BMC Bioinformatics

Research

DNA motif alignment by evolving a population of Markov chains

Chengpeng Bi[1,2]

Address: [1]Bioinformatics and Intelligent Computing Lab, Division of Clinical Pharmacology, Children's Mercy Hospitals, Kansas City, Missouri, USA and [2]Schools of Medicine, and Computing and Engineering, University of Missouri, Kansas City, Missouri, USA

Email: Chengpeng Bi - cbi@cmh.edu

from The Seventh Asia Pacific Bioinformatics Conference (APBC 2009)
Beijing, China. 13–16 January 2009

Published: 30 January 2009

BMC Bioinformatics 2009, **10**(Suppl 1):S13 doi:10.1186/1471-2105-10-S1-S13

This article is available from: http://www.biomedcentral.com/1471-2105/10/S1/S13

Abstract

Background: Deciphering *cis*-regulatory elements or *de novo* motif-finding in genomes still remains elusive although much algorithmic effort has been expended. The Markov chain Monte Carlo (MCMC) method such as Gibbs motif samplers has been widely employed to solve the *de novo* motif-finding problem through sequence local alignment. Nonetheless, the MCMC-based motif samplers still suffer from local maxima like EM.

Therefore, as a prerequisite for finding good local alignments, these motif algorithms are often independently run a multitude of times, but without information exchange between different chains. Hence it would be worth a new algorithm design enabling such information exchange.

Results: This paper presents a novel motif-finding algorithm by evolving a population of Markov chains with information exchange (PMC), each of which is initialized as a random alignment and run by the Metropolis-Hastings sampler (MHS). It is progressively updated through a series of local alignments stochastically sampled. Explicitly, the PMC motif algorithm performs stochastic sampling as specified by a population-based proposal distribution rather than individual ones, and adaptively evolves the population as a whole towards a global maximum. The alignment information exchange is accomplished by taking advantage of the pooled motif site distributions. A distinct method for running multiple independent Markov chains (IMC) without information exchange, or dubbed as the IMC motif algorithm, is also devised to compare with its PMC counterpart.

Conclusion: Experimental studies demonstrate that the performance could be improved if pooled information were used to run a population of motif samplers. The new PMC algorithm was able to improve the convergence and outperformed other popular algorithms tested using simulated and biological motif sequences.

Background

Discovering *cis*-regulatory elements or DNA motifs in genomic sequences is fundamental to build genetic networks and important to understand gene regulation in biological and pathological processes [1]. Although much algorithmic effort has been expended, it still remains challenging (see recent reviews in [2,3]). Multiple sequence local alignment coupling with position weight matrix

(PWM) updating, or PWM-based technique for short, has been widely used to solve the *de novo* motif discovery problem [2,3]. This problem is of combinatorial optimization, and it has proven NP-complete [4]. A motif model or PWM can be uniquely defined by a local alignment, and the PWM updating is applied to progressively calibrate the alignments until some specified criterion is met. In the following text, the terms local alignment and motif discovery are often interchangeable. Basically, the PWM updating technique consists of two broad approaches to approximating the local alignment solutions, deterministic and stochastic motif-finding algorithms. The deterministic local alignment algorithm developed by Lawrence and Reiley in 1990 [5] is rooted on the Expectation Maximization (EM) method [6]. EM has been widely applied in various scientific computing since its introduction due to its simplicity and efficiency [7]. The EM motif algorithm has spawned a plethora of its variations (see review in [8] and references therein), for example, a popular motif algorithms called MEME is an enhanced EM version [9].

The stochastic local alignment method is the second approach in the PWM updating. It has its origin in Markov chain Monte Carlo (MCMC) methods [10] pioneered by Metropolis et al. in 1953 [11] and generalized later on by Hastings [12]. The earliest MCMC-based motif discovery algorithm is called Gibbs motif sampler developed in 1993 [13], and later many other MCMC-based motif samplers have been investigated such as BioProspector [14]. Moreover, the EM algorithm relates to the MCMC algorithms in the sense that it can be viewed as a forerunner of the Gibbs sampler in its data augmentation version [10,15,16], also as seen in a recent comparative study on these two approaches to detecting protein domains [17]. The MCMC algorithms have some appealing features better than EM, for example, they can escape from the local optima suffered by the EM algorithms. Most importantly, MCMC can be used as a general framework for solving a wide range of complex problems where EM often fails, because for these problems computing the expectation or maximization steps as required in EM often become infeasible. In particular, MCMC provides a framework for drawing samples from a complicated target distribution that cannot be sampled with simpler deterministic methods.

In practical sequence alignment, the MCMC sampling algorithms may also suffer from local maxima like EM [3]. To remedy this limitation, a MCMC motif algorithm is often run a multitude of times each starting from different points, that is, there is a population of Markov chains running in parallel. This simple strategy is strongly recommended and should be able to efficiently explore the probability landscape with multiple modes [3]. The population-based Monte Carlo methods serve two purposes, that is, to improve or diagnose convergence [10,18] and escape from local maxima [3]. It can be classified into two categories: (1) running multiple independent Markov chains (IMC) without any information exchange between chains, and (2) enabling information exchange among a population of Markov chains (PMC). It is straightforward to implement IMC for any MCMC-based samplers. In real world, IMC is often encouraged in running MCMC motif samplers. However, as up to now PMC is rarely concerned with in sequence local alignment [3]. In literature, PMC exists in a diversity of forms. For example, parallel tempering (PT) evolves R Markov chains each attaching a different temperature [19]. Chains in PT exchanges information by swapping two states that are generated by mutation, and accepted by a modified Metropolis-Hasting rule. Note that PT is the population-based simulated annealing algorithm and it only exchanges information pairwise. The real PMC algorithm should be able to exchange information at the population level, for example, the evolutionary Monte Carlo (EMC) [20] and population MCMC [21]. This paper presents the PMC motif-finding algorithm, a novel local alignment method. It evolves a population of Markov chains (PMC) with information exchange, each chain being updated according to the population-based or pooled proposal distributions and Metropolis-Hasting rule. Experimental studies demonstrate that the new algorithm was able to improve the convergence as well as evolve some better local alignments while compared to IMC and other motif algorithms tested.

Results

Metropolis-Hastings sampler for local alignment

A Metropolis-Hastings motif sampler (MHS) was first devised to carry out multiple local alignment. MHS run a single Markov chain initialized from a random alignment seed, which is the so-called initial alignment ($\mathbf{A}^{(0)}$ with t = 0 step). A motif model or PWM matrix $\Theta^{(t)}$ at step t can be derived from such a sequence alignment ($\mathbf{A}^{(t)}$). Then a Monte Carlo simulation is processed as follows: first scanning all sequences by the motif model built at step t. The scanning process simply calculates the probabilities of each potential motif sites on the input sequences using equation (1). Second, new samples are drawn from each sequence according to proposed distributions built from the sites scanned in the step t. Third, a energy function as defined in equation (3) is calculated, and the Metropolis-Hastings acceptance rate defined in equation (6) is applied to decide whether or not the new samples (i.e. new alignment) are passed to the next iteration. Finally, the above procedure is iterated for a number of iterations or cycles (C). A parallel MHS (i.e IMC) run the MHS motif sampler a multitude of times (R) independently without any information exchange, whereas the population-based

MHS (i.e. PMC) run the same MHS sampler *R* times with information exchange enabled among all chains.

