Pacific Symposium on Biocomputing 5:389-400 (2000)

STOCHASTIC HEURISTIC ALGORITHMS FOR TARGET MOTIF IDENTIFICATION (EXTENDED ABSTRACT)

H.T. WAREHAM, T. JIANG, X. ZHANG
Department of Computing and Software, McMaster University, Hamilton, ON, L8S 4L7 Canada

C.G. TRENDALL
Department of Computer Science, University of Toronto, Toronto, ON, M5S 3G4, Canada

Target motifs are motifs that are "close" to one or more substrings in each sequence in one given set of sequences but are far from every substring in another given set of sequences. Target motifs have pharmaceutical applications; unfortunately, the problem of identifying target motifs is NP-hard and is thus unlikely to have efficient optimal solution algorithms. In this paper, we propose a set of simple modifications to the Gibbs Sampling heuristic for finding motifs which allows this heuristic to detect target motifs. We also present the results of several experiments relative to both simulated and real datasets which suggest that this modified heuristic is good at detecting target motifs under a variety of conditions.

1 Introduction

An important problem in computational biology is that of finding a *motif* for a given set of strings, *e.g.*, finding a string that is "close" (relative to some distance measure d on pairs of strings) to one or more substrings in each of the sequences in the given set. Some applications require motifs that are close to one or more substrings in each sequence in one given set of sequences (*target sequences*) but are far from every substring in another given set of sequences (*avoidance sequences*). Such *target motifs* can be used in the design of group-specific diagnostics, *e.g.*, diagnostics for the presence of pathogenic strains of *E. coli* in an environment known to contain multiple strains of *E. coli*, or as candidate sequence targets in the design of broad spectrum drugs, *e.g.*, antibiotics that disrupt particular genes or gene products in pathogenic strains of *E. coli* while leaving the corresponding genes and gene products in human beings and normal human-internal microbial flora intact. These applications and others are discussed in more detail in Lanctot *et al.* [5]

As the problem of identifying target motifs is NP-hard in the restricted case when motifs are encoded as strings and the distance function is Hamming distance,[5] target motif identification is probably NP-hard in general, and hence does not have polynomial-time optimal-solution algorithms unless $P = NP$.

Research should therefore focus on polynomial-time approximation algorithms and heuristics. Though a number of heuristics are known for identifying motifs, no heuristics have been proposed for target motif identification. One can trivially derive such heuristics by embedding a motif-finding algorithm that operates on the given target sequences in a postprocessing loop which terminates only when the produced motif is far from every substring of the the given avoidance sequences. However, such naive algorithms are problematic aesthetically, because they do not integrate the avoidance sequence information into the search for target motifs, and operationally, because they may be misled into returning a strongly-conserved non-target motif instead of a weakly-conserved target motif.

In this paper, we suggest some simple modifications to the Gibbs Sampling motif-finding heuristic[6,7,11,14] that use avoidance sequence information to help detect target motifs. This modified heuristic runs in low-order polynomial time and is the only known algorithm that can detect gapped target motifs. We also present the results of several experiments on simulated and real datasets which suggest that this modified heuristic is good at detecting target motifs under a variety of conditions.

• **Terminology** A *motif* is essentially a pattern that occurs in one or more given sequences. A motif can be encoded as a string, an alignment of strings, or a profile, *i.e.*, a table giving the probabilities of occurrence of all symbols in the sequence-alphabet at each position in the motif (see Gusfield[2] and references). A motif *matches* a string if the distance between the motif and the string is below a specified threshold value. When motifs are encoded as strings, two popular matching functions are *Hamming distance* (which is the number of positions at which symbols differ in two strings of equal length) and *edit distance* (which is essentially the minimum number of symbol substitutions, insertions, and deletions that must be applied to transform one of the given strings into the other). A motif that matches one or more substrings of a given sequence is said to *appear* in that sequence.

A motif can allow gaps, which correspond to positions at which insertions or deletions can occur in the matching of that motif to a given string. These gaps can be explicit in the motif itself (either as special symbols in a string or profile or as the gaps in an alignment of strings) or implicit in that motif's associated matching function, *e.g.*, edit distance. If a motif incorporates gaps, it is a *gapped motif*; else, it is an *ungapped motif*.

Finally, in the target motif identification problem, a motif that appears in both the target and avoidance sequences is a *non-target motif* and a motif that appears only in the target sequences is a *target motif*.

