Neural Networks for Determining Protein Specificity and Multiple Alignment of Binding Sites

John M. Heumann[1], Alan S. Lapedes[2] and Gary D. Stormo[3]
[1]Hewlett-Packard Co., Manufacturing Test Division
P.O. Box 301, Loveland, CO 80539-0301
phone: 303-679-3761, FAX: 303-679-5961
heumann@hpmtljh.lvld.hp.com
[2]Theoretical Division, MS B213
Los Alamos National Laboratory, Los Alamos, NM 87544
and the Santa Fe Institute, Santa Fe New Mexico
phone: 505-667-7608, FAX: 505-665-3003
asl@t13.lanl.gov
[3]Department of Molecular, Cellular and Developmental Biology
University of Colorado, Boulder, CO 80309-0347
phone: 303-492-1476, FAX: 303-492-7744
stormo@boulder.colorado.edu

Abstract

We use a quantitative definition of specificity to develop a neural network for the identification of common protein binding sites in a collection of unaligned DNA fragments. We demonstrate the equivalence of the method to maximizing Information Content of the aligned sites when simple models of the binding energy and the genome are employed. The network method subsumes those simple models and is capable of working with more complicated ones. This is demonstrated using a Markov model of the *E. coli* genome and a sampling method to approximate the partition function. A variation of Gibbs' sampling aids in avoiding local minima.

Introduction

Regulation of gene expression often involves proteins that bind to particular regions of DNA. Determining the binding sites for a protein and its specificity usually requires extensive biochemical and/or genetic experimentation. In this paper we illustrate the use of a neural network to obtain the desired information with much less experimental effort. It is often fairly easy to obtain a set of moderate length sequences, perhaps one or two hundred base-pairs, that each contain binding sites for the protein being studied. For example, the upstream regions of a set of genes that are all regulated by the same protein should each contain binding sites for that protein. One could also collect a set of restriction fragments that show binding activity to the protein. Given such data, a set of DNA fragments each known to contain at least one binding site for the protein, we want to determine where the binding sites are on each fragment and the specificity of the protein. The two types of information we desire, the protein's specificity and the location of the binding sites, are related. If we knew the specificity we could predict the binding sites, and if we knew several examples of binding sites we could develop a model of the specificity from them (Stormo 1990).

Two other approaches have been used successfully on this problem, a greedy algorithm (Stormo & Hartzell III 1989; Hertz, Hartzell III, & Stormo 1990) and an Expectation-Maximization (EM) algorithm (Lawrence & Reilly 1990). In the greedy approach solutions are constructed from pair-wise comparison of sequences, adding in new sequences until a site has been found on each sequence or it appears that some sequences do not have binding sites. This latter ability, to detect "bad" data or multiple classes of sites, is one of its advantages. Recent advances in this method include not specifying the length of the binding site in advance, allowing gaps in the alignments of the sites, allowing correlations between adjacent positions in the sites, allowing multiple sites per fragment and working on protein as well as DNA sequences (Hertz and Stormo, in preparation).

The EM method, in contrast, is a likelihood-based method which examines all the data simultaneously. It iteratively refines the model parameters (specificity) to arrive at a maximum likelihood solution. Convergence to a locally optimal solution is assured; in favorable cases this will also be the global optimum. This method has also been extended to sites that include gaps in the alignment (Cardon & Stormo 1992). Related methods include Hidden Markov Models (HMM) which explicitly include correlations between adjacent positions (Baldi *et al.* 1994; Krogh *et al.* 1994). The method has also been modified recently to use a Gibbs sampling approach and shown to work well on finding common domains in protein sequences (Lawrence *et al.* 1993). Another approach, based on enumerating possible patterns and finding those which occur most often in the collection of sequences, has also been used to identify common motifs in protein sequences with considerable success (Henikoff & Henikoff 1991). While other approaches to multiple sequence alignment have been described, most are tailored to the "global" problem of aligning sequences along their entire lengths. Such methods usually do not perform well on the "local" alignment problems that we address in this paper.

