

# BIOINFORMATICS. AN INTRODUCTION TO MATLAB PROGRAMMING

The course will take place in five classes. Every day a sheet of exercises will be given to be solved in the class. In each sheet there will be some mandatory exercises that must be sent to [jklett@cbm.uam.es](mailto:jklett@cbm.uam.es) and [acortes@cbm.uam.es](mailto:acortes@cbm.uam.es). These problems will be explained during the classes, and there will be enough time to complete them. Students are strongly encouraged to do all problems.

Please, before sending the exercises, note:

- Programs should be sent by e-mail in separate .m files with the original exercise name.
- Student name should be the first line of the program as a comment. Comments in the programs and text messages may be in English or Spanish.

Once the exercises are corrected, a second deadline will be available for fixes. These exercises will be evaluated as passed/non-passed. All exercises must be passed in order to continue with the bioinformatics course.

At the end of the classes period, all student must perform a project related with the theoretical part of the bioinformatics course. Details on project evaluation will be discussed with the project list.

## Day 4 problems

1. **ReadProtein.** Write a function whose input is the name of a file containing the positions of all atoms of a protein (these files are in proteins.zip). It must have two outputs: a matrix with size [N 3] (where N is the number of atoms) with the (x,y,z) coordinates of each atoms, and a vector with the charges of all atoms. Note: Once you have the coordinates of the atoms, you can use plot3 (with '.' markers) to visualize the protein. Hint: Use importdata for this problem.
2. **BoxSize.** Write a script that calculates the dimensions of a box that contains the protein (you will be provided of a coordinate's file of the protein atoms). You may want to use the Max and Min functions from previous exercises. Although, you are allowed in this case to use the "max" and "min" Matlab functions. (NB: Do not worry about rotating or translating the coordinates of the protein! Just calculate the dimensions of the box with the protein coordinates as they are given).
3. **Affinity.** The affinity between the receptor and a ligand depends (approximately) on the coulombic potential between them. In order to estimate this affinity, write a function or script that:
  - a. Reads the receptor file and one ligand file (use function ReadProtein for this).

- b. Calculates the sum of coulombic interaction of all the protein atoms against all the ligand atoms. The coulombic interaction between two atoms is the following:

$$\begin{array}{l} \text{Atom 1: position } (x_1, y_1, z_1), \text{ charge } Q_1 \\ \text{Atom 2: position } (x_2, y_2, z_2), \text{ charge } Q_2 \\ \\ d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}; \\ \text{Coul} = \frac{Q_1 Q_2}{d}; \end{array}$$

Note: The affinity depends on the relative positions of receptor and ligand. But you do not need to take care of this, because the files provided correspond to a ligand which is located in the active part of the receptor (where the affinity is maximal). You may see this by plotting together the receptor and the ligand (use plot3 and different colors for each one).

4. **Affinity\_full (mandatory).** Write a script that:
- Reads the receptor file
  - Reads, one by one, all the ligands (use a “for” loop, and construct the names of the files by appending text strings to the counter. Do not forget to use num2str where appropriate).
  - Computes the coulombic potential between each ligand and the receptor. Note: You may do this after each ligand is loaded, and then overwrite the next ligand.
  - Plots the coulombic potential for each ligand with the protein.
  - Shows a message saying which ligand is binding with the lowest coulombic interaction energy (this means highest affinity).
  - Plots the receptor and the ligand with lowest coulombic potential together (in different colors).
5. **WriteProtein.** Write a function that takes the coordinates, radii and charges of a set of atoms, and creates a .dat file, which may be read by ReadProtein. Do not forget to close the file at the end of the function. You will need the functions fopen, fclose and fprintf (or fwrite).