Scanning function

The information weight function originally proposed by Kullback and Leibler in 1951 [22] is used to approximate the probability of a potential site (a_i) on sequence i (S_i) given the current motif model $\Theta^{(t)}$,

$$p(a_i \mid \mathbf{A}^{(t)}, \boldsymbol{\Theta}^{(t)}) \propto \sum_{j=a_i}^{a'} \sum_{k=A}^{T} \log \left(\frac{\boldsymbol{\theta}_{j'k}^{(t)}}{\boldsymbol{\theta}_{0k}} \right)^{\delta(S_{ij},k)} \qquad (1)$$

where $a' = a_i + w - 1$, $j' = j - a_i + 1$, and δ is the indicator function. The information weight function simply indicates how similar a motif nucleotide probability ($\vec{\boldsymbol{\theta}}_j$) is relative to the background ($\vec{\boldsymbol{\theta}}_0$). Therefore, a site gets a larger scanning probability if its motif sequence is more different than the background. The scanning function can be derived in the EM motif algorithms [5,8] or Gibbs motif samplers [13,14,17].

Scoring function

Given a set of unaligned sequences (**S**), each local alignment can be treated as a configuration (v) in the whole local alignment space. The potential energy of a configuration (i.e. a local alignment $\mathbf{A}^{(t)}$) is defined by,

$$H(v) = H(\Theta^{(t)}, \mathbf{A}^{(t)}, \mathbf{S}), \qquad (2)$$

where **A** is the missing data (i.e. the motif sites), **S** is the observed data and Θ is a parameter matrix (or a motif model) to be estimated. Note that the potential energy has its real biological meaning, for example, it may indicate the protein-DNA binding affinity [23,24] or energy of a protein three-dimensional configuration [10]. Assuming the Boltzmann distribution, one can compute the probability of such a configuration ($p(v)$). Although the energy function $H(\cdot)$ can be defined in any forms of interests, here a simple local alignment log-likelihood function [8,9] is used, which is given as,

$$H(\mathbf{A}^{(t)}) \equiv Q(\boldsymbol{\Theta}^{(t)}) = \mid \mathbf{A}^{(t)} \mid \sum_{j=0}^{w} \sum_{k=A}^{T} \tau_t(\boldsymbol{\theta}_{jk}), \qquad (3)$$

where $|\mathbf{A}|$ is the aligned motif sites, and $\tau_t(\boldsymbol{\theta}_{jk}) = -\boldsymbol{\theta}_{jk}^{(t)} \log(1.0 / \boldsymbol{\theta}_{jk}^{(t)})$ is a negative entropy function. The energy function described as above and its variants have been widely adopted in maximum likelihood [5,8,9,17] or maximum a posterior based local alignment algorithms [10,13,14].

MHS sampling

The probability of a local alignment follows the Boltzmann distribution as,

$$p(\mathbf{A}|\mathbf{S}) = Z^{-1} \exp\{-\lambda H(\mathbf{A})\}, \qquad (4)$$

where $\lambda = 1/k_B T$, and Z is a normalization constant. Given the current local alignment ($\mathbf{A}^{(t)}$, $\Theta^{(t)}$), a new local alignment ($\mathbf{A}^{(n)}$) is proposed according to the following proposal distribution,

$$P(\mathbf{A}^{(n)} \mid \mathbf{A}^{(t)}) = \prod_{i=1}^{N} p(a_i \mid \mathbf{A}^{(t)}, \boldsymbol{\Theta}^{(t)}). \qquad (5)$$

As noted, the proposal distribution assumes each sequence is independently sampled. The proposed alignment is either accepted or rejected according to the Metropolis-Hastings rule, and its acceptance rate (α_H) is defined as,

$$\alpha_H = \min \left\{ 1, \exp(\lambda \Delta H) \frac{P(\mathbf{A}^{(n)}|\mathbf{A}^{(t)})}{P(\mathbf{A}^{(t)}|\mathbf{A}^{(n)})} \right\}, \qquad (6)$$

where $\Delta H = H(\mathbf{A}^{(n)}) - H(\mathbf{A}^{(t)})$. The transition probability from alignment $\mathbf{A}^{(t)}$ to $\mathbf{A}^{(n)}$ is thus expressed as follows: $T(\mathbf{A}^{(n)}|\mathbf{A}^{(t)}) = P(\mathbf{A}^{(n)}|\mathbf{A}^{(t)})\alpha_H(\mathbf{A}^{(n)}|\mathbf{A}^{(t)})$. The MHS algorithm keeps updating the current best alignment and associated maximum energy along the sampling process or Markov chain. The output is the final best alignment ($\mathbf{A}^*$) and its corresponding maximum energy (H_{max}) as well as the associated best motif model (Θ^*). It should be pointed out that the energy maximization is equivalent to the minimization, that is, the optimizations $\max\{H\}$ and $\min\{-H\}$ are the same.

The IMC version of the MHS motif algorithm can be easily implemented in parallel. Let *R* independent Markov chains each starting from a different initial alignment, and then the IMC algorithm run *R* chains separately. Thus, one can send each run (i.e. a single MHS sampler) to a different compute node for execution. A set of the best alignments collected from all samplers in nodes can be sorted out and output the top best alignments as the potential solutions (note that the top-10 solutions are output as the default).

The PMC motif-finding algorithm

Running multiple independent Markov chains (i.e. IMC) was originally designed to evaluate convergence [15], and later was found of its efficiency in improving convergence [18]. The goal of multiple local alignment is to uncover the target motifs hidden in a set of unaligned sequences as fast as possible. This requires that a MCMC motif-finding algorithm be able to efficiently explore the alignment space and in the same time locate as many near-optimal motifs as possible. In addition, to be practical, a Markov chain should be quickly converged since biological sequence alignment is a quite large-scale problem. As mentioned before, IMC technique is used in motif-finding on regular basis. However, it does not allow information exchange between chains. The idea of the PMC motif-finding algorithm is simply proposing the sampling distribution based on the current population of alignments, rather than on each individual. Since the transition matrix is based on the current population, the transition matrix of a single chain may be not stationary, but it evolves as a whole population which is stationary. It is hoped that such population-based information exchange would be able to improve the motif-finding performance and convergence as well. The PMC motif algorithm first initializes a population of R independent alignments ($\mathbf{A}^{(r0)}$: $r = 1, \cup, R$). Each individual alignment at step t ($\mathbf{A}^{(rt)}$) can be uniquely mapped to a PWM model $\Theta^{(rt)}$, which can be used to scan sequences the same as in a single MHS motif sampler described as above. Now the proposal distribution is based on the current population of alignments $\mathbf{A}^{(rt)}$. After scanning step, each individual often generates different distributions of site probabilities, $p(a_i^r|\mathbf{A}^{(rt)})$: $r = 1, \cup, R$, here $p(a_i^r \mid \cdot)$ is the site probability on location (a_i) of sequence i in the r-th Markov chain. To summarize the current population of site probabilities on each sequence location (a_i), its expectation $\bar{p}(a_i|R, t)$ and variance $\sigma(a_i|R, t)$ can be estimated as follows,