• **Previous Results** Ito *et al*[4] gave a low-order polynomial-time algorithm for the target motif identification problem when the motif is a string, the matching function is edit distance, and the target motif is required to be a substring of each of the target sequences. Lanctot *et al* [5] considered the case in which the motif is a string, the matching function is Hamming distance, and all target sequences are of the same length as the motif. They showed that this problem (which they called DISTINGUISHING STRING SELECTION) is NP-hard and gave a high-order polynomial-time approximation algorithm that is guaranteed to produce a motif whose distance to the target sequences is within a factor of 2 of optimal.

2 Gibbs Sampling Algorithms for Motif Identification

Gibbs Sampling is essentially a general stochastic strategy for determining the parameters of a statistical model relative to a given data set.[6,7] This strategy starts with some setting of parameter-values and iteratively changes the value of one parameter at a time by assuming that the remaining parameters are correct and invoking Bayes Theorem until all parameters converge to stable (if not optimal) values. With reference to the motif identification problem, the model is a motif encoded as an alignment of strings, the parameters are the positions of the motif within each sequence in a given set S (the *motif-instances*), and the stochastic heuristic modifies these motif-instances one sequence at a time, one sequence per iteration, until the alignment of these motif-instances denotes a stable (if not optimal) motif (see Table 1).

Several Gibbs-based algorithms for identifying ungapped [6,7,11] and gapped [6,14] motifs have been described in the literature. These algorithms follow the outline given in Table 1, and differ only in their conception of what a motif model is and how such models are used to weight motif-instances in Step 7. Lack of space precludes giving algorithm details here; interested readers should consult the original papers. Though each iteration of the main **while** loop in each of these algorithms runs in low-order polynomial time and space, no upper bounds on the number of iterations is known, and hence no worst-case time complexity can be given for these algorithms. All of these algorithms use relative entropy[a] based measures to evaluate candidate motif-instances. Intuitively, these measures assess the distinctness of the collective symbol-occurrence probabilities of the derived motif from background sequence symbol-occurrence probabilities, and allow the algorithms to detect the strongest motif signals.

[a] Given two probability distributions Q and P over a domain X, the *relative entropy of Q to P* (also known as the *Kullback-Leibler distance between Q and P*) is $H(Q||P) = \sum_{x \in X} Q(x) \log Q(x)/P(x)$.

Table 1: Generic Gibbs Sampling Algorithm for Motif Identification.

```
Input:   A set $S$ of sequences over some alphabet.
Output:   A candidate motif for $S$.
begin
1. Stochastically select initial motif-instances in the sequences of $S$.
2. Create initial motif-model from motif-instances.
4. while  not finished do
5.     Select sequence $s$ from $S$.
6.     Construct motif-model from motif-instances in $S - \{s\}$.
7.     Weight all possible motif-instances in $s$ relative to the motif-model
         derived above.
8.     Stochastically select a new motif-instance $x$ for $s$ relative to
         these weights.
9.     Update motif-instance information for $s$ relative to $x$.
10.     Check if motif-model has converged and process is finished.
11. Output motif-instances for $S$.
end
```

3 Gibbs Sampling Algorithms for Target Motif Identification

Two obvious points in the algorithm given in Table 1 at which avoidance sequence information can be used to influence target motif search are Steps 1 and 7. To implement this influence, we need an easily-computable measure of how much a candidate motif-instance is like the substrings of the avoidance sequences (and hence how fervently this motif-instance must be avoided). The most obvious measure is the score of the best alignment of the motif-instance against the avoidance sequences under an appropriate distance function. Although this approach is rigorous and appropriately values approximate matches, it can be computationally prohibitive. An alternative approach is to describe substrings of avoidance sequences in terms of the parameters of a statistical model. This results in a loss of information but a corresponding increase in computational efficiency.

