7.36/7.91/20.390/20.490/6.802/6.874
PROBLEM SET 3. Gibbs Sampler, RNA secondary structure, Protein Structure with PyRosetta, Connections (25 Points)

Due: Thursday, April 3th at noon.

Python Scripts
All Python scripts must work on athena using /usr/athena/bin/python. You **may not assume availability of any third party modules** unless you are explicitly instructed so. You are advised to test your code on Athena before submitting. Please only modify the code between the indicated bounds, with the exception of adding your name at the top, and remove any print statements that you added before submission.

Electronic submissions are subject to the same late homework policy as outlined in the syllabus and submission times are assessed according to the server clock. **All python programs must be submitted electronically, as .py files** on the course website using appropriate filename for the scripts as indicated in the problem set or in the skeleton scripts provided on course website.

P1. Gibbs Sampler (10 Points).

You are studying longevity in two species, A and B. A study was recently published showing that a transcription factor called AGE is involved in regulating many aging- and stress-related pathways in both species A and B. AGE is known to affect transcription by binding to the promoters of its target genes. You have a list of aging-related genes whose expression changed (as measured by RNA-seq) in *age*(-) mutants relative to wild-type in species A. From a similar experiment, you obtained a list of genes whose expression changed in *age(-)* in species B. You want to look at the promoters of these genes to see if you can find any enriched sequence motif that might be a recognition site for AGE. You have two lists of sequences – `seqsA.fa` contains the 30bp upstream from the AGE target genes in A, and `seqsB.fa` contains 30bp upstream from the AGE target genes in B. To do this analysis, you will implement a Gibbs sampler! Once you are done modifying the script for parts A-D, please submit your completed `gibbsSampler.py` script to the Homework Submissions dropbox on the course website.

(A – 6 points) Download the skeleton code `gibbsSampler.py`. The script requires two inputs: the name of a FASTA file containing sequences believed to share a common motif, and the length of the motif. The main function `run` is called at the very bottom of the script; the argument `x` passed to `run(x)` is the number of iterations of the Gibbs sampler that will be run (initially set to 1).

You can run the script with the sequences from A as the input file and motif length 7 by running

```
% python gibbsSampler.py seqsA.fa 7
```

The skeleton code should run without errors, though it will mostly be telling you that various sub-functions haven't been implemented yet. Now fill out the skeleton code so that it successfully implements the Gibbs Sampler. Though you can use whatever approach you find best when completing the script, you may find the suggestions in `GibbsSamplerHelp.txt` helpful to walk you through which sub-functions you need to complete.

Once you've completed the code, set your Gibbs Sampler to run 1000 iterations and run it on the `seqsA.fa` file. Run the algorithm several (~10) times and pick the highest scoring run. For that run, report the background distribution, final weight matrix, motif score and relative entropy (you can just copy and paste from the output of the script as long as your solution is in readable table form; .doc provided on course website if helpful). Also, plot the relative entropy of the motif after each iteration. What is the shape of this graph? What is the consensus motif? (Hint: the highest scoring motif has a score over 600).

(B – 1 point) Weight matrices are not very visually informative for understanding a motif – Sequence Logos are more human friendly. Run your code again, using the `printToLogo()` function provided in the code to print out the final motifs for each sequence after the 1000 iterations are complete (you may want to comment out the other outputs). Go to http://weblogo.berkeley.edu/logo.cgi and paste the list of aligned motifs into the input box and hit Create Logo (note that the number of bits of information at each position doesn't take into account the background distribution of the sequences). Try this at least 3 different times with different runs of your Gibbs sampler, and print off or include a screenshot of each logo as well as the relative entropy and final score of that logo. Compare the results of your various runs. Briefly explain what types of differences you observe and why the Gibbs sampler returns these different motifs.

(C – 2 points) We now want to look for motifs in species B. Run the algorithm for length 7 and `seqsB.fa` several times and report the background distribution, final weight matrix, motif score and relative entropy (you don't need to plot RelEnt at every iteration) from a representative run. How does the relative entropy of the motif in species B compared to species A? Does this mean that the AGE motif is easier or harder to find in species B? Why?

(D – 1 point) Assuming there was still one occurrence of the motif in every sequence, what would happen if we increased the lengths of the sequences we are searching through? Run your Gibbs sampler on `seqsAext.fa`, which contains sequences of length 90 instead of 30. Run it several times and plot the relative entropy as a function of the number of iterations for a high scoring run. How is this plot different from part (a)? Can you explain the difference?

P2. RNA secondary structure prediction (5 points).

