

20.320 — Problem Set # 6

October 29th, 2010

Due on November 5th, 2010 at 11:59am. No extensions will be granted.

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 be 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 the week after they were issued. 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.

PyRosetta

For this problem set you will be using PyRosetta. Information on PyRosetta and tutorials can be found here: <http://pyrosetta.org/tutorial.html>. To help you, we will introduce here some handy functions:

- First make sure your python code contains the following two lines in the header:

```
from rosetta import *
rosetta.init()
```
- `pose = Pose()` Creates a new variable of type `pose` (variable to handle proteins).
- To read a `pdb` file using PyRosetta, use the following command. This will load your `pdb` file into the variable `pose`.

```
pose = Pose("3ihw.pdb")
```
- `make_pose_from_sequence(pose, "ACEG", "fa_standard")` creates a protein sequence ACEG and loads it into variable `pose`.
- There are then a number of actions you can perform using defined functions for this variable type:
 - `pose.sequence()` gives you the full amino acid sequence of your protein
 - `pose.sequence(4)` gives the amino acid located at position 4
 - `pose.phi(4)` gives the phi angle of the 4th residue
 - `pose.psi(4)` gives the psi angle of the 4th residue
 - `pose.set_phi(4, -60)` set the phi angle of residue number 4 to -60°
 - `pose.set_psi(4, 30)` set the psi angle of residue number 4 to 30°
 - `pose.set_chi(1,4, 30)` set the χ_1 angle of residue number 4 to 30°
- To calculate the energy of a protein, you need to use a scoring function.
 - `scorefxn = create_score_function('standard')` creates a scoring function names `scorefxn`. If you are interested in how this function is defined, you can display its parameters using the command `print scorefxn`.
 - `scorefxn(pose)` evaluates the energy of your protein stored in the variable `pose`

For plotting you may choose to use Matplotlib or any other plotting software of your choice. Tutorials can be found here: <http://matplotlib.sourceforge.net/>

1 Protein engineering scaffolds

Molecules capable of specific, high affinity binding to a molecular target are vital to targeted therapeutics, diagnostics and other biotechnology applications. Antibodies have been largely the scaffold of choice, but also have a large number of shortcomings. In this problem, we will explore the structure of different protein engineering scaffolds: Ankyrin repeat protein (PDB ID: 2JAB), Affibody (PDB ID: 2B88) and 10th Fibronectin domain 3 (PDB ID: 1TTG). Structural data for proteins can be obtained from the Protein Data Bank (PDB): www.rcsb.org

- a) Give three disadvantages for the use of antibodies as therapeutics:
- b) Write a Python program to parse the PDB files and extract the Φ and Ψ angles.
- c) Residues in alpha helices have $\phi = -55^\circ \pm 10^\circ$ and $\psi = -60 \pm 10^\circ$ while residues in beta-strands have $\phi = -90^\circ \pm 30^\circ$ and $\psi = 120 \pm 30^\circ$. Given these allowed angle ranges, what fraction of residues are in an alpha-helical conformation? What fraction are in a beta-strand conformation?
- d) If you were to engineer a protein with specific binding properties using these scaffolds starting from scratch, which regions in these proteins would you mutate?
- e) **Extra credit:** Tryptophan is one of the most commonly found residue in binding interfaces. What property of tryptophan makes it especially suited for this role?

2 Surface energy

In this problem we will explore the conformation energy of two tri-peptide as a function of the χ_1 angles of the middle residue. We will consider the AVA and ARA peptide.

- a) Give 3 components of the scoring function and discuss their relative weights.
- b) Set the angles of your peptide so that it is in alpha-helical state.
- c) For both peptide, plot the energy as a function of the χ_1 angle. To do so, vary the χ_1 angle by 10° from -180° to 180° and calculate the energy of the peptide.
- d) What do you observe?

3 Interaction prediction

A number of crystallographic studies have shown that the binding interfaces between are generally large and include many intermolecular contacts. However, structural analysis alone cannot show whether all of these contacts are important for tight binding. For the determination of residues important for a binding interaction between two proteins, alanine screening is often performed. In this technique, each amino acid is substituted individually by an alanine. The impact in the interaction can then be measured and the importance of the residue can be assessed. In this problem, you will determine the importance of certain residues of hGH for binding to its receptor. Site 1 of hGH binds to its receptor with a K_D of 0.3nM, corresponding to a binding free energy of ΔG of -12.3 kcal/mol.

- a) Your alanine screening experiment gives the result listed in the table below. Calculate the change in free energy, $\Delta\Delta G$ associated with these mutations.

Mutation	K_D (nM)
R43A	0.6
E75A	0.25
I103A	0.9
W104A	3.2
I105A	1.05

- b) Given the $\Delta\Delta G$ for these mutations, which residues are the most important for binding?
- c) A different mutation on a residue that is burried inside the protein, W23A, yields a K_D of 120nM. Why is there such a dramatic effect on the K_D when this residue is known not to interact with the the receptor?
- d) What does this tell you about the limitation of alanine scanning?

4 Protein folding

The p53 tumour suppressor protein is a sequence-specific transcription factor whose function is to maintain genome integrity. Inactivation of p53 by mutation is a key molecular event, which is detected in more than fifty percent of all human cancers. In response to stress insults, the p53 tumour suppressor protein activates a network of genes whose products mediate vital biological functions, the most critical of which are cell cycle arrest and apoptosis. The large majority of the mutations affect residues that are critical for maintaining the structural fold of this a highly conserved DNA-binding (core) domain.

