Modeling splice sites with Bayes networks

Deyou Cai[1], Arthur Delcher[2], Ben Kao[1] and Simon Kasif[1]

[1]Department of Electrical Engineering and Computer Science, University of Illinois, Chicago, IL 60607, USA and [2]Department of Computer Science, Loyola College in Maryland, Baltimore, MD 21210, USA and Celera Genomics, Rockville, MD 20850, USA

Received on November 6, 1998; revised on April 23, 1999; accepted on June 17, 1999

Abstract

Motivation: *The main goal in this paper is to develop accurate probabilistic models for important functional regions in DNA sequences (e.g. splice junctions that signal the beginning and end of transcription in human DNA). These methods can subsequently be utilized to improve the performance of gene-finding systems. The models built here attempt to model long-distance dependencies between non-adjacent bases.*

Results: *An efficient modeling method is described which models biological data more accurately than a first-order Markov model without increasing the number of parameters. Intuitively, a small number of parameters helps a learning system to avoid overfitting. Several experiments with the model are presented, which show a small improvement in the average accuracy as compared with a simple Markov model. These experiments suggest that single long distance dependencies do not help the recognition problem, thus confirming several previous studies which have used more heuristic modeling techniques.*

Availability: *This software is available for download and as a web resource at http://www.ai.uic.edu/software*

Contact: *kasif@eecs.uic.edu*

Introduction

Recent advances in biotechnology have triggered the generation of massive amounts of biological data. The size and complexity of biological sequence databases suggest that automated systems for sequence modeling and analysis will be essential for extracting scientific knowledge from biological sequence data. Such systems have already begun to demonstrate their importance in the process of biological discovery. As an example, the GLIMMER (Salzberg *et al.*, 1998) system for microbial gene finding has been adopted by several genome sequencing projects, and has already been used to find thousands of genes in the bacterium that causes Lyme disease, as well as other bacteria. Several gene-finding systems have also been built for gene finding in human DNA sequences (Kulp *et al.*, 1996; Burge and Karlin, 1997; Henderson *et al.*, 1997; Xu and Uberbacher, 1997).

A critical component in these systems is an effective model of splice junctions. These regulatory regions represent the interface between introns and exons and provide signals for nuclear machinery in the form of snRNA-proteins to excise the intron segments from pre-mRNA. The resulting ligand consists only of exons, the actual protein coding regions of the RNA transcript. From the biological standpoint, the splicing action is necessarily specific; otherwise protein consistency would be poor. It follows that the splice site must exhibit strong features which facilitate a particularly high specificity between the signal-detecting protein and the pre-mRNA. Indeed this is the case as the majority of the gene-finding systems rely heavily on splice site signals as compared with other potential signals in the DNA template.

This paper describes an application of Bayes networks in the form of trees to the problem of modeling DNA regulatory regions, in particular, those DNA segments that signal mRNA splice junction positions. These positions, commonly known as acceptor ($5'$ splice site) and donor ($3'$ splice site) sites, exhibit certain properties which facilitate their recognition by snRNA-proteins (snRNP) in the pre-mRNA splicing process. Tree networks can capture distant dependencies in DNA segments that may be omitted by other models such as conventional Markov and hidden Markov models. This paper generalizes the work reported in Salzberg (1997) (also see Reese *et al.*, 1997) where first-order Markov models were used to learn the probability distribution characterizing acceptor and donor sites. The main contribution of this paper is providing an efficient method to model biological data more accurately than a first-order model without increasing the number of parameters. The number of parameters used in automated learning systems is often critical for obtaining good generalization capability and consequently strong predictive accuracy. Intuitively, a small number of parameters helps a learning system to avoid overfitting.

