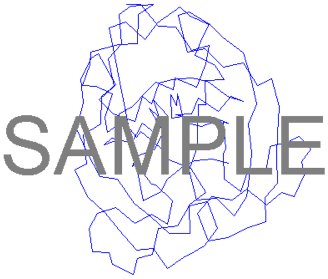
Assignment #6: File I/O and the protein data bank, due April 11th.

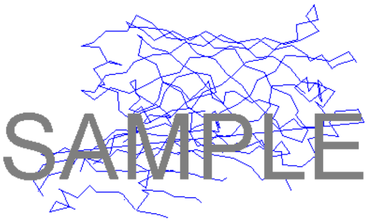
A protein is a molecule that is a sequence of amino acid residues.

The Protein Data Bank http://www.rcsb.org/pdb records the 3-d structures known for protein molecules.

Two examples are HIV Protease (7hvp) at right, an important AIDS drug target, and green fluorescent protein (1gfl) below, which earned its discoverers the Nobel Prize in 2008. Many differing 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 from these lines.



Their format is described on page 3 of this document.



Your Task:

1. Write a function **`readPDBfile('filename')`** that will read the atoms for a protein stored in a .pdb file whose name is specified in quotes. Your function will be stored in an m-file `readPDBfile.m`. The function declaration should be

```
function [anum, aname, coords] = readPDBfile(infile);
```

The output variables should be set to

`anum:` `n×1` column vector with the serial number for each atom as ***integers***
`aname:` `n×4` string array with the 4-letter atom *uppercase* name for each atom
`coords:` `n×3` matrix with xyz coordinates (in angstroms) for each atom as ***doubles***

where **`'n'`** is the number of atoms (or ATOM commands) in the .PDB file.

2. Write a function **`drawAtoms( queryName, marker, aname, coords )`** that uses the **`plot3`** command to draw (using the style in **`marker`**) any atoms with **`aname`** matching the **`queryName`**. All the other atoms should be ignored. For your report, you must draw all Carbon atoms (`' C  '`) with a **green asterisk marker**. *Note:* There is a preceding space and two trailing spaces in `' C  '`. For fun, you can use the 'Rotate 3D' plot tool to view the protein in 3D!

3. Write a function **`potentialHbonds( anum, aname, cords )`** that looks for all possible pairs of Nitrogen (second letter of **`aname`** is 'N') atoms and Oxygen (second letter of **`aname`** is **`'O'`**) atoms whose distance is between 2.6 and 3.2 angstroms, inclusive. Return a list

containing the pairs of atom numbers for hydrogen bonding pairs. *Note:* Be careful to compare the second letter in the name. For example, **`' NH1'`** should be a Nitrogen match!

4. Write a MATLAB primary script that gives a brief high level description of each function you wrote, generates 3D plots of the C atoms for both the **7hvp** and **1gfl** proteins, and outputs the number of hydrogen bonds found for each.

5. Create a .ZIP file of your 3 functions, your primary script file, and your published primary script with all your functions (use either ‘type fcnName’ within your mainscript or cut and paste them into a single document) and turn that in by 5pm on November 11th.

Opening and Reading a File:

A file such as **`‘7hvp.pdb’`** (for HIV protease) must be opened, and given a file id, before it can be read:

```
fid = fopen( '7hvp.pdb', 'rt' );
```

Then, each time you call

```
line = fgetl( fid );
```

the variable `'line'` will contain the next line read from the file as a string variable (array of characters) stored in the variable 'line'.

If the first six characters of the `'line'` are equal to **`'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.

Review the lectures on strings and/or on File I/O (or re-read chapters 6 & 8 from Attaway's book), if you are still need help on these concepts.

Other hints:

1. Don't modify data files! Your *readPDBfile* should work on any of the PDB files in the Protein Database…
2. To capture the many variables your reader function returns, you'll have to call it with a line something like this: **`[anum, aname, coords] = readPDBfile('7hvp.pdb');`**
3. **`upper()`** and **`lower()`** can change the case of strings; **`num2str()`** and **`str2double()`** can be used to change numbers to strings and strings to numbers.
4. Use string functions like **`strcmpi()`** or **`strmatch()`** to compare strings.
5. Break the problems down into small tasks. Think of what actions need to be done only once (like opening the file) and what actions have to be done repeatedly (like reading a line from the file.)

Format of ATOM records

This is from the documentation of the PDB format on the rcsb web site. I've marked in bold the items that are relevant for this assignment. MATLAB has a bioinformatics toolbox, that can read all the data from a PDB file, but for this assignment it will be much easier to write your own simple function than to try to figure out how MATLAB stores the PDB variables in structures and cells.

COLUMNS	DATA TYPE	CONTENTS
1 - 6	**Record name**	**'ATOM '** % Note the two trailing spaces
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 residue.
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 File Layout:

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
...
...
ATOM    145  N   VAL A  25      32.433  16.336  57.540  1.00 11.92      A1   N
ATOM    146  CA  VAL A  25      31.132  16.439  58.160  1.00 11.85      A1   C
ATOM    147  C   VAL A  25      30.447  15.105  58.363  1.00 12.34      A1   C
ATOM    148  O   VAL A  25      29.520  15.059  59.174  1.00 15.65      A1   O
ATOM    149  CB AVAL A  25      30.385  17.437  57.230  0.28 13.88      A1   C
ATOM    150  CB BVAL A  25      30.166  17.399  57.373  0.72 15.41      A1   C
ATOM    151  CG1AVAL A  25      28.870  17.401  57.336  0.28 12.64      A1   C
ATOM    152  CG1BVAL A  25      30.805  18.788  57.449  0.72 15.11      A1   C
ATOM    153  CG2AVAL A  25      30.835  18.826  57.661  0.28 13.58      A1   C
ATOM    154  CG2BVAL A  25      29.909  16.996  55.922  0.72 13.25      A1   C
...
...
```