

Problem Set 3

CS 174 Bioinformatics

The yeast species *Saccharomyces cerevisiae* has been used in baking and brewing for thousands of years. It is one of most intensively studied eukaryotic model organisms in molecular and cell biology. *Saccharomyces cerevisiae* was the first eukaryotic genome that was completely sequenced. The genome is composed of about 13 million base pairs and approx. 6,000 genes.

Just as in problem set 1, we will study the genome of *Saccharomyces cerevisiae* in this assignment. In particular we will identify regulatory motifs in promoter regions of the *S. cerevisiae* genes using simple enumeration-based methods.

Assignment

1. Create a file `ps3.py` which will contain your code for this assignment. Add some comment lines at the beginning of the file, indicating “CS174 Problem Set 3” and your name.
2. Download the following two files:
 - (a) http://www.ics.uci.edu/~xhx/courses/CS174/assignments/PS3/yeast_orfs_promoter.fa
 - (b) http://www.ics.uci.edu/~xhx/courses/CS174/assignments/PS3/yeast_cellcycle_gene_cluster2.txt

File (a) contains the promoter sequences of all *S. cerevisiae* genes, in FASTA format (see problem set 1). In this case, the word following the “>” symbol in each sequence’s description line is the ID for that sequence. The ID is typically the name of the gene for the sequence (e.g., YAL067C). Sometimes, however, the same promoter sequence is shared by two genes. In this case, the ID consists of the name of the two genes connected by the underscore symbol “_”. For example, a description line “>YAL065C_YAL064W-B” expresses that the sequence is the promoter of both gene YAL065C and gene YAL064W-B. (You can disregard the rest of the description line.)

File (b) contains a list of genes. These genes have been previously identified to share a similar expression pattern across different stages of the *S. cerevisiae* cell cycle. We will denote this set of genes by *G*.

3. Write a function that enumerates and returns all possible 6-mer motifs.
4. Write a function that returns the reverse complement of any 6-mer (you can reuse your code from problem set 1).
5. Write a main function that, using the previous two functions, does the following:

- (a) For each 6-mer, count the number of genes (in `yeast_orfs_promoter.fa`) whose promoter sequences contain the 6-mer or its reverse complement. Note that for each gene we only determine whether the gene contains the motif (or its reverse complement) or not: even if the gene contains multiple sites of the motif, it will be counted only once – but don't forget that some sequences are the promoter for two genes.

Extra credit: Any biological reasons to include the instances of the reverse complement? (You can add your answer as a comment in your Python code.)

- (b) Repeat the previous step but only for promoter sequences of the genes contained in the gene set G (from step 2 above).
- (c) For each 6-mer, define and calculate (using the counts you just computed) a p -value that quantifies the significance of the 6-mer's over-representation in the promoter sequences for the gene set G , with the entire gene set as a control.

As part of this you will need to compute probabilities under a hypergeometric probability distribution, as discussed in class. For easy computation of these probabilities we provide the following Python module:

- <http://www.ics.uci.edu/~xhx/courses/CS174/assignments/PS3/mystats.py>

Save this file to your working directory and add the line `from mystats import hyperGeom` to the beginning of your script `ps3.py`. In your computation of the p -values you can then call the function `hyperGeom(N, n, K, k)` to compute:

$$\text{hyperGeom}(N, n, K, k) := P(k; N, K, n) = \frac{\binom{K}{k} \binom{N-K}{n-k}}{\binom{N}{n}}$$

- (d) Rank all 6-mers with their p -values in ascending order. Output the top ten 6-mers with the smallest p -values. Record these ten 6-mers (and their respective p -values) to a file `ps3output.txt`.

6. Experiment with the MEME tool (which implements the EM algorithm):

- <http://meme.sdsc.edu/meme4.1/cgi-bin/meme.cgi>

Write down ten motifs discovered by MEME for gene set G .

Note: Unfortunately, the input on this website is limited to 60,000 characters, which does not fit all sequences for gene set G . You should therefore pick a random subset of sequences (for gene set G) that has overall length of less than 60,000 characters. Repeat this process a few times with different random subsets and merge the results.

7. Compare motifs discovered from the enumeration method you implemented against the output from MEME. Comment on the differences in a file `ps3comments.txt` (also record the ten motives you obtained from MEME).
8. Upload **all three** files `ps3.py`, `ps3output.txt`, and `ps3comments.txt` to the assignment submission folder in the EEE dropbox "CS174 Problem Set 3".

General remarks

- Please keep your code legible so we can read and grade it, if necessary. For instance, variables should have meaningful names. Use comments to explain principles and ideas, but don't comment on every single statement.
- Try to have complexity in mind when writing your code (this will become more relevant later on).
- We use **Python 2.x** for this class. Version 2.6.1 is the latest release, you can download it for free from <http://www.python.org/download/>. Version 3.0 has been released, but it is backwards incompatible, and 2.x it is still in wide practical use.