The method reported in this paper learns the best model

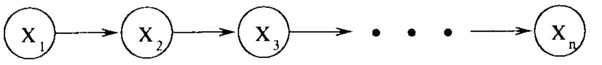

Fig. 1. First-order Markov chain model.

from a family of models (Bayes tree networks) in the sense of maximizing the likelihood of the training data given the model. Since this family of models includes first-order Markov models it is guaranteed to be at least as good as a first-order model. If, however, the best model is a standard first-order Markov model then the algorithm will produce that as its output. Tree models use the same number of parameters as first-order Markov models. As a result, the algorithm can be expected to achieve equal or better predictive accuracy.

Tree networks

An informal introduction to probabilistic tree networks in the context of DNA sequence analysis is provided here. For a more thorough coverage of probabilistic networks the reader is referred to Pearl (1991).

In order to model a DNA sequence of length N, a discrete random variable X_i is associated with each position i in a sequence. Each random variable X_i takes values from the set $\{A, C, G, T\}$. Recall that a first-order Markov model assumes that the joint probability distribution is based on the following assumption of conditional independence:

$$P(X_1, X_2, \ldots, X_n) = p(X_1)p(X_2|X_1)\ldots p(X_n|X_{n-1}) \tag{1}$$

That is, the probability distribution of bases in position i depends only on the previous base in position $i - 1$. Figure 1 shows this dependency.

Probabilistic tree networks generalize the first-order Markov model by allowing position i to depend on any position j. If the probability of variable i depends on variable j, then variable j will be referred to as the parent of i. Probabilistic tree networks allow arbitrary pairwise dependencies as long as each variable has at most one parent in the dependency tree. In other words, each position is allowed to 'depend' on a single position. However, a single position can 'influence' more than one position. For instance, Figure 2 gives a simple probabilistic network that describes a probability distribution in the form of a tree. This probabilistic model implies that the joint probability distribution on seven variables $p(X_0, X_1, X_2, X_3, X_4, X_5, X_6)$ has a simple form. Specifically, it can be factored as a product:

$$\begin{aligned} &p(X_0, X_1, X_2, X_3, X_4, X_5, X_6) = \\ &\quad p(X_0)p(X_1|X_0)p(X_2|X_0)p(X_3|X_1)\ldots \\ &\quad p(X_4|X_1)p(X_5|X_2)p(X_6|X_2) \end{aligned} \tag{2}$$

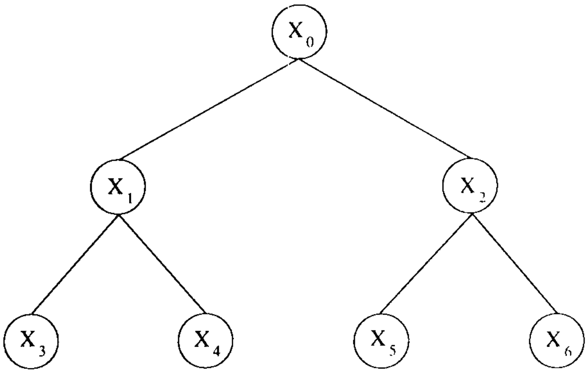

Fig. 2. Tree model.

Learning probabilistic tree models

The method for learning probabilistic tree models used here relies on a classical result by Chow and Liu (1968), who proposed a simple method of computing the best probabilistic tree from the data, in the sense of maximizing the likelihood of data given the model. The procedure to compute this tree is as follows:

- Compute the mutual information $M(i, j)$ between every pair of variables i and j. $M(i, j) = \Sigma_{x,y} p(x, y) \log p(x, y)/p(x)p(y)$, where x and y are the set of values taken by variables i and j, respectively.
- Construct a weighted graph $G = (V, E)$ where node i of the graph corresponds to random variable X_i (position i in the sequence). The edge between nodes i and j is associated with weight $W(i, j) = M(i, j)$, the mutual information between positions i and j in the sequence.
- Construct a maximum spanning tree of the graph G. A maximum spanning tree of a graph is an acyclic subgraph (i.e. a tree) that contains all nodes of the graph, where the sum of the edge weights included in the tree is maximized. Maximum spanning trees can be computed simply and efficiently using several standard algorithms (Cormen *et al.*, 1990).
- Orient the tree by choosing variable X_0 as the root of the tree and orienting all edges away from the root.
- For each pair of variables X_i and X_j such that X_i is the parent of X_j compute the conditional probability of $p(X_j|X_i)$. That is, approximate conditional probabilities by recording the empirical frequencies observed in the data.

