Bayesian inference using advanced Monte Carlo methods in bioinformatics and cheminformatics

IKEBATA HISAKI

Doctor of Philosophy

Department of Statistical Science
School of Multidisciplinary Sciences
SOKENDAI (The Graduate University for Advanced Studies)

Bayesian inference using advanced Monte Carlo methods in bioinformatics and cheminformatics

Hisaki Ikebata

Department of Statistical Science

SOKENDAI

This dissertation is submitted for the degree of

Doctor of Philosophy

March 2017

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Hisaki Ikebata

March 2017

Acknowledgements

I want to express my sincere gratitude to my supervisor, Ryo Yoshida, for his useful advice which improved the quality of my research throughout my PhD course. Without his help, I would not have obtained the fundamental knowledge and skills required to tackle difficult research challenges. My second paper “Bayesian molecular design with a chemical language model" could not have been completed without the help of my co-authors, Ryo Maezono, Tetsu Isomura, and Kenta Hongo. In particular, Kenta Hongo instructed me in quantum chemistry simulation, and assisted me in using the computer cluster and supercomputer at JAIST. I feel gratitude to the members of the dissertation committee, Kenji Fukumizu, Yoshihiro Yamanishi, and Daichi Mochihashi, who gave me many valuable comments. I also want to say thanks for all of the assistance from the staffs at the ISM and SOKENDAI. The working environment of ISM is very supportive and I believe it accelerated the progress of my research. Finally, I thank my colleagues in SOKENDAI, my friends, and my family, who helped create a comfortable, and sometimes even an enjoyable, atmosphere during my studies. I must say their support was absolutely essential for me to be able to conduct my work throughout the duration of the PhD course. Thanks a lot!

Abstract

This thesis describes how to tackle problems in bioinformatics and cheminformatics using Bayesian methods. Bayesian methods can be used to solve inverse problems in various fields in academia and industry, but their application to real-world problems is usually complex. To capture the behavior of complex systems, the models of these systems also need to be complex and often contain various unknown and interacting parameters. For these reasons, obtaining solutions to these problems is difficult, and cannot be achieved only through a conjugate prior distribution or standard Monte Carlo methods. When dealing with a problem involving a high-dimensional parameter space, simple Monte Carlo methods such as rejection sampling or importance sampling are numerically intractable due to the curse of dimensionality. Although the Markov chain Monte Carlo (MCMC) method is often used as an alternative approach, it suffers from the local-trap problem. To deal with the local-trap problem, many existing methods use a tempering technique to lower the energy barrier between two different modes. In this thesis, a new MCMC method is developed, called the repulsive parallel MCMC (RPMCMC) method. It generates parallel Markov chains, and uses repulsive forces among the chains to explore the entire sampling space. A few methods which used RPMCMC were confirmed to work well for a synthetic multi-modal target distribution when compared to a simple Metropolis sampler.

One of the main contributions of this thesis is the introduction of two novel applications of the RPMCMC method in the context of Bayesian modeling. The first application we consider is in the field of bioinformatics, and is called the motif discovery problem. The aim of this problem is to find recurring patterns of conserved short strings that appear in a large fraction of nucleotide sequences. These patterns and locations can aid in the understanding

of important biological processes since the pattern preservation indicates the important biological processes occur there. Since recent experimental technologies called ChIP-seq can produce large numbers of fractions, many existing algorithms need to be reconstructed to deal with the increasing volume of data within an acceptable time. One major drawback of the existing methods, such as Gibbs sampling, arises from the highly multimodal posterior distribution since many and diverse motifs are present in a given sequence. Once the generated Markov chain is stuck in a locally high-probability region, it is difficult for an algorithm to escape from that region within a finite time. This problem has received little attention in previous studies. The aim of the RPMCMC approach is to achieve a high detection accuracy while keeping the computational efficiency at an acceptable level. The proposed method is designed to detect a greater diversity of motifs which existing methods are unable to discover. In experiments, compared to the original method using a standard Gibbs sampler, this all-at-once interacting parallel run can detect many more diverse motifs. Furthermore, this method was comprehensively tested on synthetic promoter sequences and real ChIP-seq datasets. In a synthetic promoter analysis, the RPMCMC algorithm found around 1.5 times as many embedded motifs as existing methods. For the ChIP-seq datasets, the RPMCMC algorithm obtained far more reliable cofactors than other recently published ChIP-tailored algorithms. Computational molecular design has great potential to save time and reduce costs in the discovery and development of functional molecules. Our second objective is to discover promising molecules that exhibit various kinds of desirable properties. Some previous studies tackled this issue with genetic algorithms (GAs) and molecular graph enumeration. The primary problem with these methods, the generation of unfavorable structures, was avoided by introducing many incomprehensive rules. An alternative approach, called the fragment assembly method, suffers from restricted design space and large computational loads. Our Bayesian molecular design begins by introducing a set of machine learning models that forwardly predict properties of a given molecule for multiple design objectives. These forward models are inverted to the backward model through Bayes' law, in combination with a prior distribution. This gives a posterior probability distribution conditioned on a desired property region. Exploring high-probability regions of the posterior distribution with

the sequential Monte Carlo (SMC) method, molecules that exhibit the desired properties are identified. The most notable feature of this workflow is it's novel backward prediction algorithm. In this study, a molecule is described by an ASCII string in the SMILES format. To reduce the occurrence of chemically unfavorable structures, a chemical language model is trained, which acquires commonly occurring patterns of chemical substructures by the natural language processing for the SMILES language of existing compounds. The trained model is used in the SMC algorithm to recursively refine SMILES strings of seed molecules such that the properties of the resulting molecules fall in the desired property region while eliminating the creation of unfavorable chemical structures. The effectiveness of this method was demonstrated with case studies in multi-objective molecular design aimed at investigating the physical properties (HOMO-LUMO gap and internal energy) and bio-activities of 10 target proteins.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Bayesian analysis

To model physically realistic and complex systems with statistical models, it is necessary to use methods which have a high expressive power. One of the most widely used models is the Gaussian mixture model. Estimation and inference of the Gaussian mixture model parameters can be achieved by the EM algorithm [104, 86]; however, typically this requires iteration and may not converge to the globally optimal parameter estimates, as there are often multiple peaks in the likelihood function. In addition, estimating the number of components in the model is also nontrivial [39].

Bayesian methods provide powerful tools to solve inverse problems in scientific and industrial fields [118, 83, 114, 10, 65, 97]. Consider a typical feature of inverse problems, the number of output dimensions is more than the number of input dimensions, meaning this problem is ill-posed; however, these problems can be solved by introducing prior information in the form of the prior distribution and incorporating this information into the statistical models. Bayesian models can be fitted even when the number of observations is small, and these models avoid the overfitting often encountered with frequentist approaches such as maximum likelihood estimation (MLE). When Bayesian methods first started to gain widespread attention, computational resources were scarce, so prior distributions had to be restricted to the conjugate prior form. The conjugate prior gives a posterior distribution

having the same form as the prior distribution, thus reducing the computation time required for model fitting and inference. This, however, narrows the range of applications for the Bayesian approach.

With advances in computer power, the above restrictions are being gradually relaxed, making it easier to use more flexible Bayesian techniques, including non-linear models such as Gaussian processes [34], deep hierarchical models with complex interactions [16], or structures such as trees [19, 4] and graphs [91, 36]. A representative method using a non-conjugate prior is Monte Carlo inference, which uses particles or a sequence of random variables to approximate the posterior distribution.

However, the problem is not so simple. Most parts of things that appear in our world show diversity. This diversity is necessary to provide robustness to the biological system, resulting in the furnishing of more opportunities to survive under critical environmental changes over a long period of time. Problems can arise when we try to analyze a system containing such diversity; that is to say, the posterior distribution consists of a mixture of many components, making the shape of the distribution non-convex with multiple peaks. When dealing with the inference problem in a high-dimensional parameter space, simple Monte Carlo methods such as rejection sampling or importance sampling are numerically infeasible due to the curse of dimensionality. Although the Markov chain Monte Carlo (MCMC) method is often used as an alternative, it suffers from the local-trap problem for target distributions with multiple peaks.

To deal with the local-trap problem, many existing methods use a tempering technique to lower the energy barrier between two different modes [35]. Here, we developed the new MCMC method, called the repulsive parallel MCMC (RPMCMC), using a novel approach. It generates parallel Markov chains, and uses repulsive forces among the chains in order to explore the entire sampling space. A few methods using RPMCMC were confirmed to work well for a synthetic multi-modal target distribution, when compared with a simple Metropolis sampler [79] in an experiment.

1.2 Applications

This thesis considers applications of Bayesian modeling for problems in biology and chemistry. Past research has shown for events in these fields that latent factors have an observable effect on final outcomes [26, 75, 99].

Two novel applications based on Bayesian techniques were introduced in this thesis[63, 62]. The first problem is considered important in bioinformatics, and is called the motif discovery problem. The goal of this problem is to find recurring patterns of conserved short strings that appear in a large fraction of nucleotide sequences. Identification of these patterns can lead to the discovery and understanding of important biological processes. Recently, the experimental ChIP-seq technologies have produced many more fractions than before, requiring existing algorithms to be reconstructed to handle large volumes of data within an acceptable time. Recent *de novo* motif discovery methods can be classified into either model-based optimization ([105]) or word-count approaches (DREME [115], Hegma [61]). Although they increase computational efficiency, they reduce the accuracy in motif detection since they use heuristics to speed up their computation. For the motif discovery problem, we obtained a superior result by using the RPMCMC algorithm.

The second problem focuses on the design of new molecules having desired properties. Computational molecular design has great potential to achieve enormous savings in time and cost during the discovery and development process of functional molecules, and it can be applied to a wide range of chemicals such as drugs, dyes, solvents, polymers, and catalysts. The objective of this problem is to computationally create novel molecules that have several desired properties. Some previous studies tackled this issue using genetic algorithms (GAs) [90] and molecular graph enumeration [129]. The main drawback of these methods, the generation of unfavorable structures during operation, needs to be avoided by introducing many incomprehensive rules. An alternative set of methods, called fragment assembly methods [122], suffer from a restricted design space and large computational loads. The distinguishing feature of our proposed algorithm is that a pattern of molecules expressed by ASCII strings called SMILES is learned using a method for natural language processing. The trained model is incorporated in the sequential Monte Carlo (SMC) algorithm to recursively

refine SMILES strings of seed molecules such that the properties of the resulting molecules fall in the desired property region while eliminating the creation of unfavorable chemical structures. The effectiveness of the method was demonstrated with case studies in multi-objective molecular design aimed at obtaining desired physical properties (HOMO-LUMO gap and internal energy) and bio-activities of 10 target proteins.

1.3 Thesis outline

This thesis is divided into three parts: the development of a Bayesian sampling method to overcome the local-trap problem encountered when inferring parameters of a posterior distribution, and its applications in two distinct fields.

Chapter 2 introduces the Bayesian techniques, including approaches with a non-conjugate prior, which are used in this thesis. These techniques are grouped into two types, deterministic or stochastic algorithms. Chapter 2 will provide a few examples of both types, with a special focus on Monte Carlo inference. Beginning from elementary methods such as importance sampling, we describe the basic idea of the Markov chain Monte Carlo (MCMC) method and its variants, and how to deal with problems when considering the posterior distribution having multiple isolated peaks. Before going into the specifics of algorithms, we show that a basic MCMC algorithm based on a random walk transition is ineffective. After that, we show several approaches, including the RPMCMC method, to overcome the problem arising with the standard MCMC through a simple comparison.

Chapters 3 and 4 describe applications of the above methods to problems in biology and chemistry. Chapter 3 shows that the RPMCMC algorithm avoids the local-trap problem arising when using the standard Gibbs sampler which is a widely used MCMC algorithm for motif discovery. The RPMCMC algorithm outperformed other existing methods for both a synthetic dataset and a real dataset. Furthermore, using the RPMCMC algorithm led to previously published discoveries although other existing methods missed to find them. Chapter 4 describes an attempt to construct the model for Bayesian sampling for one of the most important problems in cheminformatics, the inverse-QSPR problem. Although it is an

important problem, few researchers have tackled it, since it entails too large a chemical space to find an optimal solution. The proposed chemical structure representation based on the statistical language model is used for constructing an informative prior distribution, and this prior can be easily incorporated into the SMC sampler to achieve the generation of diverse chemical structures from the posterior distribution corresponding to the inverse-QSPR model.

Chapter 2

Bayesian analysis and Monte Carlo methods

In this chapter, we will describe the basic ideas of the Bayesian inference and some examples strongly relating to applications described in chapter 3 and chapter 4. Some of elementary or theoretical items are left in Appendices.

2.1 Posterior inference in Bayesian models

In a Bayesian perspective, a target quantity containing uncertainties wants to be inferred from observed data. Various types of Bayesian approaches have been reported [8, 109, 9]. The Bayesian model can be generalized as follows:

1. the sampling model that data $\mathbf{Y}$ is obtained from, with unknown parameter $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ for the conditional distribution, is given by

$$\mathbf{Y} \sim p(\mathbf{Y}|\theta), \tag{2.1}$$

2. a marginal distribution $p(\boldsymbol{\theta})$ for a quantity $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, which is called the prior distribution, is given by

$$\boldsymbol{\theta} \sim p(\boldsymbol{\theta}). \tag{2.2}$$

Bayes' theorem can convert prior knowledge into posterior knowledge by incorporating observations. When observing data $\mathbf{Y}$, Bayes' theorem gives the posterior distribution as

$$p(\boldsymbol{\theta}|\mathbf{Y}) = \frac{p(\mathbf{Y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{Y})}, \tag{2.3}$$

where $p(\mathbf{Y}|\boldsymbol{\theta})$ is the likelihood function which shows how likely it is that the observations $\mathbf{Y}$ occurred under parameter $\boldsymbol{\theta}$. The denominator of the right-hand side does not depend on $\boldsymbol{\theta}$.

2.1.1 Conjugate prior

Conjugate priors are useful for simplifying the computation of the posterior distribution. When a conjugate prior is chosen for a particular model (likelihood form), the corresponding posterior distribution belongs to the same family as the prior distribution.

Multinomial likelihood and Dirichlet prior

The example shown here is a model widely used for sequence analysis such as DNA and protein sequences in bioinformatics [73, 37]. The ZOOPs model used in Chapter 3 is a variant of this model which contains latent variables.

Here, it is assumed that a DNA string with N letters, $\mathbf{y} = \{y_1, \cdots, y_N\}$ is observed, where $y_i \in \Sigma = \{\mathrm{a}, \mathrm{t}, \mathrm{c}, \mathrm{g}\}$ for every i. Suppose that these letters are $i.i.d.$ samples from the multinomial distribution with unknown parameters $\boldsymbol{\theta}$; then the corresponding likelihood function is

$$p(\mathbf{y}|\boldsymbol{\theta}) = \prod_{k\in\Sigma} \theta_k^{\sum_{i=1}^{N} I(y_i=k)}, \tag{2.4}$$

where I is the indicator function, and the parameter vector $\boldsymbol{\theta}$ must satisfy $\sum_k \theta_k = 1$. In this case, the Dirichlet prior is conjugate. Using the Dirichlet distribution with hyperparameter $\boldsymbol{\alpha}$ for the prior of $\boldsymbol{\theta}$ to determine the strength of prior belief, given by

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \propto \prod_{k\in\Sigma} \theta_k^{\alpha_k - 1}, \tag{2.5}$$

the posterior distribution can be derived by multiplication of the likelihood function and the prior distribution given by

$$p(\boldsymbol{\theta}|\mathbf{y},\boldsymbol{\alpha}) \propto \prod_{k\in\Sigma}\theta_k^{\sum_{i=1}^N I(y_i=k)} \times \prod_{k\in\Sigma}\theta_k^{\alpha_k-1} \tag{2.6}$$

$$= \prod_{k\in\Sigma}\theta_k^{\sum_{i=1}^N I(y_i=k)+\alpha_k-1}. \tag{2.7}$$

This is the Dirichlet distribution.

Bayesian linear regression with known variance

Linear regression is widely used in many research fields, and the Bayesian version can deal with additional uncertainty. Here, a simple model which assumes that the variance of the observational noise is constant is shown. In subsection 4.2, a more sophisticated model with unknown noise variance is considered to model the Quantitative Structure-Property Relationship, which has long been used in cheminformatics for predicting target properties of new chemical structures. Linear regression assumes that the response variable $y \in \mathbb{R}$ can be modeled as a linear function of the input variables $\mathbf{x} \in \mathbb{R}^d$ by

$$y = \mathbf{w}^T \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix} + \varepsilon, \tag{2.8}$$

where $\mathbf{w} \in \mathbb{R}^{d+1}$ is a vector of weights and ε is a residual assumed to be normally distributed, $\mathrm{N}(0,\sigma^2)$ where σ is a positive constant. It is easy to extend to multivariate output $\mathbf{y} \in \mathbb{R}^L$, but we show a simpler case here.

If it is assumed that observed N data are *i.i.d.*, and are represented with the design matrix $\mathbf{X} = (\mathbf{x}_1,\ldots,\mathbf{x}_N)^T \in \mathbb{R}^{N\times d}$ and n response variables $\mathbf{y} \in \mathbb{R}^N$, then the likelihood function for the linear regression model is given by

$$p(\mathbf{y}|\mathbf{F},\mathbf{w},\sigma^2) = \mathrm{N}(\mathbf{y}|\mathbf{F}\mathbf{w},\sigma^2\mathbf{I}_N)$$

$$\propto \exp(-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{F}\mathbf{w})^T(\mathbf{y}-\mathbf{F}\mathbf{w})), \tag{2.9}$$

where $\mathbf{F} = (\mathbf{1}, \mathbf{X})$ with $\mathbf{1} = (1, \ldots, 1)^T$ and $\mathbf{I}_N$ is the $N \times N$ identity matrix. The prior distribution of the weight vector $\mathbf{w}$ is introduced using the Gaussian distribution as

$$\begin{aligned} p(\mathbf{w}) &= \mathrm{N}(\mathbf{w}|\mathbf{w}_0, \mathbf{V}_0) \\ &\propto \exp(-\frac{1}{2}(\mathbf{w}-\mathbf{w}_0)^T \mathbf{V}_0^{-1}(\mathbf{w}-\mathbf{w}_0)), \end{aligned} \tag{2.10}$$

where $\mathbf{V}_0$ is a positive definite matrix.

The corresponding posterior distribution is given by the product of the likelihood and the prior as

$$\begin{aligned} p(\mathbf{w}|\mathbf{F},\mathbf{y}) &\propto p(\mathbf{y}|\mathbf{F},\mathbf{w},\sigma^2)p(\mathbf{w}|\sigma^2) \\ &\propto \exp(-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{F}\mathbf{w})^T(\mathbf{y}-\mathbf{F}\mathbf{w})) \\ &\quad \times \exp(-\frac{1}{2}(\mathbf{w}-\mathbf{w}_0)^T\mathbf{V}_0^{-1}(\mathbf{w}-\mathbf{w}_0)) \\ &\propto \exp(-\frac{1}{2}(\mathbf{w}-\mathbf{w}_*)^T\mathbf{V}_*^{-1}(\mathbf{w}-\mathbf{w}_*)), \end{aligned} \tag{2.11}$$

$$\text{where } \mathbf{w}_* = \mathbf{V}_*\mathbf{V}_0^{-1}w_0 + \frac{1}{\sigma^2}\mathbf{V}_*\mathbf{F}^T\mathbf{y}, \tag{2.12}$$

$$\mathbf{V}_*^{-1} = \mathbf{V}_0^{-1} + \frac{1}{\sigma^2}\mathbf{F}^T\mathbf{F}, \tag{2.13}$$

$$\mathbf{V}_* = \sigma^2(\sigma^2\mathbf{V}_0^{-1} + \mathbf{F}^T\mathbf{F})^{-1}. \tag{2.14}$$

For predicting the output $\tilde{y}$ for a new coming data $\tilde{\mathbf{x}}$, the posterior predictive distribution, which is obtained by integrating over the parameter $\mathbf{w}$ of the posterior, is given by

$$\begin{aligned} p(\tilde{y}|\tilde{\mathbf{f}},\mathbf{F},\mathbf{y}) &= \int \mathrm{N}(\tilde{y}|\mathbf{w}^T\tilde{\mathbf{f}})\mathrm{N}(\mathbf{w}|\mathbf{w}_*,\mathbf{V}_*)d\mathbf{w} \\ &= \mathrm{N}(\tilde{y}|\mathbf{w}_*^T\tilde{\mathbf{f}}, {\sigma^2}_*(\tilde{\mathbf{f}})) \end{aligned} \tag{2.15}$$

$${\sigma^2}_*(\tilde{\mathbf{f}}) = \sigma^2 + \tilde{\mathbf{f}}^T\mathbf{V}_*\tilde{\mathbf{f}}, \tag{2.16}$$

where $\tilde{\mathbf{f}} = (1, \tilde{\mathbf{x}}^T)^T$. The variance of the posterior predictive distribution comes from two sources. One is the observation noise and the other is the uncertainty about how close the new input $\tilde{\mathbf{f}}$ is to the observation.

2.1.2 Non-conjugate prior

When a prior distribution is not a conjugate, integration of the corresponding posterior $\int p(\theta|\mathbf{Y})d\theta$ is analytically intractable, and thus the expectation of functions over the posterior distribution is also intractable. For this case, there are two common approaches for approximating the posterior. One is to approximately transform the likelihood function to make the prior conjugate, the other is to obtain the Monte Carlo approximation of the posterior. Although the Monte Carlo approach is one of the main concerns of this thesis, we first show some examples of deterministic methods for approximating the intractable posterior such as the Laplace approximation and variational inference. Later, we show several types of Monte Carlo methods.

Laplace approximation for the Bayesian logistic regression

The first example shown here often arises in Bayesian logistic regression. Here, it is supposed that the likelihood function is given by $p(\mathbf{Y}|\boldsymbol{\theta})$, and is not conjugate with the Gaussian prior $\mathrm{N}(\boldsymbol{\theta}|\boldsymbol{\theta}_0, \mathbf{V}_0^{-1})$. In this setting, the log-posterior distribution is given by

$$\log p(\boldsymbol{\theta}|\mathbf{Y}) = -\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^{\mathrm{T}}\mathbf{V}_0(\boldsymbol{\theta}-\boldsymbol{\theta}_0) + \log p(\mathbf{Y}|\boldsymbol{\theta}). \tag{2.17}$$

However, the integration of this is not analytically available since the product of two different forms does not follow a standard distribution. The Laplace approximation tries to approximate the posterior distribution $p(\boldsymbol{\theta}|\mathbf{Y})$ with the Gaussian distribution as

$$q(\boldsymbol{\theta}) = \mathrm{N}(\boldsymbol{\theta}|\boldsymbol{\theta}_*, \mathbf{V}), \tag{2.18}$$

where $\boldsymbol{\theta}_*$ satisfies $\nabla_{\boldsymbol{\theta}_*}\log p(\boldsymbol{\theta}_*|\mathbf{Y}) = 0$ obtained by an optimization algorithm such as iterative reweighted least squares (IRLS) [24], and $\mathbf{V} = -\nabla\nabla\log p(\boldsymbol{\theta}|\mathbf{Y})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_*}$ is the Hessian of the log-posterior distribution evaluated at $\boldsymbol{\theta}_*$. This approximation, however, is not accurate when the likelihood function deviates from the Gaussian distribution.

Variational inference

Suppose that what has to be done is to infer the posterior distribution $p^*(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathscr{D})$ for observed data $\mathscr{D}$, however this quantity is hard to be evaluated since it can decompose as

$$p(\boldsymbol{\theta}|\mathscr{D}) = \frac{p(\boldsymbol{\theta},\mathscr{D})}{\int_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} p(\boldsymbol{\theta},\mathscr{D})d\boldsymbol{\theta}} \tag{2.19}$$

and the denominator $p(\mathscr{D}) = \int_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} p(\boldsymbol{\theta},\mathscr{D})d\boldsymbol{\theta}$ in the right hand of this equation is often intractable. In other words, $p(\boldsymbol{\theta},\mathscr{D})$ can be computed pointwise since it is just the product of the likelihood and the prior. Let that product $p(\boldsymbol{\theta}|\mathscr{D})p(\mathscr{D})$ be denoted as $\tilde{p}(\boldsymbol{\theta})$.

Variational inference uses a tractable distribution $q(\boldsymbol{\theta})$ with some additional free parameters to approximate the intractable distribution $p(\boldsymbol{\theta}|\mathscr{D})$. The widely used objective is the KL divergence between $p(\boldsymbol{\theta}|\mathscr{D})$ and $q(\boldsymbol{\theta})$, which is given by

$$KL(p||q) = \int_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} p(\boldsymbol{\theta}|\mathscr{D}) \log \frac{p(\boldsymbol{\theta}|\mathscr{D})}{q(\boldsymbol{\theta})}. \tag{2.20}$$

This objective is obviously intractable since $p(\boldsymbol{\theta}|\mathscr{D})$ is intractable in this setting. As an alternative, the reverse KL divergence is often used,

$$\begin{aligned}
KL(q||p) &= \int_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathscr{D})} && (2.21)\\
&= \int_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta},\mathscr{D})/p(\mathscr{D})} && (2.22)\\
&= \int_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{\tilde{p}(\boldsymbol{\theta})} + p(\mathscr{D}) && (2.23)\\
&= KL(q||\tilde{p}) + p(\mathscr{D}). && (2.24)
\end{aligned}$$

Since $p(\mathscr{D})$ does not depend on $\boldsymbol{\theta}$, minimizing this objective makes q close to the tractable target $\tilde{p}$. The most widely used form of the variational inference is the mean field approximation [95]. Assuming that the parameter of the posterior is multivariate $\boldsymbol{\theta} \in \mathbb{R}^M$, the mean field approximation uses a fully factorized form,

$$q(\boldsymbol{\theta}) = \prod_i^M q_i(\theta_i). \tag{2.25}$$

The objective is to find $q_i, \cdots, q_M$ to minimize the following expression iteratively:

$$\min_{q_1,\cdots,q_M} KL(q||\tilde{p}), \tag{2.26}$$

where for each q_i the optimization is conducted over the parameters of the marginal distribution. This approximation deviates from the true distribution if the parameters are correlated.

Monte Carlo approximation

Monte Carlo methods provide an effective alternative for approximating posterior distributions. Suppose that random samples $Z_1, \cdots, Z_N$ can be obtained from the posterior $p(z|\mathbf{Y})$ with non-conjugate prior. In the Monte Carlo approximation, the posterior distribution is represented pointwise as

$$\hat{p}(z) = \frac{1}{N}\sum_{i=1}^{N} \delta_{Z_i}(z), \tag{2.27}$$

where $\delta_Z(z)$ is the delta function which returns 1 when $z = Z$ and 0 otherwise. Using this form of the distribution, the expectation of a bounded function f with respect to the posterior distribution can be given by

$$\mathbb{E}_{p(z|\mathbf{Y})}[f(Z)] = \frac{1}{N}\sum_{i=1}^{N} f(z)\delta_{Z_i}(z) \tag{2.28}$$

$$= \frac{1}{N}\sum_{i=1}^{N} f(Z_i). \tag{2.29}$$

As simple examples, the posterior mean and variance are given by $\mu^* = \frac{1}{N}\sum_{i=1}^{N} Z_i$ and $\sigma^* = \frac{1}{N}\sum_{i=1}^{N}(Z_i - \mu^*)^2$, respectively. The details of Monte Carlo inference methods will be discussed in the next subsection.

2.2 Monte Carlo inference

To achieve efficient posterior inference, a variety of Monte Carlo sampling methods will be discussed here. In Monte Carlo methods, it is necessary to consider how to generate random variables from standard distributions such as the Gaussian, gamma, and Dirichlet

distributions, as illustrated in the examples of Chapters 3 and 4. Methods for generating various types of random variables are explained in Appendix A.

2.2.1 Importance sampling

Since, for non-conjugate distributions, analytic expressions for the corresponding posterior distributions are usually not available, the standard random variable generators shown in Appendix A cannot be used for the inference. Importance sampling is one of the simplest methods for Monte Carlo inference. Here, the goal is to determine the expectation of some bounded function f over the intractable non-standard density $p(x)$ which is called the target distribution. The expectation is given by

$$\mathbb{E}_{p(x)}[f(x)] = \int f(x)p(x)dx. \tag{2.30}$$

It is assumed that this expectation cannot be computed and nor can samples be obtained from $p(x)$ directly, but random variables can be obtained from the standard distribution $q(x)$ that should be close to $p(x)$. In importance sampling, samples from $q(x)$ are generated then used to estimate the expectation with respect to the target distribution $p(x)$ as

$$\begin{aligned}
\mathbb{E}_{p(x)}[f(X)] &= \int f(x)p(x)dx \\
&= \int f(x)\frac{p(x)}{q(x)}q(x)dx \\
&\simeq \frac{1}{N}\sum_{i=1}^{N} f(x)\frac{p(x)}{q(x)}\delta_{X_i}(x) \\
&= \frac{1}{N}\sum_{i=1}^{N} f(X_i)\frac{p(X_i)}{q(X_i)} \\
&= \frac{1}{N}\sum_{i=1}^{N} f(X_i)w_i,
\end{aligned} \tag{2.31}$$

where $X_i, \cdots, X_N$ are obtained from the standard distribution $q(x)$, and $w_i = \frac{p(X_i)}{q(X_i)} (i = 1, \ldots, N)$ are called the importance weights.

2.2.2 Sampling importance resampling

The sampling importance resampling generates samples from the unnormalized target probability density function (pdf) $p(x)$ by using the following weighted particle distribution obtained in the importance sampling introduced before, which is given as

$$\hat{p}(x) \approx \sum_i W_i \delta_{X_i}(x), \tag{2.32}$$

where W_i is the normalized importance weight for the ith particle as $\sum_{i=1}^{N} W_i = 1$. It is straightforward to show that when these particles are replaced with the probability W_i $(i = 1,\ldots,N)$ allowing duplication, the updated particles also approximately follow the target distribution,

$$\begin{aligned}
\hat{P}(x \in A) &= \sum_{i=1}^{N} I(X_i \in A) W_i \\
&= \frac{\sum_{i=1}^{N} I(X_i \in A) \frac{\tilde{p}(X_i)}{q(X_i)}}{\sum_{j=1}^{N} \frac{\tilde{p}(X_j)}{q(X_j)}} \\
&\xrightarrow[N\to\infty]{} \frac{\int I(x_i \in A) \frac{\tilde{p}(x)}{q(x)} q(x)}{\int \frac{\tilde{p}(x)}{q(x)} q(x)} \\
&= \frac{\int I(x_i \in A) \tilde{p}(x)}{\int \tilde{p}(x)} \\
&= \int I(x_i \in A) p(x) = P(x \in A),
\end{aligned} \tag{2.33}$$

where $\tilde{p}$ is the unnormalized density of distribution P.

This procedure is called sampling importance resampling [108], which is necessary for sequential approaches such as the particle filter [31, 5] and the sequential Monte Carlo (SMC) sampler [28].

2.2.3 Markov chain Monte Carlo

When generating $i.i.d.$ random samples from the target distribution is difficult due to high dimensionality or other technical issues, dependent samples under some generation rules

can be exploited to approximate the target distribution. One of the most useful concepts for generating dependent samples is the Markov chain. A Markov chain is a sequence of random variables $X_1, X_2, \cdots$ with the Markov property stipulating that the current state only depends on a finite number of past states. This relation is given by

$$P(X^{(t+1)} \in A | X^{(0)} = x^{(0)}, \cdots, X^{(t)} = x^{(t)}) = P(X^{(t+1)} \in A | X^{(t)} = x^{(t)}), \tag{2.34}$$

for all measurable sets $A \in \chi$ for time $t = 0, 1, 2, \cdots$. Here, the marginal distribution of $X^{(t)}$ over states χ at time t is written as $P_t(dx)$. From the initial distribution $P_0(dx)$, the marginal distribution of the Markov chain $\{X^{(t)}\}$ evolves from time t to time $t+1$ as

$$P_{t+1}(dx) = \int_{\chi} P_t(dz) P_t(z, dx), \tag{2.35}$$

where $P_t(z, dx)$ is called the transition kernel at time t which is the probability measure for $X^{(t+1)}$ given $X^{(t)} = z$. In particular, the Markov chain Monte Carlo (MCMC) approach uses time-homogeneous Markov chains by setting $P_t(z, dx) = P(z, dx)$ for all t. Under this setting, Eq. 2.35 becomes

$$P_{t+1}(dx) = \int_{\chi} P_t(dz) P(z, dx). \tag{2.36}$$

It is noted that when using the time-homogeneous transition kernel, $P_t(dx)$ can be uniquely determined from the initial distribution $P_0(dx)$ and the transition kernel $P(z, dx)$. From this fact, one can write the conditional distribution of $X^{(t)}$ given $X^{(0)} = x$ using $P_t(x, \cdot)$.

