ANN-SPEC: A METHOD FOR DISCOVERING TRANSCRIPTION FACTOR BINDING SITES WITH IMPROVED SPECIFICITY

C.T. WORKMAN

Center for Biological Sequence Analysis,
The Technical University of Denmark
DK-2800 Lyngby, Denmark

G.D. STORMO

Department of Genetics
Washington University School of Medicine
St Louis, MO, 63110-8232 USA

This work describes ANN-Spec, a machine learning algorithm and its application to discovering un-gapped patterns in DNA sequence. The approach makes use of an **A**rtificial **N**eural **N**etwork and a Gibbs sampling method to define the **Spec**ificity of a DNA-binding protein. ANN-Spec searches for the parameters of a simple network (or weight matrix) that will maximize the specificity for binding sequences of a positive set compared to a background sequence set. Binding sites in the positive data set are found with the resulting weight matrix and these sites are then used to define a local multiple sequence alignment. Training complexity is $O(lN)$ where l is the width of the pattern and N is the size of the positive training data. A quantitative comparison of ANN-Spec and a few related programs is presented. The comparison shows that ANN-Spec finds patterns of higher specificity when training with a background data set. The program and documentation are available from the authors for UNIX systems. Contact: workman@cbs.dtu.dk

1 Introduction

Pattern discovery remains an active area of research in computational biology. The importance of biological information in primary sequence data is widely recognized but finding this information is difficult. Much of the information inferred from sequence data has come from sequence motif or homology studies using various alignment based methods. Most often a multiple sequence alignment is desired but finding the optimal multiple sequence alignment is hard. For general alignment cost functions, this problem is known to be NP-complete. ANN-Spec indirectly learns an un-gapped local multiple sequence alignment by a method related to Expectation-Maximization (EM). In this method, sampled alignments are used to fit a set of weights and the best weights are used to define an alignment. The work presented here is an extension of the alignment method presented in[1] and is closely related to the Gibbs sampling method of[2][3]. An early version of this program was tested on short *E. coli* promoters (100-

200 bps) and employed simple background sequence models, including Markov chains. The current implementation can use real background sequences which allows the method to find patterns of greater discriminatory capability when compared to the original version and other current methods.

1.1 Transcriptional Regulation

Much of gene expression is controlled at the transcriptional level by systems of transcription factors and regulatory proteins. These factors bind DNA sequence elements proximal to the transcription initiation site and modulate the expression of that gene. Discovering conserved patterns in promoter regions can correspond to known and possibly undiscovered regulatory factors and may implicate mechanisms of co-regulation. Given a set of promoter sequences each known to contain a common binding site, we wish to find both the binding sites and a model for the proteins binding specificity. Transcription factor proteins exhibit a range of specificities. Some bind a small number of highly conserved sequences but most bind a large variety of partially conserved sequences. Highly conserved motifs can be found by word frequency analysis[4][5] [6] while weakly conserved motifs are often impossible for these methods to find [7]. The approach presented here is designed to find weakly conserved signals as well as the highly conserved ones.

New mRNA expression analysis techniques implicate sets of coregulated genes with little to no information about regulatory mechanism. Methods for discovering functional sequence elements could greatly expedite experimental identification of new regulatory genes. A few methods exist for sequence element discovery that are not based on word frequency analysis: MEME[8], CONSENSUS[9][10] and Gibbs sampler[2]. The objective functions for each method are similar, maximizing likelihoods or likelihood-ratios, but the methods for searching the space of possible alignments are very different. CONSENSUS is based on a greedy strategy that progressively adds sub-sequences to a set of alignments where each iteration extends a bounded number of partial alignments. MEME is an EM method that considers all sites of the training data simultaneously and over iterative training converges to a local maximum. The Gibbs sampler is a stochastic variant of the EM method. In this method a single sequence contributes a site to the alignment based on a weighted sampling procedure relying on site scores from the alignment of the previous iteration. Again final results are obtained through iterative training. In this work we compare the performance of ANN-Spec to CONSENSUS, MEME and the Gibbs sampler on simulated data sets.