In this paper, we use the first-order correlation model θ_A, which encodes the probabilities $P(a|b)$ of symbol a occurring in the avoidance sequences, given that symbol b occurred immediately before symbol a; in practice, $P(a|b)$ is approximated by the frequency of occurrence of substring ba in the avoidance sequences. For a string $x = x_1 \ldots x_n$, the weight of an instance of an ungapped motif relative to θ_A is $P(x|\theta_A) = \prod_{i=1}^{n-1} P(x_{i+1}|x_i)$. To evaluate such a model against an instance of a gapped motif, treat each insertion as matching all symbols, ignore all deletions, and multiply the appropriate probabilities as before. In the case of Step 1, each of the potential motif-instances is weighted relative to θ_A and motif instances can be selected stochastically relative to these weights. In the case of Step 7, the weight w_x associated with a possible motif-instance x is weighted by the odds that this motif instance doesn't appear in the avoidance sequences, *i.e.*, $w_x \cdot \frac{1-P(x|\theta_A)}{P(x|\theta_A)}$ (this assumes that the events of the motif model being close to the target sequences but distant from the avoidance sequences are independent, which seems to be a reasonable assumption).

A final modification is necessary, as distance-thresholds are not explicitly evaluated in the algorithms considered here and it is more important that a produced motif be the required distance from all substrings of the avoidance sequences. To ensure this, the algorithm as modified above is embedded in a postprocessing loop that terminates only when either the number of iterations of the postprocessing loop exceeds a user-defined bound B_R or the consensus string associated with the produced motif has distance greater than some threshold T_P to every substring in the avoidance sequences, where this distance is Hamming distance for ungapped motifs and edit distance for gapped motifs. In all experiments described in remainder of this paper, B_R was set to 10 and T_P was set such that no produced motif had more than 60% similarity to any substring in the avoidance sequences.

4 Results

4.1 Preliminaries

Both the Rocke and Tompa [14] algorithm and the version of this algorithm modified as described in Section 3 above were implemented in a set of C/C++ programs. The naive approach to finding target motifs described in the introduction was also implemented by embedding the Rocke and Tompa algorithm in the postprocessing loop described in Section 3.

Each dataset used in the experiments consists of a set of target sequences, a set of avoidance sequences, a set of non-target motifs, and a set of target motifs. In the simulated datasets, each motif has an associated randomly generated "seed string" and copies of that motif are created by performing a specified number of operations at random on that motif's seed string, where this number is the product of a specified *motif dispersion rate* and the length of the motif. In the case of ungapped motifs, all such operations are substitutions; in the case of gapped motifs, 60% of the operations are substitutions, 20% are single-symbol insertions, and 20% are single-symbol deletions. Note that the symbol insertion and substitution probabilities are equal.

Given datasets constructed as described above, we know the locations of all possible motifs and can hence unambiguously determine when motif detection does and does not occur. Given a set of sequences S, the start and stop positions of a motif m in each sequence in S, and the start and stop positions of a candidate target motif m_c in each sequence in S, define the *average overlap of* m_c *relative to* m *in* S as the average of the overlaps of m and m_c in each sequence in S. We will say that m_c *detects* m in S if the average overlap of m_c relative to m in S exceeds a threshold T_D. In all experiments reported in this section, T_D was set to 50% of the motif length.

Finally, let each execution of an algorithm on a dataset be a *run*, and each set of runs relative to a particular dataset be a *trial*.

4.2 Simulation Study #1

• **Motivation** The motif-finding algorithms underlying the naive method for target motif identification described in the Introduction should consistently prefer the best-conserved motif in the target sequences, even if that motif is a non-target motif. How often does this happen and under what conditions does it start causing serious problems for the naive method (and hence validate methods like the one proposed in this paper)?

• **Methods** Each dataset consisted of two motifs (one target) of length 20 which were embedded in 20 target sequences and 2 avoidance sequences of base length 60 (which had post-embedding lengths of 100 and 80, respectively). The non-target motif is always perfectly conserved and the target motif has a specified dispersion rate. Each trial consisted of 10 runs on a particular dataset, and the number of runs the target motif is detected in each trial was recorded. The naive and modified Gibbs algorithms were run for 100 trials apiece on protein and DNA sequence datasets with ungapped and gapped motifs generated relative to motif dispersion rates 0%, 10%, and 20%.

• **Results and Discussion** The results given in Table 2 show that both of the algorithms perform well at low motif dispersion rates, and performance falls off as the motif dispersion rate increases (particularly in the case of gapped motifs). The modified Gibbs algorithm always performs better than the naive Gibbs algorithm in the case of ungapped motifs and gapped protein motifs; however, in the case of gapped DNA motifs, the naive Gibbs algorithm performs better at non-zero motif dispersion rates. This suggests that the modified Gibbs algorithm is sensitive to the size of the sequence-alphabet in the case of gapped motifs. This may be a product of our modifications or it may be inherent in the Rocke and Tompa algorithm (as that algorithm was

Table 2: Simulation Study #1 Results.