The focus of this thesis is on Bayesian inference using Monte Carlo sampling. In order to approximate $\mathbb{E}[f(X)]$ with respect to the target distribution $\pi(x)$, a transition kernel $P(z, dx)$ is required which is invariant for the target distribution $\pi(dx)$, that is, the following balance condition

$$\pi(dx) = \int_{\chi} \pi(dz) P(z, dx) \tag{2.37}$$

should be satisfied. This means that if X_t is obtained from $\pi(x)$, then $X^{(t+1)}$ is also obtained from $\pi(x)$, but dependent on $X^{(t)}$. When the following convergence $\lim_{t\to\infty} P_t(X^{(t)} \in A|X^{(0)} = x) = \pi(x)$ for π-almost x and all measurable sets $A \in \chi$ is achieved, this $\pi(x)$ is called the equilibrium distribution of the Markov chain. The convergence properties of the MCMC approach are shown in Appendix B. In later subsections, we will introduce several methods based on the MCMC approach that are useful for solving the problems in Chapters 3 and 4.

2.2.4 Gibbs sampling

The random variable generation methods introduced in the previous subsections, such as importance sampling and rejection sampling, become infeasible when dealing with a high-dimensional space. A typical obstacle in rejection sampling is that the acceptance probability tends to zero as the number of dimensions increases because of the curse of dimensionality. Similarly, the weight distribution of the importance sampling degenerates to one particle when the number of dimensions becomes sufficiently high. The Gibbs sampling technique has been widely used for high-dimensional problems in Bayesian analysis [43, 40]. This technique is based on iterative sampling procedures from conditional distributions of the target. Let the target distribution be $f(\mathbf{x}), \mathbf{x} \in \chi$. The first step is to make a partition of $\mathbf{x}$, which has K blocks, to satisfy $dim(\mathbf{x}_1) + \cdots + dim(\mathbf{x}_K) = d$ where $\mathbf{x} = (\mathbf{x}_1, \cdots, \mathbf{x}_K)$. The second step is to obtain the corresponding conditional distributions, for example, a blocked variable $\mathbf{x}_k$ in the kth block can be expressed as

$$f(\mathbf{x}_k|\mathbf{x}_1, \cdots, \mathbf{x}_{k-1}, \mathbf{x}_{k+1}, \cdots, \mathbf{x}_K) = \frac{f(\mathbf{x})}{f(\mathbf{x}_1, \cdots, \mathbf{x}_{k-1}, \mathbf{x}_{k+1}, \cdots, \mathbf{x}_K))}, \tag{2.38}$$

for $k = 1, \cdots, K$. Under this setting, the procedure for Gibbs sampling is iterative from the K conditional distributions. We show a simple case when $K = 3$ in the following.

1. Initialize the variable $\mathbf{x}^{(0)} = (\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \mathbf{x}_3^{(0)})$
2. at time t, $\mathbf{x}_1^{(t+1)} \sim f(\mathbf{x}|\mathbf{x}_2^{(t)}, \mathbf{x}_3^{(t)})$

3. $\mathbf{x}_2^{(t+1)} \sim f(\mathbf{x}|\mathbf{x}_1^{(t)}, \mathbf{x}_3^{(t+1)})$
4. $\mathbf{x}_3^{(t+1)} \sim f(\mathbf{x}|\mathbf{x}_1^{(t+1)}, \mathbf{x}_2^{(t+1)})$
5. set $t = t+1$, then go back to step 2.

When this Markov chain satisfies the regularity conditions such as irreducibility and aperiodicity described in Appendix B, the distribution of $\mathbf{x}^{(t)}$ is considered to have converged to the target distribution $f(\mathbf{x})$.

Data Augmentation

Even if some of the data are missing, the Gibbs sampler can be used to complement them [117]. It is equivalent to the stochastic version of the missing data analysis in the EM algorithm [29]. In this thesis, this method is used to analyze motif models including unobserved motif positions in the Motif discovery problem (Chapter 3). Let $X_{obs} \in \chi_{obs}$ and $X_{mis} \in \chi_{mis}$ be the observed data and the missing data, respectively. $X = (X_{obs}, X_{mis})$ is called the complete data, assumed to come from some distribution $p(X_{obs}, X_{mis}|\boldsymbol{\theta})$ where $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ is the parameter of interest. Since X_{mis} is not observed, the goal of the Bayesian inference is to obtain the marginal posterior distribution $p(\boldsymbol{\theta}|X_{obs})$ with prior distribution $p(\boldsymbol{\theta})$.

$$p(X_{obs}|\boldsymbol{\theta}) = \int_{X_{mis}} p(X_{obs}, X_{mis}|\boldsymbol{\theta}) dX_{mis}. \tag{2.39}$$

The procedure for the data augmentation is as follows.

1. Initialize $\boldsymbol{\theta}^{(0)}$
2. At time t, obtain $X_{mis}^{(t+1)} \sim p(X_{min}|\boldsymbol{\theta}^{(t)}, X_{obs})$
3. obtain $\boldsymbol{\theta}^{(t+1)} \sim p(\boldsymbol{\theta}|X_{obs}, X_{min}^{(t+1)})$
4. $t := t+1$, then go back to 2.

This is a simple form of the Gibbs samplers with only two conditional distributions that have to be considered.

2.2.5 Metropolis-Hastings method

Gibbs sampling is effective for a variety of problems, but it can be used only when the posterior distribution has a conditional distribution that is a standard distribution such as Gaussian, Dirichlet, or a discrete distribution, whereas the Metropolis-Hastings (MH) algorithm [79, 54] can be applied to a wider variety of distributions. Suppose that the target distribution is $\pi(dx)$ with pdf $f(x)$ on sample space χ and σ-field $\mathfrak{B}_\chi$. As shown in the previous subsection, the Markov chain uses a transition kernel $P(x,dy)$ with invariant distribution $\pi(x)$,

$$\pi(dx) = \int_\chi \pi(dz)P(z,dx). \tag{2.40}$$

In the MH algorithm, the transition kernel is constructed to satisfy the reversibility condition. More precisely, a Markov chain with transition kernel $P(z,dx)$ and invariant distribution $\pi(dx)$ is reversible if it satisfies the detailed balance condition

$$\int_B \int_A \pi(dz)P(z,dx) = \int_A \int_B \pi(dx)P(x,dz), \tag{2.41}$$

for $\forall A,B \in \mathfrak{B}_\chi$. Using a form of pdf, the detailed balance condition can be

$$f(x)p(z|x) = f(z)p(x|z), \tag{2.42}$$

where $p(z|x)$ is the pdf of the transition kernel $P(x,dz)$ given a fixed x. To maintain this condition, the MH algorithm adopts the acceptance-rejection rules

1. At time t, z is generated from the proposal distribution $q(z|x_t)$
2. $x_{t+1} = z$ with acceptance probability $\alpha = \min\{1, \frac{f(z)q(z|x_t)}{f(x_t)q(x_t|z)}\}$, and $x_{t+1} = x_t$ with probability $1-\alpha$.

This acceptance probability is chosen to satisfy the following reversible condition:

$$f(x)q(z|x)\alpha(x,z) = f(z)q(x|z)\alpha(z,x). \tag{2.43}$$

Using this condition, the condition in Eq. 2.42 can be verified in the general case. Here, if it is assumed that $\alpha(z,x)$ is measurable with respect to $Q(dz|x) = q(z|x)\nu(dz)$ for some probability measure ν, then the transition kernel in the MH sampler can be shown as

$$\begin{aligned} P(x,A) &= \int_A Q(dz|x)\alpha(x,z) + I_{x\in A}\left(\int_\chi Q(dz|x)\left(1-\alpha(x,z)\right)\right) \\ &= \int_A Q(dz|x)\alpha(x,z) + I_{x\in A}\left(1-\int_\chi Q(dz|x)\alpha(x,z)\right), \end{aligned} \tag{2.44}$$

for $A \in \mathfrak{B}_\chi$, that is,

$$\begin{aligned} P(x,dz) &= Q(dz|x)\alpha(x,z) + \delta_x(dz)r(x) \\ &= q(dz|x)\alpha(x,z)\nu(dz) + \delta_x(dz)r(x), \end{aligned} \tag{2.45}$$

where $r(x)$ is $\int_\chi Q(dz|x)\alpha(x,z)$, which represents the average rejection probability for state x. Therefore, the reversibility condition of the Markov chain when using the MH kernel can be proven through the following. For $\forall A, B \in \mathfrak{B}_\chi$,

$$\begin{aligned} & \int_B\int_A \pi(dx)P(x,dz) \\ = & \int_B\int_A f(x)q(z|x)\alpha(x,z)\nu(dx)\nu(dz) + \int_B\int_A \delta_x(dz)f(x)r(x)\nu(dx)\nu(dz) \\ = & \int_B\int_A f(x)q(z|x)\alpha(x,z)\nu(dx)\nu(dz) + \int_{A\cap B} f(x)r(x)\nu(dx) \\ = & \int_B\int_A f(z)q(x|z)\alpha(z,x)\nu(dx)\nu(dz) + \int_{A\cap B} f(x)r(x)\nu(dx) \ (\text{u}\textit{sing conditon Eq.}2.43) \\ = & \int_B\int_A f(z)q(x|z)\alpha(z,x)\nu(dz)\nu(dx) + \int_{B\cap A} f(z)r(z)\nu(dz) \\ = & \int_A\int_B \pi(dz)P(z,dx) \end{aligned} \tag{2.46}$$

2.2.6 Slice sampler

The slice sampler is an alternative method for obtaining samples from a non-standard conditional distribution, when Gibbs sampling is not appropriate [55]. In the applications considered in this thesis, the slice sampler is used to sample from the full conditional

distribution in the repulsive parallel MCMC sampler for the motif discovery problem (details will be given in Section 3.2.3).

Assume that the density function of the target distribution is $f(x)$ $(x \in \chi)$. As considered in the rejection sampling (shown in Appendix A), obtaining a sample from $f(x)$ is equivalent to sampling uniformly from the following area

$$A = \{(x,u) : 0 \leq u \leq f(x)\}. \tag{2.47}$$

The solution is obtained by augmenting a random variable U from the conditional distribution $\text{Unif}(0, f(x))$ given x. Then, the joint density of (X,U) is

$$f(x,u) = f(x)f(u|x) \propto 1_{(x,u)\in A}. \tag{2.48}$$

Therefore, with the Gibbs sampler, samples from the joint distribution are obtained with the following algorithm.

1. At time t, $u_t \sim \text{Unif}(0, f(x_t))$
2. $x_t \sim \text{Unif}\{x : f(x) \geq u_{t+1}\}$
3. set $t = t+1$, then go back to 1

For a multivariate distribution, the same number of augmented variables are prepared and the above steps are repeated for each variable in turn. In practice, it is difficult to determine the region A, and thus the *stepping out* method is used to identify the interval $x_{\min} \leq x \leq x_{\max}$ [88]. Starting from x^*, the stepping out method moves the current point in both the positive and negative directions as $x^* + \Delta$, $x^* - \Delta$ for a positive constant Δ, repeatedly until those points are near the border of the region A. The slice sampler has been proven to converge. Roberts and Rosenthal showed that the slice sampler is geometrically ergodic under particular conditions [107], and Miya ad Tierney showed that the slice sampler is uniformly ergodic under slightly stronger conditions [80]. The property of ergodicity is described in Appendix B.

2.2.7 Reversible jump MCMC for Bayesian model selection

There is an issue with how many parameters are required to define a suitable model for the observed data. With too many parameters there can be issues with overfitting, while with too few parameters the model does not have enough power to fully reflect the true distribution underlying the data, which causes bias. A simple example for the linear regression problem is shown in many textbooks such as [15, 85]. In this thesis, the reversible jump MCMC (RJMCMC) method is used for the motif discovery problem (Chapter 3) to specify motif lengths using Bayesian modeling.

In the RJMCMC, let $\{M_k : k \in K\}$ be a countable set of models to fit the observation X. Each model has its own parameters $\boldsymbol{\theta}_k \in \boldsymbol{\Theta}_k$. Without loss of generality, it is supposed that each model has different parameter dimensions. The prior distribution of the number of model parameters and conditional prior of the parameters given model k are $p(k)$ and $p(\boldsymbol{\theta}_k|k)$, respectively.

Under this assumption, the target distribution with the RJMCMC method is given by

$$\pi(k, \boldsymbol{\theta}_k|Y) \propto p(Y|k, \boldsymbol{\theta}_k)p(\boldsymbol{\theta}_k|k)p(k). \tag{2.49}$$

The RJMCMC method tries to construct a homogeneous Markov chain whose invariant distribution is the target. Consider the proposal distribution $q(k^*|k)$ to generate the dimension k^* from k; if the generated dimension k^* is different from the current k, it is necessary to make those two dimensions equal using augment vectors u^* generated from a distribution $\psi_{k^t \to k^*}(\mathbf{u})$, and introduce the bijection T as

$$(\boldsymbol{\theta}_{k^*}, \mathbf{u}^*) = T(\boldsymbol{\theta}_k^{(t)}, \mathbf{u}), \tag{2.50}$$

where the concatenated vectors on both sides have the same dimensions. From the above setting, the RJMCMC algorithm can be iterated as follows.

1. At time t, obtain model $M_{k^{(t+1)}}$ from the proposal distribution $q(k^{(t+1)}|k^{(t)})$
2. generate $\mathbf{u}$ from $\psi_{k^{(t)} \to k^*}(\mathbf{u})$

3. generate $(\boldsymbol{\theta}_{k^*}, \mathbf{u}^*)$ from $T(\boldsymbol{\theta}_k^{(t)}, \mathbf{u})$
4. compute $r = \dfrac{\pi(k^*, \boldsymbol{\theta}_{k^*}^* | Y) q(k^{(t+1)} | k^{(t)}) \psi_{k^{(t)} \to k^*}(\mathbf{u}^*)}{\pi(k^{(t)}, \boldsymbol{\theta}_k^{(t)} | Y) q(k^{(t)} | k^{(t+1)}) \psi_{k^* \to k^{(t)}}(\mathbf{u})} \left| \dfrac{\partial(\boldsymbol{\theta}_{k^*}^*, \mathbf{u}^*)}{\partial(\boldsymbol{\theta}_k^{(t)}, \mathbf{u})} \right|$,
   where $\left| \dfrac{\partial(\boldsymbol{\theta}_{k^*}^*, \mathbf{u}^*)}{\partial(\boldsymbol{\theta}_k^{(t)}, \mathbf{u})} \right|$ is the determinant of the Jacobian of the transformation in Eq. 2.50.
5. $X^{(t+1)} = (\boldsymbol{\theta}_{k^*}, \mathbf{u}^*)$ with probability $\min\{1, r\}$, or $X^{(t+1)} = X^{(t)}$ with probability $1 - \min\{1, r\}$.

The RJMCMC can be said to be a MH-like procedure with an initial distribution including auxiliary variables. In many actual applications, T is an identity transformation and ψ is an independent sampler; thus, the Jacobian can be reduced to 1.

2.3 Multi-modality

When the target distribution $\pi(x)$ is multi-modal, the regular MCMC methods described above suffer from the local-trap problem. In MCMC simulations for a complex distribution having a lot of lags between multiple modes, the trajectory of the Markov chain becomes trapped in a region around a single mode, and never transits into regions around other modes. As a result, the Markov chain cannot converge to the target distribution in finite time. We show a simple example using the Metropolis algorithm for a mixture of Gaussian distributions as shown in [74].

2.3.1 Simple Metropolis-Hastings case

Here, the target pdf is a mixture of two Gaussian distributions, given by

$$\pi(\mathbf{x}) = 0.4\mathrm{N}(\boldsymbol{\mu}_1, 1.5\mathbf{I}) + 0.6\mathrm{N}(\boldsymbol{\mu}_2, 1.5\mathbf{I}), \tag{2.51}$$

where $\boldsymbol{\mu}_1 = (0,0)^T$, $\boldsymbol{\mu}_2 = (10,10)^T$. In this example, the pdf of the initial distribution from the previous state $\mathbf{x}'$ is $q(\mathbf{x}|\mathbf{x}') = \mathrm{N}(\mathbf{x}', 3^2\mathbf{I})$, and is chosen so as to achieve an acceptance ratio of around 30% as recommended in [41]. Starting from a point $\mathbf{x}_0 = (5,5)^T$, the MH

sampler is iterated 5000 times. The trajectories of four independent samplers are given in Fig. 2.1, and show that the local trap problem occurred after the burn-in period in two out of four trials. The resulting kernel density estimates (rightmost ones in Fig. 2.1) of samples generated for each Markov chain thus surely deviate from the true distribution.

Many researchers have tried to develop sampling techniques for multi-modal distributions. Below, methods based on the MCMC method or the SMC sampler targeting a sequence of annealed target distributions are outlined.

2.3.2 Simulated tempering

Suppose that the target distribution can be represented as $\pi(x) \propto \exp(-H(x))$ for $x \in \chi$, where H is an energy function. The temperature behaves as in simulated annealing [60, 70], which is a popular global optimization method. The simulated tempering uses an augmented variable T taking a finite number of discrete values $T_1 > T_2, \cdots, T_m = 1$. The final mth temperature is 1, which is treated as the target temperature; thus, the augmented target distribution is given by $\pi(x,T) \propto \exp(\frac{-H(x)}{T})$. x and T are updated iteratively in the same manner as in Gibbs sampling. The trial distribution is defined as $\pi(x,T_i) = \exp(\frac{-H(x)}{T_i})/Z_i$, where Z_i is the normalizing constant for $i = 1, \cdots, m$. In addition to the procedure in the MH algorithm using the transition kernel K_i at temperature T_i, the simulated tempering requires the proposed probability of transition between different temperatures. Let $q_{i,j}$ be the proposed probability from T_i to T_j. With no prior information, this probability is set to be uniform and the transition is restricted only to the current or neighboring states to avoid a high rejection probability, that is, $q_{i,i-1} = q_{i,i} = q_{i,i+1} = 1/3$ for $1 < i < m-1$, $q_{1,1} = q_{1,2} = q_{m,m-1} = q_{m,m} = 1/2$. Under this setting, the whole procedure for the simulated tempering can be described as follows.

1. Initialize T_{i_0} and x_0
2. At time t, transition to temperature T_j with probability $q_{i_t,j}$.
3. $x_{t+1} \sim K_j(x_t,)$

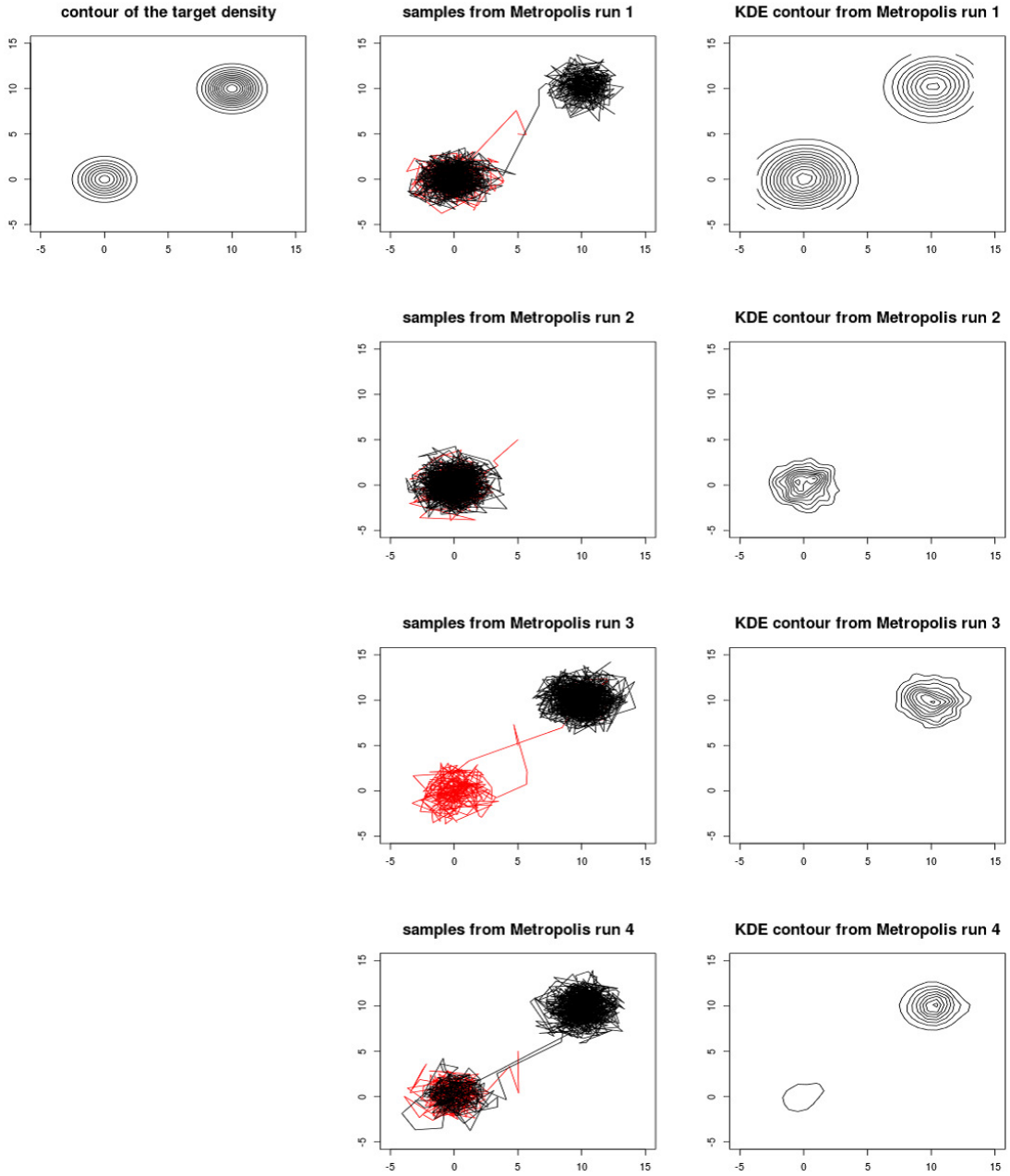


Fig. 2.1 Metropolis sampler for a bimodal distribution

4. If $j = i_t$, i_{t+1} is set to j. If not, $i_{t+1} = j$ with acceptance probability

$$\min\{1, \frac{\hat{Z}_j}{\hat{Z}_i} \exp(-H(x_{t+1}))(\frac{1}{T_j} - \frac{1}{T_i})\frac{q_{j,i_t}}{q_{i_t,j}}\}, \tag{2.52}$$

where $\hat{Z}_i$ is an estimate of Z_j using the reverse logistic regression method proposed in [44]. Otherwise, $i_{t+1} = i_t$.

5. $t := t + 1$, then go back to step 2.

This procedure is iterated until convergence. The multi-layered structure of the algorithm allows the full exploration of the target distribution.

2.3.3 Parallel tempering

Suppose that the pdf of the target distribution is $\pi(\mathbf{x})$; then a general form of the augmented distribution used in the population based MCMC is given by

$$\pi(\mathbf{x}) = \pi(x_1, \cdots, x_m) = \prod_{i=1}^{m} \pi_i(x_i), \tag{2.53}$$

where $\mathbf{x} \in \chi^m$ is the augmented random variable, and m is the population size.

In parallel tempering (PT), the target distribution is defined similarly as in the simulated tempering with a temperature ladder $T_1 > T_2 > \ldots > T_m = 1$ such that $\pi_m(x) = \pi(x)$. The parallel sequence of annealed distributions is shown below.

$$\pi_i(x) \propto \exp\left(\frac{-H(x)}{T_i}\right), \text{for } i = 1, \cdots, m. \tag{2.54}$$

where T_i is the temperature for annealing for each component. Each temperature is determined as in the simulated tempering. By increasing the temperature, each distribution is flattened, making the MH transition easier to traverse over the entire space. In addition, an exchange operation effectively transmits samples into different temperature levels.

Suppose that $\mathbf{x}^{(t)} = \left(x_1^{(t)}, \ldots, x_m^{(t)}\right)$ is the population of samples at time t. The PT iterates the following two steps.

1. Update each $x_i^{(t)}$ with the MH kernel K_i as done in the simulated tempering.

2. Replace the states of the chain i and j with the probability $q_{i,j}$. Accept this exchange with probability $\alpha = \min\{1, \frac{q_{j,i}}{q_{i,j}} \exp\left(\left[H(x_i^{t+1}) - H(x_j^{(t+1)})\right]\left[\frac{1}{T_i} - \frac{1}{T_j}\right]\right)\}$

The sequence of temperatures $T_i, \ldots, T_m$ needs to be controlled in practice. The PT has been proven effective for simulations of complicated system such as polymer dynamics [89, 133], and the spin-glass model [59, 25].

2.3.4 Sequential Monte Carlo sampler with annealing

The SMC sampler is used to obtain samples sequentially from a sequence of probability distributions [27, 31, 77]. In this thesis, this method works well for avoiding the local-trap problem, and is applied to the inverse-QSPR problem in Chapter 4, which is expected to have many isolated peaks in the target distribution (corresponding to a wide variety of chemical structures satisfying the target properties). When it is difficult to obtain samples from an intricate distribution $\pi(x)$, one can prepare a sequence of annealed target distributions as

$$\pi_t(x) \propto \pi(x)^{\beta_t}, \text{for } t = 1, \cdots, T, \tag{2.55}$$

where β_t is a non-decreasing sequence of inverse temperatures $0 \leq \beta_1 \leq \beta_2 \leq \ldots \leq \beta_{s-1} \leq \beta_s = \ldots = \beta_T = 1$. This is equivalent to the sampling version of the simulated annealing.

In the SMC sampler, the unnormalized artificial joint distribution $\tilde{\gamma}_{1:t}(x_{1:t})$ is introduced as

$$\tilde{\pi}_{1:t}(x_{1:t}) = \tilde{\gamma}_{1:t}(x_{1:t})/Z_t, \tag{2.56}$$

$$\tilde{\gamma}_{1:t}(x_{1:t}) = \gamma_t(x_t) \prod_{l=1}^{t-1} L_l(x_{l+1}, x_l), \tag{2.57}$$

where $L_{t-1} : \chi \times \chi \to [0,1]$ is the backward Markov kernel with pdf $L_{t-1}(x_t, x_{t-1})$ for $x_t, x_{t-1} \in \chi$, Z_t is a normalization constant, and $\gamma_t(x_t)$ is the true unnormalized marginal target density at time t. This artificial joint distribution admits that the marginalization of this joint distribution over the history until at time $t-1$ is equal to $\gamma_t(x_t)$.

Assume that the representation of the marginal proposal distribution at time t can be obtained pointwise as

$$\eta_t^N(x_t) = \int \eta_{t-1}^N(z) K_t(z, x_t) dz = \sum_{i=1}^{N} K_t(X_{t-1}^{(i)}, x_t), \tag{2.58}$$

where $K_t(z,x) : \chi \times \chi \to [0,1]$ is the Markov kernel which is the probability density around z. At time $t-1$, it is also assumed that the particle distribution $\{W_{t-1}^{(i)}, X_{1:t-1}^{(i)}\}(i = 1, \cdots, N, W_t^{(i)} > 0, \sum_{i=1}^{N} W_{t-1}^{(i)} = 1)$ represents the target distribution $\tilde{\pi}_{1:t-1}$, that is to say

$$\tilde{\pi}_{1:t}^N(dx_{1:t}) = \sum_{i=1}^{N} W_{t-i}^{(i)} \delta_{X_{1:t-1}^{(i)}}(dx_{1:t}). \tag{2.59}$$

This propagated density at each time can be incorporated into the importance sampling framework. The importance weights at time t are given by

$$\begin{aligned} w_t(x_{1:t}) &= \frac{\tilde{\gamma}_t(x_{1:t})}{\eta_t(x_{1:t})} \\ &= w_{t-1}(x_{1:t-1}) \tilde{w}_t(x_{t-1}, x_t), \end{aligned} \tag{2.60}$$

where

$$\tilde{w}_t(x_{t-1}, x_t) = \frac{\gamma_t(x_t) L_{t-1}(x_t, x_{t-1})}{\gamma_{t-1}(x_{t-1}) K_t(x_{t-1}, x_t)}, \tag{2.61}$$

Since the discrepancy between η_t and $\tilde{\pi}_t$ tends to increase with increasing t, the variance of the weight distribution also tends to increase. This often causes a degeneracy of the particle approximation. This degeneracy can be evaluated with the effective sample size (ESS) defined as $\{\sum_{i=1}^{N} (W_t^{(i)})^2\}^{-1}$ [76]. To prevent degeneracy, a resampling is done through the sampling importance resampling when the ESS exceeds a predefined threshold, and the weights are reset. The whole procedure for the SMC sampler is as follows.

step 1 :Initialization

- set $t = 1$.
- draw $x_1^{(i)} \sim \eta_1$ for $i = 1, \ldots, N$.

- evaluate weights $w_1 = \frac{\gamma_1(x)}{\eta_1(x)}$, and normalize weights as $W_1^{(i)} = w_1^{(i)} / \sum_j^N w_1^{(j)}$ for $i = 1 \ldots, N$.

step 2 :resampling

if ESS $< T$ (for some threshold T), resample the particles and set $W_t^{(i)} = 1/N$.

step 3 :sampling

- draw $x_t^{(i)} \sim K(X_t^{(i)}, \cdot)$ for $i = 1 \ldots, N$.
- update weights $\{\tilde{w}_t^{(i)}\}_{i=1,\ldots,N}$ by using Eq. 2.61 and normalize them as in step 1.
- set $t := t + 1$.
- go back to step 2.

2.3.5 Repulsive parallel Markov chain Monte Carlo

The RPMCMC is our proposed method, and is shown to capture the latent motif structures better than other existing methods in Chapter 3. The RPMCMC algorithm is derived by creating an augmented system $\pi_A(x_1, \ldots, x_M|\beta)$ which consists of M exact copies of the target distribution $\pi_i(x) = \pi(x)$ $(i = 1, \ldots, M)$ and the repulsive force function $\psi(x_1, \ldots, x_M)$:

$$\pi_A(x_1, \cdots, x_M|\beta) \propto \prod_{i=1}^{M} \pi(x_i) \psi(x_1, \ldots, x_M)^{\beta}, \quad \beta \geq 0. \tag{2.62}$$

Each x_i is called a *replica*. The repulsive force function ψ imposes a stronger penalty on closer replicas. The parameter β controls force intensity; that is, a greater β produces a stronger repulsion and vice versa. Drawing samples of $x_1, \ldots, x_M$ simultaneously from Eq. 2.62, the M sample paths tend to move toward different regions. Furthermore, a replica trapped in a locally high-probability state can be pushed to other regions by the repulsive force generated when approaching other replicas (Fig. 2.2). To obtain a sample from each conditional distribution, the Metropolis sampler or the slice sampler can be used. It is important to see that the use of a non-zero force severity biases the samples from π_A with

respect to the posterior distribution. When $\beta = 0$, which removes the repulsion from π_A, an unbiased sample set can be obtained.

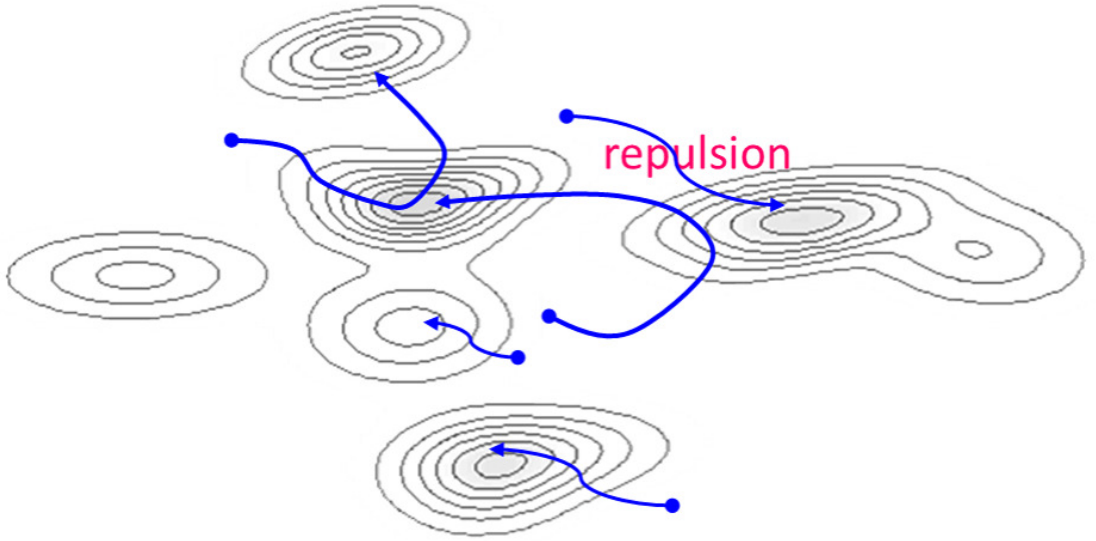


Fig. 2.2 Schematic view of the repulsive parallel MCMC algorithm

The repulsive function should be chosen such that it is bounded in the sample space. For example, for a distribution with bounded support such as the Dirichlet distribution, the following repulsive function can be used.

$$\psi(\mathbf{x}_1, \ldots, \mathbf{x}_M) = \prod_{i=1}^{M-1} \min_{j<i} \exp(||\mathbf{x}_i - \mathbf{x}_j||^2). \tag{2.63}$$

This repulsive function is effective in the motif discovery problem shown in Chapter 3. However, it cannot be applied to distributions with unbounded support such as the Gaussian distribution since the value of the repulsive function goes to infinity as an argument goes to infinity. In this case, the following repulsive function is more effective.

$$\psi(\mathbf{x}_1, \ldots, \mathbf{x}_M) = \prod_{i=1}^{M-1} \left(1 - \frac{2}{1 + \min_{j<i} \exp ||\mathbf{x}_i - \mathbf{x}_j||^2}\right). \tag{2.64}$$

This function takes a value between 0 and 1, so it is bounded even when a target distribution is supported in unbounded space. Fig. 2.3 shows the conditional distributions of the augmented Gaussian distributions ($M = 2$) given a fixed replica using the two types of repulsive functions introduced above. The upper row in Fig. 2.3 shows that repulsion from Eq. 2.63 works well since the probability of region around other replicas decreases, but when looking at a distant region from these two modes, the value of this function grows exponentially. Once a sampler tends towards these regions, it never returns to the region around the modes of the target distribution, whereas the repulsion from Eq. 2.64 remains constant when replicas are located far from each other.