2 Algorithm

2.1 Statment of the Problem

Given a positive sequence set, $\mathcal{S}$, and a background sequence set, $\mathcal{G}$, possibly representing the genome, we wish to find the model parameters that describe a DNA-binding protein with highest specificity and sensitivity for $\mathcal{S}$. The objective is to maximize the probability that each of the n sequence regions of $\mathcal{S}$ are bound by the protein.

2.2 Simple Neural Network

The neural network used in this approach is a sparsely encoded perceptron with one processing unit. Since there is a perceptron weight for each different nucleotide at each position in the pattern, a single linear perceptron has a set of weights, Ω, with the well recognized representation of a weight matrix[11][12]. A linear model, like a perceptron or weight matrix, has been found to be a good model for the binding energy of DNA-binding proteins[13][14][15]. For these proteins, the total binding energy is well approximated by the sum of partial binding energies at each nucleotide of the binding site. The perceptron or weight matrix defines a score related to the binding energy at each nucleotide in the pattern. The perceptron first calculates the linear sum of weights times their input values by the simple function $h_j = H(\Omega, X_j)$;

$$h_j = \sum_{k=0}^{l-1} \sum_{b=1}^{|B|} \omega_{k,b} x_{j+k,b} + \beta \tag{1}$$

where j specifies the offset into a sequence, and k ranges over the l positions in the pattern. Index b ranges over the alphabet ($|B| = 4$ for DNA) and both $j + k, b$ index the pattern matrix X_j ($x_{k,b} \in \{0, 1\}$). The bias term β, can be shown to contribute a constant factor to the final network output and can be set to zero without loss of generality.

2.3 Perceptrons estimate posterior probabilities

Perceptrons are linear discriminant functions which can be used to estimate posterior probability distributions. We assume the Maximum Entropy (MaxEnt) distribution for the most likely distribution for interacting molecules with energies E_α. MaxEnt[a] defines a relationship between the binding probability and $\exp(-E_\alpha)$. If we use the linear output of the perceptron to approximate

---

[a]MaxEnt is also known as the Maxwell-Boltzmann distribution.
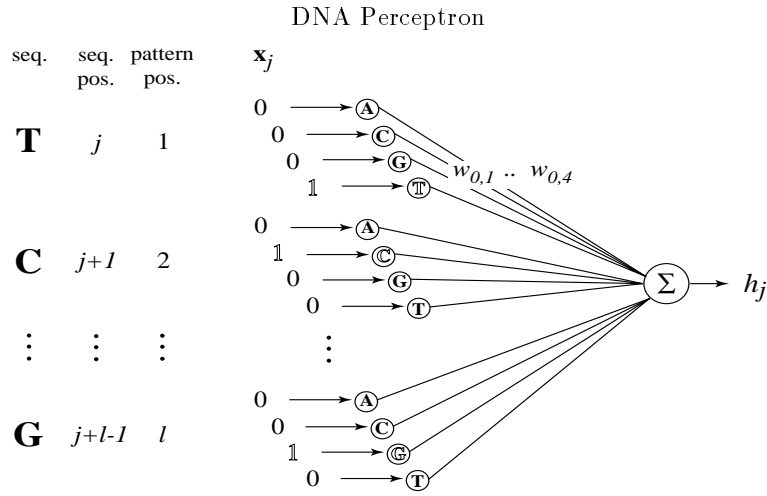
DNA Perceptron

Figure 1: Schematic representation of the DNA perceptron. Input is sparsely encoded giving 4 input nodes per pattern position. The weights of the perceptron define a weight matrix.

$-E_\alpha$ then $\exp(h_\alpha)$ can be used to estimate the MaxEnt distribution. Classification of the binding site can be viewed as a two class problem; at any given time a particular sub-sequence or word, S_α, is either bound or not bound. Given any l long sequence S_α presented to the perceptron as a binary matrix X_α, model parameters Ω and binding classes $C \in \{0, 1\}$, the perceptron estimates $P(C = 1|S_\alpha, \Omega)$.