	% Motif Dispersion	Average # Runs Target Motif Detected			
		DNA		Protein	
		Naive	Mod.	Naive	Mod.
Ungap	0	9.53	9.23	8.80	9.19
	10	9.39	9.30	8.17	8.86
	20	7.03	8.50	6.99	7.98
Gap	0	8.05	8.29	9.45	9.63
	10	6.90	5.86	7.58	8.90
	20	3.42	3.18	4.41	7.15

never tested on its ability to detect known protein (let alone known DNA) motifs [14]). In any case, these results suggest that methods that integrate avoidance-sequence information into target-motif search may be useful for detecting weakly-conserved target motifs in the presence of strongly-conserved non-target motifs.

4.3 Simulation Study #2

• **Motivation** Given that algorithms like that described in this paper are preferable to naive algorithms, how many runs of our modified Gibbs algorithm are required to detect significant numbers of target motifs in datasets that contain multiple target motifs? This is worth knowing, as it will often be the case that we won't know how many motifs are present and will need guidelines on how many times we must run our algorithm in order to find a significant number of these motifs.
• **Methods** Each dataset consisted of ten motifs (five target) of length 20 which were embedded in 20 target sequences and 2 avoidance sequences of base length 300 (which had post-embedding lengths of 500 and 400 respectively). All motifs have a common specified dispersion rate. Each trial consisted of 25 runs on a particular dataset. The modified Gibbs algorithm was run for 100 trials on protein and DNA sequence datasets with ungapped and gapped motifs generated relative to motif dispersion rates 0%, 10%, and 20%.
• **Results and Discussion** The results given in Table 3 show that, on average, the modified Gibbs algorithm recovers more than half of the target motifs after only 10 runs and almost all of the target motifs after 25 runs under all motif dispersion rates in the case of ungapped DNA and ungapped and

Table 3: Simulation Study #2 Results.

	% Motif Dispersion	Average # Target Motifs Detected					
		DNA			Protein		
		# Runs			# Runs		
		5	10	25	5	10	25
Ungap	0	3.34	4.13	4.71	3.49	4.43	4.93
	10	3.19	3.94	4.56	3.38	4.32	4.90
	20	3.03	3.61	3.98	2.96	3.95	4.81
Gap	0	2.56	3.02	3.47	3.69	4.44	4.96
	10	1.48	1.95	2.61	3.28	4.23	4.81
	20	0.28	0.37	0.48	1.86	2.80	4.21

gapped protein motifs. The previously noted sensitivity of the algorithm when confronted with gapped DNA motifs seems to be dramatically exacerbated by having multiple motifs present.

4.4 Experiments on Real Data

• **Motivation** The previous results in this section suggest that the modified Gibbs algorithm is good at identifying target motifs in simulated datasets under a variety of conditions. However, the dataset simulator is admittedly simplistic and the produced datasets probably do not exhibit crucial characteristics of real sequences and motifs. How well does our modified Gibbs algorithm perform on real datasets? Moreover, what characteristics of these datasets are correlated with (and thus may be useful in explaining) this performance?

• **Methods** We were unable to find natural examples of target motif sequence datasets; [b] hence, we constructed target motif datasets from published protein sequence alignments for the acetyltransferases,[10] cyclins,[12] protein kinases,[3,9] lipocalins,[6] and cytosine methyltransferases.[13] Each alignment contained multiple highly-conserved ungapped regions and a subset of these regions were designated as ungapped motifs. Each of these datasets was transformed into a set of two or more target motif datasets like that in simulation study #1 by selecting one of the motifs, selecting a subset of the given sequences to be avoidance sequences and then randomly permuting the

[b] Though a dataset being used in pharmaceutical research was made available to us, the associated target motifs could not be revealed due to confidentiality agreements.

Table 4: Real Data Results: Motif Detection. The annotation $(|T|, L_T; |A|, L_A)$ gives the number of and average length of the target and avoidance sequences in each dataset.

