Sampling Good Motifs with Markov Chains

Chris Peikert

December 10, 2004

Abstract

Markov chain Monte Carlo (MCMC) techniques have been used with some success in bioinformatics [LAB+93]. However, these results rely on heuristic estimates of a Markov chain's mixing time. Without provable results, we cannot accurately judge the quality of an algorithm or the output it produces.

Our goal is to remedy this situation in the context of the motif-finding problem. Using combinatorial techniques from theoretical computer science [JS96], we design a Markov chain for sampling motifs, provide rigorous bounds on its mixing time, and describe the quality of the motifs it is likely to produce.

1 Introduction

1.1 Motif Finding

Genes are regulated by transcription factors, which bind at locations upstream of the genes and repress their transcription. Groups of related genes often have similar upstream sequences, called *motifs*, which cause the genes to all be expressed or repressed at the same time, depending on the presence of the transcription factor. If we suspect that a set of genes are all regulated by the same transcription factor, we can test this hypothesis by searching for substrings within the upstream regions that are as similar as possible. This is the essence of the motif-finding problem. We note that this is a simplification of the model considered in [LAB+93], but this simpler model retains the spirit of the problem and will prove to be more amenable to analysis.

More abstractly, motif-finding can be phrased as an optimization problem: suppose we are given n sequences $s_1, s_2, \ldots, s_n$, and a length k. We would like to find indices $x = (x_1, \ldots, x_n)$ such that the substrings $s_i[x_i \ldots x_i + k - 1]$ are the "most similar" according to some measure of similarity. Of course, the problem can be solved by iterating over all x, but the size of this space is exponential in n. Instead, we would prefer an algorithm which finds a best (or an almost-best) motif, and runs in time polynomial in n.

1.2 The Markov Chain Monte Carlo Method

The *Markov chain Monte Carlo (MCMC) method* is a general sampling technique which has found uses in statistics, mechanics, computer science, and other fields. A Markov chain defines a "random walk" over some abstract state space. One starts from an arbitrary state and makes local, randomized steps as specified by the chain. As the number of steps tends to infinity (for suitably-defined chains), this process converges to a some *stationary distribution* over the state space. The

number of steps required to become "close" to the stationary distribution is known as the *mixing time*. By taking enough steps and outputting the final state, one effectively samples from a close approximation of the stationary distribution.

When applying the MCMC method, one must design a Markov chain that has the desired stationary distribution, and which converges to this distribution quickly enough to be useful. In many cases, researchers only use heuristic measures to estimate the mixing time of a chain, but in fact there are many powerful combinatorial techniques which can yield provable, quantitative bounds.

1.3 Prior Work

The seminal paper of Lawrence *et al* [LAB+93] introduced MCMC methods to bioinformatics in the context of the multiple alignment problem (which can be viewed as a more general form of motif-finding). While their techniques are very clever and sophisticated, their use of random walks is only heuristic, and they are unable to give any guarantees about the quality of their algorithm's output. This weakness also extends to most uses of the MCMC method in physics and statistics, and also to the general technique of *simulated annealing*.

Theoretical computer science has recently seen an explosion of results that use MCMC methods to efficiently approximate hard-to-compute combinatorial quantities. Using a powerful equivalence between counting and sampling, several quantities can be estimated by sampling from a Markov chain process, including: the number of perfect matchings in a graph (and more generally, the permanent of a matrix) [JS89], the number of solutions to knapsack problems [DFK+93], the number of colorings of a graph [Jer95], the volume of a convex body [BDJ96], the number of valid configurations in monomer-dimer systems [JS89], and more.

1.4 Our Approach

Our purpose is to provide a rigorous analysis of MCMC techniques in the context of the motif-finding problem. We design algorithms using the MCMC method, and obtain quantitative results about their running times and the quality of the solutions they produce.

The overall structure of our motif-finding algorithm will be as follows: based on the input DNA strands, we (implicitly) construct a Markov chain over possible sets of motif locations. We run the chain from an arbitrary initial state for a prescribed number of steps, keeping track of the best state found within the run. We then re-run the chain a prescribed number of times and output the best state seen across all runs.

The mixing time of the chain tells us the number of steps we should run it. The stationary distribution bounds the probability that a single run produces a good solution, because the chance that the run *ever* hits a good solution is at least the probability assigned to good solutions by the stationary distribution (minus a negligible approximation quantity). By conducting an appropriate number of independent runs, we obtain a good solution with overwhelming probability. Therefore, constructing a suitable chain and performing a careful analysis allows us to quantify the exact running time of the algorithm and give rigorous statements about the quality of its output.