By Bayes' theorem and the MaxEnt distribution we get the conditional probability for a binding site as

$$P(S_\alpha|C = 1, \Omega) = \frac{P(S_\alpha)P(C = 1|S_\alpha, \Omega)}{\sum_\beta P(S_\beta)P(C = 1|S_\beta, \Omega)} = \frac{(g_\alpha)e^{(-E_\alpha/\mathcal{K}\mathcal{T})}}{\sum_\beta (g_\beta)e^{(-E_\beta/\mathcal{K}\mathcal{T})}} \qquad (2)$$

The degeneracy term g_α is proportional to $P(S_\alpha)$ the probability of observing S_α, which is independent of Ω (i.e. we assume $P(S_\alpha|\Omega) = P(S_\alpha)$). Note that S_α may exist $g_\alpha > 1$ times in the genome. The likelihood that *a particular instance of* S_α is bound is given by $P(S_\alpha|C = 1, \Omega)/g_\alpha$ as each of the g_α instances are assumed to be equally likely. The binding probability at a particular site $S_{\alpha,k}$ is given by;

$$P\left(S_{\alpha,k}|C=1,\Omega\right) = \frac{P\left(S_\alpha|C=1,\Omega\right)}{g_\alpha} \approx \frac{e^{h_\alpha}}{\sum_\beta (g_\beta) e^{h_\beta}} = \frac{e^{h_\alpha}}{Z} \tag{3}$$

The partion function, Z, is the sum over all sites in the genome so β will range over the observable l long sequences. Equation 3 is the probability for a single binding site. We want to know the likelihood that one of our sequence *regions* is bound. This is simply the union or sum of individual site probabilities over the sequence region.

$$P(\mathbf{S}_i|C=1,\Omega) \approx \frac{\sum_{j=0}^{m_i} e^{h_{i,j}}}{Z} = \frac{Q_i}{Z} \tag{4}$$

Here i is an index of the positive sequence set, j ranges over the m_i positions in sequence region $\mathbf{S}_i$. An i,j pair specifies a particular l long sequence and its location in the genome (i.e. there exists a mapping $f : S_{\alpha,k} \to S_{i,j}$). A negative h will give $\exp(h)$ close to zero and only the sites that select positive weights will give $h > 0$ and significant perceptron outputs contributing to Q.

If we wish to maximize the sensitivity and specificity of our parameters for our positive sequences, then we should also maximize the probability all n of our sequences are classified as binding regions and classifying as few of the background sequences possible. This is the same as maximizing the joint probability over the n sequence regions assuming we have only n proteins.

$$P(\mathcal{S}|C=n,\Omega) \approx \prod_i \frac{Q_i}{Z} \tag{5}$$

Notice that we do not multiply by the product of $(1-Q_j/Z)$ over the negative set. This allows the method to be rather insensitive to "false positives" during training. Maximizing the product of likelihoods is the same as maximizing the sum of log-likelihoods or the mean log-likelihood.

$$U = \frac{1}{n}\sum_i ln\left(\frac{Q_i}{Z}\right) = \langle ln(Q_i)\rangle - ln(Z) \tag{6}$$

2.4   Parameter Fitting

The goal of training is to fit the free parameters, Ω, to the training data. In supervised training, one would have data consisting of a set of positive patterns (binding sites) and a set of negative patterns (non-binding sites). In our case, only the binding and non-binding regions are known while the positions of the sites are "missing data". As done in the EM algorithm, one option is to

consider all sites of the positive data as potentially binding. This would be the case here if we tried to optimize the objective function as described. For our gradient descent the gradient is derived from a modified version of Eq. 6.

First the weights are initialized to represent a randomly sampled site in $\mathcal{S}$. Alternatively, the weights may be initialized by a user-provided matrix. At each training iteration, the perceptron scores all sites of $\mathcal{S}$ and uses the $\exp(h)$ distribution for an individual sequence to Gibbs sample k sites. When Gibbs sampling, if $\exp(h_{i,j})/Q_i = 0.90$ then site j has a 90% chance of being sampled.