$$\bar{p}(a_i \mid R,t) = E[p(a_i)] = \sum_{r=1}^{R} p(a_i^r \mid \mathbf{A}^{(rt)})/R; \tag{7}$$

$$\sigma^2(a_i \mid R,t) = \sum_{r=1}^{R} \frac{[p(a_i^r|\mathbf{A}^{(rt)}) - \bar{p}(a_i|R,t)]^2}{R-1}. \tag{8}$$

One would argue that a larger variance (σ) may lead to a slow convergence. Thus the population-based proposal distribution is defined as,

$$\bar{P}(\mathbf{A}^{(rn)} \mid \mathbf{A}^{(rt)}) = \prod_{i=1}^{N} \bar{p}(a_i \mid R,t), \tag{9}$$

where $r = 1, \cup, R$. Each individual alignment so proposed is either accepted or rejected according to the Metropolis-Hastings rule, and the population-based acceptance rate (α_P) is defined as follows,

$$\alpha_P = \min\left\{1, e^{\lambda\Delta H^r}\frac{\bar{P}(\mathbf{A}^{(rn)}|\mathbf{A}^{(rt)})}{\bar{P}(\mathbf{A}^{(rt)}|\mathbf{A}^{(rn)})}\right\}, \tag{10}$$

where $\Delta H^r = H(\mathbf{A}^{(rn)}) - H(\mathbf{A}^{(rt)})$. The α_P rule is independently applied to all chains in the population.

This would ensure that each chain evolves independently, but adaptively progresses as a member in the population towards a unified direction.

Comparison of sampling trajectories

The trajectories were plotted using the original CRP binding data set. The IMC version run the MHS algorithm 5 times (i.e. $R = 5$), each being independent. The PMC algorithm was run in a population of either 5 or 10 individual chains, and the population evolved together. The sampling trajectories were plotted in Figure 1A–1C. Five chains were largely differing in the IMC case, that is, some converged quickly whereas others were very slow (Figure 1A). Note that these chains started from different initializations and ended up with distinct trajectories or equilibrium points. For the population-based motif algorithm (i.e. the PMC version), all chains (i.e. $R = 5$ or 10) quickly converged to the same global equilibrium level, as displayed in Figure 1B and 1C. This nicely illustrated that PMC improved the convergence in each chains. Moreover, the PMC alignment with $R = 5$ reached its maximum H's around iteration 24, whereas the PMC alignment with $R = 10$ achieved its maxima around iteration 45. This reveals that a larger population may slow down the convergence possibly due to a bigger variance (σ^2). Besides, more chains in the population may generate more dominant motif sites that compete for the equilibrium points. As a result, it should spend more time in burn-in phase before reaching a global equilibrium as more chains are added to the population. However, a larger population could explore more space. It is very hard to determine an optimal size, as it may be data-dependent.

On the other hand, the IMC version took the longest to come up with the best alignment at iteration 93 in 3 out of 5 chains. In addition, the IMC chains ended up with different results (Figure 1A), because some initial seeds result in good alignments with high energy (H), but oth-

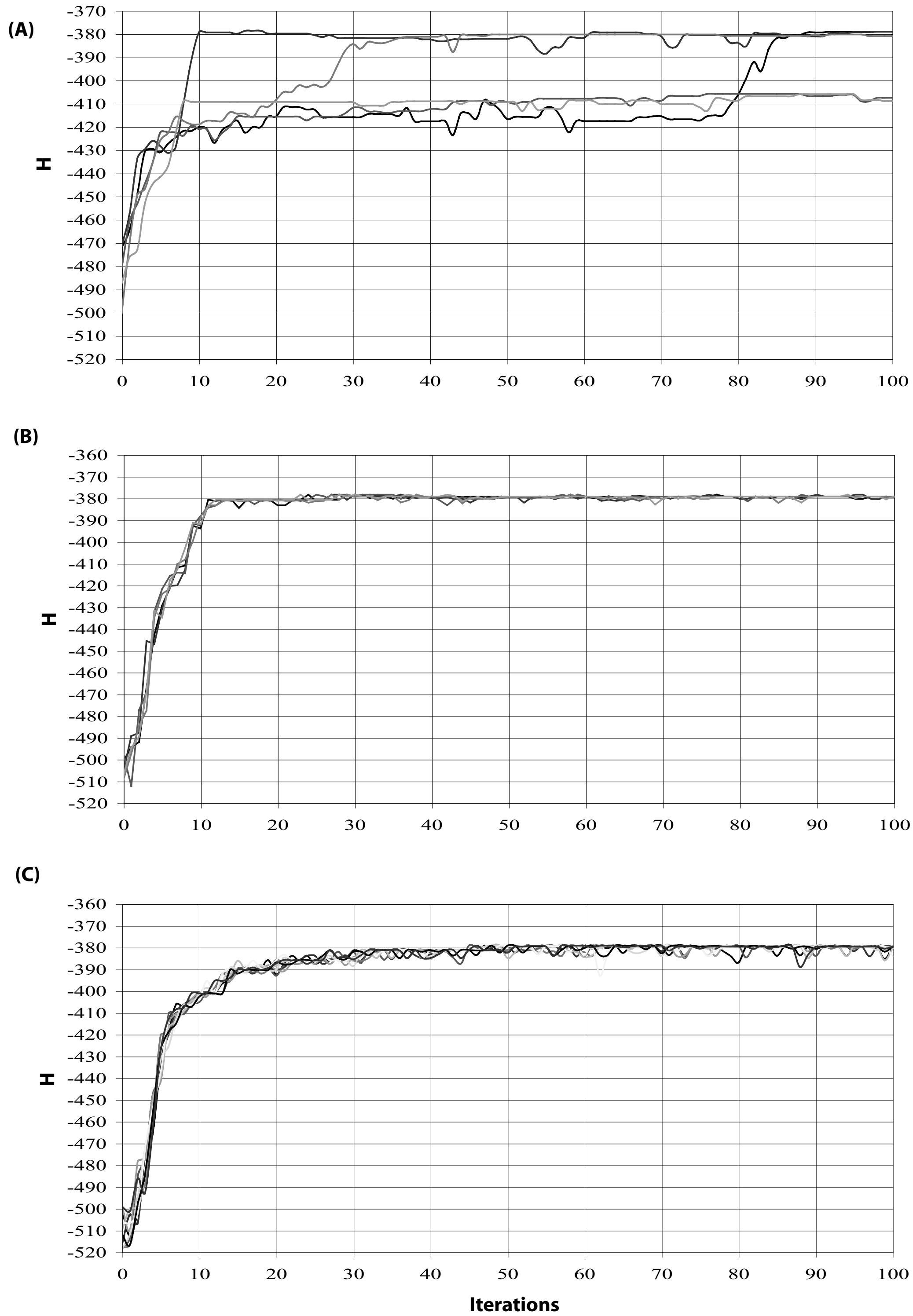

Figure 1

Trajectories of MC chains. The trajectories were plotted using the CRP binding data. Both IMC and PMC run the same MHS algorithm *r* times, but with different strategies as described in the text. (A) IMC with *r* = 5, (B) PMC, *r* = 5, (C) PMC, *r* = 10. Each arrow points to the alignment with the highest likelihood H_{max}.

