PROTEIN REPORT

1. Introduction

The biological plasticity of proteins allows their sequences of amino acids to fold into different secondary structures depending on several factors. Some of them are the physicochemical properties of the amino acids - where the hydrophobic effect plays a major role (Chandler, 2005) -, the nature of the environment that surrounds them, or the electrostatics, van der Waals and hydrophobic interactions (Dill, 1990). Apart from these aspects, and as Anfinsen postulated, the own sequence of amino acids encodes the potential secondary structure of a protein. Yet, this statement seems to dilute when comes to consider the sequence role in protein folding through evolution. It has been demonstrated that sequences evolve faster than structures, mainly due to mutations (Illegard *et al.,* 2009), so it is possible for two different sequences to have the same secondary structure due to divergent evolutionary phenomena.

The goal of the present task is to examine the importance of the sequence similarity in protein folding at polypeptides level by analyzing how often two identical small sequences of amino acids reach the same secondary structure in comparison with random sequences. This variability will be studied through the RMSD (***R**oot **M**ean **S**quare **D**eviation*) measure between several pairs of pentapeptides. The lower this measure is after the superposition of two fragments, the more likely is that they reach a similar structure in a 3D space (Coutsias *et al.,* 2004).

2. Material and methods

2.1. *Data set*

The current study has been launched by analyzing one hundred of PDB (***P**rotein **D**ata **B**ank*) (Berman *et al.,* 2000) files that contain the information extracted from the crystallization of the protein, including its sequence of residues and the 3D atomic coordinates of each one of these residues. By performing a parsing of these files, it is possible to extract these atomic coordinates and perform the RMSD calculations over them.

2.2. *Python programming language and associated modules*

The main task has been effectively realized through an own-developed script in Python language that, making use of different modules, can be summed up in 5 steps:

2.2.1. Parsing of the PDB files

2.2.2. Extraction of fragments

2.2.3. Matches finding

2.2.4. RMSD computation

2.2.5. Histogram building

It is important to point out here the effectiveness of building a dictionary in this script to perform steps 2.2.2 and 2.2.3 simultaneously, as the fragments with the same sequence are associated to a same key, so the matching pentapeptides are easily found as well as the computational time of the operation is greatly reduced (about 10 seconds). After this crucial step, a double pipeline has been followed: the extraction of real pairs of pentapeptides, the computation of their RMSD and their representation in a first histogram, and the extraction of the equivalent number of first sequences of the dictionary (randomized), the computation of their RMSD and their plot in a second histogram.

The main module that has been used to complete this script is *Bio.PDB*, easy to handle to work with PDB files properly, by parsing them and extract both their sequences and atomic coordinates (Hamelryck & Manderick, 2003). The most relevant aspect of the module in the elaboration of the script is its ability to deal with the sequences of proteins as *Seq* Biopython-defined objects, with several methods and functions associated that allow to extract the sequences (*.get_sequence*) or check if the residues saved in the file can be treated as actual non-damaged amino acids (*.is_aa*)

It is also important to point out the utilization of the *Numpy* module, that behaves as an ideal tool to work with the atomic coordinates of the residues as matrices, and makes available a myriad of different functions and operators, key at the implementation of the RMSD calculation and the SVD (***S**ingle **V**alue **D**ecomposition*); as well as the *Matplotlib* module (Hunter, 2007), able to get the RMSD information and summarize it in form of different plots. In this assignment, two histograms with different graphical options have been constructed and used to extract the final conclusions.

