Parsing regulatory DNA: General tasks, techniques, and the PhyloGibbs approach

Rahul Siddharthan
The Institute of Mathematical Sciences, Chennai 600 113, India
(Fax, 91 44 2254 1586; Email, rsidd@imsc.res.in)

In this review, we discuss the general problem of understanding transcriptional regulation from DNA sequence and prior information. The main tasks we discuss are predicting local regions of DNA, *cis*-regulatory modules (CRMs) that contain binding sites for transcription factors (TFs), and predicting individual binding sites. We review various existing methods, and then describe the approach taken by PhyloGibbs, a recent motif-finding algorithm that we developed to predict TF binding sites, and PhyloGibbs-MP, an extension to PhyloGibbs that tackles other tasks in regulatory genomics, particularly prediction of CRMs.

[Siddharthan R 2007 Parsing regulatory DNA: General tasks, techniques, and the PhyloGibbs approach; *J. Biosci.* **32** 863–870]

1. Introduction

An important problem in computational biology is understanding the regulation of genes from DNA sequence and other information. Most cellular biological processes require that certain sets of genes be turned on, and other genes turned off, in a carefully orchestrated manner. Gene regulation can occur at various levels: by chemical modification of the DNA and chromatin itself, at the transcriptional level (controlling or preventing the recruitment of the RNA polymerase that transcribes a messenger RNA molecule), or at the post-transcriptional level (by targeting the mRNA molecule). In recent years much attention has been paid to the post-transcriptional regulatory machinery in cells, mediated by RNA molecules (for recent reviews, *see* He and Hannon 2004; Matzeke and Birchler 2005), and there are suggestions that, at least in eucaryotes, transcription happens much more widely than previously believed and post-transcriptional regulation plays a more important role than had been appreciated (see Pearson 2006 for a review). Nevertheless, transcriptional regulation is more basic and better understood, and we focus on it here.

Genes are transcriptionally regulated by specialised proteins (in particular, transcription factors, TFs) – a nomenclature we will use for any protein that binds to DNA and plays a regulatory role in the transcription process) that recognise specific short sequences in the DNA, bind to them, and help recruit the RNA polymerase (or, in some cases, inhibit it). In eucaryotes, genes are in general controlled by several transcription factors, whose combinatorial logic can be quite complex. TFs generally bind upstream of the gene, but often also in introns, downstream or occasionally even within coding sequence.

In bacteria, and in the simplest eucaryotes like yeast, there is very little intergenic sequence (a few tens of bases in bacteria, a few hundreds in yeast), and all of it could potentially be regulatory. In most other organisms, there can be many thousands of bases between genes, only some of which plays a role in gene regulation. TF binding sites are typically found either in the promoter region near the start site of a gene, or in localised "*cis*-regulatory modules" (CRMs), typically 1–2 kilobases long, that contain binding sites for various factors that control the gene during specific cellular situations. These may be found a few kilobases from the start site, or in introns or downstream. Two key

Keywords. PhyloGibbs; regulatory DNA; transcription factors

Abbreviations used: CRMS, *cis*-regulatory modules; MCMC, Markor Chain Monte Corlo; PWMS, position weight matrices; TFs, transcription factors

computational tasks are, thus, to predict CRMs and to predict binding sites for transcription factors.

In addition, many enhancers are known to be very far away from the genes they regulate – up to a megabase away (Lettice *et al* 2003). (This is in sequence space: in the actual nucleus, depending on how the chromatin is packaged, they may be very near the target gene.) These are difficult to determine ab initio, but if they have been discovered by other means, they can be examined computationally for binding sites.

This article is an expository discussion of the general techniques used to tackle these tasks, and the specific approach of a recent program we wrote, PhyloGibbs (Siddharthan *et al* 2005) (and its successor, PhyloGibbs-MP (R Siddharthan, unpublished results). For technical details of PhyloGibbs, the reader is referred to the original papers, and for a practical guide on how to use PhyloGibbs, there is an upcoming review article (Siddharthan and van Nimwegen 2007) as well as documentation (including manual pages) in the PhyloGibbs program distribution itself.

2. General considerations in computational regulatory genomics

2.1 *Posterior probabilities and Bayes' theorem*

Probability calculations generally calculate the probability of an event given a model for that event. In biological sequence analysis, the event (sequence data) is given to us, and we are required to find an appropriate model for that event (location of regulatory regions or binding sites). In other words, given a data set S, and given a number of possible models h_1, h_2, . . . , each of which could have given rise to S with probabilities $P(S \mid h_1)$, $P(S \mid h_2)$, . . . , we want to evaluate $P(h_i \mid S)$: the probability of each model given the data S.