ers in bad alignments with low *H*. This nicely demonstrates that a prerequisite for achieving good alignment solutions would be running multiple times of a MCMC-based sampler, which provides more opportunities of being able to search more space and thus escape from local maxima, as already suggested in the literature [3,17]. As noted in Figure 1B–1C, all PMC chains quickly converged to the same equilibrium, which produced a global optimal solution rather than a set of different solutions exhibited in IMC (Figure 1A).

Figure 2 summarizes the algorithm performance in different length groups of simulated CRP binding data sets. Both IMC and PMC versions successfully detected the planted motifs when $L \leq 400$ bp. PMC performed slightly better than IMC in each case. Overall, both performance dropped off as the background sequence becomes longer. They failed when the sequence length $L \geq 800$ bp. This is because a long random background sequence provides high chance of generating decoy signals that are highly conserved. To systematically test this kind of subtle motif discovery and further show the capability of the PMC motif-finding algorithm, the following section tested the well-formulated subtle motif problem.

Subtle motif discovery

The planted (*l*, *d*)-motif problem is a typical subtle motif-finding problem first considered by Sagot [25], and later was rigorously formulated by Pevzner and Sze [26]. Here the motif width is denoted as *l* instead of *w* in order to be consistent with the original problem definition. Since this problem came to exist, it has been challenging every motif-finding algorithms without exception [27]. It is commonly thought that position weight matrix (PWM)-based motif algorithms such as the EM motif-finders [5,9] and Gibbs motif samplers [13,14] are unable to solve the problem [26,27]. This section revisited the planted motif problem using the IMC and PMC algorithms developed in the paper, and compared with two other algorithms, Weeder [28] and Projection [27], which were specifically designed for tackling the planted problem. Table 1 summarized the performance comparison of the five motif discovery algorithms tested in the 10 planted (*l*, *d*)-motif cases. One can divide the 10 cases into two categories: the 5 easy cases, i.e. (11,2), (13,3), (15,4), (17,5) and (19,6), and the 5 tough cases, i.e. (10,2), (12,3), (14,4), (16,5) and (18,6). The easy cases are those planted motifs with slightly high conservation, whereas the tough cases with relatively low conservation. Overall, the PMC algorithm outperformed other algorithms with the grand average *nla* = 0.65. Assuming that a case is successfully detected if more than half planted sites are uncovered (*nla* ≥ 0.5). The PMC algorithm successfully detected almost all planted motifs with *nla* ranging from 0.55 to 0.93, and it failed in two very tough cases, (16,5) and (18,6). Among these detected cases, PMC predicted better than others in most cases (see the bolded numbers in Table 1) except for two cases, that is, Projection had the best *nla* = 0.84 in (17,5) and IMC with the best with *nla* = 0.96 in (11,2). Nonetheless, PMC still performed quite well in the two cases with *nla* = 0.93 in (11,2) and *nla* = 0.76 in (17,5). Projection is the second best predictor with average performance of *nla* = 0.53. Projection effectively located the 4 easy cases, (11,2), (13,3), (17,5) and (19,6) with *nla* = 0.95, 0.85, 0.84, 0.64, respectively. PMC achieved higher precision than Projection in the tough (10,2) case with *nla* = 0.73 vs. 0.53, as well as in the (19,6) case with *nla* = 0.75 vs. 0.64 as manifested in Table 1. Projection did not succeed in uncovering the remaining 4 tough cases: (12,3), (14,4), (16,5) and (18,6). Notably, previous report on Projection demonstrated its excellent performance in all planted cases [27], whereas this report showed discrepancies on it using the new unbiased motif data with the carefully chosen background sequences (see Methods). The difference may imply that Projection's performance is likely heavily dependent on the composition of the planted motifs used and their background sequences as well. Notice that although Projection adopted the random projection technique to pre-select a set of initial aligning seeds, it runs the EM algorithm as its final step that still suffers from local optimum. Taken together, results show that Projection did significantly improve the EM algorithm's performance (compare the average *nla*: 0.53 vs. 0.41), nonetheless, it still suffers from local maxima especially in subtle cases. The experiments allowed Projection to know the exact *d*'s in each cases, that is not required in other algorithms (and thus it may be not fair to other algorithms compared).

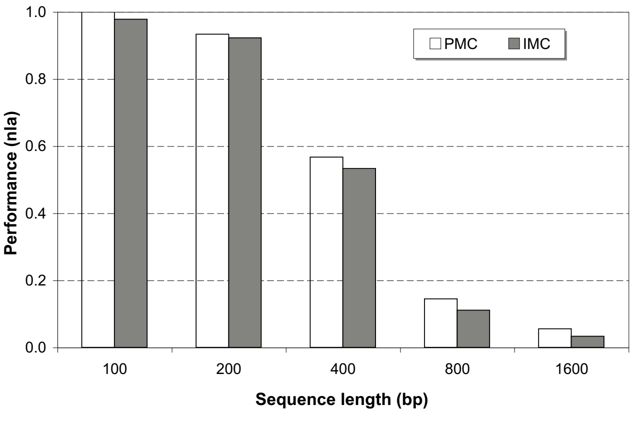

Figure 2
Performance comparison. The CRP binding motifs were planted in simulated background sequences with different lengths. Both IMC and PMC showed the same trend, that is, their performances go down as the sequence becomes longer.

Table 1: Comparison using (*l, d*)-motifs. Note that here EM is DEM [8]. The number in bold corresponds to the best predictor in that case row.

