Hidden Markov Models for biological sequences

A.R.Rao
Indian Agricultural Statistics Research Institute, New Delhi – 110 012

1. Introduction

Very efficient programs for searching a text for a combination of words are avail-able on many computers. The same methods can be used for searching for patterns in biological sequences, but often they fail. This is because biological 'spelling' is much sloppier than English spelling: proteins with the same function from two different organisms are almost certainly spelled differently, that is, the two amino acid sequences differ. It is not rare that two such homologous sequences have less than 30% identical amino acids. Similarly in DNA many interesting signals vary greatly even within the same genome.

A Hidden Markov Model (HMM) is a statistical model, which is very well suited for many tasks in molecular biology, although they have been mostly developed for speech recognition since the early 1970s. The most popular use of the HMM in molecular biology is as a 'probabilistic pro-file' of a protein family, which is called a profile HMM. From a family of proteins (or DNA) a profile HMM can be made for searching a database for other members of the family. Profile HMM treats gaps in a systematic way. HMMs are particularly well suited for problems with a simple 'grammatical structure,' such as gene finding. In gene finding several signals must be recognized and combined into a prediction of exons and introns, and the prediction must conform to various rules to make it a reasonable gene prediction. An HMM can combine recognition of the signals, and it can be made such that the predictions always follow the rules of a gene.

2. From regular expressions to HMMs

Regular expressions are used to characterize protein families, which is the basis for the PROSITE database [Bairoch *et al.*, 1997]. Using regular expressions is a very elegant and efficient way to search for some protein families, but difficult for other. As already mentioned in the introduction, the difficulties arise because protein spelling is much more free than English spelling. Therefore the regular expressions sometimes need to be very broad and complex. Let us imagine a DNA motif like this:

```
A  C     A     -     -     -     A     T     G
T  C     A     A     C     T     A     T     C
A  C     A     C     -     -     A     G     C
A  G     A     -     -     -     A     T     C
A  C     C     G     -     -     A     T     C
-------------------------------------------------------------------
T  G     C     T     -     -     A     G     G
-------------------------------------------------------------------
A  C     A     C     -     -     A     T     C
-------------------------------------------------------------------
```

A regular expression for this is

[AT] [CG] [AC] [ACGT]* A [TG] [GC],

meaning that the first position is A or T, the second C or G, and so forth. The term '[ACGT]*' means that any of the four letters can occur any number of times. The problem with the above regular expression is that it does not in any way distinguish between the highly implausible sequence

T G C T - - A G G

which has the exceptional character in each position, and the consensus sequence with the

A C A C - - A T C

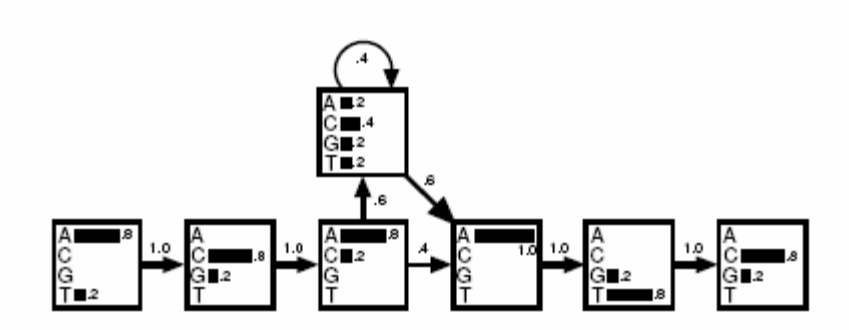

Figure 1: A hidden Markov model derived from the alignment discussed in the text above. The transition probabilities are shown with arrows. In each state the histogram shows the probabilities of the four nucleotides

most plausible character in each position (the dashes are just for aligning these sequences with the previous ones). It is possible to make the regular expression more discriminative by splitting it into several different ones, but it easily becomes messy. The alternative is to score sequences by how well they fit the alignment. To score a sequence, there is a probability of 4/5=0 8 for an A in the first position and 1/5=0 2 for a T, because out of 5 letters 4 are As and one is a T. Similarly in the second position the probability of C is 4/5 and of G 1/5 and so forth. After the third position in the alignment, 3 out of 5 sequences have 'insertions' of varying lengths, so the probability of making an insertion is 3/5 and thus 2/5 for not making one. To keep track of these numbers a diagram can be drawn with probabilities as in Fig. 1.

