LIGAND BINDING SITE PREDICTOR - TUTORIALS AND EXAMPLE ANALYSES

TUTORIALS

To work with this model, it is necessary to download all the files provided:

- setup.py
- __init__.py
- BS_predictor.py
- data directory, which contains:
    - classes.py
    - atom_types.csv
    - train_pdbs directory, which contains 300 PDB files

Moreover, note that it is also necessary to install DSSP (Define Secondary Structure of Proteins) for the program to work properly. In Linux this can be done using the following commands on the terminal:

```
sudo apt-get install dssp
sudo ln -s /usr/bin/mkdssp /usr/bin/dssp
```

There are two possible ways to run the program:

- **Running the BS_predictor.py Python script:**

  The BS_predictor.py Python script is used to run the program. Simply type “python3 BS_predictor.py” to launch the program, and then specify the **options -p** followed by the name of a valid PDB file, and **-o** followed by the name of an output file to save the result, as shown below:

```
python3 BS_predictor.py -p <input_PDB_file> -o <output_PDB_file_name>
```

  When using this way of running BS_predictor, it is also necessary to have the following packages and versions downloaded:

  - pandas == 2.0.0
  - biopython == 1.81
  - networkx == 3.1
  - scipy == 1.10.1
  - sklearn (scikit-learn) == 1.2.2
  - numpy == 1.24.2

- **Installing the program BS_predictor:**

  1. Run setup.py in the command line:

```
python setup.py install
```

     This installs the program BS_predictor.py, making the command BS_predictor available in the command line.

  2. Run BS_predictor in the command line using the following syntax:

```
BS_predictor -p <input_PDB_file> -o <output_PDB_file_name>
```

***Attention** : we do not recommend the option of installing the program due to in our case after installing it the BS_predictor.py does not recognize the data folder.

ANALYSES OF EXAMPLES

We have used the program BS_predictor.py to obtain the predicted ligand binding sites of the following proteins:

- 4ins.pdb

This structure corresponds to 2ZN pig insulin, which is a dimer.

```
python3 BS_predictor.py -p ./4ins.pdb -o ./4ins_result.pdb
```

```
Computing exposure information of protein 4ins...
Computing residue properties of protein 4ins...
Computing interactions of protein 4ins...
Predicting the binding sites of protein 4ins...
The list of predicted binding sites of protein 4ins is the following:
['4ins_A_GLY_1', '4ins_A_GLN_5', '4ins_B_GLY_23', '4ins_C_GLY_1', '4ins_D_GLY_23']
A new pdb file has been generated with the predicted binding sites of protein 4ins.
Program finished correctly.
```

The program predicts the following 4 residues as possible binding sites: GLY 1 (chain A), GLN 5 (chain A), GLY 23 (chain B), GLY 1 (chain C), GLY 23 (chain D).

Using Chimera, we can visualize the structure of the protein (4ins.pdb file) in blue, and the predicted binding sites according to BS_predictor (4ins_result.pdb file) in yellow.

- 4hg0.pdb

This structure corresponds to Magnesium and cobalt efflux protein CorC of *Escherichia coli*. It corresponds to target T0652 from Critical Assessment of Structure Prediction (CASP) and we have used it to compare the results we obtained with BS_predictor and the binding residues from CASP.