The gradient that is used is the derivative of the new objective U^* calculated from the alignment sites A and a Q_i^* based only on the sites in A. This gives the gradient a strong bias for the Gibbs sampled sites. As a result of the frequent sub-optimal samplings, the perceptron is able to wander through the different patterns found in the positive sequences. Patterns that are conserved in the positive sequences and have a low background frequency will be sampled more often and be learned by the weights. The partition function estimates the expected number of binding regions. When $ln(Z)$ is much larger than n, then the current pattern is expected to occur often in the background and would result in poor binding specificity. In this implementation, Z is either estimated based on a sampling of $\mathcal{G}$ or calculated analytically assuming a random background. When estimating Z, a fixed number of sites proportional to the size of $\mathcal{S}$ are sampled at random. A new random sampling is done for each training iteration. For each alignment and estimate of Z, a weight decay is applied and the weights are moved a fixed step size in the direction of the gradient. The weight change is specified as;

$$\Delta\Omega = \eta \left(\frac{\partial U^*}{\partial \Omega} \right) - \lambda \Omega \qquad (7)$$

where η is the learning rate or step size and λ is the decay rate. Convergence criteria are not practical due to the sampling procedure. Instead, a fixed number of training iteration are performed and the best answer is reported based on the objective in Eq. 6. The training performs an extensive search of the parameter space, but probably only comes close to the true local optimum. If desired one could start from the best answer and do a local search to refine the parameters to that optimum.

The training time scales linearly with the input data N and more specifically by $O(lN)$ where N is the size of $\mathcal{S}$. The most expensive steps, scoring the positive set and estimating the partition function, are $O(lN)$ each. The analytical partition estimate is only $O(l|B|)$ but this does not change the asymptotic complexity. Typically, on the order of 10^3 iterations are performed per epoch

and ten to a hundred epochs are recommended, for a total of $\approx 10^5$ iterations. In practice the constants associated with training are significant and fast processor speeds and long runtimes may be required for thorough analysis.

3   Methods

3.1   Programs and Data

For the comparison the newest version of each program was obtained: version 6 of CONSENSUS, version 2.2.2 of MEME and an unreleased version 1.01.009 of Charles Lawrences' motif sampler (we will call Gibbs).

For a given number of sequences, n, and $m = 500$ nucleotides, ten different random data sets with the biased nucleotide priors of yeast (32% A T, 18% C G) were generated. A simple random sequence generator was used which gave "zero order" randomization [b]. In each data set, a site was implanted in each sequence. Individual sites were variations of a common consensus sequence. Each position of the aligned sites was assigned a random mutation probability which caused some positions to be more conserved than others. The alignment was mutated a fixed number of times based on a "mutation rate", r, where the number of substitutions was defined as $\lfloor rnl \rfloor$.

Substitution probabilities were defined by the nucleotide priors and reflexive substitutions were disallowed. Eight different mutation rates were applied to each of the ten different alignments giving 80 alignments. The value of r was varied from 0 to 0.4 to provide a range of pattern conservation from completely conserved to very noisy. The mutated instances of each alignment were substituted into the sequence sets at random positions and orientations so that each sequence was given an instance. Positive data sets were generated in this way for $n = (10, 20, 40, 80)$.

Background data were generated in two ways. The simplest was to create random sequences with the same priors as the positive data sets. We also wanted to test the ability of the program to identify patterns that occur only in the positive data sets even when another common pattern exists in both the positive and background sets. In that case we implanted an initial common pattern in both positive and background sets, following the same procedure as described above for the positive pattern. Therefore, the non-random background contains a strong bias for one motif but is otherwise random. The common signal was not necessarily a low complexity pattern and could be discriminatory if it were not present in all regions. A non-random background data set consisting of 1000 sequences of 500 base pairs was generated. Positive

---

[b]That is to say, no dinucleotide or higher order biases were imposed.

data sets were then built from this non-random model that also contained a second pattern with varying mutation rates as before. Care was taken that the specific implant did not overlap the originally inserted pattern.

