Computational Biology
Lecture 9: CpG islands, Markov Chains, Hidden Markov Models HMMs
Saad Mneimneh

Given a DNA or an amino acid sequence, biologists would like to know what the sequence represents. For instance, is a particular DNA sequence a gene or not? Another example would be to identify which family of proteins a given protein (amino acid sequence) belongs to.

In both cases above, we have a sequence of symbols from some alphabet and we are required to say something about the structure of that sequence. The same problem also arises in other areas of research such as pattern and speech recognition. For instance, a common task in speech recognition is to recognize the spoken word or sentence from a signal represented as a sequence of labels (quantization levels).

Therefore, we can adapt some of the techniques used in those areas, namely Markov Chains and Hidden Markov Models, which will serve as probabilitic models of sequences. We will concentrate on a famous biological instance of the general problem: to identify CpG islands in a DNA sequence where the alphabet consists of the four nucleotides A, G, C, and T. Next, we define what a CpG island is.

CpG islands

The CG pair of nucleotides (dinucleotide) is the most infrequent dinucleotide in many genomes. We will denote this dinucleotide by CpG to distinguish it from a C-G pair across the two strands of the DNA (recall that C binds with G).

The reason why the CpG dinucleotide is infrequent in the genome goes back to the fact that the C in CpG has a tendency to become methyl-C, by a process called methylation. Methyl-C in turn has a high chance in mutating to a T.

Therefore, due to the methylation process, the CpG dinucleotide is rarer than would be expected by the independent probabilities of C and G. The methylation process is suppressed in areas around genes and hence these areas contain a relatively high concentration of the CpG dinucleotide. Such regions are called CpG islands, whose length varies from few hundreds to few thousands bases.

The presence of a CpG island can be an indication to the start of a gene. Therefore, identifying CpG islands helps to determine the location of genes across the DNA. We would like to answer the following two questions:

- Question 1: given a short sequence, is it from a CpG island or not?
- Question 2: given a long sequence, does it contain a CpG island or not?

Markov Chains

Since we are looking for the CpG dinucleotide, modeling dinucleotides (rather than just individual symbols) in a sequence is now important. Previously, we modeled pairs of symbols using a joint probability distribution of alignments. This joint probability distribution was estimated from statistical data pertaining to alignments of many sequences. Why don't we just do the same here? In other words, why don't we just estimate the frequency of every dinucleotide and obtain a probability distribution for dinucleotides? We need to model pairs of symbols in the *same* sequence now, and not as a pair across two strands. Therefore, the second symbol of one pair, is the first symbol of the other, and there is clearly a new dependence that need to be captured.

Therefore, we need to build a model that generates sequences in which the probability of a symbol depends on the previous symbol. This way we will be able to capture the notion of two consecutive nucleotides on a strand. A suitable model for this purpose would be a Markov Chain, which is a collection of states with transition probabilities between states. In a Markov chain, the probability of the next state depends on the current state we are in. Here's the definition of a Markov chain:

Markov Chain

- Set of states Q
- For each pair of states i and j, a *transition* probability a_{ij}
- $\sum_j a_{ij} = 1$

We transition from one state to another in discrete time steps $n = 1, 2, 3, \ldots$. If we are in state i at time step n, we go to state j in time step $n+1$ with probability a_{ij} (the transition probability from state i to state j). We will also assume that the state at time n, x_n depends on the states x_0, x_1, x_2, ... only through the most recent state. We call this the Markov property.

Markov property: $p(x_n = j | x_0, x_1, x_2, ..., x_{m-1}, x_m = i) = p(x_n = j | x_m = i), m < n$. If $m = n - 1$ this is a_{ij}.

Figure 1 below shows an example of a Markov chain with two states.

