BMC Bioinformatics

BioMed Central

Review

Finding regulatory elements and regulatory motifs: a general probabilistic framework

Erik van Nimwegen

Address: Biozentrum, University of Basel, and Swiss Institute of Bioinformatics, Klingelbergstrasse 50/70, Basel, Switzerland

Email: Erik van Nimwegen - erik.vannimwegan@unibas.ch

Published: 27 September 2007

BMC Bioinformatics 2007, **8**(Suppl 6):S4 doi:10.1186/1471-2105-8-S6-S4

This article is available from: http://www.biomedcentral.com/1471-2105/8/S6/S4

Abstract

Over the last two decades a large number of algorithms has been developed for regulatory motif finding. Here we show how many of these algorithms, especially those that model binding specificities of regulatory factors with position specific weight matrices (WMs), naturally arise within a general Bayesian probabilistic framework. We discuss how WMs are constructed from sets of regulatory sites, how sites for a given WM can be discovered by scanning of large sequences, how to cluster WMs, and more generally how to cluster large sets of sites from different WMs into clusters. We discuss how 'regulatory modules', clusters of sites for subsets of WMs, can be found in large intergenic sequences, and we discuss different methods for *ab initio* motif finding, including expectation maximization (EM) algorithms, and motif sampling algorithms. Finally, we extensively discuss how module finding methods and *ab initio* motif finding methods can be extended to take phylogenetic relations between the input sequences into account, i.e. we show how motif finding and phylogenetic footprinting can be integrated in a rigorous probabilistic framework. The article is intended for readers with a solid background in applied mathematics, and preferably with some knowledge of general Bayesian probabilistic methods. The main purpose of the article is to elucidate that all these methods are not a disconnected set of individual algorithmic recipes, but that they are just different facets of a single integrated probabilistic theory.

The weight matrix representation of regulatory sites

The first step in any algorithm for identifying regulatory sites in DNA or RNA is to decide on a mathematical representation of the binding sites. For definiteness, let us assume we are considering a DNA binding factor which, when bound to DNA, covers a DNA segment of l base pairs long. For any length-l sequence s there will be a well-defined (but generally unknown) binding free-energy $E(s)$ to the regulatory factor. A key assumption [1] that is introduced at this point is that the energy $E(s)$ can be written as the sum of independent contributions $E_i(s_i)$ from each of the bases s_i in segment s, i.e.

$$E(s) = \sum_{i=1}^{l} E_i(s_i). \quad (1)$$

This assumption of course generally only holds to some extent. Large-scale *in vitro* studies have shown that the binding energies can deviate from this simple additivity assumption [2]. However, these deviations are typically small, and moreover they seem generally restricted to segments with low binding-energy [3]. At this point it is not yet clear to what extent and for what fraction of regulatory factors, the additivity assumption holds. Some researchers believe that, at least for some factors, functional binding

sites deviate significantly from this assumption, and this may well be the case. However, it is this author's experience that in collections of experimentally determined binding sites there is little evidence of correlations between the nucleotides occurring at different positions which, as we will see below, supports the additivity assumption for functional binding sites.

The crucial assumption which underlies the whole idea of 'finding regulatory sites' is that the set of all 4^l possible segments s can be meaningfully divided into 'binding sites' and all other sequences. Since this is not *a priori* clear at all, it is good to consider what this assumption entails. At a given concentration c of the regulatory factor, the probability that a sequence segment s will be bound by the factor is given by an expression of the following form [4,5]

$$P_{\text{bound}}(s) = \frac{ce^{\beta E(s)}}{ce^{\beta E(s)} + K}, \qquad (2)$$

where $\beta = 1/(kT)$ is the inverse temperature, and K is a constant. (We here ignore the fact that the factor may bind at segments that overlap s, which would prevent the factor from binding at s. Below we will derive the general solution that takes this complication into account.) The expression (2) is an s-shaped function that goes from 0 to 1 as $ce^{\beta E(s)}$ goes from much smaller than K to much larger than K. Therefore, at a given concentration c one can naturally separate sequences s into binders, i.e. those with $E(s) > \log(K/c)/\beta$ and non-binders with $E(s) < \log(K/c)/\beta$. If the concentration of the (active) regulatory factor were to vary continuously between different cellular states, then the set of sites bound by the factor would also vary continuously and it would not make much sense to divide segments s into binders and non-binders. However, if in physiological conditions the regulatory factor primarily switches between an 'off' state, i.e. low concentration c_{off}, and an 'on' state, i.e. high concentration c_{on} than there would be a well defined set of sites that are bound when the factor is 'on' and unbound when the factor is 'off', i.e. those with energies in the range

$$\frac{1}{\beta}\log\left(\frac{K}{c_{\text{on}}}\right) < E(s) < \frac{1}{\beta}\log\left(\frac{K}{c_{\text{off}}}\right). \qquad (3)$$

Therefore, this set of binding sites may be characterized by a typical energy that lies somewhere in the middle of this range.The assumption that is thus generally made [1] is that binding sites are characterized by an average binding energy $\bar{E}$. We now want to derive the probability $P(s)$ that a randomly chosen binding site will have sequence s, given only the constraint that the average energy of the sites is $\bar{E}$. The maximum entropy formalism [6], i.e. as applied in statistical mechanics, prescribes that distribution $P(s)$ is given by

$$P(s) = \frac{e^{\lambda E(s)}}{\sum_{s'} e^{\lambda E(s')}} = \prod_{i=1}^{l}\left[\frac{e^{\lambda E_i(s_i)}}{\sum_{\alpha} e^{\lambda E_i(\alpha)}}\right], \qquad (4)$$

where the sum over s' is over all length-l sequences, and the sum over α is over the four bases. The Langrangian multiplier λ is chosen such that $\langle E \rangle = \sum_s E(s)\, P(s) = \bar{E}$. Note that this is the same functional form as the well-known Boltzmann distribution. To avoid confusion, note also that equations (2) and (4) are probability distributions over entirely different spaces. The former takes a fixed sequence segment s and compares the probabilities of the bound and unbound states for this sequence segment, whereas the latter assigns the probabilities that a binding site will take on any of the 4^l possible sequences. In equation (4) the bases at different positions are independent, i.e. $P(s) = \prod_{i=1}^{l} P_i(s_i)$ with

$$P_i(s_i) = \frac{e^{\lambda E_i(s_i)}}{\sum_{\alpha} e^{\lambda E_i(\alpha)}}. \qquad (5)$$

This property allows us to define a *position specific weight matrix* (WM) w with components

$$w_{\alpha}^{i} = \frac{e^{\lambda E_i(\alpha)}}{\sum_{\alpha'} e^{\lambda E_i(\alpha')}}. \qquad (6)$$

That is, we can represent regulatory sites by WMs, and find the following expression for the probability that a binding site has sequence s:

$$P(s \mid w) = \prod_{i=1}^{l} w_{s_i}^{i}. \qquad (7)$$

Finally, note that $P(s|w)$ gives the probability that a given binding site will have sequence s, which should be carefully distinguished from the probability $P(w|s)$ that a sequence segment s is a binding site for w. The latter cannot be calculated without specifying how likely s is to arise under alternative hypotheses as will be discussed in detail below.

Weight matrices are probably the most commonly used representation of regulatory sites and, as has just been shown, can be derived under the assumptions that the contribution to the binding energy from bases at different positions in the site are independent, and that functional

binding sites are characterized by a given average binding energy. In this chapter we will focus on regulatory motif finding methods that use WMs. It should be noted, however, that in some circumstances regulatory sites can be adequately represented by either specific DNA words, i.e. when the regulatory factor recognizes essentially only a single sequence segment, or by regular expressions, and there is a substantial amount of work on motif finding in this context. There is also a moderate amount of work on more complex representations of regulatory sites, such as hidden Markov models that allow sites of varying length and correlations between bases at neighboring positions [2,7].

Finding WM matches

Assume that we are in possession of a WM w that summarizes the binding specificity of a regulatory factor. One of the simplest applications is to 'scan' one or more sequences for 'matches' to this WM. Let s denote some sequence of length L, where L is typically much larger than the length l of the WM. We now want to infer if one or more sites for this WM occur in this sequence. Probabilistic inference always [6] takes the following general form

1. Enumerate all possible hypotheses H that could have accounted for the data D.

2. Assign prior probabilities $P(H)$ to each of these hypotheses.

3. Define a likelihood model that gives the probability $P(D|H)$ of producing the entire data D under each of the hypotheses H.

4. The posterior probability $P(H|D)$for each of the hypotheses is then given by Bayes' theorem:

$$P(H \mid D) = \frac{P(D \mid H)P(H)}{\sum_{\tilde{H}} P(D \mid \tilde{H})P(\tilde{H})}. \quad (8)$$

For example, assume that we have prior information that precisely one site for WM w occurs in s and that the other bases in s were drawn from a background model b. For simplicity we will assume that under this background model b, each letter has a probability b_α to be base α. In this situation all possible hypotheses are simply all possible locations i at which the binding site might start. If we have no information to suggest that the site is more likely to occur at some places than others we use an uniform prior $P(i)$ = constant. The likelihood $P(D|i)$ of the data, i.e. sequence s, given the corresponding hypothesis is given by the product of probabilities that the bases from 1 up to i derive from the background model, that the segment from $i + 1$ through $i + l$ derives from the WM w, and that bases $i + l + 1$ through L again derive from the background model.

The probability $P(D|i)$, as illustrated in Fig. 1, is given by

$$P(D|i) = P(s_{[0,i]}|b)P(s_{[i,l]}|w)P(s_{[i+l,L-i-l]}|b), \quad (9)$$

where $s_{[i,l]} = s_{i+1}s_{i+2}...s_{i+l}$ is the length-l segment in s starting after position i with

$$P(s_{[i,l]} \mid w) = \prod_{k=1}^{l} w^k_{s_{i+k}}, \quad (10)$$

and the background probabilities are given by

$$P(s_{[0,i]} \mid b) = \prod_{k=1}^{i} b_{s_k}$$
$$P(s_{[i+l,L-i-l]} \mid b) = \prod_{k=i+l+1}^{L} b_{s_k}. \quad (11)$$

With the uniform prior, the posterior probability $P(i|D)$ that the site occurs at i is

$$P(i \mid D = \frac{P(D \mid i)}{\sum_{j=0}^{L-l} P(D \mid j)}. \quad (12)$$

In general we of course do not know that there is precisely one site in s. Therefore, we generally want to consider the extended set of hypotheses that consists of all possible configurations of binding sites that can be assigned to sequence s. Figure 2 shows a possible configuration containing 3 hypothesized binding sites.

Generally, each possible configuration with n sites can be denoted by a vector $i = (i_1, i_2,...,i_n)$ which denotes the positions at which the binding sites occur. The probability of the data given a configuration i is now given by

$$P(D \mid i) = \left[\prod_{\sigma \in B_i} b_\sigma \right] \prod_{s \in S_i} P(s \mid w), \quad (13)$$

where B_i is the set of background bases and S_i is the set of hypothesized sites in configuration i.

To assign prior probabilities $P(i)$ to all possible configurations one generally assumes that the data D was produced through a stochastic process where at each step with probability $(1 - \pi)$ a single background base is emitted, and with probability π a length-l binding site is emitted. Under this model the prior probability $P(i)$ for a configuration i

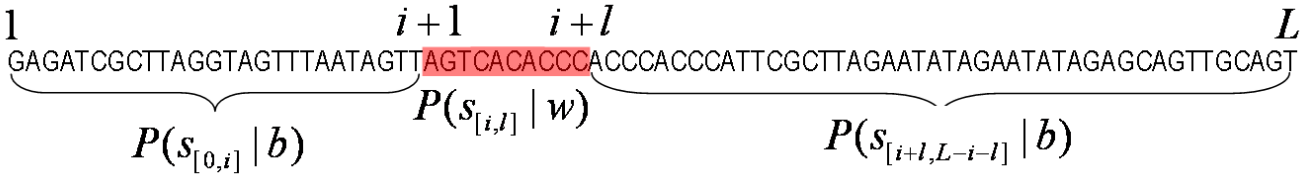

Figure 1
A dataset *D* consisting of a single sequence *s* of length *L*, with a single site hypothesized immediately after position *i*.

depends only on the number of sites $n(i)$ that occurs in the configuration, and is given by

$$P(i) \propto \pi^{n(i)} (1 - \pi)^{L-ln(i)} \quad (14)$$

Using this the posterior probability of configuration i given the data becomes

$$P(i \mid D) = \frac{P(D \mid i)\pi^{n(i)}(1-\pi)^{L-ln(i)}}{\sum_j P(D \mid j)\pi^{n(j)}(1-\pi)^{L-ln(j)}}, \quad (15)$$

where the sum in the denominator is over all possible binding site configurations j.

Even though the total number of configurations grows faster than exponential with the sequence length L, the sum in the denominator can be easily calculated using dynamic programming as follows. Let F_n denote the sum of the likelihoods of all configurations up to position n in s. We have the recurrence relation

$$F_n = F_{n-1}(1 - \pi)\ b_{s_n} + F_{n-l}\pi P(s_{[n-l,\ l]}|w), \quad (16)$$

as illustrated in Fig 3.

Notice that the sum over all configurations is just F_L, i.e. $F_L = \sum_j P(D|j)P(j)$, which can be calculated in a time $O(L)$ using the above recurrence relation. Similarly, we can move backward from the end of the sequence to have a recurrence relation for the sum of likelihoods of all configurations of positions n through L of s:

$$R_n = b_{s_n}\ (1 - \pi)R_{n+1} + P(s_{[n-1,\ l]}|w)\pi R_{n+l}. \quad (17)$$

Finally, instead of calculating the posterior $P(i|D)$ for a particular configuration i, we can also calculate the posterior probability that a site occurs at a given position, independent of the rest of the configuration. Let us denote by $\{n\}$ the set of all configurations that have a site at segment $s_{[n,l]}$ .The posterior probability $P(\{n\}|D)$ is given by the sum of posterior probabilities of all configurations in $\{n\}$, i.e.

$$P(\{n\} \mid D) = \sum_{i \in \{n\}} P(i \mid D). \quad (18)$$

It is easy to see that this sum can be expressed in terms of F_n and R_n as follows:

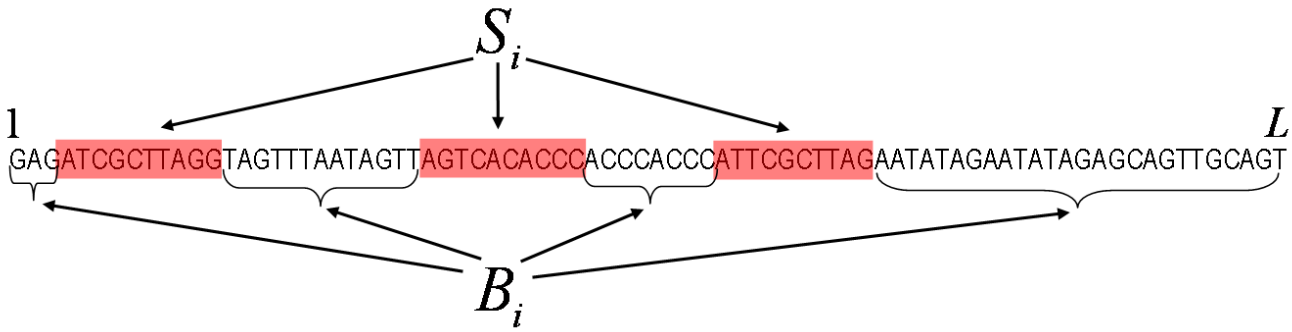

Figure 2
A configuration *i* with 3 hypothesizes sites. S_i denotes the set of hypothesized sites and B_i the background bases.

