20.320 Problem Set 3
September 25, 2009

#

General Instructions:

1. You are expected to state all your assumptions and provide step-by-step solutions to the numerical problems. Unless indicated otherwise, the computational problems may be solved using Python/MATLAB or hand-solved showing all calculations. Both the results of any calculations and the corresponding code must be printed and attached to the solutions. **For ease of grading (and in order to receive partial credit), your code must be well organized and thoroughly commented, with meaningful variable names.**

2. You will need to submit the solutions to each problem to a separate mail box, so please prepare your answers appropriately. Staples the pages for each question separately and make sure your name appears on each set of pages. (The problems will get sent to different graders, which should allow us to get the graded problem set back to you more quickly.)

3. Submit your completed problem set to the marked box mounted on the wall of the fourth floor hallway between buildings 8 and 16.

4. The problem sets are due at noon on Friday, October 2nd. There will be no extensions of deadlines for any problem sets in 20.320. Late submissions will not be accepted.

5. Please review the information about acceptable forms of collaboration, which was provided on the first day of class and follow the guidelines carefully.

20.320 Problem Set 3
Question 1

#

In class we discussed the role of GCSF in binding both bone marrow precursors and neutrophils. The interaction of interest is between the GCSF and the GCSF receptor (GR). Given this table from *Layton et al. JBC 274:25 pp.17445-17451* answer the following questions.

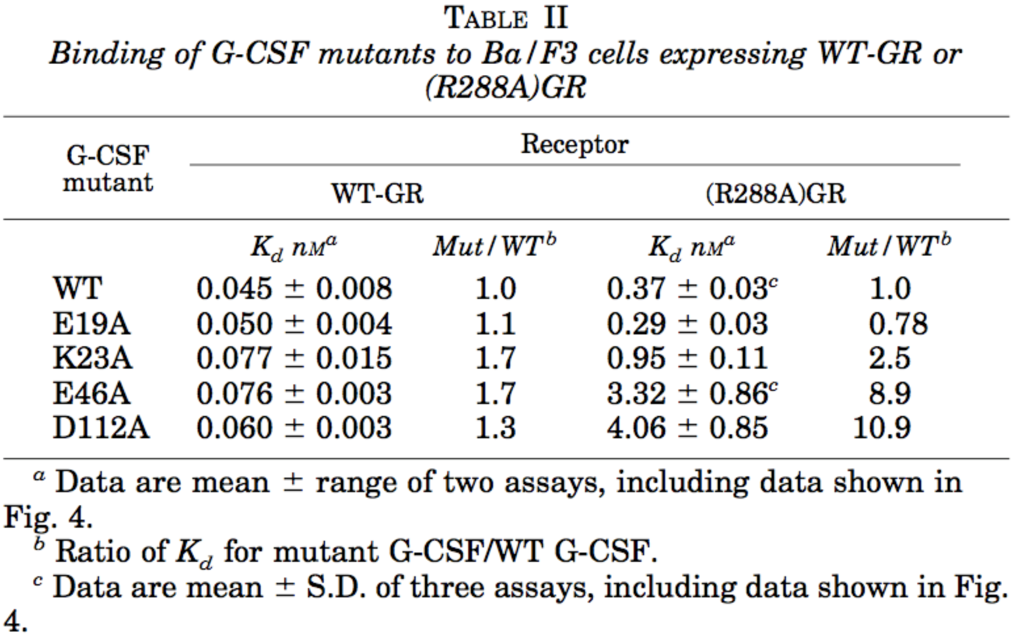


TABLE II
Binding of G-CSF mutants to Ba/F3 cells expressing WT-GR or (R288A)GR

G-CSF mutant	Receptor			
	WT-GR		(R288A)GR	
	K_d nM[a]	*Mut/WT*[b]	*K_d nM*[a]	*Mut/WT*[b]
WT	0.045 ± 0.008	1.0	0.37 ± 0.03[c]	1.0
E19A	0.050 ± 0.004	1.1	0.29 ± 0.03	0.78
K23A	0.077 ± 0.015	1.7	0.95 ± 0.11	2.5
E46A	0.076 ± 0.003	1.7	3.32 ± 0.86[c]	8.9
D112A	0.060 ± 0.003	1.3	4.06 ± 0.85	10.9

[a] Data are mean ± range of two assays, including data shown in Fig. 4.