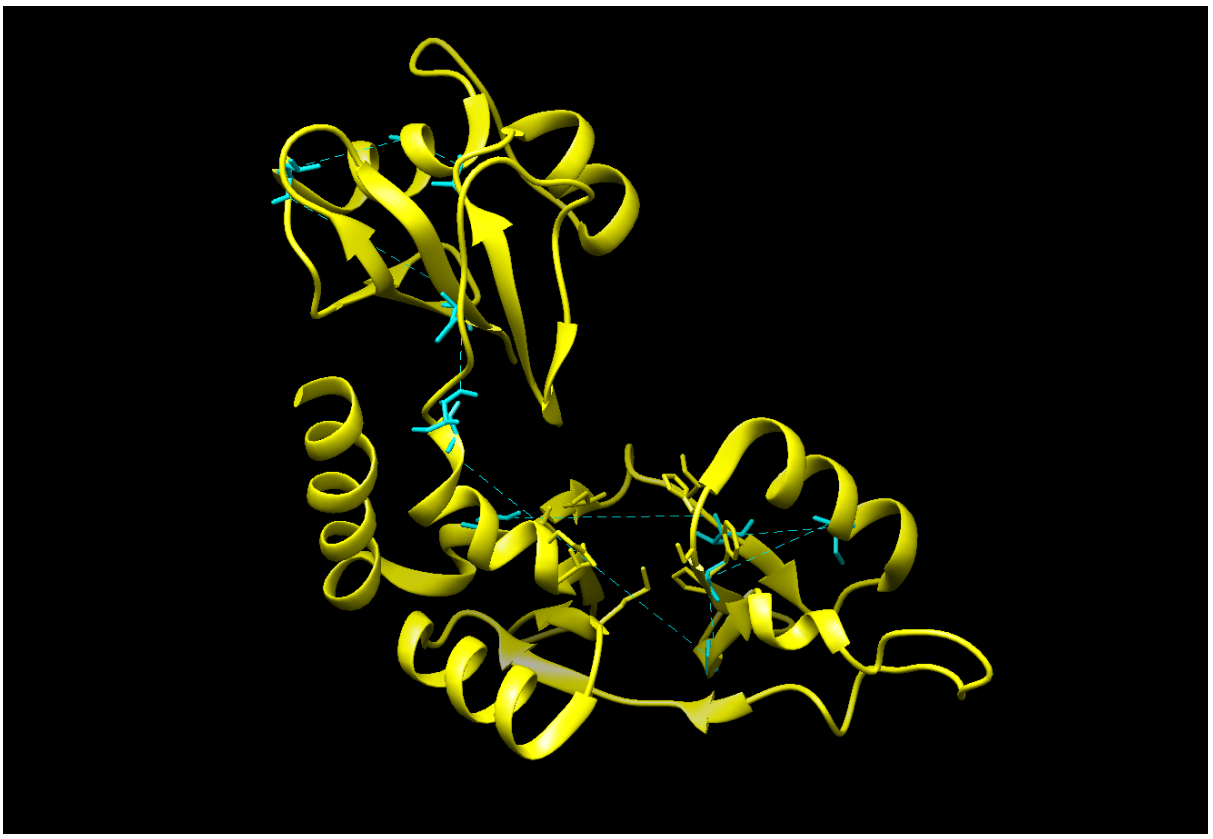
```
python3 BS_predictor.py -p ./4hg0.pdb -o ./4hg0_result.pdb
```

```
Computing exposure information of protein 4hg0...
Computing residue properties of protein 4hg0...
Computing interactions of protein 4hg0...
Predicting the binding sites of protein 4hg0...
The list of predicted binding sites of protein 4hg0 is the following:
['4hg0_A_ILE_72', '4hg0_A_THR_81', '4hg0_A_CYS_91', '4hg0_A_PHE_103', '4hg0_A_GLY_115', '4hg0_A_GLY_190', '4hg0_A_ILE_192'
 '4hg0_A_ALA_216', '4hg0_A_GLY_241', '4hg0_A_LEU_250', '4hg0_A_GLY_261', '4hg0_A_HIS_278']
A new pdb file has been generated with the predicted binding sites of protein 4hg0.
Program finished correctly.
```

The program predicts the following 12 residues as possible binding sites: ILE 72 (chain A), THR 81 (chain A), CYS 91 (chain A), PHE 103 (chain A), GLY 115 (chain A), GLY 190 (chain A), ILE 192 (chain A), ALA 216 (chain A), GLY 241 (chain A), LEU 250 (chain A), GLY 261 (chain A), HIS 278 (chain A).

We know that the predicted binding sites according to CASP are 74, 79, 80, 99, 100, 101, 102, 103, 104, 165, 180, 182 and 183. We can see that most of these are different from the ones that our program predicts, which means that our model does not work optimally in this case.

Using Chimera, we can visualize the structure of the protein (4hg0.pdb file) in yellow, the predicted binding sites according to BS_predictor (4hg0_result.pdb file) in blue and the ligand in gray (in the case of the first image).



- 4hqo.pdb

This structure corresponds to thrombospondin repeat anonymous protein (TRAP) from *Plasmodium vivax*. In this case, it corresponds to target T0686 from Critical Assessment of Structure Prediction (CASP) and we have used it to compare the results we obtained with BS_predictor and the binding residues from CASP.

