## Background

In almost all eukaryotic organisms, the 2 base sequence of CG is the rarest of all of the nucleotide pairs (called dinucleotides). It is typically denoted "CpG", and the reason it is rare is because it often gets methylated, with the methylated Cs then becoming prone to mutating into Ts.

However, since methylation is typically suppressed around genes, CpG dinucleotides have a relatively high frequency near gene regions, and these clusters of CpGs are called CpG islands. The presence of a CpG island can therefore be an indicator of the start of a gene sequence, so when you are trying to find where the genes are in a genome, looking for CpG islands is a good place to start.

## Questions

Start by downloading the files that you will need to complete the exercise:

1. A list of 100 dinucleotides sampled from known CpG island regions on human chromosome 1: [cpg100.txt](https://uncc.instructure.com/courses/192992/files/20807195?wrap=1)

2. A list of 100 dinucleotides sampled from known non-CpG island regions on the same chromosome: [non100.txt](https://uncc.instructure.com/courses/192992/files/20807196?wrap=1)

3. An 1000bp long unknown sequence to test: [unknownSeq.txt](https://uncc.instructure.com/courses/192992/files/20812654?wrap=1)

### Question 1: Calculate the transition probabilities for the CpG data set. *[2 pts total]*

In the CpG data set, you have a list of 100 dinucleotides (1 per line) that have been sampled from regions of human chromosome 1. To build a Markov model for CpG islands, you need to use this data set to calculate the transition probabilities for each dinucleotide pairing. In other words, we want to know what the probability is of seeing nucleotide *j* next to nucleotide *i*.

Your transition matrix will look like this:

|  | **A** | **C** | **G** | **T** |
| --- | --- | --- | --- | --- |
| **A** | 3/16 = 0.1875 | 4/16 = 0.25 | 7/16 = 0.4375 | 2/16 = 0.125 |
| **C** | 5/35 = 0.1429 | 13/35 = 0.3714 | 10/35 = 0.2857 | 7/35 = 0.2 |
| **G** | 6/35 = 0.1714 | 12/35 = 0.3429 | 13/35 = 0.3714 | 4/35 = 0.1143 |
| **T** | 1/14 =  .0714 | 6/14 =  .4286 | 6/14 =  .4286 | 1/14 =  .0714 |

To calculate each transition probability, use the formula:

To help you check your method, I've given you the probability for pAA.

When you are done, show your completed transition probability matrix in your lab report.

### Question 2: Calculate the transition probabilities for the non-CpG data set. *[2 pts total*]

Do the same thing that you did in Question 1 to get the transition probability matrix for the non-CpG dinucleotide list.

|  | **A** | **C** | **G** | **T** |
| --- | --- | --- | --- | --- |
| **A** | 10/29 = 0.3448 | 5/29 = 0.1724 | 7/29 = 0.2414 | 7/29 = 0.2414 |
| **C** | 7/20 = .35 | 5/20 = .25 | 1/20 = .05 | 7/20 = .35 |
| **G** | 6/20 = .3 | 4/20 = .2 | 5/20 = .25 | 5/20 = .25 |
| **T** | 6/31 = .1935 | 6/31 = .1935 | 7/31 = .2258 | 12/31 = .3871 |

### Question 3: Get the log odds ratio for a short test sequence: **CGCG** *[2 points total].*

Recall that the equation for the log odds ratio is

In this case, P(x | model A) will be the probability of the sequence CGCG using the transition probabilities from Question 1 (the CpG model), and P(x | model B) will be the probability of the sequence CGCG using the transition probabilities from Question 2 (the non-CpG model).

Since we know that CGCG is a CpG sequence, we expect the CpG model to perform better than the non-CpG model, so this is a way to check that we set our models up correctly and they give us the right answer.

You do NOT need to worry about n-step probabilities, or steady-state calculations; this is just the standard probability of a sequence of events.

*Don't forget that we are interested in the probability of seeing one nucleotide come after another, so there are 3 "transitions" in a 4bp sequence.*

log(P(CG)/P(CG))+ log(P(GC)/P(GC))+log(P(CG)/P(CG)) = 1.748

### Question 4: Get the log odds ratio for a 100 bp sequence *[4 points total].*

Now, do the same calculation that you did in Question 3 for a longer CpG sequence:

CGGTCGGCTCAGCGGCTCCTGCCCTGGTCAGGGGGCGCCAGGTCCTGCCCCTCCTGGGGAGGGCGGGGGGCGAGAAGGGCGATTCTGGGGGCGGTTGCTC

The primary purpose of doing this is for you to come up with a more automated method to perform the probability calculations. In the next question, you will need to get the log odds ratio for multiple 100 bp sequences, so you do not want to do all of those calculations by hand!

This question also serves as another check that the models you set up in questions 1 and 2 can correctly identify a longer CpG sequence.

The overall log odds ratio for the 100 bp sequence is: 13.158617058006827.

### Question 5: Find the CpG island region in an unknown sequence *[5 pts total]*

For the last part of this lab, you will do a log odds ratio test in a *sliding window* across a 1000 bp sequence of unknown composition. You should use a window size of 100 bp, and a step size of 50 bp. The idea is to scan this sequence for regions that might be CpG islands (rather than test the whole thing at once).

#### Question 5A. How many total windows are you going to need to test? *[1 pt]*

There are 19 total windows which need testing

*Even though you don't have to, you may find it helpful to make a table or list of all of the start and end positions you'll need for each window. Even if you can calculate the number of windows without doing this, you might find this list useful for doing to full sliding window analysis.*

#### Question 5B. Does the first 100 bp window fit the CpG model or the non-CpG model? *[1 pt]*

#### *No it does not fit either the CpG model or non-CpG model*

*If you're unsure how to get the first window, then here are 3 different ways to do it in bash:*

*Option 1: Using the "cut -c" command will pull out a specific range of characters from 1 to 100:*

cut -c1-100 unknownSeq.txt

*Option 2: Using the "fold -w" command breaks a string into all substrings of the length you specify with "-w", and then piping (|) into the "head -n1" command gets the first one:*

fold -w100 unknownSeq.txt | head -n1

*Option 3: Reading the sequence into a unix variable lets you use Unix's internal substring command. In this example, I'll use the backticks (*

*) to capture the output of the cat command as a variable, then I use ${variable:position:length} to get the substring I want:*

seq=`cat unknownSeq.txt`

echo ${seq:0:100}

#### Question 5C. Calculate the log odds ratio for all of the windows. How many have an odds ratio >0? Which window has the highest odds ratio value? *[2 pts]*

I calculated **9** windows that have an odds ratio > 0.

The highest odds ratio value is in window **12**.

#### Question 5D. Roughly **where** in the unknown sequence is the location of the CpG island? (i.e., what are the start and end position(s) of the windows that show evidence for the CpG model)? *[1 pt]*

In the 1000bp “unknown sequence”, the CpG island appears to start at Window 11 (position 500-600) and continue to somewhere in Window 15 (position 700-800).