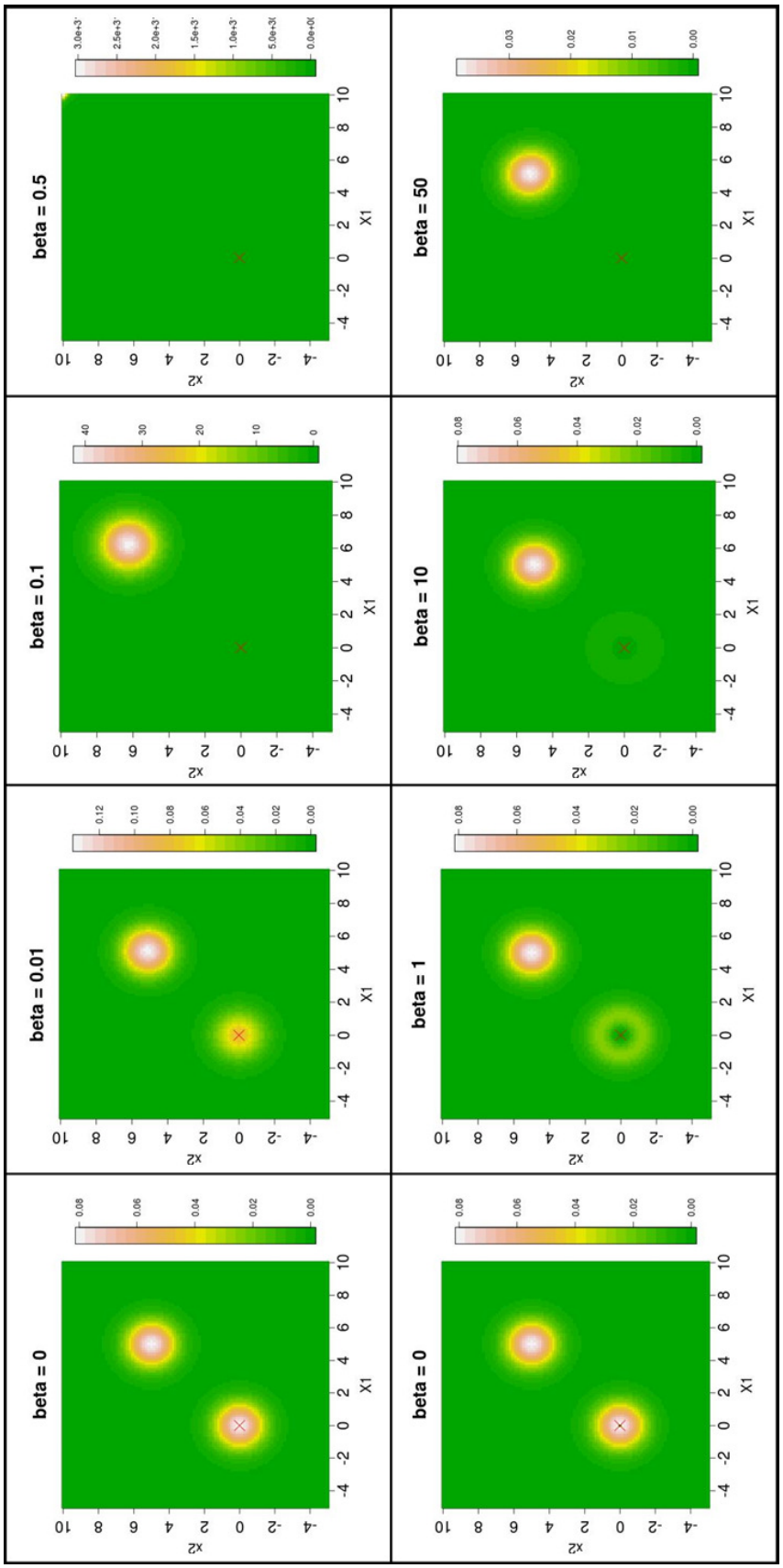


Fig. 2.3 Upper and lower rows represent the conditional distributions of the augmented Gaussian distributions given a fixed replica (red cross) using repulsive functions in Eq. 2.63 and Eq. 2.64 with different repulsive force β, ($M = 2$).

Table 2.1 Mean vectors of the components of the mixture Gaussian distribution

k	$\boldsymbol{\mu}_1$	$\boldsymbol{\mu}_2$	k	$\boldsymbol{\mu}_1$	$\boldsymbol{\mu}_2$	k	$\boldsymbol{\mu}_1$	$\boldsymbol{\mu}_2$	k	$\boldsymbol{\mu}_1$	$\boldsymbol{\mu}_2$
1	2.18	5.76	6	3.25	3.47	11	5.41	2.65	16	4.93	1.50
2	8.67	9.59	7	1.70	0.50	12	2.70	7.88	17	1.83	0.09
3	4.24	8.48	8	4.59	5.60	13	4.98	3.70	18	2.26	0.31
4	8.41	1.68	9	6.91	5.81	14	1.14	2.39	19	5.54	6.86
5	3.93	8.82	10	6.87	5.40	15	8.33	9.50	20	1.69	8.11

Suppose that we are given a sample set of size $N \times M$ from Eq. 2.62 with nonzero β, denoted by $\{x_i^{(j)} | i = 1, \ldots, M, j = 1, \ldots, N\}$ where each $x_i^{(j)}$ denotes the jth sample of the ith replica. Obviously, the repulsive force leads to biased samples with respect to the target $\pi_A(x_1, \cdots, x_M|0)$ in Eq. 2.62 when the repulsive force is zero. To correct this bias, importance sampling is used, which assigns weights to each sample by

$$w_i^{(j)} = \frac{\pi_A(x_1^{(j)}, \ldots, x_M^{(j)}|0)}{\pi_A(x_1^{(j)}, \ldots, x_M^{(j)}|\beta)} \propto \frac{1}{\psi(x_1^{(j)}, \ldots, x_M^{(j)})}. \tag{2.65}$$

The ratio between the target (zero force) and the biased distribution ($\beta > 0$) becomes the inverse of the repulsive force function. Note that the M replicas $x_i^{(j)}$ ($i = 1, \ldots, M$) in the jth ensemble share the same weight.

2.3.6 Experiment

In the following experiment, the objective is not to compare the algorithms, but to confirm that the RPMCMC and the SMC can sample from the multi-modal distributions as effectively as the PT does. Consider a synthetic 2D mixture of Gaussian distributions, as

$$\pi(\mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma} \sum_{k=1}^{20} w_k \exp\{-\frac{1}{2\sigma^2}(\mathbf{x} - \boldsymbol{\mu}_k)^T(\mathbf{x} - \boldsymbol{\mu}_k)\}, \tag{2.66}$$

where $\sigma = 0.1, w_i = 0.05$ for all i, ([74]). The mean values $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{20}$ are shown in Table 2.1. This distribution is shown in Fig. 2.4.

In this experiment, with initial values generated from the uniform distribution $\mathrm{Unif}(0,10)^2$, each sampler is run for the target distribution $\pi(\mathbf{x})$. The setting used for each method is shown in the following list.

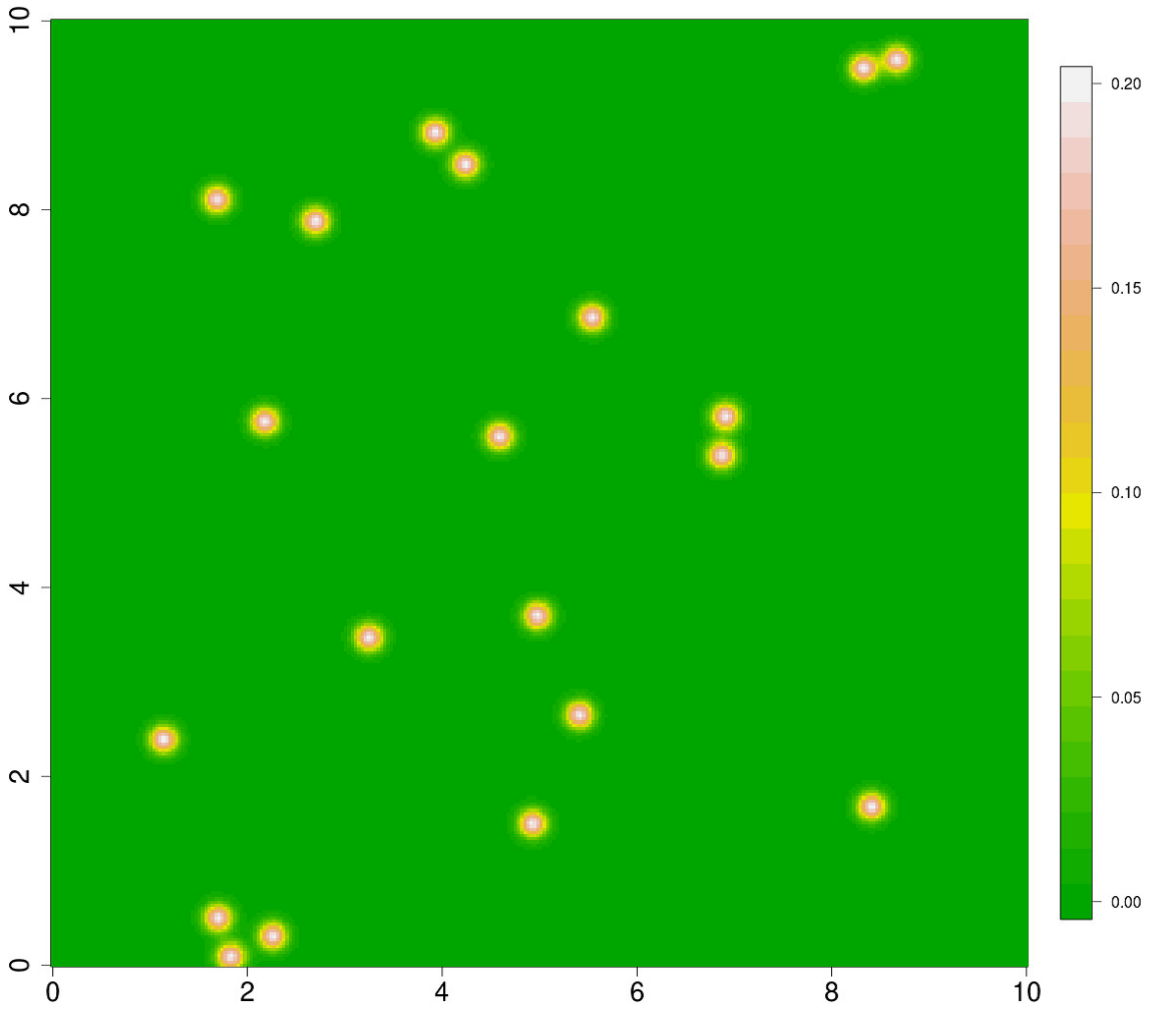


Fig. 2.4 Mixture of 20 Gaussian distributions used in the experiment.

1. RPMCMC with 40 replicas with repulsive function $\psi(\mathbf{x},\mathbf{y}) = 1 - \frac{2}{1+\exp(||\mathbf{x}-\mathbf{y}||)}$ where $||\cdot||$ is a Euclidean norm, generates 3.0×10^4 samples (5.0×10^3 samples were discarded in the burn-in period).

Table 2.2 Comparison three Monte Carlo methods for multi-modal distribution with the standard Metropolis sampler

		RPMCMC		SMC		PT		Metropolis	
parameters	true values	est.	sd	est.	sd	est.	sd	est.	sd
μ_1	4.48	4.54	0.083	4.53	0.294	4.69	0.406	4.62	2.09
μ_2	4.91	4.90	0.118	4.98	0.445	4.72	0.441	3.90	1.76
Σ_{11}	5.61	5.55	0.111	5.45	0.71	5.79	0.753	2.22	1.23
Σ_{22}	9.77	9.84	0.234	9.49	0.67	8.55	0.742	5.41	3.14
Σ_{12}	2.63	2.59	0.148	2.00	1.03	-0.13	1.192	0.55	1.40

2. PT with 20 parallel chains as in [74] generated 1.8×10^5 samples (3.0×10^4 samples were discarded as in the burn-in period) taking almost the same computation time as the RPMCMC did. The temperatures used in the PT were equally spaced between 1 and 5. The proposal distribution is $\mathrm{N}(\mathbf{x}^{'}|\mathbf{x}, 0.25^2 T\mathbf{I})$, where T is the temperature.

3. The SMC sampler iterates 200 steps with 2500 particles and $\kappa(t) = 50^{0.97^t}$. The transition of particles is followed by the MCMC kernel [28] with the proposal distribution $\mathrm{N}(\mathbf{x}^{'}|\mathbf{x}, 0.25^2 T\mathbf{I})$ where T is the temperature.

4. The Metropolis sampler generated 3.6×10^6 samples (6.0×10^5 samples were discarded as in the burn-in period) taking almost the same computation time as the RPMCMC did. The proposal distribution is the Gaussian distribution $\mathrm{N}(\mathbf{x}^{'}|\mathbf{x}, 0.25^2\mathbf{I})$.

Table 2.2 shows the estimates for the global mean, variance, and covariance of the target distribution $\pi(\mathbf{x})$. Here, est. denotes the mean of the 10 independent trials and sd denotes their standard deviations. The estimates by the RPMCMC, SMC, and PT methods have much smaller standard deviations than the regular Metropolis sampler. The Metropolis sampler gives different results for the estimation each time due to the local-trap problem, and only generated samples around one mode in three of 10 trials. For the RPMCMC, SMC, and PT methods, the local-trap problem was not observed in these 10 trails, and thus these methods can be said to avoid it.

Chapter 3

Motif discovery problem

3.1 Introduction

The motif discovery problem is a research field that is focus on by many researcher for a long time but is recently receiving renewed attention since recent experimental technologies, such as ChIP-seq, posed new challenges. A genome-wide ChIP study produces thousands or more DNA fragments consisting of several hundred base pairs, which cover the binding sites of a transcription factor (TF). This mechanism is very important for biological systems from the fact that once the TF binds on the target region called the transcription factor binding sites (TFBSs), the corresponding gene is activated, leading to the next biological process for surviving (Fig. 3.1). The problem can be identified as finding recurring patterns of conserved short strings that appear in a large fraction of nucleotide sequences (Fig.3.2) since important regions tend to be preserved in the process of evolution. Motifs are visualized by using SeqLogos of the position weight matrix (PWM) as shown in Fig.3.3. By discovering motifs in the given sequences, which are associated with known TF-binding motifs in a database, e.g. JASPAR [110], TRANSFAC [127], we can predict not only the regions bound by the primary TF but also the cofactors that modulate the TF activity. Finding cofactor can also be important issue to elucidate complicated biological processes [115, 45, 112].

Early methods of *de novo* motif discovery can be classified into either a model-based (MEME [116], AlignACE [58], ANN-Spec [130]) or a word-count approach (Weeder [96]).

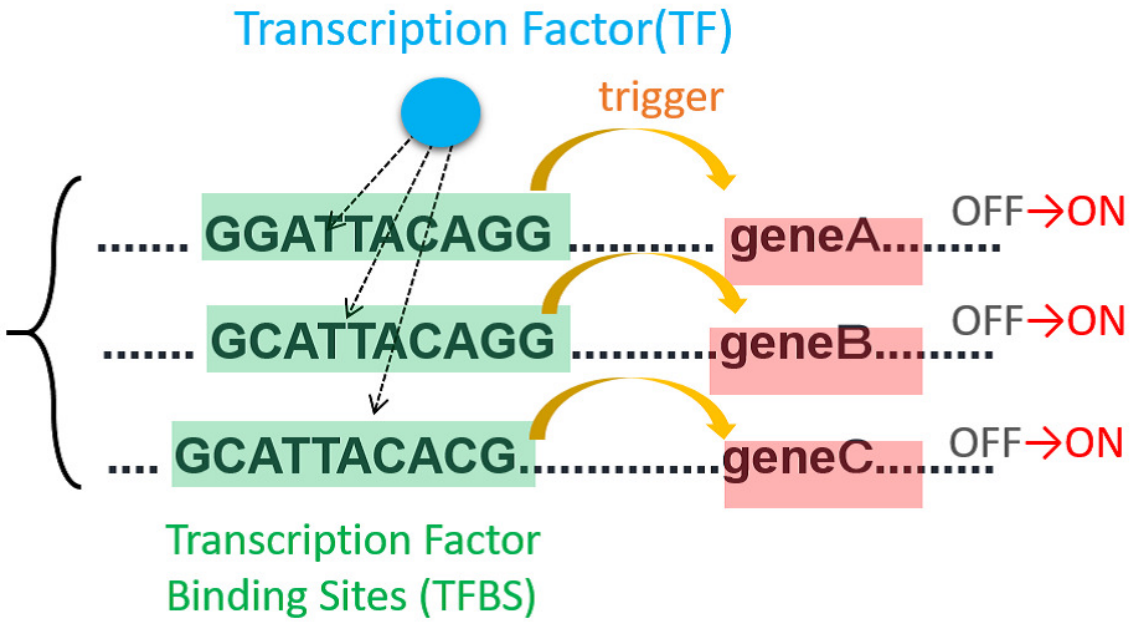


Fig. 3.1 A schematic view of the gene activation by binding a transcription factor to the transcription factor binding site. Once the transcription factor binds to the transcription factor binding sites, it triggers to activate the corresponding genes.

Seq_1 : GGGGCGCGATTCCAGGGGGCGCGGGAGGGG
Seq_2 : CCGGATGGCACCCCCGGCCGGTGTGCCCGGC
Seq_3 : GGCCGGTGTGGGGGCGCGGGAATGCCACCCG
Seq_4 : GGGGGCGCCGGATGCTACCGGCCGGTGCCGG

Fig. 3.2 Under an assumption that important structures should be preserved, the objective here is to find similar subsequences called motifs (colored in blue) on the upper region of genes in which transcription factors usually bind.

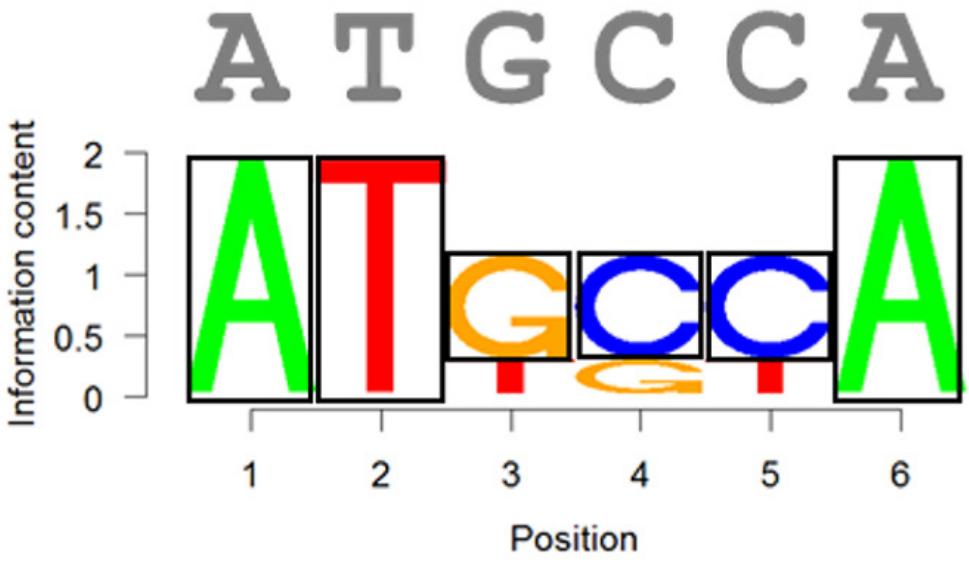


Fig. 3.3 The representation of the position weight matrix. The vertical axis represents the information content for each letter (shown with the size of each letter). The horizontal axis indicates the position in a motif. This example shows that this motif tends to have A, T and A at the first, third and sixth position, respectively.

These methods were designed on the assumption that the input sequences of $\sim 10^3$ base pairs would range in size between 10^2 and 10^3. Hence, they do not scale to larger size of data from ChIP-seq experiment and their fundamental methodologies have undergone reconstruction. However, most ChIP-tailored algorithms emphasize computational efficiency, and they sacrifice accuracy of motif detection because they use heuristics to speed up their computation.

The model-based methods employ either the EM algorithm [116] or Gibbs sampling [73]. The main computational load arises in the process of computing the posterior probabilities over all fixed-length subsequences at each iteration. STEME [105], a ChIP-tailored version of MEME, uses a branch-and-bound technique so that negligible oligomers with significantly low probabilities are effectively removed. The word-count methods, regardless of old or new, rely on essentially the same strategy. All possible oligomers are counted with exact or the fuzzy matching for input sequences. Then, overrepresented oligomers are determined against background sequences. Similar motifs are merged to generate output motifs. To reduce the computational load in the counting operation, DREME [115] and CisFinder [111] adopt similar strategies. Starting from $\simeq 100$ oligomers with no wildcards, each oligomer is either left or removed recursively by adding a wildcard and by assessing its significance in turn. Such methods run the risk of missing important motifs in earlier steps of the recursion. Hegma [61] is the fastest of current developed algorithms. A highly specific strategy based on Gray codes [46] is employed to avoid fuzzy matching so as to speed up the merging of similar motifs. However, this novel idea results in a degradation of the detection accuracy as will be shown later.

The aim of this study is to confirm high detection accuracy of a new proposed algorithm while keeping the computational efficiency at an acceptable level. In particular, the proposed method is designed to detect *many diverse* motifs that previous methods are unable to discover. The RPMCMC algorithm shown in the chapter 2 is a parallel version of the widely-used Gibbs motif sampler. One critical drawback of the standard Gibbs sampling, as with the EM algorithm, arises from the following fact: the posterior distribution can be considered highly multimodal because many diverse motifs are possibly present in given sequences. Once the

generated Markov chain is stuck in a locally high probability region, it is difficult to escape from that region within a finite time. This problem has received little attention in previous studies. MEME adopts a serial implementation of the EM algorithm that repeats the search with different initial conditions [116]. To reduce the possibility of becoming trapped in the same local optima, low prior probabilities are assigned to already-discovered motif sites in consecutive serial runs. However, such iterative methods take too long to be used for large ChIP data.

Again, RPMCMC is run on parallel interacting Gibbs samplers. A repulsive force comes into play when the trajectories of different chains near each other. Therefore, different chains are facilitated to explore entire regions. Compared to the original method using a single chain, this all-at-once interacting parallel run can detect much more diverse motifs. In addition, the proposed method has other unique characteristics, for instance, automated control of variable-length motifs, and the fast-clustering algorithm for many generated motifs in the summarization step. The method was comprehensively tested on synthetic promoter sequences and 228 TF ChIP-seq datasets of the ENCODE project. In the synthetic promoter analysis, RPMCMC found around 1.5 times as many embedded motifs as existing methods did. For the ChIP-seq datasets, the RPMCMC algorithm reported 444 reliable cofactors in total, 219 of which were not discovered by either of the recently published ChIP-tailored algorithms: DREME and Hegma.

3.2 Proposed method

3.2.1 ZOOPS model

We use the ZOOPS model [116] that allows zero or one motif occurrence per sequence. Assume that we are given a set of n sequences, $S^+ = \{s_1^+, \ldots, s_n^+\}$, where sequence s_i^+ is of length L_i ($i = 1, \ldots, n$). The reverse complement of the given sequence set is denoted by $S^- = \{s_1^-, \ldots, s_n^-\}$. Our model uses the set of n concatenated sequences, $S = \{s_1, \ldots, s_n\}$, where $s_i = (s_i^+, s_i^-)$ ($i = 1, \ldots, n$). The motif presence indicator z_i takes the value one or zero according to the presence or absence of a motif in sequence s_i. In a sequence s_i with $z_i = 1$,

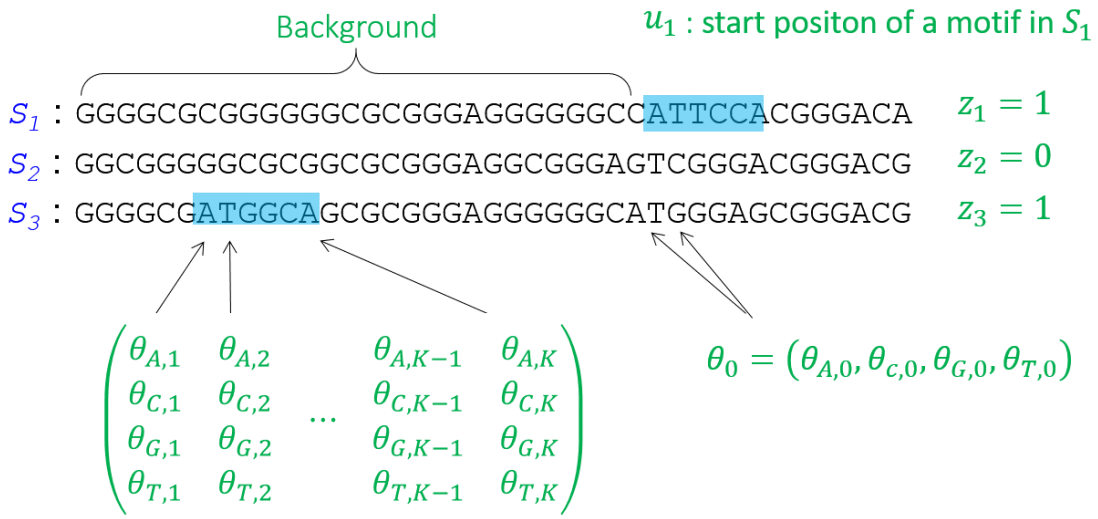


Fig. 3.4 The schematic view of the ZOOP model. Here, although all sequences look same length, there is no such a restriction.

a K-mer motif is positioned at the start site $u_i \in \{1, \ldots, L_i - K + 1, L_i + 1, \cdots, 2L_i - K + 1\}$. The kth element of the motif follows the position-specific multinomial distribution with $\theta_k = (\theta_{k,a}, \theta_{k,c}, \theta_{k,g}, \theta_{k,t})^\mathsf{T}$, which represents the nucleotide preference of the kth element to A, C, G, T. Thus, we have $\Theta = (\theta_1, \ldots, \theta_K)$, a position probability matrix (PPM). We treat the motif length K as an unknown parameter. The background sequences are assumed to follow independent multinomial trials with the background probability denoted by $\theta_0 = (\theta_{0,a}, \theta_{0,c}, \theta_{0,g}, \theta_{0,t})^\mathsf{T}$.

Given an input S, the objective is to estimate the PPM Θ with the unknown motif length K and the background probability θ_0 where the latent variables comprise $U = \{u_1, \ldots, u_n\}$ and $Z = \{z_1, \ldots, z_n\}$. The likelihood is then

$$p(S|U, Z, K, \Theta, \theta_0) \propto \prod_{\sigma \in \{a,c,g,t\}} \theta_{0,\sigma}^{\sum_{i=1}^{n} \sum_{j=1}^{2L_i} I(s_{i,j} = \sigma)} \times \prod_{k=1}^{K} \prod_{\sigma \in \{a,c,g,t\}} \left(\frac{\theta_{k,\sigma}}{\theta_{0,\sigma}} \right)^{\sum_{i=1}^{n} z_i I(s_{i,u_i+k-1} = \sigma)}, \tag{3.1}$$

where $s_{i,j}$ denotes the types of bases at the jth position in s_i, and $I(\cdot)$ is the indicator function. The first component of the right-hand side in the first line is the probability of all letters in the n input sequences, which is calculated under the background multinomial distribution. The second component is the likelihood ratio that assesses overrepresentation of the K-mer segmented sequences against the background.

As the priors for the multinomial likelihood, we use the Dirichlet distributions

$$\begin{aligned} p(\Theta|K) &\propto \prod_{k=1}^{K} \prod_{\sigma\in\{a,c,g,t\}} (\theta_{k,\sigma})^{\beta_{k,\sigma}-1}, \\ p(\theta_0) &\propto \prod_{\sigma\in\{a,c,g,t\}} (\theta_{0,\sigma})^{\alpha_\sigma-1}, \\ p(K=j) &= \frac{I(K_{\min}\le j\le K_{\max})}{K_{\max}-K_{\min}+1}, \end{aligned}$$

where $\beta_k=(\beta_{k,a},\beta_{k,c},\beta_{k,g},\beta_{k,t})^{\mathsf{T}}$ $(k=1,\cdots,K)$ and $\alpha=(\alpha_a,\alpha_c,\alpha_g,\alpha_t)^{\mathsf{T}}$ are the concentration parameters fixed at set values. The prior on Θ is conditioned by the motif length K. The equal probabilities are assigned to any K with a range between the predetermined minimum and maximum motif lengths, $K_{\min}$ and $K_{\max}$.

To complete the joint posterior of all the unknown parameters, we prescribe the priors on U and Z as follows:

$$\begin{aligned} p(u_i=u|K) &= \frac{1}{2(L_i-K+1)} \text{ for } i=1,\ldots,n, \\ p(z_i=1|K) &= \gamma^K \text{ for } i=1,\ldots,n. \end{aligned}$$

The start site u_i of a motif occurs with equal probability in all the possible positions in sequence s_i. The motif presence indicator z_i follows the binomial distribution with the success probability γ^K for each i $(0\le\gamma\le 1)$.

Note that although a specific type of modeling is presented here, our current program allows for a certain amount of flexibility in the model specification. For instance, a user can choose a higher-order Markov background model up to third order [23] and a position specific prior for the motif start sites [7].

3.2.2 Multi-modality of posterior distribution

Let x denote all the unknowns, U, Z, K, Θ and θ_0. To obtain an approximate estimate for the posterior $p(x|S)$, we can employ the Gibbs sampler with an data augmentation explained in chapter 2.2.4. However, the Gibbs motif sampler has a serious drawback in that inherent presence of a great many different motifs causes a complex energy landscape of the posterior distribution. In particular, once the trajectory of a Markov chain comes around a locally high-probability region which corresponds to one of the existing motifs, it is difficult to effect a transfer into another region within a finite runtime. The EM algorithm might exhibit the same defects.

As an illustration, we show a result of the simple Gibbs motif sampling. The dataset consists of 10^3-bp-long synthetic promoter sequences of 300 human genes. The Gibbs sampling was repeated 20 times under different initial conditions. As shown in Fig. 3.5, *all* the chains were trapped at similar AT-rich motifs for a long duration. Exceedingly high probabilities might be concentrated on the AT-rich segments and all the chains were absorbed to those domains of the posterior distribution. This is a typical scenario. Fig. 3.5 also shows the result of RPMCMC, which was run with 20 interacting parallel Gibbs samplers as described in chapter 2.3.5. By performing just an all-at-once parallel run, RPMCMC could capture much more diverse motifs than the independent Gibbs sampling could.

3.2.3 Inference with the RPMCMC

The RPMCMC algorithm is again shown here to denote an augmented system $\pi_A(x_1, \ldots, x_M|\beta)$ which consists of M exact copies $\pi_i(x) = p(x|S)$ $(i = 1, \ldots, M)$ of the posterior distribution and the repulsive force function $\psi(x_1, \ldots, x_M)$:

$$\pi_A(x_1, \cdots, x_M|\beta) \propto \prod_{i=1}^{M} \pi(x_i)\psi(x_1, \ldots, x_M)^{\beta}, \quad \beta \geq 0. \tag{3.2}$$

The repulsive force function is defined as a function of PPMs, $\psi(x_1, \ldots, x_M) \equiv \psi(\Theta_1, \ldots, \Theta_M)$. Let $D(\Theta_i, \Theta_j)$ be an increasing function of the dissimilarity between Θ_i and Θ_j. With this,

Fig. 3.5 A drawback of the independent Gibbs motif sampler, which is highlighted on 300 promoter sequences. The top and bottom panels display the processes of produced PPMs (sequence-logos) for RPMCMC with 20 replicas and independent Gibbs sampling under 20 different initial conditions. Five of the 20 sampling paths are shown for each method.

