02-510/710: Computational Genomics, Spring 2025

HW3: Epigenetics

Version: 1
Due: 23:59 EST, Mar 30, 2025 on Gradescope

Topics in this assignment:

1. Haplotype Inference
2. Metagenomics
3. Hidden Markov Models

What to hand in.

- One write-up (in pdf format) addressing each of following questions.
- All source code. If the skeleton is provided, you just need to complete the script and send it back. Your code is tested by autograder, please be careful with your main script name and output format.

Submit the following file which contain the completed code and the pdf file to gradescope separately.

./S2025HW3.pdf

Please note that all the solutions must be your own. We will check for plagiarism after the final submission.

1. **[20 pts] Haplotyping by hand**

   This will be a simple exercise in haplotyping that should be done by hand. Please show your work.

   After mapping reads and performing variant calling, you have filtered down to only heterozygous sites and encoded the variants in 0/1 format to denote wild-type/variant.

   Below each read is listed on a line, where '-' means that the read doesn't cover that variant locus.

```
----------------1101011111----------------------------
----------1010010010100001001111O----------------------
-----001000101101101-----------------------------------
0011000100010------------------------------------------
------101110100100101----------------------------------
------------------------------001010001010101----------
------------------------------1101011101010------------
-------------------------000010100010---------------
----------------------00000011110101------------------
-----------------------------------------01010001010---
-------------------------------------------1110101011
-----------------------------------------010001110100
----------------------------------00010101010---------
1100111010101------------------------------------------
---------------001010000000-------------------------
---------------------1111110010101--------------------
-------------------------------------10101010111010----
----------01011011010---------------------------------
```

   (a) (10pts) Assuming the reads come from a diploid organism, determine the two haplotypes.

   Solution

   Haplotype 1: 00110001000101101101011111110010101000101010100011101 00
   Haplotype 2: 11001110111010010010100000001101010111010101011100010 11

   (b) (5pts) What is the MEC score of your proposed diploid haplotypes?

   Solution

   10

   (c) (5pts) You should have noticed that in your solution, the two haplotypes are entirely complementary (if there is a 0 in haplotype A, then there is a 1 in haplotype B). Explain why this is the case.

   Solution

   Heterozygous sites in a diploid organism have different alleles on the two chromosomes. Since we're using 0/1 encoding to represent wild-type/variant, every position must have one 0 and one 1 across the two haplotypes.

2. **[40 pts] Metagenomic read binning**

In this problem, we will be building a simple metagenomic read classifer from scratch.

(a) (10pts) Inside the data folder is 4 genome fasta files, each of which are 1,000,000 bp long. Using $k = 4$, report the 10 most common kmers for each genome.

Solution

caat: 4104, gtgt: 4052, attc: 4043, atag: 4041, tccg: 4040,
ggag: 4039, cgga: 4037, taga: 4033, aggg: 4026, agag: 4023

(b) (10pts) The reads.fa file contains 1000 reads from the synthetic genomes. Using the most common kmers from each genome, design a classifier to determine the source genome of the reads. Describe your method below, and implement it with code. State the relative frequency of each genome in the reads file.

Solution

Approach: For each reference genome, extract the top 20 most frequent ("signature") 4-mers. Then, for each read, count all 4-mers in the read. Next, calculate a score for each genome by summing the occurrences of that genome's signature k-mers in the read and assign the read to the genome with the highest score.
Genome1 Relative Frequency: 0.2270 (22.70%),
Genome2 Relative Frequency: 0.2610 (26.10%),
Genome3 Relative Frequency: 0.2770 (27.70%),
Genome4 Relative Frequency: 0.2350 (23.50%)

(c) (10pts) An alternative method for binning metagenomic reads is to match them to discriminative kmers. A kmer will be discriminative if it is present in only 1 of the reference genomes. Using k = 10, report the 5 most common discriminative kmers for each genome.