2 The Details

2.1 Background

Basic definitions. Let Ω be the state space of a Markov chain. $P : \Omega^2 \to [0,1]$ gives the transition probabilities, such that for all $x \in \Omega$, $\sum_{y\in\Omega} P(x,y) = 1$. It is known that if the chain is ergodic (i.e., irreducible and aperiodic), it has a unique *stationary distribution* π over Ω, which is approached as the number of steps tends to infinity (regardless of initial state). Let $P^t(x,y)$ be the probability that the chain occupies state y when started from a particular state x and run for exactly t steps. The *variation distance* from stationarity is defined as

$$\Delta_x(t) = \frac{1}{2}\sum_{y\in\Omega} \left|P^t(x,y) - \pi(y)\right|,$$

and the rate of convergence to π is measured by the *mixing time*

$$\tau_x(\epsilon) = \min\{t \ : \ \Delta_x(t') \leq \epsilon \ \forall t' \geq t\}.$$

For $e = (x,y) \in \Omega^2$, define $Q(e) = Q(x,y) = \pi(x)P(x,y)$. We say that the chain is *time-reversible* if $Q(x,y) = Q(y,x)$ for all $x, y \in \Omega$ (this is also known as the *detailed balance* condition). Finally, let $f : \Omega \to \mathbb{R}$ be some scoring function over the states.

Bounding the mixing time. The speed at which a chain approaches its stationary distribution is related to the eigenvalues of the matrix of transition probabilities. Specifically, the mixing time is related to the second-largest eigenvalue, in absolute value (the largest eigenvalue being 1). Analyzing the eigenvalues directly is often too difficult, but a variety of combinatorial techniques can be used to bound the eigenvalues or reason about the mixing time directly. Among these approaches are *coupling* and/or *path coupling* arguments, using *canonical paths* to bound the *edge congestion*, and analysis of the chain's *conductance*.

In this work, we will use the canonical paths technique to bound the edge congestion and thereby the chain's mixing time. In this approach, we identify the Markov chain with a graph $G = (\Omega, E)$, where $(x,y) \in E$ iff $Q(x,y) > 0$. In our case, G will be undirected because our chain will be time-reversible. For every ordered pair $(x,y) \in \Omega^2$, we describe a *canonical path* γ_{xy} through G from x to y. We will desire a set of canonical paths that does not rely on any single edge too heavily; intuitively, the existence of such a set of paths means that the graph is very "well-connected," therefore the Markov chain should mix quickly. Formally, we define the *congestion* to be the following quantity:

$$\rho = \max_{e\in E} \frac{1}{Q(e)} \sum_{\gamma_{xy}\ni e} \pi(x)\pi(y)|\gamma_{xy}|,$$

where $\gamma_{xy} \ni e$ means that γ_{xy} uses the directed edge e, and $|\gamma_{xy}|$ is the length of the path. With this definition, we have the following:

Proposition 1 ([Sin92]). *Let $\mathbb{M}$ be a finite, time-reversible, ergodic Markov chain over Ω with self-loop probabilities $P(x,x) \geq 1/2$ for all $x \in \Omega$ and stationary distribution π. If the congestion of $\mathbb{M}$ is ρ, then the mixing time of $\mathbb{M}$ satisfies $\tau_x(\epsilon) \leq \rho(\ln \pi(x)^{-1} + \ln \epsilon^{-1})$, for any choice of initial state x.*

2.2 A Useful Chain

Our overall goal is to define a Markov chain from which we can sample good motifs. We therefore desire a chain whose stationary distribution is highly non-uniform, so that states with high scores are assigned much greater probability mass.

Here is one general way of compactly defining such a Markov chain, independent of the particulars of the motif-finding problem. Suppose we start from a connected, undirected graph $G = (\Omega, E)$ of uniform degree D. We can then define the following (oft-used) Markov chain $\mathbb{C}(\lambda)$, which depends on a parameter λ whose role will become clear below:

From state x,

1. With probability $1/2$, stay at x. Otherwise,
2. Let y be a uniformly random neighbor of x. Go to y with probability $\min\{1, \lambda^{f(y)-f(x)}\}$.

Note that the chain always accepts transitions to states y that have better scores than x (according to f), and probabilistically rejects transitions to states having worse scores. When $\lambda = 1$, the chain always accepts transitions without regard to score, while under larger values of λ, the probability of rejection tends to 1. Therefore in the limit as $\lambda \to \infty$, the chain simply becomes a "randomized greedy" walk which only accepts transitions that improve the score. This trade-off suggests that we must carefully choose an intermediate value of λ, so that high-scoring states are favored without the chain becoming stuck in a local optimum.