Despite the existence of some good methods, neural network approaches have advantages that may prove important, at least for some binding proteins. One of the disadvantages of all the previous methods, at least as utilized thus far, is the restriction to models whereby the protein's affinity for the binding sites is additive over the positions of the site. That is, the models require that the positions of the site interact with the protein independently of one another. This is due to the function being optimized, either "Information Content" or the closely related log-likelihoods, being summed over all of the positions. Even with the extension of the greedy approach to allow neighboring correlations or the use of such correlations in the HMM, this really only allows the additive components to be dinucleotides rather than single bases. Modifications of HMM for finding common RNA structures allow for correlations between complementary pairs in the sequences, but this still has limited flexibility (Eddy & Durbin 1994; Sakakibara *et al.* 1994). Neural networks, on the other hand, have the ability to detect and utilize important correlations in the data without specifying them in advance. Even single layer networks, perceptrons, can utilize correlational information unavailable to methods assuming strict independence (Stolorz, Lapedes, & Xia 1992). With multi-layer networks it is possible to find arbitrarily complex mappings between the input and output.

In this paper we demonstrate the flexibility of neural network methods, combined with a Gibb's sampling procedure, for the problem of simultaneously determining the specificity of a DNA-binding protein and locating its binding sites in a set of fragments. The approach is expected to be easily extendible to problems of finding common domains in proteins or RNAs.

Specificity

The specificity of a DNA-binding protein is often described qualitatively as the "consensus sequence" or preferred binding site sequence. Specificity can also be defined quantitatively based on the relative binding affinities to all possible sites (Stormo & Yoshioka 1991; Stormo 1991). We need to use that quantitative definition because our neural network searches the space of binding parameters to find those which maximize the difference in binding energy between the DNA fragments given as the data in the problem and the rest of the genome of the organism. The remainder of this section defines specificity as it is used in the objective function of our neural network.

Assume that the DNA site that is recognized and bound by the protein is a fixed length of l bases; there are 4^l different sequences to which the protein could bind. We denote by $\{S_i \mid 1 \leq i \leq 4^l\}$ the set of all possible binding site sequences. When explicit representation of a sequence is required we use a matrix $\mathbf{S}_i$ of 0's and 1's with $S_i(b,m) = 1$ if base b occurs at position m of sequence i, for $b \in \{A,C,G,T\}$ and $1 \leq m \leq l$.

We want to determine the binding function $\mathbf{H}$ for the protein which will return the binding energy, H_i, to any sequence S_i. That is, we want to find $\mathbf{H}(S_i) = H_i$. We know such a function exists because, in the worst case, it can simply be the list of binding energies to all possible sequences, so that $\mathbf{H}(S_i)$ would return the ith value on the list. In general we hope to find a more "compact" function with adequate accuracy. An example would be a function which is additive over the positions of the binding site, such as used in models by Berg and von Hippel (Berg & von Hippel 1987) and Stormo (Stormo 1991). In these models each base at each position in a binding site contributes some partial energy to the binding, and the total binding energy is the sum of those partial binding energies. This is essentially the model used in the greedy and EM approaches described above; the neural network based analysis below subsumes this model and allows for more complicated models.

In the following, we will use two types of notation simultaneously, one based on thermodynamics and the other on Bayesian statistics. This is to accommodate readers familiar with one type but not the other, and to point out the close relationship between these approaches. The Bayesian notation will be in brackets for clarity.

Imagine an experiment in which a single molecule of the protein was present in a reaction mixture with each of the 4^l sequences. Further assume that the different sequences were present in proportions P_i. The probability that, at any moment when the protein was bound to a sequence, it was bound to sequence S_i is:

$$F_i = [P(S_i|B=1)] = \frac{P_i e^{-H_i}}{\sum_i P_i e^{-H_i}} \qquad (1)$$

$$= \left[\frac{P(B=1|S_i)P(S_i)}{P(B=1)}\right] = \frac{P_i e^{-H_i}}{Y} \qquad (2)$$