			Algorithms		
(*l, d*)	WEE	PRO	EM	IMC	PMC
10,2	0.46	0.53	0.32	0.42	**0.73**
11,2	0.74	0.95	0.47	**0.96**	0.93
12,3	0.27	0.30	0.29	0.22	**0.59**
13,3	0.44	**0.85**	0.43	0.72	**0.85**
14,4	0.29	0.25	0.31	0.21	**0.55**
15,4	0.27	0.67	0.40	0.50	**0.75**
16,5	0.20	0.16	0.32	0.13	0.28
17,5	0.23	**0.84**	0.59	0.39	0.76
18,6	0.20	0.12	0.34	0.14	0.32
19,6	0.20	0.64	0.58	0.26	**0.75**
ave	0.33	0.53	0.41	0.40	**0.65**

The IMC and EM motif-finders performed nearly equal on average (i.e. 0.40 vs. 0.41). These poor results achieved are in tune with previous reports [26,27], that is, the EM and MCMC-based motif algorithms had difficulties in solving the planted motif problem. IMC only correctly predicted three easy cases: (11,2), (13,3) and (15,4) with average *nla* = 0.96, 0.72 and 0.50, respectively, whereas EM barely uncovered two easy cases with long motif widths: (17,5) and (19,6) with *nla* = 0.59 and 0.58, respectively. Notice that the PMC algorithm is significantly better than its independent counterpart IMC in almost all the simulated cases. It is evident that the PMC motif algorithm is superior to IMC, see the 25% increment on average performance (i.e. 0.65 vs. 0.40). However, in finding highly conserved motifs, IMC and PMC may perform approximately the same, as will be shown in the JASPAR benchmarks.

Weeder only detected the (11,2) case and failed the remaining cases with average performance of *nla* = 0.33. Previous studies showed that Weeder was one of the best motif predictors in mammalian promoter regions [29], it may imply that a highly heterogeneous word frequency distribution made a major contribution to its success in real biological cases, whereas the word frequencies in the simulated backgrounds were nearly uniformly distributed, that may render the motif-finding much more difficulty to Weeder. In addition, Weeder may be preferred in non-oops motif discovery, because its strategy is that selecting as many significant words as possible and put all of them in the candidate list. In difficult situations, while other algorithms totally fail, Weeder may still have the chance to find some meaningful signals. However, such winning strategy becomes the weakness in *oops* motif discovery, because the more motifs reported, the higher false positives incurred. Currently, there is no way to force Weeder run as an *oops* model.

Testing JASPAR benchmark

In order to test the algorithm performance using the JASPAR data sets, one needs to generate simulated promoter sequences since the original JASPAR sequences are too short (about 25 bp on average). The simulated eukaryotic promoter sequences were generated by zero-order Markov model and they were composed of 45% GC content, which conforms to the ENCODE promoter regions [1]. Each JASPAR binding site was implanted into one simulated promoter sequence of length 500 bp. Note that the longer the sequence, the harder to locate the target as demonstrated in Figure 2. The 500-bp long sequence is approximately the same size as in a ChIP-chip data set, and should be a moderate size to challenge each algorithms tested. The Weeder and Projection programs were run the same as in the planted motif cases. For Projection, when l = 9 bp, d is still set to 2 mismatches, and when l = 20 – 22 bp, d = 7 – 8 mismatches as an extension to the original problem. Here the EM motif algorithm used is the popular MEME [9] run as its defaults.

Table 2 summarized the results of the five motif-finding algorithms. Compared to the planted motif problem, all algorithms tested using the JASPAR data carried out excellent jobs of finding almost all the experimentally verified motifs, the overall performance ranging from *nla* = 0.58 to 0.84. The PMC algorithm is still the best predictor with the average *nla* = 0.84, and IMC is very close to it with accuracy at *nla* = 0.82. MEME and Projection performed equally well with the average accuracy at *nla* = 0.75. Like in the planted motif cases, Weeder is the poorest predictor with average *nla* = 0.58 in the JASPAR data sets. In fact, Weeder predicted about 67% sites correctly, however, it also predicted more non-sites, and thus its average performance was decreased a bit. Notice that in the w = 9 case, both PMC and IMC predicted correctly whereas others failed.

The above experimental results illustrated the ability of the PMC algorithm to predict the binding sites not only in the simulated subtle motif data, but also in the experimentally verified data. PMC did outperform its independent counterpart IMC. In other words, information exchange between multiple Markov chains as imple-

Table 2: Comparison using JASPAR. Note that here EM is MEME [9].

			Algorithms		
w	WEE	EM	PRO	IMC	PMC
9	0.38	0.39	0.43	0.53	0.53
10	0.55	0.65	0.63	0.65	0.77
11	0.48	0.73	0.74	0.80	0.80
12	0.73	0.79	0.84	0.88	0.90
13	0.21	0.75	0.77	0.82	0.85
14	0.65	0.82	0.82	0.91	0.87
15	0.90	0.90	0.90	0.97	0.97
16	0.53	0.79	0.73	0.85	0.95
20	0.82	0.90	0.89	0.94	0.94
ave	0.58	0.75	0.75	0.82	0.84

mented in PMC improved the convergence and the motif prediction as well. If a population of MHS motif samplers evolves towards a common target function, it would be expected that the performance might be improved, because pooled information is used to inform each individual proposal distribution. Suppose an individual chain from one sampler starts from a bad point and it definitely results in a poor solution while evolving independently. Now, if the population information is used to define its proposal distribution, that would render the poor chain much better with the benefit of being a member, as illustrated in Figure 1.

Discussion

This paper presents a novel PMC motif-finding algorithm by evolving a population of Markov chains. The PMC motif algorithm exchanges local alignment information between individual chains by applying a pooled proposal distribution to all chains, and thus each individual chain can adaptively evolve towards a population-level equilibrium or global target function. Experimental studies demonstrate that the new PMC algorithm was able to improve the convergence and evolve better alignment solutions while compared to its multiple independent Markov chains method (IMC) and other algorithms. Further investigation into the population MCMC methods may be as follows: (1) disturbance added to the PMC chains may assist in exploring the alignment space, for example, one can incorporate genetic operators such as crossover and mutation into the PMC procedure; (2) fine-tuning sampling parameters might be informative to boost the accuracy of subtle motif discovery.

On the other hand, genetic algorithms [30], a class of adaptive global search methods modeled after biological systems, have been recently tried to overcome the limitations inherent in EM [31] as well as in MCMC [32]. It can be expected that computational intelligence techniques, which include genetic algorithms, neural networks and swarm intelligence, are likely to play important roles in sequence motif alignment.

Methods

Multiple sequence local alignment

The motif discovery problem is simply defined as: finding some recurrent short sequence patterns or motifs that are likely embedded in a given set of biological related sequences (**S**), for example, upstream promoter regions of co-regulated genes or enriched binding sequences in ChIP-chip experiments among others. Multiple local alignment is the most widely used method to locate over-represented sites in a given sequence set. The aligned over-represented sites are then used to build a frequency matrix that depicts a conserved domain or motif. Let $\mathbf{S} = \{S_1, ..., S_i, ..., S_N\}$ denote the sequence data set. Let L_i be the length of the sequence i (S_i) and S_{ij} denote a residue symbol taking on a value in K, for instance, $K = \{A, C, G, T\}$ is an alphabet of DNA sequences. If only one motif per sequence (i.e. *oops* model) is assumed, there are N motifs in total for N sequences. A zero or one motif per sequence (i.e. *zoops*) model is also frequently used. Nonetheless, both *oops* and *zoops* models assume that sequence data come from a two-component multinomial mixture model: (1) the background model, assuming that residues at non-motif positions follow an independent and identical multinomial distribution ($\vec{\boldsymbol{\theta}}_0$); and (2) the w-mer motif model (or w-motif), assuming that residues within the motif are independent but not identical, in other words, residues at different motif positions come from different multinomial distributions ($\vec{\boldsymbol{\theta}}_j$).