Solution

Genome1 - 1. agacgcgaca; 2. gaacatttgc; 3. tagtgcgact; 4. taccatttgg; 5. tcaaaccgag
Genome2 - 1. ctaccccgta; 2. cttgaaaggg; 3. cgggcagcgg; 4. cagaaattgg; 5. agatgagaac
Genome3 - 1. catattggct; 2. gtggaatggc; 3. tagccaccat; 4. ggagacaggt; 5. ttagcgtaac
Genome4 - 1. cacgtacttg; 2. tcattcgttg; 3. tctagatgca; 4. agcataacca; 5. tacatgttcg

(d) (10pts) Similar to part b), design a new classifier, this time using discriminative kmers from each genome. Describe your method below, and then state the relative frequency of each genome in the reads file. (Hint: for this method, would recommend using the entire set of discriminative kmers for a given genome)

Solution

Approach: First, extract all possible k-mers from each genome and identify the discriminative k-mers. Then, for each of the reads, we extract all its k-mers and count how many discriminative k-mers from each genome are present. The read is assigned to the genome with the highest number of matching discriminative k-mers.
Genome1 Relative Frequency: 0.2570 (25.70%)
Genome2 Relative Frequency: 0.3080 (30.80%)
Genome3 Relative Frequency: 0.2060 (20.60%)

Genome4 Relative Frequency: 0.2290 (22.90%)

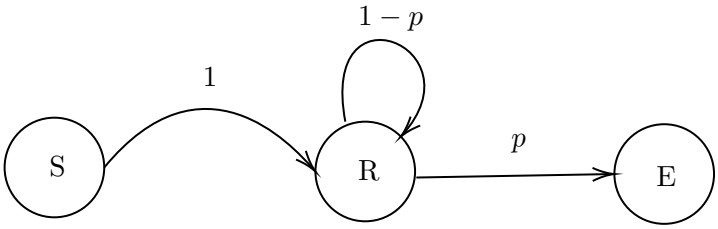


Figure 1: Random Genome HMM

3. **[40 points] Hidden Markov Models**

Warm-Up

(a) Consider the state transition diagram for the very simple HMM shown in Figure 3. The state S is a silent start state, and the state E is a silent end state. The state R, which stands for *random*, emits nucleotides with the following probabilities:

nucleotide	emission probability
A	0.15
C	0.35
G	0.35
T	0.15

The human genome is approximately 3.2 billion base pairs long. Suppose we generate a genome using the HMM in Figure 3. Find the value of p such that the expected length of this random genome is equal to the size of the human genome.

Solution

The expected length of the genome is determined by how many times we expect to visit state R before transitioning to the end state. Staying in R for $(n-1)$ steps gives probability $(1-p)^{(n-1)}$, and then the transition from R to E on the nth step has probability p. This gives us that generating a genome of length n has the following probability: $P(length = n) = (1-p)^{(n-1)} \cdot p$. The expected length is $E[len] = \sum_{n=1}^{\infty} n \cdot (1-p)^{(n-1)} \cdot p = \frac{1}{p}$. Now, $3.2 \times 10^9 = \frac{1}{p} \implies p = 3.125 \times 10^{-10}$

(b) Find the expected GC content of the genome generated in the previous part (write the answer as a percentage).

Solution

Probability of emitting C + Probability of emitting G = 0.35 + 0.35 = 0.70, so GC content = 70%.

Supervised Learning with HMMs

(c) Consider the problem of supervised learning using HMMs. Specifically, we are given a set of n observation sequences $O^{(i)} = o_1^{(i)}, \ldots, o_{T_i}^{(i)}$ drawn from an alphabet set O, along with state annotation data $Q^{(i)} = q_1^{(i)}, \ldots, q_{T_i}^{(i)}, (i = 1, \ldots, n)$, from a set of possible states Q. Here T_i is the length of the i-th observation. Show that the maximum likelihood estimates for the HMM are

$$\hat{a}_{st} = \frac{\sum_{i=1}^{n} \sum_{j=2}^{T_i} \mathbb{I}\left\{q_{j-1}^{(i)} = s, q_j^{(i)} = t\right\}}{\sum_{i=1}^{n} \sum_{j=2}^{T_i} \sum_{t' \in Q} \mathbb{I}\left\{q_{j-1}^{(i)} = s, q_j^{(i)} = t'\right\}},$$

and

$$\hat{e}_s(b) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{T_i} \mathbb{I}\left\{q_j^{(i)} = s, o_j^{(i)} = b\right\}}{\sum_{i=1}^{n} \sum_{j=1}^{T_i} \sum_{b' \in O} \mathbb{I}\left\{q_j^{(i)} = s, o_j^{(i)} = b'\right\}}.$$

Here, $\mathbb{I}$ is the indicator function, which takes value 1 when its argument is true and 0 otherwise.