In order to measure the free energy of unfolding, one often uses denaturing conditions such as urea treatment. The free energy change for unfolding, ΔG_u at each denaturant concentration can be calculated as follows:

$$\Delta G_u = -RT \ln K_u = -RT \ln \frac{[U]}{[F]}$$

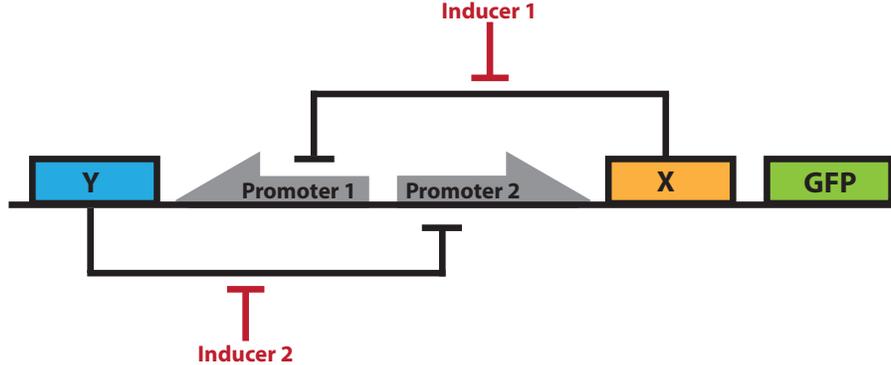
Where [U] represents the concentration of unfolded protein and [F] of folded protein. However, we are interested in the unfolding free energy in absence of denaturing conditions. Denaturant-induced unfolding transitions are most commonly interpreted assuming a linear relationship for the denaturant (C):

$$\Delta G_u(C) = \Delta G_u^0 - m \cdot C$$

Given the data measured below at 25°C, what is the free energy of unfolding of this protein?

[Urea] (M)	Fraction of unfolded protein
2.0	0.15
2.5	0.28
3.0	0.35
3.5	0.60
4.0	0.72
5.0	0.83

5 Qualitative analysis of a nonlinear biochemical circuit



Consider the synthetic genetic circuit above. X and Y are homooligomeric transcription factors mutually repressing each other's expression, with $x = [pX]$ and $y = [pY]$. Green fluorescent protein is a readout for the state of the system. Inducers 1 and 2 can be added to perturb the system.

- a) Recall this differential equation for transcriptional self-activation with a monomeric active form of the TF:

$$\dot{x} = \beta_0 + \beta_x \frac{x}{K_D + x} - \alpha x$$

Write down a set of two analogous differential equations describing homooligomeric, mutual repression of two TFs (no derivation required).

- b) To simplify the equations, make the following assumptions:

- Both proteins are degraded or diluted at the same rate $\alpha_x = \alpha_y = 1$.
- For both promoters, the maximum rate of expression is $\beta_x = \beta_y = 10$.
- Basal (leaky) expression for both promoters is $\beta_{0,x} = \beta_{0,y} = 0$.
- The effective K_D is $K_{D,\text{eff},x} = K_{D,\text{eff},y} = 1$.
- The active forms of both repressors are homodimeric.

Write down the simplified set of ODEs.

- c) Find the critical points (x_0, y_0) of the system where $(\dot{x}, \dot{y})|_{x_0, y_0} = (0, 0)$.

HINTS: 1. You will come across a higher-order polynomial equation which you will need to solve. It is solvable by hand, but do feel free to use <http://www.numberempire.com/equationsolver.php> or an online equation solver such as <http://www.numberempire.com/equationsolver.php>.

2. Only positive, real-valued roots of that polynomial are biochemically meaningful steady states. Solutions which are negative or complex are mathematically legitimate, but such concentrations of the species in question will never occur; thus, these solutions can be disregarded in the rest of this problem.

- d) Find the Jacobian matrix $\mathbf{A} = \begin{pmatrix} f_x & f_y \\ g_x & g_y \end{pmatrix}$, where $\dot{x} = f(x, y)$ and $\dot{y} = g(x, y)$.

- e) Find the trace τ and the determinant Δ of the Jacobian matrix. Then, evaluate both at each of the critical points.

- f) For each critical point,
- i) Calculate the eigenvalues λ of the Jacobian matrix at that point.
 - ii) Use the eigenvalues to classify the critical point as a stable or unstable equilibrium, a stable or unstable spiral, the center of a periodic orbit, or a saddle.
- g) By hand or with the help of MATLAB (but *without* numerical simulation), sketch a phase portrait of the system:
- i) Use x as your horizontal and y as your vertical axis.
 - ii) Sketch the nullclines $f(x, y) = 0$ and $g(x, y) = 0$ and mark all critical points. MATLAB might help to do this accurately.
 - iii) Use your earlier analysis of the stability of each critical point to sketch local, nearby trajectories around each critical point (best done by hand).
 - iv) Connect the local trajectories to extend throughout the phase plane.
 - v) Looking at this phase portrait, can you guess why this circuit might be preferable to the bistable switch implemented via sigmoidal self-activation in the last problem set?
- h) All else remaining unchanged, what happens to the system if $\beta_y = 4$?

MIT OpenCourseWare
<http://ocw.mit.edu>

20.320 Analysis of Biomolecular and Cellular Systems
Fall 2012

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.