where B is a binary indicator with $B = 1$ being the bound state of the DNA, and $B = 0$ being the unbound state. F_i can also be considered to be the time-averaged fraction of the total binding that is occupied by sequence S_i. The terms in brackets are simply a statement of Bayes theorem for conditional probabilities. $Y = \sum_i P_i e^{-H_i}$ is the partition function over the distribution of sequences; it assures that $\sum_i F_i = 1$. It is important to note that in equation 1, F_i is the fraction of the time for which *some* sequence S_i will be bound. If there are multiple copies of sequence S_i in the experiment we may wish to know the fraction of time that a particular one of them will be bound. This would be the typical case in thinking about gene regulation; there may be many occurrences of a particular binding site sequence in the genome, but only one of them is appropriately located to regulate the expression of a particular gene. For example, imagine that a protein must bind to a particular site X, with sequence S_i, in order to regulate the expression of a particular gene. If the sequence S_i is common in the genome then the affinity for sequence S_i *and* the concentration of the protein must be high enough to assure the appropriate level of binding at X. We need to determine the amount of binding to particular binding sites compared to the background of all possible sites, including others with the same sequence:

$$\frac{F_i}{P_i} = \frac{K_i}{Y} = \left[\frac{P(S_i|B=1)}{P(S_i)}\right] = \left[\frac{P(B=1|S_i)}{P(B=1)}\right] \quad (3)$$

where, for convenience, we use $H_i = -\ln K_i$, which also means that $Y = \sum_i P_i K_i = [\sum_i P(B = 1|S_i)P(S_i)]$. K_i/Y measures the specificity of the protein for every sequence S_i. If we know that for every sequence and we know the distribution of sequences, P_i, we can calculate the fraction of time each sequence is bound to the protein, F_i. Additional information about the protein concentration would be sufficient to predict the occupancy of any site in the genome, with the caveat that we are ignoring complications due to cooperative interactions with other binding sites and other proteins.

In these definitions we have not specified a temperature. We imagine that the binding experiment is performed at a fixed temperature which merely contributes to the units of energy. However, if we wanted to apply this methodology to different temperatures, or even do an "annealing" procedure, we could replace H_i by H_i/T in all of the equations. We could also replace our function **H** by another **H′** such that $H_i' = H_i + c$ without affecting the results because the e^c term drops out. If we choose $c = \ln Y$ then $\sum_i P_i e^{-H_i'} = 1$ and equation 3 becomes

$$\frac{F_i}{P_i} = e^{-H_i'} = K_i' \quad (4)$$

That is, we are free to choose our "baseline" of energy such that the probability of any particular site being bound is simply the negative logarithm of that energy, K_i'.

Determining H and Identifying Sites

In the problem under consideration, the data provided is a set of fragments and the knowledge that each contains at least one good binding site for the protein; we are essentially informed that some site on each fragment has a high value of K_i/Y. Assume there are N fragments, each L long, and denote by s_{jk} the number of the site sequence which begins at position k of sequence j ($1 \leq j \leq N$ and $1 \leq k \leq L - l + 1$). For example, the sequence of the first possible site on the first fragment is $S_{s_{11}}$. We want to find a function **H** such that the there is a high probability of binding to fragment 1 *and* fragment 2 ... *and* fragment N. For simplicity, assume that we need only maximize the best binding site (*i.e.* the one with the highest probability) on each fragment. This means we should maximize

$$\prod_j \max_k \frac{K_{s_{jk}}}{Y} \quad (5)$$

or, instead, maximizing the logarithm

$$\ln \prod_j \max_k \frac{K_{s_{jk}}}{Y} = \sum_j [\max_k \ln K_{s_{jk}}] - N \ln Y \quad (6)$$

which is equivalent to minimizing

$$U = \frac{1}{N} \sum_j \min_k H_{s_{jk}} + \ln Y \quad (7)$$

This is the objective function for maximizing specificity of sites from each fragment. Note that minimizing U maximizes the difference in binding energy between the chosen sites and the genome as a whole. Our neural network actually uses a slightly modified procedure to avoid local minima. At each iteration, the binding energy $H_{s_{jk}}$, and corresponding probability, $P_{s_{jk}}$, are computed for each possible site on each fragment. Rather than choosing the minimum energy sites, however, a site from each fragment is sampled randomly according to the computed distribution of probabilities, as in Gibb's sampling (Geman & Geman 1984). We then adjust the network weights so as to minimize U. The simple models typically used with Gibb's sampling permit immediate extraction of the optimal parameters (weights) from a set of sampled sites. We wish to extend the method to models in which this is no longer possible, however. As a result, we use fixed-step-size, gradient descent with weight decay at each iteration. With this procedure, any binding function **H** yielding a finite, differentiable U is acceptable. In this paper, we use a neural network whose output models the binding strength $K_i = e^{-H_i}$.