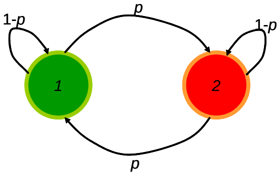


Figure 1: A Markov chain

For the Markov chain of Figure 1, $a_{11} = 1 - p$, $a_{12} = p$, $a_{21} = p$, and $a_{22} = 1 - p$. A Markov chain can be also represented as a matrix. For instance, the Markov chain in Figure 1 can be represented by the matrix

$$P = \left[\begin{array}{cc} 1-p & p \\ p & 1-p \end{array} \right]$$

The matrix representation P of a Markov chain is useful; for instance, it is easy to show that $p(x_n = j | x_m = i) = P_{ij}^{n-m}$, $m < n$.

Let us now look at the probability of obtaining a sequence $x_1 ... x_n$ from our Markov chain. This is basically the probability of a path $x_1 ... x_n$ in the chain. This can be expressed as the probability of starting in state x_1 and making successive transitions to x_2, x_3, ..., x_n. This is justified by the Markov property: $p(x_1 ... x_n) = p(x_n | x_1 ... x_{n-1}) p(x_1 ... x_{n-1}) = p(x_n | x_{n-1}) p(x_1 ... x_{n-1})$ $(Markov\ property) = a_{x_{n-1} x_n} p(x_1 ... x_{n-1})$.

Therefore, this probability can be expressed as:

$$p(x_1 ... x_n) = p(x_1) a_{x_1 x_2} ... a_{x_{n-1} x_n} = p(x_1) \prod_{i=1}^{n-1} a_{x_i x_{i+1}}$$

But what is the probability of starting in state x_1? This has to be given, so we must have a probability distribution for the starting state. Alternatively, we can model this by explicitly adding a start state with transition probabilities to all other states. We will always start with that special start state. Let the start state be denoted by 0, then $p(x_1) = a_{0x_1}$:

$$p(x_1 ... x_n) = p(x_0 = 0, x_1 ... x_n) = a_{0x_1} a_{x_1 x_2} ... a_{x_{n-1} x_n} = \prod_{i=0}^{n-1} a_{x_i x_{i+1}}$$

where $x_0 = 0$.

Similarly, we can explicitly add an end state (also denoted by 0 for simplicity). Although not needed, having an end state will help us model the length of the sequences too, becaise it will induce a probability distribution for the length of a path in the Markov chain. Therefore, each state (including the start state, i.e. empty sequence) will have a transition probability to that special end state. The probability of ending a sequence in state x_n is $a_{x_n 0}$. The probability of a path will be:

$$p(x_1 ... x_n) = p(x_0 = 0, x_1 ... x_n, x_{n+1} = 0) = a_{0x_1} a_{x_1 x_2} ... a_{x_{n-1} x_n} a_{x_n 0} = \prod_{i=0}^{n} a_{x_i x_{i+1}}$$

where x_0 and x_{n+1} are both 0.

We can model a sequence of dinucleotides as the Markov chain shown in Figure 2 below. Adding a start state and a possible end state to the Markov chain is also shown.

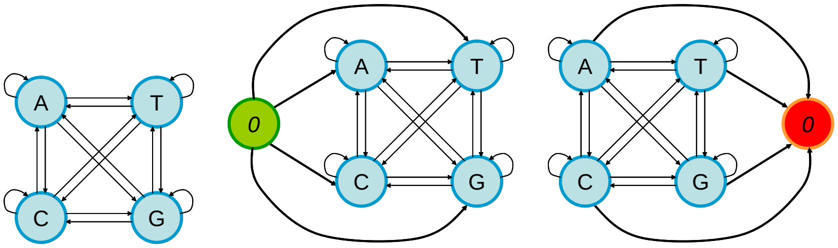


Figure 2: Markov chain for dinucleotides

Figure 2 shows the structure of the Markov chain. We still need to determine the parameters of the Markov chain, namely the transition probabilities between the states. These transition probabilities will have to differ in the case of CpG islands and non-CpG islands. For instance, in CpG islands we expect to see high transition probabilities to nucleotides (states) C and G.

