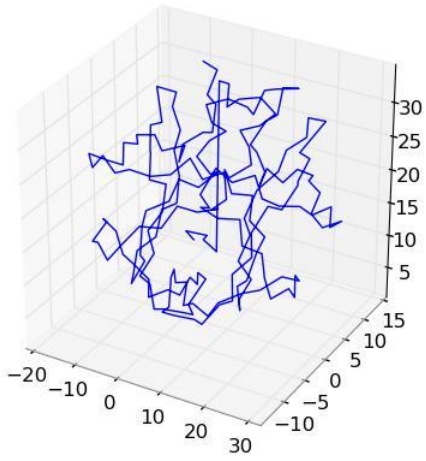
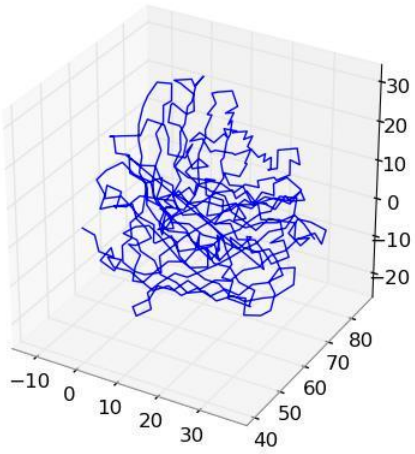
Assignment 5

Due on April 9 at 11:55 p.m.

A protein is a molecule that is a sequence of amino acid residues. The Protein Data Tank records the 3D structures known for protein molecules (85,058 as of 02 October 2012). Two examples are HIV Protease (7hvp), an important AIDS drug target, and green fluorescent protein (1gfl), which earned its discoverers the Nobel Prize. Many types of information are stored in a PDB file; we will be interested only in lines that start with 'ATOM', and only in certain columns of these lines. Their format is described later.





Tasks

1. Inside the A4 folder you'll find the two .pdb files and a script template to get you started. You should edit this script to produce the results requested below. When the I run your script it should produce the graphs requested and it should report how many hydrogen bonding pairs there are for each protein.

2. Write a function `readPDBfile(filename)` that will read the atoms for a protein stored in pdb file whose name is given as the argument. Your function should return a Python tuple containing 4 values: `(anum, aname, resno, coords)`.
   a. `anum` should be an array with the serial number for each atom.
   b. `aname` should be an array of strings giving the 4-letter atom name for each atom. The strings in `aname` should be in upper case.
   c. `resno` is an array with a residue sequence number for each atom.
   d. `coords` is an n-by-3 array with xyz coordinates (in angstroms) for each atom.

3. Write a function `drawCA(aname, coords)` that uses `Axes3d.plot` to draw the Calpha backbone of the protein: it should connect atoms with name `' CA '` in

sequence, and ignore all the other atoms.

4. Write a function `Hbonds(anum, aname, resno, coords)` that looks for pairs with a Nitrogen (second letter of the name is `'N'`) and Oxygen (second letter is `'O'`) atoms whose distance is between 2.6 and 3.2 angstroms, inclusive. Such a pair is deemed to form a hydrogen bond if the residue sequence numbers of these atoms differ by at least 2. Return a list containing the pairs of atom numbers for hydrogen bonding pairs.

Opening and Reading a File

Reading a file such as '7hvp.pdb' (for HIV protease) is easy in Python. You can simply say:

```
for line in file('7hvp.pdb', 'r'):
    # do something for every line
```

The first argument is a string giving the name of the file. The second argument to the file function is a string indicating the file mode. We're using `'r'` here because we are reading the file. Other possible values include `'w'` for creating a new file to write and `'a'` for appending to an existing file.

If the first six characters of the line are `'ATOM '` then that line has interesting information. The numbers and names occupy fixed positions on a line, so you can extract them with indexing and convert them from strings to numbers, if necessary, in your reader.

Other hints

1. Don't modify data files! Your `readPDBfile` should work on any of the files.
2. Your reader returns many variables. To capture them all, you'll have to call it with a line something like
   ```
   a, an, rn, c = readPDBfile('7hvp.pdb')
   ```
3. To plot the 3D line drawings requested use the `Axes3D` object like
   ```
   fig = pylab.figure()
   ax = Axes3D(fig)
   ax.plot(x, y, z) # x, y, and z are 1D arrays giving the
   coordinates of the points
   ```
4. String methods `.upper()` and `.lower()` can change the case of strings.
5. Functions `int()` and `float()` can be used to change strings to numbers.
6. Compare strings with the usual comparison operators we used for numbers.
7. Break the problems down into small tasks. When making your loops, think of what has to be done once (like opening the file) and what has to be done repeatedly (like processing the line.)
8. I get 124 as the number of hydrogen bonding pairs for 7HVP.

Format of ATOM records

This is from the documentation of the PDB format on the rcsb web site.

COLUMNS	DATA TYPE	CONTENTS
1- 6	Record name	"ATOM "
7 - 11	Integer	Atom serial number.
13 - 16	Atom	Atom name.
17	Character	Alternate location indicator.
18 - 20	Residue name	Residue name.
22	Character	Chain identifier.
23 - 26	Integer	Residue sequence number.
27	AChar	Code for insertion of residues.
31 - 38	Real(8.3)	Orthogonal coordinates for X in Angstroms.
39 - 46	Real(8.3)	Orthogonal coordinates for Y in Angstroms.
47 - 54	Real(8.3)	Orthogonal coordinates for Z in Angstroms.
55 - 60	Real(6.2)	Occupancy.
61 - 66	Real(6.2)	Temperature factor (Default = 0.0).
73 - 76	LString(4)	Segment identifier, left-justified.
77 - 78	LString(2)	Element symbol, right-justified.
79 - 80	LString(2)	Charge on the atom.

Example:

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
ATOM      1  N   PRO A   1      -3.260   7.392  33.952  1.00 24.47           N
ATOM      2  CA  PRO A   1      -2.317   6.655  34.795  1.00 22.40           C
ATOM      3  C   PRO A   1      -0.919   6.658  34.208  1.00 20.82           C
ATOM      4  O   PRO A   1      -0.802   7.111  33.058  1.00 21.07           O
ATOM      5  CB  PRO A   1      -2.897   5.256  34.804  1.00 23.12           C
ATOM      6  CG  PRO A   1      -4.336   5.353  34.377  1.00 24.12           C
ATOM      7  CD  PRO A   1      -4.607   6.783  33.948  1.00 24.43           C
```