The determination of Y, the partition function over the distribution of sequences, is also essential to the method. If the data provided are regulatory sites from a particular organism, then Y should be calculated from the genomic sequences of that organism. If one assumes that the genome is a random sequence with a particular composition, and that the binding energy is additive across the positions, then Y can be calculated analytically, and the solution maximizes Information Content, as shown below. If the sequence of the genome is known then Y could be determined directly from it, although this might be computationally expensive. Currently the *E. coli* genome is nearly 50% sequenced, and that large sample could be used to approximate the entire genome. Much smaller samples may also be sufficient providing they are good approximations to the genome. Below we illustrate this approach using a Markov model of the *E. coli* genome.

Templated Exponential Perceptron

The simplest neural network is a perceptron: a single, artificial neuron which applies a monotonic function to a weighted sum of its inputs (Minsky & Papert 1988). If a negative exponential function is chosen, the model becomes

$$H_i = \sum_m \sum_b W(b,m) S_i(b,m) = \mathbf{W} \cdot \mathbf{S}_i \qquad (8)$$

where $\mathbf{W}$ are the perceptron weights. This model is exact when each base position contributes linearly and independently to the binding energy.

Rather than using the classical perceptron training algorithm, we train the network to minimize $\mathcal{U}$, as described above. We use the term "templated perceptron" to indicate the idea that the perceptron represents the binding specificity of the protein; it can be thought of as "sliding along" the sequence and providing the binding energy to each possible binding site. Although the perceptron assumes linear, independent contributions to binding energy from each base, there is still more flexibility in this model than allowed with the strict requirement of independent base interactions used by the greedy and EM approaches. Requiring independent base interactions fixes the linear weights to specific values related to base frequencies. Training the network as described above, however, without the assumption of independence, allows the weights to assume values that optimize the objective function. This results in a perceptron capable of picking up some of the higher-order correlations in the data, and in having those correlations contribute to the linear terms of the model (Stolorz, Lapedes, & Xia 1992).

Training to minimize $\mathcal{U}$ limits the type of correlations which may be captured by the perceptron. Let the sequence number of the optimum site on fragment j be s_{j*}. With a templated exponential perceptron, minimizing $\mathcal{U}$ then becomes equivalent to minimizing

$$\begin{aligned} \mathcal{U} &= \frac{1}{N} \sum_i \mathbf{W} \cdot \mathbf{S}_{s_{j*}} + \ln Y & (9) \\ &= \mathbf{W} \cdot <\mathbf{S}_{s_{j*}}> + \ln Y & (10) \end{aligned}$$

$<\mathbf{S}_{s_{j*}}>$, the average binding site, is a matrix with elements $<S_{s_{j*}}> = \mathcal{F}(b,m)$ equal to the frequency with which base b occurs at position m of the sites. Since only the "average site" is visible to the perceptron, correlations between bases in the sites can not be exploited. This limitation can be overcome by replacing the perceptron with a multi-layer network. Even with a templated exponential perceptron and the training scheme described, however, correlations visible in the entire genome (*i.e.* those effecting Y) can be utilized. The Markov model of *E. coli* discussed below provides an example of such correlations.

Random Genomes

Consider application of the templated exponential perceptron to a collection of possible binding sites devoid of correlations, and a genome in which each sequence S_i occurs with the frequency which would be computed assuming independent, identically distributed bases with composition $p(b)$ at every position:

$$P_i = \prod_m \prod_b p(b)^{S_i(b,m)} \qquad (11)$$

We refer to such collections as random genomes, and they permit explicit calculation of Y. Letting $K(b,m) = e^{-W(b,m)}$

$$P_i K_i = \prod_m \prod_b [p(b) K(b,m)]^{S_i(b,m)} \qquad (12)$$

$S_i(b,m)$ acts as a "selector" such that only one value of $p(b)K(b,m)$ is used in the product for each position m. Summing over all sequences to compute Y, however, leads to terms for each possible base at every position. As a result

$$\begin{aligned} Y &= \sum_i P_i K_i & (13) \\ &= \sum_i \prod_m \prod_b [p(b) K(b,m)]^{S_i(b,m)} & (14) \\ &= \prod_m \sum_b p(b) K(b,m) & (15) \end{aligned}$$