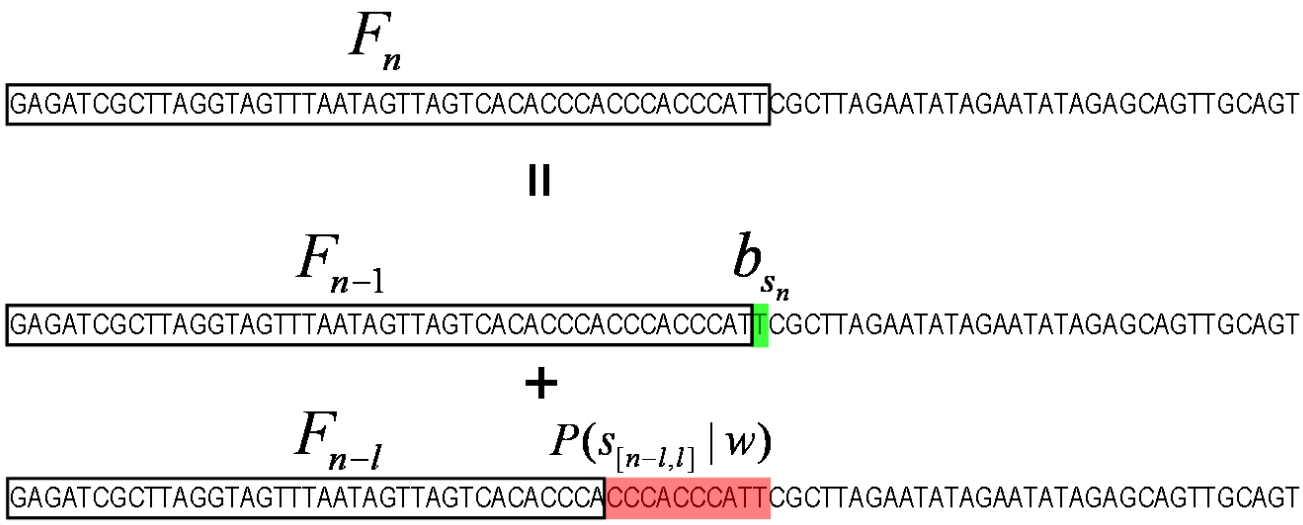

Figure 3
Illustration of equation (16). The black rectangle indicates the sum F_n of probabilities $P(D|i)$ for all binding site configurations i for the sequence within the rectangle. Any configuration in F_n is obtained either through adding a single background base at n to any of the configurations in F_{n-1}, or by adding a site from $n - l + 1$ through n to any configuration in F_{n-l}.

$$P(\{n\} \mid D) = \frac{F_n P(s_{[n,l]} \mid w) \pi R_{n+l+1}}{F_L}, \qquad (19)$$

where the numerator corresponds to the sum over all configurations that have a site at $s_{[n,l]}$.

Here it is useful to note that, formally speaking, the model that we have introduced is a hidden Markov model and that the expressions (16), (17), and (19) are essentially the same as the so-called forward-backward algorithms of hidden Markov model theory [8,9]. Researchers with a background in statistical physics tend to think of F_L as a *partition sum* and the recurrence relations are essentially what is known as the *transfer matrix* technique.

Given the WM w for a regulatory factor we may use equation (19) to scan any sequence s for positions at which functional binding sites for the factor are likely to occur. The likelihood of success of this procedure critically depends on the density of true sites in the input sequence s. That is, even in a sequence generated entirely from the background model, segments that are indistinguishable from binding sites will occur by chance at a certain rate. For example, let's assume that such chance 'binding site lookalikes' occur once every 500 bps on average, and assume that we are looking for between 1 and 3 functional binding sites in an intergenic region of length 250 which stems from a bacterial genome. In this case we expect less than 1 binding site to occur by chance, and so we will likely be able to accurately determine the location of the 1 to 3 functional sites. In contrast, assume we are looking for 1 to 3 functional sites in the introns and upstream regions of a human gene, which together might contain as many as 100,000 bps of non-coding DNA. It is clear that in this case the functional sites will 'drown' in a sea of about 200 binding site lookalikes.

At this point the reader may ask how *the cell* distinguishes functional binding sites from mere 'lookalikes'. Comparing equation (2) with (6) and (7), we see that $P_{\text{bound}}(s)$ can be written in terms of $P(s|w)$ and c. In other words, two segments s and s' for which $P(s|w) = P(s'|w)$ necessarily have $P_{\text{bound}}(s) = P_{\text{bound}}(s')$, and one may thus wonder why two segments that are equally likely to be bound by the regulatory factor are not equally functional. There are a number of reasons. First, in eukaryotic genomes DNA is wrapped up in chromatin and so different sites may have different accessibility to the regulatory factor. Second, binding of the regulator may by itself not guarantee functionality, i.e. a regulatory effect. A number of additional constraints typically have to be satisfied. The site may need to occur in the vicinity of specific other regulatory sites, e.g. to mediate interactions between different factors bound at the different sites. The site may need to occur at a particular distance from the basal promoter and in a particular orientation to be able to interact with the basal machinery, and other constraints currently not yet understood. When we want to look for functional sites in long sequences we thus generally have to use information that goes beyond the probabilities $P(s|w)$ of the individual sequence segments given individual WMs w. One type of

additional information that can be used is that in some cases functional binding sites are known to cluster on the genome. We now discuss approaches to incorporating this information.

Finding clusters of binding sites: regulatory modules

It has been well-established that in higher eukaryotic organisms transcription regulation is often implemented through regulatory 'modules' in which multiple binding sites for multiple regulatory factors cluster together relatively tightly in intergenic regions [10]. In some cases one may even know the subsets of regulatory factors that tend to cooperate in regulatory modules for particular biological pathways. For example, a large body of work has identified the sets of transcription factors that are involved in segmentation of the early Drosophila embryo, e.g. see [11].

One approach to distinguishing functional binding sites from nonfunctional ones is to look for such regulatory modules. That is, the idea is to start with a set of WMs $\{w\}$, preferably from a set of regulatory factors that are believed to interact in regulatory modules, and to look for relatively short genomic segments in which there is a surprisingly high density of sites for the WMs from $\{w\}$. As far as this author is aware, this general idea was introduced around the same time by a number of groups [12-15]. The implementation we discuss here is most closely-related to the approaches of refs. [14,15].

The first thing to note is that the dynamic programming solution introduced in the previous section can be easily extended to multiple WMs w (potentially of different lengths). We now assume that the data is produced through a stochastic process where at each step with probability π_{bg} a background base is generated, and with probability π_w a WM segment from WM w with length l_w is generated. The priors of course satisfy the normalization $\pi_{\mathrm{bg}} + \sum_w \pi_w = 1$. For notational simplicity we can consider the background b to just be one of the WMs (with length $l = 1$) in the set $\{w\}$. In this more general model the recurrence relation for F_n becomes

$$F_n = \sum_w F_{n-l_w} \pi_w P(s_{[n-l_w, l_w]} \mid w), \qquad (20)$$

where the background b is now one of the WMs w.

The second thing to note is that the sum over all configurations $F_L = \sum_c P(D|c)P(c|\{\pi_w\})$ is formally the likelihood of the data D under our entire set of hypotheses c, that is, it is the probability to obtain the data under the assumed stochastic model. Note that in this expression we have indicated explicitly that this probability depends on the priors $\{\pi_w\}$. The quantity F_L thus summarizes how well the sequence can be explained in terms of the set of WMs in the model. The basic idea of the regulatory module detecting algorithms in [14,15] is to identify putative regulatory modules with sequence segments that have a high value for the sum F_L of probabilities of all binding site configurations in the segment.

The procedure works as follows. One starts with an intergenic region upstream of a gene of interest in a higher eukaryotic genome. Such intergenic regions are typically quite large, i.e. from 10 Kbps in flies to over 100 Kbps in humans. One then slides a windows of length between 200 and 500 bps or so over this long intergenic region. For each window one then determines the set of priors $\{\pi_w\}$ that maximize F_L for the sequence σ in the window, and calculates the value of F_L at this maximum. One also calculates the probability $P(\sigma|b)$ for the sequence in the window deriving entirely from the background model. The ratio $X = F_L/P(\sigma|b)$ then quantifies the 'score' for the window in question. Finally, the predicted regulatory models are all windows for which X is larger than some prespecified cut-off, and for which the score X is larger than the score for any other window overlapping it.

A key step in this procedure is maximizing F_L with respect to the prior $\{\pi_w\}$. Different regulatory modules may have different densities of sites and we thus want to allow for different priors $\{\pi_w\}$ within different windows. Since we do not know the $\{\pi_w\}$ for each segment, from the point of view of probability theory one should strictly speaking not maximize with respect to $\{\pi_w\}$ but rather integrate over all possible priors $\{\pi_w\}$. However, the resulting expressions no longer allow for an effective dynamic programming solution and this would thus make the problem computationally intractable. However, if the function F_L has a sharp peak with respect to the $\{\pi_w\}$ then the height of the maximum is representative for the value of the integral and one can thus think of the maximization of F_L with respect to the $\{\pi_w\}$ as an approximation to doing the full integral.

Assuming that segment σ is of length L the set of equations specifying the maximum with respect to the $\{\pi_w\}$ are

$$\text{constant} = \frac{d\log(Z)}{d\pi_w} = \frac{d\log(F_L)}{d\pi_w} = \frac{\langle n(w)\rangle}{\pi_w}, \qquad (21)$$

where $\langle n(w)\rangle$ is the expected number of binding sites for WM w averaged over all configurations, each weighted by its probability. The last equation follows from the fact that, the prior $P(c|\{\pi_w\})$ is given by

$$P(c \mid \{\pi_w\}) = \prod_w (\pi_w)^{n(w,c)}, \qquad (22)$$

where $n(w, c)$ is the number of sites for w in configuration c. The derivative then becomes

$$\frac{d\sum_c P(\sigma \mid c)P(c \mid \{\pi_w\})}{d\pi_w} = \frac{1}{\pi_w}\sum_c n(w,c)P(\sigma \mid c)P(c \mid \{\pi_w\}). \quad (23)$$

Thus, from (21) and that fact that the π_w are normalized to sum to 1 we have

$$\pi_w = \frac{\langle n(w)\rangle}{\sum_{\tilde{w}}\langle n(\tilde{w})\rangle}. \quad (24)$$

Typically this maximum is found through *expectation maximization* (EM). Starting from an initial guess of the $\{\pi_w\}$ we calculate $\langle n(w)\rangle$ for all w and set a new set of priors $\{\pi_w\}$ using equation (24). Under iteration this is guaranteed to lead to an optimum in F_L, although not necessarily the global optimum.

Motif finding

Up to now we have assumed that we are in possession of the WMs w representing the sequence-specificities of the regulatory factors. However, unless one has experimental data that directly measures binding affinities of different sequence segments we generally do not possess such detailed information. Typically the best situation encountered is that we have a collection S of sequences that have been determined to be functional binding sites for the regulatory factor. So we now ask what we know about the WM w given such a set of sequences S, i.e. we aim to calculate $P(w|S)$.

Equation (7) gives the probability that a binding site for w will have sequence s. This can be trivially extended to *sets* of sequences. That is, the probability to obtain the set of n length-l sequences S when sampling n sequences from the WM w is given by

$$P(S \mid w) = \prod_{s\in S} P(s \mid w) = \prod_{i=1}^{l}\prod_{\alpha}(w_\alpha^i)^{n_\alpha^i(S)}, \quad (25)$$

where in the last equation we have defined $n_\alpha^i(S)$ as the number of times the letter α occurs at position i in the sequences S. Thus, the probability to obtain sequences S when sampling from the WM w depends only on the counts $n_\alpha^i(S)$.

Using Bayes' theorem the posterior probability $P(w|S)$ for the WM given the set of sites S is formally given by

$$P(w \mid S) = \frac{P(S \mid w)P(w)}{P(S)}. \quad (26)$$

In this equation $P(w)$ is the *prior* probability that the WM is given by w. The denominator is a normalizing constant, which does not depend on the WM (we discuss its meaning in a minute). The prior $P(w)$ represents our prior information about the WM w before we see any sites. As will become clear below, the computations are analytically most easily tractable if we use so-called Dirichlet priors that have the following general form

$$P(w) = \prod_{i=1}^{l} P(w_i) = \prod_{i=1}^{l} c_i \prod_{\alpha}(w_\alpha^i)^{\gamma_\alpha^i - 1}, \quad (27)$$

where c_i is a normalization constant for column i, and the γ_α^i are constants that determine the prior. Notice that for the particular choice $\gamma_\alpha^i = 1$ we obtain a *uniform* prior that makes all WMs a priori equally likely, which can be argued to reflect a state of complete ignorance about the WM. In reality, however, we know that for most positions in the site, regulatory factors tend to have distinct preferences for certain bases. That is, we a priori know that a WM column $w^i = (0.25, 0.25, 0.25, 0.25)$ is not very likely. To reflect this information we can choose $\gamma_\alpha^i < 1$. This will put more weight on WM columns that are 'skewed', i.e. giving low probability to some bases and high probabilities to others. Sometimes we have even more pertinent information. It has, for example, been argued recently that groups of related TFs show the same pattern of highly and less skewed columns [16]. If we are inferring the WM of such a TF we can thus reflect that information by setting γ_α^i small for those positions i that are known to be highly skewed and $\gamma_\alpha^i \approx 1$ for columns that are known not to be very skewed (for example because TFs of that family do not touch the DNA at that position).