We now dispense with some technical details about $\mathbb{C}(\lambda)$. First, observe that it is ergodic: $P(x, x) \geq 1/2$ for all x, so the chain is aperiodic, and G is connected, so the chain is irreducible. Next, define a distribution π over Ω: $\pi(x) = \lambda^{f(x)}/Z(\lambda)$, where $Z(\lambda) = \sum_{u \in \Omega} \lambda^{f(u)}$ is just a normalizing factor. An easy calculation verifies that π is the stationary distribution and that the chain is time-reversible. Therefore, for $\lambda > 1$, the stationary distribution favors higher-scoring states, indicating that $\mathbb{C}$ may be useful for solving optimization problems. We now apply it to the specific problem of finding good motifs.

2.3 Applying the Chain to Motif-Finding

Given this definition of $\mathbb{C}(\lambda)$, we need only define an underlying graph $G = (\Omega, E)$ specifically related to motifs within our input DNA sequences. We do so as follows: the states are n-tuples $x = (x_1, \ldots, x_n)$ specifying the substrings within each of the n DNA sequences. For simplicity, we assume that each DNA sequence has length $n+k-1$, so there are only k possible offsets within each DNA sequence, and $\Omega = [k]^n$. Two states x, y are neighbors if they differ in *at most one* index (so every x is a neighbor of itself, which simplifies the analysis). We note that every state has exactly kn neighbors. The scoring function $f(x)$ can be any function that evaluates the "similarity" of the substrings specified by x. We assume that $f(x) \in [0, 1]$ for all $x \in \Omega$, and that more-similar substrings are assigned higher scores. Several popular scoring methods (e.g., information content, consensus scoring) can easily be adapted to satisfy these constraints.

Proposition 2. *If λ is polynomial in n and k, the chain $\mathbb{C}(\lambda)$ is rapidly mixing. Specifically, the mixing time of $\mathbb{C}(\lambda)$ satisfies $\tau_x(\epsilon) \leq 2\lambda^2 n^2(\ln \lambda + n \ln k + \ln(1/\epsilon))$ for any $x \in \Omega$.*

Proof. We bound the mixing time of $\mathbb{C}(\lambda)$ using the technique of canonical paths and edge congestion. The canonical paths we choose are simple: the path γ_{xy} from any state x to any state y is

the path which takes n steps, changing x_i to y_i on the ith step. (If $x_i = y_i$ for some i, the ith step is just a self-loop.)

Recall that the mixing time of the chain is related to the maximum edge congestion:

$$\rho = \max_e \frac{1}{Q(e)} \sum_{\gamma_{xy} \ni e} \pi(x)\pi(y)|\gamma_{xy}|.$$

We will give an upper bound on ρ via the following steps: first, we observe that $|\gamma_{xy}| = n$. Next, we will give a lower bound on $Q(e)$ for every e. Next, we will give an upper bound on $\pi(x)\pi(y)$. Finally, we will count the number of paths $\gamma_{xy} \ni e$ and give a lower bound on the value $Z(\lambda)$.

Here is a lower bound on the value of $Q(e)$ for any $e = (u, v)$ where $v \neq u$:

$$\begin{aligned} Q(u,v) &= \pi(u)P(u,v) \\ &= \frac{\lambda^{f(u)}}{Z(\lambda)} \cdot \frac{\min\{1, \lambda^{f(v)-f(u)}\}}{2nk} \\ &= \frac{\lambda^{\min\{f(u),f(v)\}}}{2nkZ(\lambda)} \\ &\geq \frac{1}{2nkZ(\lambda)} \end{aligned}$$

(Of course, $Q(u, u) \geq 1/2Z(\lambda)$, which is even better.)

Now we bound $\pi(x)\pi(y)$:

$$\begin{aligned} \pi(x)\pi(y) &= \frac{\lambda^{f(x)+f(y)}}{Z(\lambda)^2} \\ &\leq \frac{\lambda^2}{Z(\lambda)^2}. \end{aligned}$$

Next, notice that for any edge $e = (u, v)$ where u and v (possibly) differ only in the ith index, the only canonical paths γ_{xy} that pass through e are those for which $x_j = u_j$ for $j = i, \ldots, n$, and for which $y_j = v_j$ for $j = 1, \ldots, i$. Therefore there are only k^{i-1} choices for x, and k^{n-i} choices for y, for a total of k^{n-1} paths $\gamma_{xy} \ni e$. Also observe that $Z(\lambda) = \sum_{u \in \Omega} \lambda^{f(u)} \geq |\Omega| = k^n$.

