Homework Assignment #7

DUE in class, Tuesday, November 14

To date we have been using force fields to calculate the **potential energy** of a molecular configuration or conformation at each step. If we were to calculate the potential energy of a native, folded protein structure and the potential energy of the same protein in an extended, unfolded conformation the difference would not tell us very much about the stability of the folded protein because we would not know what the difference in entropy is.

To calculate the stability of a protein we need to know the **free energy** difference between the two states. It turns out that a major contribution to the free energy change during protein folding involves a change in interaction of the protein with water which can be described by the **solvation free energy**. This is because the entropy of the solvation water changes dramatically upon protein folding. In fact, just calculating the difference for water-protein interaction between folded and unfolded states gives us a rough estimate for the total free energy of protein folding. Of course, this would ignore internal hydrogen bonds, internal VDW interactions and protein charge-charge interactions. But it turns out that hydrophobic effects along with entropy effects are the major driving forces for protein folding so this method gives a rough estimate of the folding free energy change for a protein.

Solvent accessible surface area (ASA) has been used successfully to estimate the solvation free energy of small molecules, especially hydrophobic molecules. It has been found that the solvation free energy is approximately +0.023 kcal/mol/$Å^2$ of hydrophobic accessible surface area (i.e. it is unfavorable). Solvation free energy is a ΔG, the difference between two states.

When a protein folds from a well solvated, extended conformation to the native compact globular conformation significant hydrophobic surface area is buried in the interior of the protein – an energetically favorable process. The removal of hydrophobic surface from solvent exposure contributes to the favorable free energy of folding and helps stabilize the protein.

Your task is to calculate the extent of hydrophobic surface buried upon folding of a protein and calculate the associated favorable free energy contribution to folding. Assume that all carbon atoms are hydrophobic and all other atoms are polar (an oversimplification but good enough to make the point here). Therefore, you will have to sum the the ASA values for all carbon atoms, C, CA, CB, etc. in both the folded and unfolded states of a protein and calculate the difference in ASA. The tarred archive on the cluster, */home/compbio2/Shared/asa.tar*, contains the following files,

1. *A.pdb* – Protein in the folded conformation
2. *B.pdb* - Protein in the unfolded conformation
3. *asa.py* – Program to calculate the ASA (python2 code)
4. *average.py* – Program to calculate statistics of numerical vectors (python3 code)

Unpack this archive in your */home/combio2/[JHEDID]/* directory on the cluster for completion of this assignment.
(You are encouraged to fetch these two PDB files back to the Mac to look at them with PyMOL. Although this is not necessary to complete the assignment, it's always a good idea to know what your PDB file looks like in graphics before using it for analysis.)

Hand in a hardcopy printout containing the following columns of information for **all** atoms in the PHE 30 residue in the above protein structures:

```
Atom#  AtomName ResidueName FoldedASA UnfoldedASA ASADifference
```

and at the bottom provide a value for the calculated free energy change due to hydrophobic burial in kcal/mol for the **entire** protein. (Note the sign of your numerical answer – this is the free energy of hydrophobic **burial**).

Tools for this assignment are the following:

- The program **asa.py** takes a PDB file as input and will print to the screen a new PDB file in which the ASA ($Å^2$) of each atom is in the B-factor column. (Note: use python2 for this script; you can use the master node)
- The program **average.py** takes as input a file containing a one dimensional numerical array (i.e. a one column list of numbers) and will print to the screen the N, Sum, Mean, S.D., Minimum, Maximum and |Sum|. (Note: use python3 for this script; you can use the master node)
- The command, **`grep ' C' filename`**, (note the white space before the C) will print to the screen all lines in a file containing a capital letter "C" preceded by a blank space.
- The command, **`awk '{print $10}' filename,`** will print column 10 of a file to the screen.
- You can combine grep and awk commands using the pipe. For example the following command,
  `grep 'PHE  30' A.asa | awk '{print $10}'`
  will print to the screen column 10 (B-factor) for the phenylalanine_30 atoms. (Note that there are 4 spaces after PHE).
- The command, **`paste filename1 filename2 > filename3`**, will combine side by side the contents of *filename1* and *filename2* into a new file called *filename3*.
- The command, **`awk '{print ($1 - $2)}' filename`**, will print the difference of columns 1 and 2 to the screen if both columns are numbers.