2.3. *RMSD algorithm*

The RMSD (Kabsch, 1976) is an algorithm that can be used to measure the distance between two sets of vectors and their similarity. In the structural Bioinformatics context, it is used to compare the structure of biomolecules - in this specific case, pairs of pentapeptides - translating and rotating the first structure and superimposing it on the top of the other (Coutsias *et al.,* 2004). The two arrays of atomic coordinates are used to minimize the RMSD between them, and the structures will be more similar the lower is their RMSD measure:

$$\mathrm{RMSD}(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1} |x_i - y_i|^2}$$

Figure 1*: Main equation of the RMSD algorithm, where in this case X and Y are the two arrays of atomic coordinates of each pentapeptide*

3. Results

The main results obtained with the Python script implementation are in form of the following two histograms:

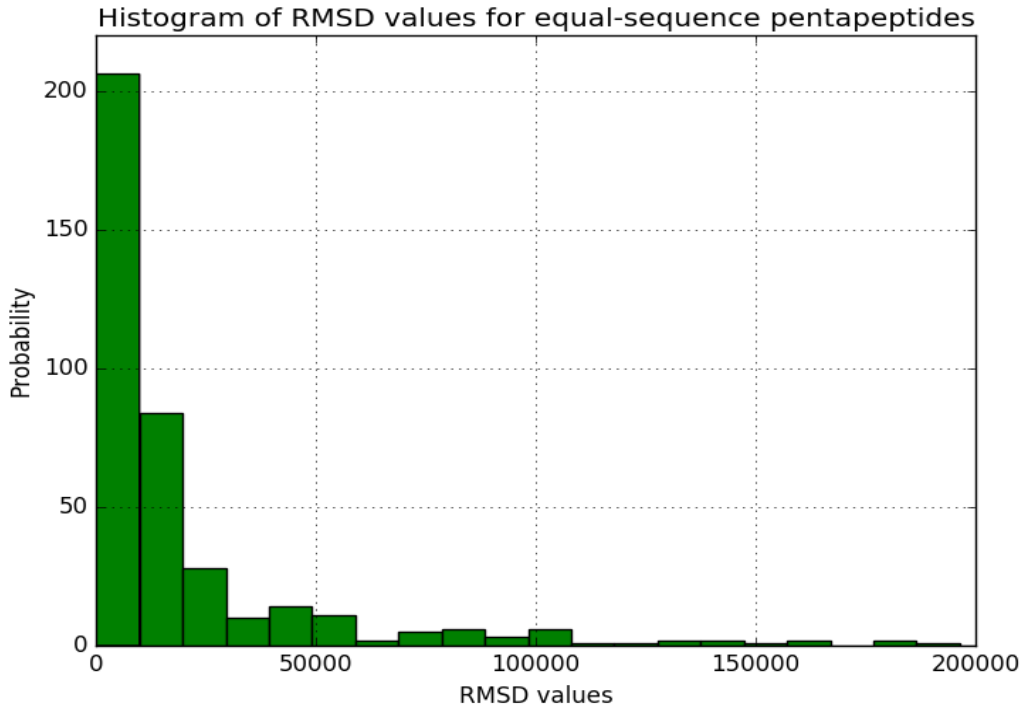

***Figure 2**: Histogram of RMSD values for 400 pentapeptide pairs with the same sequence. In the X - axis the RMSD value is represented, and in the Y - axis the frequency of each RMSD value is represented.*