With a Dirichlet prior of the form (27) equation (26) becomes

$$P(w \mid S) = C\prod_{i=1}^{l}\prod_{\alpha}(w_\alpha^i)^{n_\alpha^i(S)+\gamma_\alpha^i - 1}, \quad (28)$$

where C is an overall normalization constant. Equation (28) shows why the γ_α^i are often called pseudocounts. Increasing γ_α^i by 1 has the same effect on the posterior

$P(w|S)$ as adding 1 to the number of times $n_\alpha^i(S)$ that letter α was observed at position i. Put another way, the posterior $P(w|S)$ has exactly the same functional form as the prior $P(w)$, i.e. both are of the form $\prod_\alpha (w_\alpha)^{x_\alpha}$ with x_α the 'count' of base α. Priors that have this property are called *conjugate* priors. In this particular case it means that one may think of the posterior $P(w|S)$ as the prior for another problem with 'pseudocounts' $\tilde{\gamma}_\alpha^i = n_\alpha^i(S) + \gamma_\alpha^i$. How to use the distribution $P(w|S)$ in practice? In order to estimate the WM one could for example determine the WM w that maximizes $P(w|S)$. This maximum posterior probability WM has components

$$w_\alpha^i = \frac{n_\alpha^i(S) + \gamma_\alpha^i - 1}{n^i(S) + \gamma^i - 4}, \tag{29}$$

with $n^i(S) = \sum_\alpha n_\alpha^i(S)$ and $\gamma^i = \sum_\alpha \gamma_\alpha^i$. Note that with a uniform prior $\gamma_\alpha^i = 1$ the maximum occurs when the WM entries match the observed frequencies. This means, for example, that if a given base α is not observed at all at some position i, i.e. $n_\alpha^i(S) = 0$, we will assume that it is *impossible* for α to occur at position i. This is true even if the set S contains only very few sites.

Alternatively we may estimate the w_α^i by their expected values under the distribution $P(w|S)$. To calculate these expectation values we have to integrate $P(w|S)$ over all possible WMs. That is, for each position i the integral is over the simplex:

$$\sum_\alpha w_\alpha^i = 1, w_\alpha^i \geq 0 \forall \alpha. \tag{30}$$

The solution to such integrals is given by the following general identity

$$\int (w_1)^{x_1 - 1} \ldots (w_n)^{x_n - 1} dw_1 \ldots dw_n = \frac{\prod_{i=1}^n \Gamma(x_i)}{\Gamma(\sum_{i=1}^n x_i)}, \tag{31}$$

where the integral is over the simplex $\sum_{i=1}^n w_i = 1$. Using this identity we first find the normalization constant of equation (28). That is, by demanding that $\int P(w|S) dw = 1$ we obtain

$$\frac{1}{C} = \prod_{i=1}^l \frac{\prod_\alpha \Gamma(n_\alpha^i(S) + \gamma_\alpha^i)}{\Gamma(n^i(S) + \gamma^i)}, \tag{32}$$

and using this (plus the general identity $\Gamma(x + 1) = x\Gamma(x)$) we find for the expectation values

$$\langle w_\alpha^i \rangle = \int w_\alpha^i P(w \mid S) dw = \frac{n_\alpha^i(S) + \gamma_\alpha^i}{n^i(S) + \gamma^i}. \tag{33}$$

Note that in this estimate of the w_α^i no component gets probability zero if we use a prior with $\gamma_\alpha^i > 0$ for all i and α.

In the previous section we repeatedly made use of the expression $P(s|w)$, i.e. the probability to obtain sequence s when sampling from the WM. We now calculate an analogous expression $P(s|S) = \int P(s|w)P(w|S)dw$, which is the probability to obtain sequence s when sampling from the same WM as the one from which the set S derived (without ever specifying precisely what this WM is, i.e. we integrate over all possible w). Using again the general identity (31) we obtain

$$P(s \mid S) = \prod_{i=1}^l \frac{n_{s_i}^i(S) + \gamma_{s_i}^i}{n^i(S) + \gamma^i} = \prod_{i=1}^l \langle w_{s_i}^i \rangle. \tag{34}$$

That is, we find that $P(s|S)$ is precisely the probability that would be obtained from expression $P(s|w)$ when using the expectation values $\langle w_\alpha^i \rangle$ as an estimate for the WM w.

Up to now we assumed that we were given a set S of length-l sequences that were sampled from the WM. Except in cases where we have, for example DNase footprinting data that give the precise locations of the regulatory sites, such specific data are again generally rare. It is much more common that we have a set of n longer sequences that we know (or strongly suspect) to contain one (or more) regulatory site(s) each for a common regulatory factor. In this situation we simultaneously need to infer *where* in the sequences the sites occur and what the WM is from which they derive.

To be explicit, let's assume we have a dataset D that consists of n length-L sequences, and we know that each sequence contains precisely one binding site of length l for a common regulatory factor. The set of hypotheses for this problem then corresponds to all combinations (w,i) of a WM w and a vector $i = (i_1, i_2, \ldots i_n)$ that denotes the positions where the regulatory sites occur, i.e i_1 is the position

of the site in the first sequence, i_2 the position of the site in the second sequence, etcetera. We now first calculate the probability $P(D|w, i)$ of the data given (w, i). Let S_i denote the set of n length-l segments that make up the hypothesized binding sites with positions i and let B_i denote all background nucleotides in the data D *outside* of these segments. In analogy with equation (13) the probability $P(D|w, i)$ is then given by

$$P(D \mid w,i) = \left[\prod_{\sigma \in B_i} b_\sigma\right] \prod_{s \in S_i} P(s \mid w), \tag{35}$$

where the first product is over all nucleotides outside of the hypothesized binding sites, and the second product is over all hypothesized binding sites s.

At this point there are two possible approaches. In the first approach one calculates the probability $P(D|w)$ of the data given the weight matrix only by summing over all possible binding site configurations i:

$$P(D \mid w) = \sum_i P(D \mid w,i) P(i), \tag{36}$$

where $P(i)$ is a prior probability distribution over vectors of site assignments, and the sum is over all possible vectors. One then next searches the space of all possible WMs w for those with high $P(D|w)$. In the second approach one calculates the probability $P(D|i)$ of the data given the vector of site positions only by integrating over all possible weight matrices. Formally [6] this probability is given by

$$P(D|i) = \int P(D, w|i) dw = \int P(D|w, i) P(w) dw, \tag{37}$$

and next the set of all site positions i is searched for those with high $P(D|i)$. We now discuss these approaches in turn.

Maximizing P(D|w) *through Expectation Maximization*

In the first approach one attempts to find the weight matrix w that maximizes the probability of the data $P(D|w)$. Note that, as we have seen in section "Finding WM matches", the sum over all possible site configurations i can be easily performed through dynamic programming once the matrix w is given. For the particular case we are considering, i.e. assuming precisely one site per sequence, the probability $P(D|w)$ is given by the product of the probabilities for the individual sequences

$$P(D \mid w) = \prod_{m=1}^{n} P(D_m \mid w), \tag{38}$$

with

$$P(D_m \mid w) = \frac{1}{L-l+1} \sum_{i_m} \left[P(s_{[i_m,l]} \mid w) \prod_{\sigma \notin s_{[i_m,l]}} b_\sigma \right], \tag{39}$$

where D_m is the mth sequence, the product over σ is over all bases outside of the site (i.e. the background), and we have used the uniform prior $P(i_m) = 1/(L - l + 1)$ over the binding binding site position i_m.

To find the WM w that maximizes $P(D|w)$ we proceed analogously as we did for finding the set of priors $\{\pi_w\}$ in equations (21) through (24). For each column k of the WM we have the four equations

$$\text{constant} = \frac{d \log[P(D \mid w)]}{dw_\alpha^k} = \frac{\langle n_\alpha^k \rangle}{w_\alpha^k}, \tag{40}$$

where $\langle n_\alpha^k \rangle$ is the number of times letter α is expected to occur at position k of the regulatory sites under posterior distribution $P(i|D,w)$.

To derive the last equality, first note that derivative is a sum of independent terms

$$\frac{d \log[P(D \mid w)]}{dw_\alpha^k} = \sum_{m=1}^{n} \frac{d \log[P(D_m \mid w)]}{dw_\alpha^k}, \tag{41}$$

and that each term is again a sum of independent terms

$$\frac{d \log[P(D_m \mid w)]}{dw_\alpha^k} = \frac{(L-l+1)^{-1}}{P(D_m \mid w)} \sum_{i_m} \frac{dP(D_m \mid w, i_m)}{dw_\alpha^k}. \tag{42}$$

Now if the base $s(i_m + k)$ at position $i_m + k$ of sequence m is equal to α, then the last derivative on the right simply divides $P(D_m|w, i_m)$ by w_α^k, and else the derivative is zero. We thus have

$$\frac{dP(D_m \mid w, i_m)}{dw_\alpha^k} = \frac{\delta(s(i_m + k), \alpha)}{w_\alpha^k} P(D_m \mid w, i_m), \tag{43}$$

where the delta-function is one if $s(i_m + k) = \alpha$ and zero otherwise. We thus find

$$\frac{d \log[P(D_m \mid w)]}{dw_\alpha^k} = \frac{\sum_{i_m} \delta(s(i_m + k), \alpha) P(i_m \mid D, w)}{w_\alpha^k}. \tag{44}$$

Note that the numerator of the right-hand side of this equation is just the expected number of times letter α occurs at position k of the binding sites in D_m under the posterior distribution $P(i_m|D,w)$. Summing over all sequences m we thus obtain

$$\frac{d\log[P(D\,|\,w)]}{dw_\alpha^k} = \frac{\langle n_\alpha^k \rangle}{w_\alpha^k}. \tag{45}$$

Using the fact that the WM columns are normalized, we find that at the maximum of $P(D|w)$ the weight matrix components obey the equalities

$$w_\alpha^k = \frac{\langle n_\alpha^k \rangle}{n} \tag{46}$$

As in section "Finding clusters of binding sites: regulatory modules" one can use EM to solve these equations. We start with a randomly chosen WM w and calculate

$\langle n_\alpha^k \rangle$ for that WM. We then update the WM components using equation (46) and repeat until the WM no longer changes. This procedure is guaranteed to converge to a local optimum of $P(D|w)$.

Note that in the above we assumed just one site per sequence but it is easy to extend these derivations to arbitrary configurations, using the identities derived in section "Finding WM matches". Probably the first algorithm developed to find regulatory motifs in this way is the well-known MEME algorithm [17], and by now there are quite a number of algorithms that have been developed using this general idea, e.g. MDScan [18].

Once an optimal WM w_* is found it is straightforward, i.e. using equation (19), to calculate the posterior probabilities $P(i_m|D, w_*)$ that a site occurs at position i_m in sequence m and this allows one to distinguish between high confidence and low confidence sites. Programs that use the EM approach to motif finding often report such probabilities. Note, however, that the posterior probabilities $P(i_m|D, w_*)$ should not be confused with the posterior probabilities $P(i_m|D)$ which give the posterior probability that a site occurs at i_m *independent* of what the WM w is (we derive an expression for this probability below). The latter quantifies how much evidence there is in the data D that a site occurs at i_m, whereas $P(i_m|D,w_*)$ assumes in addition that the inferred WM w_* is correct. Since in many cases there is a reasonably high probability that w_* does not match precisely the WM from which the site derives, the probabilities $P(i_m|D, w_*)$ will typically be significantly larger than $P(i_m|D)$.

Finally, it would even be straightforward to extend the EM approach to multiple WMs using the expressions of section "Finding clusters of binding sites: regulatory modules". One could then, in principle, simultaneously find the set of priors $\{\pi_w\}$ and the set of WMs $\{w\}$ that maximize the overall probability $P(D|\{w\}, \{\pi_w\})$ of the data. For each W M w the expectation-maximization update equation of the WM components would take on the form

$$w_\alpha^k = \frac{\langle n_\alpha^k(w) \rangle}{\langle n(w) \rangle}, \tag{47}$$

where $\langle n(w) \rangle$ is the expected total number of sites for WM w that occur in D and $< n_\alpha^k\,(w)>$ is the expected number of those sites that have a base α at position k. The problem with this approach is that EM will very often lead to a local rather than the global optimum (it roughly speaking moves uphill from the starting point to the nearest local optimum). So depending on the initial sets $\{w\}$ and $\{\pi_w\}$ the EM procedure may lead to very different optima and the higher the dimension of the search-space, the more serious this problem becomes. Therefore, in practice algorithms such as MEME do not search for multiple WMs simultaneously but rather find one WM at a time. In addition, programs like MEME will start from many different initial WMs w and perform EM for each of them, reporting the best optimum found in any of these EMs.

Motif sampling

The second approach to motif finding focuses on the probability $P(D|i)$. To calculate (37) we substitute (35) for the likelihood and first note that it can be separated in a part $P(B_i|b, i)$ that depends only on the background, and a part $P(S_i|i)$ that is given by an integral, i.e. $P(D|i) = P(B_i|b,i)P(S_i|i)$ with

$$P(B_i\,|\,b,i) = \prod_{\sigma\in B_i} b_\sigma = \prod_\alpha (b_\alpha)^{n_\alpha(B_i)}, \tag{48}$$

where $n_\alpha(B_i)$ is the number of times base α occurs in the background B_i, and

$$P(S_i\,|\,i) = \int P(S_i\,|\,w,i)\,P(w)\,dw = \int P(w)\prod_{s\in S_i} P(s\,|\,w)\,dw. \tag{49}$$

For the prior $P(w)$ we use a Dirichlet prior as in (27), and use the general identity (31) to calculate the integral, which results in

$$P(S_i \mid i) = \prod_{k=1}^{l}\left[\frac{\Gamma(\gamma^k)}{\Gamma(n+\gamma^k)}\prod_{\alpha}\frac{\Gamma(n_\alpha^k(S_i)+\gamma_\alpha^k)}{\Gamma(\gamma_\alpha^k)}\right], \tag{50}$$

where $n_\alpha^k(S_i)$ is the number of times base α occurs at position k of the sites in S_i, and the γ_α^k are again the pseudocounts of the Dirichlet prior. The most common situation is that we know little about the WM that can be expected and in such situations either a uniform prior $\gamma_\alpha^k = 1$ or one that biases toward the corners of the simplex, e.g. $\gamma_\alpha^k = 0.5$, are reasonable choices. However, as we mentioned in the discussion of equation (28), if we already have a set of known sites S_{known} for the motif, in which base α appears m_α^k times at position k, then the posterior probability for the WM has the same form as a prior with counts $\gamma_\alpha^k = m_\alpha^k + 1$. Using this posterior as a prior in equation (50) we can thus also calculate the probability of obtaining the sequence segments in S_i when sampling from the same WM as the WM from which the set S_{known} derived. That is, equation (50) easily allows for the incorporation of prior knowledge about the WM w.

The meaning of equation (50)

Since the expression (50) is central in all motif sampling strategies we will divert here to discuss its meaning in a little more detail. First, note that $P(S_i|i)$ is a product of independent factors for each column k. We thus focus on a single column only. In addition, we will assume a uniform prior over WMs, i.e. $\gamma_\alpha^k = 1$. The expression for a single column then takes on the simpler form

$$P(S) = \frac{3!\prod_\alpha n_\alpha!}{(n+3)!} = \frac{3!n!}{(n+3)!}\frac{\prod_\alpha n_\alpha!}{n!}, \tag{51}$$

where we used that $\Gamma(x + 1) = x!$ for integer x. The second equality on the right is to clarify that $P(S)$ can be written as the product of two factors. The first of these factors, $3!n!/(n + 3)!$, is the inverse of the binomial coefficient $\binom{n+3}{3}$. This binomial coefficient corresponds to the number of different sets of counts $\{n_\alpha\}$ that are possible. That is, it counts the number of vectors of integers (n_a, n_c, n_g, n_t) such that $\sum_\alpha n_\alpha = n$.