Solution

For transition probabilities a_{st} (from state s to state t, and emission probabilities $e_s(b)$ probability of emitting observation b in state s), the likelihood of observing our data given the HMM parameters is:

$$L(a, e) = \prod_{i=1}^{n} P(O^{(i)}, Q^{(i)} | a, e)$$

We have the state sequences $(Q^{(i)})$, so we can decompose this:

$$L(a, e) = \prod_{i=1}^{n} \left[\prod_{j=2}^{T_i} a_{q_{j-1}^{(i)} q_j^{(i)}}\right] \left[\prod_{j=1}^{T_i} e_{q_j^{(i)}}(o_j^{(i)})\right]$$

Taking the log likelihood:

$$l(a, e) = \sum_{i=1}^{n} \left[\sum_{j=2}^{T_i} \log a_{q_{j-1}^{(i)} q_j^{(i)}} + \sum_{j=1}^{T_i} \log e_{q_j^{(i)}}(o_j^{(i)})\right]$$

We can rewrite this using indicator functions:

$$l(a, e) = \sum_{s \in Q} \sum_{t \in Q} \left(\sum_{i=1}^{n} \sum_{j=2}^{T_i} \mathbb{I}\{q_{j-1}^{(i)} = s, q_j^i = t\}\right) \log a_{st} + \sum_{s \in Q} \sum_{b \in O} \left(\sum_{i=1}^{n} \sum_{j=1}^{T_i} \mathbb{I}\{q_j^{(i)} = s, o_j^i = b\}\right)$$

Now, to maximize the likelihood estimates, we apply to constraints $\sum_{t \in Q} a_{st} = 1 \quad \forall s \in Q$ and $\sum_{b \in O} e_s(b) = 1 \quad \forall s \in Q$. For a_{st}, we take the derivative with respect to a_{st} and set to zero:

$$\frac{\sum_{i=1}^{n} \sum_{j=2}^{T_i} \mathbb{I}\{q_{j-1}^{(i)} = s, q_j^i = t\}}{a_{st}} - \lambda_s = 0$$

Solving for a_{st} gives:

$$a_{st} = \frac{\sum_{i=1}^{n} \sum_{j=2}^{T_i} \mathbb{I}\{q_{j-1}^{(i)} = s, q_j^i = t\}}{\lambda_s}$$

Applying the constraint that $\sum_{t \in Q} a_{st} = 1 \quad \forall s \in Q$ gives:

$$\lambda_s = \sum_{t' \in Q} \sum_{i=1}^{n} \sum_{j=2}^{T_i} \mathbb{I}\{q_{j-1}^{(i)} = s, q_j^{(i)} = t'\}$$

Plugging λ_s back into the expression for $a_s t$, we get:

$$\hat{a}_{st} = \frac{\sum_{i=1}^{n} \sum_{j=2}^{T_i} \mathbb{I}\left\{q_{j-1}^{(i)} = s, q_j^{(i)} = t\right\}}{\sum_{i=1}^{n} \sum_{j=2}^{T_i} \sum_{t' \in Q} \mathbb{I}\left\{q_{j-1}^{(i)} = s, q_j^{(i)} = t'\right\}}$$

and similarly, we have:

$$\hat{e}_s(b) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{T_i} \mathbb{I}\left\{q_j^{(i)} = s, o_j^{(i)} = b\right\}}{\sum_{i=1}^{n} \sum_{j=1}^{T_i} \sum_{b' \in O} \mathbb{I}\left\{q_j^{(i)} = s, o_j^{(i)} = b'\right\}}$$

