A Better Method for Length Distribution Modeling in HMMs and Its Application to Gene Finding

Broňa Brejová and Tomáš Vinař

School of Computer Science, University of Waterloo, ON N2L 3G1, Canada
{bbrejova, tvinar}@math.uwaterloo.ca

Abstract. Hidden Markov models (HMMs) have proved to be a useful abstraction in modeling biological sequences. In some situations it is necessary to use generalized HMMs in order to model the length distributions of some sequence elements because basic HMMs force geometric-like distributions. In this paper we suggest the use of an arbitrary length distributions with geometric tails to model lengths of elements in biological sequences. We give an algorithm for annotation of a biological sequence in $O(ndm^2\Delta)$ time using such length distributions coupled with a suitable generalization of the HMM; here n is the length of the sequence, m is the number of states in the model, d is a parameter of the length distribution, and Δ is a small constant dependent on model topology (compared to previously proposed algorithms with $O(n^3m^2)$ time [10]). Our techniques can be incorporated into current software tools based on HMMs.

To validate our approach, we demonstrate that many length distributions in gene finding can be accurately modeled with geometric-tail length distribution, keeping parameter d small.

Keywords: computational biology, hidden Markov models, gene finding, length distribution

1 Introduction

Hidden Markov models (HMMs) have been used in bioinformatics for such tasks as gene finding, prediction of protein secondary structure, and sequence alignment. HMMs have many features suitable for modeling biological sequences, such as a sound probabilistic basis, and ease in expressing signals and content preferences. Efficient algorithms exist for making predictions and for training HMMs on unannotated or only partially annotated sequences.

One disadvantage of Hidden Markov models is their deficiency in modeling regions with different types of length distribution. If an HMM can stay in one state for more than one step in a row, the probability that it stays in this state for l steps is $(1-p)p^{l-1}$ for some constant p. This gives rise to a geometric probability distribution on the length of the region generated by that state.

However, many elements of biological sequences have length probability distributions that are inadequately approximated by a geometric distribution. Still, such elements may need to be represented in a model by a state or a small group of states. Several approaches have been used to overcome this problem, but none of them is completely satisfactory.

1.1 Current approaches

Durbin et al. [5] discuss several ways to model non-geometric length distributions by replacing a single state by a group of states. Transitions are added inside this group so that the probability of staying within the group for l steps is close to the probability that the modeled feature has length l. This technique does not require any changes in HMM modeling language or in algorithms for training and decoding. However, the running time and needed memory will increase with the number of states, and the conceptual complexity of the model can become unwieldy.

Another disadvantage of this approach is that the additional transitions introduce many paths through the model corresponding to the same sequence annotation. As pointed out in [6], this affects the accuracy of the standard Viterbi algorithm for decoding, which considers only the most probable path through the model. While a natural response would be to look for the most probable annotation instead of the most probable path, this problem was recently proved NP-hard [9].

Rabiner [10] presents generalized HMMs to address the problem of length distributions for speech recognition. A state in a generalized HMM does not generate one character at a time but a region of arbitrary length. The number of characters produced in each state is determined according to a given arbitrary length distribution. Applications of this approach in bioinformatics include the gene prediction tools Genie [7] and GENSCAN [2, 1].

Generalized Hidden Markov models allow for non-geometric length distributions, but give slower decoding algorithms. For a pure HMM, the running time of decoding (prediction) grows linearly with the length of the sequence. For each state and each position, the probability of generating the symbol at that position is evaluated once. In a generalized HMM, a single state can produce an arbitrarily long region. There are $\Theta(n^2)$ regions in a sequence of length n. If the time needed to evaluate a probability of generating one region in a given state grows linearly with the region's length, this gives an $O(n^3m^2)$ running time. In some cases it is possible to organize the computation so that the probability for one region is obtained in constant time, but the running time is still quadratic, which is infeasible for computation on genome-size sequences.