The second factor in equation (51), $\prod_\alpha n_\alpha!/n!$, is the inverse of the multinomial coefficient $n!/(\prod_\alpha n_\alpha!)$ which gives the number of different ways that n objects can be distributed over 4 boxes such that n_a objects are in the first box, n_c in the second, n_g in the third, and n_t in the fourth. Thus, the probability $P(S)$ for a column of n bases is inversely proportional to the number of ways in which the counts $\{n_\alpha\}$ of this column can be realized. In summary, there are 4^n possible outcomes for the n bases in the column. The probability distribution $P(S)$ assigns a probability to each of these that is precisely inversely proportional to the number of the 4^n outcomes that lead to the counts $\{n_\alpha\}$. As a result, the total probability to obtain an outcome with counts $\{n_\alpha\}$ is *constant* for all $\binom{n+3}{3}$ possible counts (because we have to sum $P(S)$ over all possible outcomes that lead to the same set of counts).

For large n we can approximate the multinomial coefficient using Stirling's approximation to find

$$\frac{\prod_\alpha n_\alpha!}{n!} \approx e^{-nH(\{n_\alpha\})}, \tag{52}$$

where $H(\{n_\alpha\})$ is the entropy of the distribution n_α/n:

$$H(\{n_\alpha\}) = -\sum_\alpha \frac{n_\alpha}{n}\log\left(\frac{n_\alpha}{n}\right). \tag{53}$$

Thus, the probability $P(S)$ is largest for sets of sequences whose base distributions have lowest entropy.

Back to motif sampling

We now return to our motif sampling calculations. Using (50) and (48) we obtain $P(D|i)$ in terms of the counts $n_\alpha^k(S_i)$ and $n_\alpha(B_i)$. Finally, using a uniform prior over hypotheses i, the posterior $P(i|D)$ becomes simply

$$P(i \mid D) = \frac{P(D \mid i)}{\sum_j P(D \mid j)}, \tag{54}$$

where the sum in the denominator is over all possible assignments $j = (j_1,..., j_n)$ for the positions of the binding sites.

Ideally we would now either find the configuration of site positions i_* that maximizes $P(i|D)$, or we would for each position i_k calculate the posterior probability $P(i_k|D)$ that a site occurs at position i_k in sequence k, which is formally given by

$$P(i_k \mid D) = \sum_{i_1,\ldots,i_{k-1},i_{k+1},\ldots,i_n} P(i \mid D). \qquad (55)$$

Unfortunately, since $P(i|D)$ is a complicated nonlinear function of the base counts $n_\alpha^k(S_i)$ and $n_\alpha(B_i)$ we cannot separate it easily into contributions from the different hypothesized sites in i and there is generally no way to calculate sums like (55) other than explicitly summing over all $(L - l + 1)^{n-1}$ states. To find site configurations i with high $P(i|D)$ researchers have in in general resorted to Markov chain Monte-Carlo techniques for sampling the distribution $P(i|S)$[19]. The most commonly used way of sampling the distribution $P(i|S)$ is through so-called *Gibbs sampling* [20] and consists of iterations of the following steps, which are illustrated in Fig. 4

1. Randomly select one of the n sequences with uniform probability.

2. If sequence number m was selected, remove the segment s located at position i_m from the set of sites S_i of the current configuration. Denote this set of $(n - 1)$ sequences as S_i^- and the new configuration as i^-.

3. For every position $i_m = 0$ through $i_m = L - l$ denote the new configuration that results from placing the site at i_m in sequence m as (i^-, i_m) and calculate $P(D|i^-, i_m)$.

4. Select a new configuration by sampling the position of the site in sequence m according to the probability distribution

$$P(i_m \mid D, i^-) = \frac{P(D \mid i^-, i_m)}{\sum_{j_m=0}^{L-l} P(D \mid i^-, j_m)}. \qquad (56)$$

using (48) and (50) one finds that this probability is proportional to

$$P(i_m \mid D, i^-) \propto \prod_{k=1}^{l} \frac{n_{s(i_m+k)}^k(S_i^-) + \gamma_i^k}{b_{s(i_m+k)}(n - 1 + \gamma^k)}, \qquad (57)$$

where $s(i_m + k)$ is the base that occurs at position $i_m + k$ in sequence m. Note that this expression is precisely the ratio between the probability $P(s_{[i_m,l]}|S_i$ of the site at i_m deriving from the same WM as the others in S_i^-, i.e. as in equation (34), and the probability $P(s_{[i_m,l]} \mid b)$ of this segment under the background, i.e.

$$P(i_m \mid D, i^-) \propto \frac{P(s_{[i_m,l]} \mid S_i^-)}{P(s_{[i_m,l]} \mid b)}. \qquad (58)$$

By iterating these steps one can sample the entire distribution $P(i|D)$ and, for example, estimate the posterior probability $P(i_m|D)$ that a site occurs at position i_m in sequence m, i.e. by the fraction of time a site occurs at i_m during sampling. The probabilities $P(i_m|D)$ rigorously quantify the evidence in D that a site occurs at position i_m. Thus, whenever $P(i_m|D)$ is large we can be confident that a site does occur at i_m.

To make a single prediction for the set of regulatory sites in D one searches for the configuration i_* that maximizes $P(i|D)$. In some approaches, e.g. [21], this is done simply by keeping track of the highest probability configuration that was observed during sampling. However, more accurate determination of the optimal configuration i can be obtained through *simulated annealing* [22]. One introduces a parameter β and instead of sampling from $P(i|D)$ one samples from a probability distribution which is proportional to $P(i|D)^\beta$. At the start of the search β is set to a small number and then β is slowly increased with time. As β increases more weight will be put on configurations with high probability and eventually the sampler will 'freeze' into a state with locally optimal probability $P(i|D)$. Provided the annealing is done slowly enough the optimum will correspond to the globally optimal state. This is for example the approach taken by the PhyloGibbs algorithm [23].

Once an optimal state i_* is found through simulated annealing one can of course use normal sampling, i.e. with $\beta = 1$, to obtain the posterior probabilities of the sites in i_*. Given the optimal configuration i_* one can of course also report the expected WM given this configuration, which has components

$$\langle w_\alpha^k \rangle = \frac{n_\alpha^k(S_{i_*}) + \gamma_\alpha^k}{n + \gamma^k}. \qquad (59)$$

Instead of assuming that there is precisely one site in each of the n sequences we can of course also sample much more general configurations c. Most generally, one could allow varying numbers of sites for multiple WMs. The top left panel in figure 5 shows such a general configuration with sites for 3 different motifs (red, blue, and green). If we assume the same kind of priors as we used in section "Finding clusters of binding sites: regulatory modules" then the prior probability for a particular configuration c,

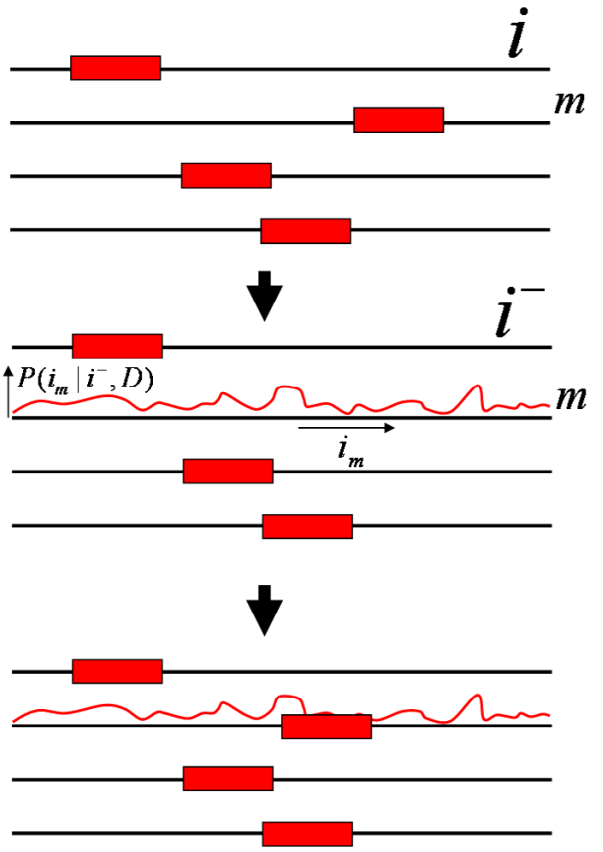

Figure 4
Illustration of the steps of the Gibbs sampling algorithm. The red profile indicates the posterior probability $P(i_m|D, \vec{r})$ and in the last step a new position is sampled from this distribution.

which has $n(w, c)$ sites for WM w and $n(b, c)$ bases in background, is proportional to

$$P(c \mid \{\pi\}) \propto \left[(\pi_{\text{bg}})^{n(b,c)} \right] \prod_w (\pi_w)^{n(w,c)}. \qquad (60)$$

If we denote the set of sites for WM w in configuration c by S_w and the set of background nucleotides as $B(c)$ we obtain for the likelihood of the data given the configuration

$$P(D \mid c) = \left[\prod_{\sigma \in B(c)} b_\sigma \right] \prod_w P(S_w), \qquad (61)$$

where for each group of sites the probability $P(S_w)$ is given in complete analogy with (50) by

$$P(S_w) = \prod_{k=1}^{l} \left[\frac{\Gamma(\gamma^k)}{\Gamma(n(w)+\gamma^k)} \prod_\alpha \frac{\Gamma(n_\alpha^k(S_w)+\gamma_\alpha^k)}{\Gamma(\gamma_\alpha^k)} \right], \qquad (62)$$

where $n(w)$ is the total number of sites in group S_w and $n_\alpha^k(S_w)$ is the number of times base α occurs at position k of the sites in S_w.

The posterior probability $P(c|D)$ of a configuration is simply proportional to the product of (60) and (61), i.e. $P(c|D) \propto P(D|c)P(c|\{\pi\})$. To sample from the posterior probability $P(c|D)$ over all possible configurations we need a more extensive set of 'moves' then the one described in the Gibbs sampler above. This can be done in a number of ways [24]. One possibility is to pick a sequence at random, to remove all sites currently located in it, and to sample from all ways of putting a new set of sites in, see [25] for details. The set of moves implemented by the PhyloGibbs algorithm [23] is illustrated in Fig. 5. These moves are:

1. Resampling a segment: Pick a sequence m at random and a random position i_m in it. Check if there is a site overlapping the region from $i_m + 1$ to $i_m + l$ in the current configuration c. If so, do nothing, i.e. move from c to c. If the region is free (or a site occurs precisely at $i_m + 1$ through $i_m + l$) calculate the probabilities $P(c'|D)$ for all configurations c' that are obtained by putting a site for any of the WMs w at i_m, including putting no site at all or putting a site for a new motif. Finally, sample one of these configurations c' with probability proportional to $P(c'|D)$.

2. Moving a site: Pick one of the sites occurring in c and remove it creating configuration c^-. Find all sequence segments s of length l in c^- that are not overlapping any site. Calculate the probability $P(c'|D)$ for all configurations that can be obtained by placing a new site for the same WM at any of the free segments s. Sample one of these configurations c' in proportion to $P(c'|D)$.

3. Shifting a site group: Pick one of the sets of sites S_w at random. Check how far the sites in S_w can be shifted to the left and right without colliding with other sites in the current configuration c. Denote these maximal shifts by $l_{\max}$ and $r_{\max}$. For every shift h between $h = l_{max}$ and $h = r_{max}$ calculate the probability $P(c'|D)$ of the configuration that would result if all sites in S_w were shifted by an amount h. Sample one of the configurations c' in proportion to probability $P(c'|D)$.

One of the main advantages of the motif sampling approach over EM algorithms is that it is much less likely

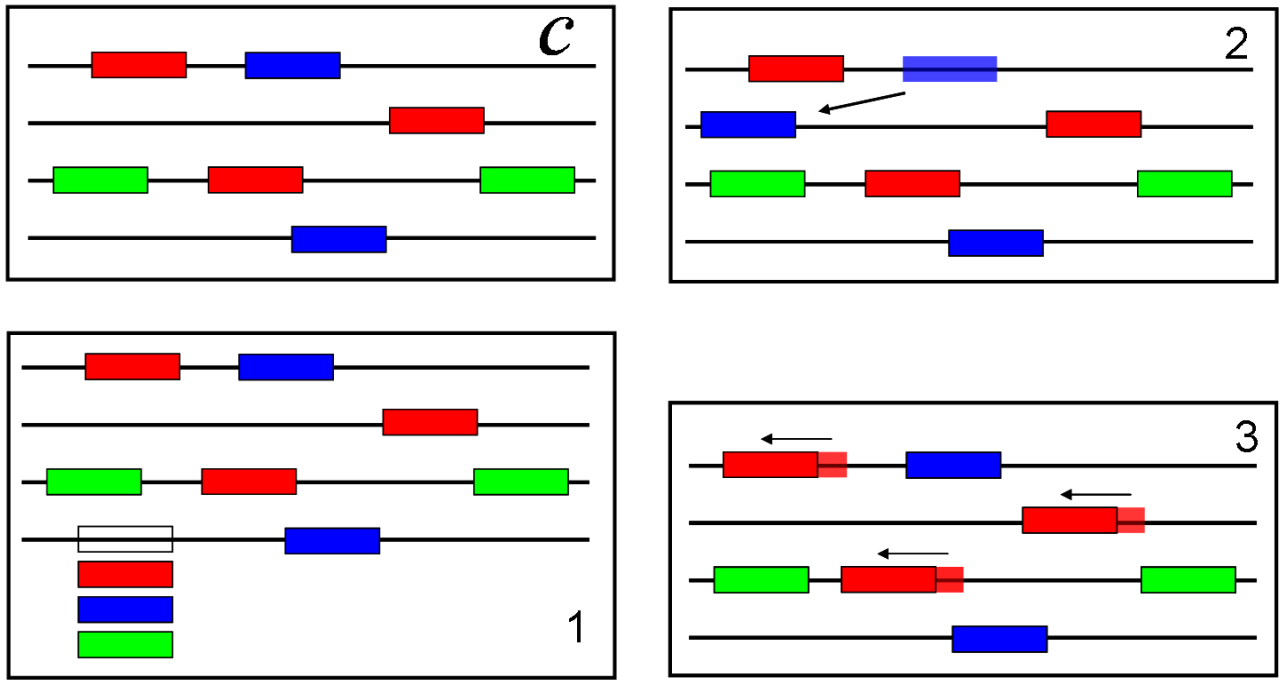

Figure 5
Illustration of a general configuration with varying site numbers for multiple motifs (upper left) and examples of moves used to sample all possible configurations. In 1 a randomly chosen segment is 'recolored', leaving it either blank (background), coloring with any of the existing motifs, or coloring it with a new color (new motif). In 2 a colored segment is chosen at random and moved to another location. In 3 all segments in a motif are shifted by the same amount.

to get stuck in local optima. In particular, one can sample multiple motifs without becoming trapped in bad local optima. Another advantage is that one can obtain rigorous posterior probabilities for sites appearing at different positions which allows for a more reliable separation of trustworthy predictions from spurious ones (see [23]). As for the single motif, i.e. our discussion below equation (50), we can here also use 'informative' priors for each of the motifs. That is, if we have a set motifs for which known sites are available we can use the base counts in these sites as 'pseudocounts' γ_α^k of priors for corresponding motifs in the binding sites configurations. That is, apart from inferring multiple new motifs, we can use informative priors to discover new sites for known motifs at the same time. This can be especially useful when we are trying to find a new motif in a set of sequences that also contains sites for a number of known motifs. If we were to search this data for a single motif then it is quite likely that the search would return one of the known motifs. By searching for multiple motifs at the same time and using informative priors for each of the known motifs we can make sure that known sites will automatically associate with the known motifs, and that the remaining motifs are indeed new motifs. Finally, under the sampling approach one can use arbitrarily complicated priors $P(c)$ on configurations, including priors that demand that certain combinations of sites occur at certain specified distances of each other, in particular orientations, etcetera. In the EM approach such complex priors would typically cause the dynamic programming solution to summing over all configurations to break down.