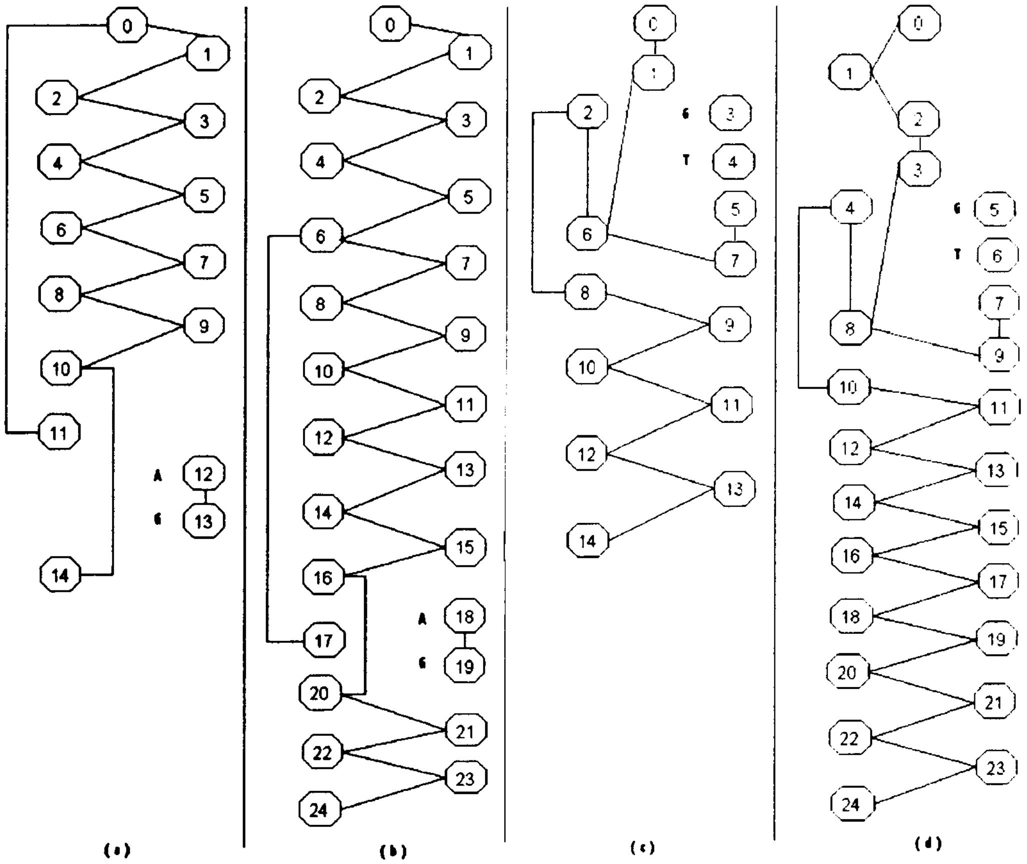

Fig. 3. Tree models for acceptor and donor sites using 15 and 25 base window sizes. Node numbers denote position in the sequence. (a) Acceptor model at window size 15. (b) Acceptor model at window size 25. (c) Donor model at window size 15. (d) Donor model at window size 25.

It is possible to prove (Pearl, 1991) that the above construction maximizes the probability of the data given a tree model. Since the first-order Markov chain is a special case of the tree model, it remains a possible solution of this algorithm assuming it is, in fact, the best model of the data.

