COMP 364: Homework 2

Matplotlib & Biopython (25 points)

Assigned: Feb. 15, 2013
Due on March 2, 2013 by email before 23:59pm

Submit all solutions in the form of python script files that the TA can run. Zip your solutions up into a compressed file called `<your name>_hw2.zip` *and email it to* `jerome.waldispuhl@mcgill.ca` *(indicate COMP364 in the subject).*

1 Heatmap gradients (5 pts)

1. Write a script `rb_gradient.py` that generates a vertical 100x100 red to blue (no black in between) (5 pts)

2. Write a script `bw_gradient.py` that generates a 100x100 black to white gradient along the diagonal (top left to bottom right) of a figure (5 pts)

2 Heatmap manipulation (5 pts)

In this problem you will work with the gene expression data we used in class (Use only the first 100 time points of the experiment). The negative values indicate under-expressed genes and the positive ones represent over-expressed genes.

1. Write a script `hmap.py` that takes your data file name as a command-line input and plots it as a red-black-green heatmap. (1 pts)

2. Write a script `hmap.py` that takes your data file name as a command-line input and plots the log of the data points as a red-black-green heatmap. As you may have noticed some values are negative. Let x be this value. Then, you will print the log of $x + 1$ the data if $x > 0$ (i.e. over-expressed) and the log of $\frac{1}{1-x}$ (or $-\log(1-x)$) if $x < 0$. What should be the value to plot if $x = 0$? Treat this case explicitly in your script. (2 pts)

3. Write a script `base_hmap.py` that takes two command line arguments: (1) your data file name and (2) the baseline (i.e., the baseline is no longer the median value between the minimum and maximum values). Your script should generate a plot such that the red-black-green heatmap has black (i.e. the median value) corresponding to the new baseline. Under- (in red) and over-expressed genes (in green) will thus be determined by this new baseline. To this end, you can use the optional arguments `vmax` and `vmin` when you call `imshow()` (`http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.imshow`) to center your scale. Ensure that at least the minimal *or* the maximal value is reached. (2pt)

3 Manipulating GenBank data (7 pts)

Download the GenBank record available at `http://www.ncbi.nlm.nih.gov/Sitemap/samplerecord`. Parse it and store in into a `Sequence` object and:

- Retrieve and print the complete gene sequence (1 point).
- Retrieve and print the coding (protein) sequences (there are 3 of them) included in that file (1 points).
- Retrieve and print the start position, end position, offset (i.e. codon short) and strand direction of each coding sequences. You should use the field `.features`. More informations can be found at : `http://biopython.org/DIST/docs/tutorial/Tutorial.html#htoc37` (2 points).
- Extract the coding sequences from the gene and transcribe them using the information retrieve above. Compare your results to the coding sequences (3 points).

4 Analyzing protein structures (8 pts)

Go to the Protein DataBank (`http://www.rcsb.org`) and Download the PDB data of the "Solution structure of the RMM-CTD domains of human LINE-1 ORF1p". Indicate its PDB id number.

- Parse the structure using the PDBparser implemented in Biopython (1 point).
- How many models this file contains? Print this number (1 point).
- We say that two residues are in contact their C_α are distant by at most 8 Å. Calculate and print the list of contacts for each model (2 points).
- Calculate and print the list of contacts that are conserved in all the models and half of the models (2 points).
- For each pair of indices, calculate the number of occurrence of this contact in the models. Print your results in a heat map named "contacts.pdf" (2 points).