Introduction to Python
Structural Biology

Final Project

Programming in python a **standalone program** for solving a specific problem in the **Structural Biology**.

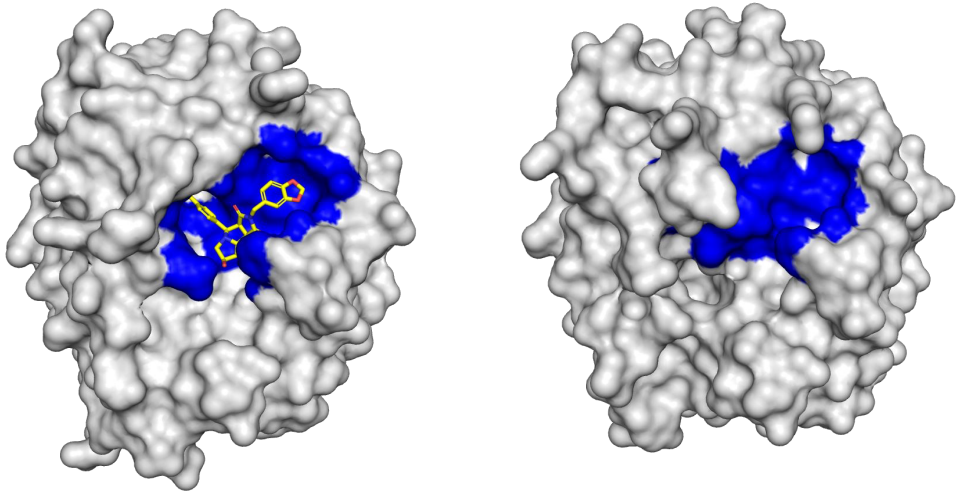
Independent evaluation for both subjects

upf.

Objective

- **Objective:** Implement a **ligand binding site** predictor for proteins.

A specific region on a protein where a small molecule (ligand) binds through non-covalent interactions.



Thrombin with Inhibitor

Objective

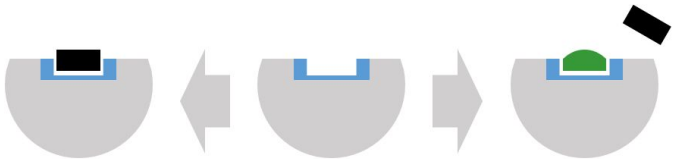
- **Objective:** Implement a **ligand binding site** predictor for proteins.
- **Input:** Protein structure in **pdb** format

  Structure-based approach.
- **Output:** Predicted ligand binding sites.
  - Lists of amino acids involved in each detected site.
  - File that can be visualized in a molecular graphics software (Chimera, PyMOL) .

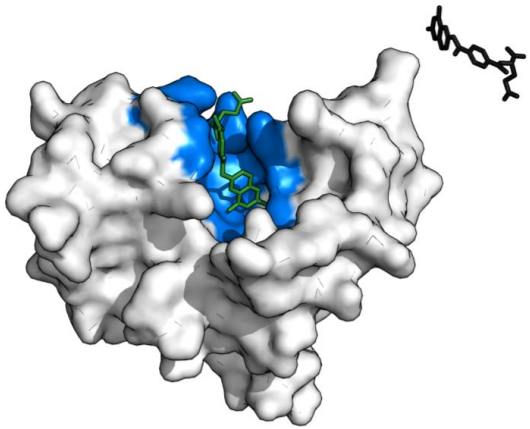
upf.

First approach to the project

Different approaches based on:



- **Geometry**
- **Machine Learning**
- **Deep Learning**
- **Other**



Start with an initial bibliography research and choose **one method** to implement.

After PYT Session 11 different objectives will be proposed to help in the development of the program.

PYT Evaluation

1. **Final project (30%). Groups. (*)**

   The exercise will be guided in terms of the program structure.

   PYT evaluation will take into account:

   - That the code works and does what is expected . (40%)
   - Program structure (functions, classes, modules...) (20%)
   - Readability and documentation. (20%)
   - Reusability, use of libraries. (10%)
   - Efficiency of the code (10%)

upf.

SBI evaluation

Groups of up to **2-3 students** do the work project. It consists on programming a server or **standalone program** on the analysis of the structure of biomolecules.

The project is evaluated in line with the course of Python. This implies the following requisites:

- The team has to have at least one member registered in the course of “structural bioinformatics” and one in “Introduction to Python”

SBI evaluation

The evaluation of the project is independent, but a percentage of the mark will be from the mark in python (quality of the program). The marks will be based on:

a) Quality:	0-10 pts (from python's mark)
b) Executable:	0/1 (works/doesn't)
c) Installation	0/1 (installs/doesn't)
d) Tutorial	0-10 pts (Tutorial with examples on how to use the program)
e) Theory	0-10 pts (Background and scientific explanation)
f) Analysis	0-10 pts (Analyses of examples)
g) Applicability	0-1 (can the program be applied to all/many/few examples)

Mark= (3a + 2d + 4e + 3f +10b + 10c + 10g)/15

upf.