Dataset	Motif Code	Motif Length	Average # Runs Motif Detected
Acetyl. (6,173;3,184)	A	23	0.08
	B	18	0.00
Cyclin (8,200;2,193)	I	35	8.96
	II	16	6.44
	III	26	8.00
	IV	21	0.00
Kinase (10,274;2,276)	I	10	0.16
	IV	20	8.06
	VI	10	1.20
	VII	10	3.70
Lipocalin (4,185;1,180)	A	16	5.54
	B	16	2.82
Methyl. (4,379;1,477)	I	20	0.42
	IV	24	8.26
	VI	20	3.78
	VIII	20	5.26

amino acids comprising all occurrences of the selected motif in all of the avoidance sequences; this effectively "erased" occurrences of the selected motif in the avoidance sequences and rendered it a target motif. These constructed datasets were run against the modified Gibbs algorithm as in simulation study #1, with each dataset being run for 50 trials.

Each motif in the datasets constructed above was characterized in terms of its length and the following additional quantities:

1. *Degree of Motif Conservation*: The relative entropy measure was used to compute the conservation of the overall symbol-distribution in the motif relative to the background sequence symbol-distribution (*Ovr*) and the average conservation (over all motif-positions) of the symbol-distribution for each position of the motif relative to the background sequence symbol-distribution (*Avg*).

2. *Degree of Target-Sequence Interference*: Unlike the simulated datasets examined in previous sections, real datasets may have multiple partially-conserved copies of a particular motif in each sequence. Call these

partially-conserved copies *ghosts*. Given that our criterion of motif-detection measures overlap of a candidate target-motif with only one of these copies in each target sequence and the stochastic nature of our algorithm makes it possible that our algorithm may accidentally lock onto sufficiently-conserved ghosts in various target sequences, the presence of sufficiently-conserved ghosts in a sufficient number of target sequences may force the detection criterion to reject what is otherwise (purely in terms of sequence-pattern) the correct target motif. We computed two measures of the similarity of ghosts in the avoidance sequences to known motifs, $DTMG_L(m) = (\sum_{i=1}^{N} W - \max_{s \in s(T_i), s \neq m_i} d(m_i, s))/N$ and $DTMG_G(m) = (\sum_{i=1}^{N} d(c, m_i) - \max_{s \in s(T_i), s \neq m_i} d(c, s))/N$, where m is the motif, c is the consensus string associated with m, m_i is the occurrence of m in the ith target sequence, $d(x, y)$ is the Hamming similarity of strings x and y, W is the length of m, and $s(T_i)$ is the set of all substrings of length W in the ith target sequence. Intuitively, the former measures the average similarity of ghosts to known occurrences of motifs within individual sequences, while the latter measures the average difference between both known occurrences of motifs and ghosts within a selected sequence and a motif-model based on the remaining sequences (and hence assesses, in a sense, how likely it is that a number of ghosts may be accidentally chosen as motif-instances by our algorithm).

3. *Degree of Avoidance-Sequence Interference*: Ghosts may also occur in avoidance sequences, and the presence of a single sufficiently-conserved ghost in any avoidance sequence may cause the postprocessing loop to reject what is otherwise the correct target motif. We computed a measure of the similarity of ghosts in the avoidance sequences to known motifs, $DAMG(m) = (\sum_{i=1}^{N} d(c, m_i) - \max_{s \in s(A_i)} d(c, s))/N$, where $s(A_i)$ is the set of all substrings of length W in the ith avoidance sequence.

The values of these quantities for each motif were subsequently correlated with the detectability of each motif using Pearson correlation coefficients,[8] whose values range from -1 (strong negative correlation) to 0 (no correlation) to 1 (strong positive correlation).

• **Results and Discussion** The raw detectability results in Table 4 show that the modified Gibbs algorithm detects two-thirds of the target motifs some of the time and half of the motifs most of the time. This performance is adequate but not as good as that reported for the simulation studies in previous sections. The results in Table 5 (Column "All") suggest that the most important factors in explaining this discrepancy are motif length and the closeness of target sequence ghosts. As the interference caused by target ghosts

Table 5: Real Data Results: Motif Detection / Characteristic Correlations. The probability α that each correlation coefficient is not significantly different from a hypothesis of no correlation was calculated using the t-test described on pages 250 and 251 of May *et al.* [8] These probabilities are represented by starred annotation as follows: **** $\rightarrow \alpha = 0.05$, *** $\rightarrow \alpha = 0.1$, ** $\rightarrow \alpha = 0.2$, and * $\rightarrow \alpha = 0.5$. See main text for explanation of terms.