We are now ready to combine all these bounds:

$$\begin{aligned} \rho &\leq \max_e 2n^2 k Z(\lambda) \sum_{\gamma_{xy} \ni e} \frac{\lambda^2}{Z(\lambda)^2} \\ &\leq 2n^2\lambda^2 k \frac{k^{n-1}}{Z(\lambda)} \\ &\leq 2n^2\lambda^2. \end{aligned}$$

To apply Proposition 1, we simply observe that $\pi(x) \geq 1/Z(\lambda) \geq 1/\lambda k^n$, and we are done. □

Proposition 2 guarantees that the chain $\mathbb{C}(\lambda)$ will be rapidly mixing. However, it guarantees nothing about the *quality* of the motifs that are produced by the Markov chain process. We now turn our attention to that issue.

2.4 Approximate Optimality of the Output

Recall that the stationary distribution of the chain $\mathbb{C}(\lambda)$ is $\pi(x) = \lambda^{f(x)}/Z(\alpha)$. We would like to argue that in the stationary distribution, the total probability mass over optimal states is some non-negligible quantity δ. If this were the case, then running the chain $O(1/\delta)$ times and taking the best answer among all runs would yield an optimal motif with overwhelming probability.

Unfortunately, a naive analysis of the stationary distribution cannot yield such a strong result. Consider a scoring function f which assigns value 1 to a single state x, and .99 to all others. Then the probability mass on the optimal state is about $\lambda^{.01}/k^n$, which is exponentially small for any λ that is polynomial in n (which is required for our rapid-mixing analysis to go through).

However, we can still achieve meaningful results if we slightly relax our requirements, and if we make some (hopefully valid) assumptions about our input and scoring function. Instead of requiring an optimal solution, it may be enough to find an *approximately optimal* one, i.e. one whose score is only slightly smaller than that of an optimal solution. Depending on the application, an approximation could still be useful:

- It might still identify a common subsequence in most of the DNA strands, narrowing the search for a common transcription factor.
- To test the hypothesis that a group of genes contain a common motif, it may be enough to find a motif whose score is much larger than what one would expect by chance alone. According to most popular scoring functions, if a solution with a very high score exists, then several solutions with reasonably large scores also exist. Finding any of these solutions might be enough to reject the null hypothesis.

By including approximately-optimal solutions in our set of "good" motifs, we can significantly increase the probability mass that is assigned to good solutions by the stationary distribution.

A complementary approach would be to assume some random distribution on the parts of the DNA strands that lie *outside* the motif. Under such an assumption, one could potentially prove that most states have very low scores, thereby decreasing the relative weight assigned to "bad" solutions by the stationary distribution.

Here we describe some sufficient conditions to establish that the Markov chain produces a good solution with good probability. We stress than none of these conditions appear to be necessary; we are only providing some general conditions that could potentially be met by a careful analysis of the scoring function and the source of input.

Consider, as a function of score s, the number of states that have score approximately s. If this function grows only geometrically as s *decreases*, then we can prove that the stationary distribution assigns enough weight to good solutions. More formally, let $N(a,b) = |\{x \in \Omega \ : \ f(x) \in [a,b)\}|$. Then we have:

Proposition 3. *Suppose there exists a polynomial $T(n,k)$ and constant $\epsilon > 0$ such that*

$$\frac{N(1-(j+1)\epsilon, 1-j\epsilon)}{N(1-j\epsilon, 1-(j-1)\epsilon)} \leq T(n,k)$$

for every integer $j > 0$ such that $N(1-j\epsilon, 1-(j-1)\epsilon) > 0$. Then for $\lambda = T(n,k)^{1/\epsilon}$, $\mathbb{C}(\lambda)$ is rapidly mixing, and the stationary distribution of $\mathbb{C}(\lambda)$ assigns probability mass at least $\epsilon/(T(n,k)+\epsilon)$ to states whose scores are within ϵ of optimal.

Proof. First observe that because T is a polynomial and ϵ is constant, $\lambda = T(n,k)^{1/\epsilon}$ is also a polynomial. By Proposition 2, $\mathbb{C}(\lambda)$ is rapidly mixing.