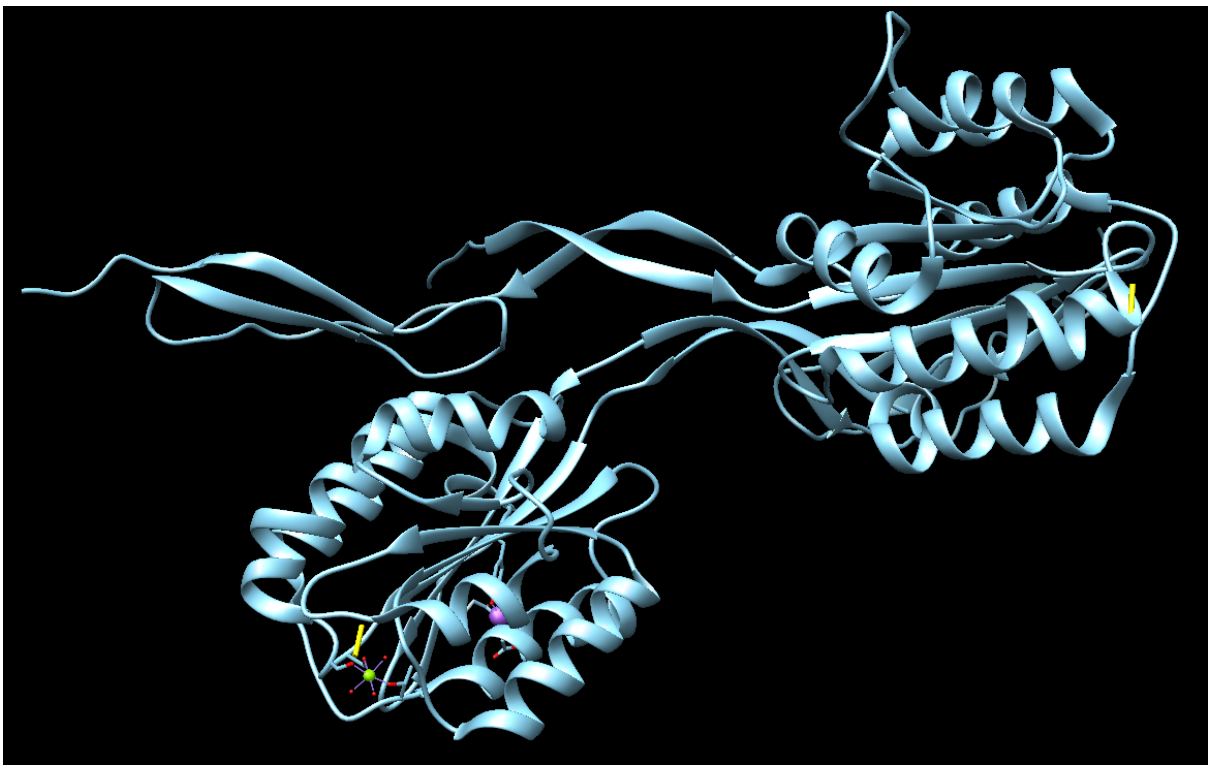
```
python3 BS_predictor.py -p ./4hqo.pdb -o ./4hqo_result.pdb
```

```
Computing exposure information of protein 4hqo...
Computing residue properties of protein 4hqo...
Computing interactions of protein 4hqo...
Predicting the binding sites of protein 4hqo...
The list of predicted binding sites of protein 4hqo is the following:
['4hqo_A_GLY_51', '4hqo_A_SER_52', '4hqo_A_GLY_56', '4hqo_A_ASN_83', '4hqo_A_SER_91', '4hqo_A_ILE_105', '4hqo_A_VAL_114', '
4hqo_A_THR_130', '4hqo_A_GLY_159', '4hqo_A_ALA_167', '4hqo_A_ALA_171', '4hqo_A_GLY_185', '4hqo_A_MET_273', '4hqo_B_GLY_51',
 '4hqo_B_GLY_56', '4hqo_B_TYR_57', '4hqo_B_VAL_64', '4hqo_B_GLY_100', '4hqo_B_LYS_107', '4hqo_B_VAL_114', '4hqo_B_GLY_125',
 '4hqo_B_GLY_159', '4hqo_B_GLY_187', '4hqo_B_ILE_227']
A new pdb file has been generated with the predicted binding sites of protein 4hqo.
Program finished correctly.
```

The program predicts the following 24 residues as possible binding sites: GLY 51 (chain A), SER 52 (chain A), GLY 56 (chain A), ASN 83 (chain A), SER 91 (chain A), ILE 105 (chain A), VAL 114 (chain A), THR 130 (chain A), GLY 159 (chain A), ALA 167 (chain A), ALA 171 (chain A), GLY 185 (chain A), MET 273 (chain A), GLY 51 (chain B), GLY 56 (chain B), TYR 57 (chain B), VAL 64 (chain B), GLY 100 (chain B), LYS 107 (chain B), VAL 114 (chain B), GLY 125 (chain B), GLY 159 (chain B), GLY 187 (chain B), ILE 227 (chain B).

We know that the predicted binding sites according to CASP are 28, 30 and 103. From this example we can see that our program is getting many false positives, which could be due to the fact that we have given much more weight to the class of binding sites than to the one of non-binding sites during the development of the machine learning model. Therefore, we should try to improve our program in order to obtain more accurate results.