The main disadvantage of the motif sampling approach is of course speed. To obtain accurate statistics one needs to sample for a long time, and the time necessary grows with the product of the size of the data-set D, the total number of sites, and the number of motifs. In contrast, the dynamic programming approaches outlined in section "Finding WM matches" allow for efficient computation of sums over all possible configurations even for very large input data, allowing one to search very large sequences for matches to sets of WMs, which is computationally infeasible with motif sampling algorithms.

As mentioned already, motif sampling was introduced more than a decade ago [20]. Since then a significant

number of algorithms has been developed including [21,26-28], and probably many more. The PhyloGibbs algorithm [23] introduces several extensions such as simulated annealing to find the configuration with maximal probability, simultaneously sampling multiple motifs, and taking the phylogenetic relationships between the sequences into account (discussed below).

Clustering sites and motifs

There are several situations in which we may want to cluster sets of binding sites. This demand for instance arises whenever we have obtained a set of sequence segments that are thought to each have regulatory function, without knowing the specific function of any of the segments. For example, several researchers have used so-called 'phylogenetic footprinting', the identification of short overly conserved segments in alignments of orthologous intergenic regions from related genomes, to gather large collections of putative regulatory sites [29-32]. It is reasonable to assume that most of these short segments contain a regulatory site for some regulatory factor, but we do not know which sites are sites for the same factor nor how many different regulatory factors are represented in data.

Formally, given a dataset D of sequence segments, we want to partition this dataset into subsets such that all segments within a subset contain a regulatory site for a common regulatory factor, and different subsets correspond to different regulatory factors. In addition, we want to multiply align all the segments within each subset. Thus, for this problem the set of hypotheses is all possible ways in which the set D can be partitioned into subsets, and all possible ways in which the sequences in each subset can be multiply aligned. Let us denote possible configurations by C. Each configuration C consists of a set of subsets $c \in C$ that each consist of a collection of sequences from D. The union of these subsets c of course equals D. In addition C specifies, for each subset c, an alignment S_c of sequence segments that are taken from the sequences in c. For simplicity we will assume that all these sequence segments are of fixed length l in all subsets. That is, C specifies a partition of the sequences in D into subsets c, and it specifies where in each of the sequences the regulatory site of length l occurs, thereby specifying length-l alignments S_c for each subset c. We now want to calculate the probability $P(D|C)$ of the data given a configuration C. We can generally separate $P(D|C)$ into a contribution of the sites (those segments from the sequences in hypothesized regulatory sites) and the bases outside these segments that are scored according to a background model.

$$P(D|C) = P(D_{sites}|C)P(D_{bg}|C). \quad (63)$$

For simplicity we will use a background model that assigns a probability 1/4 to each base (extensions to more complex background models are straight forward). In that case the contribution $P(D_{\text{bg}}|C)$ is constant, i.e. does not depend on C and we just consider $P(D_{\text{sites}}|C)$. This probability can be written as a product of independent contributions from each subset

$$P(D_{\text{sites}} \mid C) = \prod_{c \in C} P(S_c), \quad (64)$$

where S_c is the alignment of sites in subset c. The probability $P(S_c)$ is just the probability that all sequence segments in S_c derive from a common WM. The probability $P(S_c)$ is simply given by replacing S_i with S_c in the right-hand side of equation (50).

To obtain the posterior probability

$$P(C \mid D) = \frac{P(D \mid C)P(C)}{\sum_{C'} P(D \mid C')P(C')} \quad (65)$$

we also need a prior $P(C)$ over partitions. The simplest prior is of course to assign a uniform prior $P(C)$ = constant. Note however that, a uniform prior over partitions may correspond to a very peaked prior with respect to the number of clusters. That is, given a dataset with 100 sequences there are astronomically more partitions of the data into, say, 30 subsets than there are partitions of the data into, say, 2 subsets. If one wants a uniform prior over the number of clusters one needs to assign a probability $P(C) \propto 1/S_{|C|}^{|D|}$, where $|D|$ is the total number of sequences in D, $|C|$ is the number of subsets in C, and $S_{|C|}^{|D|}$ is the number of possible partitions of $|D|$ objects into $|C|$ subsets, which is called a Stirling number of the second kind [33]. Note that with this prior a particular configuration C with, say, 2 subsets will have a much higher a priori probability than a configuration with, say, 30 subsets. That is, it is impossible to be a priori completely ignorant about partitions in general *and* about the number of subsets at the same time. Again there is no easy way to find the configuration C with maximal posterior probability. A fast procedure for determining a state C with high posterior probability is through hierarchical clustering. One starts out with each sequence in D forming a subset on its own. For every pair of sequences s, and s' one then calculates the probability of the configuration $C(s, s', i, i')$ that is obtained when the subsets s and s' are joined into a cluster, putting the hypothesized sites at positions i and i' respectively. We then find the combination (s, s', i, i) with maximal $P(C(s, s', i, i')|D)$ and create the corresponding

state $C(s, s', i, i')$. This procedure is repeated, i.e. at each iteration two subsets are fused so as to maximize $P(C|D)$. The iteration stops when there is no more subset merger that would increase $P(C|D)$. The great disadvantage of this procedure is that it generally leads to highly suboptimal local optima in $P(C|D)$.

A better alternative is to use Markov chain Monte-Carlo sampling and simulated annealing. A simple and effective move-set is as follows

1. Select one of the sequences in D at random and remove it from its current subset thereby creating a configuration C^-.

2. For each of the subsets in C^- consider the configuration $C(c, i)$ when the removed sequence s is put into subset c and the length-l site is s is started at position i. Also consider the configuration $C(0)$ which is obtained by putting the sequence s in a subset of its own. Calculate $P(C|D)$ for all these configurations and sample one of the configurations in proportion to these probabilities.

These steps are illustrated in Fig. 6.

By repeating these two steps one can sample from the posterior distribution $P(C|D)$ over all possible configurations. Through simulated annealing, i.e. sampling from $P(C|D)^\beta$ and slowly increasing β, one can attempt to locate the configuration C^* which globally maximizes $P(C|D)$. The PROCSE software [34] implements such a Markov chain Monte-Carlo scheme for simultaneously clustering and aligning sets of sequences that are thought to contain regulatory sites and it has been used to predict regulons in bacteria genome-wide. It has also been used to automatically curate sets of experimentally determined binding sites [35]. PROCSE first determines a 'reference configuration' C^* through simulated annealing and then performs another sampling run, i.e. with $\beta = 1$, to determine the posterior probabilities of the clusters that occur in the reference state C^*.

An almost identical procedure as just described can be used to cluster motifs or arbitrary combinations of motifs and sequences. Application of different motif finding algorithms to the same dataset, or application of the same algorithm to related datasets, often results in sets of inferred motifs that show clear commonalities. One is thus often interested in analyzing sets of motifs to identify which motifs are really different, and which motifs might represent a common underlying WM.

As we have seen in section "Finding WM matches" all our information about a motif, i.e. a WM w, can be represented by counts n_α^k that represent the number of observations of base α at position k of the sites. So more generally, we will assume that when we are given 'a motif' this information can always be represented by a set of counts n_α^k. For example, when we are given WM components w_α^k then we transform this into a set of counts n_α^k by specifying the pseudocounts γ_α^k of a prior, and the effective total number of observations n on which the w_α^k are based:

$$n_\alpha^k = w_\alpha^k (n + \gamma^k) - \gamma_\alpha^k. \quad (66)$$

Without loss of generality we can thus think of these counts n_α^k as deriving from an alignment S of sites for the motif. That is, we can generally specify our knowledge about a 'motif' by specifying an alignment S of sites drawn from the motif WM.

Such alignments S can be clustered and aligned with each other completely analogously to the procedure just described for single sequences. That is, one can think of an alignment S as a set of individual sequences s that have already been clustered and aligned with each other. When multiple such alignments S are mutually aligned and clustered into a larger alignment S_c then we calculate the probability $P(S_c)$ that all sequences in S_c derive from a common WM exactly like we did before, i.e. equation (49). Thus, the only difference between clustering and aligning single sequences, and clustering and aligning 'motifs' is that motifs are represented by sets of multiply aligned sequences, and that these motif alignments are so to speak 'glued together' in that these sequences will never be repartitioned during the sampling. The PROCSE software also allows for such pre-clustered and pre-aligned sequences to be submitted as input. In this way arbitrary combinations of single sequences and motifs can be aligned and clustered simultaneously.

Incorporating phylogeny

In all our approaches so far we have assumed that different sequences that contain binding sites can be considered independent samples from a WM w. In addition, the motif finding approaches that we discussed all presume that one is given sets of sequences that are likely to contain sites for common regulatory factors. In many cases researchers use independent biological evidence, such as expression data, to collect such sets of sequences that appear 'co-regulated' [27,36]. Apart from expression data, more recently ChIP-on-chip techniques have been used to collect sets of

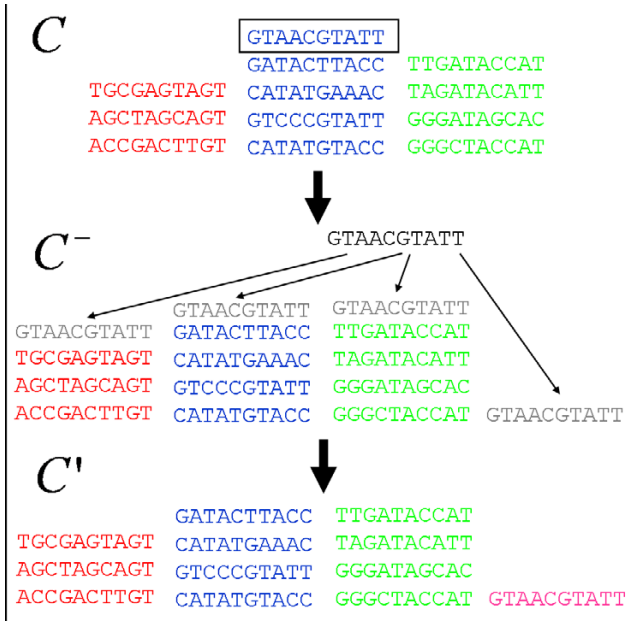

Figure 6

Illustration of the move-set for binding site clustering. Starting from a configuration *C* with three clusters, the top sequence in the blue cluster is chosen for resampling. It is removed from its cluster to produce configuration *C*-. Probabilities are then calculated for all configurations that would be obtained by inserting the sequence into any of the clusters or a new cluster (gray sequences), and finally one of these (*C*') is sampled. In this example the sequence was placed in a new cluster. For illustration purposes we have assumed all sequences in *D* have precisely the length *l* of the hypothesized site, so that each sequence can only be aligned in one way with any cluster. In general the sequences in *D* will be longer than *l* and one would also sample over all ways that the sequence can be aligned with each of the clusters.

sequences that appear to be bound by a common regulatory factor, see e.g. [37,38].

Another possibility is to collect sets of orthologous intergenic regions from related species. It is often reasonable to assume that many of the regulatory sites occurring in the ancestor of these species have been maintained and are shared by all or most of the descendants. Therefore, orthologous intergenic sequences can generally be expected to contain sites for common regulatory factors. However, in contrast to sites in collections of upstream regions of genes from a single species, these sites cannot be considered *independent* samples from a common WM. That is, the orthologous sites are related evolutionary, and their sequences will therefore generally be more correlated than independent samples from a WM. Therefore, to correctly analyze orthologous intergenic regions we need to take the phylogenetic relationships of the species into account.

Binding site evolution

Let us consider a single position in a regulatory site whose WM has components w_α at that position. We now want to calculate the probabilities $P_{\alpha\beta}(w, t)$ that over an evolutionary time t this position in the site evolves from base β to base α.

There is a long history of such models for the evolution of amino acids, e.g. see [39,40]. For our application to nucleotide evolution a general treatment of this problem was given by the model of Halpern and Bruno [41]. The rate $u_{\alpha\beta}$ at which base β is substituted by base α during evolution is written as the product of an instantaneous rate of mutation $\mu_{\alpha\beta}$ from β to α, and the probability $f_{\alpha\beta}$ that a mutation from β to α will be fixed in the population (which depends on selection), i.e.

$$u_{\alpha\beta} = f_{\alpha\beta}\mu_{\alpha\beta}. \tag{67}$$

Under this general model the probabilities $P(\alpha|\beta, w, t)$ are the solution of the differential equations

$$\frac{dP_{\alpha\beta}(w,t)}{dt} = \sum_{\gamma\neq\alpha}[u_{\alpha\gamma}P_{\gamma\beta}(w,t) - u_{\gamma\alpha}P_{\alpha\beta}(w,t)]. \tag{68}$$

Note that in the limit of long time the probabilities $P_{\alpha\beta}(w, t)$ become independent of time, i.e. memory of the start state is lost, and by the definition of the WM components the probabilities $P_{\alpha\beta}(w, t)$ limit to w_α, i.e.

$$\lim_{t\to\infty} P_{\alpha\beta}(w,t) = w_\alpha. \tag{69}$$

Assuming that the rates $\mu_{\alpha\beta}$ are given one can then solve [41] for the substitution rates $u_{\alpha\beta}$ that will lead to the limit distribution (69):

$$u_{\alpha\beta} = \mu_{\alpha\beta}\frac{\log\left[\frac{\mu_{\beta\alpha}w_\alpha}{\mu_{\alpha\beta}w_\beta}\right]}{1-\frac{\mu_{\alpha\beta}w_\beta}{\mu_{\beta\alpha}w_\alpha}}. \tag{70}$$

To solve equation (68) we note that it can be written as a matrix equation. Define the rate matrix **U** through

$$U_{\alpha\beta} = u_{\alpha\beta} - \delta_{\alpha\beta}\sum_\gamma u_{\gamma\alpha}. \tag{71}$$

In terms of this matrix **U** equation (68) becomes

$$\frac{d\mathbf{P}(t)}{dt} = \mathbf{U}\cdot\mathbf{P}(t), \tag{72}$$

with matrix $\mathbf{P}(t)$ having components $P_{\alpha\beta}(t)$. Using the boundary condition $P_{\alpha\beta}(0) = \delta_{\alpha\beta}$ the solution is given by

$$P_{\alpha\beta}(t) = (e^{\mathbf{U}t})_{\alpha\beta}. \quad (73)$$

In this general model one thus solves for $P_{\alpha\beta}(w, t)$ by first determining $\mathbf{U}$ using equation (70), and determining its eigenvalues and eigenvectors. However, note that in general the solution is a complicated function of the WM components w_α which is not easily amenable to further analysis.