We will estimate the transition probabilities from statistical data about CpG islands and non-CpG islands. We will therefore build two Markov chains, one for each. Then given a sequence, we compute the probability p of obtaining the sequence in the CpG island Markov chain, and the probability q of obtaining the seuqnece in the non-CpG island Markov chain. The odds ratio or log-odds ratio of these two probabilites can be used to determine whether the sequence is coming from a CpG island or not. Here are the steps:

- Bring a set of short DNA sequences labeled + for CpG islands and - for non-CpG islands (therefore these sequences are known to be either coming from CpG islands or not)
- For the CpG island Markov chain, estimate a^+_{ij} using $a^+_{ij} = \frac{c^+_{ij}}{\sum_k c^+_{ik}}$, where c^+_{ij} is the number of times nucleotide j follows nucleotide i in the sequences labeled $+$.
- For the non-CpG island Markov chain, estimate a^-_{ij} is a similar way.

Now given a sequence x, compute $p(x)$ for each Markov chain, denote these by $p(x|+)$ and $p(x|-)$. Then we use the log-odds ratio $\log \frac{p(x|+)}{p(x|-)}$ to determine if x is coming from a CpG island or not: If $\log \frac{p(x|+)}{p(x|-)} > 0$, the x is coming from a CpG island. Assuming that the transitions from the start state and to the end state are the same in both cases, the log-odds ration can be expressed as:

$$\log \frac{p(x|+)}{p(x|-)} = \log \frac{\prod_{i=0}^{n} a^+_{x_i x_{i+1}}}{\prod_{i=0}^{n} a^-_{x_i x_{i+1}}} = \sum_{i=1}^{n-1} \log \frac{a^+_{x_i x_{i+1}}}{a^-_{x_i x_{i+1}}}$$

Here's an example of transition probabilities for each of the two chains:

$$P^+ = \left[\begin{array}{ccccc} & A & C & G & T \\ A & 0.18 & 0.27 & 0.43 & 0.12 \\ C & 0.17 & 0.37 & 0.27 & 0.19 \\ G & 0.16 & 0.34 & 0.37 & 0.13 \\ T & 0.08 & 0.36 & 0.38 & 0.18 \end{array}\right] \quad P^- = \left[\begin{array}{ccccc} & A & C & G & T \\ A & 0.30 & 0.20 & 0.29 & 0.21 \\ C & 0.32 & 0.30 & 0.08 & 0.30 \\ G & 0.25 & 0.25 & 0.29 & 0.21 \\ T & 0.18 & 0.24 & 0.29 & 0.29 \end{array}\right]$$

Note that for the '+' chain (CpG islands), the transition probabilities to C and G are higher. Consider the sequence $CGCG$. The log-odds ration for this sequence is

$$\log \frac{0.27}{0.08} + \log \frac{0.34}{0.25} + \log \frac{0.27}{0.08} > 0$$

Therefore, $CGCG$ is judged to be coming from a CpG island.

We have developed a startegy to answer Question 1. What about Question 2? We can use the dual Markov chain model that we developed above to find CpG islands in a long sequence of nucleotides. Here's how: consider windows of small size, say 100, in the long sequence. For each window (a short sequence), compute the log-odds ratio as above. Therefore, we can identify windows with positive log-odds ratio and then merge intersecting windows to determine which parts of the long sequence are CpG islands.

The disadvantage of the above approach to Question 2 is that CpG islands tend to have variable length, and a window of 100 might not be appropriate to judge: If the window is too small, then we tend to have every occurence of CG as an island by itself. If the window is too large, then we do not achieve enough discrimination (the extreem case being whether the whole sequence is a CpG island or not corresponding to a window size equal to the length of the sequence).

A better way is to incorporate both models (CpG islands and non-CpG islands) into one model. Therefore, we will build a single Markov model consisting of both chains (+) and (-) described above as sub-chains, and with small transition probabilities between the two sub-chains. This is shown below. We rename the states by adding '+' and '-' labels to distinguish them. This relabeling is critical; otherwise, we cannot distinguish states of the new model.

