CHAPTER IV

HIDDEN MARKOV MODELS IN CPG ISLANDS
IV. HIDDEN MARKOV MODELS IN CPG ISLANDS *

Abstract

We estimate the average length of a CpG islands and average length between two CpG islands. We detect the CpG island computationally by HMM, using Viterbi method, forward method, backward method and Baum Welch method. In learning problem Baum Welch method is a better method than others.

Literature review of HMM in CpG Islands

Rabiner.L.R 1989, described about Hidden Markov models and selected applications in biological sequences.


Tubingen 2004 gave details about HMM about sequences.

Daniel Huson 2005 has discussed CpG Markov chains.

C.Dirterich 2007 were discussed HMM in CpG Islands.

We discuss CpG Islands computationally by HMM algorithms.

* It is accepted for publication in Medwell Online International Research Journal
4.1 Introduction

In the human genome wherever the dinucleotide CG occurs (frequently written CpG to distinguish it from the C-G base pair across the two strands) the C nucleotide (cytocine) is typically chemically modified by methylation. There is a relatively high chance of this methyl-C mutating into a T, with the consequence that in general CpG dinucleotides are rarer in the genome than could be expected from the independent probabilities of the C and G. For biologically important reasons the methylation process is suppressed in short stretches of the genome, such as around the promoters or "start" regions of many genes. In these regions we see many more CpG dinucleotides than elsewhere, and in fact more C and G nucleotides in general. Such regions are called CpG islands. They are typically a few hundred to a few thousand bases long (Raúl Lozano, Julián Cobos).

A HMM can be seen like a finite state machine in which the following state only depends on the present state and we have a observations or parameters vector associated to each transition between states. It is possible thus to be said that a model of Markov have two processes associated: one hidden, no directly observable, corresponding to the transitions between states, and another observable one, whose accomplishments are the vectors of parameters that take place from each state and which they form the pattern to recognize (C.Dirterich).
Now, we can detect the CpG island computationally, here the different models of the problem and see which of them models it accurately enough to get the best results [24,76].

4.2 Model I

Merge the two Markov chains model+ and model− to obtain a so-called Hidden Markov Model. A hidden Markov model is similar to a discrete-time Markov chain, but more general and thus more flexible. The main difference is that when states are visited, these “emit” letters
from a fixed time independent alphabet. In our case of the localization of CpG-islands, we want to have both Markov chains within one model, with a small probability of switching from one chain to the other at each transition point. But here we face a problem: for each letter from our alphabet (i.e. nucleotide) we have two states, depending from which model they come. We resolve this by re-labeling the states such that we have the states $A^+, C^+, G^+$ and $T^+$ which emit $A$, $C$, $G$ and/or $T$ within CpG-regions, and vice versa states $A^-$, $C^-$, $G^-$ and $T^-$ which emit $A$, $C$, $G$ and/or $T$ within non-CpG-regions (Daniel Huson).

Figure 4.2 : Model for CpG Island
Application:

A table of transition probabilities generated for the CpG island states.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>.33</td>
<td>.37</td>
<td>.12</td>
<td>.16</td>
</tr>
<tr>
<td>C</td>
<td>.27</td>
<td>.12</td>
<td>.42</td>
<td>.18</td>
</tr>
<tr>
<td>G</td>
<td>.7</td>
<td>.27</td>
<td>.18</td>
<td>.36</td>
</tr>
<tr>
<td>T</td>
<td>.12</td>
<td>.16</td>
<td>.37</td>
<td>.33</td>
</tr>
</tbody>
</table>

Table 4.1: Transition probabilities

we have in all 32 (16*2) transition state without considering the transition for the CpG region to non CpG region. A simple way to find out if a given region is part of CpG island or not, we find the log likelihood by calculating the ration of the log odds of the two probabilities for the CpG and non CpG regions.

\[ L(M, N) = \log \left( \frac{p(MN/+)}{P(MN/-)} \right) \]

Given a region \( x = x_1 \ldots x_N \)

A quick-&-dirty way to decide whether entire \( x \) is CpG

\[ P(x \text{ is CpG}) > P(x \text{ is not CpG}) \Rightarrow \Sigma L(x_i, x_{i+1}) > 0 \]

So now if this value > 0, then we can say that the region is a CpG island else if this value < 0, We can consider it to be a non CpG island.
4.3 MODEL II

Let $P^+$ represents of staying within the CPG island states.

$P^-$ Represents the probability of staying the non CPG island states.

Each transition probability among states in the CPG will be scaled by $P^+$.

i.e., $a_{(A^+G^+)} = P^+ \cdot a_{(A^+G^+)}$

Let us consider transitions among the CPG and non CPG states. So the idea is when moving from a state in a CPG Island to one in non CPG there are two factors to account for

1. The probability of going to the non CPG region ($1-P^+$).
2. The probability of occurrence of that state in the non CPG island region

For example

$Q_{(A^+A^-)} = (1-P^+) \cdot q_{A^-}$

This is the proportion of A in the non CPG region. Similarly we can calculate the transition probabilities to non CPG region. But we need some way of estimating the value of $P^+$ and $P^-$ to find out these transition probabilities.
To find the length of distribution of each state in an HMM.