Using Chimera, we can visualize the structure of the protein (4hqo.pdb file) in blue, and the predicted binding sites according to BS_predictor (4hqo_result.pdb file) in yellow.



- 4fgm.pdb

This structure corresponds to the aminopeptidase N family protein Q5QTY1 from *Idiomarina loihiensis*. In this case, it corresponds to target T0726 from Critical Assessment of Structure Prediction (CASP).

```
python3 BS_predictor.py -p ./4fgm.pdb -o ./4fgm_result.pdb
```

```
Computing exposure information of protein 4fgm...
Computing residue properties of protein 4fgm...
Computing interactions of protein 4fgm...
Predicting the binding sites of protein 4fgm...
The list of predicted binding sites of protein 4fgm is the following:
['4fgm_A_ILE_22', '4fgm_A_LEU_31', '4fgm_A_SER_40', '4fgm_A_GLY_52', '4fgm_A_LEU_53', '4fgm_A_ASN_58', '4fgm_A_GLN_66', '4fgm_A_GLY_104', '4
fgm_A_SER_110', '4fgm_A_CYS_112', '4fgm_A_LEU_113', '4fgm_A_ALA_133', '4fgm_A_SER_151', '4fgm_A_GLY_171', '4fgm_A_SER_190', '4fgm_A_GLU_217'
, '4fgm_A_SER_242', '4fgm_A_CYS_247', '4fgm_A_SER_248', '4fgm_A_ILE_253', '4fgm_A_LEU_270', '4fgm_A_SER_278', '4fgm_A_PHE_313', '4fgm_A_TYR_
316', '4fgm_A_SER_338', '4fgm_A_GLY_345', '4fgm_A_SER_350', '4fgm_A_SER_354', '4fgm_A_THR_360', '4fgm_A_PRO_370', '4fgm_A_GLY_380', '4fgm_A_
LEU_382', '4fgm_A_SER_386', '4fgm_A_ALA_402', '4fgm_A_SER_416', '4fgm_A_GLY_418', '4fgm_A_ASN_426', '4fgm_A_ASN_429', '4fgm_A_SER_437', '4fg
m_A_GLY_488', '4fgm_A_ALA_493', '4fgm_A_GLY_497', '4fgm_A_LEU_498', '4fgm_A_ASN_502', '4fgm_A_ILE_523', '4fgm_A_SER_532', '4fgm_A_GLY_583',
'4fgm_A_THR_586']
A new pdb file has been generated with the predicted binding sites of protein 4fgm.
Program finished correctly.
```

The program predicts the following 48 residues as possible binding sites: ILE 22, LEU 31, SER 40, GLY 52, LEU 53, ASN 58, GLN 66, GLY 104, SER 110, CYS 112, LEU 113, ALA 133, SER 151, GLY 171, SER 190, GLU 217, SER 242, CYS 247, SER 248, ILE 253, LEU 270, SER 278, PHE 313, TYR 316, SER 338, GLY 345, SER 350, SER 354, THR 360, PRO 370, GLY 380, LEU 382, SER 386, ALA 402, SER 416, GLY 418, ASN 426, ASN 429, SER 437, GLY 488, ALA 493, GLY 497, LEU 498, ASN 502, ILE 523, SER 532, GLY 583 and THR 586 (all of them from chain A).

We know that the predicted binding sites according to CASP are 273, 277 and 307. In this case, again, we can see that our program is getting many false positives. Thus, it is necessary to improve the machine learning model to avoid this situation.

Using Chimera, we can visualize the structure of the protein (4fgm.pdb file) in blue, and the predicted binding sites according to BS_predictor (4fgm_result.pdb file) in yellow.

- 1ags.pdb

This structure corresponds to a surface mutant (G82R) of a human alpha-glutathione S-transferase. We have chosen this structure because it is an example of a PDB file that only contains information about alpha Carbons and, therefore, the program will raise a ValueError since it is not meant to work with this type of PDB files.

```
python3 BS_predictor.py -p ./1ags.pdb -o ./1ags_result.pdb
```

```
Traceback (most recent call last):
  File "/home/yolanda/Documentos/casi_def_2/BS_predictor.py", line 95, in <module>
    main(sys.argv[1:])
  File "/home/yolanda/Documentos/casi_def_2/BS_predictor.py", line 44, in main
    target_protein = Input_Protein(protein_file)
                     ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/yolanda/Documentos/casi_def_2/data/classes.py", line 105, in __init__
    self.init_matrix()
  File "/home/yolanda/Documentos/casi_def_2/data/classes.py", line 125, in init_matrix
    raise ValueError("PDB file only contains information about C-alpha atoms.")
ValueError: PDB file only contains information about C-alpha atoms.
```