The above mentioned figure indicates a hidden Markov model. A box in the drawing is called a state, and there is a state for each term in the regular expression. All the probabilities are found by counting in the multiple alignment how many times each event occur, just as described above. The only part that might seem tricky is the 'insertion', which is represented by the state above the other states. The probability of each letter is found by counting all occurrences of the four nucleotides in this region of the alignment. The total counts are one A, two Cs, one G, and one T, yielding probabilities 1/5, 2/5, 1/5 and 1/5 respectively. After sequences 2, 3 and 5 have made one insertion each, there are two more insertions (from sequence 2) and the total number of transitions back to the main line of states is 3. Therefore there are 5 transitions in total from the insert state, and the probability of making a transition to itself is 2/5 and the probability of making one to the next state is 3/5.

It is now easy to score the consensus sequence ACACATC. The probability of the first A is 4/5. This is multiplied by the probability of the transition from the first state to the second, which is 1. Continuing this, the total probability of the consensus is

$$P(\text{ACACATC}) = .8 \times 1 \times .8 \times 1 \times .8 \times .6 \times .4 \times .6 \times 1 \times 1 \times .8 \times 1 \times .8 = 4.7 \times 10^{-2}$$

Making the same calculation for the exceptional sequence yields only $0\ \ 0023 \times 10^{-2}$ which is roughly 2000 times smaller than for the consensus. This way one can achieve the goal of getting a score for each sequence, a measure of how well a sequence fits the motif.

Table 1: Probabilities and log-odds scores for the 5 sequences in the alignment and for the consensus sequence and the 'exceptional' sequence.

	Sequence	P x 100	Log odds
Consensus	A C A C — — A T C	4.7	6.7
Original sequences	A C A — — — A T G	3.3	4.9
	T C A A C T A T C	0.0075	3.0
	A C A C — — A G C	1.2	5.3
	A G A — — — A T C	3.3	4.9
	A C C G — — A T C	0.59	4.6
Exceptional	T G C T — — A G G	0.0023	-0.97

Table 1 shows the calculated probabilities of other four original sequences in the alignment. The probability depends very strongly on the length of the sequence. Therefore the probability itself is not the most convenient number to use as a score, and the log-odds score shown in the last column of the table is usually better. It is the logarithm of the probability of the sequence divided by the probability according to a null model. The null model is one that treats the sequences as random strings of nucleotides, so the probability of a sequence of length L is $0\ 25^{L}$. Then the log-odds score is

$$\text{Log-odds for sequence S} = \log \frac{P(S)}{0.25^{L}} = \log P(S) - L \log 0.25.$$

One can also use other null models instead. Often one would use the over-all nucleotide frequencies in the organism studied instead of just 0.25. For instance, the calculation of the log-odds of the consensus sequence is

$$\text{Log-odds(ACACATC)} = 1.16 + 0 + 1.16 + 0 + 1.16 - 0.51 + 0.47 - 0.51 + 1.39 + 0 + 1.16 + 0 + 1.16 = 6.64$$

If the alignment had no gaps or insertions we would get rid of the insert state, and then all the probabilities associated with the arrows (the transition probabilities) would be 1 and might as well be ignored completely. Then the HMM works exactly as a weight matrix of log-odds scores, which is commonly used.

3. Profile HMMs

A profile HMM is a certain type of HMM with a structure that in a natural way allows position dependent gap penalties. A profile HMM can be obtained from a multiple alignment and can be used for searching a database for other members of the family in

the alignment very much like standard profiles [Gribskov *et al*, 1987]. The structure of the model is shown in Fig. 2. The bottom line of states are called the main states, because they model the columns of the alignment. In these states the probability distribution is just the frequency of the amino acids or