Let A_i be an indicator variable drawn from the location space $\{0,1\}^{(L_i - w+1)}$ of sequence i, $\mathbf{A} = [A_1, ..., A_i, ..., A_N]^T$ be the set of indicator variables representing the motif start sites (i.e. a local alignment) in the sequences, and w be the motif width. The total number of local alignments (V) can be generally expressed as: $V = \prod_{i=1}^{N} \binom{L_i - w + 1}{|A_i|}$, and here the number of motif sites on sequence i is defined as: $|A_i| = \Sigma_l A_{il}$. Therefore, if $|A_i| = 1$ for all i, it is an *oops* model, otherwise it is a *zoops* or multiple-site model. The total number of motif sites is $|\mathbf{A}| = \Sigma_i |A_i|$. Alternatively, an indicator variable $a_i = l$ is used to represent the motif starting at position l on sequence i, which is equivalent to $A_{il} = 1$. Note that $a_i = 0$ means no motifs found on sequence i. If multiple sites occur on a sequence, a vector ($\vec{a}_i$) is used to store all the positions. Obviously a motif indicator vector $\vec{a}_i \in \mathcal{P}_i$ $(\{1, 2, ..., L_i - w + 1\})$, here $\mathcal{P}_i$ is the power set of the i-th sequence motif sites. The alignment of motif sites is initialized by randomly generating a set of motif start sites (i.e. $\mathbf{A}^{(0)}$ or equivalently $[\vec{a}_1^{(0)}, ..., \vec{a}_N^{(0)}]^T$) and then it is progressively refined until convergence.

Evaluating the weight function

The information weight function defined in equation (1) needs estimation of two parameters: motif ($\vec{\boldsymbol{\theta}}_j$) and background ($\vec{\boldsymbol{\theta}}_0$). Suppose a counting matrix derived from a DNA motif alignment, $\{c_{jk}\}^{w \times 4}$, here c_{jk} is the number of the nucleotide k counted on position j in the motif alignment used, and then the information weight function (τ_{jk}), as originally defined by Kullback and Leibler in 1951 [22], is computed as,

$$\tau_{jk} = \log \frac{\boldsymbol{\theta}_{jk}}{\boldsymbol{\theta}_{0k}} = \log \frac{c_{jk} + \beta_k}{\sum_{k=1}^{|K|} c_{jk} + |\vec{\beta}|} - \log \boldsymbol{\theta}_{0k}, \tag{11}$$

where τ_{jk} is the information weight of the base k on motif position j. The pseudo-counts $\vec{\beta}$ are added to avoid zero counting in a motif alignment due to small sample, β_k is the pseudo-count of the base k, and $|\vec{\beta}| = \Sigma_k \beta_k$. It is often set as follows: $\beta_k = 1.25/4$ for a DNA sequence [33]. The background distribution $\vec{\boldsymbol{\theta}}_0$ can be estimated from the sequences or user-specified. Note that the expectation of the information weight defines the relative entropy function [34] or information content [24].

Overall objective function

Suppose each alignment is thought of as a hidden state in the alignment space, the motif discovery problem can be formulated as finding the optimized alignment state (v^*) among the entire alignment space. Index a state by $v \equiv [\vec{a}_1, ..., \vec{a}_i, ..., \vec{a}_N]^T = \mathbf{A}^{(v)}$, and let the potential energy of state v be $H^{(v)} = H(\mathbf{S}, \mathbf{A})$ where $\mathbf{A}$ is the alignment corresponding to the state v. The energy may be related to an alignment score or the motif sequence specificity/binding energy [23]. Then at equilibrium the probability of state v, $p(v)$, is calculated as,

$$p(v) = \frac{1}{Z(\lambda)} \exp\{-\lambda H(\mathbf{S}, \mathbf{A})\} \tag{12}$$

where the normalizing constant is defined as,

$$Z(1) = \int_{\mathcal{P}_1} \cdots \int_{\mathcal{P}_N} e^{-1H(\mathbf{S},\mathbf{A})} d\vec{a}_1 \cdots d\vec{a}_N.$$

Therefore, the optimized alignment state (v^*) is the one with the maximum probability (p^*). If v^* is found, then the parameter estimation (Θ^*) is done. However, computing the partition function or normalized constant (i.e. Z) is commonly intractable, because the alignment problem has proven *NP*-complete [4]. Markov chain Monte Carlo methods are frequently applied to approximate this kind of hard problem. This paper defined the maximum log-likelihood as the potential energy function, see equation (3).

Metropolis-Hastings sampler

Since it is hard to draw a sample directly according to equation (11), the Monte Carlo optimization methods such as the Metropolis-Hastings sampling (MHS) algorithm are often employed to iteratively draw a series of samples according to a proposed probability distribution such as the one in equation (5). Therefore, the sequence local alignment can be formulated as a Markov chain Monte Carlo (MCMC) optimization problem. Note that MCMC is a general framework for approximation methods in search of a large space characterized by complicated or unknown functions.

In solving the sequence local alignment problem, MHS generates samples (e.g. $\mathbf{A}$) from a probability distribution $p(\cdot)$, and explores the local alignment space $\Omega(\mathbf{S})$ using a Markov chain. MCMC does not sample directly from $p(\cdot)$, but only requires that the density $p(v)$ can be evaluated within a multiplicative constant $p(v) = \tilde{p}(v)/Z$ and $\tilde{p}(v)$ is the un-normalized target distribution. In equation (4), $\tilde{p} = \exp\{-\lambda H(\cdot)\}$. A Markov chain is a discrete-time stochastic process $\{\mathbf{A}^{(0)}, \mathbf{A}^{(1)}, \cup\}$ with property that the state $\mathbf{A}^{(t)}$given all previous values $\{\mathbf{A}^{(0)}, \mathbf{A}^{(1)}, \cup\ \mathbf{A}^{(t-1)}\}$ only depends on $\mathbf{A}^{(t-1)}$: $\mathrm{P}(\mathbf{A}^{(t)}|\mathbf{A}^{(0)}, \mathbf{A}^{(1)}, \cup\ \mathbf{A}^{(t-1)}) = P(\mathbf{A}^{(t)}|\mathbf{A}^{(t-1)})$. We call $P(\cdot|\cdot)$ the transition matrix of the Markov chain.