Two approaches have been taken to reduce the running time. First, the running time can be restricted by imposing an upper bound of d on the number of characters produced by each state [10], which gives an algorithm with running time $O(ndm^2)$ or $O(nd^2m^2)$ depending on the type of model used. In speech recognition applications, it is usually possible to keep the bound d relatively small, again yielding a practical running time. However, such assumption cannot be made in the case of elements of biological sequences.

A different approach is taken by the gene finder GENSCAN [2]. Here, practical running time is achieved by using biological knowledge to limit the number of regions that need to be considered for each generalized state. A disadvantage of this approach is that only some elements of the sequence (exons) can be characterized in this way. For the other elements (introns, intergenic regions), geometric distribution is used to achieve a practical running time.

1.2 Our approach

In this paper, we introduce a technique for modeling length distribution in HMMs that addresses some of the problems of the existing techniques mentioned above. We propose a geometric-tail (GT) distribution which can model length distributions of many elements of biological sequences more accurately than the geometric distribution, while still being handled efficiently in decoding algorithms. In the GT distribution long regions have geometric distribution and the probability of shorter lengths can be arbitrary. We describe a general class of models, which we call boxed HMMs, which can be used in conjunction with the GT length distribution. Then, we give an $O(ndm^2\Delta)$ time decoding algorithm, where n is the length of the sequence, m is the number of states of the HMM, d is a parameter of the GT distributions used, and Δ is a small constant dependent on model topology[1].

The rest of the article is organized as follows. In Section 2 we define the GT distribution and provide an algorithm to estimate its parameters from a given sample. In Section 3 we introduce boxed HMM models. In Section 4 we give $O(ndm^2\Delta)$ modification of Viterbi algorithm to decode a boxed HMM with the GT distribution. Finally, in Section 5 we give a brief overview of experimental results that validate our approach and discuss the application of our approach to gene finding.

2 The GT length distribution and its estimation

The *geometric-tail length distribution* (GT distribution) joins two distributions: the first part is an arbitrary length distribution, and the second part is a geometric tail (see example in Figure 1a).

Definition 1 *A* GT *distribution δ of order t is defined by a t-tuple $(d_1, d_2, \ldots, d_t)$ and a parameter q, where $0 \le q < 1$ and $\left(\sum_{i=1}^{t-1} d_i\right) + d_t \cdot \frac{1}{1-q} = 1$. In this distribution,*

$$\delta(x) = Pr[X = x] = \begin{cases} d_x, & \text{if } x \in \{1, \ldots, t\}, \\ d_t \cdot q^{x-t}, & \text{if } x > t. \end{cases}$$

[1] In our gene finding experiments $d \approx 187$, and $\Delta = 3$.

This variety of distribution is suitable for modeling the lengths of many biological sequence elements that have most of the probability mass concentrated in relatively short lengths, but where much longer sequences do sporadically occur. The geometric tail assigns a non-zero probability to a region of any length, so we do not exclude any region by setting the upper bound too small. Also, the geometric tail has only one parameter q, so we do not need dense data to estimate it. On the other hand, densely populated medium length region of the distribution can be estimated simply by the smoothed empirical distribution. If appropriate, we can also stipulate a lower bound on the length of the region by assigning zero probability to all smaller lengths.

Assume we have a training sample of lengths, in which length i occurs $\gamma(i)$ times. We want to use this sample to estimate the parameters $(d_1, \ldots, d_t)$ and q of the GT distribution δ of order t. We want to optimize the parameters subject to the following criteria: First, the relative frequency of observing a length at least t is the same in the distribution δ and the observed sample γ. Second, the likelihood of independently generating lengths in the sample γ from distribution δ is maximized.