(A - 3 points) Use the Nussinov algorithm to find the secondary structure that maximizes the number of Watson-Crick base-pairs in the following RNA sequence (show your work):

CGAGUCGGAGUC

(B – 2 points) Run this sequence through the mfold RNA folding server (default parameters) at:

http://mfold.rit.albany.edu/?q=mfold/RNA-Folding-Form

Examine the top two structures it produces by looking at the "pdf"s under "View Individual Structures:". Notice that they don't match the structure predicted by the Nussinov algorithm. Examining the examples of real RNA secondary structures shown in lecture (slides 12, 32, 39 of Lecture 11), generate a hypothesis for which criteria used by the mfold algorithm to describe RNA thermodynamics prevents this algorithm from predicting the structure returned by the Nussinov algorithm in part (A). Test your hypothesis by strategically inserting adenosines at locations (e.g. in loops, between stems, across from bulges) necessary in the sequence above and finding the minimum number that must be inserted so that the top mfold-predicted structure has pairs between the same bases as in your Nussinov base pair-maximization structure.

(A – 1 point) Go to the Protein Data Bank (http://www.rcsb.org/pdb/home/home.do) and search for the protein we'll be working with – 1YY8. What is this molecule? By looking at the "3D View" tab of this protein, what is the predominant secondary structure (α-helix or β-sheet)?

(B – 1 point) In order to open and edit the `pyRosetta_1YY8.py` file, you will need to use a text editor on Athena's Dialup Service, which doesn't register commands from the mouse. If you're familiar with emacs or vim, these are great options, otherwise we recommend nano, a friendly text editor that can be navigated with the arrow keys. Open the file for editing by typing:

`nano pyRosetta_1YY8.py`

You can scroll with the arrow keys, and additional commands are shown at the bottom of the Terminal, where `^` is the Control key. Once you've made any changes, to exit and save, type "`Control + x`" followed by "`y`" and then Enter to signify yes, you want to save your changes.

Look over how the code for `part_b()` loads the PDB file `1YY8.clean.pdb` (which has been cleaned to remove non-atomic annotation information) and prints out the protein's angles and energy score. Then run the script from the command line with `python pyRosetta_1YY8.py` (note that it may take up a couple of minutes to load the module the first time you run it; it will also print out many lines of initialization warnings before printing out what is in the code). What is the energy of the structure, and what categories are the primary contributors (weighted score) for and against the total energy? (describe the categories in words as on the lecture slides, not just the shortened names that appear)

(C – 1 point) In the code for `part_c()`, Monte-Carlo side-chain packing is implemented. At the indicated region, add code to print out the post-packed energy score. At the bottom of the script, uncomment `part_c()` and run the script. What is the energy of the structure after packing, and which two categories have decreased the most compared to part (b)? (describe the categories in words as on the lecture slides, not just the shortened names that appear)

(D – 1 point) The `1YY8.rotated.pdb` structure is the same as `1YY8.clean.pdb`, except that one residue's phi or psi angle has been changed. Fill in the code for `part_d()` to load the rotated structure and perform side-chain packing on it, printing out the energy of the structure before and after side-chain packing. What is the energy before and after? Why is the post-side-chain packing energy not the same as that of part (c)?

(E – 1 point) Add code to `part_e()` to determine which angle (phi or psi, as well as which residue number this angle belongs to) is changed in the rotated structure. Which residue/angle is it?

(F – 1 point) Using the residue and angle you determined in part (e), add code to `part_f()` to determine the energy of the structure with that angle changed to each of the possible angles [-180,-179,….,180]. Which of these angles has the lowest energy, and does this agree with the corresponding angle in the original structure from part (b)?

Add each of these energies to the `energies_of_angle` list, and then modify `set_title` to include the residue and angle that you changed. If you've done this correctly, an `energy_vs_angle.pdf` plot will be written to the directory where the script is. If you're at an Athena workstation, you can view this PDF directly (Home Folder > pyRosetta_materials); if you've ssh'ed into Athena from a Terminal on your local computer, you can copy this PDF to your local computer and then open it in your computer's PDF viewer with the following commands using `scp` (secure copy):

```
<in a new Terminal on your computer, cd into your local computer's directory where you want to
download the PDF>

scp <your Kerberos username>@athena.dialup.mit.edu:~/pyRosetta_materials/energy_vs_angle.pdf .
```

P4. Queuing theory/connections (4 points).

A small bank hires a management consultant to figure out if they can afford to advertise free checking for one year (a $150 value) to any customer who has to wait in line more than 15 minutes. The consultant observes that, each minute that the bank is open, the waiting line gets longer by one customer with probability ¼, and – if there is a line – the line gets shorter by one customer (because a customer is served by a teller) with probability ¾. Let X be the probability that the line never gets longer than 10 people in a 12-week period (this is a reference case, whose empirical frequency is known to the bank), and let Y be the probability that the line never gets longer than 15 people in a 12-week period (the proposed duration of the promotion). The bank is open 2400 minutes every week.

Use an equation that was covered in class to calculate $\ln(X) / \ln(Y)$.

7.91J / 20.490J / 20.390J / 7.36J / 6.802J / 6.874J / HST.506J Foundations of Computational and Systems Biology
Spring 2014