ProGen: GPHMM for prokaryotic genomes

Sharad Akshar Punuganti

May 10, 2011

Abstract

ProGen is an implementation of a Generalized Pair Hidden Markov Model (GPHMM), a model which can be used to perform both the tasks of gene-finding and alignment simultaneously. EM-based and viterbi algorithms have been developed and implemented to train the model and find the genes respectively. ProGen models prokaryotic genome (hence the name, "ProGen") structure and, as preliminary results suggest, performs better alignment of DNA sequences than other substitution-score based alignment methods. Partial genome of Escherichia coli 101-1 has been used to test the gene-finding ability of ProGen. Few results are reported.

1 Introduction

Generalized Hidden Markov Models have been successfully used to model genomic structure and locate (or find) genes in the genome. Examples of such systems include GENSCAN [1] and TWINSCAN [2]. Pair Hidden Markov Models can be used for performing the alignment task of finding the best alignment between any two given sequences. A Generalized Pair Hidden Markov Model (GPHMM) naturally combines a Generalized Hidden Markov Model and a Pair Hidden Markov Model and can be used to perform the task of gene-finding and sequence alignment simultaneously.

One prime difference between a GPHMM based gene-finder and others is that a GPHMM-based gene-finder takes in two (orthologous) sequences rather than just one. And naturally because of the extra information of homology that a GPHMM-based gene-finder has, one can guess that it performs the task of gene-finding better than those that do not have this information. This has been shown to be true in the case of TWINSCAN. TWINSCAN is

a natural extension of GENSCAN and exploits homology between two (input) related genomes. TWINSCAN outperformed GENSCAN in terms of percentages of specifity and sensitivity of exact gene prediction acuracy.

SLAM [3] is a GPHMM based gene-finder and models eukaryotic genome structure. While SLAM has been shown to achieve good results compared to existing gene-finders of eukaryotic genomes, it has a slight disadvantage over other gene-finders: the input sequences to SLAM need to pre-processed. Essentially the input sequences need to be "approximately aligned" inorder to speed up the process of gene-finding and alignment. They make use of AVID [6] genome-scale aligner to do this. The task of alignment of genomic sequences is quite difficult in itself and is further complicated due to the complex genomic structure of eukaryotic genomes. This is evident in the exact gene finding accuracy of these systems which are in the range of 20% - 30%. A natural question would be to explore the capability of GPHMMs in the domain of prokaryotic genomes as the genomic structure of prokaryotes is relatively simple when compared to eukaryotes. Unlike eukaryotes, the DNA of prokaryotic genes do not have exons and introns, but just a start codon followed by coding sequence and ending with a stop codon. Hence the intricacies and issues surrounding splice junctions, acceptor sites and donor sites are vanquished in this domain. It becomes interesting to explore GPHMMs in this simple domain. As of writing this report no such work has been reported in literature. *ProGen* is a GPHMM based gene-finder and aligner for prokaryotic genomes and preliminary results show its effectiveness (and hence that of GPHMMs in general) in locating genes in the genomic region of the bacterial strain Escherichia coli 101-1. Short sequence data influenced from the evolutionary orthologous sequences from Eschericia coli and from the closely related bacterium Shigella have been manually created and input to ProGen. The alignment results of the ProGen system are compared with classical Dynamic Programming based aligners such as Needleman and Wunsch *et al* [5]. A few of the cases where ProGen performed better in terms of alignment are reported in this report.

ProGen has been developed from scratch entirely as part of this project without the use of any external aid or libraries. SLAM is the only known GPHMM based system reported in literature and does not provide enough details about the algorithms used in their system. A major part of my work has been dedicated to developing EM-based algorithm, to train the ProGen system, and a viterbi algorithm for gene-finding, as these algorithms are non-trivial in a GPHMM based model.