$P(S \mid h_i)$ is called the "likelihood" of S given h_i, and $P(h_i \mid S)$ is called the "posterior probability" of h_i given S. Generally, not all hypothesis are equally likely *a priori*, so we also want to know, upfront, the "prior probability" $P(h_i)$ of each hypothesis. Then, Bayes' Theorem gives the posterior probability:

$$P(h_i|S) = \frac{P(S|h_i)P(h_i)}{\sum_j P(S|h_j)P(h_j)} \quad (1)$$

where the denominator sums over all hypothesis, and thus evaluates to the "prior probability" of S, which can usually be treated as a constant and ignored. When all hypotheses are equally likely *a priori*, the posterior probability is proportional to the likelihood $P(S \mid h_i)$.

Bayes' theorem is discussed in most statistics texts; in the frequentist limit, where there is a large number of trials and all prior probabilities and likelihoods can be rigorously calculated, it is easily provable. But it is widely used (and, arguably, most useful) in situations not amenable to a frequentist approach.

2.2 *Describing binding sites*

Transcription factors (TFs) recognize short sequences in DNA (typically about 15–25 bp long in bacteria, 7–15 bp in eucaryotes). Typically, also, these are not exact strings or even "strings with k mismatches" and it is hard to apply standard fast substring-finding algorithms (e.g.Ukkonen 1995; Sagot 1998; Amir *et al* 2004) from computer science to them. They are usually represented as "position weight matrices" (PWMs), which are 4 x N matrices where N is the width of a binding site and the element W_{an} gives the frequency or probability of nucleotide α at position n in a binding site.

There is an underlying assumption, usually not too bad, that the columns in a weight matrix are independent. Given the limited amount of data in most cases, it is hard to go beyond this assumption, to incorporate (for instance) dinucleotide correlations.

If the weight matrix columns are normalised (so that the elements are probabilities), given a string S of length n, we may be interested in the probability that it is a binding site for a TF described by the weight matrix W. What we can calculate more readily is the likelihood of observing the string S given that it is a binding site for W; this is $\Pi_n W_{S_n n}$. This needs to be compared with the "background" probability of this string. In the simplest case, if the string is either a "binding site" (with probability p_w) or "background" (with probability $p_b = 1 - p_w$), given a "background model" we can calculate the posterior probability of the site being a binding site represented by W, using Bayes' theorem:

$$P(W|S) = \frac{P(S|W)p_w}{P(S|W)p_w + P(S|B)p_b} \quad (2)$$

(where the calculation of $P(S \mid B)$ is discussed further in the next section). This formula works well by itself in practice. But, in general, there are other possibilities: the string may be a binding site for some other factor W′, it may partially overlap a binding site for W, and so on. All these possibilities are taken care of elegantly in the Stubb algorithm for predicting *cis*-regulatory modules.

2.3 *Describing background DNA*

By "background DNA" we mean DNA of unknown function, whose statistical properties may be taken to be generic. The simplest background model is to assign a probability of 0.25 for each nucleotide. The next simplest is to assign

non-uniform probabilities, to take account of the fact that certain nucleotides occur more often than others (for example, A and T usually occur more frequently than C and G). Here we also need to note variation in the frequencies of nucleotides in different parts of the genome: for example, the frequencies may be different in coding and non-coding DNA, in heterochromatin and euchromatin, and so on.

For example, with a background model where the probability of each nucleotide is 0.25, for a string S of length L, $P(S \mid B) = 0.25^{L}$. If the probabilities of A and T are 0.3 and the probabilities of C and G are 0.2, for a string S of length L containing N_{AT} A's and T's and $N_{GC} = L - N_{AT}$ G's and C's, $P(S|B) = 0.3^{N_{AT}}\, 0.2^{N_{GC}}$.

In actual DNA, it turns out that dinucleotide correlations are important: the frequencies of, say, C and A do not accurately predict the frequency of the dinucleotide CA. Therefore, it is common to use a Markov model for the DNA background, where the probability of each base depends on its predecessor. Then,

$$P(S|B) = p(S_1)\prod_{n=2}^{L} p(S_n \mid S_{n-1}), \qquad (3)$$

where $p(\alpha)$ is the background probability of the base α, and $p(\alpha|\alpha')$ is the conditional probability of α given that its predecessor was α'. Motif-finders commonly include such a Markovian background, which can be generalised to higher orders (so that the probability of a base depends conditionally on its m predecessors).