![Figure 4.3 Geometric distribution of States](image)

**Length distribution of region X:**

\[
P[l_X = 1] = 1-p,
\]

\[
P[l_X = 2] = p(1-p),
\]

...  

\[
P[l_X = k] = p^{k-1}(1-p)
\]

\[
E[l_X] = 1/(1-p)
\]

Geometric distribution, with mean \(1/(1-p)\) we can calculate the average length of this state as \(1/ (1-P)\) which shows that the length distributions of each state tend to fall geometrically (Tubingen).

We can now estimate the average length of CPG Island and the average length between two CPG islands to compute the required probabilities. Given such a model we can use the HMM algorithms is to compute the following.
Viterbi Algorithm

Initialization:
\[ V_0(0) = 1 \]
\[ V_k(0) = 0, \text{ for all } k > 0 \]

Iteration:
\[ V_l(i) = e_l(x_i) \max_k V_k(i-1) a_{kl} \]

Termination:
\[ P(x, \pi^*) = \max_k V_k(N) \]

Forward Algorithm

Initialization:
\[ f_0(0) = 1 \]
\[ f_k(0) = 0, \text{ for all } k > 0 \]

Iteration:
\[ f_l(i) = e_l(x_i) \Sigma_k f_k(i-1) a_{kl} \]

Termination:
\[ P(x) = \Sigma_k f_k(N) a_{k0} \]

Backward Algorithm

Initialization:
\[ b_k(N) = a_{k0}, \text{ for all } k \]
Iteration:
\[ b_l(i) = \sum_k e_l(x_{i+1}) a_{kl} b_k(i+1) \]

Termination:
\[ P(x) = \sum_k a_{0k} e_k(x_1) b_k(1) \]

1. Viterbi: To compute the best parse for a given sequence which can give us whether a particular nucleotide is in a CpG region or not given the complete DNA Sequence.

2. Posterior Decoding: If we want to find a more locally optimum prediction for a state given the sequence using forward/backward algorithms.

### 4.4 Learning problem in HMM

A newly sequenced genome of an organism, which we do not have much prior information. So, for example we do not know the CpG island regions in the genome but would surely like to train the parameters of our model to learn these values. So the learning problem in HMM involves updating the parameters \( \theta \) of the model to maximize the quantity \( P(x|\theta) \) (Daniel Huson).

There are two different types of learning scenarios depending on the type of data available with us. These are in some ways related to supervised and unsupervised learning scenarios in machine learning.
CASE I

**Estimation when the “right answer” is known**

We are given a set of training sequences along with labeled parser for the sequences. So the data set consists of a sequence $X = x_1 x_2 \ldots x_n$ for which the true parse $\pi = \pi_1 \pi_2 \ldots \pi_n$ is known.

In this case, the new parameters $\theta$ maximize the quantity $P(x, \pi | \theta)$. So, given labeled data, we can find out a very intuitive way to define the transition and emission probabilities of our HMM. If $A_{kl}$ represents the number of times $k \rightarrow l$ transitions occur in $\pi$ and $E_k(b)$ represents the number of times state $k$ in $\pi$ emits $b$ in $x$, The maximum likelihood parameters can just be the frequency count from this data.

So,

$$a_{kl} = A_{kl} / \sum_i A_{il} \quad \text{and} \quad e_k(b) = E_k(b) / \sum_c E_k(c)$$

The drawback of this scheme is that the parameters might be over fitted for the given training data so when new and more generalized data is provided, the model will not work well in those cases. Also, some of the probabilities might end up with a value of zero which is incorrect and must be set to a very small value in these cases. So, a way around this is to use “pseudo counts” and add them to the values in the equations above which might depend on the prior beliefs we have regarding these values.

**Examples:**

In general

$$A_{kl} = (\text{the number of times } k \rightarrow l \text{ transitions occur in } \pi) + n_{kl}$$
Given 10 casino rolls, we observe

\[ x = 2, 1, 5, 6, 1, 2, 3, 6, 2, 3 \]


Then:

\[ aFF = 1; \quad aFL = 0 \]

\[ eF(1) = eF(3) = 0.2; \]

\[ eF(2) = 0.3; \quad eF(4) = 0; \quad eF(5) = eF(6) = 0.1 \]

Given 10,000,000 casino rolls, we observe

a genomic region \[ x = x_1...x_{10,000,000} \] where we have good (experimental) annotations of the CpG islands

the casino player allows us to observe him, as he changes dice and produces 10,000 rolls

**Intuition:** When we know the underlying states,

Best estimate is the average frequency of transitions & emissions that occur in the training data