Outline The next section describes the GPHMM model of the ProGen system and the EM-based and viterbi algorithms that have been implemented as part of the system. Section 3 describes the experiments that have been carried out to evaluate the correct working of the system. Section 4 discusses a few important issues concerning the ProGen system and a few implementation details. The report ends with a conclusion and the possibility of future work in this direction.

2 Model

The basic GPHMM model used for the project is shown in Figure 1. This model is slightly similar to the model used in the GeneMark.hmm system [7] for modeling prokaryotic genome structure. GeneMark.hmm is a Generalized HMM based model developed for gene-finding in prokaryotic genomes. Note that GeneMark.hmm [7] does not perform alignment as it does not incorporate the Pair HMM in its model.

Note that this model models both the direct and reverse strands of the DNA. There are 7 states in the ProGen model. The first state corresponds to the intergenic region, the region in between genes. State 2 corresponds to the direct start codon. State 3 corresponds to the coding region of the gene. As explained previously the genes in the prokaryotic genomes are not spliced as exons and introns, as is the case in eukaryotic genomes, but have a simple structure: a start codon followed by the coding region of the gene and ending with a stop codon. State 4 corresponds to the stop codon. A similar set of three states, 5, 6 and 7 are included to incorporate the genes on the reverse strand appropriately in the reverse direction: state 5 for reverse stop codon, state 6 for the coding region in the reverse strand and state 7 for reverse start codon.

State 1 is a normal state as in that it emits just 1 pair of nucleotides (one each for the two sequences). Then the system makes a transition either onto the same state, 1, or to the states 2 or 5 if it encounters a start codon on the direct strand or a stop codon on the reverse strand. In these states (i.e. in 2 and 5) and also in states 4 and 7, the model emits a fixed length (3 pairs) of nucleotides and transits to states 3 or 6 respectively. States 3 and 6 are generalized. In these states the system first decides on a duration, d, modeled as:

$$p(d) = Nc * (d/Dc) * exp(-d/Dc) \tag{1}$$

which models the γ distribution over durations. This identifies one of the important advantages of a Generalized HMM over a non-generalized HMM. A state that is not generalized is bound to be distributed according to a

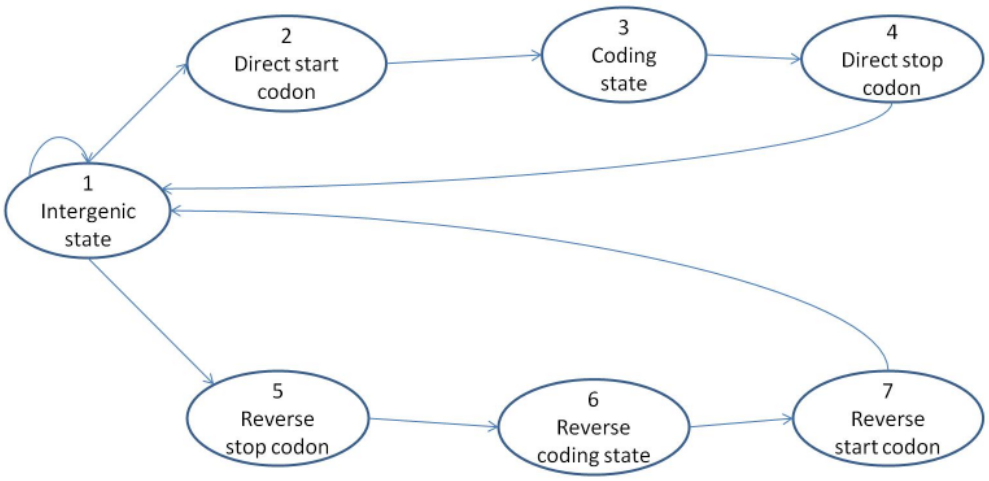

Figure 1: GPHMM model of ProGen. The duration in State 1 is modeled as a geometric distribution, while the durations in states 3 and 6 are modeled as gamma distributed. Codon states 2, 4, 5 and 7 are modeled according to a fixed probability distribution.

geometric distribution while a Generalized HMM allows the states to be modeled according to any choice of distribution *p(d)* for the duration d.