2.4 *Predicting* cis-*regulatory modules*

Since *cis*-regulatory modules contain a larger-than-normal density of binding sites, when one knows a candidate set of transcription factors and the position weight matrices that describe them, an obvious approach to detecting CRMs is to look for predicted binding sites for these PWMs and see where they cluster unusually strongly.

Nearly all CRM-detection programs do this, but a naive approach doesn't work very well, because there are too many predicted sites for each factor, many of which are false positives, and it is difficult to determine "enriched" regions for these sites. It turns out that including phylogenetically close species helps a lot. Cis-Analyst (Berman *et al* 2002, 2004) clusters predictions from multiple species using various stringency criteria, with some success at predicting known modules. Stubb (Sinha *et al* 2003, 2004, 2006) takes into account exclusion between different factors, and calculates a free energy of binding, for each factor and overall, in user-determined "windows"; it, too, can take advantage of one additional input species, and performance improves dramatically.

A recent alternative approach (Pierstorff *et al* 2006) uses conserved sequence (local ungapped sequence) alone to predict CRMs, and previously characterised PWMs are not needed at all.

In contrast, PhyloGibbs-MP looks for clusters of binding sites that it itself predicts *ab initio*. It can take previously characterised PWMs as "prior information", but typically these only improve the confidence level of predictions, and do not actually change the predictions themselves. This is discussed further below.

2.5 *Predicting binding sites* ab initio

Once a stretch of regulatory DNA (a promoter, or CRM, or enhancer) is identified, it is of interest to predict individual binding sites in that region. When several sites for a TF are already known and a reasonable PWM can be constructed, binding sites can be predicted using that PWM, as described in § 2.2. However, an important problem is to predict binding sites *ab initio*, that is, without the help of a previously constructed PWM (which in many cases is not available).

The task, given a stretch of DNA sequence (or several stretches), is to determine which short, fuzzy sequences are statistically overrepresented: that is, to partition the input sequence into likely "binding sites" (samples from a PWM) and "background" (represented by the background model). Typically we have restrictions on allowed parses, such as the number of different factors (motifs) expected and the number of binding sites per factor. Even with these restrictions, the space of possible parses is too large to search exhaustively. Two common approaches are expectation maximisation on mixture models [used by the MEME algorithm (Bailey and Elkan 1994], which we don't discuss here, and Gibbs sampling.

Gibbs sampling is a version of Markov-chain Monte-Carlo (MCMC) sampling, and was first used in the context of biological motif-finding by Lawrence *et al* (1993). Other implementations have been made over the years; in the following, we describe our implementation, PhyloGibbs, and a forthcoming upgrade, PhyloGibbs-MP.

PhyloGibbs is a Gibbs-sampling motif finder that we introduced recently (Siddharthan *et al* 2005), containing several enhancements over other motif-finders:

- It (optionally) uses orthologous sequence from closely-related species, since orthologous genes are expected to be regulated similarly. It scores phylogenetically-related regions differently from independently-evolved sequence.
- Unlike "footprinting" methods, it searches in both conserved and unconserved regions. This is important because binding sites are known to evolve, even in closely related species (Dermitzakis *et al* 2003; Emberly *et al* 2003; Tanay *et al* 2005).

- It uses a two-stage strategy that finds a best set of predictions by simulated annealing, and then statistically evaluates their significance by sampling.

We showed that, on test data sets of experimentally annotated binding sites in yeast, PhyloGibbs predicts known sites with much better specificity (at a given sensitivity) than four other programs we benchmarked.

3. The Gibbs sampler algorithm as implemented in PhyloGibbs

3.1 *Markov Chain Monte Carlo sampling*

Gibbs sampling is a particular case of Markov Chain Monte Carlo (MCMC) sampling; we briefly review it here. (For a more through review of MCMC methods in general, *see* Smith and Roberts 1993.) In general, this approach is needed when one is confronted with the problem of searching for an optimal state, or averaging a weighted function over all states, in a state space that is too large to search exhaustively. For example, each state S may have a probability $P(S)$ associated with it, and we may either want to find the "most probable state" that maximises $P(S)$, or find the expectation value of a function $f(S)$ in this space, $\langle f \rangle = \Sigma_{s'} f(S')P(S')$.