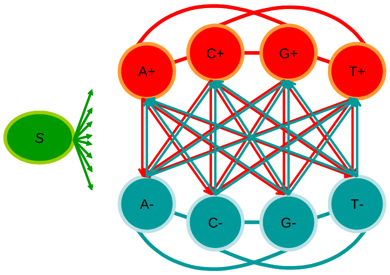


Figure 3: Combining both models

The advantage of this model is its trainability to reflect reality. As we have done before, we can estimate the transition probabilities between the two sub-chains by relying on known annotated sequences with all their transitions between CpG and non-Cpg islands. This way we remove the dependence on a particular window size.

The problem that we face with this new model, however, is that there it not a one-to-one correspondence between the states and the symbols of the sequence. For instance, the symbol C can be generated by both states C^+ and C^-. Hence, a sequence does not correspond to a path in the model anymore, but to multiple paths. In other terms, a sequence $x_1 ... x_n$ does not uniquely determines the path in the model. The states are *hidden* in the sense that the sequence itself does not reveal how it was generated. Therefore, we need to develop a slightly different theory for this new model, called Hidden Markov Model.

Hidden Markov Models

A Hidden Markov Model is defined as:

- A Markov Chain
  - Set of states Q
  - (transition probabilities) For each pair of states i and j, a *transition* probability a_{ij}
  - $\sum_j a_{ij} = 1$
- An alphabet of symbols $\sum$
- (emission probabilities) For each state k, and symbol b, $e_k(b) = p(x_i = b|\pi_i = k)$ (now we use variable π for states and variable x for symbols)
- $\sum_b e_k(b) = 1$ for each state k

Therefore, an HMM incorporates a Markov chain and decouples the states from the symbols: the probability of seeing symbol b in state k is called the *emission* probability of symbol b in state k, and denoted by $e_k(b)$. Therefore, a symbol can be emitted by many states with different probabilities. The emission probabilities for our HMM of dinucleotides shown in Figure 3 are straight forward. For instance $e_{A^+}(A) = 1$, $e_{A^+}(G) = 0$, $e_{A^+}(C) = 0$, and $e_{A^+}(T) = 0$. The transition probabilities between the (+) states as well as between the (-) states were determined previously. The transition probabilities between (+) and (-) states can be determined, as mentioned earlier, from previously annotated DNA sequences where transitions from CpG to non-CpG islands and vice-versa are known. Later we will look at the general problem of determining the parameters of an HMM. We also assume that the Markov property holds:

Markov porperty: $p(\pi_n = j|x_0, x_1, x_2, ..., x_m, \pi_0, \pi_1, \pi_2, ..., \pi_{m-1}, \pi_m = i) = p(x_n = j|\pi_m = i), m < n$. If $m = n - 1$ this is a_{ij}.

For now, we can consider the following three questions related to HMMs.

- evaluation: given a sequence x, what is the probability $p(x)$ of obtaining x in the model.
- decoding: given a sequence x, what is the most probably path that prodoces x in the model.
- learning: given a sequence x, what are the parameters (transition probabilities and emission probabilities) that will maximize $p(x)$ in the model.

We will start with the decoding question. After all, this is what we are interested in for our Question 2 of the CpG island problem: for a DNA sequence x, what is the most probable path that produced x in the given HMM. Recall that now the path is not unique, and one possible way of explaining how x was produced (hence knowing its composition of CpG islands) is to look at the most probable path in the model. We will later look at other ways of interpreting x, not necessarily looking for its most probable path. But now, let us concentrate on this task.

Before presenting the algorithm for obtaining the most probable path of x and computing its probability, let us consider a concrete example of an HMM known as the Dishonest Casino.