The current system models only 3 start codons ATG, GTG, and TTG with probabilities of 0.905, 0.09 and 0.005 respectively and the stop codons considered for the project are TAA, TAG and TGA with emission probabilities of 0.4, 0.4 and 0.2 respectively. These probabilities have been estimated by counting the number of their occurences in an entire genome [7].

ProGen incorporates a pair HMM too, with the states 3 and 6 modeled internally as described in [4].

2.1 Training the system

The three different notions of:

- Pair HMM

- Generalized HMM
- Different states having different duration distributions

make the task of training the system difficult. An Expectation Maximization based algorithm has been developed to train the system. Basically the states are divided into three categories: (1), (2,4,5,7) and (3,6).

The 1^{st} state has the forward and backward variables estimated in the E-step in the standard way as described in Rabiner et al [8]. The equations for recomputing the forward and backward variables for the rest of the states have been altered to suit the system so as to reduce the time complexity:

$$f(2,t) = f(1,t-3) * a(1,2) * b(2, Obs(t-2:t)) \tag{2}$$

where $f(2,t)$ is the forward variable of state 2 at time t, $a(1,2)$ is the transition probability of transiting from state 1 to 2, and $b(2, Obs(t-2:t))$ is the emission probability of emitting observation sequence $Obs(t-2:t)$ at time instants $(t-2)$, $(t-1)$ and t. Similar equations have been developed for other codon states 4, 5 and 7.

The forward and backward variables for states 3 and 6 are computed according to [8] with duration distributions modeled according to equation 1. The minimum duration in these states is set to 30 nt and the maximum to 7155 nt.

2.2 Viterbi algorithm for ProGen

A viterbi algorithm has been developed and implemented succesfully for ProGen. The important detail is the combination of the viterbi algorithm as described in Durbin et al [4] for a Pair HMM and the viterbi algorithm for a Generalized HMM as described in [8]. The basic idea is that the viterbi forward variable for a state, say 3, is computed by running the viterbi algorithm for the Pair HMM for that state for all possible combination of pair durations (for each of the sequences).

Note that more than 3 start codons can be modeled by the system by appropriately setting up the EM-algorithm to train the parameters of the model. Also for the intergenic region only a gap penalty has been used but not the space penalty as it is modeled to emit only one pair of nucleotides at each time step.

The model has been implemented in Matlab and the source code is available at http://pages.cs.wisc.edu/ akshar/progen/. Please read the "Readme" file for instructions on how to run the code.

3 Experimental evaluation

3.1 Gene-finding

The gene-finding capability of the model was tested by running it against the genomic data of the bacterial strain Escherichia coli 101-1. The genomic data was downloaded from the ASAP database (using the option "All feature types"). Note that for the task of gene-finding only the Generalized HMM of the system was used as the GPHMM took a lot of time to process a data set of 10000nt. To give an idea, the GPHMM would have taken of the order 10^{12} seconds for processing sequences of length 10000nt, as the complexity of the GPHMM viterbi algorithm is $O(D^4 * T1 * T2)$ where D is the maximum duration in a state (which is 7155 for states 3 and 6), $T1$ and $T2$ are the lengths of the two sequences. Table 1 shows the results of the gene-finding task.

Number of nucleotides	10000
Number of genes present	13
Number of genes found by ProGen	9

Table 1: Gene-finding results

A prediction is defined to be correct if the system correctly predicts the start location of the gene. An initial examination of the 4 genes that could not be correctly located in the genome reveals that these genes have either a start codon or a stop codon that was not included as part of the system emission probability parameters (i.e. the probability for these codons were set to 0).