Instead of searching, or calculating the average over, every state S, the idea is to pick a subset of states whose distribution is given by $P(S)$. That is, each state is selected with a probability $P(S)$. MCMC is a method of constructing such a set of states, by starting from an initial state S_1, using it to select a new state S_2, using S_2 to select S_3 and so on. The selections are done according to a "transition probability" $T(S_j \leftarrow S_i)$. The sequence of states thus obtained is a Markov chain (the probability of picking any state depends only on the predecessor of that state), and if the transition rate T satisfies suitable conditions, the long-time distribution of states approaches $P(S)$.

The conditions required are (i) Ergodicity, and (ii) General balance.

Ergodicity means that, using the transition rule, starting from any state in state space, it is possible to reach any other state. With a poorly chosen transition rule that breaks ergodicity, some states may never be reached at all.

General balance is simply the condition that the limiting probability distribution $P(S)$ should remain invariant on applying the transition T:

$$\sum_{S'} P(S')T(S \leftarrow S') = P(S). \tag{4}$$

In other words, if we start with a distribution of states, and then change every state according to T, the distribution should remain unchanged. It can be shown that if instead we start with a single state, and repeatedly apply a transition T whose stationary distribution is P and which is ergodic, the chain of states thus obtained will have a distribution approaching P in the infinite-time limit.

In practice, general balance is a difficult condition to impose, so a stricter (but more enforceable) condition, "detailed balance", is preferred: for any two states S and S',

$$P(S')\,T(S \leftarrow S') = P(S)\,T(S' \leftarrow S) \tag{5}$$

(Summing this equation over S' yields the general balance criterion; detailed balance is thus sufficient but not necessary.)

Average quantities may be computed by maintaining a running total for the quantity of interest over a large number of steps, and finally dividing by the total number of steps. Optimal states may be reached by "simulated annealing", where $P(S)$ is replaced by $P(S)^{1/T}$ where T is a fictitious temperature, and is slowly lowered to zero. ("Quenching", or setting $T = 0$ directly, tends to get the system stuck in local minima, and usually fares poorly in finding global minima.)

3.2 *Gibbs sampling*

The art of MCMC lies in choosing $T(S' \leftarrow S)$, the transition rule or "moveset". Typically, S can be further parametrised by other variables, $S = (x_1, x_2, \ldots)$ (let us assume here that the variables are discrete), and a move consists of picking one of these variables and altering it. The most common choice is the Metropolis algorithm, where one variable, say x_i, is picked at random, and is randomly altered to a new value x_i' (so that the new state $S' = (x_1, x_2, \ldots, x_i', \ldots)$. Then if $P(S') > P(S)$, the move is accepted. Otherwise, the move is accepted with a probability $P(S') / P(S)$. It is easily verified that this move satisfies detailed balance; moreover, in many cases it is computationally very efficient (requiring only one function evaluation, for $P(S')$), and is thus a very popular algorithm.

In the motif-finding case, however, $P(S)$ tends to have high values only for a very few configurations, and is low for the vast majority of phase space. Therefore, Metropolis sampling tends to converge very slowly (the majority of moves either get rejected, or get accepted but do not greatly improve the configuration), and an alternative approach, Gibbs sampling, is used. In Gibbs sampling, one variable (say x_i) is picked at random, and then every possible new value x_i' is considered for that variable; a new value is selected from all these possibilities, according to the probabilities $P(S)$ for the new states. In other words, suppose there are N variables, the i'th variable x_i is picked, and the new value for this variable is x_i, with a corresponding state $S' = (x_1, x_2, \ldots, x_i, \ldots)$. Then,

$$T(S' \leftarrow S) = \frac{1}{N} \frac{P(S')}{\sum_{S''} P(S'')}, \tag{6}$$

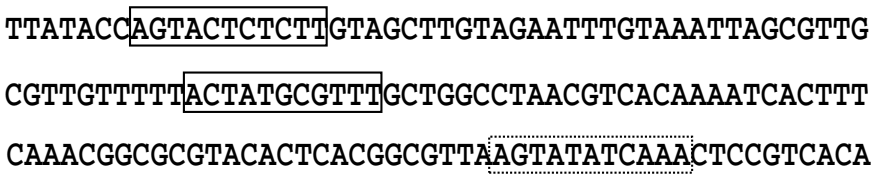

Figure 1. Putative binding sites are represented by "windows" (boxes). At each step of the Gibbs sampler, one binding site is selected and removed (dotted box), and a replacement site is selected, from all possible choices, according to the posterior probabilities of those choices.

where the 1/*N* prefactor reflects the probability of selecting the *i* th variable (x_i), and *S′* are all the possible states from changing x_i) (including the original state *S*). Substituting this into equation (5), we see that detailed balance is immediately satisfied.

