CS 309: Computational Biology

Fall, 2021

Lab 2: DosR binding site motif search in *Mycobacterium tuberculosis*

Due Wednesday, October 6

For this lab, you will work in groups of 2–3. All members of each group are expected to contribute equally to all aspects of the lab.

Introduction

Mycobacterium tuberculosis (MTB) can persist in a latent state in humans for many years before causing disease. Latency has been found to be linked to hypoxia (lack of oxygen) in the host. (Read pages 105–107 for more background.). You suspect that genes that are activated in hypoxia are regulated by a common transcription factor, so you collect the upstream sequences for all of the MTB genes that are upregulated in hypoxia, looking for the motif that corresponds to the binding site for the transcription factor regulating these genes. Your biologist colleague tells you that you should look at the 250 bp upstream region of each gene (which have been conveniently compiled for you in a FASTA file named `upstream250.fasta`). Your colleague also tells you that the motif is probably about 20 bp long.

Experimental Objectives

1. Carefully read Chapter 2 and then write a function that implements the randomized motif search algorithm that uses Gibbs sampling (page 101). Your function should return the best set of motifs.

2. Run your function with motif length 20 on the 25 MTB upstream sequences in the accompanying file `upstream250.fasta`. Experiment with different values of N to find one that seems to works well and also finishes in a reasonable amount of time.

   The following code snippet will read all of the sequences in the FASTA file into a list of strings:

   ```
   from Bio import SeqIO

   sequenceDict = SeqIO.index('upstream250.fasta', 'fasta')

   sequences = []
   for id in sequenceDict:
       sequences.append(str(sequenceDict[id].seq))
   ```

   The `index` function reads the sequences into a dictionary, and then the loop extracts them into a list of strings.

3. You should notice that your function gives you different results every time you run it. Randomized algorithms like this really need to be executed many (thousands) of times to get

reliably good results. Write another function that calls your motif search function repeatedly and keeps track of the best set of motifs over all of those trials. How many trials are necessary to get reliable results? Document your best set of motifs and their score.

4. Compute the consensus sequence for your motifs.

5. Also, use the WebLogo tool at `http://weblogo.threeplusone.com` to create a sequence logo for your motifs.

6. Next, you will use an online tool to perform the same analysis and compare the results. Upload the `upstream250.fasta` to the Consensus tool at `http://stormo.wustl.edu/consensus/html/Html/main.html`. Set the desired pattern width equal to 20 (keep all other parameters the same) and click "Submit." After the program has run, scroll to the bottom of the page and click "Next." Under "Matrix 1," you will see 19 sequences generated as a motif matrix. The elements in the column to the left of these sequences have the form `XXX/YYY`, where `XXX` is the sequence number and `YYY` is the starting position of each motif in the original string of length 250.

   Document these motifs, compute their score, and generate a consensus sequence and sequence logo for them, as you did above.

7. Finally, compare and contrast your results with those given by the Consensus tool. In your discussion, explore why Consensus only gave 19 motifs instead of 25.

Submission

In lieu of a formal lab report, supply your results/answers for each of the parts above. You may hand in one document for your group, but every individual is expected to contribute to all parts of the lab.

Please submit your report and the file containing your program.