3.2 Alignment

For testing the accuracy of alignments produced by ProGen, the orthologous gene alignments were compared with the alignments produced by the Dynamic Programming approach [7]. The orthologous genes were chosen from the bacterial genomes of Escherichia Coli B171 and the closely related Shigella. As mentioned previously it was computationally intensive to align sequences more than 100 nucleotides long using the GPHMM model. The orthologous sequences were considerably shortened to lengths of the order of a few 10s but also making sure that the major evolutionary mutations between the two sequences were maintained. Several tests were conducted to prove the effectiveness of ProGen over DP algorithm. Some of the results are presented below:

```
CATGAC  TTAAA
CATGACTTTTAAA
1222333334441

CATGACTT  AAA
CATGACTTTTAAA
```

In the above alignments the 1st pair of aligned sequences is the alignment produced by ProGen (the viterbi path of states is also shown) and the 2nd pair is the alignment produced by DP algorithm. Also the 1^{st} sequence in both the alignments is derived from E.Coli and the 2^{nd} sequence from Shigella. Clearly, the stop codon (TAA) of the gene in E.Coli is torn-apart and misaligned by the DP-based program. But ProGen correctly aligns the stop codons in both the sequences. Here is another example of sample alignments of two other sequences:

```
CATGAC     TAAG
CATGACTAGAGTAAA
122233333334441

CATGACTA AG
CATGACTAGAGTAAA
```

The alignment score for the 1st alignment assigned by the DP program is less than the alignment score for the second alignment. Hence the DP-program chooses the second alignment as the better of the two thereby commiting a mistake. ProGen correctly aligns the two sequences (the 1st alignment, (the viterbi path is shown too)) as it emits stop codons in the same state and (hence aligning them together) in both the sequences. In fact any score based algorithm will fail to align the above two sequences. This shows the effectiness of ProGen as it uses the homology information of the sequences.

4 Discussion, Conclusion and Future Work

The system has several limitations. The major drawback, as discussed earlier, is the time complexity of the system which is $O(D^4T_1T_2)$ where D is the maximum duration of the states. This makes it highly infeasible to work with genome size data. SLAM [3] uses AVID [6] to perform an initial alignment of the input sequences and then work with the aligned sequences to find the

genes. Naturally, this idea can be extended to ProGen and hopefully may help reduce the time complexity of the system. The model has an inherent inability concerning alignment when dealing with sequences containing overlapping genes.

For better results as suggested by GeneMark.hmm, more states can be added to account for atypical genes (genes that are orthologously transferred into the genome). These genes contribute upto 15% of all the genes in the genome and thus can significantly improve the accuracy of the gene-prediction. Also the addition of more start codons or stop codons will improve the accuracy of the gene-predictions by the system further as shown in Section 3.1. For e.g. 'GGC' another start codon (Escherichia coli B171, Feature ID:ADN-0005082) is not included in the model but has been found in a few known genes.

However the system, as demonstrated in Section 3.2, as is can be used for accurate aligning of small orthologous gene sequences.

5 Acknowledgments

I wish to thank Prof. Mark Craven for helpful suggestions and support.

References

[1] Burge, C. and Karlin, S. 1997. Prediction of Complete gene structures in human genomic DNA. *J. Mol. Biol.* 268:78–94.

[2] Korf, I., Flicek, P., Duan, D. and Brent, M. 2001. Integrating genomic homology into gene structure prediction.*Bioinformatics* Vol 17, 140-148

[3] Alexanersson, M., Cawley, S., Pachter, L. 2003. SLAM: Cross-species gene finding and alignment with a GPHMM. *Genome Research*

[4] Durbin., R., Eddy., S., Krogh., A., Mitchison, G. 1998. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids.* Cambridge University Press.

[5] Needleman, B., Wunsch, D. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Bio..* 48: 443:53

[6] Bray,.N, Dubchak, I., and Pachter, L. AVID: A Global Alignment Program. *Genome Research* 13:97-102, 2003.

[7] Lukashin, A., and Borodovsky, M. 1998. GeneMark.hmm: new solutions for gene finding. *Nucleic Acids Research* 1998, Vol 26, No. 4

[8] Rabiner, L. 1989. A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *IEEE Proceedings* 1988.