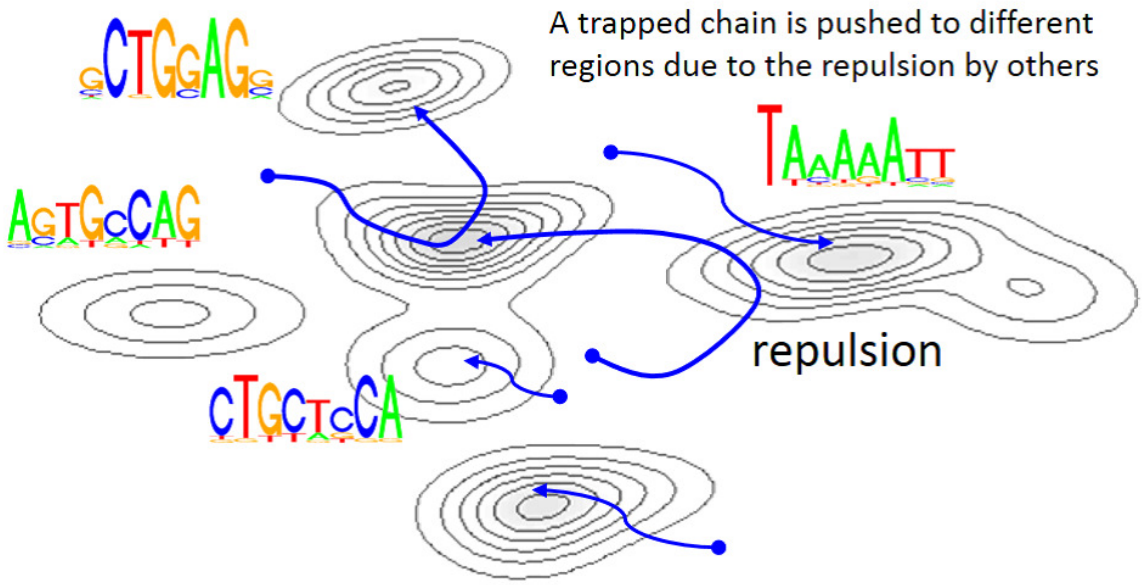


Fig. 3.6 A schematic view of the RPMCMC algorithm.

the repulsion is modeled by

$$\psi(x_1,\ldots,x_M)=\prod_{i=1}^{M}\exp(\min_{j:j<i}D(\Theta_i,\Theta_j)). \tag{3.3}$$

Replica i interacts with its nearest neighbor j^* such that $j^*=\arg\min_{j:j<i}D(\Theta_i,\Theta_j)$. The dissimilarity D is measured by

$$\begin{aligned} D(\Theta_i,\Theta_j) &= \frac{1}{K^*}\Big(\min_{(k,h)\in\mathscr{A}}\|\Theta_{i,k:k+K^*}-\Theta_{j,h:h+K^*}\|_F \\ &\quad + c\times|K_i-K_j|\Big), \end{aligned} \tag{3.4}$$

where $\mathscr{A}=\{(k,h)|k=1,\ldots,\max(1,K_j-K_i+1),h=1,\ldots,\max(1,K_i-K_j+1)\}$ and $K^*=\min\{K_i,K_j\}$. In general, K_i and K_j, the column sizes of Θ_i and Θ_j, are different. The distance of the PPMs is assessed after associating a smaller-sized PPM with the same-sized submatrix of the other as in Fig.3.7, $\Theta_{i,k:k+K^*}$ and $\Theta_{j,h:h+K^*}$, and by choosing the smallest Frobenius norm in all possible alignments $(k,h)\in\mathscr{A}$. The second term is a gap penalty for the difference of motif lengths where $c>0$.

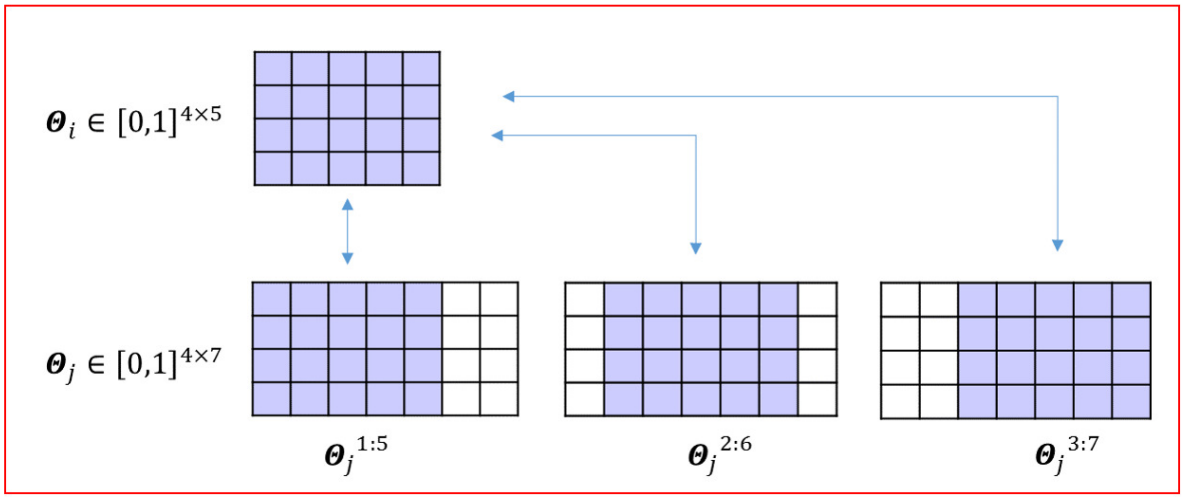


Fig. 3.7 The pairing rule when measuring the distance between two different sized matrices. In this example, Θ_j has larger column, it is considered to have three positions to be aligned. The first term in Eq. 3.4 returns the minimum values of Frobenius norms of differences in three pairs of matrices.

To obtain an estimate from the augmented posterior, Gibbs sampling is combined with several techniques such as the reversible-jump MCMC method [47] and the slice sampler [88]. The full details of the RPMCMC procedure are described in the following paragraph.

The RPMCMC algorithm renews each of the M replicas, $x_m = \{U^m, Z^m, K^m, \Theta^m, \theta_0^m\}$ ($m = 1, \ldots, M$) by alternatively drawing a sample from the full conditional distribution of the extended target (Eq. 3.2) while fixing all others at the current values. The procedure for sampling U^m, Z^m, and θ_0^m is the same as that for the conventional Gibbs motif sampler because these components are independent of the repulsive force function. The procedure for updating Θ and K involves a reversible-jump MCMC algorithm and a slice sampler, which are used to treat the varying-width PPM and the repulsive force acting upon the PPMs. Each sampling procedure is detailed while omitting the replica index ($m = 1, \ldots, M$) except for the description of Θ and K:

Motif start site: The conditional posterior probability of u_i $(i = 1, \ldots, n)$ given the others is

$$p(u_i|x \setminus \{u_i\}, S) \propto \prod_{k=1}^{K} \prod_{\sigma \in (a,c,g,t)} \left(\frac{\theta_{k,\sigma}}{\theta_{0,\sigma}} \right)^{I(\sigma = s_{i,u_i+k-1})}, \quad (3.5)$$
$$u_i \in \{1, \ldots, L_i - K + 1, L_i + 1, \ldots, 2L_i - K + 1\}.$$

We draw a value of u_i after normalizing the above so as to sum up to one over $u_i \in \{1, \ldots, L_i - K + 1, L_i + 1, \ldots, 2L_i - K + 1\}$. The process of sampling u_i $(i = 1, \ldots, n)$ can be fully-parallelized into n independent calculations. We run this with an OpenMP implementation of multi-threading.

Motif presence indicator: The conditional posterior of each z_i $(i = 1, \ldots, n)$ is

$$p(z_i|x \setminus \{z_i\}, S) \propto \begin{cases} \gamma^K \prod_{k=1}^{K} \prod_{\sigma} (\frac{\theta_{k,\sigma}}{\theta_{0,\sigma}})^{I(s_{i,u_i+k-1} = \sigma)} & (z_i = 1) \\ 1 - \gamma^K & (z_i = 0) \end{cases}$$

This process was also parallelized into n independent threads in the OpenMP implementation.

PPM: For the mth replica, the kth column θ_k^m of PPM is updated by drawing a sample from the following full conditional distribution:

$$\theta_k^m \sim p(\theta_k^m | x_m \setminus \{\theta_k^m\}, S) \propto \prod_{\sigma \in \{a,c,g,t\}} \theta_{k,\sigma}^{m \sum_{i=1}^{n} I(s_{i,u_i^m+k-1} = \sigma) + \beta_{k,\sigma} - 1} \exp(\min_{j: j \neq m} D(\Theta^m, \Theta^j)),$$

for $k = 1, \ldots, K$.

Here, it is impossible to obtain samples directly from this analytically intractable distribution which is the product of a Dirichlet density function and the repulsive function. The current version of RPMCMC uses a slice sampler.

Without loss of generality, we let $\theta_{k,t}^m = 1 - \theta_{k,a}^m - \theta_{k,c}^m - \theta_{k,g}^m$ be fixed. In the slice sampler, for each $\sigma \in a, g, c$, a new $\theta_{k,\sigma}^m$ is sampled from the uniform distribution in the range between $\theta_{\min}$ and $\theta_{\max}$ as follows:

(i) Generate ε from the uniform distribution in the range $[0, 1]$.

(ii) Find $\theta_{\max}$ such that $p(\theta_{\max}|x \setminus \theta^m_{k,\sigma}, S) = \varepsilon \times p(\theta^m_{k,\sigma}|x \setminus \theta^m_{k,\sigma}, S)$ by incrementing $\theta_{\max}$ from the current $\theta^m_{k,\sigma}$ by a small value δ.

(iii) Find $\theta_{\min}$ such that $p(\theta_{\min}|x \setminus \theta^m_{k,\sigma}, S) = \varepsilon \times p(\theta^m_{k,\sigma}|x \setminus \theta^m_{k,\sigma}, S)$ by decrementing $\theta_{\min}$ from $\theta^m_{k,\sigma}$ by δ.

(iv) Generate the new $\theta^m_{k,\sigma}$ from the uniform distribution in the range $[\theta_{\min}, \theta_{\max}]$.

Background probability: θ_0 is updated by drawing a Dirichlet random variable from the full conditional distribution

$$p(\theta_0|x \setminus \{\theta_0\}, S) \propto \prod_{\sigma \in \{a,c,g,t\}} \theta_{0,\sigma}^{\sum_{i=1}^{n}(\sum_{j=1}^{2L_i} I(s_{i,j}=\sigma) - \sum_{k=1}^{K} I(s_{i,u_i+k-1}=\sigma)) + \beta_{0,\sigma} - 1}.$$

Motif length: Θ and K are updated by the reversible jump MCMC algorithm [47], as shown in the following procedure:

Suppose that the currently obtained Θ has the width $K = k$. To renew $\{\Theta, k\}$ to $\{\Theta^*, k^*\}$ at a step, we first generate a candidate according to the proposal distribution:

$$q(\Theta^*, K^* = k^* \mid \Theta, K = k) = \begin{cases} q_{lc} & k^* = k-1 : \Theta^* = \Theta_{2:k} \\ q_{rc} & k^* = k-1 : \Theta^* = \Theta_{1:k-1} \\ q_0 & k^* = k : \Theta^* = \Theta \\ q_{le} & k^* = k+1 : \Theta^* = (\theta^*, \Theta) \\ q_{re} & k^* = k+1 : \Theta^* = (\Theta, \theta^{**}) \end{cases}$$

The first two transitions indicate the contraction proposals that drop the first and last columns of $\Theta = (\theta_1, \Theta_{2:k}) = (\Theta_{1:k-1}, \theta_k)$, respectively, from the current Θ. The third transition is to retain the current state. The last two transitions expand the size of the current PPM to $K^* = k+1$ by adding the new components, θ^* and θ^{**}, to the leftmost and rightmost columns of Θ. Conditioned by Z, U, and $K = k$, these additional components are given by the frequencies of each nucleotide at the first and last elements of the currently occupied

motif region:

$$\theta^*_\sigma = \frac{\sum_{i=1}^n I(s_{i,u_i-1}=\sigma)}{\sum_{\sigma'}\sum_{i=1}^n I(s_{i,u_i-1}=\sigma')} \text{ and } \theta^{**}_\sigma = \frac{\sum_{i=1}^n I(s_{i,u_i+k}=\sigma)}{\sum_{\sigma'}\sum_{i=1}^n I(s_{i,u_i+k}=\sigma')}.$$

After one of the move types is chosen according to the probabilities $q_{lc}, q_{rc}, q_0, q_{le}, q_{re}$, we determine the acceptance or rejection according to the probability,

$$\alpha(\Theta^*) = \min\left(1, \frac{p(U,Z,K=k^*,\Theta^*,\theta_0|S)q(\Theta,K=k|\Theta,K=k)}{p(U,Z,K=k,\Theta,\theta_0|S)q(\Theta^*,K=k^*|\Theta^*,K=k^*)}\right).$$

If accepted, $\Theta^* \to \Theta$, and otherwise $\Theta \to \Theta$.

We can discover a much wider variety of motifs with an all-at-once interacting parallel simulation than with the independent method. Conventional Gibbs sampling with M different initial seeds (as shown in the previous subsection) can be derived by setting the zero force severity, $\beta = 0$, to RPMCMC.

Our current implementation does not parallelize the process of updating the M Markov chains. We use multi-core processors only for counting the nucleotide frequencies when renewing the motif start sites.

Post-processing: Clustering and Ranking

RPMCMC produces many redundant outputs with slight variations. We reduce the redundancy by grouping the outputs into g clusters, $C_1, \ldots, C_g$, based on the dissimilarity of the sampled PPMs. The procedure is as follows (see Fig. 3.8 for a schematic illustration):

(i) Samples of size $p = M \times N$ are arranged as $\eta = \{x^{(1)}, \ldots, x^{(p)}\}$ by sorting realized values of the likelihood (Eq. 3.1) in decreasing order.

(ii) Set $\lambda > 0$, a threshold for the within-cluster variability.

(iii) Set $k = 1$, and repeat (a)-(d) until no samples are left:

  (a) Initiate the kth cluster $C_k = \{x^{(1)}\}$ by a singleton of the sample that is ranked first in η. Let $\mu_k = x^{(1)}$ be the cluster representative.

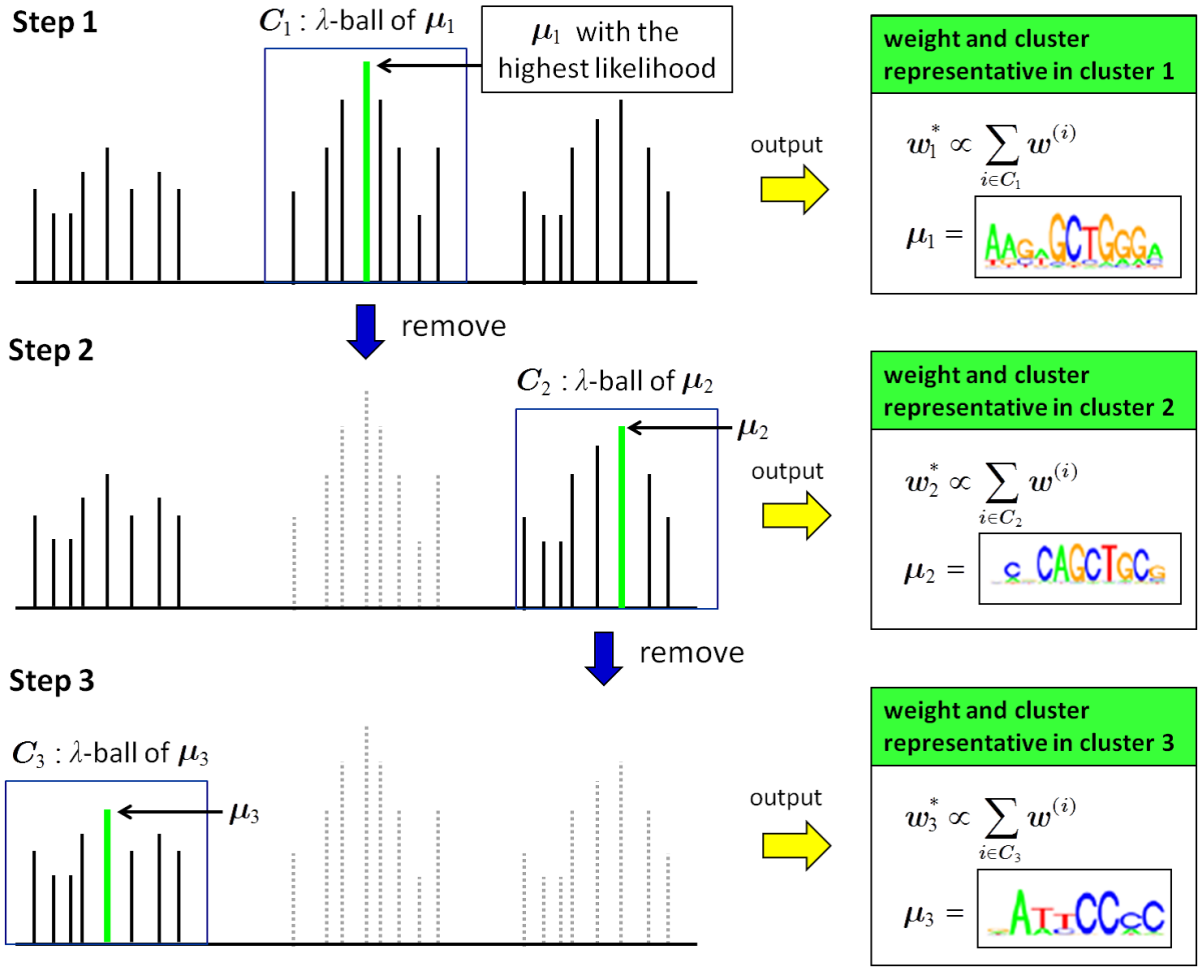


Fig. 3.8 A schematic illustration of the post-processing process.

(b) Collect all samples satisfying the condition $D(\Theta^{(1)}, \Theta^{(i)}) \leq \lambda$ where $\Theta^{(i)}$ denotes the PPM of $x^{(i)}$. These samples are integrated into cluster C_k; $C_k = \{x^{(i)} | D(\Theta^{(1)}, \Theta^{(i)}) \leq \lambda, i = 1, \ldots, p\}$.

(c) Discard the collected samples in C_k from the ordered sequence; $\eta \leftarrow \eta \setminus C_k$. Let p be the length of η, and rearrange η according to the likelihood values.

(d) If η is empty, terminate the computation. Otherwise, let $k \leftarrow k+1$ and go back to step (a).

The method operates with a single input parameter λ that controls the number g of clusters. Samples within $D \leq \lambda$ are assigned to the cluster representative μ_k, which is the one to achieve the highest likelihood within the kth cluster members.

Denote the $p = M \times N$ samples and their importance weights by $\{x^{(i)}, w^{(i)}\}_{i=1}^{p}$. With the g reduced samples $\{\mu_1, \ldots, \mu_g\}$, we define an approximated posterior distribution by

$$\hat{p}(x|S) \propto \sum_{k=1}^{g} I(x = \mu_k) w_k^*, \quad w_k^* \propto \sum_{i \in C_k} w^{(i)}.$$

This is a mixture of the g probability mass functions $I(x = \mu_k)$ at μ_k. Mixing rate w_k^* is the sum of the importance weights associated with the corresponding cluster C_k. PPMs and the motif start sites in $\{\mu_1, \ldots, \mu_g\}$ are of primary interest for motif discovery. We generate a ranked list of the reduced discovered motifs which are ordered according to the weights w_k^*.

3.3 Experiment

Performance Evaluation

We report the performance of several motif discovery algorithms on two types of data: (i) promoter sequences into which strings generated from PPMs in the JASPAR CORE database are planted, and (ii) 228 TF ChIP-seq datasets of the ENCODE project. We evaluate the performance for each type of data as follows:

(i) Given the nucleotide positions of known and predicted motifs, recall (SN, sensitivity) and precision (PPV, positive predicted value) are evaluated at a nucleotide level. These criteria have commonly been used, for instance, in [121] (we use the abbreviations SN and PPV according to convention). For given J known motifs, we define slightly modified SN and PPV for the evaluation of multiple output motifs.

Let p_j be the output that achieves the most overlapping predicted sites with the jth known motif among the g outputs (if there are two or more outputs having the same number of overlapped nucleotides, the one with the higher rank given by a motif finder is chosen). Then,

the recall and the precision are computed as

$$\begin{aligned}
\mathrm{SN} &= \frac{1}{J}\sum_{j=1}^{J}\mathrm{SN}_j \text{ and } \mathrm{PPV} = \frac{1}{J}\sum_{j=1}^{J}\mathrm{PPV}_j, \\
\mathrm{SN}_j &= \frac{\#\text{ of nucleotides in motif } j \text{ overlapped by output } p_j}{\#\text{ of nucleotides in motif } j}, \\
\mathrm{PPV}_j &= \frac{\#\text{ of nucleotides in motif } j \text{ overlapped by output } p_j}{\#\text{ of nucleotides in output } p_j}.
\end{aligned}$$

A low SN statistic indicates the lack of ability to discover planted multiple motifs and a low PPV statistic can be a signal for less identification accuracy, for instance, the occurrence of over- or under-estimates of the planted motif regions.

(ii) From contiguous segments around the TFBSs of the primary TF in each dataset, we obtain a list of cofactor interacting motifs and their annotations that are implicated in the regulatory module of the primary TF. To identify the cooperative cofactors of the primary TF, each predicted motif (PPM) is matched to JASPAR CORE motifs by using the TOMTOM program [49]. For a given predicted PPM, TOMTOM outputs the matching scores to all annotated TFBSs (the name of TFs) in JASPAR with the statistical significances (E-values). For each algorithm, a diversity of the discovered motifs is evaluated with the number of known motifs in JASPAR CORE that are matched significantly to the produced PPMs with the acceptable level of significance at E-value less than 0.05.

In addition, we use the log-likelihood ratio (LLR) to evaluate K-mer binding sites of a predicted motif:

$$\mathrm{LLR}(U,K) = \sum_{k=1}^{K}\sum_{\sigma\in\{a,c,g,t\}} n' f_{k,\sigma}\log\left(\frac{f_{k,\sigma}}{b_\sigma}\right),$$

where $f_{\sigma,k}$ ($\sigma \in \{a,c,g,t\}$, $k \in \{1,\ldots,K\}$) is the relative frequency of nucleotides at each position in a predicted site, $b = (b_a, b_c, b_g, b_t)^{\mathsf{T}}$ is the relative frequency of nucleotides of the background. The output consists of n' motif subsequences. A higher LLR indicates a better likeliness of the K-mer instances to be a motif in terms of a combined characterization on the degree of overrepresentation relative to the background and the total information content.

3.3.1 Synthetic dataset

The performance of RPMCMC was tested on synthetic datasets against two ChIP-tailored algorithms, DREME and Hegma, and a classical algorithm, Weeder. The datasets were derived from non-redundant sets of randomly selected $n \in \{300, 600, 1200, 2500, 5000\}$ promoter sequences obtained from UCSC.hg19 with two different kinds; one composed of fixed-length sequences of 1,000 bp and the other of variable-length sequences varied between 200 and 2,000 bp. Oligomers generated by randomly chosen 10 JASPAR CORE PPM collections were planted into randomly selected start sites so that each sequence has eight motifs on average. For each data size n, we prepared 20 different sequence sets. With this ground truth, we measured the change in recall and precision. All parameters of RPMCMC and the specified Weeder options are shown in Table 3.1. For DREME and Hegma, we employed the default parameters. The parameters of RPMCMC were empirically chosen.

Table 3.1 Default parameters of RPMCMC and Weeder options that were used in all experiments. Hegma and DREME were executed using the default settings.

RPMCMC	
parameter	*value*
prior on z_i	$\gamma = 0.755$
max/min motif width	$K_{\max} = 8, K_{\min} = 15$
Dirichlet priors	$\alpha_\sigma = 1, \beta_{k,\sigma} = 1$
# of replicas	$M = 50$
# of MCMC iterations	$N = 520$
burn-in period (fixed)	$N \leq 20$
repulsive force severity	$\beta = 10 \times \sum_i z_i$
motif clustering	$\lambda = 0.3$
gap penalty	$c = 0.3$

Weeder	
option	*value*
species code	HS
analysis type	medium

Fig. 3.9(a) summarizes the SN and PPV values as a function of n for RPMCMC, Hegma and Weeder. DREME was removed from this figure because there was no way of calculating

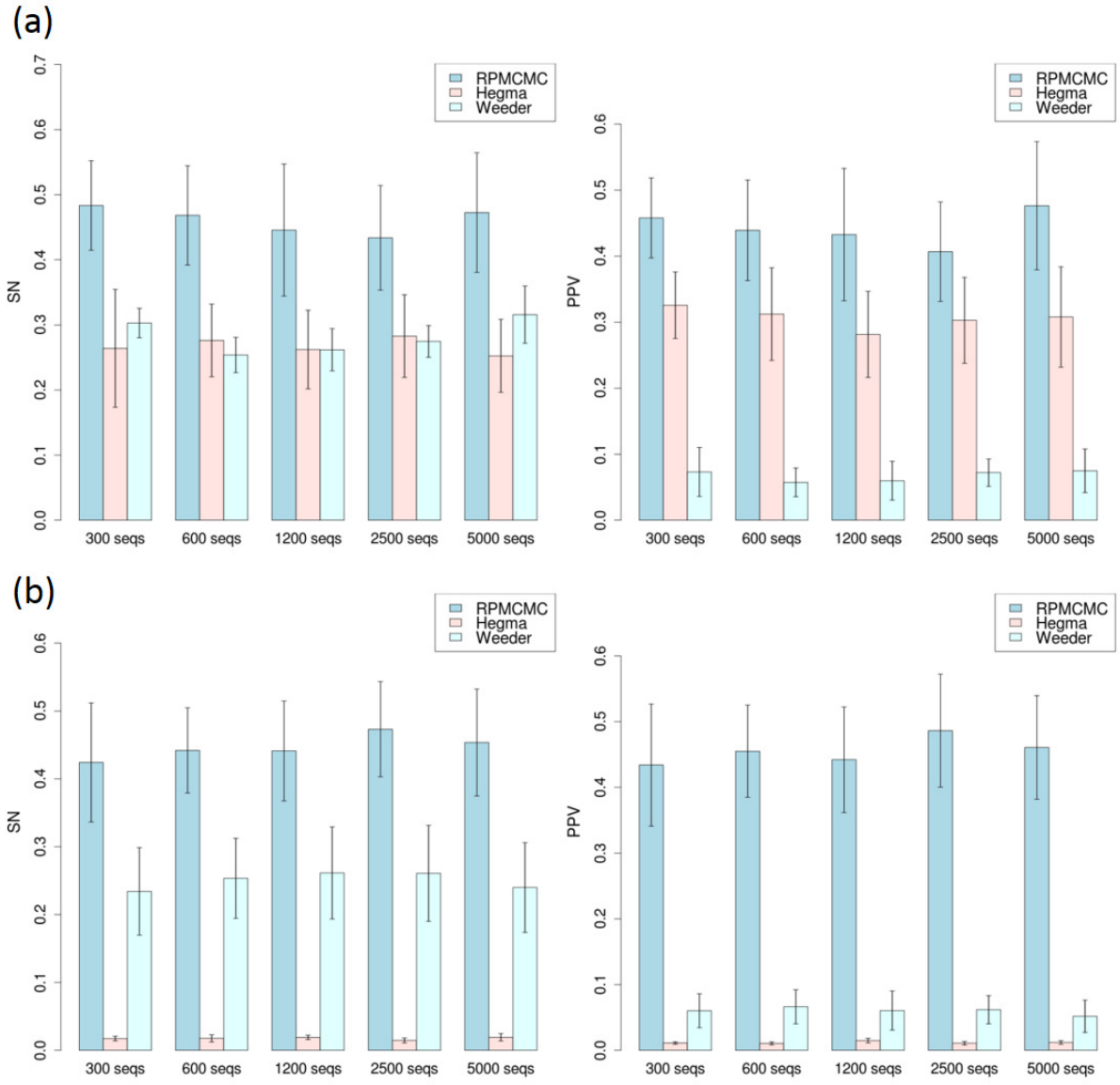


Fig. 3.9 Performance comparison among RPMCMC, Hegma and Weeder on synthetic datasets: (a) fixed-length sequence sets and (b) variable-length sequence sets. Motifs were generated according to the JASPAR CORE PPM collection and were inserted randomly into a set of promoter sequences. SN (left) and PPV (right) values of each method are plotted against the varying sequence sizes, $n \in \{300, 600, 1200, 2500, 5000\}$.

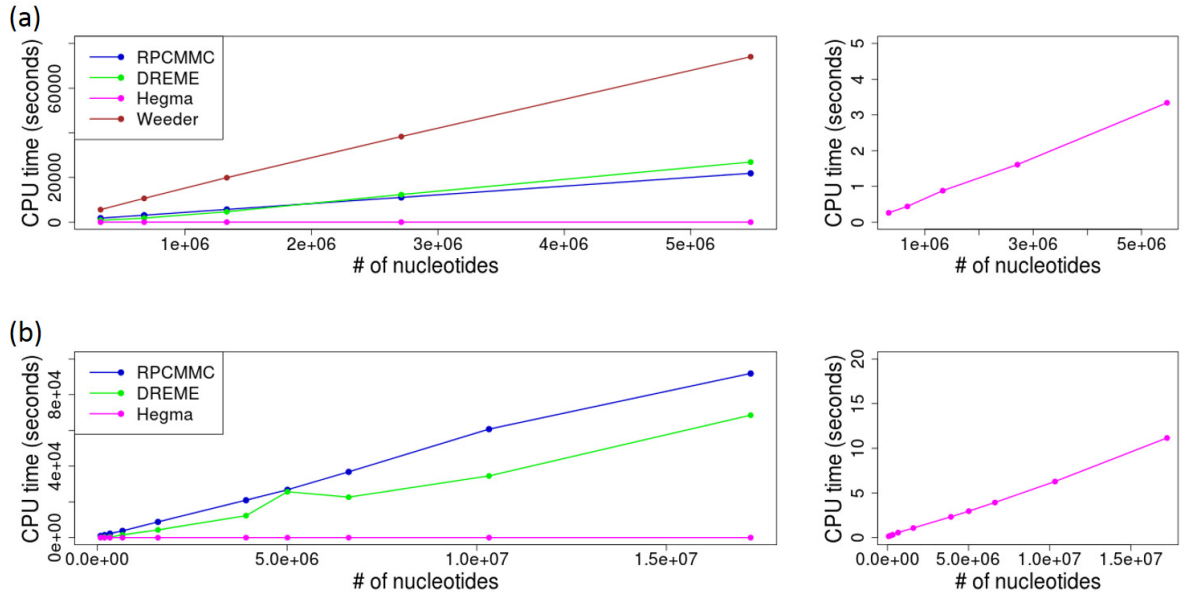


Fig. 3.10 Computational efficiency of RPMCMC, Hegma, DREME and Weeder (a) the synthetic promoter sequence and (b) the ChIP-seq datasets, shown as a function of the number of nucleotides. The vertical axis indicates CPU times. The right figure is an enlarged display of the left figure to make clear the computation time of Hegma.

SN and PPV due to the lack of outputs on motif sites in the distributed program. The numbers of outputs from RPMCMC, Hegma and Weeder were 85.7, 214.76 and 13.3 on average, respectively. It can be seen that RPMCMC outperformed the other methods. For the fixed-length datasets, RPMCMC delivered SN values around 1.7 times higher than those of the other two methods. The PPVs of RPMCMC were around 1.5 times higher than those of Hegma. As shown in Fig. 3.9(b), the results on the variable-length datasets were similar to those on the fixed-length datasets except that the performance of Hegma was significantly degraded.

We analyzed the cause of the observed low SN and PPV statistics for Hegma and Weeder, as illustrated with the results on the fixed-length datasets. It was found that Hegma has a strong tendency to divide planted regions of a motif into a few different predicted motifs. Such incorrectly fragmented outputs acted to increase PPV slightly but resulted in the observed low SN. A distinctive characteristic of Weeder is the fairly low PPV while several comparative studies reported Weeder to be one of the best performing algorithms among early motif finders [121]. A region predicted by Weeder tends to include not only a planted motif region but also many background regions. RPMCMC could achieve much higher SN and PPV than the others could.