[b] Ratio of K_d for mutant G-CSF/WT G-CSF.

[c] Data are mean ± S.D. of three assays, including data shown in Fig. 4.

This question will focus mainly on GCSF variants with wild-type GCSF receptor.

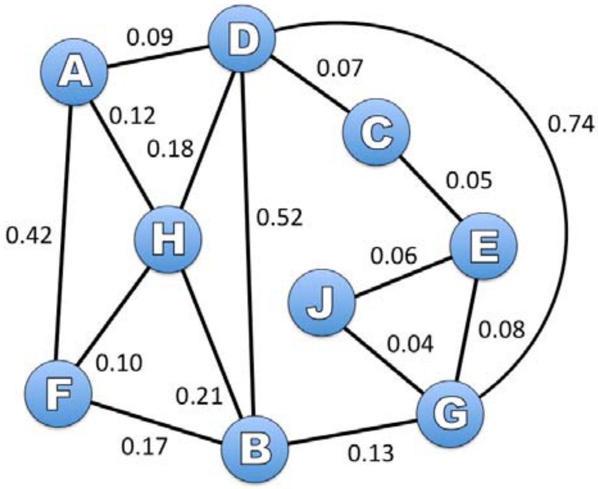
a) Draw out the four thermodynamic cycles for different GCSF mutants binding to the wild-type GR. Note that this question asks you only to draw the cycles – no calculations are required, but be sure to label the variables in your figures logically and accurately.

b) Given 100 pM concentration of WT-GR, calculate the fractional site saturation assuming an excess of ligand for wild-type GCSF and each GCSF mutant.

c) Compute the ΔΔGs in kcal/mol between all mutants (6 total ΔΔGs) at normal body conditions (37°C and 1 atm pressure). Note that $\Delta\Delta G^\circ_{i,j} = \Delta G^\circ_{\text{bind},j} - \Delta G^\circ_{\text{bind},i}$.

20.320 Problem Set 3
Question 2

In lecture, we discussed the use of graphs for representing biomolecular interactions. In a protein-protein interaction graph the ‘nodes’ represent proteins. Two nodes are connected by an edge if there is evidence that the two proteins interact. Consider the protein-protein interaction (PPI) network graph shown below. Protein nodes are colored blue and denoted by a one-letter character. Each edge is associated with a probability of interaction as shown.



To answer the questions below, you will need the NetworkX Python package, which is installed on Athena. If you wish to try this on your personal computer, the NetworkX zip file can be downloaded from *http://pypi.python.org/pypi/networkx/*. The folder networkx contained in the zip file will need to be placed in the same folder as the starter code provided.

a) As you can observe from the PPI network graph provided, there may be multiple ‘paths’ to reach any node from any other node of the network. Assume that each path represents a potential signal transduction pathway. The “length” of a path between two nodes is defined as the sum of the edge weights along that path. If the edge weights are set to the negative log of the probabilities ($w_{ij} = -\log_{10} p_{ij}$) show that the shortest path will be the one with the maximum joint probability.

b) Write a small piece of python code to accept the given graph as input and calculate the shortest path matrix (SPM) as output. The SPM is a matrix-representation format that is convenient to analyze PPI networks. Each element of the SPM represents the joint probability of the set of protein-protein interactions along the shortest path between two nodes as in the following equation:

$$\mathrm{SPM}(i,j) = \max_{P\ \in\ \text{paths from } i \text{ to } j}\left(\Pi_{e_{k\in P}}\, prob_k\right)$$

where the product is over all edges *k* in *P*, a path from *i* to *j* and *prob*$_k$ is the probability of edge *k*. You may assume that proteins of this network do not self-interact – that is no protein has any interaction with itself (0 probability). The website contains some code to help you get started with this problem called `networkstarter.py` You will have to submit your fully commented python code to receive credit for this problem.