Now let j_0 be the integer such that $\max_{x\in\Omega} f(x) \in [1 - j_0\epsilon, 1 - (j_0 - 1)\epsilon)$, i.e. j_0 identifies the interval containing the optimal score. Say that any state whose score lies outside this interval is "bad," while the remainder are "good." Clearly all good states have score within ϵ of optimal.

Let $M(a,b)$ be the probability mass assigned by the stationary distribution to states x such that $f(x) \in [a,b)$. It will be sufficient to prove that for all $j > j_0$,

$$\frac{M(1 - j\epsilon, 1 - (j-1)\epsilon)}{M(1 - j_0\epsilon, 1 - (j_0 - 1)\epsilon)} \leq T(n,k).$$

Because there are only $1/\epsilon$ such j, the total mass assigned to bad states is at most $T(n,k)/\epsilon$ times the mass assigned to good states. The result immediately follows.

Let us now prove the desired inequality. Note that

$$M(1 - j\epsilon, 1 - (j-1)\epsilon) \leq \frac{\lambda^{1-(j-1)\epsilon}}{Z(\lambda)} \cdot N(1 - j\epsilon, 1 - (j-1)\epsilon). \tag{1}$$

Now, by assumption on N and by taking a telescoping product, we know that

$$N(1 - j\epsilon, 1 - (j-1)\epsilon) \leq T(n,k)^{(j-j_0)} \cdot N(1 - j_0\epsilon, 1 - (j_0 - 1)\epsilon).$$

But by our choice of λ, $T(n,k)^{(j-j_0)} = \lambda^{\epsilon(j-j_0)}$. Replacing terms in (1), we get

$$\begin{aligned} M(1 - j\epsilon, 1 - (j-1)\epsilon) &\leq \lambda^{\epsilon} \cdot \frac{\lambda^{1-j_0\epsilon}}{Z(\lambda)} \cdot N(1 - j_0\epsilon, 1 - (j_0 - 1)\epsilon) \\ &\leq T(n,k) M(1 - j_0\epsilon, 1 - (j_0 - 1)\epsilon), \end{aligned}$$

as desired. □

3 Further Work

We note that at this time we are unable to prove that any efficient algorithm unconditionally gives a good approximation of an optimal motif (though we are able to give reasonable conditions which guarantee positive results). We see a few technical hurdles standing in the way of this goal: first, the canonical paths technique tends to give loose bounds on the mixing time. In addition, we point out that the approach seems inherently limited in our particular chain: consider two high-scoring but completely different motifs represented by states x and y. Any path from x to y will go through an intermediate edge $e = (u,v)$ whose endpoints represent *bad* motifs by any reasonable scoring function. Therefore, e's congestion will be "charged" an amount of $\pi(x)\pi(y)/Q(e)$, which is a relatively large quantity. It is possible that other analytical techniques such as coupling, or analyzing the conductance, could circumvent these problems.

We also would like to prove that the stationary distribution assigns large weight to good solutions. This requires knowing something about the structure of the scores. Our Proposition 3 supposes one useful structure, though there are undoubtedly others. In particular, it would be helpful to study a particular scoring function and argue that it does not penalize "near-misses" of an optimal motif by too much. Using such a result, one could argue that there are many solutions with scores almost as good as optimal, and hence that large mass is assigned to them.

References

[BDJ96] R. Bubley, M. Dyer, and M. Jerrum. A new approach to polynomial-time random walks for volume computation. Manuscript, 1996.

[DFK+93] M. Dyer, A. Frieze, R. Kannan, A. Kapoor, L. Perkovic, and U. Vazirani. A mildly exponential time algorithm for approximating the number of solutions to a multidimensional knapsack problem. *Combinatorics, Probability and Computing*, 2:271–284, 1993.

[Jer95] M. Jerrum. A very simple algorithm for estimating the number of k-colourings of a low-degree graph. *Random Structures and Algorithms*, 7:157–165, 1995.

[JS89] M. Jerrum and A. Sinclair. Approximating the permanent. *SIAM Journal on Computing*, 18:1149–1178, 1989.

[JS96] Mark Jerrum and Alistair Sinclair. *Approximation algorithms for NP-hard problems*, chapter 12, pages 482–520. PWS Publishing Co., 1996.

[LAB+93] Charles E. Lawrence, Stephen F. Altschul, Mark S. Boguski, Jun S. Liu, Andrew F. Neuwald, and John C. Wootton. Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, October 1993.

[Sin92] A. Sinclair. Improved bounds for mixing rates of markov chains and multicommodity flow. *Combinatorics, Probability and Computing*, 1:351–370, 1992.