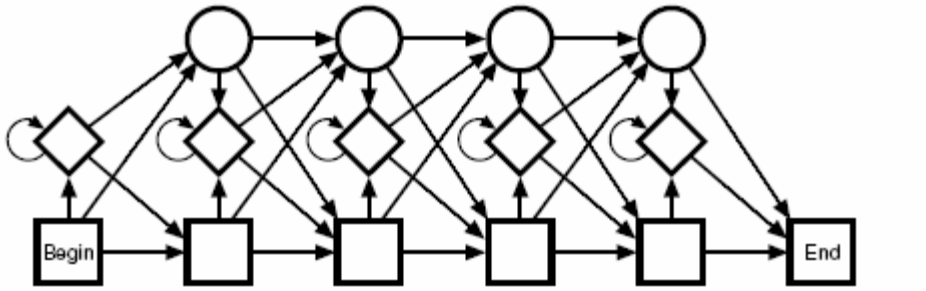

Figure 2. The structure of the profile HMM

nucleotides as in the above model of the DNA motif. The second row of diamond shaped states is called insert states and is used to model highly variable regions in the alignment. They function exactly like the top state in Fig. 1, although one might choose to use a fixed distribution of residues, *e.g.* the overall distribution of amino acids, instead of calculating the distribution as in the example above. The top line of circular states is called delete states. These are a different type of state, called a silent or null state. They do not match any residues, and they are there merely to make it possible to jump over one or more columns in the alignment, *i.e.*, to model the situation when just a few of the sequences have a '–' in the multiple alignment at a position.

As an example consider a multiple alignment as shown in Fig. 3. A region of this alignment has been chosen to be an 'insertion,' because an alignment of this region is highly uncertain. The rest of the alignment (shaded in the figure) is the columns that will correspond to main states in the model. For each non-insert column we make a main state and set the probabilities equal to the amino acid frequencies. To estimate the transition probabilities we count how many sequences use the various transitions, just like the transition probabilities were calculated in the DNA motif example. The model is shown in Fig. 4. There are two transitions from a main state to a delete state shown with dashed lines in the figure, that from *begin* to the first delete state and from main

```
G G W W R G d y . g g k k q L W F P S N Y V
I G W L N G y n e t t g e r G D F P G T Y V
P N W W E G q l . . n n r r G I F P S N Y V
D E W W Q A r r . . d e q i G I V P S K - -
G E W W K A q s . . t g q e G F I P F N F V
G D W W L A r s . . s g q t G Y I P S N Y V
G D W W D A e l . . k g r r G K V P S N Y L
G D W W E A r s l s s g h r G Y V P S N Y V
G D W W Y A r s l i t n s e G Y I P S T Y V
G E W W K A r s l a t r k e G Y I P S N Y V
G D W W L A r s l v t g r e G Y V P S N F V
G E W W K A k s l s s k r e G F I P S N Y V
G E W C E A q t . k n g q . G W V P S N Y I
S D W W R V v n l t t r q e G L I P L N F V
L P W W R A r d . k n g q e G Y I P S N Y I
R D W W E F r s k t v y t p G Y Y E S G Y V
E H W W K V k d . a l g n v G Y I P S N Y V
I H W W R V q d . r n g h e G Y V P S S Y L
K D W W K V e v . . n d r q G F V P A A Y V
V G W M P G l n e r t r q r G D F P G T Y V
P D W W E G e l . . n g q r G V F P A S Y V
E N W W N G e i . . g n r k G I F P A T Y V
E E W L E G e c . . k g k v G I F P K V F V
G G W W K G d y . g t r i q Q Y F P S N Y V
D G W W R G s y . . n g q v G W F P S N Y V
Q G W W R G e i . . y g r v G W F P A N Y V
G R W W K A r r . a n g e t G I I P S N Y V
G G W T Q G e l . k s g q k G W A P T N Y L
G D W W E A r s n . t g e n G Y I P S N Y V
N D W W T G r t . . n g k e G I F P A N Y V
```

Figure 3. An alignment of 30 short amino acid sequences. Shaded area represents most conserved region and is the mainstates in the HMM. The unshaded area represents insert states

state 12 to delete state 13. Both of these correspond to dashes in the alignment. In both cases only one sequence has gaps, so the probability of these delete transitions is 1/30. The

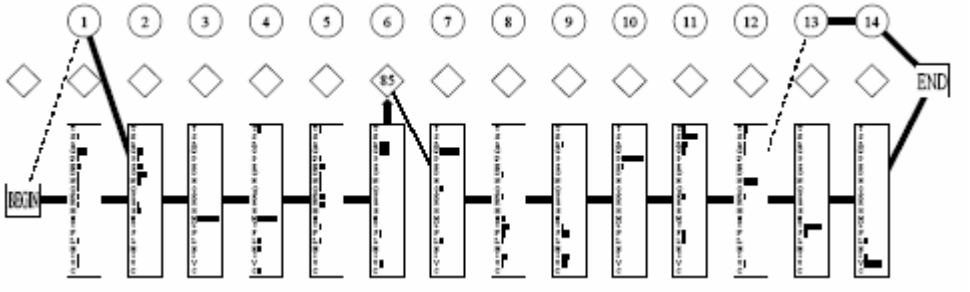

Figure 4. A profile HMM made from the alignment shown in Figure 3.

fourth sequence continues deletion to the end, so the probability of going from delete 13 to 14 is 1 and from delete 14 to the end is also 1.