In particular, let $x = [\sum_{i=t}^{\infty} \gamma(i)]/[\sum_{i=1}^{\infty} \gamma(i)]$ be the relative frequency of observing length at least t. Then we want to maximize $\prod_{i=t}^{\infty} \delta(i)^{\gamma(i)}$ subject to $x = \sum_{i=t}^{\infty} \delta(i)$. We can easily show that the maximum is achieved when

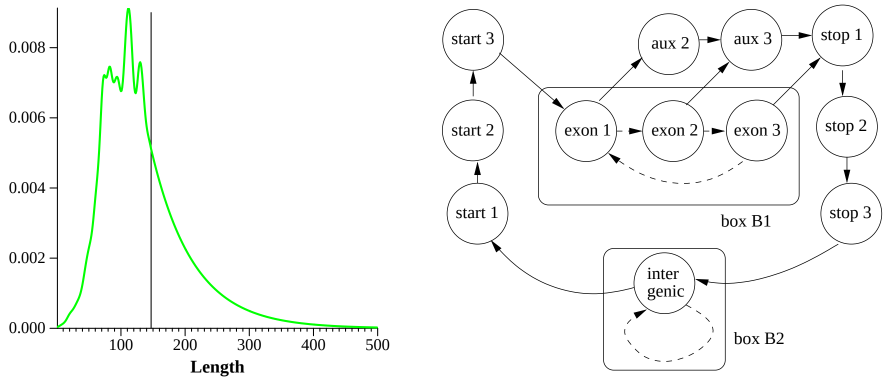
$$q = \left[\sum_{i=0}^{\infty} i \cdot \gamma(i+t)\right] \Big/ \left[\sum_{i=0}^{\infty} (i+1) \cdot \gamma(i+t)\right] \tag{1}$$

$$d_t = x \cdot (1-q) \tag{2}$$

Parameters $d_1, \ldots, d_{t-1}$ can be set to the relative frequency of observing corresponding lengths in training data. However, it is better to smooth these values to avoid overfitting to training data. We used a smoothing procedure introduced in GENSCAN [2]. In this method, every point from the training data set is replaced by a normal distribution centered in this point. We obtain the resulting smooth distribution as a sum of these normal distributions.

As we will show in Section 4, the running time for decoding of an HMM, where lengths of the sequence elements are modeled by GT distribution, grows linearly with t. Therefore, for small values of t, the algorithm runs fast, but the distribution is close to geometric, which may not fit the data very well. For large values of t, the decoding will be slower, but the distribution might fit the data better. On the other hand, increasing the value of t increases the number of parameters to be estimated (parameters $d_1, \ldots, d_{t-1}$), but the amount of data available remains the same. Therefore the length distribution obtained for large values of t may not generalize well to examples not present in the training set.

In our tests, to compute a good value of t for a given training data, we first divided data randomly into k equal size parts. In each step, we set aside one part as a testing data set, and we used the $k-1$ remaining parts to estimate parameters of a GT distribution for all values of t. The parameters were then tested on the testing data set and the likelihood score of the testing data was



(a) (b)

Fig. 1. (a) An example of a GT distribution consisting of an arbitrary length distribution for lengths up to 146 and a geometric tail starting at length 147. **(b) A simple boxed HMM model** for DNA sequences consisting of single-exon genes on the forward strand only. Internal transitions are marked by dashed lines. All transitions have probability 1. Box B_1 models the region of exon between the start and stop codons, box B_2 models intergenic regions. Both boxes can use non-geometric length distributions. Here a boxed HMM is a more accurate model than the corresponding pure HMM.

computed. We repeated this process k times, where in each step, one of the parts served as testing data. Finally, we chose the value of t with the best overall performance.

In practice, such process can yield large values of t, so of course one can specify an upper bound T based on constraints on running time of the decoding algorithm.

3 HMMs and boxed HMMs

A hidden Markov model is a generative probabilistic model, consisting of *states* and *transitions*. Each state u is a Markov chain of order o_u. Let $e_u(x_1, \ldots, x_{o_u}, x)$ be the emission probability of generating character x, provided that the last o_u symbols generated are $x_1, \ldots, x_{o_u}$. After a symbol is generated, one of the transitions is followed to another state. We denote the probability of a transition from state u to state v as $a_{u,v}$.