The Dishonest Casino: A fair die is used most of the times. However, the casino switches to a loaded die with a small probability, say 0.05, and switches back to the fair die with probability 0.1. The loaded die has $p(1) = p(2) = p(3) = p(4) = p(5) = 0.1$, and $p(6) = 0.5$. Given a sequence of rolls, can you tell when the fair die was used and when the floaded die was used? [try to think about the similarity between this problem and Question 2 of the CpG island problem]

The Dishonest Casino can be represented as an HMM with two states Fair and Loaded, each with different emission probabilities for the symbols 1 through 6.

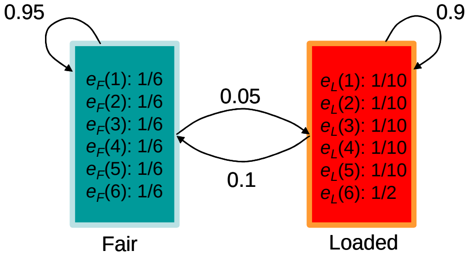


Figure 4: The Dishonest Casino HMM

Therefore, for a given sequence of rolls, we can ask what the most probable path is.

Now let us come back to our CpG island problem and consider the sequence $CGCG$. The probability of sequence $CGCG$ being emitted by the state sequence C^-, G^-, C^+, G^+ in our HMM model is

$$a_{0C^-}.1.a_{C^-G^-}.1.a_{G^-C^+}.1.a_{C^+G^+}.1.a_{G^+0}$$

In general, the probability of obtaining a sequence x on a particular path π is:

$$p(x_1...x_n, \pi_1...\pi_n) = p(x_1...x_n, \pi_0 = 0, \pi_1...\pi_n, \pi_{n+1} = 0) =$$

$$a_{0\pi_1} e_{\pi_1}(x_1).a_{\pi_1\pi_2} e_{\pi_2}(x_2)...a_{\pi_{n-1}\pi_n} e_{\pi_n}(x_n).a_{x_n 0} =$$

$$a_{0\pi_1} \prod_{i=1}^{n} e_{\pi_i}(x_i) a_{\pi_i \pi_{i+1}}$$

where $\pi_{n+1} = 0$. There are many state sequences (paths) that generate the same sequence $CGCG$ with different probabilities. We would like to find the most probable path $\pi^* = \arg\max_\pi p(\pi|x)$. Since we know how to compute $p(x,\pi)$ for a given sequence x and a path π (see above), it is useful to note that $\arg\max_\pi p(\pi|x) = \arg\max_\pi p(x,\pi)$. The proof of this fact is left as an exercise.

Of course computing $p(x,\pi)$ for every possible π and choosing the π that maximizes it is not efficient since we have an exponential number of paths. We will revert to a dynamic programming technique (yes... again).

Let $v_l(i)$ be the probability of the most probable path $\pi = \pi_1 ... \pi_i$ that generates $x_1 ... x_i$ and ends in state $\pi_i = l$. Obviously, we want the prbability $max_k v_k(n)$ that corresponds to the most probable path for generating $x_1 ... x_n$ (and ending in any state). We can express $v_l(i)$ as follows:

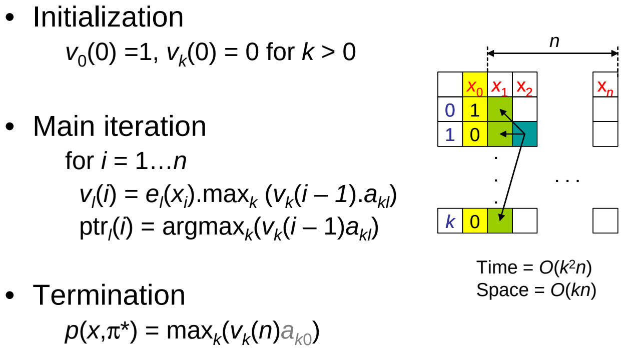
$$v_l(i) = e_l(x_i). \max_k(v_k(i-1)a_{kl})$$

The equation above says that the most probable path for generating $x_1 ... x_i$ ending in state l has to emitt x_i in state l (hence the emission probability $e_l(x_i)$) and has to contain the most probable path for generating $x_1 ... x_{i-1}$ ending in any state k, followed by a transition from state k to state l (hence the transition probability a_{kl}).