From equation 10, minimizing U for a random genome therefore becomes equivalent to minimizing

$$\mathbf{W} \cdot <\mathbf{S}_{s_{j*}}> + \ln \prod_m \sum_b p(b) K(b,m) \qquad (16)$$

Alternatively, since we are free to choose the baseline of $\mathbf{H}$ such that $Y = 1$, we may minimize $\mathbf{W} \cdot <\mathbf{S}_{s_{j*}}>$ subject to the constraint

$$\prod_m \sum_b p(b) K(b,m) = 1 \qquad (17)$$

Applying a Lagrange multiplier, λ, to this constrained minimization we obtain

$$\mathcal{F}(b,m) = \lambda \frac{p(b) K(b,m)}{\sum_b p(b) K(b,m)} \qquad (18)$$

which has a solution

$$\begin{aligned} W(b,m) &= -\ln \frac{\mathcal{F}(b,m)}{p(b)} & (19) \\ \lambda &= \sum_b p(b) K(b,m) = 1 & (20) \end{aligned}$$

For random genomes, our approach is therefore equivalent to finding the alignment which maximizes the Information Content, as was done with the greedy algorithm (Stormo & Hartzell III 1989; Hertz, Hartzell III, & Stormo 1990).

Results

We have tested this approach on the same data that were used in the original greedy and EM papers (Stormo & Hartzell III 1989; Lawrence & Reilly 1990), which is a collection of DNA fragments, each 105 long, that contain one or more binding sites for the *E. coli* CRP protein. We refer readers to Figure 2 of (Stormo & Hartzell III 1989) to see the data used as input. In the first test we calculated Y exactly, using equation 15 and assuming the *E. coli* genome to be random and equimolar (*i.e.* $p(b) = 0.25\ \forall b$). We chose a binding site size of 22 bases and did multiple runs with pseudo-random initial weights ranging from -0.005 to 0.005. Correct identification of the sites is usually obtained, although it is not uncommon for all sites to be shifted one or two bases in either direction. Shifted solutions can arise due to local minima, and because the ends of the 22-base sites are less highly conserved than the 16-base central "core". The correct alignments all give essentially the same weights; these are similar to those obtained from the greedy algorithm but not identical because that algorithm used different values of $p(b)$ based on the composition of the fragments themselves.

In a second test we estimated Y by computing the binding to pseudo-random samples from a model of the *E. coli* genome. The model genome was again assumed to be random and equimolar. We used three different sample sizes (G): 1024, 2048 and 4096. This was done by generating a "genome" of $G+l-1$ bases and calculating Y as in equation 13 from this sample of G sites. A new sample was generated for each iteration of training. The results are somewhat more variable now, due to the variability in the "genome sample" used in estimating the partition function, but generally all of the correct sites are still found; again some solutions are shifted, but the correct solutions are easily obtained from those.

In a third test we estimated Y by sampling a more complicated model of the genome. It is known that the *E. coli* genome is not well modeled by a random sequence of the known composition, but that a third-order (or higher) Markov model represents it quite well (Phillips, Arnold, & Ivarie 1987). Therefore we generated sample genomes, sized as before, from a third-order Markov model and estimated Y from those samples. Estimation is now required, since Y can not be calculated directly using this Markov model of the genome. Again the correct sites are usually identified, some of them shifted. The fact that identical alignments were obtained in all three experiments is encouraging, and suggests that estimating Y by sampling is a viable strategy. This is important since analytical calculation of Y is possible only for the simplest models of binding, and simple, well-characterized genomes.

Table 1: Average Correlation Coefficients Between Solution Vectors

	A	N_{10}	N_{11}	N_{12}	M_{10}	M_{11}	M_{12}
A	1.00						
N_{10}	0.88	0.79					
N_{11}	0.91	0.81	0.83				
N_{12}	0.90	0.81	0.82	0.83			
M_{10}	0.89	0.81	0.80	0.82	0.81		
M_{11}	0.90	0.82	0.82	0.83	0.84	0.87	
M_{12}	0.91	0.82	0.82	0.82	0.84	0.86	0.85

Finally, we repeated the experiments described above using a site width of 16 bases. While local minima are more frequently encountered with this model, the shifted sites are no longer found among the lowest energy solutions; those are now always the correct, unshifted solutions.