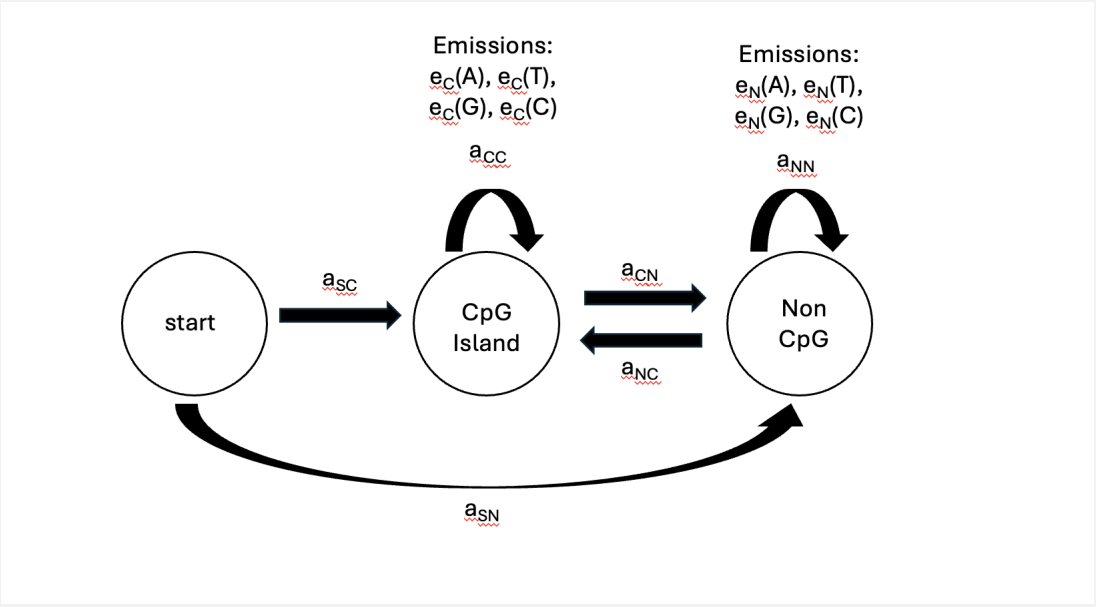
HMMs Application

CpG islands are DNA regions rich in cytosine (C) and guanine (G) nucleotides, commonly associated with gene promoter regions and regulation. Suppose you want to build a Hidden Markov Model (HMM) to recognize CpG islands within genomic DNA sequences.

(d) Design a simple Hidden Markov Model to model genomic sequences for CpG island detection. Using a graphical illustration of your HMM design and clearly define the hidden states and the possible observations.

Solution

The HMM has a silent start state denoted 'start', and two hidden states, 'CpG Island' and 'Non CpG'. The observations are the four nucleotides A,T,G, and C. As shown in the figure, $e_C(A), e_C(C), e_C(G), e_C(T)$ are the probabilities of emitting each nucleotide in the CpG Island state, and $e_N(A), e_N(C), e_N(G), e_N(T)$ are the probabilities of emitting each nucleotide in the Non-CpG state.



(e) After training your HMM, you receive an unlabeled genomic DNA sequence. Describe two methods you could use to identify and locate CpG islands within this sequence. Briefly explain how each method works.

Solution

1. Viterbi Algorithm - uses dynamic programming techniques to calculate the highest probability path ending in each possible state. For each nucleotide, it considers both the emission probability and the transition probability from the previous state. After processing the entire sequence, it traces backwards from the highest probability final state to reconstruct the complete state sequence.

2. Posterior Decoding - calculates the probability of being in the CpG Island state at each individual position in the DNA sequence, without committing to a single path through the model. It combines the forward algorithm (computes the probability of observing the sequence up to a position and being in a particular state) with the backward algorithm (computes the probability of observing the remainder of the sequence given the current state).

(f) Consider now that you have two separate biological conditions, e.g. healthy vs. diseased tissues, each potentially having different distributions of CpG islands. You wish to construct two HMMs, one per condition, from unlabeled sequences. What algorithm should you use to estimate the transition and emission probabilities for each model and why?

Solution

The Baum-Welch algorithm would be ideal for this scenario because it allows us to learn HMM parameters from unlabeled data through an iterative process. This is perfect when we have sequences from each condition but lack state annotations, i.e., don't know which regions are actually CpG islands. Applying this algorithm to sequences from healthy and diseased tissues allows us to develop two distinct models that capture the unique CpG island patterns in each condition.

(g) Suppose you have trained the above 2 HMMs. You wish to determine if a new given unlabeled genomic sequence is more likely to originate from healthy or diseased tissue. What method would you use to do this?

Solution

The Forward algorithm