To allow more analytic flexibility we have developed a simpler model of the evolution of binding sites [23,42] that assumes that all mutations are introduced at the same rate, i.e. $\mu_{\alpha\beta} = \mu$, and that the probability of fixation $f_{\alpha\beta}$ depends only on the target base α i.e. $f_{\alpha\beta} = w_\alpha$. Under these assumption the differential equations become

$$\frac{dP_{\alpha\beta}(w,t)}{dt} = \mu \sum_{\gamma\neq\alpha} [w_\alpha P_{\gamma\beta}(w,t) - w_\gamma P_{\alpha\beta}(w,t)] = \mu w_\alpha - \mu P_{\alpha\beta}(w,t) \quad (74)$$

This equation can be easily solved to give

$$P_{\alpha\beta}(w, t) = \delta_{\alpha\beta} e^{-\mu t} + w_\alpha(1 - e^{-\mu t}). \quad (75)$$

Note that $e^{-\mu t}$ is the probability that no mutations have taken place during time t. We call this no-mutation-probability the *proximity* $q = e^{-\mu t}$ between the ancestor and the descendant [23]. In terms of the proximity the solution becomes

$$P_{\alpha\beta}(w, q) = \delta_{\alpha\beta} q + (1 - q) w_\alpha. \quad (76)$$

This expression has a nice simple interpretation. With probability q no mutations have taken place in going from β to α and the bases are identical. With probability $(1 - q)$ one or more mutations took place and the probability that one then ends up with base α is simply the WM component w_α.

Probability of an orthologous set of bases

Assume that we have a set of orthologous intergenic regions and assume that we know the phylogenetic tree T that relates the species from which the regions derive. Consider now a set of orthologous bases S from these intergenic regions. That is, the bases in S have evolved from a common ancestor base in the common ancestor of the species according to the tree T. We now calculate the probability $P(S|T, w)$ that, when evolving from a common ancestor under one of the evolutionary models just discussed, and according to the given phylogenetic tree T, the set of bases S will result at the leafs of the tree.

Note that the set S only specifies the bases at the leafs of the tree T, i.e. the bases at the internal nodes are unknown. If we also knew all the bases at the internal nodes we could calculate $P(S|T, w)$ simply by multiplying the probabilities $P_{\alpha\beta}(w, t)$ for each branch, i.e.

$$P(S \mid T, w) = \prod_n P_{s_n s_{a(n)}}(w, t_n), \quad (77)$$

where the product is over all nodes n, s_n is the base at node n, $a(n)$ is the ancestor of node n, and t_n is the length of the branch from $a(n)$ to n. This is illustrated in the left panel of Fig. 7.

However, as we do not know the identities of the bases at the internal nodes, we thus have to sum over all possibilities. This can be done using a dynamic programming scheme first presented by Felsenstein [43]. We denote by $D_\alpha(n, w)$ the probability to observe all bases of S that are descendants of node n of the tree given that node n has base α. For nodes n that are leafs, i.e. bases of S, we of course have $D_\alpha(n, w) = \delta_{\alpha sn}$. We can determine $D_\alpha(n, w)$ for all nodes using the following recursion relation

$$D_\alpha(n, w) = \prod_{m\in c(n)} \left[\sum_\beta P_{\alpha\beta}(w, t_m) D_\beta(m, w) \right]. \quad (78)$$

where $c(n)$ is the set of children of node n, and t_m is the length of the branch connecting m to its parent n. This basic recursion is illustrated in the middle panel of Fig. 7. Starting from the leafs we can use (78) to calculate $D_\alpha(n, w)$ for all nodes up to the root of the tree. Finally the probability $P(S|T, w)$ for the whole tree is obtained by summing over the bases of the root node r, noting that the prior probability that root r has base α is w_α. This gives

$$P(S \mid T, w) = \sum_\alpha w_\alpha D_\alpha(r, w). \quad (79)$$

In complete analogy we can calculate the probability $P(S|T, b)$ of the column of bases S assuming that they evolved under a background model b. which is given by background probabilities b_α. To obtain $P(S|T, b)$ we just replace $P_{\alpha\beta}(w, t)$ with $P_{\alpha\beta}(b, t)$ for each branch of the tree in equation (78) and replace w_α with b_α in (79). Finally, we can also easily accommodate cases in which the regulatory site has been maintained in some but not all species. That is, we can have some branches of the tree T evolve according to the background model b whereas other branches evolve according to the WM column w, simply by using $P_{\alpha\beta}(w, t)$ for each branch evolving according to the WM, and using $P_{\alpha\beta}(b, t)$ for each branch evolving according to the background. An example of such a

more complicated 'selection pattern' is shown in the right panel of Fig. 7.

Finding sites and modules in multiple alignments

To apply the probabilities $P(S|T, w)$ and $P(S|T, b)$ to a set of orthologous intergenic regions we of course first have to identify which sets of bases in these sequences form orthologous groups. That is, we have to produce a multiple alignment of the orthologous intergenic regions. Given a multiple alignment we can then assume that every column of the alignment corresponds to a set of orthologous bases. The problem of producing accurate multiple alignments of non-coding sequences is extremely challenging and is beyond the scope of this article. There are now a number of algorithms available that focus specifically on alignment of non-coding DNA [44-46], although our personal experience is that consistency based methods [47,48] and evolutionary explicit progressive alignment [49] often outperform these methods significantly. From this point on we will assume that a global multiple alignment of the orthologous intergenic regions is given and that we can assume that vertically aligned bases in this alignment are orthologous.

We can use the probabilities $P(S|T, w)$ and $P(S|T, b)$ that we derived above to extend the formalism of sections "Finding WM matches" and "Finding clusters of binding sites: regulatory modules" to multiple alignments. The simplest way of doing this is to take one of the sequences in the multiple alignment as a *reference* sequence and to consider all binding site configurations for this reference sequence. This is often natural since in many cases we are really only interested in finding regulatory sites in one particular species and it is thus natural to take this species as a reference.

Let $s_{[i,l]}$ denote a segment of length l in this reference sequence, and let $S_{[i,l]}$ denote the corresponding block in the multiple alignment. To calculate the probability that a regulatory site occurs at $s_{[i,l]}$ we will now calculate the probabilities of observing the alignment segment $S_{[i,l]}$ under different assumptions for the selection that was operating at each branch of the tree T relating the species in the alignment. The simplest assumptions about the selection are that either all sequences in $S_{[i,l]}$ evolved according to the background model, i.e. using expression $P(S|T, b)$ for each column S in $S_{[i,l]}$, or that all sequences evolved according to WM w, i.e. using $P(S|T, w)$ for each column S in $S_{[i,l]}$. Many algorithms [23,50,51] in fact restrict themselves to these two possibilities. However, there are many other possibilities. If there are B branches in the tree then there are in principle 2^B possible ways of assigning selection to the branches, i.e. either WM w or background b for each branch. Formally, to calculate the probability that a regulatory site occurs at $s_{[i,l]}$ we would want to consider all 2^{B-1} 'selection patterns' σ for which $s_{[i,l]}$ is under selection of the WM w. We would want to assign prior probabilities $P(\sigma)$ to all 2^B possible selection patterns σ, and calculate the probabilities $P(S_{[i,l]}|T, \sigma)$ for each. Finally, by summing $P(S_{[i,l]}|T,\sigma)P(\sigma)$ over all selection patterns for which $s_{[i,l]}$ is under selection of the WM w one would obtain the total probability of the data $S_{[i,l]}$ under the assumption that a regulatory site occurs at $s[i,l]$. Unfortunately, there is no simple way of determining a reasonable distribution $P(\sigma)$ and the sum would generally involve a large number of terms. This author is not aware of any algorithm that currently implements this general scheme.

In the MotEvo algorithm [35] a single selection pattern σ_* is chosen that best fits the alignment and the sequences in it. Note first that, since WMs have a fixed width, a site in the reference species can only occur in another species if the corresponding segment in that species is gaplessly aligned with the site in the reference species. Therefore, we first check which of the other sequences in $S_{[i,l]}$ are gaplessly aligned with the reference sequence and which are not. For those sequences not gaplessly aligned with the reference we assign the background evolution model to the branches leading to these sequences. For each of the other sequences s in $S_{[i,l]}$ we calculate the probability $P(s|w)$ of the sequence under the WM w, and the probability $P(s|b)$ of the sequence under the background model b. Whenever $P(s|w) > P(s|b)$ we assign the WM model to the branch leading to s, and for all others we assign the background model. Finally, we assume that an internal node evolved according to the WM if any of its descendants do. This defines a unique selection pattern σ for $S_{[i,l]}$ and we calculate $P(S|T, w)$ using this selection pattern. The procedure is illustrated in Fig. 8.

We also calculate $P(S_{[i,l]}|T, b)$ assuming all branches evolved according to background. Finally, if we assign a prior probability π that a site occurs at $S[i,l]$ the posterior probability $P(\text{site}|S_{[i,l]})$ that the reference species has a functional site at i becomes

$$P(\text{site} \mid S_{[i,l]}) = \frac{P(S_{[i,l]} \mid T, w)\pi}{P(S_{[i,l]} \mid T, w)\pi + P(S_{[i,l]} \mid T, b)(1-\pi)}. \quad (80)$$

This is essentially the expression used by the MotEvo algorithm [35] to find regulatory sites. The MONKEY algorithm finds regulatory sites in a very similar manner. Instead of the simple evolutionary model (76) MONKEY uses the more general Halpern/Bruno model (70). However, MONKEY does not consider the possibility that the site is conserved in some but not all of the aligned species, i.e. it assumes that either all branches of the tree evolve

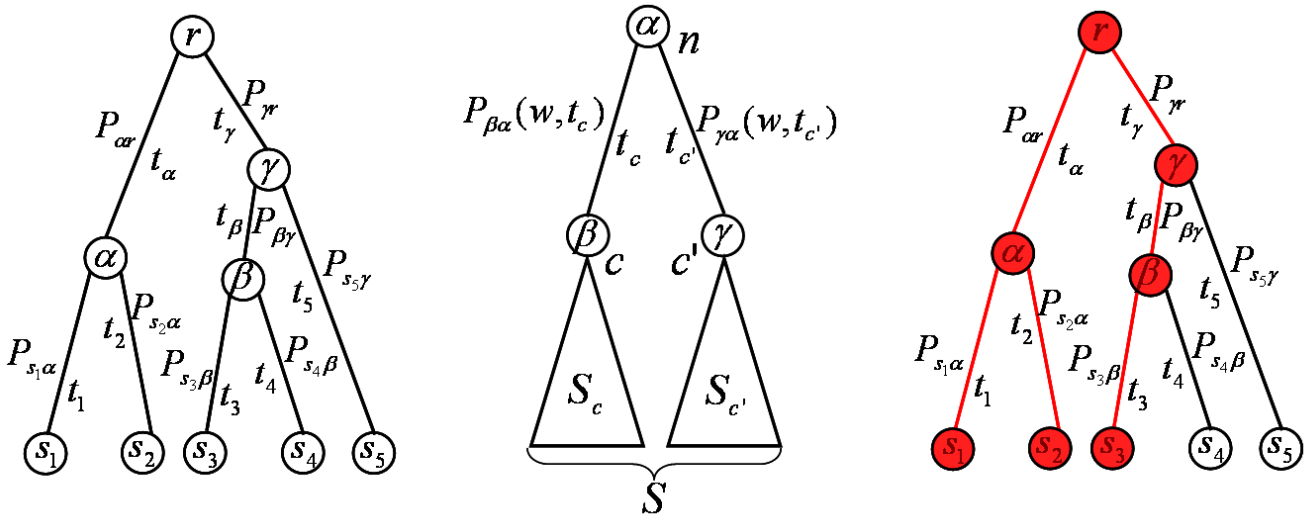

Figure 7
The evolution of a set of orthologous bases along a phylogenetic tree. In the left panel the expression (77) is illustrated. For notational simplicity we write $P_{\alpha\beta}$ for $P_{\alpha\beta}(w, t)$. The middle panel illustrates the recurion relations (78) with *c* and *c*' the children of node *n*, S_c the set of bases in *S* that descend from *c* and $S_{c'}$ the set of bases in *S* that descend from *c*'. The right panel shows expression (77) for a more complex selection pattern with branches evolving according to the WM in red, and those evolving to the background in black.

according to the WM, or all branches evolve according to background.

Instead of looking at one sequence segment at a time, we can of course also use this formalism to calculate sums of the probabilities of all possible binding site configurations as in section "Finding WM matches". Instead of calculating the probability $P(s_{[i,l]}|w)$ of a single sequence segment under the WM we instead calculate the probability $P(S_{[i,l]}|w)$ of the ungapped alignment block at that location using the procedure just outlined. That is, for every segment $s_{[i,l]}$ we find which other sequences are ungapped at the segment and choose which of these are evolving according to the WM based on the probabilities of the individual sequence segments under the WM. The generalization of equation (20) is then simply

$$F_n = F_{n-1} P(S_{[n-1,1]} \mid T, b)\pi_{\text{bg}} + \sum_w F_{n-l_w} P(S_{[n-l_w,l_w]} \mid T, w)\pi_w. \tag{81}$$

Note that position *n* here always refers to the *n*th base in the reference sequence.

Finally, using this formalism we can of course also search for regulatory modules in multiple alignments in complete analogy with the equations in section "Finding clusters of binding sites: regulatory modules".

This procedure has been implemented for two-species alignments in the Stubb algorithm [42]. Applying the Stubb algorithm to predict developmental regulatory modules in *Drosophila* it was shown in [52] that using two-species alignments improves predictions of the locations of regulatory modules over the single species algorithms.

Motif finding incorporating phylogeny

In section "Motif finding" we discussed two approaches to motif finding, one based on maximizing the probability $P(D|w)$ of the data given a WM *w* using expectation maximization, and one using Markov chain Monte-Carlo sampling to find the site configuration *c* that maximizes the posterior $P(c|D)$. These methods can also be extended in a straightforward way to multiple alignments and we now discuss these in turn.