**Drawback:**

Given little data, there may be over fitting:

\[ P(x|\theta) \] is maximized, but \( \theta \) is unreasonable

0 probabilities – VERY BAD
CASE II

Estimation when the “right answer” is unknown

When we are given a set of training examples but without label, we thus have no knowledge of the true parse of the sequences and have to learn the parameters just by using all the sequences provided. So the $A_{kl}$ and $E_k(b)$ values cannot be directly computed as was possible in the previous case. The EM or Expectation Maximization algorithm is a general method of finding the maximum-likelihood estimate of the parameters of an underlying distribution from a given data set when the data is generally incomplete or missing. For our scenario the EM algorithm can be briefly outlined as:

1. Estimate $A_{kl}$, $E_k(b)$ in the training data
2. Update $\theta$ according to $A_{kl}$, $E_k(b)$
3. Repeat steps (1) and (2) until convergence

Examples

The porcupine genome, we don’t know how frequent are the CpG islands there, neither do we know their composition.

10,000 rolls of the casino player, but we don’t see when he changes dice.
4.5 Baum Welch Algorithm

This is a variant of the EM algorithm and its details are given below.

Algorithm steps
i. Initialize the parameter models arbitrarily
ii. Given a q
We have to estimate the values for $A_{kl}$ and $E_{l}(b)$ as in the previous case.
The idea is to find the probability of a $k \rightarrow 1$ state transition at a
particular position $i$ in the parse corresponding to the sequence. At each
position $i$ of sequence $x$, find probability transition is used. So,

$$P(\pi_i = k, \pi_{i+1} = 1 \mid x) = \frac{1}{P(x)} \times P(\pi_i = k, \pi_{i+1} = 1, x_1 \ldots x_N) = Q/P(x)$$

( By Conditional Probability rule)

Where

$$Q = P(x_1 \ldots x_i, \pi_i = k, \pi_{i+1} = 1, x_{i+1} \ldots x_N)$$

$$= P(\pi_{i+1} = 1, x_{i+1} \ldots x_N \mid \pi_i = k) P(x_1 \ldots x_i, \pi_i = k)$$

$$= P(\pi_{i+1} = 1, x_{i+1} x_{i+2} \ldots x_N \mid \pi_i = k) f_k(i)$$

$$= P(x_{i+2} \ldots x_N \mid \pi_{i+1} = 1) P(x_{i+1} \mid \pi_{i+1} = 1) P(\pi_{i+1} = 1 \mid \pi_i = k) f_k(i)$$

$$= b_l(i+1) e_l(x_{i+1}) a_{kl} f_k(i)$$

By using the Markov property and separating out the terms, we get
terms representing forward and backward computations. The figure
explains the formula well describing the transition from $k \rightarrow l$ being
composed of all the previous states and emissions followed by the
transition from $k$ to $1$ with the emission at $1$ and then all remaining state
transitions and emissions (Vignesh Ganapathy).
So, $A_{kl}$ is the $E[# \text{times transition } k \rightarrow l, \text{given current } \theta]$

$$A_{kl} = \sum_i P(\pi_i = k, \pi_{i+1} = l | x, \theta) = \sum_i \frac{f_k(i) a_{kl} e_l(x_{i+1}) b_l(i+1)}{P(x | \theta)}$$

\[ f_k(i) a_{kl} e_l(x_{i+1}) b_l(i+1) \]

\[ \sum_i P(\pi_i = k, \pi_{i+1} = l | x, \theta) = \sum_i \frac{f_k(i) a_{kl} e_l(x_{i+1}) b_l(i+1)}{P(x | \theta)} \]

\[ \text{Figure 4.4 Computational involved in Baum Welch Algorithms} \]

$$E_k(b) = \frac{1}{P(x | \theta)} \sum_{i | x_i = b} f_k(i) b_k(i)$$

So, compute forward, backward parts and using above formulae, compute $A$ and $E$

iii. The next step is to update the parameter model using these values and computing the log likelihood i.e $P(x | \theta_{\text{new}})$

iv. Repeat steps ii. and iii. till the difference in the results does not vary by much.

1. The algorithm requires the same running time as the forward and backward algorithms take for computation i.e is $O(k^2 N)$
2. Since this is based on the EM algorithm, it is guaranteed to improve the log likelihood result during each iteration.

3. The algorithm suffers from the drawback that it might end up at a local maximum instead of finding the optimum global maximum result.

4.6 Viterbi Algorithm

Simple approach to solve the problem is using the Viterbi algorithm. Our problem was mainly estimating the values of $A_{kl}$, $E_{i}(b)$ without knowing the actual parse for the given sequences. The Viterbi actually does give us the most likely parse for a sequence. So the steps for the algorithm are as follows.

i. Find the best parse $p^*$ using Viterbi.

ii. Now, similar to the first case, estimate the values for $A_{kl}$, $E_{i}(b)$ using the frequency count.

iii. Update the values of the HMM model based on these values

Repeat the steps ii and iii till convergence.

Conclusion

In Hidden Markov Models in CpG Islands, we have estimated the average length of a CpG islands and average length between two CpG islands. We detect the CpG island by HMM, using Viterbi method, forward method, backward method and Baum Welch method. Baum Welch method is a better method than others.