$P(\cdot|\cdot)$ is stationary, that is to say, independent of time or step (t).

The MHS sampler starts with any alignment $\mathbf{A}^{(0)}$ by randomly initializing a seed. Notice there exists a one-to-one map: $\mathbf{A}^{(t)} \to \Theta^{(t)}$. Then the MHS algorithm iterates the following two steps: (1) Propose a random perturbation of the current state (t), i.e., $\mathbf{A}^{(t)} \to \mathbf{A}'$, where $\mathbf{A}'$ can be viewed as generating from a proposal distribution $P(\mathbf{A}^{(t)} \to \mathbf{A}') = P(\mathbf{A}'|\mathbf{A}^{(t)})$ by a series of independent random sampling. Then, one can toss a Bernoulli coin with probability of α_H coming up heads: if heads show up, accepts the new move: $\mathbf{A}^{(t+1)} = \mathbf{A}'$; otherwise stay as it was: $\mathbf{A}^{(t+1)} = \mathbf{A}^{(t)}$.

Parallel MHS samplers

MCMC algorithms are often sequentially applied. A single chain is run for a long time until it converges to a stationary distribution. The states visited during the initial stage (i.e. burn-in phase) are usually thought to be unreliable and thus samples are discarded if reconstructing the density. However, this burn-in time should be short enough (i.e. well-mixing) to ensure the computational efficiency. The burn-in time is indeed depending on the initial points. A helpful strategy to select a good seed is simply running multiple independent MCMC chains, each of which starts with a different seed [18]. Suppose there are R parallel chains each of which is independent, and let the ensemble of alignments be $\vec{v} = (v_1, \cup, v_R)$, then the Boltzmann distribution of the ensemble is given rise to,

$$p(\vec{v}) = \prod_{r=1}^{R} p(v_r) = \frac{1}{Z^R} \exp\{-\sum_{r=1}^{R} \lambda H(v_r)\}. \qquad (13)$$

The above function treats the temperature (T) as constant and can be easily extended to a temperature variable (T_r) attached to a MCMC chain [20]. Notice that there are no information exchange among these chains. The population-based MCMC (i.e. PMC) simulates samples according to a pooled proposal distribution, that is, v_r comes from the $\bar{p}\,(v_r)$ defined in equation (7) rather than the $p(v_r)$ in equation (4) as information exchange occurs among chains.

Performance evaluation

The nucleotide-level accuracy (*nla*) metric is defined to evaluate the motif-finding algorithm performance based on the known alignment (**O**) and predicted alignment (**A**). Let o_i be the known motif position of sequence i, and a_i be the predicted motif position on the same sequence. The function *nla* is defined as,

$$nla(\mathbf{O}, \mathbf{A}) = \frac{1}{|\mathbf{A}|} \sum_{i=1}^{|\mathbf{A}|} \frac{|a_i \cap o_i|}{w}, \qquad (14)$$

where $|\mathbf{A}|$ is the number of motif sites, $|a_i \cap o_i|$ is the size of overlapping block between the predicted and observed motifs from the sequence i. The predicted alignment may exactly match the observed (nla = 1.0) or have some phase shifts (0.0 <nla < 1.0) or completely misaligned (nla = 0.0).

Motif-finders and their parameter settings

Both IMC and PMC versions run the same MHS motif-finding algorithms, and they were executed in the same parameter settings. The PMC population size was R = 20 as the default in all cases if not specified otherwise. The MEME algorithm (version 3.5.4) was also tested in the JASPAR data sets, and its parameters were set as its defaults except that MEME was notified of running the *oops* model, and the exact motif lengths were given as its input like other algorithms tested. The Projection algorithm [27] was specifically designed to deal with the planted (l, d)-motif discovery, and it requires the input of the l and d. The Projection program version 0.42 was tested with the same simulation data. Note that other algorithms tested do not require the d input. For example, to run the planted (10,2)-motif case, Projection executes the following command: "*findmotif -l 10 -d 2 -M 20 -s seqfile*". Projection can be downloaded from the website [35].

The Weeder algorithm [28] is one of the best motif predictors in mammalian promoters [29], however, its performance may be largely dependent on the nucleotide composition of real promoter regions. Several pre-calculated tables of word frequencies for various species (6-mers and 8-mers) come with the Weeder software. To cope with this requirement, new word frequency tables (6-mer and 8-mer words) were generated, for example, in the uniform case, each word was simply given frequency at 1. For other zero-order Markov backgrounds, first compute a w-word with minimum frequency: $f_{min} = (\theta_{0k'})^w$, here k' is the nucleotide with the least probability and adjust the value to 1.0, and accordingly other words with the frequency f_{word} are set to <Fences>Qf_{word}/f_{min} + 0.5<Fences>N (personal communication with Dr. Giulio Pavesi). The Weeder program version 1.31 was run with all the simulated data like this: "*weederlauncher.out seqfile bk large A*", here *bk* is the generated word frequency table. The Weeder user's manual can be found on the website [36].

Experimental data sets tested

The first example used is the gold standard CRP data set [5,8,9], which consists of 18 experimentally determined binding sequences (w = 22 bp) each being embedded within a 105 bp promoter sequence. The motif width is set as 22 base pairs long according to the verified site length. The local alignment is performed on both forward and reverse DNA strands. The motif discovery problem becomes more difficulty as the sequence length gets longer. To test the length (L) effect, the 18 verified sites were planted into different data sets, each being different sequence length: i.e. L = 100, 200, 400, 800 and 16,000 bp long. The simulated background sequences were generated based on the zero-order Markov model ($\vec{\boldsymbol{\theta}}_0$ = [0.302, 0.183, 0.209, 0.306]^T), which was estimated according to the original CRP data set.

Besides the length effect, other major factors such as the degree of motif conservation also have heavily impact on the performance. The data sets of the planted (l, d)-motif problem were generated as follows: given a fixed l-mer short motif (l = 10 to 19 bp), one generates 20 background sequences (N = 20) each with 600 bp long (L = 600), and each background sequence was randomly embedded with a variant of the consensus. A variant motif is a substring derived from the consensus with exactly d-position mutation (d = 2 to 6). The background distribution used in simulation is $\vec{\boldsymbol{\theta}}_0$ = [0.3, 0.2, 0.2, 0.3]^T. Each simulation was repeated 20 times and all simulated data sets were fed to the five algorithms tested. All computing jobs were submitted to the Apple Xserve cluster facility located in Children's Mercy Hospital's main campus. Notice that the same motif sets were also planted in two other backgrounds: [0.25, 0.25, 0.25, 0.25] and [0.2, 0.3, 0.3, 0.2], and similar conclusions could be drawn (data not shown.)