Typically each sequence element is modeled by a state or a group of states. For example, in gene finding, groups of states correspond to introns, exons and intergenic regions. Given a biological sequence, we can compute the path through the model which has the highest probability of generating the sequence. This can be done using the well-known Viterbi algorithm (see [5]) in $O(nm^2)$ time, where n is the length of the sequence, and m is the number of states in the HMM. This most probable path predicts the positions of the sequence elements.

HMMs impose a geometric distribution on the lengths of the sequences generated by individual states. This is inappropriate in many applications, and therefore generalized Hidden Markov models are often used. In a generalized HMM each state produces several characters instead of one. The number of characters produced in each state is determined according to a given arbitrary length distribution.

We will consider the following generalization of HMMs, which we call *boxed HMMs*. States of a boxed HMM are organized in boxes. Each state corresponds to a Markov chain of some order. A box is a subset of states; each state belongs to at most one box (some states may be unboxed). Denote $B(u)$ the box containing state u. Each box of a boxed HMM is assigned a length distribution for how long the HMM stays in the box. Denote δ_B the length distribution associated with box B.

Each state located in a box has two sets of outgoing transitions: *internal* transitions which lead only into states grouped in the same box, and *external* transitions which can lead to any state. The probability of an internal transition from state u to state v is denoted by $a'_{u,v}$, and the probability of an external transition is denoted $a_{u,v}$. For every state u, $\sum_v a_{u,v} = 1$ and for every boxed state u, $\sum_v a'_{u,v} = 1$. Unboxed states, of course, have only external transitions.

The boxed HMM as a generative model works as follows. Unboxed states are treated as in ordinary HMMs. Once a box B is entered, the length m is generated according to the length distribution δ_B. For the next $m-1$ steps the internal set of transitions is used in each state (i.e. the HMM will remain in box B for m steps). In the m-th step the external set of transitions is used to take the model to a new box or unboxed state (or to the same box, if that is chosen).[2] A simple example of a boxed HMM is given in Figure 1b.

It might seem that the actual length distribution is a combination of the explicit distribution δ_B associated with a box B and the implicit geometric-like distribution induced by the internal transitions inside the box. However, for a given length m and start state in box B, the sum of probabilities over all strings of length m generated by box B is 1, and the internal transitions do not constitute any implicit distribution of lengths.

Boxes of a boxed HMM correspond to states in other classes of generalized HMMs. However, each state inside a box can have a different set of external transitions and transitions entering a box specify which state of the box is to be used as the first. This feature is very useful for example for modeling frame dependencies between exons in genes.

[2] This implicitly assumes that each state inside the box has at least one external transition. In some cases this might be undesirable. However, there are several ways to relax this restriction to satisfy modeling needs, and the algorithms we present can easily be modified to handle such extensions.

4 Algorithms for boxed HMMs

In this section we introduce an efficient algorithm which can be used to decode boxed HMMs with GT length distributions. The algorithm is a modification of the well-known Viterbi algorithm. Our modifications can be applied in similar way to other generalized HMM models (and thus incorporated in other software tools based on HMMs), as well as to other algorithms related to HMMs (e.g. computing forward and backward probabilities).

Let the input sequence be $x = x_1 x_2 \ldots x_n$. The Viterbi algorithm uses dynamic programming to compute, for each position in the sequence i and each state u, the probability of the most probable path generating the sequence $x_1 x_2 \ldots x_i$, and ending in state u. In our algorithm, let $P(u,i)$ be the probability of such a path with additional constraint that if u is a part of a box B, then the segment generated by B ends at position u.

In addition to $P(u,i)$, we compute two auxiliary quantities Q and p to make the algorithm more efficient. Let δ_B be the GT distribution associated with box B, and t_B be the order of this distribution. The value $Q(u,i)$ is defined similarly to $P(u,i)$, except that we consider only paths, in which the segment generated by B has length at least t_B ($Q(u,i)$ is undefined if state u is not part of any box). The value of $Q(u,i)$, representing all lengths greater than t_B, can be computed in a single step, rather than checking all such lengths one-by-one.