Fig. 3.10(a) gives the computation time for each method. RPMCMC was implemented in `C++`. We used the `C` programs for DREME, Hegma and Weeder, which are available on the authors' websites. All the tests were conducted on Intel Xeon Phi™coprocessors with 61-core CPUs and 48 GB of main memory. In terms of computation efficiency, Hegma outperformed the others, and RPMCMC was comparable to DREME. In particular, the computation times of RPMCMC and DREME were about a ten-thousandth those of Hegma. RPMCMC would sustain an acceptable level of computation time, and furthermore, it might be possible to render the algorithm more efficient. The bottleneck in RPMCMC is in the process of calculating the posterior probabilities of the motif start sites u_i (see details in Supplementary Method S1): with a given PPM, $K \times \sum_i 2(L_i - K + 1)$ times calculations were necessary to perform in every iteration over all possible K-mer consecutive subsequences in S. This process can fully be parallelized into independent processing elements. Alternatively, we

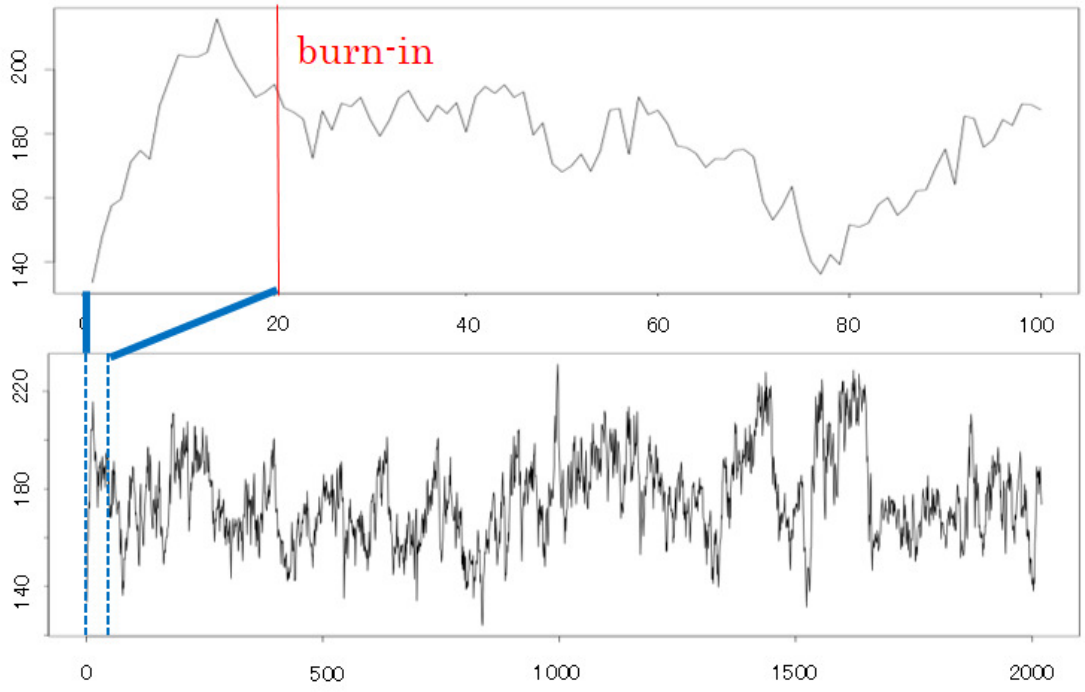


Fig. 3.11 Series of the likelihood values in RPMCMC for a synthetic dataset with 300 sequences. Default burn-in is set at 20 steps (red vertical line).

could use a branch-and-bound technique as in STEME that effectively prunes subsequences with negligibly low probabilities.

We remark on the difficulty in detecting the burn-in time for RPMCMC. An initial portion of the Markov chain samples should be discarded because the chain approaches its stationary distribution [22] following a sufficient burn-in period. Fig. 3.11 shows the process of evolving the likelihood during a RPMCMC run. The series of the likelihood values remained instable, which indicates a fairly slow mixing of the Markov chain because the target distribution was inherently multimodal and the parallel interacting chains switched their target local modes successively. In general, it is difficult to deal with a diagnostic of burn-in periods that looks for multimodality of the posterior distribution. At the current moment, we do not have a specific idea other than an obvious approach of giving as long as possible for a trial move.

3.3.2 ChIP-Seq data

Using RPMCMC with the default parameters given in Table 3.1, we predicted the cofactor motifs of the primary TF for each of the 228 datasets of ChIP-seq experiments in the ENCODE project [20]. FASTA files were produced by clipping the sequences of UCSC.hg19 at the locations recorded in SYDH TFBS narrowPeak files (available from NCBI's Gene Expression Omnibus using the accession number GSE31477). We removed datasets that had only a few sequences after removing fragments with lengths less than 200 or more than 500 from the obtained FASTA files. Also, we removed datasets which have more than one percent of sequences including blacklist regions reported on
https://sites.google.com/site/anshulkundaje/projects/blacklists. In this way, we obtained the 228 datasets from the total of 359 datasets. The numbers of the input sequences ranged from 205 to 49,211. RPMCMC produced 51–149 output motifs for each dataset. A discovered motif, for instance, $\{U_k, \Theta_k\}$ in μ_k, was regarded as being significantly enriched if it appeared in 5% or more of the input sequences, i.e. $\sum_{i=1}^{n} z_i/n \geq 0.05$. At the acceptable level of significance on the TOMTOM's E-values $\leq$0.05, approximately 15 significantly enriched outputs on average could have correspondence to one of the experimentally validated TFBSs in JASPAR CORE.

In the experiments, Hegma produced a far greater number of outputs (1,081 outputs on average over all datasets) than RPMCMC (110 outputs) and DREME (49 outputs). The outputs of Hegma possibly included many redundant motifs. Removing motifs with $\sum_{i=1}^{n} z_i/n < 0.05$ from the total outputs, the average numbers of outputs of Hegma, RPMCMC and DREME dropped to 24, 110 and 33, respectively.

The computation times of each algorithm for 10 selected datasets including the smallest and the largest dataset are shown in Fig. 3.10(b). Compared to the experiment with the synthetic datasets, the computation times of RPMCMC were a little inferior to those of DREME for the ChIP-seq datasets. RPMCMC would still sustain an acceptable level of computation time. As discussed in the previous subsection, the current implementation of the RPMCMC algorithm is yet to be optimized for speed.

			P/A	
predicted motif	*E-value*	*ranking*	*Hegma*	*DREME*
SP1	2.71×10^{-4}	1	P	P
EGR1	1.09×10^{-3}	1	P	P
SP2	1.10×10^{-3}	1	P	P
KLF5	3.75×10^{-3}	1	P	P
NRF1	3.80×10^{-9}	2	P	P
FUS3	5.07×10^{-3}	2	P	P
E2F4	4.52×10^{-2}	45	A	P
REST	3.92×10^{-3}	47	A	A
GABPA	1.57×10^{-2}	51	A	A
DAF-12	1.41×10^{-2}	56	A	A
MET31	1.56×10^{-2}	62	A	A
RPN4	3.49×10^{-2}	62	A	A
TYE7	1.43×10^{-3}	70	A	A
PIF5	4.35×10^{-2}	70	A	A
USF1	4.56×10^{-2}	70	A	A
RDS1	3.37×10^{-2}	71	A	A

Table 3.2 A list of 16 predicted motifs obtained by RPMCMC that are implicated in the transcriptional module of NRF1 in HepG2. NRF1 is the ChIPed TF and the rest are the predicted cofactors. All motifs, which could be annotated at E-value ≤ 0.05 according to JASPAR, are shown with the E-values of TOMTOM (second column) and the ranking by RPMCMC (third column). The last two columns indicate the presence (P) or absence (A) of the motif in the outputs of Hegma and DREME, respectively.

As shown in Fig. 3.12(a), the numbers of known motifs significantly matched to the outputs of RPMCMC (E-values ≤ 0.05) were larger than those of Hegma and DREME for 74% of the 228 datasets. While RPMCMC produced the largest numbers of outputs among the three methods, the LLR values of the discovered motifs of RPMCMC were much higher than those of the others as in Fig. 3.12(b). This indicates that RPMCMC has a great potential to mine many reliable diverse motifs that are undetected by the existing methods.

Table 3.2 shows 15 cofactors that were predicted by RPMCMC on a ChIP dataset (wgEncodeSydhTfbsHepg2Nrf1IggrabPk) in which the binding sites of NRF1 were studied in HepG2. The binding sites of RPN4 and USF1 were detected only by RPMCMC. It was reported that both RPN4 and NRF1 are involved in the same proteasome activity [102, 131], and the interaction of USF1 and Nrf1 is involved in the transcriptional regulation of FMP1 gene [98].

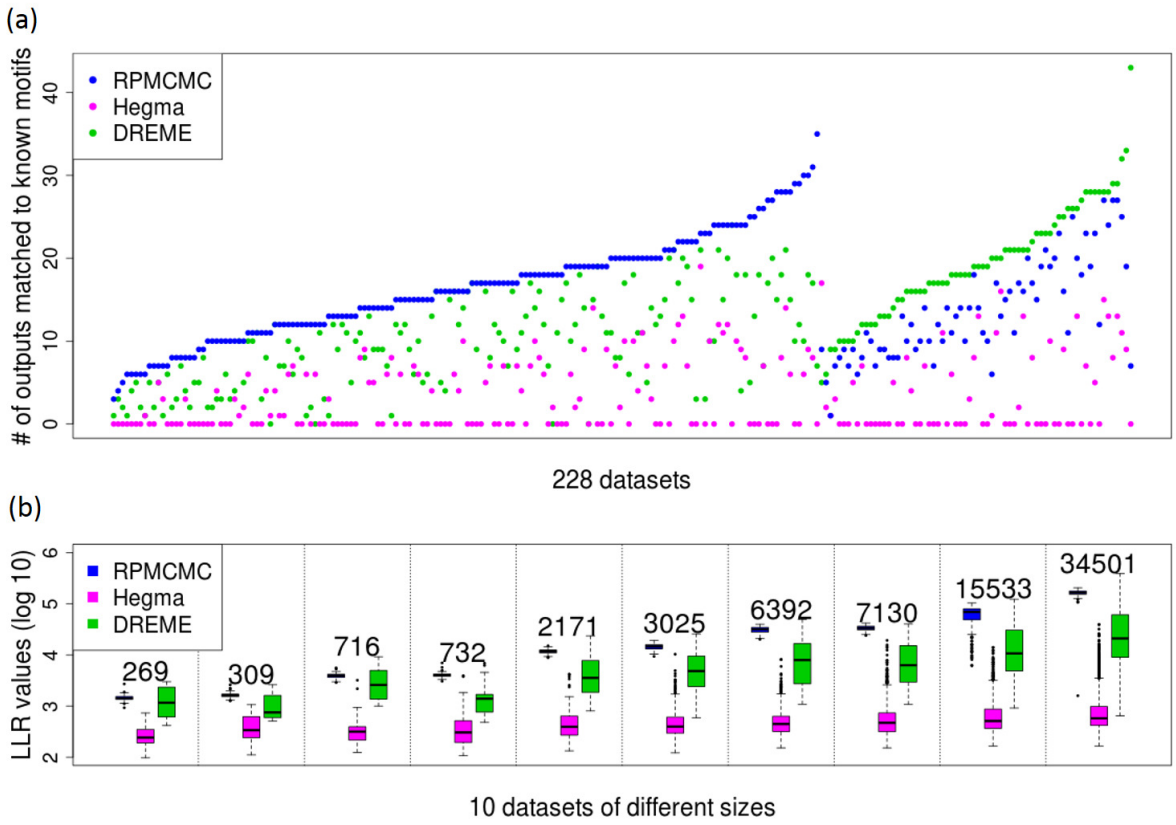


Fig. 3.12 Comparison of RPMCMC with Hegma and DREME on the 228 ENCODE datasets. (a) The number of motifs in JASPAR CORE that were matched to outputs of each algorithm for each of the 228 datasets (blue: RPMCMC; magenta: Hegma; green: DREME). The datasets are arranged by gathering together the subsets with which each method achieved the most matching to JASPAR. (b) The LLR values of the predicted sites are shown across arbitrary-chosen 10 datasets with different sizes ($\log_{10}$). Each number on the box indicates the number of sequences in each dataset.

Fig. 3.13 summarizes the detection ability to discover diverse motifs based on a Venn diagram of all matching motifs produced from the analyses of the 228 datasets. The outputs of RPMCMC contained almost all of the outputs of DREME and Hegma, and, notably, 219 annotated cofactors were uniquely discovered by RPMCMC.

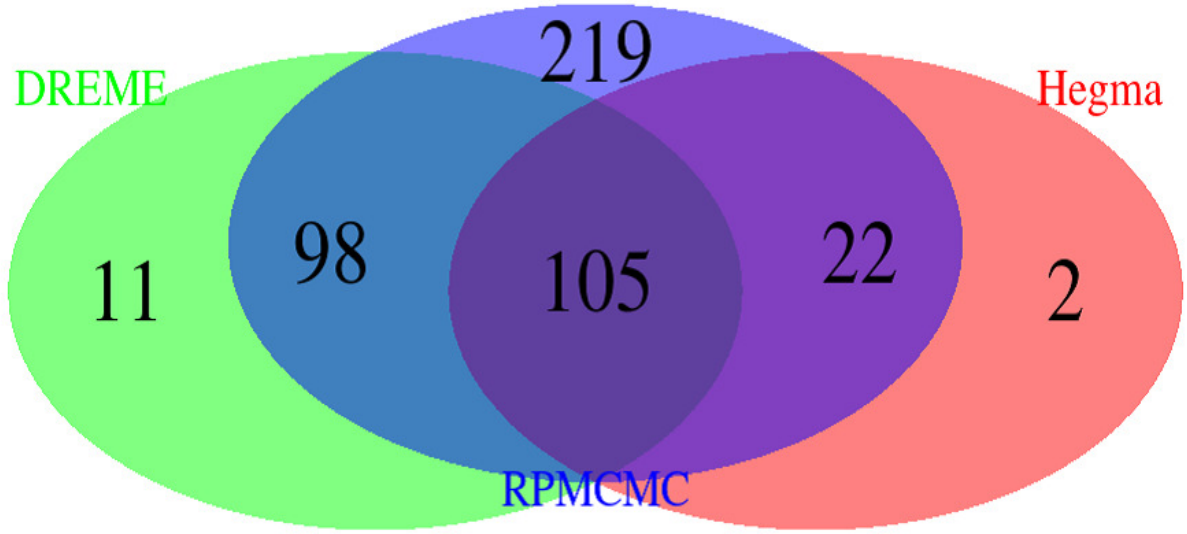


Fig. 3.13 Venn diagram for total numbers of significantly annotated motifs over all the 228 datasets, reported by RPMCMC, Hegma and DREME.

3.4 Conclusion

In the motif discovery problem, the direct use of a Gibbs sampling method revealed an inability to find latent diverse motifs even in a fairly small number of input sequences. In the application for only 300 input sequences, all simulations with different initializations became trapped in the AT-rich motifs which are of little significance in practice. This highlighted a critical drawback of the Gibbs sampling methods. The same is true for the EM algorithm. Because biological sequences generally contain rather diverse conserved patterns, which are sometimes biologically meaningless, the posterior distribution exhibits a very complex landscape as it includes many locally high probability regions. Our view is that solving this problem is the essence of improving the accuracy of motif discovery. Motivated by this, we presented a new motif discovery method called RPMCMC, which is a parallel variant of the widely used Gibbs motif samplers. The rather simple idea is to run the Gibbs motif samplers in parallel by making use of the repulsive force on different samplers. With

all-at-once sampling, we could discover diverse motifs by which the parallel samplers divide their responsibility in the overall search region.

As another contribution, we provided a list of predicted cofactor motifs that were overrepresented in the 228 ENCODE ChIP-seq datasets. RPMCMC can potentially mine promising annotated motifs which other word-count methods fail to find. To narrow down things to truly functional cofactor sets, it is necessary to conduct further validation experiments.

Chapter 4

Molecular design problem

4.1 Introduction

Computational molecular design has a great potential to promote enormous savings in time and cost in the discovery and development of functional molecules and assembles including drugs, dyes, solvents, polymers, and catalysis. The objective is to computationally create novel promising molecules that exhibit desired properties of various kinds, simultaneously. For instance, the chemical space of small organic molecules is known to consist of more than 10^{60} candidates. The problem entails a considerably complicated multi-objective optimization where it is impractical to fully explore the vast landscape of structure-property relationships. In general, the molecular design process involves two different types of prediction; the forward prediction is aimed at predicting physical, chemical and electric properties of a given molecular structure, and the backward prediction is to inversely identify appropriate molecular structures with the given desired properties. While the former design process is referred to as the quantitative structure-property relationship (QSPR) analysis, the latter is known as the inverse-QSPR analysis [14, 90, 68, 122, 123, 81, 66, 129]. In this study, a Bayesian perspective is employed to unify the forward and backward prediction processes. Therefore, the present method is called the Bayesian molecular design.

In the relevant areas called chemical or materials informatics, there have been extensive studies on the forward prediction; however, there has been considerably less progress made

in the backward prediction. An obvious approach to the inverse problem is the use of combinatorial optimization techniques. The objective is to minimize the difference between given desired properties and those attained by the designed molecules. Some previous studies tackled this issue with genetic algorithms (GAs) [90, 122, 123, 81, 66, 32, 87, 71] and molecular graph enumeration [129, 3]. Graph enumeration is generally less effective due to the combinatorial complexity of the design space. To narrow down the candidates, several ways of introducing a restricted class of molecular graphs have been investigated [129, 3]. Using GAs [125], which have been more intensively studied, searches for optimal or suboptimal designs by successively modifying chemical structures with genetic operators consisting of mutation, crossover, and selection.

The major difficulty of using a GA lies in the procedure of mutating molecules such that unfavorable structures are successfully excluded, for instance, unfavorable and/or unrealistic chemical bonds such as F-N and C=O=C. This issue is common to the graph enumeration. To avoid the emergence of unfavorable structures, exclusion rules were employed in some studies, particularly those aimed at the design of drug-like molecules [56, 67]. However, such rules might be incomprehensive, and it is impractical to establish a general rule of chemically favorable structures. A promising alternative is fragment assembly methods [122, 123, 81, 66, 30, 38]. In a structure manipulation step of these methods, randomly chosen substructures are replaced by fragments of existing compounds. While the fragment assembly methods have a certain appeal, as is evident from their widespread use, they suffer from critical disadvantages: (i) the design space is restricted to possible combinations of collected fragments, (ii) the use of a vast amount of fragments entails unacceptably large computational loads to homology search in the fragment exchange operation, and (iii) mutation and crossover operations require computationally intractable graph manipulations. The proposed method circumvents all these issues.

The Bayesian molecular design begins by obtaining a set of machine learning models that forwardly predict properties of a given molecule for multiple design objectives. These forward models on QSPR are inverted to the backward model through Bayes' law, combined with a certain prior distribution. This gives a posterior probability distribution for the inverse-

QSPR analysis, which is conditioned by a desired property region. Exploring high-probability regions of the posterior with the sequential Monte Carlo (SMC) method [28], molecules that exhibit the desired properties are computationally created. The most distinguished feature of this workflow lies in the backward prediction algorithm. In this study, a molecule is described by a ASCII string according to the well-known SMILES chemical language notation. To reduce the occurrence of chemically unfavorable structures, a chemical language model is trained, which acquires commonly occurring patterns of chemical substructures by the natural language processing of the SMILES language of existing compounds. The trained model is used to recursively refine SMILES strings of seed molecules such that the properties of the resulting molecules fall in the desired property region while eliminating the creation of unfavorable chemical structures. The key contributions of the newly proposed method are summarized below.

- *String-based structure refinement.* The string representation of molecules enables much faster structure refinements in the backward prediction than those based on graph representation.

- *Generator for chemically favorable structures.* The method is designed according to a fragment-free strategy. Structural patterns of known compounds or implied contexts of 'chemically favorable structures' are captured by the probabilistic model. Afterward, the resulting SMILES generator will be shown to be very effective in creating chemically plausible hypothetical molecules. The trained model serves as a substitute for a fragment library, and also forms the prior distribution in the Bayesian analysis.

The forward and backward predictions are pipelined in the R package *iqspr* which is provided at the CRAN repository [100]. The proposed method is illustrated through the design of small organic molecules exhibiting properties within prescribed ranges of HOMO-LUMO gap and internal energy. The properties of created molecules are verified with a quantum chemistry calculation based on density functional theory (DFT) as done for the design of organic photovoltaics in [51]. Finally, through case studies, we highlight a

significant issue and the ultimate goal of the machine learning-driven material discovery to which less attention has so far been paid.

4.2 Bayes law for molecular design

The objective of the backward prediction is to create a chemical structure S with p properties $\mathbf{Y} = (Y_1, \ldots, Y_m)^{\mathrm{T}} \in \mathbb{R}^m$ lying in a desired region U. The Bayesian molecular design relies on the statement of Bayes' law, which is sometimes called the inverse law of conditional probability,

$$p(S|\mathbf{Y} \in U) \propto p(\mathbf{Y} \in U|S)p(S). \tag{4.1}$$

This law states that the posterior distribution $p(S|\mathbf{Y} \in U)$ is proportional to the product of the likelihood $p(\mathbf{Y} \in U|S)$ and the prior $p(S)$. Exploring high-probability regions of the posterior, we aim to identify promising hypothetical structures S that exhibit the desired U.

Along with Eq. 4.1, three internal steps linking the forward and backward analyses are outlined (see also Fig. 4.1):

- *Forward prediction.* A set of QSPR models on the p properties is trained with structure-property relationship data. This defines the forward prediction model $p(\mathbf{Y}|S) = \prod_{j=1}^{p} p(Y_j|S)$ on the right-hand side of Eq. 4.1.

- *Prior.* The prior distribution $p(S)$ serves as a *regularizer* that imposes low probability masses on chemically unfavorable structures in the posterior distribution.

- *Backward prediction.* Bayes' law inverts the forward model $p(\mathbf{Y}|S)$ to the backward $p(S|\mathbf{Y} \in U)$ in which a desired property U is specified for the conditional. A Monte Carlo calculation is conducted to generate a random sample of molecules $\{S^r|r = 1, \ldots, R\}$ of size R according to the posterior distribution.

In this study, a chemical structure is described by a SMILES string. As will be detailed, a chemical language model defines the conditional distribution $S' \sim p(S'|S)$ to which the current structure S is randomly modified to a new S'. By the machine learning of the SMILES

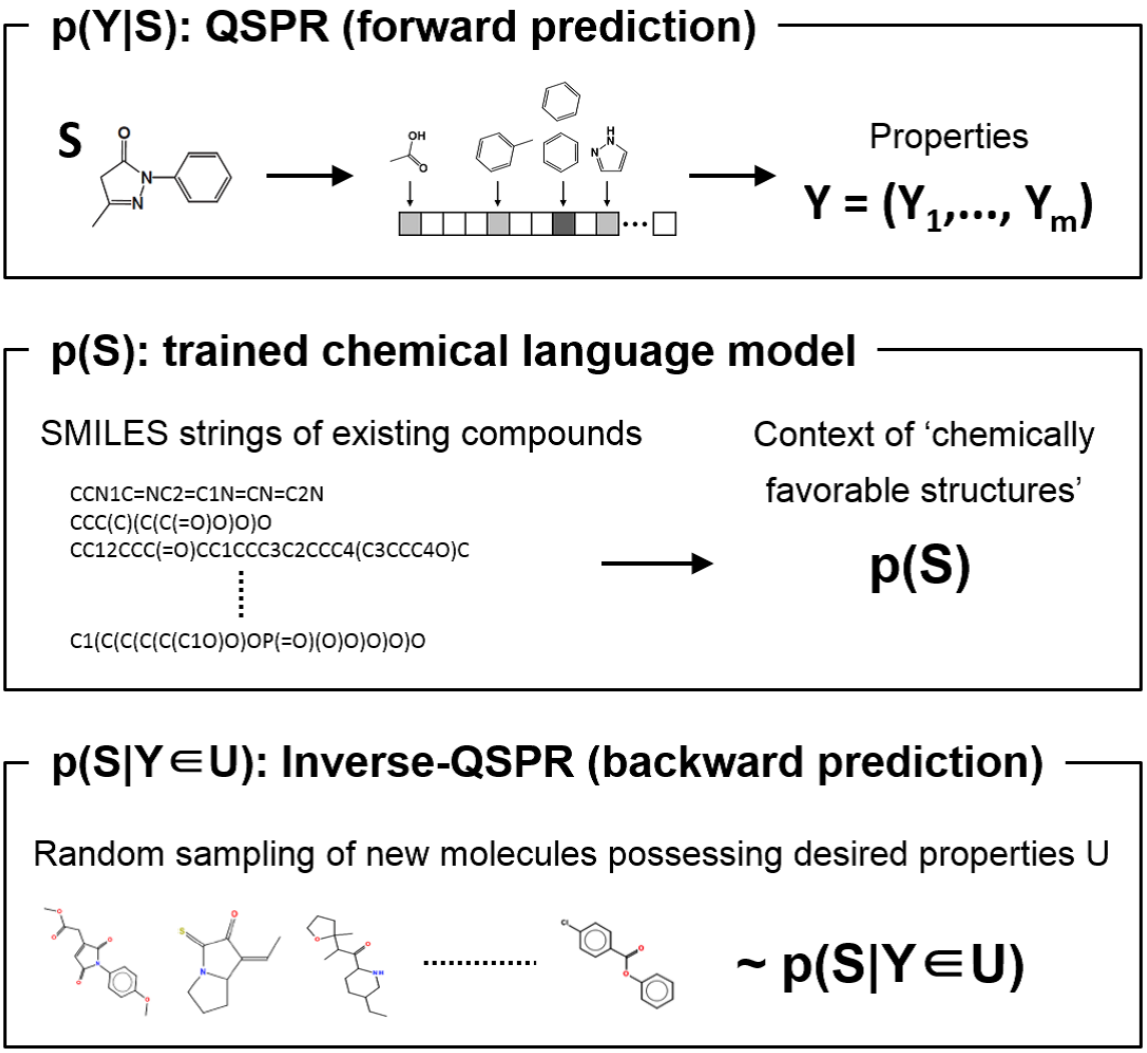


Fig. 4.1 Outline of the Bayesian molecular design method

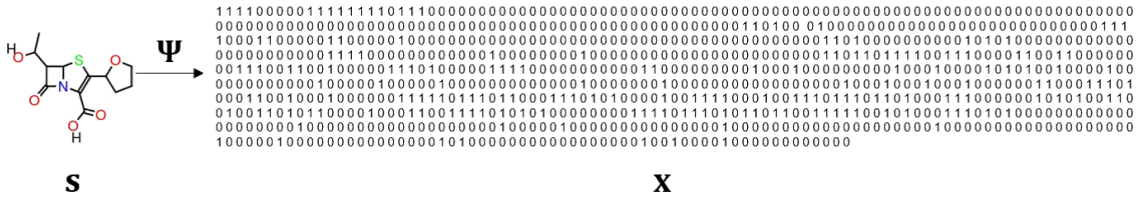


Fig. 4.2 The schematic description of the finger print.

language in tens of thousands of existing compounds, structural patterns of real molecules are compressed to the probabilistic language model. In combination with SMC, the trained model, which acquires the implicit meaning of *chemically unfavorable structures*, is utilized to modify SMILES strings under a given U while reducing the emergence of structures unlikely to occur. Furthermore, the trained language model serves as the prior in Eq. 4.1.

4.3 QSPR model

4.3.1 Bayesian linear regression

A data set $\mathscr{D}_j = \{Y_{ij}, S_i | i = 1, \ldots, N\}$ on property j is given where $Y_{ij} \in \mathbb{R}^1$ and S_i consist of the ith sample. With the N observations, the forward model is trained by a Bayesian linear regression $Y_j = \mathbf{w}_j^{\mathrm{T}} \boldsymbol{\psi}_j(S) + \varepsilon$ with a d-dimensional fingerprint descriptor $\boldsymbol{\psi}_j(S) \in \{0,1\}^d$ which contains 2D or 3D information of chemical structures as shown in Fig.4.2. To simplify the notation, the property index j is omitted here. The noise ε is independently and identically distributed according to the normal distribution $\mathrm{N}(\varepsilon|0,\sigma^2)$. The unknown parameters consist of the coefficient vector $\mathbf{w} \in \mathbb{R}^d$ and the noise variance $\sigma^2 \in \mathbb{R}^1_+$. Placing the normal and inverse-gamma priors to the unknowns, we derive the predictive distribution $p(Y|S,\mathscr{D})$ on the property Y with respect to an arbitrary input S.

Putting the normal prior $\mathbf{w} \sim \mathrm{N}(\mathbf{w}|\mathbf{0}, \mathbf{V}_0^{-1})$, and the inverse gamma prior $\sigma^2 \sim \mathrm{IG}(\sigma^2|a,b)$ on the unknown parameters of the linear regression model, we derive the predictive distribu-

tion on the property Y with respect to an arbitrary input S:

$$\begin{aligned} p(Y|S,\mathscr{D}) &= \mathrm{T}_{2a_*}\left(Y\middle|\mathbf{w}_*^{\mathrm{T}}\boldsymbol{\psi}(S),\frac{b_*}{a_*}(1+\boldsymbol{\psi}(S)^{\mathrm{T}}\mathbf{V}_*\boldsymbol{\psi}(S))\right), \\ \mathbf{V}_* &= (\mathbf{V}_0^{-1}+\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}, \\ \mathbf{w}_* &= \mathbf{V}_*\boldsymbol{\Psi}^{\mathrm{T}}\mathbf{y}, \\ a_* &= a+N/2, \text{ and} \\ b_* &= b+\frac{1}{2}(\mathbf{y}-\boldsymbol{\Psi}\mathbf{w}_*)^{\mathrm{T}}(\mathbf{I}-\boldsymbol{\Psi}\mathbf{V}_*\boldsymbol{\Psi}^{\mathrm{T}})^{-1}(\mathbf{y}-\boldsymbol{\Psi}\mathbf{w}_*), \end{aligned}$$

where $\boldsymbol{\Psi}^{\mathrm{T}}=(\boldsymbol{\psi}(S_1),\ldots,\boldsymbol{\psi}(S_N))$ and $\mathbf{y}^{\mathrm{T}}=(Y_1,\ldots,Y_N)$. Here, $\mathbf{I}$ denotes the identity matrix, and $\mathrm{T}_\nu(Y|\mu,\lambda)$ denotes the density function of the t-distribution with mean μ, scale λ and the degree of freedom ν. The predicted value of the property is given by the mean $\mathbf{w}_*^{\mathrm{T}}\boldsymbol{\psi}(S)$ of the predictive distribution.

The prediction models on the p properties, $p(Y_j|S,\mathscr{D}_j)$ $(j=1,\ldots,p)$, are obtained individually from the respective training sets. We then define the likelihood in Bayes' law with a desired property region $U=U_1\times\ldots\times U_p$ as

$$p(\mathbf{Y}\in U|S)=\prod_{j=1}^{p}\int_{U_j}p(Y_j|S,\mathscr{D}_j)\mathrm{d}Y_j. \tag{4.2}$$

For brevity, we write $p(\mathbf{Y}\in U|S)=p(\mathbf{Y}\in U|S,\mathscr{D})$.

Though a simple instance of QSPR models is described here, we can exploit more advanced techniques of supervised learning such as state-of-the art deep learning or a class of the ensemble learning algorithms. When dealing with a discrete-valued property, the regression should be replaced by a classification model such as the logistic regression model as shown in the next subsection. This study is developed along with the use of conventional fingerprints as the descriptor, but it is highly beneficial in practice to use more advanced descriptors, for example, molecular graph kernels coupled with kernel machine learning[103, 82, 132].

4.3.2 Logistic regression

Supposed that all the same type of dataset but the binary variables $Y_i \in \{0,1\}^d$ are obtained, the logistic regression is widely used to quantify the relationship between features of chemical structures and response variables in output as shown below.

$$p(Y=1|S) = \sigma(\mathbf{w}_*^{\mathrm{T}} \boldsymbol{\psi}(S)), \tag{4.3}$$

where $\sigma(\cdot)$ is the sigmoid function $\{1+\exp(\cdot)\}^{-1}$. Here, no prior is assumed for $\boldsymbol{w}$ and the estimation of $\boldsymbol{w}$ with the dataset $\mathscr{D}$ is done to maximize the following a cross entropy $J(\boldsymbol{w})$

$$\mathbf{w}_* = \underset{\mathbf{w}}{\operatorname{argmax}} J(\boldsymbol{w}) \tag{4.4}$$

$$= -\underset{\mathbf{w}}{\operatorname{argmax}} \sum_{i=1}^{N} \{y_i \log(\sigma(\mathbf{w}^{\mathrm{T}} \boldsymbol{\psi}(S_i))) + (1-y_i)\log(1-\sigma(\mathbf{w}^{\mathrm{T}} \boldsymbol{\psi}(S_i)))\} \tag{4.5}$$

This is the convex function with respect to $\mathbf{w}$ but analytically intractable to find the point to maximize J. This optimization is regularly achieved by the iteratively reweighted least squares (IRLS) [24] and Quasi-Newton methods such as BFGS [92].

