Finding CpG islands in genomic sequence data using Hidden Markov Models

Badrish Chandramouli
Department of Computer Science, Duke University
badrish@cs.duke.edu

December 5, 2002

1. Introduction

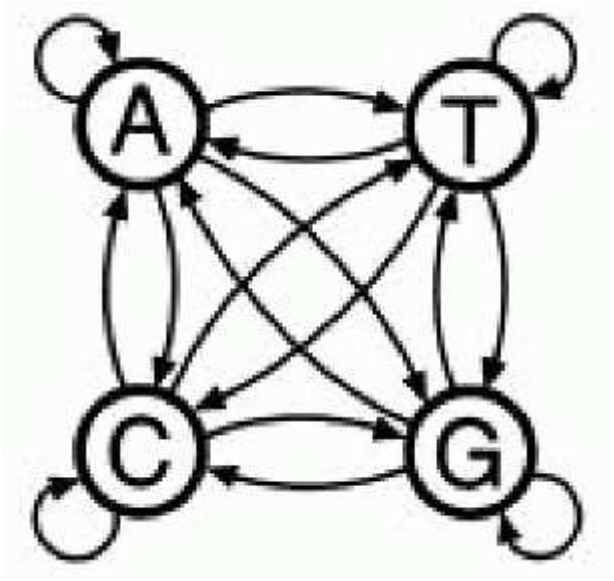
CpG islands are discrete DNA regions in the genome in which an unmethylated CpG dinucleotide frequently occurs. It is known that due to biochemical considerations CpG, the chances of the pair of nocleotides C and G appearing successively in this order along one DNA strand, is relatively rare. However there exist particular sub−sequences, which are several hundreds of nucleotides long, where the couple CpG is more frequent. These sub−sequences, called CpG islands, are known to appear in the biologically more significant parts of the genome. The ability to identify CpG islands in the DNA will therefore help us spot the more significant regions of interest along the genome. In this project, I start off with a description of the algorithmic details of using Hidden Markov Models (HMMs) to detect CpG islands. This includes parameter estimation using various methods. I then give a description of an Expectation−Maximization (EM) algorithm to detect CpG islands. I also give details of the code that I developed in order to implement these techniques and the results I obtained by running them on actual genomic sequence data. Finally, I describe some standard CpG island detection tools that are available online and compare the results I obtained with those obtained by using these programs.

2. Markov chains and hidden Markov models

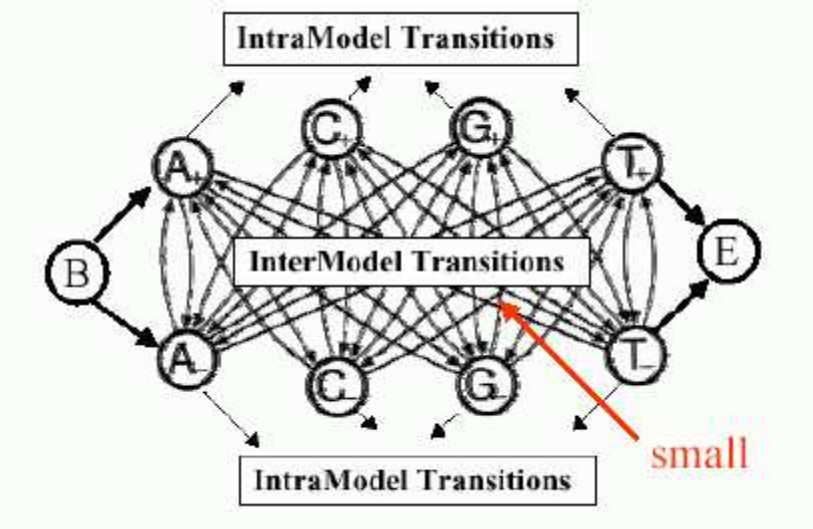
A Markov chain is a simple model in which the probability of a symbol depends on its previous symbol. Its key property is that the probability of each symbol x_i depends only on the value of the preceding symbol x_{i-1} and not on the entir e previous sequence i.e.