The maximal spanning tree is unoriented, thus in order to determine the probability of a particular sequence being generated by the tree model, the tree must first be oriented. The orientation of the tree is arbitrary since relationships between nodes remain constant regardless of how the tree is oriented; orientation merely facilitates the scoring process. For consistency, the variable associated with the first position of the sequence is chosen to be the root of the tree.

Given an oriented tree network as in Figure 2, it is easy to compute the probability that a particular sequence was generated by model M by multiplying the appropriate conditional probabilities. For instance, in order to score a particular sequence $(x_0, x_1, x_2, x_3, x_4, x_5, x_6)$, simply multiply the conditional probabilities as follows:

$$\begin{aligned} &p(x_0, x_1, x_2, x_3, x_4, x_5, x_6|M) = \\ &\quad p(x_0)p(x_1|x_0)p(x_2|x_0)p(x_3|x_1)p(x_4|x_1)p(x_5|x_2) \\ &\quad p(x_6|x_2) \end{aligned} \tag{3}$$

where $p(x_i|x_j)$ refers to the conditional probability (as approximated by the empirical frequency in the data) of nucleotide x_i in position i given nucleotide x_j in position j.

Experiments

The procedures used here generally follow the experimental methodology outlined in Salzberg (1997). A set of donor/acceptor sequences and a large set of non-splice junction DNA were extracted from the Genie database, available at http://www.ucsc.edu.

The set of true splice sites was partitioned into 80% training data and 20% testing data while the set of false splice sites was partitioned into 20% training data and

80% testing data, due to its size. Each set of training data is then used to train two different models: a model for true sites, M_T, and a model for false sites, M_F. An unknown sequence S can then be classified by calculating the probability ratio:

$$\frac{p(S|M_\mathrm{T})}{p(S|M_\mathrm{F})} \tag{4}$$

and subsequently comparing this with an empirically determined threshold value. The algorithm then varies the threshold in fixed increments. For each given value of a threshold as determined using the training set, the algorithm computes the false negative and false positive rate on the testing set.

Given a training data set D comprised of K sequences, each of length n, the goal is to compute the probability distribution P that best fits D. Results obtained by applying probabilistic tree networks to the problem of modeling acceptor and donor sites in DNA sequences are summarized below.

The experimental results are summarized in Tables 1 and 2. These tables compare the predictive accuracy of three different models: the above-described tree model (Tree); a conventional first-order Markov chain model (Chain); and a model using independent probabilities for each position in the sequence (Independent). Each table shows the percentage of false positives and false negatives in the testing set for sequences of lengths 15, 20 and 25 bases around the splice site. Table 1 shows results for acceptor sites, and Table 2 shows corresponding results for donor sites. The trees used by the tree model are shown in Figure 3 for both donor and acceptor sites using sequences lengths of 15 and 25 bases. Figures 4 and 5 plot the tradeoff between false negatives and false positives for the three models.

In Table 3 we provide the results of running a variant of the model developed by Burge and Karlin (1997) on our data. This model is based on probabilistic decision trees which support high-order Markov models. For instance, a full tree of depth three (64 leaves) can express a third-order model. The model proposed in Burge and Karlin (1997) is a carefully designed model that focuses only on some of the most prominent dependencies in the data. It appears that the model is providing a slightly better false positive rate than both the chain model and the tree model when no false negatives are allowed in the training set. It is slightly inferior to the chain model in the false negative category as it missed 2/355 donors on average. For the other settings of the threshold the results are somewhat incomparable. We comment that because different models are attached to the leaves of the maximal dependence decomposition (MDD) model it is somewhat difficult to define thresholds for each model to provide a proper comparison with the single model used in the other methods.