Typically, x_i can have many different values (a few thousand in typical applications in biological motif finding), so each step of Gibbs sampling is a few thousand times slower than a Metropolis step would be. But convergence per step is, in practice, much faster in biological motiffinding problems.

3.3 *PhyloGibbs*

Given a sequence *S* of DNA, whose *n*th base is s_n, the hypothesis is that some sites are binding sites for transcription factors TFs, while the remaining sequence is represented by a background model.

Let a "configuration" *C* be a particular selection of motif sites, the boxes in figure 1. In PhyloGibbs terminology, we call them "windows". Their width, in PhyloGibbs, must be specified *a priori;* 8–14 is usually a reasonable value for eucaryotes, and 16–25 for procaryotes. Given a weight matrix w_{i,α_i} and a background model b_α, we can calculate the probability ("likelihood") of the sequence that we have, that is, the probability that the motif sites were drawn from the PWM and the other bases were drawn from the background model:

$$P(S|C) = P(\text{windows from WM}) \times P(\text{rest from background})$$

$$= \prod_{n \text{ in window}} w_{n-n_0,S_n} \prod_{n \text{ not in window}} b_{S_n}, \quad (7)$$

where n_0 is the starting base of the window in which *n* appears.

If we do not already know the PWM w, we integrate over the space of all possible w's, column by column, with the provision that each column sums to 1. For each column i and each window, the integrand will contain a factor $w_{i,\alpha}$ where α is the base at position i in that window. This yields a monomial $\prod_\alpha w_{i\alpha}^{n_\alpha}$, where n_α is the total number of bases α at position *i* in all windows, and $\Sigma_\alpha n_\alpha = N$ is the total number of windows. This needs to be integrated over w with the constraint $\Sigma_\alpha w_{i,\alpha} = 1$. This can be done exactly:

$$\int_w \prod_a w_{i,a}^{n_a} = \frac{3!\prod_a n_a!}{(N+3)!}. \quad (8)$$

Thus, $P(S \mid C)$ can be calculated for any configuration *C*; but what we really require is $P(C|S)$, the posterior probability of the configuration *C* given the sequence that we are provided, *S*. Bayes' Theorem supplies the answer, in terms of the "prior probability" of a configuration $P(C)$:

$$P(C|S) = \frac{P(S|C)P(C)}{\sum_{C'} P(S|C')P(C')}. \quad (9)$$

This, if the prior probability $P(C)$ is constant, is proportional to $P(S \mid C)$. Typically, we take $P(C)$ to be constant within some constraints – for example, the number of different types of motifs, and the number of binding sites, could be fixed – but other choices are possible, and are in fact tremendously valuable in making use of prior information.

Let us assume, for simplicity, that the total number of windows (sites), and the maximum number of colours (different kinds of motifs), are fixed. Then the Gibbs sampling proceeds as follows: An initial configuration satisfying these constraints is selected at random. Then, at each step, one window is selected at random, removed, and a replacement (in any allowed colour) is sampled. That is, all possible replacement windows (including the window that was removed), with all possible colours (including its original colour), are considered, their posterior probabilities calculated, and the new window and colour picked from this posterior probability distribution. This is the move described in § 3.2, with a state being parametrised by the position and colour of each window. It is ergodic and satisfied detailed balance, and therefore, if repeated long enough, each configuration *C* in the configuration space will be visited with a frequency proportional to $P(C \mid S)$.

In practice, one other move is needed for speedier convergence. Single-window moves like the above can get stuck in minima where an entire set of sites is selected with an "offset" (that is, they all partially cover the true binding sites and need to all be shifted by a fixed amount for an optimal answer). Such an offset will take very long

to remove with single window shifts (though it will happen eventually), so we include a "global shift" move, where a colour is chosen at random and then all possible "global shifts" for all windows in that colour are sampled.

PhyloGibbs includes other moves, and there are various subtleties relating to how these moves are applied; for a discussion of those details, the reader is referred to our original paper (Siddharthan *et al* 2005).

3.4 *Tracking: significance assessment*

A simulated anneal obtains an optimal set of states, and "tracking" (prolonged sampling, and measuring, for each possible site, how often it is co-clustered with one of the optimal sites) provides a significance assessment.