3.2 Training

Each algorithm was run/trained assuming one occurrence per sequence: option "-N" for CONSESNSUS, "-mod oops" for MEME, "site sampler" option for Gibbs. The stochastic methods, ANN-Spec and Gibbs were run ten times for each data set and the top ranked answer was selected. Deterministic methods, CONSENSUS and MEME, were allowed to report ten results and the top ranked (usually the first result) was taken. Since all methods report a different statistic, it was convenient to compare the information content (IC) [9] [13] of the resulting alignments. The information content calculation implicitly assumes a random background sequence model and therefore is appropriate in this case. The IC of each discovered n membered alignment can then be compared to the IC of the alignment known to be contained in that data set. The first phase of this comparison allows the methods to assume random background sequence. This is the only current option for Gibbs and CONSENSUS but ANN-Spec and MEME allow training with a background data set. A second comparison was done on the simple non-random data sets for Gibbs, MEME, and ANN-Spec allowing MEME and ANN-Spec to train with the non-random background data. For each prediction from non-random data, all specificities, sensitivities and correlation coefficients were calculated. For example, given one alignment matrix, all n values of specificity and sensitivity were calculated using n different log-likelihood score thresholds to determine Tp, Tn, Fp, Fn counts. We used a site score threshold for each $Tp = (1..n)$ and assumed $(Tp \cup Fn) \in \mathcal{S}$ and $(Tn \cup Fp) \in \mathcal{G}$. Both IC and specificity statistics were needed to assess results on non-random data.

4 Results

4.1 Assuming a Random Background

For each data set size, the best 80 resulting alignments were analyzed for information content. Figure 2 compares the resulting IC from the predicted alignment to the expected IC of the inserted alignment. Points below the diagonal line are runs finding an alignment with less information than known to exist. From this set of plots we see that ANN-Spec and Gibbs compare well while CONSENSUS and MEME show a significant number of failures. Gibbs does perform slightly better than ANN-Spec and this may be due to
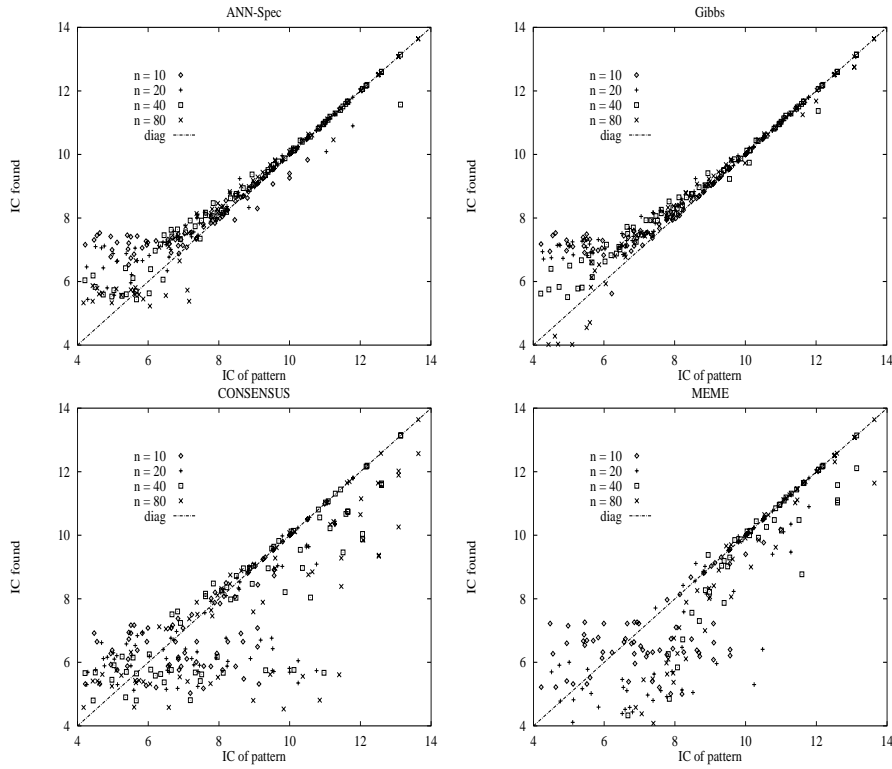
Figure 2: Plots of the discovered information content (IC) versus the expected IC for each method. Discovered ICs were calculated from the top ranking alignment predicted from each data set. These plots show the correlation to the expected ICs calculated from the inserted alignments. Each plot shows all data points for $n = 10, 20, 40, 80$ sequences.