The JASPAR web server [37] provides experimentally verified binding sites data sets of eukaryotic transcription factors (TFs). However, the vast majority of these binding data were obtained by the SELEX method [24] that tested randomly generated double-strand oligonucleotides rather than genomic DNA sequences (personal communication with Dr. Boris Lenhard). Note that these verified motifs are highly conserved, each being short sequence (i.e. 25 bp on average). To challenge the *de novo* motif algorithms, these verified sites were planted into randomly simulated background sequences, each being 500 bp. A subset of the JASPAR binding sites was used to test the algorithms in the paper, that includes those TFs with more than 25 verified binding sequences and the motif width is at least 9 bp (i.e $w \geq 9$), as detailed in Table 2.

Competing interests

The author declares that they have no competing interests.

Authors' contributions

CB performed the work and wrote the manuscript.

Acknowledgements

The author would like to express his thanks to Dr. Boris Lenhard for discussing on the JASPAR data as well as to Dr. Giulio Pavesi for providing guidance on generating word frequency tables of simulated sequences.

This article has been published as part of *BMC Bioinformatics* Volume 10 Supplement 1, 2009: Proceedings of The Seventh Asia Pacific Bioinformatics Conference (APBC) 2009. The full contents of the supplement are available online at http://www.biomedcentral.com/1471-2105/10?issue=S1

References

1. Birney E, Stamatoyannopoulos JA, Dutta A, Guigo R, Gingeras TR, Margulies EH, Weng Z, *et al.*: **Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project.** *Science* 2007, **447:**799-816.
2. MacIsaac KD, Fraenkel E: **Practical strategies for discovering regulatory DNA sequence motifs.** *PLoS Computat Biol* 2006, **2:**e36.
3. Ji H, Wong WW: **Computational biology: Towards deciphering gene regulatory information in mammalian genomes.** *Biometrics* 2006, **62:**645-663.
4. Wang L, Jiang T: **On the complexity of multiple sequence alignment.** *J Comput Biol* 1994, **1:**337-348.
5. Lawrence CE, Reilly AA: **An expectation maximization algorithm for the identification and characterization of common sites in unaligned biopolymer sequences.** *Proteins: Structure, Function and Genetics* 1990, **7:**41-51.
6. Dempster AP, Laird AM, Rubin DB: **Maximum likelihood from incomplete data via the EM algorithm (with discussion).** *J Roy Statist Soc Ser B* 1977, **39:**1-38.
7. Meng XL, Dyk D: **The EM algorithm – an old folk-song sung to a fast new tune.** *J Roy Statist Soc Ser B* 1997, **59:**511-567.
8. Bi CP: **SEAM: A stochastic EM-type algorithm for motif-finding in biopolymer sequences.** *J Bioinform Comput Biol* 2007, **5:**47-77.
9. Bailey T, Elkan C: **Unsupervised learning of multiple motifs in biopolymers using expectation maximization.** *Machine Learning* 1995, **21:**51-80.
10. Liu JS: *Monte Carlo Strategies in Scientific Computing New York: Springer-Verlag*; 2002.
11. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller A, Teller H: **Equations of state calculations by fast computing machines.** *J Chem Phys* 1953, **21:**1087-1091.
12. Hastings WK: **Monte Carlo sampling methods using Markov chains and their applications.** *Biometrika* 1970, **57:**97-109.
13. Lawrence CE, Altschul SF, Boguski MS, Liu JS, Neuwald AF, Wootton JC: **Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment.** *Science* 1993, **262:**208-214.
14. Liu X, Brutlag DL, Liu JS: **BioProspector: Discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes.** *Proc. Pacific Symposium on Biocomputing (PSB)* 2001:127-138.
15. Robert CP, Casella G: *Monte Carlo Statistical Methods New York: Springer*; 1999.
16. van Dyk DA, Meng XL: **The art of data augmentation.** *J Comput Graph Stat* 2001, **10:**1-50.
17. Bi CP: **Data augmentation algorithms for detecting conserved domains in protein sequences: A comparative study.** *J Proteome Res* 2008, **7:**192-201.
18. Gilks WR, Richardson S, Spiegelhalter DJ: *Markov Chain Monte Carlo in Practice New York: Chapman and Hall*; 1996.
19. Geyer CJ: **Markov chain Monte Carlo maximum likelihood.** *Computing Science and Statistics: Proc. of the 23rd Symposium on the Interface* 1991:156-163.
20. Liang F, Wong WH: **Evolutionary Monte Carlo: Applications to C_p model sampling and change point problem.** *Statistica Sinica* 2000, **10:**317-342.
21. Laskey KB, Myers J: **Population Markov chain Monte Carlo.** *Machine Learning* 2003, **50:**175-196.
22. Kullback S, Leibler RA: **On information and sufficiency.** *Ann Math Statist* 1951, **22:**79-86.

23. Berg O, von Hippel PH: **Selection of DNA binding sites by regulatory proteins: Statistical-mechanical theory and application to operators and promoters.** *Journal of Molecular Biology* 1987, **193:**723-750.
24. Stormo GD: **DNA binding sites: representation and discovery.** *Bioinformatics* 2000, **16:**16-23.
25. Sagot MF: **Spelling approximate repeated or common motifs using a suffix tree.** *Theoretical informatics, 1380* 1998:111-127.
26. Pevzner P, Sze SH: **Combinatorial approaches to finding subtle signals in DNA sequences.** *Proc. First ISMB Conference* 2000, **1:**269-278.
27. Buhler J, Tompa M: **Finding motifs using random projection.** *Journal of Computational Biology* 2002, **9:**225-242.
28. Pavesi G, Mauri G, Pesole G: **An algorithm for finding signals of unknown length in DNA sequences.** *Bioinformatics* 2001, **17(suppl 1):**S207-S214.
29. Tompa M, Li N, Bailey TL, Church GM, De Moor B, *et al.*: **Assessing computational tools for the discovery of transcription factor binding sites.** *Nat Biotechnol* 2005, **23:**137-144.
30. Holland JH: *Adaptation in Natural and Artificial Systems Ann Arbor: Michigan: The University of Michigan Press*; 1975.
31. Bi CP: **A genetic-based EM motif-finding algorithm for biological sequence analysis.** *Proceeding of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, IEEE* 2007:275-282.
32. Bi CP: **Evolutionary Metropolis sampling in sequence alignment space.** *Proc. 2008 IEEE Congress on Evolutionary Computation (CEC), IEEE* 2008:189-194.
33. Frith MC, Hansen U, Spouge JL, Weng Z: **Finding functional sequence elements by multiple local alignment.** *Nucleic Acids Res* 2004, **32:**189-200.
34. MacKay DJC: *Information Theory, Inference, and Learning Algorithms New York: Cambridge University Press*; 2003.
35. **The Projection Software** [http://www.cse.wustl.edu/~jbuhler/pgt/]
36. **The Weeder Software** [http://159.149.109.9/modtools/]
37. Sandelin A, Alkema W, Engstrom P, Wasserman WW, Lenhard B: **JASPAR: an open-access database for eukaryotic transcription factor binding profiles.** *Nucleic Acids Res* 2004, **32:**D91-D94.