Bio322 Assignment 2: Calclulate the R_G from PDB data and compare to experimental measures

Chaitanya A. Athale

October 20, 2014

1 Before you begin

Software: Download and install Octave on the computer you are using. Octave is a high-level language (like Python) focussed on numerical computing. Avoid the use of spreadsheet programs like MS Excel due to problems with numerical accuracy [1]. Additionally MATLAB/PyLab/Octave are fully programmable.

The tutorial in class should help you with the basics. For more questions refer to the EXTENSIVE help provided together with Octave and online.

Installation quick links:

- for MAC OSX please refer to http://hpc.sourceforge.net/
- for use Octave ver. 3.2.4 (refer to http://wiki.octave.org/Octave_for_Windows). Download and install the Setup file from http://wiki.octave.org/Octave_for_Windows#Octave_3.2.4_for_Windows_MinGW32.
- for Linux (ubuntu or related flavors) ensure the OS is updated and install the Octave package using the local software package installer

2 Data

The protein database (PDB) coordinate files are obtained from http://www.rcsb.org/pdb by searching. The files you need to get:

- GFP (1GFL)
- *E. coli* lactose inhibitor (lacI) protein (DOI:10.2210/pdb2xrs/pdb)
- Dengue non-structural protein NS1 (DOI: 10.1126/science.1247749)

3 Calculating the radius of gyration of each molecule

Write an Octave script to calculate the radius of gyration. You will use the script read_pdb.m. This stores the variables in a structure when called. Referring to the structure elements as follows will give you xyz (3D) coordinates of the atoms:

```
octave:2> s  = read_pdb('1GFL.pdb')
octave:2> s.acoord
ans =

   3.7642e+01   4.5936e+01   6.0270e+00
   3.8888e+01   4.6634e+01   6.4710e+00
   3.8955e+01   4.6782e+01   7.9960e+00
   4.0033e+01   4.6670e+01   8.5830e+00
   ...
```

Write a small script to automatically calculate the radius of gyration.

The first step will be to calculate the coordinate of the centre of mass (R_{CM}). This is nothing but the average position of all atoms and can be calculated (here vector notation is used for the 3-coordinate euclidean system):

$$\vec{R}_{CM} = \frac{1}{N}\Sigma_{i=1}^{N}\vec{R}_i \tag{1}$$

Use the expression discussed in class for the calculation of the radius of gyration (R_G):

$$\langle R_G^2 \rangle = \frac{1}{N}\Sigma_{i=1}^{N}\langle(\vec{R}_i - \vec{R}_{CM})^2\rangle \tag{2}$$

where R_i is the coordinate in space of the i^{th} atom and R_{CM} is the position of the centre of mass of the protein. Refer to the chapter on random walk polymers for more [2].

The simplest structures you will need for this are:

1. matrix: since we will be using 2D matrices only the thing you need to remember is that it has rows and columns. Use the row-index (1,...N) to store the atomic coordinates (i in Equation 1). For example, if I want to store roll numbers and ages I need to set a matrix as:

   ```
   >>myMat=[201001, 21.5; 201002, 22; 201005, 22.2; 201001, 23;
   201101, 21.2]
   ```

   If you press enter (return) at the end of that line, you should get the following output:

   ```
   ages =

      201001        24
      201002        22
      201005        23.2
   ```

```
201011          24.1
201112          22
```

As you might notice, the semi-colon (;) symbol sets a new row, while a comma (,) indicates a new column. Matrices that have multiple rows need to have the same number of columns per row. Now if you want to say access the age (in years) of the candidate in the 3rd row, all you need to do is call it as:

```
>> ages(3,2)
ans =  23.2
```

2. The second thing you need is to be able to loop (i.e. iteratively go through values in a matrix, and do something to them). The simplest loop in Octave is the *for* loop. So lets say we want to find the mean age from the example above, you can use simple array functions (thus avoiding loops altogether) as:

```
>>mean(myMat(:,2))
```

or you can explicitly go through each value:

```
>>avgAge=0;
>>for n=1:1:length(ages)
> avgAge = avgAge + myMat(n,2);
> end
```

To check the sum is actually the sum,

```
>> avgAge
avgAge =  115.30
```

Now all that remains is to divide by the number of entries (i.e. no. of rows):

```
>>avgAge = avgAge/length(ages)
```

The function length automatically chooses the size of the longer dimension (row, column). So sometimes to know what's happening, its better to use $r = size(ages, 1)$ where you mean r to be the number of rows.

4 Your submission

Submit a **SINGLE** document with the following:

1. a table with three columns: protein name, no. of atoms and radius of gyration
2. Include the script you wrote at the end of the report

It's ok to consult your friends how to do it, but do the task yourself, to find out if you can do it at all, and see if **you** have problems.

The subject of your email should read: "A02:RG-yourname" and the submitted file (PDF) should be (Pune) "A02-P-yyyymmdd-Yourname.pdf" or (Thiruvananthapuram) "A02-T-yyyymmdd-Yourname.pdf".
Submit the assignment by email to bio322@students.iiserpune.ac.in.

References

[1] Collins, C.J. (2014) Bugged by excel's calculation errors. Mar 2014 J. Accountancy

[2] Phillips, Kondev and Theriot (2008) Physical Biology of the Cell. Garland Science, New York. Illustr. by Orme.

[3] Teemu Ikonen tpikonen@pcu.helsinki.fi