its ability to correct the pattern frame during training. By inspection it was found that most of the high IC results falling just below the diagonal were correctly predicted patterns in the wrong frame (i.e. plus or minus one nucleotide position). ANN-Spec would require more training epochs to learn the correct pattern frame in these cases. CONSENSUS often found a higher IC on alignments with less than n sites. These results were forced to include the best sites from the remaining sequences but this often reduced the IC significantly. Interestingly, results in the low IC range tend to drift above the diagonal due to the low IC of the inserted pattern. In these cases the inserted pattern contained less information than what could be found by random chance[16].
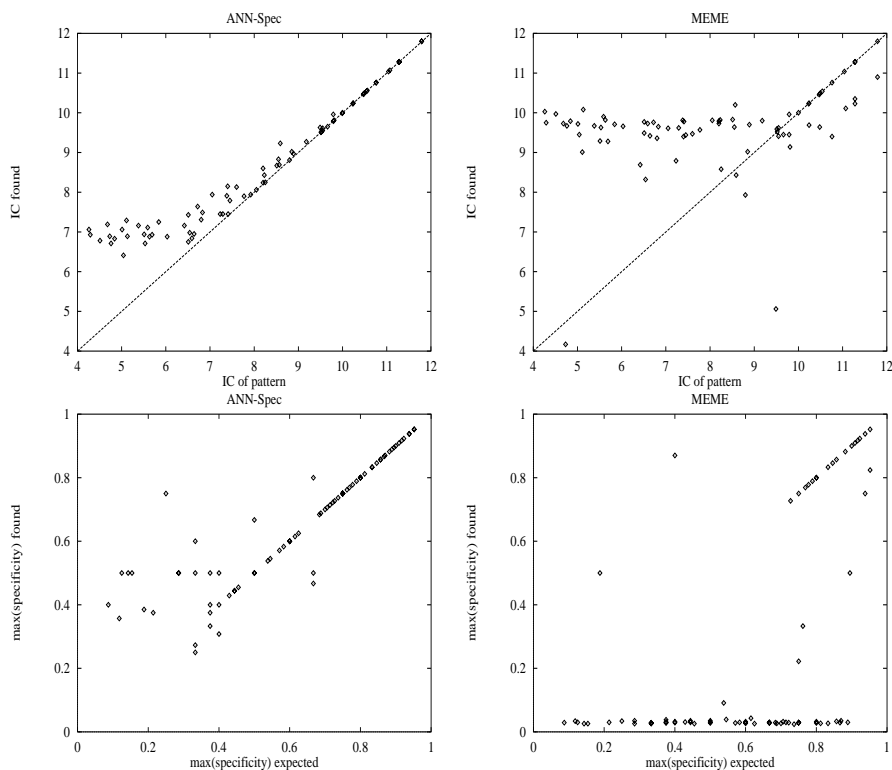
Figure 3: Results obtained with the non-random background sequence data. The top two plots show discovered IC versus the expected IC for the non-common pattern. Here ANN-Spec (left) is shown to find less information than MEME when training with a non-random background (the desired result). The bottom two plots show the maximum specificity found versus the maximum specificity expected from the same alignment models. The MEME results show that high IC does not necessarily correlate to high specificity. Each non-random positive data set contained 20 sequences.

4.2 Non-random Background

One of the goals of ANN-Spec is to find a discriminatory signal as well as one with good information. Does ANN-Spec or MEME find better discriminatory patterns with knowledge of the background? What if a pattern with a higher specificity has a lower information content? The non-random data sets were designed to answer these questions. Predictions for the non-random data sets were analyzed in the same way as the single implant results. The expected IC