Motif Characteristic		Correlation With Motif Detectability			
		All		Partial	
Motif Length		0.5061	****	0.4166	*
Motif Conservation	Ovr	-0.4231	**	-0.5047	**
	Avg	0.4581	***	0.1366	
Target Ghosts	$DTMG_L$	0.5301	****	0.6307	****
	$DTMG_G$	0.6642	******	0.4646	**
Avoidance Ghosts		-0.3128	*	0.0873	

may be masking the effects of other factors, we removed the data for the five motifs with the closest target ghosts and recomputed the correlation coefficients (Table 5, Column "Partial"). This caused a reduction in the significance of many previous correlations, though correlations with target sequence ghosts remain strong. More motifs cannot be removed from a dataset this small to explore the effects of other factors without rendering the derived correlation coefficients indistinguishable from the effects of sampling error. Hence, future research should look at redoing the experiments described in this section with more real datasets.

4.5 Integrating Symbol Similarity Information

Amino-acid similarity matrices and, to a lesser extent, nucleotide mutation models have been used profitably in many sequence alignment and pattern construction algorithms.[2] Can such information be used to improve target motif detection? It is not obvious how such information can be integrated into either the Lawrence *et al*[6] or Rocke and Tompa algorithms. The solution adopted here is to partition the amino acids into 7 biochemically-based classes (namely, $\{A, I, L, M, F, W, V, Y\}$ (hydrophobic), $\{S, T, N, Q\}$ (polar but uncharged), $\{K, R, H\}$ (positively charged), $\{E, D\}$ (negatively charged), and $\{C\}$, $\{G\}$, and $\{P\}$) such that protein sequences are mapped into sequences over a 7-symbol alphabet prior to motif search. We ran a simulation study to compare the performance of our modified Gibbs algorithm in unmapped and mapped mode relative to mapped-biased sequence datasets, and we also re-ran the real data experiments relative to mapped mode. Details will be given in the journal version of this paper.[15] For now, we can say that mapped mode boosts performance dramatically for certain real datasets, *e.g.*, Acetyltransferase A was detectable in 6.32 runs under mapped mode, and the decreased performance of mapped mode for gapped motifs in the simulation study supports our algorithm's conjectured sensitivity to sequence-alphabet size.

5 Concluding Remarks

Given the sensitivity of our modified Gibbs algorithm to target sequence ghosts, future research should focus on applying the modifications described in Section 3 to more complex Gibbs-based motif identification algorithms that allow multiple copies of a motif to occur in a sequence.[7,11] One of the authors (Trendall) has also used these modifications to implement a Hidden Markov Model[1] (HMM) algorithm for target motif identification, and preliminary tests relative to ungapped motif DNA sequence datasets have shown that this algorithm has performance comparable to the modified Gibbs algorithm. Future work should test this modified HMM algorithm against other simulated and real datasets and compare the derived results with those reported here.

References

1. R. Durbin *et al*, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids* (Cambridge University Press, 1998).
2. D. Gusfield, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology* (Cambridge University Press, 1997).
3. S.K. Hanks *et al*, *Science* **241**, 42 (1988).
4. M. Ito *et al* in *CPM'94* (Springer-Verlag, Berlin, 1994).
5. J.K. Lanctot *et al* in *SODA'99* (ACM Press, New York, 1999).
6. C.E. Lawrence *et al*, *Science* **262**, 208 (1993).
7. J.S. Liu *et al*, *JASA* **90(432)**, 1156 (1995).
8. R.B. May *et al*, *Applications of Statistics in Behavioral Research.* (Harper and Row, New York, 1990).
9. M.A. McClure *et al*, *Mol. Biol. Evol.* **11(4)**, 571 (1994).
10. A.F. Neuwald and P. Green, *J. Mol. Biol.* **239**, 698 (1994).
11. A.F. Neuwald *et al*, *Prot. Sci.* **4**, 1618 (1995).
12. P. O'Farrell and P. Léopold, *Cold Spring Qual. Biol.* **LVI**, 83 (1991).
13. J. Pósfai *et al*, *Nucleic Acids Research* **17(7)**, 2421 (1989).
14. E. Rocke and M. Tompa in *RECOMB'98* (ACM Press, New York, 1998).
15. H.T. Wareham *et al*, submitted to *J. Comp. Bio.*