4.4 Chemical language model

4.4.1 N-gram model

With the SMILES chemical language, a molecule is translated to a linearly arranged string $S = s_1 s_2 \ldots s_g$ of length g. A SMILES string consists entirely of symbols that indicate element types, bond types, and the start and terminal for ring closures and branching components. The start and terminal of a ring closure is designated by a common digit, '1', '2', and so on. A branch is enclosed in parentheses, '(' and ')'. Substrings corresponding to multiple rings and branches can be nested or overlapped. In addition to the formal rule, all strings are revised as ending up with the termination code '$'. Inclusion of this symbol is necessary to automatically terminate a recursive string elongation process. For instance, once a string

Table 4.1 Correspondence table between the formal and modified rules of SMILES

Type	Original	Modified
Start of a ring closure	`n` $\in \{1, 2, \ldots\}$	`&`
End of a ring closure	`n` (same to the start)	`&`$_i$ for the i-th ring terminators to the last of a string
Bond followed by atom A	`=A` (double), `#A` (triple)	`=A` or `#A` form a single character
Terminal character of a molecule	N/A	`$`
String in a square bracket	`[abcde]`	`[abcde]` form a single character

pattern `...CCC=O` is present, any further elongation is prohibited and should be terminated at once by appending '$'. In addition, digits indicating the starts and terminals of rings are represented by '&'. The revised representation rule is listed in Table 4.1 .

With no loss of generality, the prior $p(S)$ can be expressed as the product of the conditional probabilities:

$$p(S) = p(s_1)\prod_{i=2}^{p} p(s_i|s_{1:i-1}). \tag{4.6}$$

The occurrence probabiquivalent forms that correspond to different atom orderings. We treat such structurally equivalent strings as different S.

The fundamental idea of the chemical language modeling is as follows: (i) the conditional probability $p(s_i)$ depends on the preceding $s_{1:i-1} = s_1 \ldots s_{i-1}$. In general, the non-canonical SMILES encodes a chemical structure into many $p(s_i|s_{1:i-1})$ is estimated with the observed frequencies of substring patterns in known compounds, and (ii) the trained model is anticipated to successfully learn an implied context of the chemical language. For a given substructure $s_{1:i-1}$, the model is used to modify the rest of the components: until the termina-

tion code appears, subsequent characters are recursively added according to the conditional probabilities while putting the acquired chemical reality into the resulting structure.

The SMILES generator should create grammatically valid strings. In particular, we focus on two technical difficulties to be addressed, which are relevant to the rules of grammar on the expression of rings and branching components.

- Unclosed ring and branch indicators must be prohibited. For instance, any strings extended rightward from a given $s_{1:6} = \texttt{CC(C(C}$ should contain two closing characters, '`)`', somewhere in the rest.

- Neighbors in a chemical string are not always adjacent in the original molecular graph. Consider a structure expressed by `CCCCC(CCCCC)C`. The substring in the parentheses is a branch of the main chain. The main chain consists of six tandemly arranged carbons that are split into before and after the branch. In this case, the occurrence probability of the final character $s_{13} = \texttt{C}$ should be affected more by characters in the main chain than those in the branch. In other words, the conditional probability of s_i should depend selectively on a preferred subset of the conditional $s_{1:i-1}$ according to the overall context of $s_{1:i-1}$ and s_i. The same holds when one or more rings appear in the conditional, *e.g.*, `c1ccc2ccccc2c1C`.

To remedy these issues, the conditional probability in Eq. 4.6 is modeled as

$$p(s_i|s_{1:i-1}) = \prod_{k=1}^{20} p(s_i|\phi_{n-1}(s_{1:i-1}), \mathscr{A}_k)^{I(s_{1:i-1}\in\mathscr{A}_k)}, \tag{4.7}$$

where $I(\cdot)$ denotes the indicator function which takes value one if the argument is true and zero otherwise. One of the 20 different models $p(\cdot|\cdot, \mathscr{A}_k)$ $(k = 1, \ldots, 20)$ becomes active when the state of the preceding sequence $s_{1:i-1}$ falls into any of the mutually exclusive "conditions" $\mathscr{A}_k$ $(k = 1, \ldots, 20)$. The 20 $(= 2 \times 10)$ conditions are classified according to the presence or absence of unclosed branches and the numbers $\{0, 1, \ldots, 9\}$ of unclosed ring indicators in $s_{1:i-1}$. For instance, if $s_{1:i-1}$ contains two unclosed ring indicators, *e.g.*, `CCCC(CC(`, the corresponding models should be probabilistically biased toward producing the two terminal

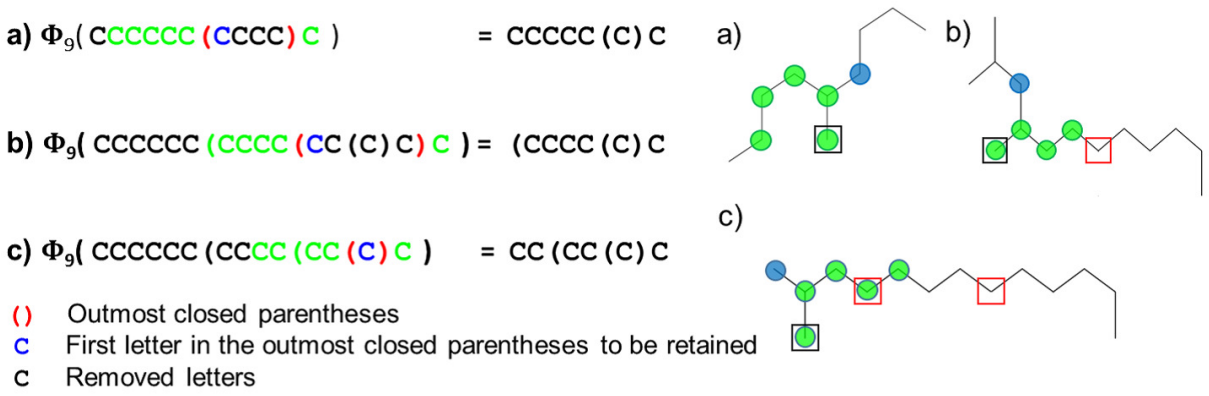


Fig. 4.3 Illustration of the substring selector $\phi_{n-1}(\cdot)$ with three examples. In the contraction operation, a substring inside of the outermost closed parentheses (red) is reduced to the character in its first position (blue). The extraction operation is to remove the rest (green) of the last $n-1$ ($=9$) characters from the reduced string. The corresponding graphs are shown on the right where the atoms in the boxes indicate the last characters in the inputs of $\phi_{n-1}(\cdot)$ (left).

characters '`)`' in subsequent characters. In addition, the substring selector $\phi_{n-1}(s_{1:i-1})$ is introduced for the treatment of the second problem. The definition is as follows:

- *Contraction.* Suppose that $s_{1:i-1}$ contains a substring $t = t_1 \dots t_q$ enclosed by the closed parentheses such that t itself is never enclosed by any other closed parentheses. In other words, t is a substring inside of the outermost closed parentheses. The substring is then reduced to be $t \to t' = t_1$ by removing all characters in t except for the first character, t_1. In other words, t_1 is the character that is the right-hand neighbor of the opening '`(`' of the outermost closed parentheses.

- *Extraction.* The selector $\phi_{n-1}(s_{1:i-1})$ outputs the last $n-1$ characters in the reduced string of $s_{1:i-1}$.

The substring selector is illustrated with several examples in Fig. 4.3. This operation reduces a substring in any nested closed parentheses to a single character that indicates the atom adjacent to the branching point. The occurrence probability of s_i is then conditioned by its $n-1$ preceding characters in the reduced strings that correspond to neighbors in the molecular graph.

Under the maximum likelihood principle, the conditional probability for $\mathscr{A}_k$ in Eq. 4.7 was estimated by the relative frequency of co-occurring n-gram, s_i and $\psi_{n-1}(s_{1:i-1})$, in training instances of known compounds as follows.

Let $f_{\mathscr{A}_k}(s_i, \phi_{n-1}(s_{1:i-1}))$ denote the count of the n-grams in which the conditional string $s_{1:i-1}$ is in condition $\mathscr{A}_k$. We then conduct the back-off procedure [17] separately with all possible substrings $s_{1:i}$ whose the conditionals $s_{1:i-1}$ belong to $\mathscr{A}_k$:

$$p(s_i|\phi_{n-1}(s_{1:i-1}), \mathscr{A}_k) = \begin{cases} \frac{f_{\mathscr{A}_k}(s_i, \phi_{n-1}(s_{1:i-1}))}{\sum_{s_i \in \Sigma} f_{\mathscr{A}_k}(s_i, \phi_{n-1}(s_{1:i-1}))} & \text{if} \sum_{s_i \in \Sigma} f_{\mathscr{A}_k}(s_i, \phi_{n-1}(s_{1:i-1})) > 0 \\ p(s_i|\phi_{n-2}(s_{1:i-1}), \mathscr{A}_k) & \text{otherwise} \end{cases},$$

where Σ denotes the set of all possible characters. This is a recursive formula across $n = 1, 2, \ldots, n_{\max}$. In the upper formula, the estimate is given by the relative frequency of each instance of an n-gram in the $\mathscr{A}_k$-conditioned substrings. If there are no instances, the estimate at the previous $(n-1)$-gram is substituted as in the lower formula.

To determine the order n of the chemical language model and to verify its learning ability in the chemical language context, ten training sets of 1,000 compounds were randomly produced from the PubChem compounds. Each set was halved for training $\mathscr{D}_{\text{train}}$ and testing $\mathscr{D}_{\text{test}}$.

The models with varying orders, $n \in \{4, 7, 10\}$, were trained with two different procedures, the back-off (BO) and the Kneaser-Nay smoothing (KN) methods [17]. As a control group in the comparison, we added a conventional n-gram that learned the $(n-1)$-order Markov relationship among the chemical strings simply without using the stratification $\mathscr{A}_k$ ($k = 1, \ldots, 20$) and the substring selector $\phi_{n-1}(\cdot)$. Model performances were evaluated with two criteria: the perplexity measure[64] and the grammatical validity of produced chemical strings.

Perplexity is a commonly used measure in the natural language processing that evaluates the generalization capability of a language model $\mathscr{M}$ with the trained probability function

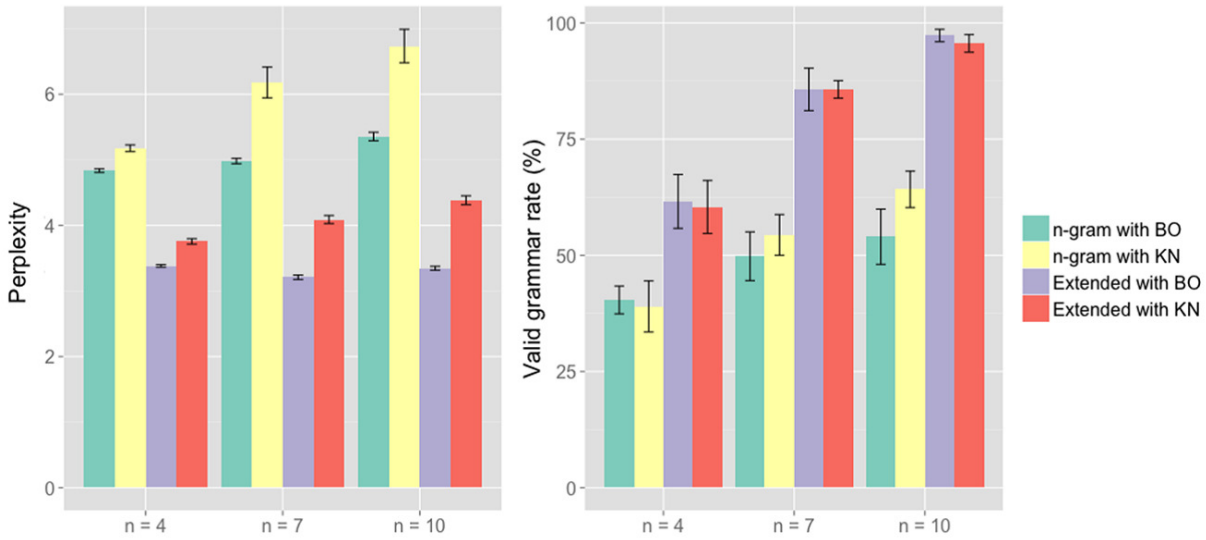


Fig. 4.4 Perplexity scores (left) and valid grammar rate (1 - the syntax error rate) (right) with respect to 1,000 SMILES strings generated from trained chemical language models. The conventional n-gram and the extended language models were trained with the BO and KN algorithms. The error bars represent the standard deviations across the 10 experiments corresponding to different training sets.

$p_{\mathscr{M}}(S)$ in Eq. 4.6,

$$\text{perplexity}(\mathscr{M}) = \exp\Big(-\frac{1}{|\mathscr{D}_{\text{test}}|}\sum_{i\in\mathscr{D}_{\text{test}}}\log p_{\mathscr{M}}(S_i)\Big).$$

For each model, the goodness-of-fit, *i.e.*, the likelihood, to the 1,000 test instances was measured. As shown in Fig. 4.4, the models resulting from BO outperformed the others in terms of perplexity. In the comparison among the BO-derived models with the different orders, there were no significant differences in the generalization capability. Furthermore, this experiment showed the significance of the stratification $\mathscr{A}_k$ ($k = 1,\ldots,20$) and the substring selector $\phi_{n-1}(\cdot)$, as significant improvements of perplexity were observed in the extended models relative to the conventional models.

In light of grammatical validity, the syntax error rates were evaluated for 1,000 hypothetical molecules generated from each of the ten trained models. The grammar check was done with the SMILES parser function 'parse.smiles' in the *rcdk* package with the option 'kekulise = TRUE'. As shown in Fig. 4.4, the error rate was monotonically reduced with an

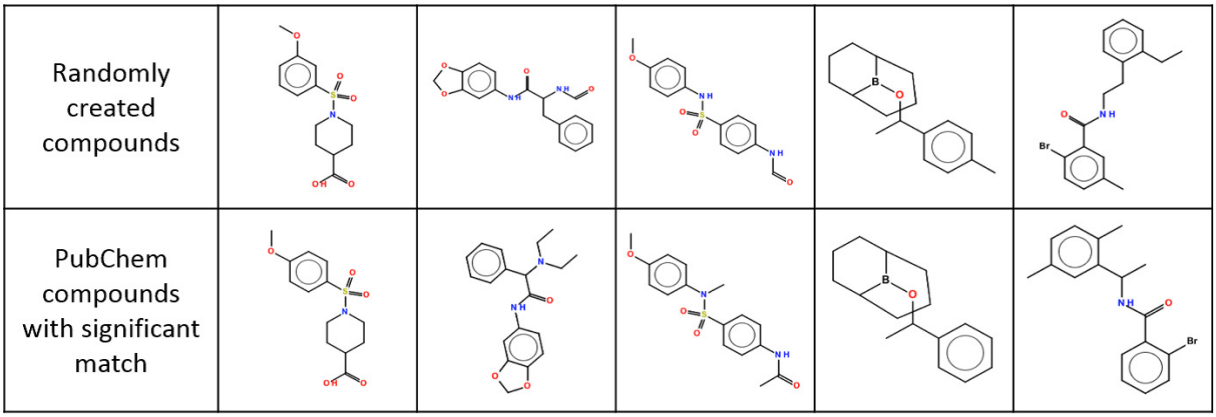


Fig. 4.5 Examples of molecules generated from the trained chemical language model with $n = 10$ (top). The bottom row displays the most similar PubChem compounds that had the Tanimoto coefficient ≥ 0.9 on the PubChem fingerprint.

increase in the Markov order in the extended models. The minimum error rate ($\leq 2.7\%$) was attained at $n = 10$. The performances of the BO and KN algorithms were much the same. In conclusion, we selected the BO-derived model with $n = 10$ on the basis of perplexity and grammatical validity.

To further validate the learning ability of the BO-derived model with $n = 10$, 50 randomly created molecules were associated with PubChem compounds in which the training compounds were removed. Approximately 72% of the 50 virtual molecules exhibited extensive similarities to one or more existing compounds meeting the acceptance criterion of the Tanimoto coefficient ≥ 0.9 on the PubChem fingerprint. Fig. 4.5 shows five instances of the created molecules; these instances indicate the great ability of the chemical language model. Conventional structure generators could never reproduce such structurally complex molecules.

4.5 Posterior inference using the sequential Monte Carlo sampler

The objective of the backward prediction is to generate chemical strings from the posterior distribution in Eq. 4.1, conditioned on a desired property region U. The forward models and the trained language model define the posterior as in Eq. 4.2 and Eq. 4.6.

We here use the SMC sampler to create novel molecules from the result in the Table 2.2 that the it produced a more accurate estimate than the simple Metropolis sampler as well as the RPMCMC. The pseudo code of the SMC algorithm that we developed is shown in Algorithm 1. In general, diverse molecules exhibit significantly high probabilities in the posterior. In order to better capture the diversity of promising structures, we create a series of tempered target distributions, $\gamma_t(S)$ ($t = 1, \ldots, T$), with a non-decreasing sequence of inverse temperatures $0 \leq \beta_1 \leq \beta_2 \leq \ldots \leq \beta_{s-1} \leq \beta_s = \ldots = \beta_T = 1$.

$$\gamma_t(S) \propto p(\mathbf{Y} \in U|S)^{\beta_t} p(S).$$

The likelihood function becomes flatter as the inverse temperature decreases, and vice versa. The algorithm begins with a small $\beta_1 \simeq 0$. The series of target distributions monotonically approaches as the iteration number increases, and bridges to the posterior at $\beta_t = 1, \forall t \geq s$.

The pseudo code of the SMC algorithm is shown in Algorithm 1. At the initial step $t = 0$, R structures $\{S_0^{(r)}|r = 1, \ldots, R\}$ are created. For each subsequent t, a currently obtained structure $S_{t-1}^{(r)}$ is transformed randomly to $S_*^{(r)}$ according to a structure manipulation model $G_\theta(S_{t-1}^{(r)}, S_*^{(r)})$ with a set of parameters, $\theta = (\kappa, \eta)$, as detailed below. This procedure is available simply by replacing $K(S_{t-1}^{(r)}, S_*^{(r)}) = G_\theta(S_{t-1}^{(r)}, S_*^{(r)})$, $L(S_*^{(r)}, S_{t-1}^{(r)}) = G_\theta(S_*^{(r)}, S_{t-1}^{(r)})$ in the sequential Monte Carlo sampler introduced in the subsection 4.2.3. The structure manipulation model $G_\theta(S, S')$ is designed with the trained SMILES generator as summarized below.

(i) Draw a uniform random number $z \sim \mathrm{Unif}(0, 1)$. If S is grammatically correct and z is less than the reordering execution probability κ ($= 0.2$), reorder the string $S \rightarrow$

S^* of length g, otherwise set the unprocessed string to S^*. With the first character chosen randomly using a uniform distribution, Open Babel 2.3.2 [94] is used from the command line with an argument '-xf' for the reordering.

(ii) Discard the rightmost b characters of the reordered string to derive $S^{**} = s^{**}_{1:g-b}$. The deletion length b is sampled from the binomial distribution $b \sim \mathrm{B}(b|L, 0.5)$ with binomial probability and the maximum length L ($= 5$ by default).

(iii) Extend the reduced string by sequentially adding a new character to the terminal point $L - b$ times. A newly added character follows the trained language model $s_i \sim p(s_i|s_{1:i-1})$. Once the termination code appears, the elongation is stopped, and then we have S'.

The reordering of strings plays a key role in preventing a series of designed molecules from getting stuck in local states. Note that temporally, the SMC algorithm can create structures containing unclosed rings and branching components. Then, the corresponding start codes for the unclosed rings or branches are temporally removed to avoid the syntax error when obtaining a descriptor for the likelihood calculation. In addition, the atom order is rearranged only when a current string is grammatically valid.

Algorithm 1 Backward prediction algorithm

Input $T, R, E, \theta, \{S_0^{(r)} | r = 1, \ldots, R\}$, $\{\beta_t | t = 1, \ldots, T\}$

Output $\{S_t^{(r)} | r = 1, \ldots, R, t = 1, \ldots, T\}$

Set $t = 0$, and $w_t^{(r)} = 1/R$ for $r = 1, \ldots, R$.

for $t = 1, \ldots, T$ **do**

for $r = 1, \ldots, R$ **do**

Transform $S_{t-1}^{(r)}$ to an intermediate state $S_*^{(r)}$ using the structure manipulation model $G_\theta(S_{t-1}^{(r)}, S_*^{(r)})$ (the procedure is detailed in the main body of this chapter).

Update the weight of the rth structure as

$$w_t^r = w_{t-1}^{(r)} \frac{\gamma_t(S_*^{(r)})}{\gamma_t(S_{t-1}^{(r)})}$$

Note that if the modified $S_*^{(r)}$ contains unclosed ring or branch specifications, those indicators must be temporally removed before the conversion of the chemical string to a descriptor in the likelihood calculation.

end for

The weights are normalized to obtain the selection probabilities $W_t^{(r)} \propto w_t^{(r)}$ such that $\sum_r W_t^{(r)} = 1$.

Calculate the effective sample size $\mathrm{ESS} = R(\sum_r {W_t^{(r)}}^2)^{-1}$.

if $\mathrm{ESS} \leq E$ **then**

Perform the resampling of $\{S_*^{(r)} | r = 1, \ldots, R\}$ with the selection probabilities.

Set $w_t^{(r)} = 1/R$ for $r = 1, \ldots, R$.

The resampled set forms a new population $\{S_t^{(r)} | r = 1, \ldots, R\}$.

else

$S_t^{(r)} = S_*^{(r)}$ for $r = 1, \ldots, R$.

end if

end for

4.6 Applications

4.6.1 Physical properties

Dataset

The molecular design process is demonstrated through the creation of small organic molecules with the design objective intended to the HOMO-LUMO gap (HL) and the internal energy (E). With the quantum chemistry calculation based on DFT, the two properties were obtained for 16,674 compounds which were selected randomly from PubChem [69]. The 16,674 pairs of structures and properties are used for learning of forward model. In addition, randomly chosen 50,000 structures from pubchem were used for learning chemical structures with the chemical language model. Before learning, molecules including one or more inorganic or ionic elements were excluded.

Estimation

Before going to backward prediction, a forward model has to be constructed. In the process to determine a combination of fingerprints, the entire data set was divided into 10,000 and 6,674 items for training and testing, respectively. As shown in Table 4.2, eight different descriptors $\psi(S)$ were derived by using six types of molecular fingerprints in combination; which these fingerprints are implemented in the R package *rcdk* [48]. The mean of the predictive distribution was employed as the predicted value of each property. The parameters of the normal and gamma priors in regression were set as in Table 4.3. The performance of the trained models was assessed with the mean absolute error (MAE). As shown in Table 4.2, the augmented descriptor that combined the 'standard', 'extended', 'circular' and 'pubchem' fingerprints delivered the highest predictive accuracy. However, the average runtime for the likelihood calculation per 100 molecules (~ 7.71 sec) was significantly greater than the others because the translation into the 'pubchem' fingerprint involves an intractable graph pattern matching. This led to a significant increase in the runtime for the backward prediction. We therefore employed the second-best descriptor containing 'standard', 'extended' and 'circular', which delivered relatively small MAEs, 0.54 eV and 23.5 kcal/mol, for the HOMO-

Table 4.2 MAEs of the QSPR models with the eight different fingerprint descriptors for the internal energy and the HOMO-LUMO gap. The six fingerprints in the *rcdk* package (bottom) and their combinations were tested. The last column denotes the average runtime for the QSPR score (likelihood) calculation per 100 molecules, which run on an Intel Xeon 2.0GHz processor with 128GB memory using the *iqspr* package

Fingerprint	Energy (kcal/mol)	HOMO-LUMO gap (eV)	Runtime (sec)
1	32.6	0.53	0.50
2	30.4	0.54	0.41
3	29.3	1.37	2.57
4	28.3	1.66	0.36
5	22.1	0.55	5.32
6	46.8	0.84	0.39
1,2,4	23.5	0.54	1.61
1,2,4,5	18.9	0.50	7.71

1. 'standard': paths of a default length (1024 bits)
2. 'extended': the 'standard' fingerprint is modified such that ring and atomic properties are taken into account (1024 bits)
3. 'maccs': MDL MACCS keys (166 bits)
4. 'circular': ECFP6 finterprint (1024 bits)
5. 'pubchem': PubChem fingerprint (881 bits)
6. 'graph': 'standard' is modified by taking into account connectivity (1024 bit)

LUMO gap and internal energy, respectively. With this, the runtime was reduced by nearly 80% (to $\sim$1.61 sec per 100 molecules), compared with the best performing model.

The performance of the backward prediction was tested on three different property regions of $\mathbf{Y} = (Y_{\mathrm{HL}}, Y_{\mathrm{E}})^{\mathrm{T}}$: (i) $U_1 = [100,200] \times [4,5.5]$, (ii) $U_2 = [250,400] \times [5,6]$, and (iii) $U_3 = [100,250] \times [2.5,3.5]$. Table 4.3 summarizes the parameters of the backward prediction. Phenol '`c1ccccc1O`' was assigned to the 100 initial structures ($R = 100$) which were refined across $t = 1,\ldots,T$ with $T = 500$ as a desired property region was sought. Fig. 4.6 shows snapshots of these processes. The created molecules underwent substantial changes in size, geometry and composition. A visual inspection of the movies verifies that backward calculation prevents structures from getting stuck in locally high-probability regions (data not shown).

Fig. 4.7 illustrates the early stages ($t \in \{1,20,50,200\}$) of the property refinements, during which they are moving in toward their respective target regions. For each t, a non-

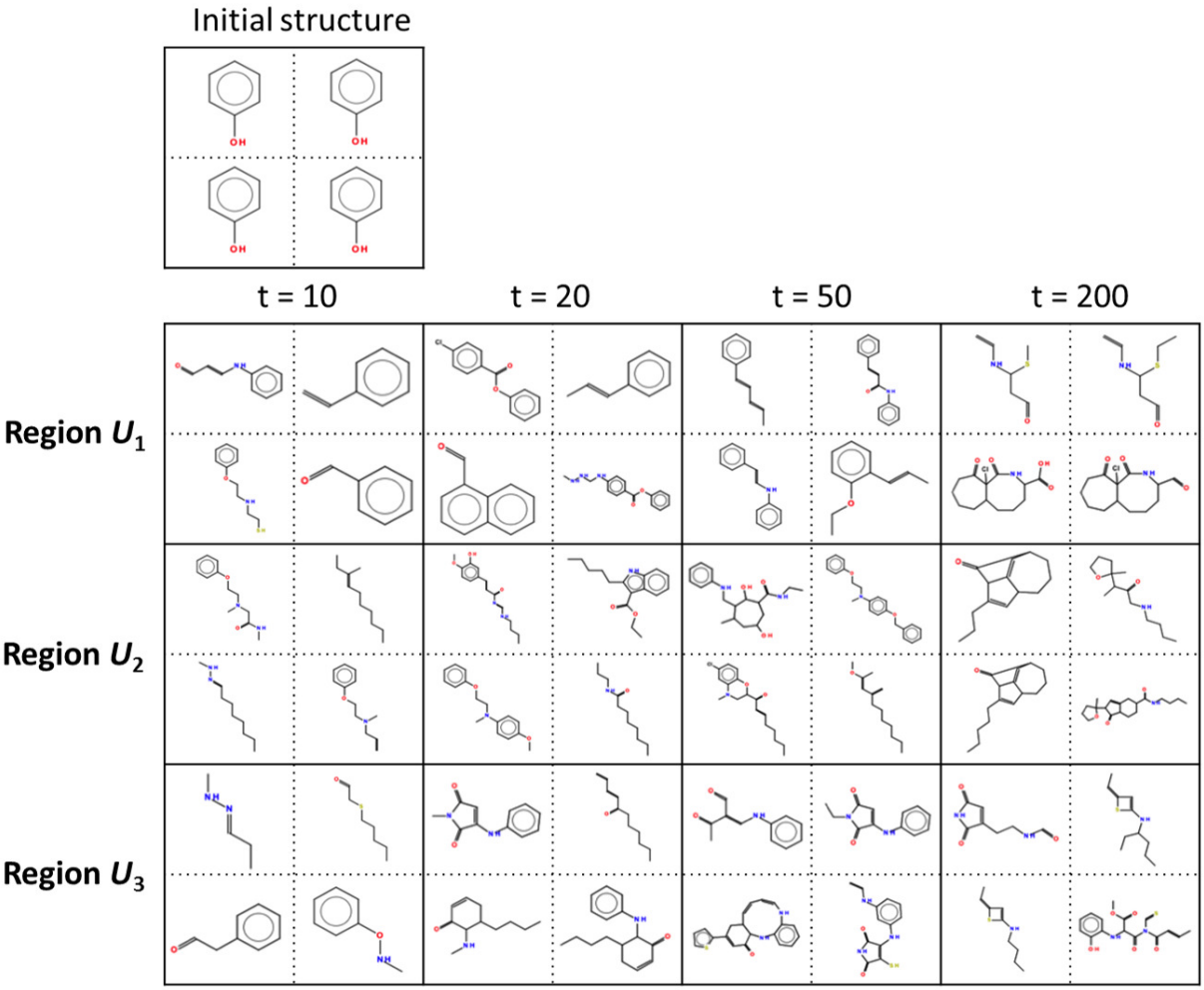


Fig. 4.6 Snapshots of structure alteration during the early phase of the inverse-QSPR calculation ($t \in \{10, 20, 50, 200\}$) with the desired property region set to U_1, U_2 or U_3. The initial molecule (phenol) is shown at the top. The created molecules shown here were those ranked in the top four by the likelihood score at each t.

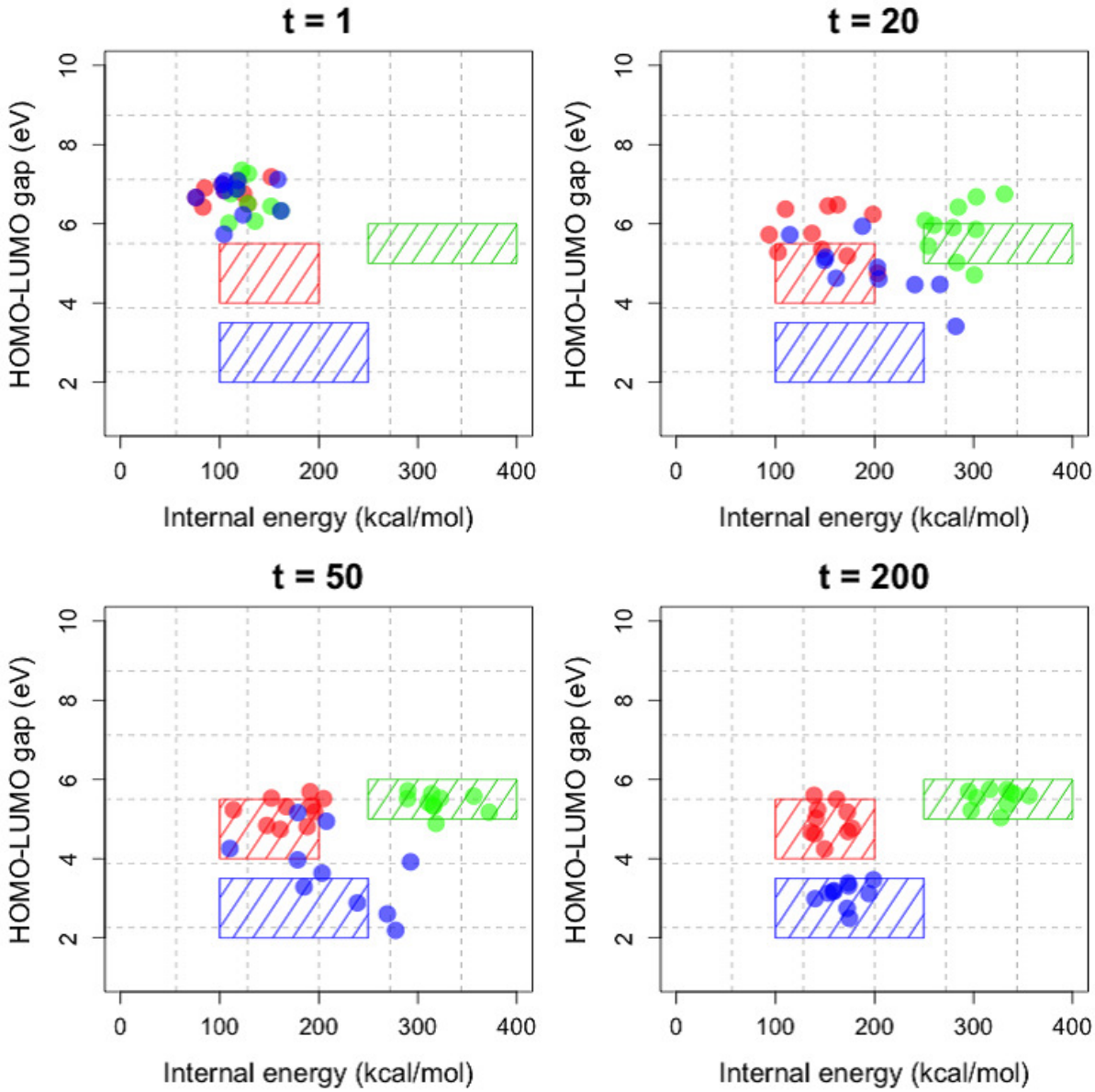


Fig. 4.7 Property refinements resulting from the backward prediction at $t \in \{1, 20, 50, 200\}$. Results on the three different property regions, U_1, U_2 and U_3, are displayed together, and color-coded by red, green and blue, respectively. The shaded rectangles indicate the target regions. The dots indicate the HOMO-LUMO gaps and internal energies of the designed molecules that were calculated by the predicted values of the QSPR models. For each U_i and t, the 10 non-redundant molecules exhibiting the greater likelihoods are shown.