was taken from the second implant that was generated with varying mutation rates. A comparison of ANN-Spec to MEME is presented in figure 3. The top two plots are like those of figure 2. Recall that each alignment generated up to n specificities one for each score threshold. The additional plots show the correlation of the maximum specificity from the predicted alignment to the maximum specificity of the varying information alignment known to exist. The results for Gibbs were not found to be significantly different from those of MEME and are not shown. It should be noted that this analysis was not fair to Gibbs as it did not train against the background data. MEME appears to always find the pattern with the most IC, even when that pattern is not specific for the positive set of sequences. The sites implanted in both the positive and background sets had an IC of about 10 (log base e "bits"), and was consistently found unless the discriminatory pattern had a higher IC. But the specificity of that pattern is quite low because it occurs in both the positive and negative sets. The two lower plots show that ANN-Spec, on the other hand, was able to identify the discriminatory pattern consistently, even when it had lower IC than the background pattern. Similar plots for maximum correlation coefficient show the same trend (data not shown). Because the objective function for ANN-Spec is designed to find patterns that distinguish the positive set from the background, it succeeds at identifying the desired patterns specific for the positive set.

5 Discussion

A serious limitation of pattern finding methods arises when low-complexity patterns are present with a high frequency in the sequence data of interest. This is the case in yeast promoter regions where ploy-A or poly-T are observed with a higher frequency than would be expected from the nucleotide composition alone. This means a conserved alignment of poly-A/poly-T can be found in almost any large set of yeast promoters where each site may align equally well in many frames. Methods that try to optimize IC alone and assume a random background model will have difficulty finding discriminatory patterns in biased data like that of yeast. But almost all classes of biological sequence data are known to be biased and not just by low complexity sequences. This work shows that even a simple pattern bias and not necessarily a low-complexity bias is enough force these methods into finding low specificity patterns.

In conclusion, Gibbs and ANN-Spec both work very well when the background is assumed to be random. ANN-Spec finds patterns with higher specificity and higher correlation coefficients when provided with background sequences. These results complement previous results on real yeast promoters[7]

where the transcription factors were known and all yeast promoters were used as background data.

Acknowledgments

References

1. Heumann,J.M., Lapedes,A.S., Stormo,G.D., ISMB, 2, 188-94 (1994).
2. Lawrence,C.E., Altschul,S.F., Boguski,M.S., Liu,J.S., Neuwald,A.F., Wootton,J.C., *Science*, 262(5131):208-14 (1993).
3. Neuwald, A.F., Liu, J.S., Lawrence, C.E. , *Prot. Sci.*, 4(8):1618-32 (1995).
4. Pesole G., Prunella N., Liuni S., Attimonelli M., Saccone C., *Nucl. Acids Res.* 20(11):2871-5 (1992).
5. van Helden,J., André,B. and Collodo-Vides,J., *J. Mol. Biol.*, 281(5):827-42 (1998).
6. Brazma,A., Jonassen,I., Vilo,J., Ukkonen,E., *Genome Res* 8(11):1202-15 (1998).
7. Workman, C.T., Stormo, G.D., *Nucl. Acids Res.* (submitted).
8. Bailey,T.L. and Elkan,C.P., ISMB, 2, 28-36 (1994).
9. Stormo,G.D. and Hartzell,G.W.III, *Proc. Natl. Acad. Sci.* USA, 86(4):1183-7 (1989).
10. Hertz,G.Z., Hartzell,G.W.III and Stormo,G.D., *Comput. Appl. Biosci.*, 6(2):81-92 (1990).
11. Stormo, G.D., Schneider, T.D., Gold, L. and Ehrenfeucht, A., *Nucl. Acids Res.* 10(9):2997-3012 (1982).
12. Stormo, G.D. *Ann. Rev. of Biophys. and Biophys. Chem.* 17:241-63 (1988).
13. Stormo,G.D. and Fields,D.S., *Trends Biochem. Sci.* 23(3):109-13 (1998).
14. Berg, O.G. and von Hippel, P.H., *J. Mol. Biol.*, 193(4):723-50 (1987).
15. Mulligan M.E., Hawley D.K., Entriken R., McClure W.R., *Nucl. Acids Res.* Jan 11;12(1 Pt 2):789-800 (1984).
16. Hertz, G.Z. and Stormo, G.D., *Bioinformatics*, 15(7):563-77 (1999).