If states u and v are in the same box, let $p(u,v,i,j)$ be the probability of the most probable path generating sequence $x_i x_{i+1} \ldots x_j$ starting in state u and ending in state v, in which only internal transitions are used (this value is undefined if u and v are not part of the same box). The values of p allow us to quickly evaluate probabilities of regions generated inside a single box.

Figure 2 shows recurrence formulas for obtaining the values of P, Q, and p. Using these formulas, we can order the dynamic programming computation so that no recursive evaluation is needed, as follows:

	Time
initialize P and Q	
for $i = 1 \ldots n$, **for** all states u	
| **if** u is a boxed state	
| | **for** $l = 0 \ldots t_{B(u)} - 1$, **for** all states v, where $B(v) = B(u)$	
| | | compute $p(v,u,i-l,i)$ (*)	$O(\|B(u)\|)$
| | compute $Q(u,i)$	$O(m \cdot \|B(u)\|)$
| compute $P(u,i)$	$O(m \cdot \|B(u)\| \cdot t_{B(u)})$
| values computed in (*) can be discarded now	

Note, that if a model does not contain any boxes, our algorithm is identical to the Viterbi algorithm.

The algorithm has time complexity $n \sum_u (m \cdot |B(u)| \cdot t_{B(u)} + |B(u)|^2 \cdot t_{B(u)}) = O(ndm^2 \Delta)$, where $d = \sum_B \frac{|B|}{m} t_B$ (weighted average of start of geometric tail in GT distribution), and Δ is the maximum number of states located in a single box. For example, in our gene finding experiments, we have used a model with $d \approx 187$ and $\Delta = 3$. The algorithm requires $O(nm)$ memory, which is the same as Viterbi algorithm for ordinary HMMs. By contrast, the algorithm presented in [10] runs in $O(nD^2m^2)$ time, where D is an upper bound on the length of duration of one state, or $O(n^3m^2)$, if such upper bound cannot be guaranteed.

The algorithm can be made more efficient if the topology of states in each box satisfies further restrictions. For example, if the states in each box are organized as in Figure 1b Box 1 (a k-periodic Markov chain), we can reduce the time complexity to $O(ndm^2)$, because for a fixed i, j, and u, only one of the values $p(*, u, i, j)$ is non-zero.

Notation. $e_u(i) = e_u(x_{i-o_u}, \dots, x_i)$.
All values not defined otherwise are considered to be 0.

Computation of $p(u, v, i, j)$*:*

If $1 \leq i < j$, $u, v \in B$: $p(u, v, i, j) = \max_{w \in B} e_u(i) \cdot a'_{u,w} \cdot p(w, v, i+1, j)$

If $u = v \in B$, $1 \leq i = j$: $p(u, v, i, j) = e_u(i)$

Computation of $Q(u, i)$*:*

If $u \in B$, $i \geq 1$: $Q(u, i) = \max \begin{cases} \max_{v \in B, w} P(w, i - t_B) \cdot a_{w,v} \cdot \delta_B(t_B) \cdot p(v, u, i - t_B + 1, i) \\ \max_{v \in B} Q(v, i-1) \cdot a'_{v,u} \cdot q_B \cdot e_u(i) \end{cases}$

Computation of $P(u, i)$*:*

If $u \in B$, $i \geq 1$: $P(u, i) = \max \begin{cases} \max_{\substack{j=1\dots t_B-1 \\ v \in B, w}} P(w, i - j) \cdot a_{w,v} \cdot \delta_B(j) \cdot p(v, u, i - j + 1, i) \\ Q(u, i) \end{cases}$

If u is not in a box, $i \geq 1$: $P(u, i) = \max_w P(w, i-1) \cdot a_{w,u} \cdot e_u(i)$

$P(\text{start}, 0) = 1$

Fig. 2. Recursive formulas used in the modified Viterbi algorithm to compute values of P, Q, and p.

5 Application to gene finding

Gene finding is the process of locating genes in DNA sequences. A gene is a region of DNA used in a cell as a template for producing a protein. A gene consists of several coding regions called exons separated by non-coding regions called introns. The DNA between genes is called "intergenic". The goal of gene finding is to label parts of the sequence as being exons, introns or intergenic regions. HMMs and generalized HMMs have been used for this task [2, 1, 7, 6].