Table 4.3 Parameters and experimental conditions for the backward prediction

Process	Description	Parameter
Forward prediction	Number of training data	$N = 10,000$
	Fingerprint descriptor	1, 2, 4
	The normal prior	$\mathbf{V} = \mathbf{I}$
	The Gamma prior	$(a,b) = (0,0)$
Chemical language model	Number of training data	50,000
	Markov-order	$n = 10$
	Estimation algorithm	Back-off method
Backward prediction	Size of population	$R = 100$
	Number of iterations	$T = 500$
	Reordering probability	$\kappa = 0.2$
	Total changing length	$L = 0.5$
	Cooling schedule	$\beta_t = 5^{0.95^{t-1}}$ for $t \leq 250$, $\beta_t = 1$ for $t \geq 251$
	Threshold on ESS	$E = 50$
	Initial structures	phenol `c1ccccc1O`

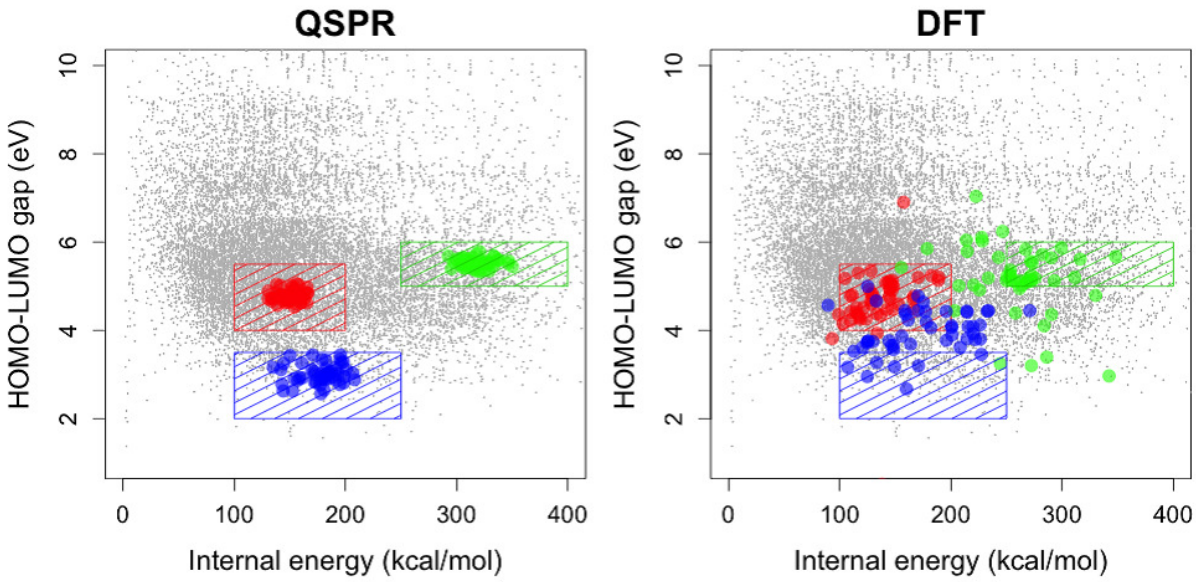


Fig. 4.8 Properties of 50 molecules which were selected from the overall backward prediction process for U_1 (red), U_2 (green), and U_3 (blue). The HOMO-LUMO gap and internal energy were calculated by the trained QSPR models (left) and the DFT calculation (right). The gray dots indicate the training data points. In each U_i, the 50 non-redundant molecules that achieved the highest likelihoods are shown.

redundant set of created molecules is shown: molecules ranked in the top 10 by the likelihood score were selected from a ranking list in which a molecule was removed from the list if its Tanimoto coefficient on the PubChem fingerprint exceeded 0.9 with respect to any of the higher ranking molecules. The reported HOMO-LUMO gap and internal energy correspond to the means of the predictive distributions for the trained forward models. At $t = 1$, the properties were very far from the desired regions. As the calculation proceeds, the resulting properties approached the targets quite rapidly. At $t = 200$, almost all of the created molecules have properties falling within their respective target region, U_1, U_2, or U_3. This observation indicates that the proposed method is capable of drastic and rapid refinements of the properties of seed molecules.

Fig. 4.8 shows the properties of molecules created at $t = 251$ and 500 with their verifications by the DFT calculation. In the same way described above, 50 non-redundant molecules were selected from the likelihood-based prioritized list of 25,000 candidates: similar to the results shown in Fig. 4.7, 50 non-redundant molecules were selected, in this case selected from a prioritized list of the 25,000 candidates corresponding to the 100 particles produced between $t = 251$ and 500. The physical properties were evaluated by the QSPR models (left) and the DFT calculation (right). For the DFT calculation, the created SMILES strings were first converted into the 3D structures by using OpenBabel with the '-gen3d' option. Such initial conformations were fully optimized using Gaussian09 with B3LYP/6-31+G(d). Finally, the electronic properties at the equilibrium geometries were computed at the same level of theory. As shown, all the QSPR-derived properties of the created molecules fell within the respective desired regions. However, in the verification by the DFT calculation, the arrival rates for U_2 and U_3 were significantly reduced to $25/50$ and $7/50$, while the high rate ($45/50$) was maintained on U_1. The cause of the performance depression in the former cases is apparent. As shown in Fig. 4.8, the number of known compounds used for the training was fairly small in neighborhoods of U_2 and U_3. By necessity, the trained forward models had much lower accuracies in prediction in neighborhoods of U_2 and U_3 relative to U_1. The ability of the backward prediction therefore declined as the desired properties were placed within regions where data are more sparsely populated. The proposed method has a great

ability to discover molecules when a desired property lies within a region where enough data are given, but the creation of truly novel molecules that reside in a far tail of the distribution of known molecules is an issue yet to be addressed. This will be discussed more in later section.

The novelty of derived molecules was investigated by seeking structurally similar compounds in PubChem. For a created S that appeared in U_i in terms of DFT, we calculated the Tanimoto coefficient $T(S,S^*)$ on the PubChem fingerprint with respect to all PubChem compounds S^* after removing the training instances. Under the acceptance criterion $T(S,S^*) \geq 0.9$, significantly similar known compounds were identified for S. Fig. 4.9 illustrates an instance of promising hypothetical molecules and the results of the similarity search. Thus, it has been confirmed that the proposed method is capable of reproducing the highly complex and diverse molecules in the database. As expected, molecules that emerged in U_2 and U_3 were less well matched to existing compounds. More importantly, it has been proved that various types of molecules can exist in the same property region and that many of these have yet to be identified. In practice in science and industry, such molecules could be truly important candidates for further testing and synthesis.

The backward prediction algorithm was run on an Intel Xeon 2.0GHz processor with 128GB memory using the *iqspr* package. The average execution time was about five seconds per step in SMC. The essential part of the current implementation was all developed in the R language and does not support parallel processing. The development of more advanced software is a future subject.

4.6.2 Bioactivity

Dataset

Datasets for structure-bioactivity relationship were obtained from the pubchem BioAssay database [124]. This database has 500000 types of assay protocols with 5000 target proteins, 30000 target genes and over 130 million bioactivity outcomes. The 10 bioactivities are selected with keyword "qHTS". The list of targets is shown in Table 4.4. Since these dataset

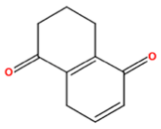
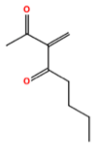
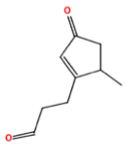
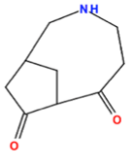
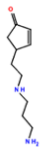
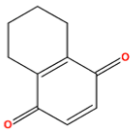
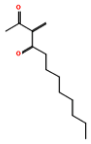
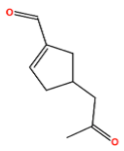
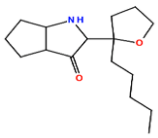
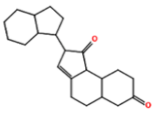
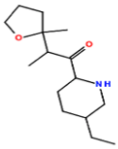
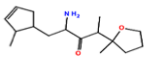
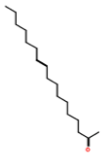
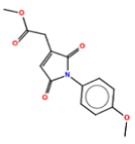
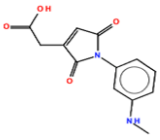
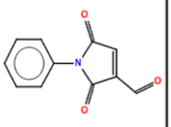
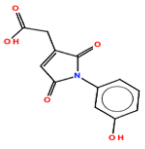
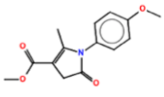
Region U_1	Discovered compounds					
	PubChem compounds with significant match				No match	No match
Region U_2	Discovered compounds					
	PubChem compounds with significant match		No match	No match	No match	No match
Region U_3	Discovered compounds					
	PubChem compounds with significant match		No match	No match	No match	No match

Fig. 4.9 Newly created molecules in the predefined property regions. The bottom row of each pair shows instances of significantly similar PubChem compounds that had the Tanimoto index ≥ 0.9.

Table 4.4 QSAR bioaasay data

Index	PubChem AID	Target name	Target activity
1	540303	Cell surface uPA generation	Active
2	588579	Polymerase Kappa	Inactive
3	588590	Polymerase Iota	Inactive
4	588591	Polymerase Eta	Inactive
5	588855	TGF-b	Inactive
6	651635	ATXN expression	Inactive
7	651768	WRN Helicase	Inactive
8	652105	PI5P4K	Inactive
9	686978	TDF1	Inactive
10	720504	PLK1-PDB	Inactive

usually contains much more negative data than positive data, positive and negative data are chosen randomly to be equal in number up to 2000, respectively.

The dataset for learning the 1922 chemical structures was obtained from the drugbank [128]. These are SMILES strings of FDA approved small molecule drugs. By learning from it, it will be expected to generate drug-like chemical structures.

Estimation

The QSPR model we used here was the logistic regression with elastic net regularization as shown in [134]. A set of the parameters used in this experiment was shown in Table 4.4. The purpose of learning QSPR is to confirm how accurate QSPR model predicts if an input molecule has the target bioactivities, which directly connects to the confidence that newly generated structures in the inverse-QSPR model have actually target bio-activity since the inverse-QSPR model is represented as the product of the QSPR model and the structure model. Fig.4.10 shows the prediction accuracy of the QSPR model using the true positive rate (TPR) and the true negative rate (TNR) with two decision boundary ($m = 0.5, 0.9$) for the target 1 and the rest of targets, respectively. The actual generated chemical structures from the inverse-QSPR model are obtained with strong confidence (high predictive probability for the target activity) in usual cases. From this result, the accuracy of QSPR model with high confidence ($m = 0.9$) was judged to be good enough (at least above 0.7) for checking in the backward prediction.

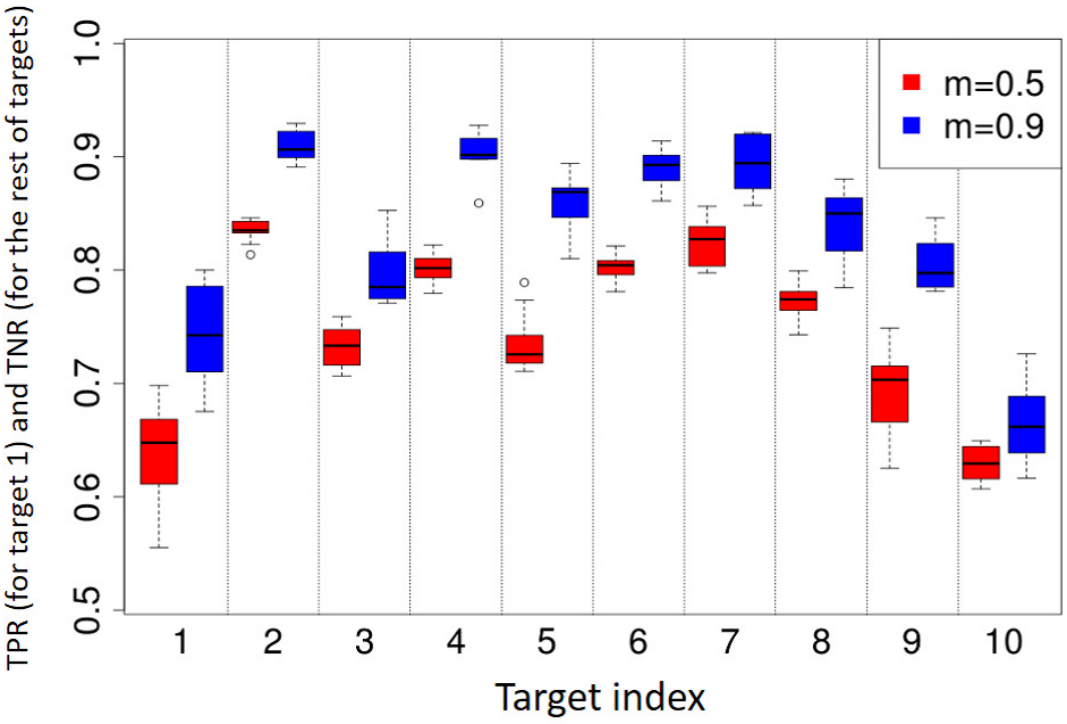


Fig. 4.10 The result of QSAR predictions for 10 targets of bioactivities.

Table 4.5 Parameters and experimental conditions for the backward prediction

Process	Description	Parameter
Forward prediction	Maximum number of training data	$N = 2,000$
	Fingerprint descriptor	1, 2, 4
	$L1$ regularization parameter	$\lambda = 0.01$
	Elastic net parameter	$\alpha = 0.1$
Chemical language model	Number of training data	1,922
	Markov-order	$n = 10$
	Estimation algorithm	Back-off method
Backward prediction	Size of population	$R = 1000$
	Number of iterations	$T = 100$
	Reordering probability	$\kappa = 0.2$
	Total changing length	$L = 5$
	Cooling schedule	$\beta_t = 20^{0.9^{t-1}}$ for $t \leq 80$, $\beta_t = 1$ for $t \geq 81$
	Threshold on ESS	$E = 50$
	Initial structures	phenol `c1ccccc1O`

For the inverse-QSPR, the score function $q(s)$ was defined as

$$q(S) = \sigma(\mathbf{w}_{1*}^{\mathrm{T}}\boldsymbol{\Psi}(S))^{r_1} \prod_{j=2}^{10} \sigma(\mathbf{w}_{j*}^{\mathrm{T}}\boldsymbol{\Psi}(S))^{r_j}, \tag{4.8}$$

where $\mathbf{w}_{j*}$ is the weight of the predictive model for activity j and $\mathbf{r} = (r_1, \ldots, r_{10})$ is the parameter to control the balance between active targets and inactive targets since inactive tend to be selected more than active to avoid the side effect caused by attacking unnecessary targets in usual settings. Here, $(r_1, \ldots, r_{10})$ is set as $(5, 5/18, 5/18, \ldots, 5/18)$.

As done for the physical properties, the SMC sampler was employed to generate chemical structures from the inverse-QSPR model. The parameter setting is shown in Table 4.6. Fig.4.11 shows the transition of score distributions. At the beginning in the SMC sampler, since all molecules are similar to the phenol, scores are almost close to zero. As time passed, their structures were gradually changed to preferred ones as shown in the previous experiment, resulting in obtaining relatively high scores until $t = 100$ in this experiment. The chemical structures in Fig.4.12 are some of the highest scored molecules with the score function (eqn. 4.8). QSPR prediction of these chemical structures are shown high activity for

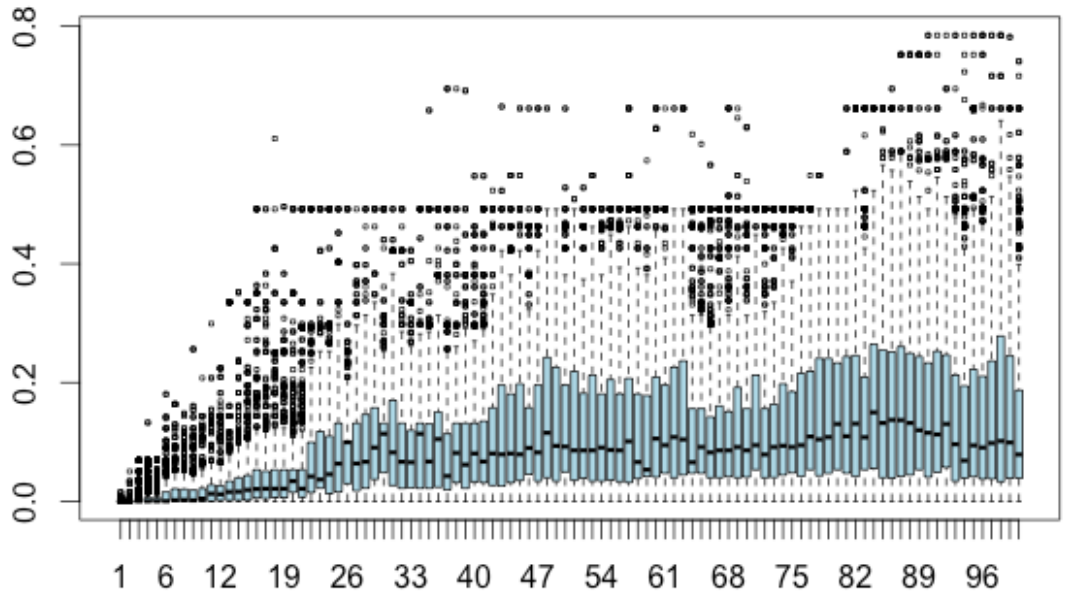


Fig. 4.11 Score distribution of particles of the SMC sampler at each time

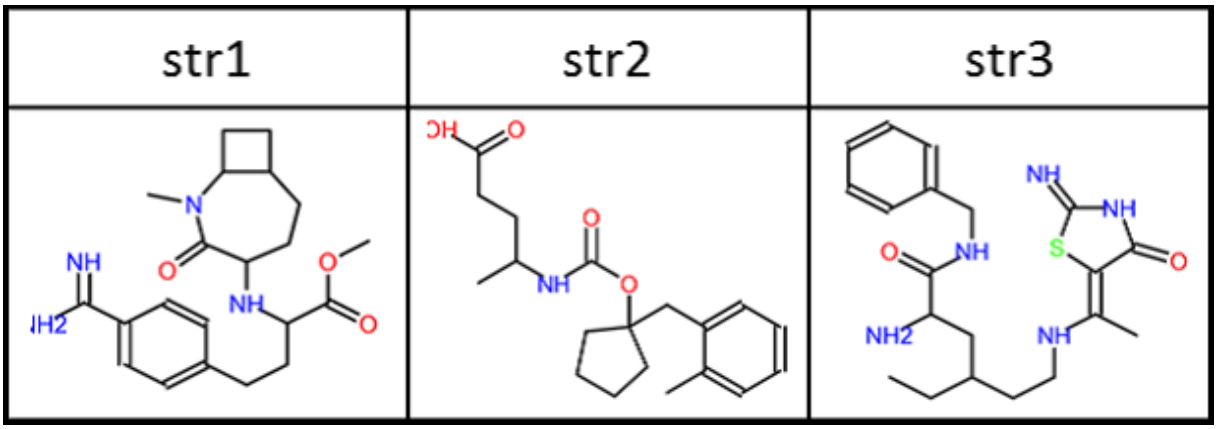


Fig. 4.12 generated three chemical structures with the highest scores

target 1 and low activities for the rest of targets (Table 4.5), thus this attempts can be said to successfully generate possible drug candidates.

Table 4.6 QSAR predictions of chosen 3 chemical structures for bioactivities in 10 target proteins

Index	score	1	2	3	4	5	6	7	8	9	10
str1	0.816	0.983	0.00	0.02	0.03	0.01	0.06	0.00	0.00	0.00	0.08
str2	0.785	0.990	0.02	0.08	0.04	0.07	0.03	0.00	0.04	0.04	0.02
str3	0.751	0.985	0.00	0.01	0.01	0.01	0.17	0.00	0.01	0.10	0.04

4.7 Conclusion

This study developed a principled approach to computational molecular design thorough a unified perspective of the Bayesian analysis to the forward and backward predictions. The method was demonstrated with case studies in multi-objective molecular design aimed at the physical properties (HOMO-LUMO gap and internal energy) and bio-activities for 10 target proteins. The presented analyses can be performed with the R package *iqspr* that we developed. Despite potentially great impacts on science and industry, practical applications of computer-aided molecular design methods have not been widely adopted. The lack of easy-to-access software and benchmark data has restrained the proliferation of the use of inverse-QSPR and the growth of methodologies and tools has been hampered due to the difficulty of performance competition.

The main contribution of this study lies in the newly proposed structure refinement algorithm based on the chemical language model. As mentioned earlier, most existing methods utilize chemical fragments of real compounds for the reduction of creating chemically unfavorable molecular graphs. The drawback of the fragment-based methods is the limited diversity of the created structures. To enhance diversity and novelty, a vast amount of fragments should be used, but this makes the operations for structure transformations in the fragment exchange process and similarity search on the large fragment library much more computationally expensive. The present study showed the great promise of a fragment-free strategy based on a chemical language model. The trained model acquired the implicit meaning of chemically favorable structures and succeeded in the creation of seemingly realistic molecules. Surprisingly, more than 70% of the generated molecules had significantly similar known compounds, and in addition, some of these were structurally very complex to the point that no conventional structure creators would ever be able to reproduce them. The proposed method demonstrated a new way to make computationally efficient structure refinements based on the string representation of molecules. It is important to see that the acquired context of the chemical language is not well defined, but rather is ambiguous. Possibly, the trained language model did not recognize higher-level chemical knowledge such as chemical

stability, synthesizability, and drug-likeliness. The creation of much more realistic and valid structures is an important consideration in future work.

As demonstrated, the backward method is enormously powerful in the exploration when enough data are observed in a neighborhood of a specified property region. However, the prediction ability declines as the desired properties are placed around regions where data are more sparse. The ultimate goal of computational molecular design is the creation of truly novel molecules that reside in an exceedingly far tail of the distribution of known molecules. The apparent cause of the limited ability is that the trained forward models become less accurate in property prediction in far tails of the training set. This is an issue common to all existing methods but less attention has been paid to this important problem. Ultimately, we wish to arrive in yet-unexplored property regions where no one has gone before. In Fig.4.13, we have provided snapshots of the property refinement process that explored a yet-unexplored property region, in order to emphasize the significance of overcoming this limitation. Within early steps, the resulting properties approached the desired region quite rapidly, but the search trajectories became more disperse as they got closer to the target.

A promising solution to this problem might be the integration of computer experiments and the backward prediction algorithm with experimental design techniques. Once created molecules get fairly close to an unexplored property region, a new set of structure-property data could be produced in a neighborhood of the region by conducting, for instance, a first-principle or a molecular dynamics computation with respect to a preferred subset of the currently created structures. Then, one could refine the forward models using the newly added data. Possibly, the query points of the computer experiment should rationally be selected under a sequential design strategy by maximizing the expected improvement of prediction under a given constraint of computational costs. The refined backward prediction might acquire a greater ability to move a step closer to the target region. The integration of the backward prediction algorithm and rationally designed adaptive data production is the next challenge in future work.

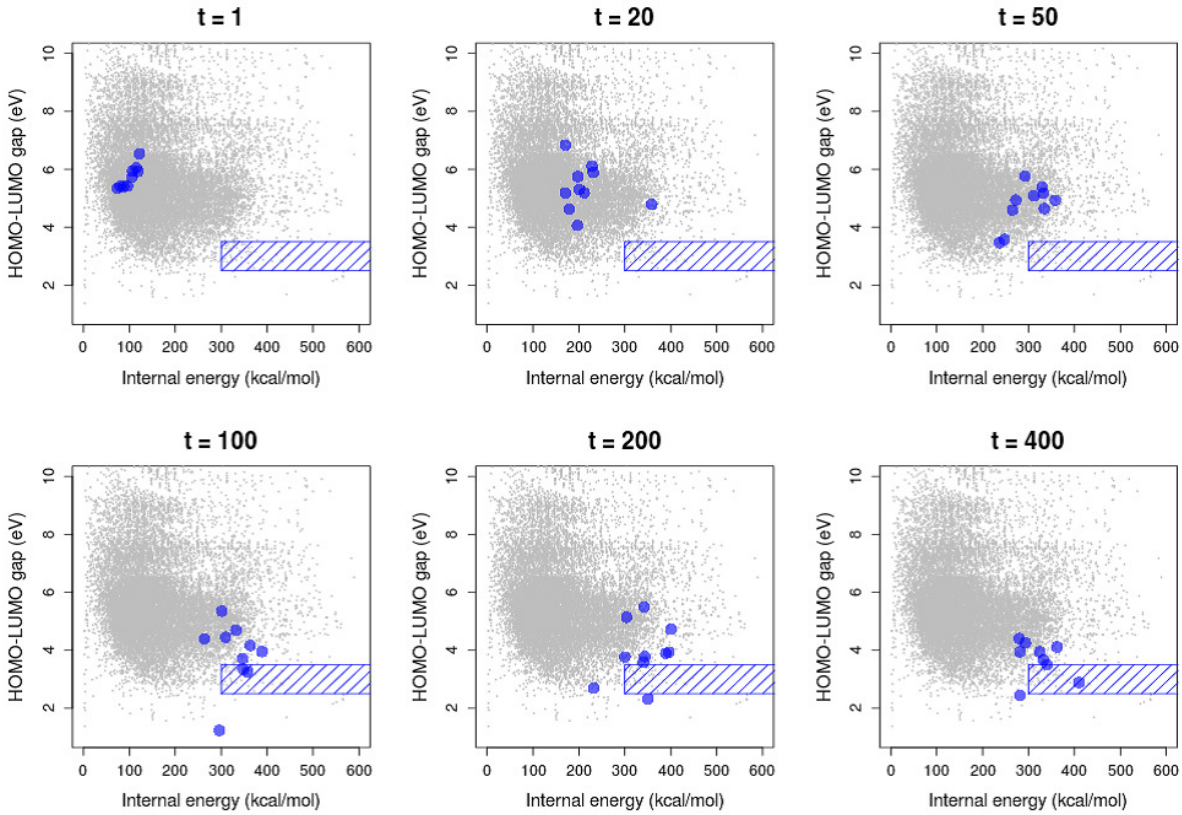


Fig. 4.13 As done in Fig. 4.7 and Fig. 4.8, properties in 10 chemical structures with highest scores at each time-step were computed with the Gaussian09.

4.8 R Package

The *iqspr* package can be installed thorough the CRAN repository. With this package, the molecular design process shown below can be reproduced. Installation of Open Babel 2.3.2 [94] is required for getting started. The package consists of a set of functions for the building of the forward prediction model (*QSPRpred reference class*) with molecular fingerprints implemented in the *rcdk* package [48], the backward prediction (*SmcChem reference class*), and the training and simulation of the SMILES generator (*ENgram reference class*) with user-specified input SMILES strings. This example is for generating chemical structures from the inverse-QSPR model with the structure - physical properties dataset.

Usage of *iqspr* package

```
install.packages("iqspr")
library(iqspr)

#obtain training data for QSPR
data(qspr.data)
idx <- sample(nrow(qspr.data), 5000)
smis <- paste(qspr.data[idx,1])
y <- qspr.data[idx,c(2,5)]

#learn a pattern of chemical strings
data(trainedSMI)
my_engram <- ENgram$new(trainedSMI, order = 10)

#learn QSPR model
my_qspr <- QSPRpred$new(smis = smis, y = as.matrix(y), v_fpnames = "graph")

#set target properties
qsprpred_EG_5k$ymin <- c(200, 1.5)
qsprpred_EG_5k$ymax <- c(350, 2.5)

#get chemical strings from the Inverse-QSPR model
smchem <- SmcChem$new(smis = rep("c1ccccc1O", 25), v_qsprpred = my_qspr,
v_engram = my_engram, temp = 3 ,decay = 0.95)
smchem$smcexec(niter = 100, preorder = 0.2, nview = 4)

#check
smiles <- get_smiles(smchem)
predict(my_qspr, smiles[1:100])
```

Chapter 5

Conclusion

The Bayesian methods presented here, combined with Monte Carlo sampling, are so powerful that they can be used to discover new structures through the Bayesian model. This thesis has mainly considered the application of these methods to problems in bioinformatics and cheminformatics.

In Chapter 3, we presented a new motif discovery method using our proposed algorithm, RPMCMC, which is a parallel variant of the widely-used Gibbs motif samplers. Compared to the standard Gibbs sampler, this all-at-once interacting parallel run of RPMCMC could detect a more diverse range of motifs. In addition, this algorithm was comprehensively tested on synthetic promoter sequences and real ChIP-seq datasets. In the synthetic promoter analysis, the RPMCMC algorithm found around 1.5 times as many embedded motifs as the existing methods. For the ChIP-seq datasets, the RPMCMC algorithm reported far more reliable cofactors than the recently published ChIP-tailored algorithms.

In Chapter 4, we developed a principled approach to computational molecular design, through a unified perspective of the Bayesian analysis, for the forward and backward predictions. The method was demonstrated with case studies in multi-objective molecular design aimed at determining the physical properties (HOMO-LUMO gap and internal energy) and bio-activities for 10 target proteins. The presented analyses can be performed with the R package *iqspr* that we developed. Despite potentially great impacts on science and industry, practical applications of computer-aided molecular design methods have not been widely adopted. The lack of easy-to-

access software and benchmark data has restricted the use of inverse-QSPR and the growth of methodologies and tools has been hampered due to the difficulty of performance comparison.

As shown in the latter part of Chapter 2, the RPMCMC and the SMC methods can correctly obtain samples from a target distribution even if the distribution has multiple peaks. These properties could accelerate the discovery of more diverse structures in inverse problems as shown in Chapters 3 and 4. This technology for structure generation is expected to be applicable to a wide variety of research fields. The structure models used in this thesis were dedicated to suit for targets by observing their patterns and compositions. This might be a problem when making a structure model for complicated objects. Recently, in the field of machine learning, many researchers have considered methods with deep neural network frameworks for generating images[101] and texts[13]. Some researchers have tried to use this framework for molecular design[50]. We should investigate the advantages and disadvantages of their methods and ours, and whether parts of their methods can be incorporated into our method.

5.1 Future work

5.1.1 Theoretical analysis of RPMCMC

One area for future work is further theoretical analysis of the RPMCMC algorithm. Here, we have only considered a simple comparison of the RPMCMC with other algorithms. Further properties remain to be investigated, for example, convergence, unbiased sampling to allow the exclusion of importance sampling, and ways to determine the strength of the repulsive force. Furthermore, our RPMCMC algorithm should be compared with other recently proposed algorithms[1, 72].

5.1.2 Possible applications in other fields

In the future, we are also interested in developing new applications in many other fields as outlined below.

- □ Tracking system [113] The Bayesian inference framework can solve tracking problems with multiple targets, multiple sensors, and multiple platforms.

- □ 3D visual generation [52] The prior information of 3D shapes and scenes is incorporated into probabilistic models. The inference algorithm tries to sample from the prior model under certain constraints, which corresponds to the posterior distribution of 3D images given the observed image/sketch.

- □ Network construction

  - Causal networks [6] With a graph sampler based on a new energy-domain Monte Carlo method, it is possible to design an efficient algorithm to generate Bayesian net structures from the marginal posterior distribution given experimental data.
  - Biological network [93] Network inference methods are widely applied to biological applications to quantify the regulatory relationships between intracellular components such as genes or proteins. Many methods have been proposed in this setting, but the connections in their statistical formulations have received less attention.

- □ Bayesian inference of phylogenetic trees [57, 33] The posterior probability of phylogenetic trees conditioned on an alignment of DNA sequences can be calculated using Bayes theorem.

- □ Medical image reconstruction

  - PET image [84] The Bayesian method was used for reconstruction of transmission and emission PET images.
  - Multi-slice CT [119] Multi-slice helical computed tomography scanning is an important technology for instant acquisition of information on internal organs, and is used for clinical diagnosis. Bayesian iterative algorithms applied to real 3D multi-slice helical data were developed to improve the image quality.