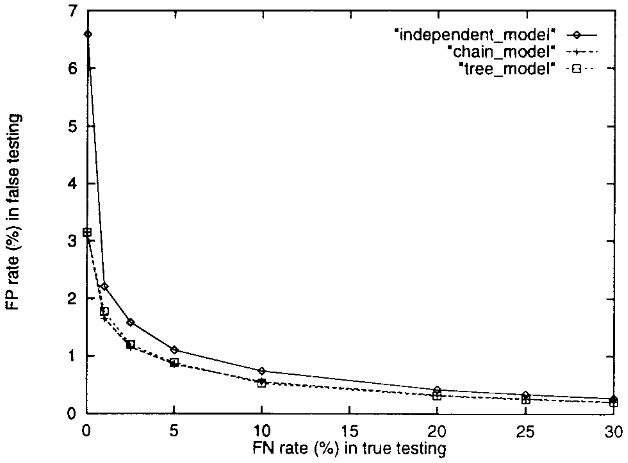

Fig. 4. Comparison of the three acceptor models.

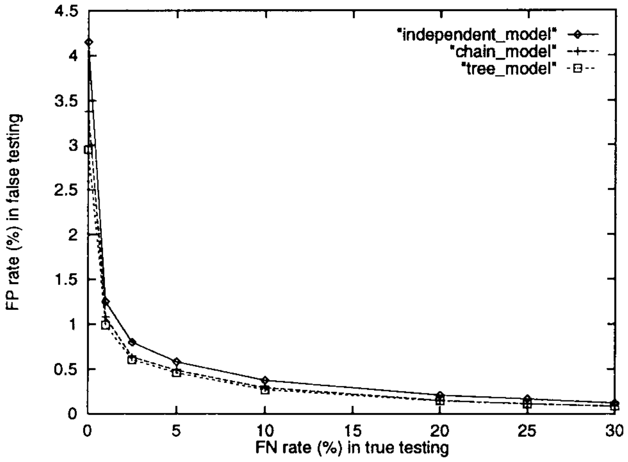

Fig. 5. Comparison of the three donor models.

These results confirm that simple first-order Markov chains are surpisingly effective for modeling splice sites. In general the variations are very small. It is important to observe that a better classification accuracy on a static classification problem is not a guarantee of improving gene recognition if the splice-site model is used in a general system such as Genscan (Burge and Karlin, 1997).

Conclusion

This paper has described a new model of splice sites in DNA sequences. In particular, the new model is capable of capturing distant dependencies between non-neighboring bases in DNA signals. These neighboring and non-neighboring dependencies are captured through a tree structure of conditional probabilities. This approach provides a generalization of previously published work

Table 1. Comparison of **acceptor** tree models at window sizes 15, 20 and 25 (FN = false negatives, FP = false positives)

		Window Size					
		15		20		25	
Model	Training FN (%)	True testing FN (%)	False testing FP (%)	True testing FN (%)	False testing FP (%)	True testing FN (%)	False testing FP (%)
Tree	0	0.85	3.15	0.85	2.92	0.56	2.84
	1	1.69	1.77	1.41	1.52	1.13	1.31
	2.5	3.94	1.19	3.66	1.01	4.23	0.936
	5	6.2	0.882	6.76	0.702	8.45	0.657
	10	15.5	0.524	13.2	0.455	14.4	0.413
	20	27.9	0.313	27	0.253	27.6	0.231
	25	32.4	0.251	34.1	0.2	35.8	0.158
	30	36.3	0.205	39.7	0.158	41.1	0.128
Chain	0	1.13	3.15	1.41	2.91	1.13	2.91
	1	1.97	1.65	1.41	1.39	1.41	1.36
	2.5	3.94	1.15	3.66	0.968	3.66	0.912
	5	7.32	0.855	6.48	0.708	8.17	0.657
	10	13.2	0.55	13.8	0.457	14.6	0.402
	20	27.6	0.326	27	0.268	28.2	0.236
	25	31.5	0.263	35.2	0.197	34.9	0.17
	30	38.3	0.209	38.3	0.157	40.6	0.129
Independent	0	0	6.59	0	6.29	0	5.81
	1	1.13	2.21	1.41	2.05	1.41	2.03
	2.5	5.07	1.58	4.23	1.37	4.23	1.37
	5	9.01	1.1	8.17	0.97	8.17	0.98
	10	13.5	0.735	13.2	0.651	14.9	0.61
	20	25.1	0.42	25.6	0.364	23.4	0.344
	25	31.5	0.337	29.9	0.282	28.7	0.272
	30	34.6	0.277	34.6	0.233	33.2	0.213
Data size	1399	355	3719401	355	3719401	355	3719401