$$P(x_i \mid x_{i-1}, \ldots, x_1) = P(x_i \mid x_{i-1}) = a_{st}$$

This is the transition probability i.e. The probability of transitioning from state s to state t. For nucleotide sequences, the Markov chain model has four states A, C, G, T. We also need to define the transition probabilitiesof switching from one state to another. The following figure shows the Markov chain for genomic sequences:



We now wish to use a variant of Markov chains for detection of CpG islands, known as Hidden Markov Models (HMMs). Suppose we split each state into two, one to represent regions of CpG islands (called as "+" states) and one to represent the other regions (called as "−" states). We can obviously build Markov models to represent each of these individually. However, we wish to build a single model that incorporates both chains, with a small probability of switching from one to the other. Hence we introduce symbols A+, C+, G+ and T+ that correspond to the CpG island regions and A−, C−, G− and T− that correspond to non−island regions. The essential difference between a Markov chain and a HMM is that for a HMM there is no one to one correspondence between the states and the symbols. It is no longer possible to tell by looking at a symbol (say C) in isolation whether it was emitted by state C+ or C−. Following figure shows a HMM for detecting CpG islands:



We see that in addition to the states for each symbol, we have added begin and end states in the model. In implementation, we usually use a common state to indicate both

these states without loss of generality. For every state (other than the begin and end), there is a finite emission probability which is either 0 or 1, based on whether on not the corresponding symbol was emitted when the model was in that state.

3. Parameter estimation when the state sequence is known

a) Maximum likelihood estimates

To estimate the parameters when the paths are known, I count the number of times each particular transition occurs in the training set of sequences (the set of sequences for which the CpG island information is known in advance). Then the transition probability for every transition is just the number of transitions from the state s to the state t, divided by the total number of transitions from state s. This technique is known as maximum likelihood estimation (MLE).

b) Maximum a posteriori estimates

In this estimation technique, we estimate the probabilities of transitions after having knowledge of the same. I implement this by adding pseudo−counts to the counts obtained by looking at the training sequences. This technique is known as Maximum a posteriori (MAP) estimation.

4. Parameter estimation when the state sequence is unknown (Baum−Welch training)

When the paths are unknown for the training sequences, a method called Baum−Welch is commonly used. This is an iterative algorithm that uses Expectation−Maximization (EM) technique. Informally, it first estimates the transition counts by considering probable paths for the training sequences using the current values of the transition and emission probabilities. Then based on these estimates, new values for transition and emission are calculated. This process is repeated until some stopping criterion is reached.

I implemented the Baum−Welch algorithm to train the HMM to detect CpG islands. To use as the convergence criterion, I calculate the log−likelihood of the sequence after each iteration and when this log−likelihood changes less than a predefined threshold value, the iterations are stopped. By this method, the overall log−likelihood of the model is increased and the process will converge to a local maximum.

5. Most probable path estimation

Once the parameters of the HMM are estimated, we can use it to detect CpG islands in new sequences. This is done by using algorithms that generate the most probable path through the model. These algorithms generate the path that has the maximum probability for the given sequence. They are both dynamic programming techniques which are explained below.

a) Using Viterbi decoding

In this dynamic programming method, we choose the path with the highest probability in the underlying state sequence. This path is found recursively. Suppose the probability of the most probable path ending in a particular state with a particular observation is known for all states k. Then these probabilities can be calculated for the next observed symbol using standard formulas. We employ this algorithm starting from the begin state (state 0). By keeping pointers backwards, the actual state sequence is found by backtracking.

b) Using Posterior decoding

In order to get the probability of a sequence x for a HMM, we must add the probabilities for all possible paths. This number increases exponentially with the length of the sequence, and so we use more efficient algorithms for the same. There exists an algorithm called the "forward algorithm" that calculates the probability moving forward through the observations. It is similar to the Viterbi algorithm, only replacing maximization steps with sums. There is another algorithm that calclulates the same starting from the last obseration and working its way backward. This is called the backward algorithm. I implemented both these algorithms using dynamic programming. Using these algorithms, it is possible to calculate the posterior distributions of the state at each step. This way we can derive the most probable path for the given sequence in the given HMM. I imlpemented the posterior decoding algorithm using the forward and backward algorithms.

6. The code I developed and the results

I developed all the code to perform automatic detection of CpG islands. I took a set of training sequences (whose CpG island information was known) and ran the MLE and MAP estimation routines to generate the transition matrix for the HMM.

