

BINF6410 Assignment 3:

Due date: **Monday, December 13th at 11:59pm**. A Dropbox folder will be available on Courselink for submission.

Your work will be evaluated on four axes:

1. **Scope:** The extent to which your code implements the features outlined in the specification.
2. **Correctness:** The extent to which your code is consistent with the specification and is free of bugs.
3. **Design:** The extent to which your code is well written (i.e. clear, efficient, elegant, logical).
4. **Style:** The extent to which your code is readable (i.e. commented, indented, containing aptly named variables and subroutines).

Background:

A gene's 5' flanking region has a major role in defining the timing, location, and level of its expression. These nucleotides include specific patterns typically 5–31 nucleotides long (mean 9.9 nt) that are binding sites for transcription factors that regulate transcript abundance. Identification of these regulatory sequences contributes to an understanding of gene regulation *in vivo* and contributes to efforts to artificially modulate gene expression. Several studies have identified putative functional regulatory sequences on the basis of their over-abundance in a set of co-regulated genes.

Transcription factors recognize conserved sequences in the promoter region upstream of a gene's transcription start site (TSS). Binding of the transcription factor initiates transcription of the downstream gene (actually, several transcription factors work in combination). If a set of genes share a common expression pattern, it is reasonable to assume that their transcription is activated by the same transcription factors, and that their promoter regions will share transcription factor binding sites (TFBSs). We call these sites "conserved non-coding regions", and because they are conserved we can find them using pattern-matching algorithms.









One strategy is to search promoter sequences for known motifs: i.e. ones that have been reported in the literature and experimentally confirmed to be transcription factor binding sites.

Assignment:

For this assignment I want you to search for known motifs in the upstream promoter regions of a set of coexpressed genes. The file `zea_mays_genes.txt` contains a list of maize (*Zea mays*) gene identifiers for a group of co-regulated maize genes. The file `promoters` contains a list of experimentally identified TFBSs that are active in maize. You will need to:

1. Obtain a fasta file containing the maize genome, version AGPv4.
ftp://ftp.ensemblgenomes.org/pub/plants/release-48/fasta/zea_mays/dna/.

You can also use ftp to collect these files.

| | | | |
|---|--|---------|----------------------|
|  | Zea_mays.B73_RefGen_v4.dna.chromosome.1.fa.gz | 86.9 MB | 7/10/20, 12:48:00 PM |
|  | Zea_mays.B73_RefGen_v4.dna.chromosome.10.fa.gz | 42.7 MB | 7/10/20, 12:48:00 PM |
|  | Zea_mays.B73_RefGen_v4.dna.chromosome.2.fa.gz | 68.9 MB | 7/10/20, 12:48:00 PM |
|  | Zea_mays.B73_RefGen_v4.dna.chromosome.3.fa.gz | 66.8 MB | 7/10/20, 12:48:00 PM |
|  | Zea_mays.B73_RefGen_v4.dna.chromosome.4.fa.gz | 69.6 MB | 7/10/20, 12:48:00 PM |
|  | Zea_mays.B73_RefGen_v4.dna.chromosome.5.fa.gz | 63.2 MB | 7/10/20, 12:48:00 PM |
|  | Zea_mays.B73_RefGen_v4.dna.chromosome.6.fa.gz | 48.3 MB | 7/10/20, 12:48:00 PM |
|  | Zea_mays.B73_RefGen_v4.dna.chromosome.7.fa.gz | 51.6 MB | 7/10/20, 12:48:00 PM |
|  | Zea_mays.B73_RefGen_v4.dna.chromosome.8.fa.gz | 51.2 MB | 7/10/20, 12:48:00 PM |
|  | Zea_mays.B73_RefGen_v4.dna.chromosome.9.fa.gz | 45.1 MB | 7/10/20, 12:48:00 PM |

2. Obtain GTF/GFF file for the same version to find the TSS for each gene.
ftp://ftp.ensemblgenomes.org/pub/plants/release-48/gff3/zea_mays
3. Write a Python script to extract the upstream non-coding sequence for each gene in `zea_mays_genes.txt`. 500nt is a reasonable size for maize promoter regions.
4. Using the list of known binding motifs in `promoters`, search for matches in the genes' promoter regions and report the number of times each motif was found.
5. Perform the same analysis on an equivalent number of randomly selected genes.
6. Compare the motif counts of the random and selected genes.

Submit any Python scripts that you use to do these tasks, as well as two separate files that list the counts of motifs from the selected genes and the random genes. The first column should be the motif and the second column should be the count, separated by tabs. If some motifs appear to be notably over or under represented amongst the coexpressed genes, please note them in a separate text file.

Hints and warnings:

1. Using SeqIO from Biopython can make reading in the large chromosome sequences faster, but one can also stick with python.
2. Be careful. Genes are also encoded on the reverse complement strand.
3. Some genes have multiple TSSs. Choose the most upstream one.
4. The promoter motifs sometimes have alternate bases, e.g. [AG]CCGAC means either ACCGAC or GCCGAC.

5. Consider double-checking that you are extracting the correct promoter sequences: e.g. blast a few of your promoter sequences against the genome and make sure that the positions in the blast output match the expected positions in the GTF/GFF file.
6. When the maize genome was assembled, missing sequence regions are replaced with a string of 100 Ns, so this pattern represents an unknown length of missing DNA. Be sure that your promoter sequences do not contain these Ns or sequences upstream of the Ns. Use a shorter promoter sequence if this happens.
7. Make sure to use functions. They can help keep your code clean and readable.
8. The randomly selected genes can be selected from the gff3 file.
9. Other than Biopython and re if desired, no modules that are not included with Python should be used.
10. Do not hard code any files or directories.