for splice site recognition. Moreover, the learning method described here for modeling splice sites is theoretically guaranteed to match or outperform previous first-order Markov-model methods in terms of likelihood of matching the data given the model.

From a biological perspective the rather surprising discovery is the result that Markov chains are surprisingly effective for modeling splice sites. The method described in the paper is guaranteed to find the best tree network, thus results suggest that distant pairwise dependencies are not sufficient for dramatic improvements in splice site recognition. This is an intriguing result which is also supported by similar studies (using different methods) found in Agarwal and Bafna (1998) and Burge and Karlin (1997).

Tree-based models, however, are likely to have additional applications in biological sequence modeling such as transcription factor binding sites where the sparsity of data precludes the use of a more complex model which has many more parameters.

While the approach described here is promising, it is important to note that it is not guaranteed to achieve a better classification accuracy (e.g. reducing error) on unseen data. The theoretical reason is simple. Recall that the original goal was to find the most likely model, M, given the data D; however, this is not readily maximizable. Thus, one must resort to using Bayes' rule to invert the expression. Now,

$$P(M|D) = P(D|M)\frac{P(M)}{P(D)} \tag{5}$$

Since $P(D)$ is constant, and assuming that all tree models are equally likely, equation (5) implies that $P(M|D)$ can be maximized by simply maximizing $P(D|M)$. Once again, this requires the assumption that all tree models for DNA sequences are uniformly distributed even though this assumption may be unrealistic. Thus, if chain-like models are slightly more likely than non-chain models the improved ability to fit the data might, in fact, be balanced by the $P(M)$ factor that is ignored in this computation.

In other words, while the number of parameters is the same as in first-order Markov models the tree model can

Table 2. Comparison of **donor** tree models at window sizes 15, 20 and 25 (FN = false negatives, FP = false positives)

		Window Size					
		15		20		25	
Model	Training FN (%)	True testing FN (%)	False testing FP (%)	True testing FN (%)	False testing FP (%)	True testing FN (%)	False testing FP (%)
Tree	0	1.13	3.09	1.13	2.95	1.13	2.62
	1	1.41	1.07	1.41	0.991	1.41	0.959
	2.5	1.97	0.7	2.82	0.608	2.54	0.672
	5	4.79	0.431	4.51	0.46	5.63	0.442
	10	11	0.26	12.7	0.27	13.8	0.277
	20	24.5	0.144	23.9	0.147	24.2	0.147
	25	30.1	0.118	32.4	0.111	32.1	0.113
	30	36.1	0.095	39.4	0.088	38	0.092
Chain	0	0	3.65	0.28	3.38	0.28	3.05
	1	1.41	0.886	1.13	1.08	1.69	1.11
	2.5	2.25	0.617	2.25	0.637	1.97	0.73
	5	3.1	0.463	3.94	0.489	4.79	0.505
	10	9.58	0.266	12.1	0.293	13.2	0.297
	20	23.7	0.146	23.4	0.152	22.8	0.159
	25	30.7	0.11	30.1	0.113	32.1	0.116
	30	34.1	0.089	37.2	0.086	37.2	0.095
Independent	0	0.28	3.9	0	4.15	0.28	3.89
	1	0.28	1.17	0.56	1.25	1.13	1.32
	2.5	1.69	0.809	2.54	0.8	3.1	0.835
	5	5.35	0.577	5.63	0.583	6.48	0.598
	10	13	0.375	11.8	0.377	11.3	0.389
	20	22.3	0.205	22	0.21	21.1	0.223
	25	27	0.16	26.8	0.168	27.3	0.17
	30	32.4	0.12	33.8	0.12	32.1	0.129
Data size	1399	355	3719401	355	3719401	355	3719401