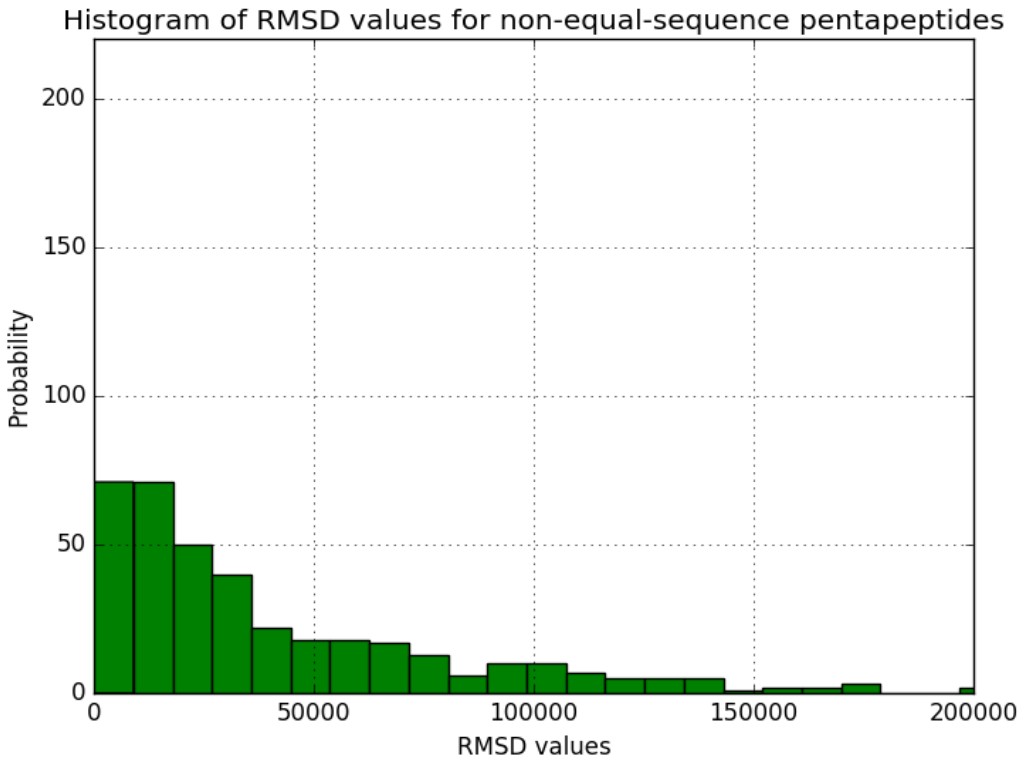

***Figure 3**: Histogram of RMSD values for 400 random pentapeptides pairs. In the X - axis the RMSD value is represented, and in the Y - axis the frequency of each RMSD value is represented. Obtained with Matplotlib module.*

In the first histogram, the RMSD values for the pairs of pentapeptides with the same sequence are concentrated between 0 and 20.000, where they form an acute peak. They appear in this region about 100 and 200 more times that in practically the rest of the histogram. On the contrary, in the second histogram, the RMSD values for the random pairs of pentapeptides are more spread and uniform between 0 and 100.000. However, they follow an exponential decaying distribution that indicates a tendency to keep a low value.

It is important to point out that with this script 398 matching pentapeptide pairs (with only two repetitions) were found. Matching triads or more have been discarded in order to keep the average of the RMSD values consistent. To elaborate the second histogram then, only 796 random pentapeptides have been selected.

4. Conclusions

With the data obtained it can be confirmed that **the RMSD values are much lower for equal-sequence pairs of pentapeptides** than for random-sequence pairs of pentapeptides, so it can be affirmed that the RMSD minimization is optimized for the pair of pentapeptides with equal sequence. This leads to the conclusion that they have a more similar 3D structure.

For this reason, it is possible to conclude that the pentapeptides with the same sequence biologically tend to fold into similar secondary structures with more probability than random peptides and **the importance of sequence similarity in protein at polypeptide level is confirmed**.

5. References

1. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E. (2000) The Protein Data Bank. *Nucleic Acids Research* **28**: 235-242

2. Chandler, D. (2005) Interfaces and the driving force of hydrophobic assembly. *Nature* **437**: 640-647

3. Coutsias, E.A., Seok, C., Dill, K.A (2004) Using quaternions to calculate RMSD. *J Comput Chem.* **25** (15): 1849–1857

4. Dill, K. (1990) Dominant forces in protein folding. *Perspectives in Biochemistry* **29**: 7133-7155

5. Hamelryck, T., Manderick, B. (2003) PDB parser and structure class implemented in Python. *Bioinformatics* **19**: 2308–2310

6. Hunter, J.D. (2007) Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* **9**: 90-95

7. Illegard, K., Ardell, D.H., Elofsson, A. (2009) Structure is three to ten times more conserved than sequence: a study of structural response in protein cores. *Proteins* **77** (3): 499-508

8. Jones E., Oliphant E., Peterson P. (2001) *SciPy: Open Source Scientific Tools for Python*

9. Kabsch, W. (1976) A solution of the best rotation to relate two sets of vectors *Acta Crystallographica* **32**: 922

10. Python Software Foundation. Python Language Reference, version 2.7. Available at http://www.python.org