Searching a database

It is discussed earlier how to calculate the probability of a sequence in the alignment by multiplying all the probabilities (or adding the log-odds scores) in the model along the *path* followed by that particular sequence. However, this path is usually not known for other sequences which are not part of the original alignment, and the next problem is how to score such a sequence. Obviously, if one can find a path through the model where the new sequence fits well in some sense, then one can score the sequence as before. All it needs is to 'align' the sequence to the model. It resembles very much the pairwise alignment problem, where two sequences are aligned so that they are most similar, and indeed the same type of dynamic programming algorithm can be used.

For a particular sequence, an alignment to the model (or a path) is an assignment of states to each residue in the sequence. There are many such alignments for a given sequence. For instance an alignment might be as follows. Let us label the amino acids in a protein as A_1, A_2, A_3, *etc.* Similarly we can label the HMM states as M_1, M_2, M_3, *etc.* for match states, I_1, I_2, I_3 for insert states, and so on. Then an alignment could have A_1 match state M_1, A_2 and A_3 match I_1, A_4 match M_2, A_5 match M_6 (after passing through three delete states), and so on. For each such path we can calculate the probability of the sequence or the log-odds score, and thus we can find the *best* alignment, *i.e.*, the one with the largest probability. Although there are an enormous number of possible alignments it can be done efficiently by the above mentioned dynamic programming algorithm, which is called the Viterbi algorithm. The algorithm also gives the probability of the sequence for that alignment, and thus a score is obtained. The log-odds score found in this manner can be used to search databases for members of the same family. A typical distribution of scores from such a search is shown in Figure 5.

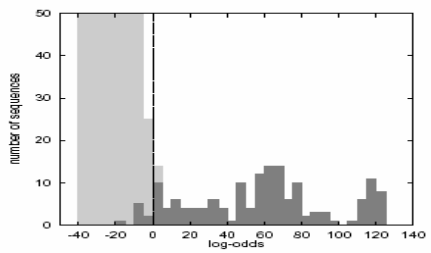

Figure 5. The distribution of log-odds scores from a search of Swissprot with a profile HMM of the SH3 domain. The dark area of the histogram represents the sequences with an annotated SH3 domain, and the light those that are not annotated as having one.

Model estimation

As presented so far, one may view the profile HMMs as a generalization of weight matrices to incorporate insertions and deletions in a natural way. There is however one interesting feature of HMMs, which has not been addressed yet. It is possible to estimate the model, i.e. determine all the probability parameters of it, from unaligned sequences. Furthermore, a multiple alignment of the sequences is produced in the process. Like many other multiple alignment methods this is done in an iterative manner. One starts out with a model with more or less random probabilities, or if a reasonable alignment of some of the sequences is available, a model is constructed from this alignment. Then, when all the sequences are aligned to the model, we can use the alignment to improve the probabilities in the model. These new probabilities may then lead to a slightly different alignment. If they do, we then repeat the process and improve the probabilities again. The process is repeated until the alignment does not change. The alignment of the sequences to the final model yields a multiple alignment. Although this estimation process sounds easy, there are many problems to consider to actually making it work well. One problem is choosing the appropriate model length, which determines the number of inserts in the final alignment. Another severe problem is that the iterative procedure can converge to suboptimal solutions. It is not guaranteed that it finds the optimal multiple alignment, i.e. the most probable one.

4. HMMs for gene finding

One ability of HMMs, which is not really utilized in profile HMMs, is the ability to model grammar. Many problems in biological sequence analysis have a grammatical structure, and eukaryotic gene structure is one such example. If one can consider exons and introns as the ‘words’ in a language, the sentences are of the form exon-intron-exon-intron...intron-exon. The ‘sentences’ can never end with an intron, at least if the genes are complete, and an exon can never follow an exon without an intron in between. Obviously this grammar is greatly simplified, because there are several other constraints on gene structure, such as the constraint that the exons have to fit together to give a valid coding region after splicing. In Figure 6 the structure of a gene is shown with some of the known signals marked.