Table 3. False negative (FN) and false positive (FP) rates using the maximal dependence decomposition (MDD) method of Burge and Karlin (1997) for predicting donor sites

	Model		
	Training FN (%)	True testing FN (%)	False testing FP (%)
	0	0.56	2.31
	1	1.97	1.87
	2.5	2.82	1.16
	5	3.66	0.69
	10	9.01	0.41
	20	20.0	0.20
	25	26.2	0.16
	30	31.3	0.13
Data size	1399	355	3719401

still overfit the training set since it searches for models over a wider family.

Nevertheless, this is a promising approach for DNA sequence modeling, which might have additional applications in other domains such as transcription-factor binding-site modeling, promoter modeling and protein modeling.

Finally, it is noted that using tree models it is possible to generalize the notion of the consensus sequence. Recall that a consensus matrix as described in many previous studies (Bucher, 1990; Hertz *et al.*, 1990) is a zeroth-order Markov chain. The consensus sequence for a zeroth-order Markov model is given by independently computing the most likely nucleotide in each position. A similar computation of consensus sequences using tree models can also be done efficiently by a simple procedure.

The software used in these experiments is available by sending an e-mail to kasif@eecs.uic.edu.

Acknowledgments

The authors thank Steven Salzberg, Gary Stormo and the reviewers for many useful suggestions. The comparison with the Burge and Karlin model was proposed by Gary Stormo. This research is supported in part by NSF grant IRI-9529227.

References

Agarwal,P. and Bafna,V. (1998) Detecting non-adjoining correlations within signals in DNA. In Istrail,S, Pevzner,P. and Waterman,M. (eds), *RECOMB 98: Proceedings of the Second Annual International Conference on Computational Molecular Biology.* ACM Press, New York, pp. 2–8.

Bucher,P. (1990) Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences. *J. Mol. Biol.*, **212**, 563–578.

Burge,C. and Karlin,S. (1997) Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.*, **268**, 78–94.

Chow,C.K. and Liu,C.N. (1968) Approximating discrete probability distributions with dependence trees. *IEEE Trans. Information Theory*, **14**, 462–467.

Cormen,T.H., Leiserson,C.E. and Rivest,R.L. (1990) *Introduction to Algorithms*. MIT Press, Cambridge, MA.

Henderson,J., Salzberg,S. and Fasman,K. (1997) Finding genes in human DNA with a hidden Markov model. *J. Computat. Biol.*, **4**, 127–141.

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

Kulp,D., Haussler,D., Reese,M.G. and Eeckman,F.H. (1996) A generalized hidden Markov model for the recognition of human genes in DNA. In *ISMB-96: Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology.* AAAI Press, Menlo Park, CA, pp. 134–141.

Pearl,J. (1991) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, revised 2nd edn. Morgan Kaufmann, San Mateo, CA.

Reese,M., Eeckman,F., Kulp,D. and Haussler,D. (1997) Improved splice site detection in Genie. *J. Computat. Biol.*, **4**, 311–323.

Salzberg,S. (1997) A method for identifying splice sites and translational start sites in eukaryotic mRNA. *Comput. Appl. Biol. Sci.*, **13**, 365–376.

Salzberg,S.L., Delcher,A.L., Kasif,S. and White,O. (1998) Microbial gene identification using interpolated Markov models. *Nucleic Acids Res.*, **26**, 544–548.

Xu,Y. and Uberbacher,E. (1997) Automated gene identification in large-scale genomic sequences. *J. Comput. Biol.*, **4**, 325–338.