At the end of the anneal, the annealed colours are set up as "labelled lists" of windows to track. Then, for every window w in the system, and for every labelled list A, we set up a "tracking counter" $N(w, A)$ to see how often it's co-clustered with that list.

Then, at each time step, for each labelled list A, we associate one of the current colours with that list, $C(A)$. This is that colour which presently has the greatest presence of windows from that labelled list. (It need not correspond to the colour that originally gave rise to that labelled list.) For every window in that colour, we update its tracking counter: $N(w, A) \leftarrow N(w, A) + 1$ for each $w \in C(A)$.

At the end of the tracking phase (a predetermined, large number of moves), we divide each $N(w, A)$ by number of timesteps, and for every label A, sort all windows in order of the tracking score. This gives a measure of significance for how often each window was co-clustered with that labelled list.

The tracking phase can thus pick up sites that were not found in the simulated anneal (for example, because of a poor guess of initial parameters).

3.5 *Incorporating phylogenetic information*

For many organisms today, sequences of several closely related organisms are also available. One motivation for developing PhyloGibbs was to take advantage of such information. The assumption is that the organisms are sufficiently closely related that the regulation of genes has not diverged significantly. But with such close relatives, much orthologous sequence is highly conserved. It is then important to score binding sites in conserved sequence correctly: a site occurring in a conserved block in five species should not be considered as five independent instances.

As a pre-processing step, PhyloGibbs uses a multiple sequence alignment program to identify conserved blocks. [Any program with multi-fasta output will do. We originally used Dialign (Morgenstern 1999), and more recently this author has developed a special-purpose program for non-coding sequence, Sigma (Siddharthan 2006).

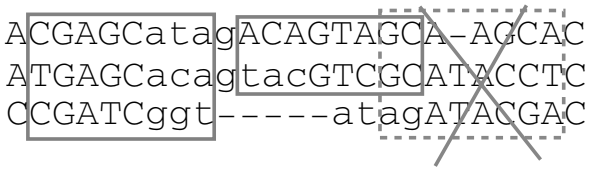

Figure 2. With phylogeneticallyrelated sequence, a multiple alignment program is used as a preprocessing step. Vertically aligned bases are treated as descendants from a common ancestor. "Windows" (boxes in the figure) can now encompass multiple sites descended from a common ancestor. The alignment within a window must be consistent: gaps within a window aren't allowed (dashed box). Lowercase letters are not treated as aligned, and can be exchanged with adjacent gaps.

We then assume that all bases inside conserved blocks (as pointed out by the alignment program) arose from an ancestral sequence and are not independent. PhyloGibbs generalises a "window" from a single binding site to a binding site in an aligned block (figure 2), thus possibly encompassing multiple species. We then modify the scoring for multi-sequence windows, while scoring single-sequence windows as before, and then sample on posterior probabilities as before.

The scoring is as follows: let us assume a "star topology" here, where all sequences descended independently from one common ancestor (as described in Siddharthan *et al* 2005, this can be generalized to arbitrary phylogenetic trees).

We assume a mutation rate m_i for species i, and a divergence time t from the common ancestor. Then the probability of a given base being conserved from its ancestor (its "proximity" to the ancestor) is $e^{-m_i t} = q_i$, and the probability that it's mutated is $1 - q_i$.

Next, we define a "transition probability" $T(\alpha_i \mid a)$ that an ancestor a evolved into a base α_i within a binding site represented by w:

$$T(\alpha_i \mid a, q_i) = [\delta_{a\alpha_i} q_i + (1-q_i) w_{\alpha_i}] \tag{10}$$

This rule says that if the base is not mutated (probability q_i), it must be the same as the ancestral base, while if it has mutated (probability $1 - q_i$), it has had sufficient time for "fixation" and is therefore a sample from the same weight matrix w. This has appropriate limits as $q \to 0$ and as $q \to 1$, and, on inserting an intermediate unknown ancestor b, has the correct multiplication rule:

$$\sum_b T(\alpha_i \mid b, q_1) T(b \mid a, q_2) = T(\alpha_i \mid a, q_1 q_2). \tag{11}$$

Then, the probability that all four bases in a column of a window W evolved from a common ancestor a and are

represented by a WM w is

$$P(W|w) = \sum_{a=A,C,G,T} w_a \prod_{i=1}^{N} T(\alpha_i | a, q_i). \quad (12)$$

For each column, we multiply such a factor for all windows of a colour; we multiply over all columns, and integrate over the space of weight matrices w as before, to obtain the posterior probability of the configuration. For the background probability, we replace w by the background probability b and do not integrate.