We have conducted experiments to verify that the GT distributions are suitable for modeling length distributions occurring in the context of gene finding. For this purpose, we have extracted the lengths of introns, intergenic regions and exons from annotated DNA sequences. We distinguish among the first exon in a gene, internal exons and the last exon in a gene, as the first and the last exon of a gene are processed differently in a cell and their length distribution is different.

At the end of the section we also discuss the effects of incorporation of GT distributions into an existing gene finder.

5.1 Fitting GT distribution to gene structure elements

We have performed a simple analysis to determine how well the geometric-tail distribution can fit the different elements of a gene structure and what are suitable values of t (start of the geometric tail). The analysis was done using the Sanger Centre annotation of human chromosome 22 [4]. We have extracted the observed length distribution for each of the gene-structure elements from the annotated sequences.

The results are presented in Figure 3. The length distribution for internal, first, and last exons can be modeled quite accurately by a GT distribution with small values of t (we have used values of t up to 150). Note, that in the case of length distribution for last exons, there is a significant local maximum centered at position approximately 300 which was not captured by the GT distribution. In order to capture this local maximum well, we would need to choose t at least

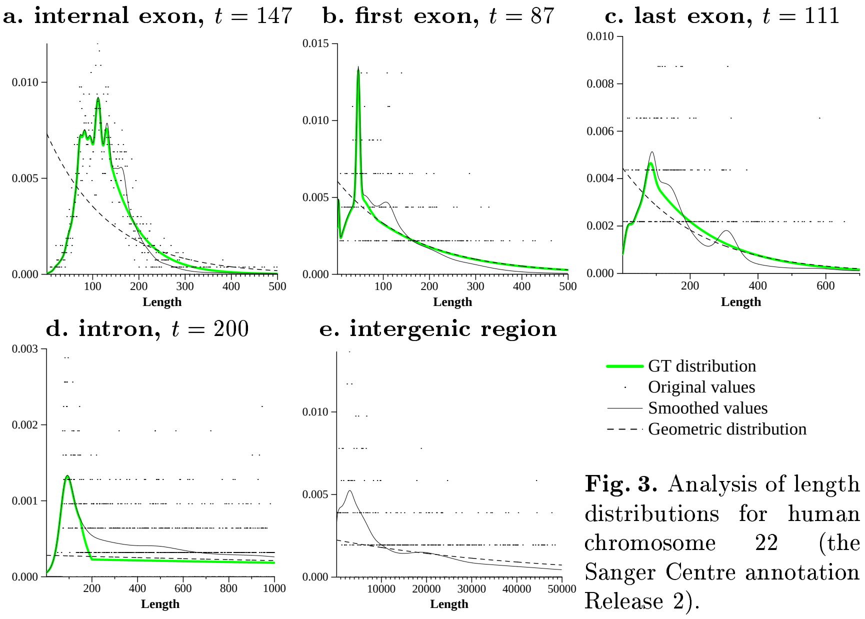


Fig. 3. Analysis of length distributions for human chromosome 22 (the Sanger Centre annotation Release 2).

370. We have observed similar behavior of last exon length distribution in other species as well (arabidopsis, rice), however this phenomena is less pronounced there (results not shown). This may point to a presence of several types of last exons in a genome.

The observed length distribution for introns appears to contain a very long tail that decreases slower than exponentially. Thus the intron tail starts below the observed values to compensate for the rest of the tail (using a very high geometric coefficient $q \approx 0.9997$). This behavior of the intron length distribution can be explained by presence of two types of introns – short and long – which are spliced by different splicing mechanisms (the distribution of short introns is centered around 100bp, while the distribution of long introns is centered around 1000bp [8]).

To capture the mode of the distribution for intergenic region it would be necessary to use impractically large values of t. Thus in this case we have retreated to a geometric distribution.