Comparing the Solutions

Besides comparing the sites that are identified, which were usually the same in each test, we can compare the energy functions directly. The weights of each network, $W(b, m)$, define the energy function and describe a vector in the $4l$-dimensional space of l-long sequences. We can compare any two energy vectors by calculating the correlation coefficient between the two vectors, which in these examples is approximately equal to the cosine of the angle between them (because the sum of the weights in any solution is approximately 0). We collected the 5 lowest-energy, unshifted solutions from 100 runs of each test method and sample size. The average correlation coefficients between all of these solutions are shown in Table 1: A is the solution from calculating Y analytically; N_{10}, N_{11} and N_{12} are the non-Markov sampled solutions for sample sizes of 1024, 2048 and 4096, respectively; M_{10}, M_{11} and M_{12} are the Markov sampled solutions for the same sample sizes, respectively. The diagonal shows the amount of variability within each set of solutions. The 2048 size sample is more consistent (*i.e.* there is less variability between the solutions) than the 1024 size sample, but the further increase to 4096 did not provide greater consistency. The first column shows the similarity to the analytical solution. The fact that these numbers are higher than the correlations between different solutions from the same test indicates that each sampled solution is close to the analytical solution and that they form a neighborhood around it. The same conclusions are obtained from the solutions using a site size of 16 except that the correlations are all somewhat higher, showing more consistency among the various solutions.

An alternative comparison can be made between the

Table 2: Correlation Coefficients Between Average Solution Vectors

	A	N_{10}	N_{11}	N_{12}	M_{10}	M_{11}	M_{12}
A	1.00						
N_{10}	0.97	1.00					
N_{11}	0.98	0.96	1.00				
N_{12}	0.97	0.95	0.95	1.00			
M_{10}	0.97	0.96	0.94	0.96	1.00		
M_{11}	0.96	0.95	0.93	0.95	0.96	1.00	
M_{12}	0.97	0.96	0.94	0.95	0.98	0.97	1.00

average solutions from each test. The individual solutions from each test form a cluster of vectors in the space. The average solution from a test is "centered" in that cluster. By comparing the averages we are asking how close together those centers are. The results are shown in Table 2. The diagonal is now always 1.00 because these are correlations of single vectors with themselves, the average vector for each test. The fact that all of the cross correlations are higher than the averages shown in Table 1 demonstrates that the average solutions are quite similar to one another.

A more subtle, but equally important, fact is evident in Table 2. The average vectors from the different N_x tests still form a cluster with the analytical solution in the center; that is, each is more similar to the analytical solution than to the other N_x solutions. In fact, if one average vector is made from all of the N_x solutions it has a correlation of 0.99 with the analytical solution. On the other hand, the solutions from the Markov samples are as similar to each other as they are to the analytical solution. This means the analytical solution is not in the center of the neighborhood defined by those solutions, but rather they are in a nearby cluster with a distinct center. If a single average vector is made from all of the M_x solutions it maintains only a 0.97 correlation with the analytical solution, while it has a 0.99 correlation with each of the separate M_x average solutions. These results are only marginally significant statistically but are, at least, consistently true in our small data set. Furthermore they are what we expect since modeling the genome with Markov correlations has an effect on the frequency of different sequences and therefore on the partition function Y. So while the identified sites are the same, the energy function varies slightly in response to the change in Y. Remember that the form of all of the solutions is a matrix, $W(b, m)$, with the binding energy to any sequence, H_i, defined as the sum of the terms corresponding to that sequence, as in equation 8. The Markov-generated solution vectors incorporate some of the information from the genome correlations even though the model remains linear. The ability of a perceptron to include such terms from high level correlations into a linear model has been described previously (Stolorz, Lapedes, & Xia 1992).