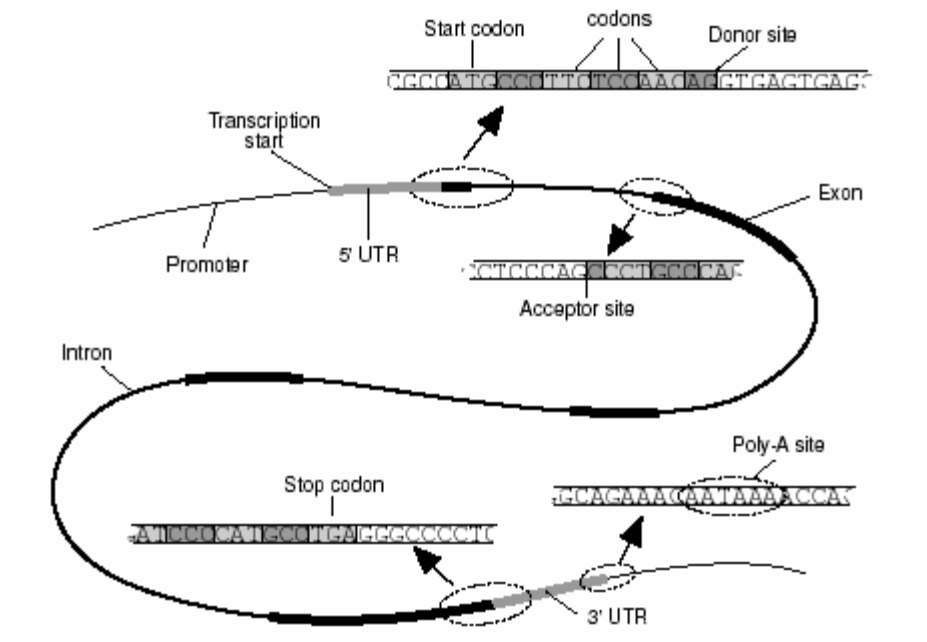

Figure 6: The structure of a gene with some of the important signals shown.

Formal language theory applied to biological problems is not a new invention. In particular David Searls (1992) has promoted this idea and used it for gene finding [Dong and Searls, 1994], but many other gene finders use it implicitly. Formally the HMM can only represent the simplest of grammars, which is called a regular grammar, but that turns out to be good enough for the gene finding problem, and many other problems. Krogh (1998) outlined an approach to find genes with the weight on the principles rather than on the details.

Signal sensors

One may apply an HMM similar to the ones already described directly to many of the signals in a gene structure. In Figure 7 an alignment is shown of some sequences around acceptor sites from human DNA. It has 19 columns and an HMM with 19 states (no insert or delete states) can be made directly from it. Since the alignment is gap-less, the HMM is equivalent to a weight matrix.

```
CTCCCTGTGTCCCACAGGGCT
TATTGTTTTCTTACAGGGC
GTTTCCCTTTGTTTTCTAGCAC
TGCCCTCTCCTTTTTCCAAGGGGT
TCCCTATATGTTTGACAGGGGT
TTCCTGTTTCCCGATGCCAGGGGC
TTGGGGTTTTCCTTTGCCAGAAC
CACTTTTGCCTCCCCACAGCGT
CCCCATGTGTGACCCTGCCAGGGTA
TATTTTATTTTAACCATAGGGGC
ATGTTGCCATCCCCCCCAGGAGG
TTTTTCCCTTTTTCCTACAGAAT
TCCGTGTGTGTCCTCCCCCAGCCCC
TTCCCATGTCCCTGACCAGGGTGG
ACGACATTTTTCCCACAGGAGG
GTGCCTCTCCCCTCCCAGATT
```

Figure 7: Examples of human acceptor sites (the splice site 5' to the exon). Except in rare cases, the intron ends with AG, which has been highlighted. Included in these sequences are 16 bases upstream of the splice site and 3 bases downstream into the exon.