This course makes use of Athena, MIT's UNIX-based computing environment. OCW does not provide access to this environment.

(Hint: The `NetworkX` module contains a function for computing the shortest distance between two nodes. Run `networkx.dijkstra_path_length(graph,start,stop)` to compute the sum of the edge weights along the shortest path between nodes `start` and `stop`. Remember to use weights equal to $-\log_{10} p_{ij}$ as defined in part (a).

c) Given that there are indeed multiple paths between most pairs of nodes, do you suppose that the SPM is the best indicator of protein-protein interactions in biological networks? Why or why not?

d) Another approach to analysis of PPI networks is to consider the multiple interaction paths between proteins of the network. Consider nodes A and G of the PPI network shown above and answer the following questions:

   i. List all the different paths (along with the corresponding probabilities of interaction) between nodes A and G, assuming that no path can be more than 6 nodes long, i.e. if the path includes more than 6 nodes you do not need to account for that particular path. You may represent the paths between nodes as a chain of letters the order of which gives the order of progression through the nodes (i.e. ACG means A to C to G). Further assume that no path can include any node more than once, i.e. that is paths such as ABFCADG are unacceptable since A occurs more than once.

   ii. Write an equation for the effective probability that at least one active protein-protein interaction path exists between two nodes in the network in terms of the probability of the edges. Note that prob(at least one active path exists) = 1 – prob(none of the paths are active). You may assume that each path is independent of other paths in the network.

   iii. Apply the equation you derived above to calculate the probability that at least one active path exists between nodes A and G of our network.

e) Let us now consider a medical application to this biological network analysis problem. Suppose that the interaction between protein nodes A and G of the provided PPI network is key to the progression of breast cancer. Identify a single node that, when inactivated, would maximally reduce the probability of any connection between A and G. For this problem, use the probability equations from question 2d and assume that when a protein is deactivated, the corresponding node and all its interaction edges are removed from the provided PPI network.

20.320 Problem Set 3
Question 3

#

In class, we discussed the Metropolis algorithm, which uses the Boltzmann distribution to sample states in order to find an absolute energy minimum. For this question, we will use the principles of the metropolis algorithm to find the absolute minimum of a polynomial function. To start, download the python file PolyEnergetics.py from the Course website.

The function `poly_energy()` in PolyEnergetics.py takes a single number as input and returns the value of the function $3x^4$-$4x^3$-$12x^2$+11 for that number.

a) Plot this polynomial function for *x* between -2 and 3. What are the minima of this function?

b) What minimum would you find if you implemented a gradient descent search starting at *x* = -2? What is the drawback in using gradient descent searches for energy minimization?

c) Complete the code `run_metropolis()` in PolyEnergetics.py to implement metropolis algorithm criteria to sample states (an x-value and it's `poly_energy` value) with the following specifications:

- The only input should be a float corresponding to a value of KT.
- The search should start at an x-value of -2.0
- Select the next *x*-value to test by generating a random number between 0 and 1. If the number is less than 0.5, decrease x by 0.1. Otherwise, increase x by 0.1.
- Decide whether or not to move to the test state based on the Metropolis criteria discussed in class.
- The function should run for 1000 cycles.
- The function should return a list of the energy values at the end of each cycle through the algorithm (i.e. 1000 entries per run).

d) Plot the list of energy values encountered during a search vs. cycle number at a KT of 0.1 and 5. Compare the behavior of the algorithm for these two KT values.

e) Run the `run_metropolis()` function 1000 times at a KT of 0.7, 2.0, and 5.0. How often does the function get within 0.1 of the global minimum at some point during the search? Explain your results.

20.320 Analysis of Biomolecular and Cellular Systems
Fall 2012