I used the CpG islands provided with the training sequences to estimate the 8x8 transition matrix using MLE. I then estimated the transition matrix with (n−1) sequences, and used the result to predict CpG islands in the sequence I left out. This was repeated for each of the sequences. Prediction was done using Viterbi and Posterior decoders that I developed. This technique is called "leave−one−out cross−validation". I implemented Viterbi decoding in log−probability space, to avoid underflow. Forward and backward algorithms were implemented using scaling. The posterior decoder uses the results of forward and backward algorithms. The program writes results in CSV format which are readable by MS Excel to generate graphs. The graphs and code are attached. The resulting CpG islands were compared with the actual islands. The results matched to a great extent, as can be seen from the graphs plotted.

I added pseudo−counts while calculating the MAP estimates, as described in the Durbin book. I varied the pseudo−counts from 1 to 100, in steps of 5, and predicted the CpG islands for the first annotated sequence. I then compared the results graphically for the first annotated sequence with the corresponding graph generated using MLE estimation (the MAP estimates were computed on the other sequences.)

The graphs and code are attached. In case of MAP estimates, I observed that for small pseudocounts, the detection of CpG islands corresponded roughly with that of MLE estimation. As I increased the pseudocount, some CpG islands which were previously undetected began to get detected with increasing probability. On the flipside, the number of false positives also increased with increasing pseudocounts. This can be seen from the graphs attached.

For the Baum−Welch training, I used 100 unannotated DNA sequences to calculate the transition matrix using the Baum−Welch algorithm. During each iteration, the transition matrix was refined and its log−likelihood was computed. I continued iterating until the change in log−likelihood fell below the threshold that was defined. I used the transition matrix generated using the Baum−Welch algorithm to predict CpG islands in a set of unknown sequences provided using the Viterbi and Posterior decoders. The results were plotted graphically and are attached.

In order to test the validity of the Baum−Welch implementation, I ran it on the annotated sequences, without providing information about CpG islands. I used the result to predict CpG islands in two of the same set of annotated sequences. The results tallied with the actual islands, to a great extent. The graphs of the same are attached.

7. Comparison with the results of online detection tools

There are some online detection tools available, which generate CpG islands for input data. I found the following tools online:

- GrailEXP − http://grail.lsd.ornl.gov/grailexp/
- Emboss − http://www.ebi.ac.uk/emboss/cpgplot/

I ran the unknown sequences on the two online detection tools mentioned. I noticed a difference in the set of CpG islands detected by the tools and those detected by the decoders. The differences are due to the following factors:

- My transition matrix was trained using the set of 100 unannotated sequences. The online tools did not have this prior knowledge.
- There were various parameters in the online tools, such as Obs./Exp., Min. length of CpG islands, organism type, source database etc. My decoders do not consider any such parameters.

8. Conclusions

I found that it is possible to use a variant of Markov chains i.e. Hidden markov models to detect CpG islands in genomic sequences with a high degree of accuracy. The main issue is getting the trnsition matrix for the HMM that correctly reflects the presence and absence of CpG islands. If we have a set of sequences which CpG islands annotated then we can use MAP or MLE estimates to get the transition probabilities. On the other hand, if we do not have such annotated sequences we can use the Baum−

Welch algorithm to get the transition probabilites. The degree of accuracy obtained by the algorithms is very high.

9. References

[1] Durbin, R. M., Eddy, S. R., Krogh, A., and Mitchison, G. 1998. *Biological Sequence Analysis. Probabilistic models of proteins and nucleic acids*. Cambridge University Press.

[2] Rabiner, L. R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77(2):257−286.

[3] Churchill, G. A. 1992. Hidden markov chains and the analysis of genome structure. *Computers and Chemistry* 16(2):107−115.

[4] Krogh, A. 1994. Hidden Markov models for labeled sequences. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, 140−144. Los Alamitos, California: IEEE Computer Society Press.

[5] Eddy, S. R. 1996. Hidden Markov models. *Current Opinion in Structural Biology* 6:361−365.

[6] L. R. Rabiner and B. H. Juang. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, 3(1):4−16, January 1986.