

# Biological Computation 20.181

## Homework 8

Using code other programmers have written is key to being an effective software developer, but learning to use someone else's codebase requires an initial time investment. In this homework, you will take advantage of an existing codebase to perform some basic protein modeling tasks. In the next homeworks, you will extend this codebase to build protein structures. First, however, this codebase does not have the capability to build structures. You will need to supplement this code with your code from HW7. Your function should follow the header found in the file "Structure.py".

```
def setVectorFromDihedral(v1, v2, v3, len34, ang234, dih1234):  
    """ This method takes 3 dependent atoms and returns vector of target  
        atom's location given bond length, angle and dihedral.  
    """
```

First, take a quick survey of the new modules and data files provided. Take a close look at the documentation provided in the headers to each function and at the top of each file. There are two important data files ("top\_all27\_prot\_na.inp", and "bbind02.May.lib"). The first "topology" file details the chemistry and geometry of building atoms, and the second "rotamer" file is a list of possible rotamers (torsion angles) for each amino acid type.

Now take a close look at the `__init__` function for creating Protein objects. Create a Protein with sequence "ACDEFGHIKLMNPQRSTVWY" and assign it to the variable `p` in the `hw8.py` script. Next, create a PDB file describing this structure called "allRes.pdb".

Now look at your newly created protein! (You may find the visualization software at <http://pymol.sourceforge.net/> handy for this, but feel free to use any protein/pdb viewing software you prefer). It should look like an extended backbone with one of each amino acid. If it doesn't, you may need to doublecheck your function for building atoms.

Finally, you will load in a protein from a pdb file, and measure its rotamer angles. You will need to complete the following code for the output as shown below:

```
#loop over residues in q  
# print out amino acid:list of dihedral angles in rotamer 0  
for res in q.getResidues():  
#insert your code here
```

### example output:

```
P:PRO: [16.088498149864904, -14.920954996245952]  
S:SER: [60.85353155923768]  
N:ASN: [51.850742669823859, -32.006293389354667]  
Y:TYR: [-92.806750276209897, 59.460942679610618]  
V:VAL: [-60.839303316853218]  
A:ALA: []
```