Motif EM incorporating phylogeny

The PhyME algorithm implements an extension of the MEME algorithm to multiple alignments of orthologous intergenic sequences from related species. It uses a reference species and considers all configurations of binding sites that can be assigned to the reference species in the same way as discussed in the previous section, i.e. it uses equation (81) to calculate the overall likelihood $P(D|w)$

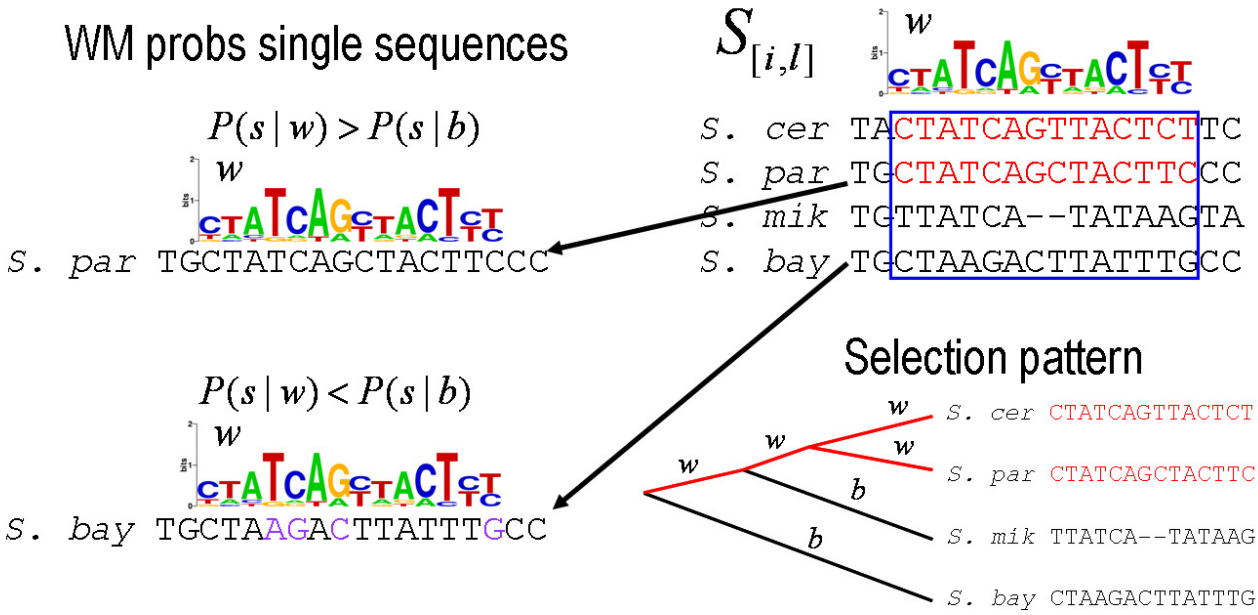

Figure 8
Probability for an alignment block assuming a site occurs in the reference sequence. In the top right an alignment segment $S_{[i,l]}$ is shown for the species *S. cerevisiae* (the reference), *S. paradoxus*, *S. mikatae*, and *S. bayanus*. First we check which sequences are gaplessly aligned with the reference. In this case *S. mikatae* contains a gap and the background model is assigned to this sequence. The reference has the WM model assigned by default (indicated in red). In the left the probabilities of the sequences from *S. paradoxus* and *S. bayanus* are compared with the WM (shown as a logo). It turns out the *S. paradoxus* sequence scores better for the WM than for background but the *S. bayanus* sequence scores better to background than to the WM, because of some mismatches to the WM consensus (bases in purple). Finally, on the bottom right the phylogenetic tree is indicated with the branches that evolve according to the WM in red, and those evolving according to the background in black.

of the alignment given the WM w. The evolutionary model that is used by PhyME to score ungapped alignment blocks $P(S_{[i,l]}|w)$ is precisely the simplified model of equation (76). However, like MONKEY and in contrast to MotEvo, PhyME assumes that either all branches in the tree evolved according to the WM model, or that all evolved according to background.

To maximize $P(D|w)$ with respect to the WM PhyMe needs to solve, for each column k in the WM, the equations

$$\frac{dP(D \mid w)}{dw_\alpha^k} = \text{constant } \forall \alpha. \qquad (82)$$

Note that for the single sequence case, the derivative of $P(s|w)$ with respect to the WM components w_α^k, was very simple, i.e. see (43). In contrast,the derivative $dP(S_{[i,l]}T, w)/d\,w_\alpha^k$ is a much more complicated function of the WM components w_α^k which needs to be calculated recursively just as $P(S_{[i,l]}|T, w)$ itself. Here it becomes particularly advantageous that in the simplified model (76) the probability $P_{\alpha\beta}(w,t)$ is such a simple function of the WM components. We do not discuss the mathematical details of solving (82) here except for mentioning the fact that it involves an iterative procedure similar to EM that leads to a local optimum in $P(D|w)$.

Motif sampling incorporating phylogeny

We now discuss extending the motif sampling approach of section "Motif sampling" to alignments of phylogenetically related sequences. Remember that in the motif sampling approach, instead of summing over all possible binding site configurations to calculate the probability $P(D|w)$ conditioned on the WM, we condition on a particular binding site configuration c and calculate the probability $P(D|c)$ by integrating over all possible WMs w.

Instead of a set of single sequences the input will now generally consist of a set of multiple alignments of orthologous non-coding sequences or a combination of multiple alignments and single sequences. As in section "Motif sampling" we want to consider all possible configurations c of binding sites that can be assigned to the input data D, and calculate the probability of the data $P(D|c)$ for each possible configuration. Whereas for single sequences the space of all possible configurations existed simply of all ways in which sets of non-overlapping windows can be assigned to the sequences, i.e. see Fig. 2, for multiple alignments the situation is a bit more complicated and illustrated in Fig. 9.

Above we assumed that a reference sequence s is given for each multiple alignment and that the set of binding ste configurations for the alignment is simply the set of all binding site configurations for the reference species. In the PhyloGibbs algorithm [23] there is no reference sequence and each sequence in the multiple alignment is treated the same. A site can be hypothesized to occur at any position of any of the sequences. By definition the algorithm assumes that, whenever a site occurs in one species, it will also occur in all other species that are gaplessly aligned with it at that location. That is, sites are automatically extended to all species that are mutually gaplessly aligned at that position, see Fig. 9. The algorithm makes sure to only allow configurations in which none of the sites overlap.

Next we need to calculate $P(D|c)$ for every possible such configuration c. This probability $P(D|c)$ is given by an equation essentially identical to equation (61). However, instead of single background bases σ with probability b_σ we will now have alignment columns S with probability $P(S|T, b)$ as calculated in section "Probability of an orthologous set of bases". The set of sequences S_w assigned to a WM w will now generally consist of several ungapped segments from the multiple alignments, i.e. alignment blocks, and possibly some single sequences as well, see Fig. 9. The probability $P(S_w)$ will again be an integral over all possible WMs but the integrand in this case will be considerably more complicated. For simplicity let's focus on a single column from the set S_w of sequence segments and alignment blocks. For simplicity assume that this column from S_w contains two independent columns S, and $\tilde{S}$ from the multiple alignments, see Fig. 9. The probability $P(S_w)$ would then be formally given by

$$P(S_w) = \int P(S \mid T, w)\, P(\tilde{S} \mid \tilde{T}, w)\, P(w) dw, \tag{83}$$

where T is the phylogenetic tree of alignment column S, $\tilde{T}$ the phylogenetic tree of alignment column $\tilde{S}$, and the expressions $P(S|T,w)$ and $P(\tilde{S} \mid \tilde{T}, w)$ are given as in equations (78) and (79). To calculate the integral notice that, formally, the expression $P(S|T,w)$ is a polynomial in the WM components of the following form

$$P(S \mid T, w) = \sum_k c_k \prod_\alpha (w_\alpha)^{m_\alpha^k} \tag{84}$$

where the prefactors c_k depend on the branch lengths in the tree and the m_α^k are sets of integers. The expression $P(\tilde{S} \mid \tilde{T}, w)$ can of course also be written in this form. Denote its prefactors $\tilde{c}_{\tilde{k}}$, and its exponents $\tilde{m}_\alpha^k$. Using this the integral can be rewritten as

$$P(S_w) = \sum_{k,\tilde{k}} c_k \tilde{c}_{\tilde{k}} \int \prod_\alpha (w_\alpha)^{m_\alpha^k + \tilde{m}_\alpha^{\tilde{k}} + \gamma_\alpha - 1} dw. \tag{85}$$

Note that each monomial term of the form $\prod_\alpha (w_\alpha)^{m_\alpha^k + \tilde{m}_\alpha^{\tilde{k}} + \gamma_\alpha - 1}$ can be easily integrated using the general expression (31). We then obtain for the integral

$$P(S_w) = \sum_{k,\tilde{k}} c_k \tilde{c}_{\tilde{k}} \frac{\Gamma(\gamma)}{\Gamma(m^k + \tilde{m}^{\tilde{k}} + \gamma)} \prod_\alpha \frac{\Gamma(m_\alpha^k + \tilde{m}_\alpha^{\tilde{k}} + \gamma_\alpha)}{\Gamma(\gamma_\alpha)}. \tag{86}$$

So in principle we can analytically determine the value of the integral $P(S_w)$ in this way. However, the number of terms in the above sum grows exponentially both with the number of sequences in each alignment and, more importantly, with the number of alignments under the integral. That is, if the configuration c contains 10 multiple alignment segments for WM w, then even if there were only 10 terms for each alignment column $P(S|T, w)$, there would still be 10^{10} terms in total. In practice we thus have to resort to approximations of the above integral. The approach that is taken in the PhyloGibbs algorithm is to approximate the expression $P(S|T, w)$ with a monomial for each alignment column, i.e.

$$P(S \mid T, w) \approx c \prod_\alpha (w_\alpha)^{x_\alpha}, \tag{87}$$

where the x_α may be non-integer. The prefactor c and the exponents x_α are set such that the first moments of the

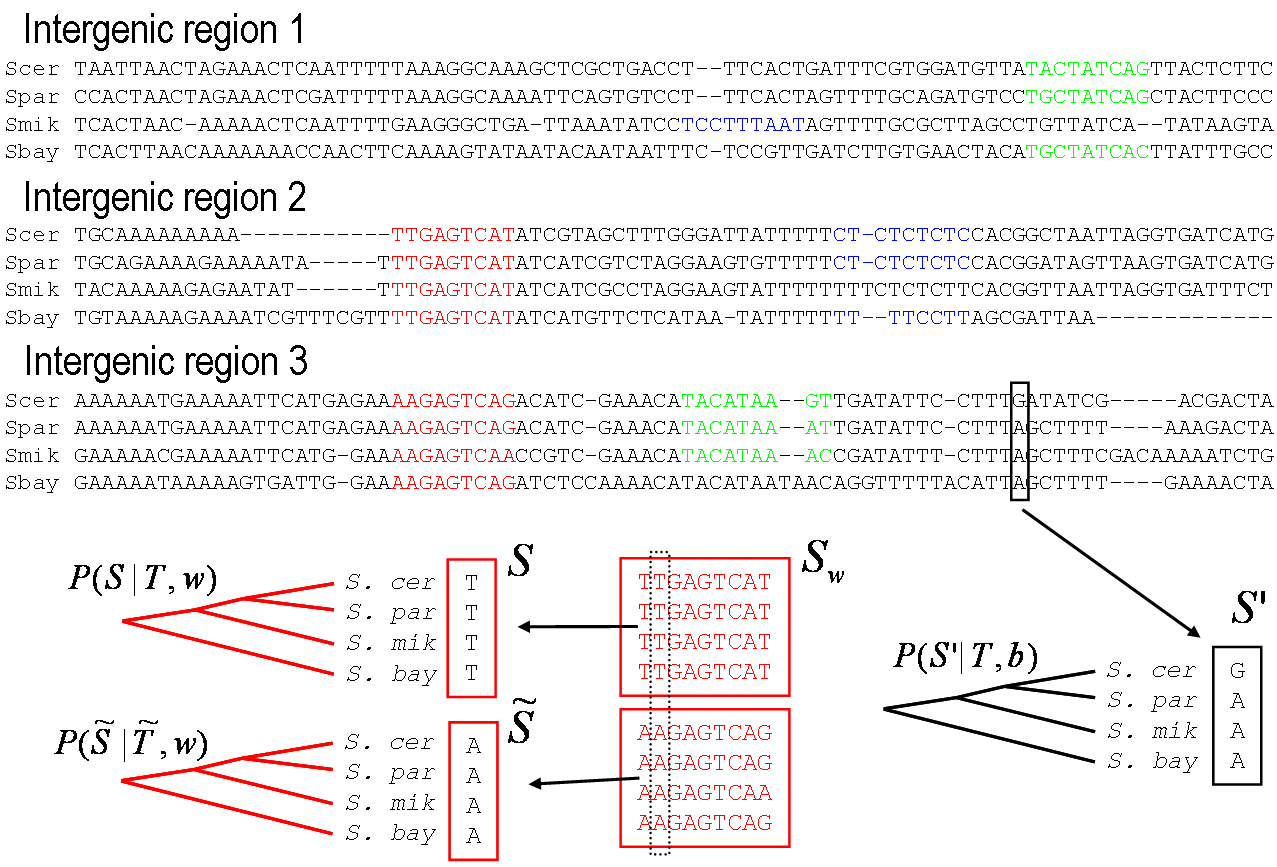

Figure 9

An input data-set consisting of the multiple alignments of 3 sets of orthologous intergenic regions from *S. cerevisiae, S. paradoxus, S. mikatae*, and *S. bayanus*. A binding site configuration *c* with sites for three motifs (red, green, and blue) is indicated. Note that each site is extended over all sequences that are locally gaplessly aligned. Most columns in the data are scored according to the background model in this configuration. On the lower right one example of an aligment column *S'* that is scored according to the background is shown. On the lower left the alignment S_w of sequences assigned to the red motif *w* is shown. A single column from this alignment consists of two independent columns, *S* and $\tilde{S}$, that derive from the multiple alignments of intergenic regions 2 and 3 respectively. The trees on the left show that under this configuration, the columns *S* and $\tilde{S}$ are both assumed to have evolved according to the same WM *w*, as indicated by the red branches on their phylogenetic trees *T* and $\tilde{T}$.

approximation match those of $P(S|T, w)$. That is, we demand that

$$c\int\prod_{\alpha}(w_\alpha)^{x_\alpha}\,dw=\int P(S\mid T,w)dw. \qquad (88)$$

and

$$c\int w_\beta\prod_{\alpha}(w_\alpha)^{x_\alpha}\,dw=\int w_\beta P(S\mid T,w)dw \qquad (89)$$

for all β. As shown in [23] this fixes *c* and the relative sizes of the x_α but leaves $\sum_\alpha x_\alpha$ still free. The absolute magnitude of the x_α we set so as to approximate the second moments, i.e. such that

$$c\int w_\beta w_\gamma\prod_{\alpha}(w_\alpha)^{x_\alpha}\,dw\approx\int w_\beta w_\gamma P(S\mid T,w)dw. \qquad (90)$$

for all combinations of β and γ. With these approximations the integral for $P(S_w)$ becomes simply

$$P(S_w) = c\tilde{c}\frac{\Gamma(\gamma)}{\Gamma(x+\tilde{x}+\gamma)}\prod_\alpha \frac{\Gamma(x_\alpha+\tilde{x}_\alpha+\gamma_\alpha)}{\Gamma(\gamma_\alpha)}, \quad (91)$$

where $x = \sum_\alpha x_\alpha$ and the variables with a tilde are those of the approximation to $P(\tilde{S} \mid \tilde{T}, w)$. The crucial point of this approximation procedure is that, at the start of the algorithm, we can determine these approximations, i.e. the values of the $x_{\alpha'}$ for every multiple alignment column S that occurs in the input data once and store the results. We thus replace the complex expression $P(S|T, w)$ with the simple expression $c\prod_\alpha (w_\alpha)^{x_\alpha}$ for each alignment column S. After that, when we are sampling different configurations, the expression $P(S_w)$ can be as efficiently calculated as for single sequences. That is, we can simply use equation (62), where $n_\alpha^k(S_w)$ is now the sum over the x_α of all the alignment segments that occur in S_w.