In these research fields, there is a massive amount of accumulated knowledge. To make a model of the inverse problem, some of this knowledge may be necessary. It might be useful for many researchers who are not familiar with statistics or machine learning to use a framework based on

the deep neural network to achieve a properly working generative model. We will endeavor to create a more general environment for *iqspr* which non-specialists from other fields can easily use to generate informative structures.

Appendix A

Random sample generation from standard distributions

In this appendix, various random variable generation methods, especially ones used in this thesis such as for the Gaussian, gamma, and Dirichlet distributions, will be explained. Before going into detail, we introduce a method for generating a pseudo random field on $[0,1]$. A sequence of random numbers generated by a computer is not genuinely random but pseudo random since it is completely determined by a relatively small number of initial values called the seeds. One of the most widely used random number generation methods is the Mersenne Twister [78]. This method was designed to correct most of the flaws found in older pseudo random generators, and is used as the default pseudo random generator in many programming languages, including Python and R, which are widely used for statistical computation.

A.1 Sampling from the inverse cdf (Exponential distribution)

If a cdf $F(x) = \int_{-\infty}^{x} p(z)dz$ for a pdf $p(x)$ is available, there is a very simple way to generate samples from the pdf. This basic idea is used in many other methods, including in this thesis for the slice sampler (Chapter 2.2.6). Once $Z \sim \text{Unif}(0,1)$ is obtained, its inverse $F^{-1}(Z)$ from $p(z)$ can also be obtained.

Proof

$$\begin{aligned} Pr(F^{-1}(Z) < z) &= Pr(Z < F(z)) && \text{(A.1)} \\ &= F(z) \text{ (since } Z \sim \text{Unif}(0,1) \text{)} && \text{(A.2)} \end{aligned}$$

Here, a simple example of a pdf is the exponential distribution $p(x) = \lambda e^{-\lambda x} (x > 0)$. In this case, the inverse cdf $F^{-1}(x) = \frac{-\log(1-x)}{\lambda}$ $(x > 0)$. This technique, however, only applies to a univariate distribution since it requires the cdf; thus it is difficult to use for inference of complicated models.

A.2 Box-Muller method (for sampling from the Gaussian distribution)

The Gaussian distribution is used for a wide range of applications. In order to obtain random variables from the Gaussian distribution, the Box-Muller method is useful. This involves transformations of continuous random variables. Suppose there exists a simple one-to-one transformation g from a random variable X to Y; then $h(y)$ which is a pdf of Y transformed using g from a random variable X that comes from pdf $f(x)$ is as follows:

$$h(y) = f(g^{-1}(x))|\frac{d}{dy}g^{-1}(y)|, \tag{A.3}$$

where $|\frac{d}{dy}g^{-1}(y)|$ is the Jacobian representing the change in volume between two spaces. If X and Y are both 2D random variables, then this Jacobian J is a determinant of the transformation matrix as follows.

$$|J| = \begin{vmatrix} \frac{\partial g^{-1}(\mathbf{y})_1}{\partial y_1} & \frac{\partial g^{-1}(\mathbf{y})_1}{\partial y_2} \\ \frac{\partial g^{-1}(\mathbf{y})_2}{\partial y_1} & \frac{\partial g^{-1}(\mathbf{y})_2}{\partial y_2} \end{vmatrix} \tag{A.4}$$

In the Box-Muller method, random variables U_1 and U_2 are obtained from the uniform distribution $\text{Unif}(0,1)$. The following transformation changes them into Gaussian random variables.

$$X_1 = \sqrt{-2\log U_1}\cos 2\pi U_2, \tag{A.5}$$

$$X_2 = \sqrt{-2\log U_1}\sin 2\pi U_2, \tag{A.6}$$

The proof can be obtained by using the characteristic function. A characteristic function of U_1 and U_2 is

$$\phi_{X_1}(\xi) = \int_0^1 \int_0^1 e^{-i\xi\sqrt{-2\log u_1}\cos 2\pi u_2} du_1 du_2. \tag{A.7}$$

Here, when using the transformation $u_1 = e^{-\frac{r^2}{2}}$, $u_2 = \frac{\theta}{2\pi}$, the Jacobian matrix is $|J| = \frac{r}{2\pi}e^{-\frac{r^2}{2}}$. The transformed characteristic function is

$$\phi_{X_1}(\xi) = \frac{1}{2\pi}\int_0^\infty \int_0^{2\pi} e^{-i\xi\cos\theta - \frac{r^2}{2}} dr d\theta. \tag{A.8}$$

Furthermore, when transforming variables $z = r\cos\theta$, $w = r\sin\theta$ with the Jacobian $|J| = r$, the characteristic function is

$$\begin{aligned}
\phi_{X_1}(\xi) &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-i\xi z - \frac{z^2+w^2}{2}} dz dw \\
&= \frac{1}{2\pi}e^{-\frac{\xi^2}{2}}\int_{-\infty}^{\infty} e^{-\frac{(z-i\xi)^2}{2}} dz \int_{-\infty}^{\infty} e^{-\frac{w^2}{2}} dw \\
&= \frac{1}{2\pi}e^{-\frac{\xi^2}{2}}.
\end{aligned} \tag{A.9}$$

This is just the characteristic function of the Gaussian distribution. Therefore, the transformed samples from Eqs. A.5 and A.6 are Gaussian random variables.

A.3 Rejection sampling (for sampling from the gamma distribution)

In rejection sampling, the first step is to find a proposal distribution $q(x)$, from which it is easy to obtain samples, satisfying

$$p(x) \leq cq(x), \tag{A.10}$$

where c is a constant and $p(x)$ is the target distribution. The sampling step starts by generating samples from the proposal distribution $q(x)$, then follows the weight computation $w(x) = \frac{p(x)}{cq(x)}$,

and discards a sample with probability $1-w(x)$. Obviously, finding a proposal $q(x)$ that is close to the target makes this sampling procedure more efficient.

Here we show a very important example to obtain gamma random variables. The gamma random variables and subsequently obtained Dirichlet random variables described later in this section are necessary for the Monte Carlo inference in the motif discovery problem considered in Chapter 3. This is achieved by using rejection sampling. The pdf of the gamma distribution is given by

$$f(x) = \frac{\beta^{-\alpha}}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} (x > 0), \tag{A.11}$$

where $\Gamma(\cdot)$ is the gamma function. This pdf is a monotonically decreasing function when $0 < \alpha < 1$ and the unimodal function has a mode at $\alpha - 1$. For this reason, two different approaches are used for the difference interval of α. For $0 < \alpha < 1$, Ahrens and Dieter [2] proposed that $cq(x)$ in the rejection sampling be taken as

$$cq(x) = \begin{cases} \frac{x^{\alpha-1}}{\Gamma(x)} & (0 < x < 1) \\ \frac{e^{-x}}{\Gamma(x)} & (x \geq 1) \end{cases}. \tag{A.12}$$

Here, c is $\int cq(x)dx = \frac{e+\alpha}{e\alpha\Gamma(\alpha)}$. Thus, the proposal distribution $q(x)$ is given by

$$q(x) = \frac{e}{\alpha+e} \alpha x^{\alpha-1} I(0 < x < 1) + \frac{\alpha}{\alpha+e} e^{-x+1} I(x \geq 1). \tag{A.13}$$

From the above, the proposed sample is obtained from $\alpha x^{\alpha-1}$ by using the inverse cdf with probability $\frac{e}{\alpha+e}$, and the exponential distribution with probability $\frac{\alpha}{\alpha+e}$. The corresponding acceptance probabilities are

$$w(x) = \begin{cases} e^{-x} & (0 < x < 1) \\ x^{\alpha-1} & (x \geq 1) \end{cases}. \tag{A.14}$$

For $\alpha \geq 1$, Cheng proposed that the proposal distribution $q(x)$ be the log-logistic function

$$q(x) = \lambda\mu \frac{x^{\lambda-1}}{(\mu + x^{\lambda})^2}, \tag{A.15}$$

where $\mu = \alpha^{\lambda}$, $\lambda = (2\alpha - 1)^{\frac{1}{2}}$ [18]. A random variable from $q(x)$ can be obtained through the inverse cdf. The cdf $\int_{-\infty}^{x} q(z)dz = \frac{x^{\lambda}}{\mu + x^{\lambda}}$, so using $U \sim \text{Unif}(0,1)$, the random variable X from the proposed distribution $q(x)$ is given by

$$X = \left(\frac{\mu U}{1-U}\right)^{\frac{1}{\lambda}}. \tag{A.16}$$

In this case, the constant c in the rejection sampling is $\frac{4\alpha^{\alpha}e^{-\alpha}}{\Gamma(\alpha)\lambda}$, and the quantity $\frac{g(x)}{cg(x)}$, which represents the acceptance probability, is 0.68, 0.8, 0.85, and 0.87 for αs of 1, 2, 5, and 10, respectively.

Dirichlet random variables are necessary to obtain the posterior distribution in the ZOOPs model used in Chapter 3 with the Gibbs sampler, although the full conditional distribution with repulsion is different from the Dirichlet variables, and thus the slice sampler was used. In addition to motif modeling, Dirichlet random variable generation is useful to infer natural language models such as the N-gram model and the topic model [12, 11], which may potentially be an extension of our chemical language model. Dirichlet random variables can be obtained by normalizing independent gamma random variables [126]. Let $X_1, \ldots, X_d$ be random variables from the gamma distribution $Gamma(\alpha_1, 1), \ldots, Gamma(\alpha_d, 1)$, respectively. Then $Y_1, \ldots, Y_d (Y_i = \frac{X_i}{\sum_j^N X_j})$ follow the d-dimensional Dirichlet distribution $Dir(\frac{\alpha_1}{\sum_j \alpha_j}, \ldots, \frac{\alpha_d}{\sum_j \alpha_j})$.

Appendix B

Convergence of Markov chain Monte Carlo

Some conditions for the convergence of the Markov chain Monte Carlo method are irreducibility and aperiodicity. A Markov chain having invariant distribution $\pi(dx)$ is *irreducible* if for any initial state, positive probabilities are assigned on sets where $\pi(dx) > 0$. A chain is called *periodic* if there is a part of the state space χ that is visited at certain regularly spaced times. If not, that chain is called *aperiodic*. A fundamental result, which will be proven in the later subsection as in [120], is that if a chain has a proper invariant distribution $\pi(dx)$ and is also irreducible and aperiodic, then $\pi(dx)$ is the unique equilibrium distribution of the chain.

To determine the convergence of the Markov chain, the following concepts of *recurrence* and *ergodicity* are introduced.

B.1 Definition of recurrence

Assume that $\{X_n\}$ is a π-*irreducible* chain with invariant distribution $\pi(\cdot)$.

1. The chain is recurrent if for $\forall B$ with $\pi(B) > 0$

$$Pr(X_n \in B \; i.o. | X_0 = x) > 0 \; \text{for} \; all x$$

and

$$Pr(X_n \in B \; i.o. | X_0 = x) = 1 \; for \; \pi - almost x,$$

where $\{X_n \in A \; i.o.\}$ denotes $\sum_i I(X_i \in A) = \infty$ with probability 1, which demonstrates that a particular set appears in the sequence infinitely often.

2. The chain is Harris recurrent if $Pr(X_n \in B \; i.o | X_0 = x) = 1$ for π-almost all x

B.2 Definitions of ergodicity

The total variation norm between two measures on the same space is used for the definition of ergodicity. The total variation norm for a signed measure λ on $A \in \chi$ is

$$||\lambda|| = \sup_{A \in \chi} \lambda(A) - \inf_{A \in \chi} \lambda(A). \tag{B.1}$$

Here, we show three different ways to define the ergodicity of the Markov chain.

1. A Markov chain is ergodic if it is positive Harris recurrent and aperiodic.

2. An ergodic chain with invariant distribution $\pi(dx)$ is called geometrically ergodic if there exists a real-valued function M such that $M(x) > 0$ and $E(|M(X)|) < \infty$, and for a positive constant $r < 1$ satisfying

$$||P^n(x,) - \pi|| \leq M(x) r^n. \tag{B.2}$$

3. The chain in 2) is called uniformly ergodic if there exists a constant $M < \infty$, and for a positive constant $r < 1$ satisfying

$$||P^n(x,) - \pi|| \leq M r^n. \tag{B.3}$$

B.3 Convergence result

The result regarding the convergence that will be shown in this subsection is from Tierney [120], and Hernandez-Lerma and Lasserre [53]. Suppose that the transition kernel $P(x,dy)$ is π-irreducible and π-invariant; then $P(x,dy)$ is positive recurrent and $\pi(dx)$ is the unique invariant distribution of $P(x,dy)$. If $P(x,dy)$ is also aperiodic, then for π-almost all x,

$$||P^n(x,) - \pi|| \to 0, \tag{B.4}$$

where $||\cdot||$ is the total variance norm. If $P(x,dz)$ is Harris recurrent, convergence occurs for all x.

B.4 Asymptotic behavior of the expectation

Revez (1975) first showed the following asymptotic behavior of the expectations [106]. Assume that X_n is ergodic with equilibrium distribution $f(x)$ and $h(x)$ is a real-valued function satisfying $E_f[|h(X)|] < \infty$; then for any initial distribution,

$$\bar{h}_n = \frac{1}{n}\sum_{i=m}^{n} h(X_i), \tag{B.5}$$

$$\mathbb{E}_\pi[h(X)] = \frac{1}{N-m+1}\sum_{i=m}^{N} h(X_i), \tag{B.6}$$

where m is the burn-in period, and goes to $E_f(h(X))$ almost surely.

B.5 Central limit theorem

The central limit theorem has also been proven in several different forms, but all of them require stronger assumptions for the Markov chain than the law of large numbers shown above. Suppose that the Markov chain X_n is uniformly ergodic with its equilibrium distribution $f(x)$ and a real-valued function $h(x)$ such that $E_f[h^2(X)] < \infty$; then, for any initial distribution, there exists a

positive constant σ_h and the following quantity

$$\sqrt{n}(\bar{h}_n - E_f[h(X)]) \tag{B.7}$$

converges weakly to a normal distribution with mean 0 and variance σ_h^2.

B.6 Convergence diagnostics

When using MCMC for inference, it is necessary to decide until when to throw away samples in the burn-in period and until when to generate samples to ensure the Markov chain has converged. Although there have been many criteria proposed by many researchers [42, 21], these criteria cannot be used to determine whether the Markov chain gets stuck in some local modes.

References

[1] Ahn, S., Chen, Y., and Welling, M. (2013). Distributed and adaptive darting monte carlo through regenerations. In *JMLR Workshop and Conference Proceedings*, volume 31, pages 108–116.

[2] Ahrens, J. H. and Dieter, U. (1974). Computer methods for sampling from gamma, beta, poisson and binomial distributions. *Computing*, 12:223–246.

[3] Akutsu, T. and Nagamochi, H. (2007). Comparison and enumeration of chemical graphs. *Comput Struct Biotechnol J*, 5:1–9.

[4] Ané, C., Larget, B., Baum, D. A., Smith, S. D., and A., R. (2007). Bayesian estimation of concordance among gene trees. *Molecular biology and evolution*, 24:412–426.

[5] Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. (2002). Tutorial on particle filters for online nonlinear/nongaussian bayesian tracking. *IEEE Trans. on Signal Processing*, 50:174–189.

[6] B., E. and Wong, W. H. (2008). Learning causal bayesian network structures from experimental data. *Journal of the American Statistical Association*, 103:778–789.

[7] Bailey, T. L. *et al*. (2010). The value of position-specific priors in motif discovery using meme. *BMC Bioinformatics*, 11:179.

[8] Berger, J. (1985). Statistical decision theory and bayesian analysis, 2nd edn.

[9] Bernardo, J. M. and Smith, A. F. M. (1984). Bayesian theory.

[10] Bertero, M. and Boccacci, P. (1998). Introduction to inverse problems in imaging.

[11] Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4):77–84.

[12] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

[13] Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. (2015). Generating sentences from a continuous space.

[14] Brown, N., McKay, B., and Gasteiger, J. (2006). A novel workflow for the inverse qspr problem using multiobjective optimization. *J Comput Aided Mol Des*, 20:333–341.

[15] C, B. (2010). *Pattern Recognition and Machine Learning*. New York: Springer.

[16] Chen, B., Polatkan, G., Sapiro, G., Blei, D., Dunson, D., and Carin, L. (2013). Deep learning with hierarchical convolutional factor analysis. *IEEE transactions on pattern analysis and machine intelligence*, 35:1887–1901.

[17] Chen, S. and Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Comput Speech Lang*, 13:359–394.

[18] Cheng, R. C. H. (1977). The generation of gamma variables with non-integral shape parameters. *Applied statistics*, pages 71–75.

[19] Chipman, H. A., George, E. I., McCulloch, R. E., Pavlis, G. L., Booker, J. R., Kalra, P., Mahapatra, P. B., and Aggarwal, D. K. (1998). Bayesian cart model search. *Journal of the American Statistical Association*, 93(443):935–948.

[20] Consortium", T. E. P. (2012). An integrated encyclopedia of dna elements in the human genome. *Nature*, 489:57–74.

[21] Cowels, M. K. and Carlin, B. P. (1996). Markov chain monte carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, 91:883–904.

[22] Cowles, M. and Carlin, P. (1996). Markov chain monte carlo convergence diagnostics: A comparative review. *J. Amer. Statist. Assoc.*, 91:883–904.

[23] da Fonseca *et al*. (2008). Efficient representation and p-value computation for high-order markov motifs. *Bioinformatics*, 24:i160–i166.

[24] Daubechies, I., DeVore, R., Fornasier, M., and Güntürk, C. S. (2010). Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63:1–38.

[25] de Candia, A. D. and A., C. (2002). Spin and density overlaps in the frustrated ising lattice gas. *Physical Review E*, 65:016132.

[26] Dede, C., Salzman, M. C., Loftin, R. B., and Sprague, D. (1999). Multisensory immersion as a modeling environment for learning complex scientific concepts. pages 282–319. Springer, New York.

[27] Del Moral, P. (2004). *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. New York: Springer.

[28] Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential monte carlo samplers. *JR Statist Soc B*, 68:411–436.

[29] Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *JR Statist Soc B*, 39:1–38.

[30] Dey, F. and Caflisch, A. (2008). Fragment-based *de novo* ligand design by multiobjective evolutionary optimization. *J Chem Inf Model*, 48:679–690.

[31] Doucet, A., de Freitas, J. F. G., and Gordon, N. J. (2001). *Sequential Monte Carlo Methods in Practic*. New York: Springer.

[32] Douguet, D., Thoreau, E., and Grassy, G. (2000). A genetic algorithm for the automated generation of small organic molecules: drug design using an evolutionary algorithm. *J Comput Aided Mol Des*, 14:449–466.

[33] Drummond, A. J. and Rambaut, A. (2007). Beast: Bayesian evolutionary analysis by sampling trees. *BMC evolutionary biology*, 7:1.

[34] E., R. C. (2006). *Gaussian processes for machine learning*.

[35] Earl, D. J. and Deem, M. W. (2005). Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23:3910–3916.

[36] Eaton, D. and Murphy, K. (2012). Bayesian structure learning using dynamic programming and mcmc. *arXiv preprint*, arXiv:1206:5247.

[37] Eddy, S. R. (1998). Profile hidden markov models. *Bioinformatics*, 14(9:755–763.

[38] Fechner, U. and Schneider, G. (2006). Flux (1): a virtual synthesis scheme for fragment-based *de novo* design. *J Chem Inf Model*, 46:699–707.

[39] Fraley, C. and Raftery, A. E. (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal*, 41:578–588.

[40] Gelfand, A. E. and Smith, A. F. M. (1990). Sampling based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409.

[41] Gelman, A., Roberts, R. O., and Gilks, W. R. (1996). Efficient metropolis jumping rules. pages 599–607.

[42] Gelman, A. and Rubin, D. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–511.

[43] Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.

[44] Geyer, C. J. and Thompson, E. A. (1995). Annealing markov chain monte carlo with applications to ancestral inference. *Journal of the American Statistical Association*, 90:909–920.

[45] Goi, C. *et al*. (2013). Cell-type and transcription factor specific enrichment of transcriptional cofactor motifs in encode chip-seq data. *BMC Genomics*, 14:S2.

[46] Gray, F. (1947). Pulse code communication. u.s. patent 2632058.

[47] Green, P. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82:711–732.

[48] Guha, R. (2007). Chemical informatics functionality in r. *J Stat Softw*, 18:1–16.

[49] Gupta, S. *et al*. (2007). Quantifying similarity between motifs. *Genome Biol.*, 8:R24.

[50] Gómez-Bombarelli, R. *et al*. (2016). Automatic chemical design using a data-driven continuous representation of molecules.

[51] Hachmann, J. *et al*. (2011). The harvard clean energy project: Large-scale computational screening and design of organic photovoltaics on the world community grid. *J Phys Chem Lett*, 2:2241–2251.

[52] Han, F. and Zhu, S. C. (2003). Bayesian reconstruction of 3d shapes and scenes from a single image. In *Higher-Level Knowledge in 3D Modeling and Motion Analysis, 2003. HLK 2003*, pages 12–20.

[53] Harnandez-Lerma, O. and Lasserre, J. B. (2001). Further criteria for positive harris recurrence of markov chains. In *Proceedings of the American Mathematical society*, volume 129, pages 1521–1524.

[54] Hastings, W. (1970). Monte carlo sampling methods using markov chain and their applications. *Biometrika*, 57:97–109.

[55] Higdon, D. M. (1998). Auxiliary variable methods for markov chain monte carlo with applications. *Journal of the American Statistical Assosiation*, 93:585–595.

[56] Huang, Q., Li, L., and Yang, S. (2010). Phdd: a new pharmacophore-based *de novo* design method of drug-like molecules combined with assessment of synthetic accessibility. *J Mol Graph Model*, 28:775–787.

[57] Huelsenbeck, J. P. and Ronquist, F. (2001). Mrbayes: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17:754–755.

[58] Hughes, J. *et al*. (2000). Computational identification of cis-regulatory elements associated with groups of functionally related genes in saccharomyces cerevisiae. *J. Mol. Biol.*, 296:1205–1214.

[59] 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.

[60] Hwang, C. R. (1988). Simulated annealing: theory and applications. *Applicandae Mathematicae*, 12:108–111.

[61] Ichonose, N. *et al*. (2012). Large-scale motif discovery using dna gray code and equiprobable oligomers. *Bioinformatics*, 28:25–31.

[62] Ikebata, H., Hongo, K., Isomura, T., Maezono, R., and Yoshida, R. (2017). Bayesian molecular design with a chemical language model. *J Comput Aided Mol Des*.

[63] Ikebata, H. and Yoshida, R. (2015). Repulsive parallel mcmc algorithm for discovering diverse motifs from large sequence sets. *Bioinformatics*, 31(10):1561–1568.

[64] Jurafsky, D. and Martin, J. H. (2008). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 2nd edition*. Prentice-Hall, New Jersey.

[65] Kalra, P., Mahapatra, P. B., and Aggarwal, D. K. (2006). An evolutionary approach for solving the multimodal inverse kinematics problem of industrial robots. *Mechanism and machine theory*, 41:1213–1229.

[66] Kawai, K., Nagata, N., and Takahashi, Y. (2014). *De novo* design of drug-like molecules by a fragment-based molecular evolutionary approach. *J Chem Inf Model*, 54:49–56.

[67] Kawai, K., Yoshimaru, K., and Takahashi, Y. (2011). Generation of target-selective drug candidate structures using molecular evolutionary algorithm with svm classifiers. *J Comput Chem Jpn*, 10:79–87.

[68] Kawashita, N. *et al*. (2015). A mini-review on chemoinformatics approaches for drug discovery. *J Comput Aided Chem*, 16:15–29.

[69] Kim, S. *et al*. (2015). Pubchem substance and compound databases. *Nucleic Acids Res*, 44:D1202–1213.

[70] Kirkpatrick, S. (1984). Optimization by simulated annealing: Quantitative studies. *Journal of statistical physics*, 34:975–986.

[71] Lameijer, E., Kok, J., Bäck, T., and Ijzerman, A. (2006). The molecule evoluator. an interactive evolutionary algorithm for the design of drug-like molecules. *J Chem Inf Model*, 46:545–552.

[72] Lan, S., Streets, J., and Shahbaba, B. (2014). Wormhole hamiltonian monte carlo. In *Proceedings of the AAAI Conference on Artificial Intelligence. . AAAI Conference on Artificial Intelligence*, page 1953. NIH Public Access.

[73] Lawrence, C. *et al*. (1993). Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment. *Science*, 262:208–214.

[74] Liang, F., Liu, C., and R., C. (2011). Advanced markov chain monte carlo methods: learning from past samples.

[75] Lipton, P. (2003). Inference to the best explanation.

[76] Liu, J. and Chen, R. (1998). Sequential monte carlo for dynamic systems. *J. Am. Statist. Ass.*, 93:1032–1044.

[77] Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. New York: SpRoberts99ringer.

[78] Matsumoto, M. and Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8:3–30.

[79] Metropolis, N. *et al*. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091.

[80] Mira, A. and Tierney, L. (2002). Efficiency and convergence properties of slice samplers. *Scand J Stat*, 29:1–12.

[81] Miyao, T., Hiromasa, K., and Funatsu, K. (2016). Inverse qspr/qsar analysis for chemical structure generation (from y to x). *J Chem Inf Model*, 56:286–299.

[82] Mohr, J., Jain, B., and Obermayer, K. (2008). Amolecule kernels: a descriptor- and alignment-free quantitative structure-activity relationship approach. *J Chem Inf Model*, 48:1868–1881.

[83] Mosegaard, K. and Tarantola, A. (1995). Monte carlo sampling of solutions to inverse problems. *Journal of Geophysical Research: Solid Earth*, 100(B7):12431–12447.

[84] Mumcuoglu, E. U., Leahy, R., and Cherry, S. R. (1994). Fast gradi.ent-based methods for bayesian reconstruction of transmission and emission pet images. *IEEE Transactions on Medical Imaging*, 13:687–701.

[85] Murphy, P. K. (2012). *Machine Learning: A Probabilistic Perspectiveg*. Cambridge: The MIT Press.

[86] Muthén, B. and Shedden, K. (1999). Finite mixture modeling with mixture outcomes using the em algorithm. *Biometrics*, 55:463–469.

[87] Nachbar, R. (1998). Molecular evolution: a hierarchical representation for chemical topology and its automated manipulation. *Genetic Programming 1998: Proceedings of the Third annual Conference*, pages 246–253.

[88] Neal, R. (2003). Slice sampling. *Ann. Stat.*, 31:705–767.

[89] Neirotti, J. P., Freeman, D. L., and Doll, J. D. (2000). Approach to ergodicity in monte carlo simulations. *Physical Review E*, 62:7445–7461.

[90] Nicolaou, C., Apostolakis, J., and Pattichis, C. (2009). *De novo* drug design using multiobjective evolutionary graphs. *J Chem Inf Model*, 49:295–307.

[91] Nielsen, T. D. and Jensen, F. V. (2009). *Bayesian networks and decision graphs*.

[92] Nocedal, J. (1980). Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35:773–782.

[93] Oates, C. J. and Mukherjee, S. (2012). Network inference and biological dynamics. *The annals of applied statistics*, 6:1209.

[94] O'Boyle, N. M. *et al*. (1999). Open babel: An open chemical toolbox. *J Cheminform*, 13:359–394.

[95] Opper, M. and Saad, D. (2001). *Advanced mean field methods: Theory and practice.* MIT press.

[96] Pavesi, G. *et al*. (2001). An algorithm for finding signals of unknown length in dna sequences. *Bioinformatics*, 17:S208–214.

[97] Pavlis, G. L., Booker, J. R., Kalra, P., Mahapatra, P. B., and Aggarwal, D. K. (1980). The mixed discrete-continuous inverse problem: Application to the simultaneous determination of earthquake hypocenters and velocity structure. *Journal of Geophysical Research: Solid Earth*, 85(B9):4801–4810.

[98] Prasad, S. and Singh, K. (2008). Interaction of usf1/usf2 and alpha-pal/nrf1 to fmr-1 promoter increases in mouse brain during aging. *Biochem Biophys. Res. Commun.*, 376:347–351.

[99] Psillos, S. (2005). Scientific realism: How science tracks truth.

[100] R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

[101] Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks.

[102] Radhakrishnan, S. *et al*. (2010). Transcription factor nrf1 mediates the proteasome recovery pathway after proteasome inhibition in mammalian cells. *Mol Cell.*, 38:17–28.

[103] Ralaivolaa, L., Swamidassa, S. J., Saigo, H., and Baldi, P. (2005). Graph kernels for chemical informatics. *Neural Networks*, 18:1093–1110.

[104] Redner, R. A. and Walker, H. F. (1984). Mixture densities, maximum likelihood and the em algorithm. *SIAM review*, 26:195–239.

[105] Reid, J. and Wernisch, L. (2011). Steme: efficient em to find motifs in large data sets. *Nucleic Acids Res.*, 39:e126.

[106] Revuz, D. (1975). *Markov chains*. Amsterdam: North-Holland.

[107] Roberts, G. and Rosenthal, J. (1999). Convergence of slice sampler markov chains. *JR Statist Soc B*, 61:643–660.

[108] Rubin, D. (1998). Using the sir algorithm to simulate the posterior distributions. In *Bayesian Statistics 3*.

[109] Rubin, D. B. (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *Annal of Statistics*, 12:1151–1172.

[110] Sandelin, A. *et al*. (2004). Jaspar: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res.*, 32(Database issue):D91–94.

[111] Sharov, A. and Ko, M. (2009). Exhaustive search for over-represented dna sequence motifs with cisfinder. *DNA Res.*, 16:261–273.

[112] Smith, A. *et al*. (2009). Mining chip-chip data for transcription factor and cofactor binding sites. *Bioinformatics*, 21:403–412.

[113] Stone, L. D., Streit, R. L., Corwin, T. L., and Bell, K. L. (2013). *Bayesian multiple target trackingc*. Artech House.

[114] Stuart, A. M. (2010). Inverse problems: a bayesian perspective. *Acta Numerica*, 19:451–559.

[115] T., B. (2011). Dreme: motif discovery in transcription factor chip-seq data. *Bioinformatics*, 348:1653–1659.

[116] T., B. and C., E. (1994). Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc. of the 2nd Int. Conf. on Intelligent Systems for Molecular Biology*, pages 28–36.

[117] Tanner, M. A. and Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82:528–550.

[118] Tarantola, A. (2005). *Inverse problem theory and methods for model parameter estimation*. siam.

[119] Thibault, J. B., Sauer, K. D., Bouman, C. A., and Hsieh, J. (2007). A three-dimensional statistical approach to improved image quality for multislice helical ct. *Medical physics*, 34:4526–4544.

[120] Tierney, L. (1994). Markov chains for exploring posterior distributions (with discussion). *Annals of Statistics*, 22:1701–1762.

[121] Tompa, M. *et al*. (2005). Assessing computational tools for the discovery of transcription factor binding sites. *Nat. Biotechnol.*, 23:137–144.

[122] Venkatasubramanian, V., Chan, K., and Caruthers, J. (1994). Computer-aided molecular design using genetic algorithms. *Comput Chem Eng*, 18:833–844.

[123] Venkatasubramanian, V., Chan, K., and Caruthers, J. (1995). Evolutionary design of molecules with desired properties using the genetic algorithm. *J Chem Inf Comput Sci*, 35:188–195.

[124] Wang, Y., Suzek, T., Zhang, J., Wang, J., He, S., Cheng, T., Shoemaker, B., Gindulyte, A., and Bryant, S. (2014). Pubchem bioassay: 2014 update. *Nucleic Acids Res.*, 42:D1075–1082.

[125] Whitley, D. (1994). A genetic algorithm tutorial. *Stat Comput*, 4:65–85.

[126] Wilks, S. (1962). *Mathematical statistics.* New York-London: John Wiley and Sons, Inc.

[127] Wingender, E. *et al*. (1995). Transfac: a database on transcription factors and their dna binding sites. *Nucleic Acids Res.*, 24:238–241.

[128] Wishart, D. S., Knox, C., Guo, A. C., Shrivastava, S., Hassanali, M., Stothard, P., Chang, Z., and Woolsey, J. (2006). Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Res.*, 34(Database issue):D668–72. 16381955.

[129] Wong, W. and Burkowski, F. (2009). A constructive approach for discovering new drug leads: using a kernel methodology for the inverse-qsar problem. *J Cheminform*, 1:1.

[130] Workman, C. and Stormo, G. (2000). Ann-spec: a method for discovering transcription factor binding sites with improved specificity. *Pac. Symp. Biocomput.*, 5:467–478.

[131] Xu, H. *et al*. (2011). The ccaat box-binding transcription factor nf-y regulates basal expression of human proteasome genes. *Biochim Biophys Acta.*, 1823:818–825.

[132] Yamashita, H., Higuchi, T., and Yoshida, R. (2014). Atom environment kernels on molecules. *J Chem Inf Model*, 54:1289–1300.

[133] Yan, Q. and de Pablo, J. J. (2000). Hyperparallel tempering monte carlo simulation of polymer systems. *Journal of Chemical Physics*, 113:1276–1282.

[134] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67:301–320.