Discussion

We have used a quantitative definition of specificity to develop a neural network for the problem of identifying the binding sites in common to a collection of unaligned DNA fragments. The weights of the network define an energy function which maximizes the discrimination between binding to sites on the collection of fragments from binding to the genome as a whole. For additive energy functions and position independent genomes this is equivalent to finding the set of sites with maximum Information Content. However, more complicated genome models can easily be used, as the Markov model used in this paper. Real genomic samples could also be used. More complicated energy models are also possible, such as with specific correlations between positions that would correspond to RNA structures. We could also use multi-layer neural networks which could identify unknown correlations that are important to the specificity of the sites. It is not clear at this time how such generalizations of the method will affect the sampling requirements, but the current results are encouraging. The method should also be extendible to the problem of finding common domains in protein sequences. Here the justification based on specificity arguments is not valid, but analogous arguments of finding the optimal alignment compared to all possible alignments can be used instead.

Acknowledgements

We thank Jonathon Arnold for providing us with Markov chain analysis data from *E. coli*. The authors thank the Santa Fe Institute where some of this research was performed. JMH thanks Hewlett-Packard for use of corporate resources in this research. ASL was supported by DOE/LANL funds. GDS was supported by grants from NIH (HG00249) and DOE (ER61606).

References

Baldi, P.; Chauvin, Y.; Hunkapiller, T.; and McClure, M. A. 1994. Hidden Markov models of biological primary sequence information. *Proc. Natl. Acad. Sci.* 91:1059–63.

Berg, O. G., and von Hippel, P. H. 1987. Selection of DNA binding sites by regulatory proteins. Statistical-mechanical theory and applications to operators and promoters. *J. Mol. Biol.* 193:723–750.

Cardon, L. R., and Stormo, G. D. 1992. An expectation-maximization (EM) algorithm for identifying protein-binding sites with variable lengths from unaligned dna fragments. *J. Mol. Biol.* 223:159–170.

Eddy, S. R., and Durbin, R. 1994. Analysis of RNA sequence families using adaptive statistical models. *Nucl. Acids Res.* 22:in press.

Geman, S., and Geman, D. 1984. Stochastic relaxations, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* 6:721–742.

Henikoff, S., and Henikoff, J. G. 1991. Automated assembly of protein blocks for database searching. *Nucl. Acids Res.* 19:6565–6572.

Hertz, G. Z.; Hartzell III, G. W.; and Stormo, G. D. 1990. Identification of consensus patterns in unaligned DNA sequences known to be functionally related. *Comput. Appl. Biosci.* 6:81–92.

Krogh, A.; Brown, M.; Mian, I. S.; Sjölander, K.; and Haussler, D. 1994. Hidden Markov models in computational biology: Applications to protein modeling. *J. Mol. Biol.* 235:1501–1531.

Lawrence, C. E., and Reilly, A. A. 1990. An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins* 7:41–51.

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

Minsky, M. L., and Papert, R. G. 1988. *Perceptrons: an introduction to computational geometry.* MIT Press.

Phillips, G. J.; Arnold, J.; and Ivarie, R. 1987. Mono- through hexanucleotide composition of the *Escherichia coli* genome: a Markov chain analysis. *Nucl. Acids Res.* 15:2611–2626.

Sakakibara, Y.; Brown, M.; Mian, I. S.; Underwood, R.; and Haussler, D. 1994. Stochastic context-free grammars for modeling RNA. In *Proceedings of the Hawaii International Conference on System Sciences.* IEEE Computer Society Press.

Stolorz, P.; Lapedes, A.; and Xia, Y. 1992. Predicting protein secondary structure using neural net and statistical methods. *J. Mol. Biol.* 225:363–377.

Stormo, G. D., and Hartzell III, G. W. 1989. Identifying protein binding sites from unaligned DNA fragments. *Proc. Natl. Acad. Sci.* 86:1183–1187.

Stormo, G. D., and Yoshioka, M. 1991. Specificity of phage P22 Mnt repressor with its operator, determined by quantitative binding to a randomized operator. *Proc. Nat. Acad. Sci.* 88:5699–5703.

Stormo, G. D. 1990. Consensus patterns in DNA. In *Molecular Evolution: Computer Analysis of Protein and Nucleic Acid Sequences, Methods in Enzymology, Vol. 183*, 211–221. Academic Press.

Stormo, G. D. 1991. Probing the information content of DNA binding sites. In *Protein-DNA Interactions, Methods in Enzymology, Vol. 208*, 458–468. Academic Press.