Unlike in the single-sequence case, the integral here is over a complicated polynomial; each monomial term can still be done exactly, but the number of terms rapidly increases with increasing number of windows, so in practice we use an approximation (described in Siddharthan *et al* 2005).

Apart from more accurate scoring, this approach to incorporating phylogeny has one other advantage: it greatly reduces the size of the configuration space to be searched. We find that the improvement in performance is significant, compared to naively running on all sequences without considering their phylogeny.

3.6 *Enhancements in PhyloGibbs-MP*

PhyloGibbs-MP (Siddharthan R, un-published results) is an update to PhyloGibbs that extends the capabilities of the program beyond simple motif-finding:

- It can localise predictions to short subsequences (meant to model *cis*-regulatory modules) of longer input sequence, and is successful at predicting CRMs *ab initio*
- It can make use of prior information, in the form of position weight matrices for known transcription factors, to improve prediction of CRMs and binding sites
- It can predict sites that appear preferentially in one set of input motifs and not in others ("differential motif finding")
- It optionally produces annotations compatible with the Generic Genome Browser (Stein *et al* 2002) for easy visualisation of the output
- It uses an importance sampling scheme to obtain a substantial speed increase

These enhancements, and performance, are discussed in an upcoming paper (R Siddharthan, unpublished results).

4. Summary and other issues

We have discussed a widely used Bayesian approach to motiffinding, namely Gibbs sampling, from the perspective of our program PhyloGibbs. We have glossed over many practical subtleties that PhyloGibbs takes account of; these are discussed in Siddharthan *et al* (2005) and Siddharthan and van Nimwegen (2007). The great advantage of a Bayesian approach is that prior information can easily be incorporated into the scoring scheme. The (somewhat trivial) example of constraints on the number of factors and the number of sites, and the example of known weight matrices as "informative priors", have already been mentioned. Other kinds of prior information may be available, such as high-throughput gene expression (microarray) data that tells us what genes are likely to be co-regulated. PhyloGibbs could usefully be extended to use such data (at present, one needs to extract gene clusters from microarray data beforehand, and feed those manually to PhyloGibbs).

Using the model that we have described, where binding sites are represented by position weight matrices and the background represented by a Markov model, existing algorithms deal reasonably well with real-world genomic data. However, there is much scope for improvement. We conclude by pointing out a few issues.

First, real DNA isn't actually well represented by a Markov model: it has significant long-ranged correlations, whose origin remain poorly understood. Understanding this should greatly benefit the task of motif finding (and finding other features in DNA).

Second, the position weight matrices that we use are assumed to have independent columns. This is partly because we don't have enough data to do better: even obtaining good "dinucleotide" weight matrices (which give the probability of seeing particular dinucleotides, rather than nucleotides, at particular positions) would take scores of known binding sites, a luxury usually not available. It works reasonably well in practice; nevertheless, it has been attacked (Djordjevic *et al* 2003) in the past, and improving the representation of binding sites should pay dividends.

Third, chromatin remodelling plays a major role in transcriptional regulation: the regulatory DNA, as well as the coding DNA, should be made accessible to the transcriptional machinery. This again is poorly understood, but progress is being made on predicting the positions of nucleosomes in chromatin, and there are suggestions that these positions correlate (or anticorrelate) with known binding sites in yeast (Segal *et al* 2006). A better understanding of such issues will undoubtedly improve our understanding of regulatory genomics.

Acknowledgements

PhyloGibbs was developed in collaboration with Eric Siggia and Erik van Nimwegen, and a significant part of the work was done at The Rockefeller University, New York in Eric

Siggia's group. The new features in PhyloGibbs-MP were developed by me, but motivated by the requirements of ongoing collaborations with various people, in particular K VijayRaghavan, K G Guruharsha, Bhaskar Saha, all of whom offered useful inputs and suggestions.