We have conducted similar experiments on other species (arabidopsis, rice) with similar results (data not shown). In general, the geometric-tail distribution works well with practical values of t for elements where large mass probability is assigned to smaller lengths (exons and introns). However, to model the length of intergenic regions accurately, we would need to use impractically high values of t, so the GT distributions are not suitable for modeling the length of intergenic regions.

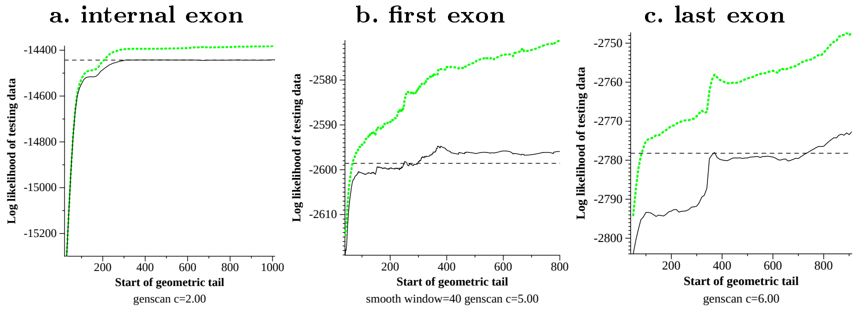
5.2 Generalization to new data

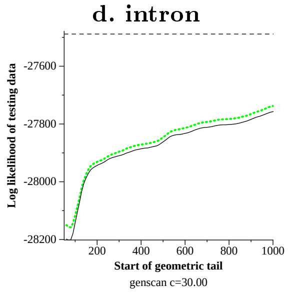
The parameters of a GT distribution are estimated by optimizing the fit to the training data. However, if the resulting distribution fits training data well, it does not mean that this would generalize to testing data.

In the second set of experiments we performed a 5-fold evaluation on the data set used in the previous section. First, the data set was split into 5 parts. In each step, four parts served as a training data and one part as a testing data. We have used our algorithm to fit the GT distribution to the training data for different values of t. Likelihood score was then evaluated for testing data set. We repeated this process 5 times, each time with a different part serving as a testing data set.

For comparison, we have also computed the likelihood score for the distribution obtained by the smoothing function from GENSCAN [2], and for the "ideal fit" distribution that is obtained by using the same data set as both training and testing data. The results are presented in Figure 4.

In all cases, the likelihood of testing data grows rapidly for small values of t. Since a geometric distribution corresponds to value $t = 1$, even small values of t give much better results than a geometric distribution. In the case of first and last exons, the GT distribution outperformed the GENSCAN smoothing for large values of t because the GENSCAN smoothing overfit the data in the sparsely populated tail sections of the length distributions.





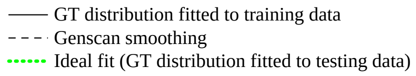


Fig. 4. Generalization capacity of GT distribution for different lengths of tail. Compare with GENSCAN smoothing algorithm and geometric distribution (GT with $t_B = 1$).

In the case of internal and first exons, we achieve performance close to GENSCAN smoothing function very fast (for values of t less than 150). For the last exon the likelihood grows in two phases. In the first phase it achieves a local maximum at $t = 111$; however, the likelihood is still significantly smaller than that of GENSCAN smoothing. The likelihood of GENSCAN smoothing is achieved at a second local maximum at $t = 370$. This behavior is explained by a local maximum centered around length of 300 in the length distribution of last exons as observed in Figure 3c. This local maximum cannot be captured by a GT distribution for $t < 300$.

For intron length distribution, the likelihood of testing data grows steadily even for large values of t (several thousands – data not shown). In this case we need to balance running time versus computation precision.

In all the cases, we see that even small values of t help significantly compared to a geometric distribution ($t = 1$). Moderate values of t allow fast running time, while yielding good results.

5.3 Incorporating GT distribution into a gene finder

We have implemented our algorithm for boxed HMMs and created a simple gene finder. Our preliminary experiments show that using GT distributions can improve the prediction accuracy compared to a geometric distribution. Sensitivity/specificity in terms of exact exons was improved by 5%/3%. The benchmark dataset by Burset and Guigó [3] was used in this experiment. The rate of improvement is consistent with results presented in [8].