For the prior over configurations $P(c)$ PhyloGibbs uses the same priors (60) as for configurations over single sequences. PhyloGibbs uses Markov chain Monte-Carlo sampling to sample the space of all binding site configurations. The move-set employed when sampling binding site configurations in multiple alignments is essentially the same as the move-set for binding site configurations in single sequences illustrated in Fig. 5. The only difference is that 'sites' now typically extend over multiple aligned sequences, as illustrated in Fig. 9. Simulated annealing is used to find a configuration c_* that maximizes the posterior probability $P(c|D)$. Finally, a further sampling run is used to calculate the posterior probabilities of the sites in configuration c_*. PhyloGibbs reports both the configuration c_* and the inferred WMs of the motifs in c_*, as well as posterior probabilities for all sites occurring in c_*. In [23] we demonstrate the performance of PhyloGibbs on synthetic data, on individual multiple alignments of orthologous intergenic regions from yeast, and on sets of multiple alignments of intergenic regions from yeast that are bound by a common regulatory factor [38]. These tests show that taking phylogeny into account significantly improves the performance in motif finding.

Finally, it is important to distinguish the motif finding methods that rigorously incorporate phylogeny by probabilistically modeling the evolution of binding sites, such as the PhyME and PhyloGibbs algorithms just discussed, from more *ad hoc* algorithms that use comparative genomic information in various ways in motif finding. This includes for example methods that simply identify significantly conserved sequence segments in multiple alignments, [30-32]. These conserved segments can then be post-processed to search for over-represented motifs. In other approaches, e.g. [29,53], orthologous upstream regions are searched in the same way as set of upstream regions of co-regulated genes from a single species would be searched, i.e. ignoring the evolutionary relationships between the sequences. In other algorithms [54,55] one only takes the topology of the phylogenetic tree into account and searches for length-l segments that occur in all orthologous sequences, such that the minimal number of mutations necessary to relate the length-l segments, i.e. the parsimony score, is under some prespecified cut-off. Another approach is to first search for significantly conserved segments in orthologous intergenic regions, and to then multiply align conserved segments from the upstream regions of co-regulated genes. This approach is taken by the PhyloCon algorithm [56] which, in spite of its name, ignores the phylogenetic relations between the species.

The biggest challenge for incorporating comparative genomic information in motif finding that is currently outstanding is the treatment of the multiple alignment. It is clear that errors in the multiple alignment can have very deleterious effects on the performance of algorithms such as PhyME, PhyloGibbs, and MotEvo. Ideally one would simultaneously search the space of all multiple alignments and all binding site configurations. However, this space is very large and it is currently unclear if and how it can be effectively searched, especially for large data-sets.

Acknowledgements

EvN thanks Ionas Erb, Nacho Molina, and Mikhail Pachkov for useful comments.

This article has been published as part of *BMC Bioinformatics* Volume 8 Supplement 6, 2007: Otto Warburg International Summer School and Workshop on Networks and Regulation. The full contents of the supplement are available online at http://www.biomedcentral.com/1471-2105/8?issue=S6

References

1. Berg OG, von Hippel PH: **Selection of DNA binding sites by regulatory proteins: Statistical-mechanical theory and application to operators and promoters.** *J Mol Biol* 1987, **193:**723-750.
2. Roulet E, Busso S, Camargo AA, Simpson AJ, Mermod N, Bucher P: **High-throughput SELEX-SAGE method for quantitative modeling of transcription-factor binding sites.** *Nat Biotechnol* 2002, **20:**831-835.
3. Benos PV, Bulyk ML, Stormo GD: **Additivity in protein-DNA interactions: how good an approximation is it?** *Nucl acids res* 2002, **30(20):**4442-4451.
4. Djordjevic M, Sengupta AM, Shraiman BI: **A Biophysical approach to Transcription Factor Binding Site Discovery.** *Genome Research* 2003, **13:**2381-2390.
5. Bintu L, Buchler NE, Garcia HG, Gerland U, Hwa T, Kondev J, Phillips R: **Transcriptional regulation by the numbers: models.** *Curr Opin Genet Dev* 2005, **15(2):**116-124.
6. Jaynes ET: *Probability Theory: The Logic of Science Cambridge University Press*; 2003.
7. Barash Y, Elidan G, Friedman N, Kaplan T: **Modeling dependencies in protein-DNA binding sites.** *RECOMB* 2003:28-37.

8. Rabiner LR: **A tutorial on Hidden Markov Models and selected applications in speech recognition.** *Proceedings of the IEEE* 1989, **77(2):**257-286.
9. Durbin R, Eddy S, Krogh G, Mitchison G: *Biological Sequence Analysis Cambridge University Press*; 1998.
10. Davidson EH: *Genomic regulatory systems San Diego: Academic Press*; 2001.
11. Rivera-Pomar R, Jackle H: **From gradients to stripes in Drosophila embryogenesis: filling in the gaps.** *Trends Genet* 1996, **12(11):**478-483.
12. Frith MC, Hansen U, Weng Z: **Detection of cis-element clusters in higher eukaryotic DNA.** *Bioinformatics* 2001, **17(10):**878-889.
13. Berman BP, Nibu Y, Pfeifferdagger BD, Tomancak P, Celniker SE, Levine M, Rubin GM, Eisen MB: **Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the Drosophila genome.** *Proc Natl Acad Sci USA* 2002, **99:**757-762.
14. Rajewsky N, Vergassola M, Gaul U, Siggia ED: **Computational detection of genomic cis regulatory modules, applied to body patterning in the early Drosophila embryo.** *BMC Bioinformatics* 2002, **3(30):**.
15. Zavolan M, Rajewsky N, Socci ND, Gaasterland T: **SMASHing regulatory sites in DNA by human-mouse sequence comparisons.** *Proc IEEE Conf on Comp Sys Bioinf* 2003.
16. Eisen MB: **All motifs are NOT created equal: structural properties of transcription factor-DNA interactions and the inference of sequence specificity.** *Genome Biol* 2005, **6(5):**P7.
17. Bailey T, Elkan C: **Fitting a mixture model by expectation maximization to discover motifs in biopolymers.** *Proceedings of the Second International Conference on Intelligent Systems for MolecularBiology* 1994, **2:**28-36.
18. Liu XS, Brutlag DL, Liu JS: **algorithm for finding protein-DNA binding sites with applications to chromatin immunoprecipitation experiments.** *Nat Biotechnol* 2002, **20:**835-839.
19. Liu JS: *Monte Carlo Strategies in Scientific Computing Springer-Verlag*; 2001.
20. 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.
21. Thompson W, Rouchka EC, Lawrence CE: **Gibbs Recursive Sampler: finding transcription factor binding sites.** *Nucl Acids res* 2003, **31(13):**3580-3585.
22. Kirkpatrick S, Jr CDG, Vecchi MP: **Optimization by Simulated Annealing.** *Science* 1983, **220(4598):**671-680.
23. Siddharthan R, Siggia ED, van Nimwegen E: **PhyloGibbs: A Gibbs Sampling Motif Finder That Incorporates Phylogeny.** *PLoS Comput Biol* 2005, **1(7):**e67.
24. Frenkel D, Smit B: *Understanding Molecular Simulation: From Algorithms to Applications Academic Press*; 1996.
25. Liu JS, Neuwald AF, Lawrence CE: **Markovian structures in biological sequence alignment.** *Journal of the American Statistical Association* 1999:1-15.
26. Roth FP, Hughes JD, Estep PW, Church CM: **Finding DNA-regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation.** *Nat Biotechnol* 1998, **16:**939-945.
27. Liu X, Liu JS, Brutlag DL: **Bioprospector: Discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes.** *Pac Symp Biocomput* 2001:127-138.
28. Thijs G, Lescot M, Marchal K, Rombauts S, Moor BD, Rouzé P, Moreau Y: **A higher order background model improves the detection of regulatory elements by Gibbs Sampling.** *Bioinformatics* 2001, **17(12):**1113-1122.
29. McCue LA, Thompson W, Carmack CS, Ryan MP, Liu JS, Derbyshire V, Lawrence CE: **Phylogenetic footprinting of transcription factor binding sites in proteobacterial genomes.** *Nucl Acids Res* 2001, **29(3):**774-782.
30. Rajewsky N, Socci ND, Zapotocky M, Siggia ED: **The evolution of DNA regulatory regions for proteo-gamma bacteria by interspecies comparisons.** *Genome Res* 2002, **12:**298-308.
31. Cliften P, Sudarsanam P, Desikan A, Fulton L, Fulton B, Majors J, Waterston R, Cohen BA, Johnston M: **Finding functional features in Saccharomyces genomes by phylogenetic footprinting.** *Science* 2003, **301:**71-76.
32. Kellis M, Patterson N, Endrizzi M, Birren B, Lander ES: **Sequencing and comparison of yeast species to identify genes and regulatory elements.** *Nature* 2003, **423:**241-254.
33. Abramowitz M, Stegun IA, Eds: *Handbook of Mathematical Functions. With Formulas. Graphs, and Mathematical Tables Dover Pubns*; 1974.
34. van Nimwegen E, Zavolan M, Rajewsky N, Siggia ED: **Probabilistic clustering of sequences: Inferring new bacterial regulons by comparative genomics.** *Proc Natl Acad Sci USA* 2002, **99:**7323-7328.
35. Erb I, van Nimwegen E: **Statistical Features of yeast's transcriptional regulatory code.** *IEE Proceedings Systems Biology ICCSB* 2006.
36. Hughes JD, Estep PW, Tavazoie S, Church CM: **Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae.** *J Mol Biol* 2000, **296:**1205-1214.
37. Lee TI, Rinaldi NJ, Robert F, Odom DT, Bar-Jospeh Z, Gerber GK, Hannett NM, Harbison CT, Thompson CM, Simon I, Zeitlinger J, Jennings EG, Murray HL, Gordon DB, Ren B, Wyrick JJ, Tange JB, Volkert TL, Fraenkel E, Gifford DK, Young RA: **Transcription regulatory neworks in Saccharomyces cerivisiae.** *Science* 2002:799-804.
38. Harbison CT, Gordon DB, Lee TI, Rinaldi NJ, Macisaac KD, Danford TW, Hannett NM, Tagne JB, Reynolds DB, Yoo J, Jennings EG, Zeitlinger J, DK DKP, Kellis M, Rolfe PA, Takusagawa KT, Lander ES, Gifford DK, Fraenkel E, Young RA: **Transcriptional regulatory code of a eukaryotic genome.** *Nature* 2004, **431:**99-104.
39. Dayhoff M, Schwartz R, Orcutt B: **A model of evolutionary change in proteins.** *Atlas of protein sequence and structure* 1978, **5:**345-352.
40. Müller T, Spang P, Vingron M: **Estimating Amino Acid Substitution Models: A Comparison of Dayhoff's Estimator, the Resolvent Approach and a Maximum Likelihood Method.** *Mol Biol Evol* 2002, **19:**8-13.
41. Halpern AL, Bruno WJ: **Evolutionary distances for protein-coding sequences: modeling site-specific residue frequencies.** *Mol Biol Evol* 1998, **15(7):**910-917.
42. Sinha S, van Nimwegen E, Siggia ED: **A probabilistic method to detect regulatory modules.** *Bioinformatics* 2003, **19(suppl 1):**i292-i301.
43. Felsenstein J: **Evolutionary trees from DNA sequences: a maximum likelihood approach.** *J Mol Evol* 1981, **17:**368-376.
44. Brudno M, Do CB, Cooper GM, Kim MF, Davydov E, Green ED, Sidow A, Batzogloue S: **LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA.** *Genome Res* 2003, **13(4):**721-731.
45. Morgenstern B, Dress A, Werner T: **Multiple DNA and protein sequence alignment based on segment-to-segment comparison.** *Proc Natl Acad Sci USA* 1996, **93:**12098-12103.
46. Bray N, Pachter L: **MAVID: Constrained Ancestral Alignment of Multiple Sequences.** *Genome Res* 2004, **14:**693-699.
47. Do C, Mahabhashyam M, Brudno M, Batzoglou S: **ProbCons: Probabilistic consistency-based multiple sequence alignment.** *Genome Research* 2005, **15:**330-340.
48. Notredame C, Higgins D, Heringa J: **T-Coffee: A novel method for multiple sequence alignments.** *J Mol Biol* 2000, **302:**205-217.
49. Loytynoja A, Goldman N: **An algorithm for progressive multiple alignment of sequences with insertions.** *Proc Natl Acad Sci USA* 2005, **102:**10557-10562.
50. Moses AM, Chiang DY, Pollard DA, Iyer VN, Eisen MB: **MONKEY: identifying conserved transcription-factor binding sites in multiple alignments using a binding site-specific evolutionary model.** *Genome Biol* 2004, **5:**R98.
51. Sinha S, Blanchette M, Tompa M: **PhyME: A probabilistic algorithm for finding motifs in sets of orthologous sequences.** *BMC Bioinformatics* 2004, **5:**170.
52. Sinha S, Schroeder MD, Unnerstall U, Gaul U, Siggia ED: **Cross-species comparison significantly improves genome-wide prediction of cis-regulatory modules in Drosophila.** *BMC Bioinformatics* 2004, **5:**129.
53. McCue LA, Thompson W, Carmack CS, Lawrence CE: **Factors influencing the identification of transcription factor binding sites by cross-species comparison.** *Genome Res* 2002, **12:**1523-1532.
54. Blanchette M, Schwikowski B, Tompa M: **Algorithms for phylogenetic footprinting.** *J Comput Biol* 2002, **9(2):**211-223.

55. Blanchette M, Tompa M: **Discovery of regulatory elements by a computational method for phylogenetic footprinting.** *Genome Res* 2002, **12(5):**739-748.
56. Wang T, Stormo G: **Combining phylogenetic data with co-regulated genes to identify regulatory motifs.** *Bioinformatics* 2003, **19(18):**2369-2380.