References

Amir A, Lewenstein M and Porat E 2004 Faster algorithms for string matching with *k* mismatches; *J. Algorithms* **50** 257–275

Bailey T L and Elkan C 1994 Fitting a mixture model by expectation maximization to discover motifs in biopolymers; *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **2** 28–36

Berman B P, Nibu Y, Pfeiffer B D, Tomancak P, Celniker S E, Levine M, Rubin G M and Eisen M B 2002 Exploiting transcription factor binding site clustering to identify *cis*-regulatory modules involved in pattern formation in the *Drosophila* genome; *Proc. Natl. Acad. Sci. USA* **99** 757–762

Berman B P, Pfeiffer B D, Laverty T R, Salzberg S L, Rubin G M, Eisen M B and Celniker S E 2004 Computational identification of developmental enhancers: conservation and function of transcription factor binding-site clusters in *Drosophila melanogaster* and *Drosophila pseudoobscura*; *Genome Biol.* **5** R61

Dermitzakis E T, Bergman C M and Clark A G 2003 Tracing the evolutionary history of drosophila regulatory regions with models that identify transcription factor binding sites; *Mol. Biol. Evol*. **20** 703–714

Djordjevic M, Sengupta A M and Shraiman B I 2003 A biophysical approach to transcription factor binding site discovery; *Genome Res*. **13** 2381–2390

Emberly E, Rajewsky N and Siggia E D 2003 Conservation of regulatory elements between two species of drosophila; *BMC Bioinformatics* **4** 57

He L and Hannon G J 2004 MicroRNAs: small RNAs with a big role in gene regulation; *Nat. Rev. Genet.* **5** 522–531

Lawrence C E, Altschul S F, Boguski M S, Liu J S, Neuwald A F and Wootton J C 1993 Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment; *Science* **262** 208–214

Lettice L A, Heaney S J H, Purdie L A, Li L, de Beer P, Oostra B A, Goode D, Elgar G, Hill R E and de Graaff E 2003A long-range Shh enhancer regulates expression in the developing limb and fin and is associated with preaxial polydactyly; *Hum. Mol. Genet.* **12** 1725–1735

Matzke M A and Birchler J A 2005 RNAi-mediated pathways in the nucleus; *Nat. Rev. Genet.* **6** 24–35

Morgenstern B 1999 DIALIGN 2: improvement of the segmenttosegment approach to multiple sequence alignment; *Bioinformatics* **15** 211–218

Pearson H 2006 Genetics: what is a gene?; *Nature (London)* **441** 398–401

Pierstorff N, Bergman C M and Wiehe T 2006 Identifying *cis*-regulatory modules by combining comparative and compositional analysis of DNA; *Bioinformatics* **22** 2858–2864

Sagot M-F 1998 Spelling approximate repeated or common motifs using a suffix tree; in *Latin 98, lecture notes in computer science* (Springer-Verlag) vol. 1380, pp 111–127

Segal E, Fondufe-Mittendorf Y, Chen L, Thastrom A, Field Y, Moore I K, Wang J-P Z and Widom J 2006 A genomic code for nucleosome positioning; *Nature (London)* **442** 772–778

Siddharthan R, Siggia E D and van Nimwegen E 2005 Phylogibbs: A gibbs sampling motif finder that incorporates phylogeny; *PLoS Comput. Biol*. **1** e67

Siddharthan R 2006 Sigma: multiple alignment of weakly-conserved non-coding DNA sequence; *BMC Bioinformatics* **7** 143

Siddharthan R and van Nimwegen E 2007 Detecting regulatory sites using phylogibbs; in *Comprehensive genomics*, *methods in molecular biology*. (ed.) N H Bergman (Humana Press) (in press)

Sinha S, Liang Y and Siggia E 2006 Stubb: a program for discovery and analysis of *cis*-regulatory modules; *Nucleic Acids Res*. **34** 555–559

Sinha S, Schroeder M D, Unnerstall U, Gaul U and Siggia E D 2004 Cross-species comparison significantly improves genome-wide prediction of *cis*-regulatory modules in *Drosophila*; *BMC Bioinformatics* **5** 129

Sinha S, van Nimwegen E and Siggia E D 2003 A probabilistic method to detect regulatory modules; *Bioinformatics* (*Suppl. 1*) **19** 292–301

Smith, A F M and Roberts G O 1993 Bayesian computation via the gibbs sampler and related markov chain monte carlo methods; *J. R. Stat. Soc. Series B (Methodological)* **55** 3–23

Stein L D, Mungall C, Shu S Q, Caudy M, Mangone M, Day A, Nickerson E, Stajich J E, Harris T W, Arva A and Lewis S 2002 The generic genome browser: a building block for a model organism system database; *Genome Res.* **12** 1599–1610

Tanay A, Regev A and Shamir R 2005 Conservation and evolvability in regulatory networks: the evolution of ribosomal regulation in yeast; *Proc. Natl. Acad. Sci. USA* **102** 7203–7208

Ukkonen E 1995 Online construction of suffix trees; *Algorithmica* **14** 249–260

*e*Publication: 5 July 2007