There is one problem: in DNA there are fairly strong dinuclotide preferences. A model like the one described treats the nucleotides as independent, so dinucleotide preferences

can not be captured. This is easily fixed by having 16 probability parameters in each state instead of 4. In column two we first count all occurrences of the four nucleotides given that there is an A in the first column and normalize these four counts, so they become probabilities. This is the conditional probability that a certain nucleotide appears in position two, given that the previous one was A. The same is done for all the instances of C in column 1 and similarly for G and T. This gives a total of 16 probabilities to be used in state two of the HMM. Similarly it can be extended to all the other states. To calculate the probability of a sequence, say ACTGTC, we just multiply the conditional probabilities

$$P(ACTGTC\ldots) = p_1(A) \times p_2(C/A) \times p_3(T/C) \times p_4(G/T) \times p_5(T/G) \times p_6(C/T) \times \ldots$$

Here p1 is the probability of the four nucleotides in state 1, p2(x/y) is the conditional probability in state 2 of nucleotide x given that the previous nucleotide was y, and so forth. A state with conditional probabilities is called a first order state, because it captures the first order correlations between neighboring nucleotides. It is easy to expand to higher order. A second order state has probabilities conditioned on the two previous nucleotides in the sequence, i.e., probabilities of the form p(x/y,z) . Small HMMs like this are constructed in exactly the same way for other signals: donor splice sites, the regions around the start codons, and the regions around the stop codons.

Coding regions

The codon structure is the most important feature of coding regions. Bases in triplets can be modeled with three states as shown in Fig. 8. The figure also shows how this model of coding regions can be used in a simple model of an unspliced gene that starts with a start codon (ATG), then consists of some number of codons, and ends with a stop codon.

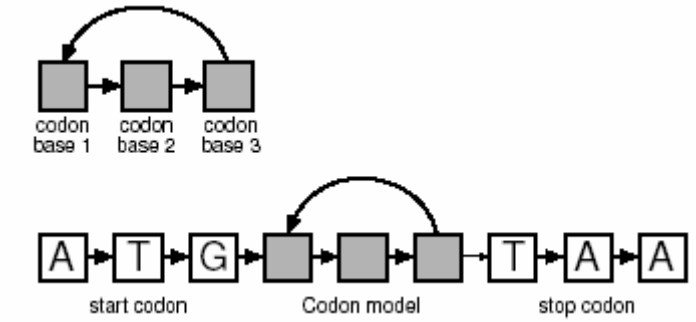

Figure 8: Top: A model of coding regions

Since a codon is three bases long, the last state of the codon model must be at least of order two to correctly capture the codon statistics. The 64 probabilities in such a state are estimated by counting the number of each codon in a set of known coding regions. These numbers are then normalized properly. For example the probabilities derived from the counts of CAA, CAC, CAG, and CAT are

P(A/CA) = c(CAA)/[c(CAA)+c(CAC)+c(CAG)+c(CAT)
P(C/CA) = c(CAC)/[c(CAA)+c(CAC)+c(CAG)+c(CAT)
P(G/CA) = c(CAG)/[c(CAA)+c(CAC)+c(CAG)+c(CAT)
P(T/CA) = c(CAT)/[c(CAA)+c(CAC)+c(CAG)+c(CAT)

where c(*xyz*) is the count of codon *xyz*. One of the characteristics of coding regions is the lack of stop codons. That is automatically taken care of, because p(A/TA), p(G/TA) and p(A/TG), corresponding to the three stop codons TAA, TAG and TGA, will automatically become zero. For modeling codon statistics it is natural to use an ordinary (zeroth order) state as the first state of the codon model and a first order state for the second. However, there are actually also dependencies between neighboring codons, and therefore one may want even higher order states.

