pi(g(x_i))\| \leq \frac{1}{\kappa}\|u(x_i)\|.\end{aligned}$$

Taking the expectation on both sides, we conclude the proof of part (i). The proofs for parts (ii) and (iii) are similar. □

PROOF: Let $u_t = \lambda\gamma_t$. Then, by (A.23), we have

$$\lambda \geq \frac{C\gamma_{t+1}^2}{\gamma_{t+1} - \gamma_t(1 - 2\delta\gamma_{t+1})}, \tag{A.38}$$

for all $t \geq t_*$, where t_* is given in (A.26). It is easy to show that for a given value of C,

$$\lambda = \frac{C}{2\delta - (1/\gamma_{t_*+1} - 1/\gamma_{t_*})}$$

will satisfy (A.38). Hence, a smaller value of C implies a smaller convergence bound of λ.

It follows from (A.12), (A.14), (A.16), (A.17) and (A.18), a lower bound of C for the Pop-SAMC algorithm can be given by

$$\begin{aligned} C_{\text{pop}} = \sup_{(\theta_{t-1},\theta_t)\in\Theta\times\Theta} \Big\{ & E(\|H(\theta_t, \boldsymbol{x}_{t+1})\|^2 + \frac{1}{\gamma_{t+1}}|\epsilon_t| E(\|\boldsymbol{P}_{\theta_t}\boldsymbol{u}(\theta_t, \boldsymbol{x}_t) - \boldsymbol{P}_{\theta_{t-1}}\boldsymbol{u}(\theta_{t-1}, \boldsymbol{x}_t)\|) \\ & + \frac{1}{\gamma_{t+1}}|(\epsilon_{t+1} - \epsilon_t)| E(\|\boldsymbol{P}_{\theta_t}\boldsymbol{u}(\theta_t, \boldsymbol{x}_{t+1})\|) + \frac{1}{\gamma_{t+1}}|\epsilon_t| E(\|\boldsymbol{P}_{\theta_{t-1}}\boldsymbol{u}(\theta_{t-1}, \boldsymbol{x}_t) - \boldsymbol{P}_{\theta_t}\boldsymbol{u}(\theta_t, \boldsymbol{x}_t)\|) \\ & + \frac{1}{\gamma_{t+1}}|\epsilon_t - \epsilon_{t+1}| \|\boldsymbol{P}_{\theta_t}\boldsymbol{u}(\theta_t, \boldsymbol{x}_t)\| \Big\}, \end{aligned}$$

where $\boldsymbol{P}$ denotes a product transition kernel, and $H(\theta, \boldsymbol{x}) = \sum_{i=1}^{\kappa}(\boldsymbol{e}_{x^{(i)}} - \boldsymbol{\pi})/\kappa$.

Similarly, for the single chain SAMC, a lower bound of C can be given by

$$\begin{aligned} C_{\text{sin}} = \sup_{(\theta_{t-1},\theta_t)\in\Theta\times\Theta} \Big\{ & E(\|H(\theta_t, x_{t+1})\|^2 + \frac{1}{\gamma_{t+1}}|\epsilon_t| E(\|P_{\theta_t}u(\theta_t, x_t) - P_{\theta_{t-1}}u(\theta_{t-1}, x_t)\|) \\ & + \frac{1}{\gamma_{t+1}}|(\epsilon_{t+1} - \epsilon_t)| E(\|P_{\theta_t}u(\theta_t, x_{t+1})\|) + \frac{1}{\gamma_{t+1}}|\epsilon_t| E(\|P_{\theta_{t-1}}u(\theta_{t-1}, x_t) - P_{\theta_t}u(\theta_t, x_t)\|) \\ & + \frac{1}{\gamma_{t+1}}|\epsilon_t - \epsilon_{t+1}| \|P_{\theta_t}u(\theta_t, x_t)\| \Big\}, \end{aligned}$$

where $H(\theta, x) = \boldsymbol{e}_{x^{(1)}} - \boldsymbol{\pi}$.

By Lemma A.3 and Lemma A.4, we have

$$C_{\text{pop}} < C_{\text{sin}},$$

where the straight inequality follows from the fact that $(\boldsymbol{e}_{x^{(i)}} - \boldsymbol{\pi})$ is not a constant vector. Hence, the Pop-SAMC algorithm has a smaller L^2 upper bound of λ than the single-chain SAMC algorithm. In other words, the Pop-SAMC algorithm can be more efficient than the single-chain SAMC algorithm. □

VITA

Mingqi Wu was born in Beijing, China. He received his B.S. and M.S. in materials science in June 2000 and June 2003 respectively, from Fudan University in Shanghai, China. He received his second M.S. in computational materials science in August 2006 from Rice University in Houston, Texas. He received his third M.S. and Ph.D. in statistics in May 2008 and December 2010 respectively, from Texas A&M University in College Station, Texas, under the direction of Dr. Faming Liang. His permanent address is: Department of Statistics, Texas A&M University, 3143 TAMU, College Station, TX, 77843-3143. Email: mqwu@stat.tamu.edu

The typist for this dissertation was Mingqi Wu.