However, this rate of improvement cannot be expected after incorporating GT distribution to gene finders based on generalized HMMs, such as GENSCAN [2], which is one of the most successful gene finders. Still, our technique can improve both speed and modeling accuracy of such a gene finder.

GENSCAN uses generalized states to model length distribution of exons. Generalized states generate exon-like sequences with the smoothed length distribution observed in the training data. This, in general, leads to a quadratic running time in the length of the sequence (which is impractical in case of long DNA sequences). However, in case of exons it is possible to use boundaries observed in DNA sequences (so called open reading frames) to limit the search in Viterbi algorithm as exons cannot extend beyond these regions. This is still slow in the case of long open reading frames.

By using GT distributions, it is possible to limit the search also by parameter t of the distribution. As we have shown before, the parameter t can be quite small (up to 200), while still modeling exon length distribution accurately. This means that the decoding can be sped up by our technique without sacrificing prediction accuracy.

GENSCAN's technique cannot be extended reasonably to other elements of DNA sequence (introns and intergenic regions). This is because there are no sequence features such as open reading frames which could be used to limit the search. Therefore in GENSCAN the authors used geometric distributions for introns and intergenic regions.

However, we have shown that a GT distribution can model intron lengths much better than a geometric distribution with reasonably small values of t (such as 200). Modeling intron lengths with GT distributions could thus improve the accuracy of GENSCAN. Still, GT distributions are impractical for intergenic regions, as we would need large values of t (more than 10000).

6 Conclusion

We have studied the problem of modeling length distributions in HMMs for annotation of biological sequences. We have described geometric-tail distributions and boxed HMMs—a class of generalized HMMs that work well with GT distributions. We have demonstrated how to estimate the parameters of a GT distribution from training data and how to modify the classical algorithms for HMMs to handle boxed HMMs with GT distributions efficiently. Our framework is flexible, and several used types of generalized HMMs can be expressed as a boxed HMM. Previously known methods are either too slow to be applied to biological data, or the algorithms have been tailored to one particular application, and hard to generalize for other cases or models.

We have also studied an application of our methods to gene finding. We provide evidence that the GT distribution is indeed a reasonable model for length distribution of introns and exons in genes. We showed that the GT distribution that fits the training data generalizes well to unseen testing data sets. We have discussed the possibility of incorporating GT distributions to current HMM based gene finders.

Acknowledgements. We would like to thank Dan Brown and Ming Li for many useful suggestions. Broňa Brejová is supported by Ontario Graduate Scholarship. Both authors are supported by NSERC research grant.

References

1. C. Burge and S. Karlin. Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology*, 268(1):78–94, 1997.
2. C. B. Burge. *Identification of Genes in Human Genomic DNA*. PhD thesis, Department of Mathematics, Stanford University, March 1997.
3. M. Burset and R. Guigó. Evaluation of gene structure prediction programs. *Genomics*, 34(3):353–357, 1996.
4. I. Dunham et al. The DNA sequence of human chromosome 22. *Nature*, 402(6761):489–495, 1999.
5. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.
6. A. Krogh. Two methods for improving performance of an HMM and their application for gene finding. In *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 179–186, 1997.
7. D. Kulp, D. Haussler, M. G. Reese, and F. H. Eeckman. A generalized hidden Markov model for the recognition of human genes in DNA. In *Proceedings of the 4th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 134–142, 1996.
8. L. P. Lim and C. B. Burge. A computational analysis of sequence features involved in recognition of short introns. *Proceedings of the National Academy of Sciences USA*, 98(20):11193–11198, 2001.
9. R. B. Lyngsøand C. N. S. Pedersen. Complexity of comparing hidden Markov models. Technical Report ALCOMFT-TR-01-144, Algorithms and Complexity – Future Technologies Project (ALCOM-FT), June 2001.
10. L. R. Rabiner. A tutorial on Hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.