Softwares and websites on HMMs

There are two program packages available free of charge to the academic community. One, developed by Sean Eddy, is called hmmer (pronounced 'hammer'), and can be obtained from his web-site (http://genome.wustl.edu/eddy/hmm.html). The other one, called SAM (http://www.cse.ucsc.edu/research/compbio/sam.html), was developed by Anders Krogh and the group at UC Santa Cruz, and it is now being maintained and further developed under the command of Richard Hughey. The gene finder sketched above is called HMMgene. The current version of HMM gene is available at the web site http://www.cbs.dtu.dk/services/HMMgene/. The first HMM based gene finder is probably EcoParse developed for E. coli [Krogh *et al.*, 1994]. VEIL [Henderson *et al.*, 1997] is a recent HMM based gene finder for human genes. The main difference from HMMgene is that it does not use high order states (neither does EcoParse), which makes good modeling of coding regions harder. Two recent methods use socalled generalized HMMs. Genie [Kulp *et al.*,1996; Reese *et al.*, 1997; Kulp *et al.*, 1997] combines neural networks into an HMM-like model, whereas GENSCAN [Burge and Karlin(1997)] is more similar to HMMgene, but uses a different model type for splice site. Also, the generalized HMM can explicitly use exon length distributions, which is not possible in a standard HMM. Web pointers to gene finding can be found at http://www.cbs.dtu.dk/krogh/genefinding .html. Other applications of HMMs related to gene finding are: detection of short protein coding regions and analysis of translation initiation sites in Cyanobacterium [Yada and Hirosawa, 1996; Yada *et al.*, 1997], characterization of prokaryotic and eukaryotic promoters [Pedersen *et al.*, 1996], and recognition of branch points [Tolstrup *et al.*, 1997]. Apart from the areas mentioned here, HMMs have been used for prediction of protein secondary structure [Asai *et al.*, 1993], modeling an oscillatory pattern in nucleosomes [Baldi *et al.*, 1996], modeling site dependence of evolutionary rates [Felsenstein and Churchill, 1996], and for including evolutionary information in protein secondary structure prediction [Goldman *et al.*, 1996].

References

Asai, K., Hayamizu, S., and Handa, K. (1993) *Computer Applications in the Biosciences* 9, 141–146.

Bairoch, A., Bucher, P., and Hofmann, K. (1997) *Nucleic Acids Research* 25, 217–221.

Baldi, P., Brunak, S., Chauvin, Y., and Krogh, A. (1996) *Journal of Molecular Biology* 263, 503–510.

Burge, C. and Karlin, S. (1997) *Journal of Molecular Biology* 268, 78–94.

Dong, S. and Searls, D. B. (1994) *Genomics* 23, 540–551.

Felsenstein, J. and Churchill, G. A. (1996) *Molecular Biological Evolution* 13.

Goldman, N., Thorne, J. L., and Jones, D. T. (1996) *Journal of Molecular Biology* 263, 196–208.

Gribskov, M., McLachlan, A. D., and Eisenberg, D. (1987) *Proc. of the Nat. Acad. of Sciences of the U.S.A.* 84, 4355–4358.

Henderson, J., Salzberg, S., and Fasman, K. H. (1997) Finding genes in DNA with a hidden Markov model Journal of Computational Biology.

Krogh, A. (1998). An introduction to Hidden Markov Models for biological sequences. In *Computational methods in molecular biology*, edited by S.L.Salzberg, D.B. Searls and S. Kasif, pp 45-63. Elsevier.

Krogh, A., Mian, I. S., and Haussler, D. (1994) *Nucleic Acids Research* 22, 4768–4778.

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 States, D., Agarwal, P., Gaasterland, T., Hunter, L., and Smith, R. (Eds.), *Proc. Conf. on Intelligent Systems in Molecular Biology* pp. 134–142 Menlo Park, CA. AAAI Press.

Kulp, D., Haussler, D., Reese, M. G., and Eeckman, F. H. (1997) Integrating database homology in a probabilistic gene structure model In Altman, R. B., Dunker, A. K., Hunter, L., and Klein, T. E. (Eds.), *Proceedings of the Pacific Symposium on Biocomputing* New York. World Scientific.

Pedersen, A. G., Baldi, P., Brunak, S., and Chauvin, Y. (1996) Characterization of prokaryotic and eukaryotic promoters using hidden Markov models In *Proc. of Fourth Int. Conf. on Intelligent Systems for Molecular Biology* pp. 182–191 Menlo Park, CA. AAAI Press.

Reese, M. G., Eeckman, F. H., Kulp, D., and Haussler, D. (1997) Improved splice site detection in Genie In Waterman, M. (Ed.), *Proceedings of the First Annual International Conference on Computational Molecular Biology (RECOMB)* New York. ACM Press.

Searls, D. B. (1992) *American Scientist* 80, 579–591.

Tolstrup, N., Rouz´e, P., and Brunak, S. (1997) *Nucleic Acids Research* 25, 3159–3164.

Yada, T. and Hirosawa, M. (1996) *DNA Res.* 3, 355–361.

Yada, T., Sazuka, T., and Hirosawa, M. (1997) *DNA Res.* 4, 1–7.