We can prove this mathematically as follows:

$$\begin{aligned}
v_l(i) = &\quad \max_{\pi_1..\pi_{i-1}} p(x_1...x_i, \pi_1...\pi_{i-1}, \pi_i = l) \\
= &\quad \max_{\pi_1..\pi_{i-1}} p(x_i, \pi_i = l, x_1...x_{i-1}, \pi_1...\pi_{i-1}) \\
= &\quad \max_{\pi_1..\pi_{i-1}} p(x_i, \pi_i = l | x_1...x_{i-1}, \pi_1...\pi_{i-1}).p(x_1...x_{i-1}, \pi_1...\pi_{i-1}) \\
= &\quad \max_{\pi_1..\pi_{i-1}} p(x_i, \pi_i = l | \pi_{i-1}).p(x_1...x_{i-1}, \pi_1...\pi_{i-1}) \\
= &\quad \max_{\pi_1..\pi_{i-2}k} p(x_i, \pi_i = l | \pi_{i-1} = k).p(x_1...x_{i-1}, \pi_1...\pi_{i-2}, \pi_{i-1} = k) \\
= &\quad \max_{\pi_1..\pi_{i-2}k} e_l(x_i)a_{kl}.p(x_1...x_{i-1}, \pi_1...\pi_{i-2}, \pi_{i-1} = k) \\
= &\quad \max_{k\pi_1..\pi_{i-2}} e_l(x_i)a_{kl}.p(x_1...x_{i-1}, \pi_1...\pi_{i-2}, \pi_{i-1} = k) \\
= &\quad \max_k \max_{\pi_1..\pi_{i-2}} p(x_1...x_{i-1}, \pi_1...\pi_{i-2}, \pi_{i-1} = k)e_l(x_i)a_{kl} \\
= &\quad \max_k(v_k(i-1)e_l(x_i)a_{kl}) = e_l(x_i)\max_k(v_k(i-1)a_{kl})
\end{aligned}$$

Therefore, we can compute $v_k(n)$ for any state k recursively to obtain the probability of the most probable path. By keeping pointers in the dynamic programming table, we can obtain the path itself. This dynamic programming algorithm in known as Viterbi decoding algorithm.



- Initialization

  $v_0(0) = 1$, $v_k(0) = 0$ for $k > 0$

- Main iteration

  for $i = 1 \ldots n$

  $v_l(i) = e_l(x_i).\max_k (v_k(i-1).a_{kl})$

  $\text{ptr}_l(i) = \text{argmax}_k(v_k(i-1)a_{kl})$

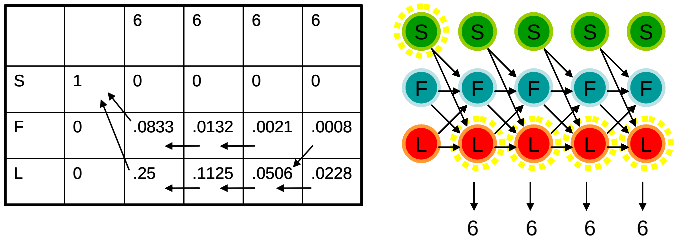
- Termination

  $p(x,\pi^*) = \max_k(v_k(n)a_{k0})$

Figure 5: Viterbi decoding algorithm

If there is no end state, then the term a_{k0} in computing the probability $p(x, \pi^*)$ is ommitted.

Here's an example of running Viterbi algorithm on the sequence 6666 for the Dishonest Casino problem:



		6	6	6	6
S	1	0	0	0	0
F	0	.0833	.0132	.0021	.0008
L	0	.25	.1125	.0506	.0228

Figure 6: Viterbi for Dishonest Casino

References

Durbin R. et al, Biological Sequence Analysis, Chapter 3.