Arnau Noguera Segura
Aina Vaquer Picó

SITE SEEKER

PREDICTION OF PROTEIN BINDING SITES

1. THEORETICAL BACKGROUND AND MODEL DESIGN

SiteSeeker is a machine learning-based program that aims for the prediction of protein-ligand binding sites based on protein structure. To design this program, we started selecting the features that we wanted to implement in our machine learning model. These were based on features described in previous literature for similar applications.

Some features were based on inherent amino acid properties, such as hydrophobicity (using both the Kyte-Doolittle scale and the Rose hydrophobicity scale, the latter being the average area of buried amino acids in globular proteins), charge, polarizability, molar mass, volume normalized by Van der Waals volume, isoelectric point, and steric parameter, which measures bulkiness of the side chain of the residues. Such features were used or described in studies such as McGreig et al. (2022), and Meiler et al. (2001).

At the same time, we used a different set of features representative of the actual conformation of each particular protein. These are characteristics of each residue that can be obtained by running DSSP on the structure of the protein, available in Python through the Bio.PDB.DSSP module. The structural features that we used for each residue are: the secondary structure, ASA (Accessible Surface Area), phi and psi angles, the energies of the interactions involving atoms of the main chain, neighbor density (implemented by ourselves as the fraction of neighboring residues per volume unit), and the ASA Z-score in respect to its neighbors. This last parameter was ideated to represent the concavity or convexity of the protein at a certain position (in the sense that a residue having more ASA than its neighbors, i.e., being more exposed to the surface, might represent convexity in the surface structure, and vice versa). Although this feature might not necessarily always represent this characteristic of the protein, particularly for inner residues, it still might be useful for the prediction. Neighbors were chosen arbitrarily as those located at a minimum distance lower than 10 Å than our residue of reference. Some of these features were used in publications like Yamaguchi et al. (2022), while others were added because they were also calculated by DSSP, and we believed they may be informative.

To train our model, we wanted to obtain a list of PDB files containing a protein and its ligand. To do that, a non-redundant list of ligands was obtained from RCSB, mainly ligands that were popular drugs, ring systems or metal-containing. The idea was to obtain protein-ligand interactions that reflected very different types of molecules, and were not overrepresentative of any single ligand type. Then, we searched for proteins that bound those ligands in RCSB through the Python package rcsbsearchapi and, for each ligand, we downloaded the PDB structure of only one protein, to avoid overfitting. Then, we calculated the features listed above for each of the residues of every protein, and we determined if the residues belonged to a binding site, to generate our training dataset. A residue was labeled as belonging to a binding site if its minimum distance to the ligand was lower than 4 Å, according to what was found in Khazanov et al. (2013).

130 proteins in PDB format were downloaded, and, after removing some proteins that were problematic for DSSP, we obtained the features and the putative binding sites of 111 proteins (72,501 residues, 5,253 of which were binding residues according to the described criteria). We divided the proteins into training and test datasets (80% and 20%, respectively). We opted to divide entire proteins instead of single residues to avoid any potential overfitting due to different residues of overlapping proteins being used both in the training and testing.

Categorical features (amino acid residue and secondary structure) were encoded as binaries to allow for incorporation into the machine learning model. The data were normalized previously to the model training because most machine learning models are sensitive to variations in feature scales. The scaling parameters were saved to be applied on the features of input proteins in the program.

We chose to use an SVM (Simple Vector Machine) to build our machine learning model, which has been used in previous articles to predict protein-ligand binding sites (Wang et al., 2013; Wong et al., 2013). The parameters of the model: C, kernel, degree, and gamma; were optimized via hyperparameter tuning to obtain the best performing model. These parameters were evaluated by iterative splitting of the training set into training and evaluation, training of an SVM model with each combination of parameters, and calculation of the accuracy, precision and recall of the predictions on the evaluation set. We chose the following parameters: C = 1, kernel = “rbf”, gamma = 1, for the final model. Degree is a specific parameter of the “poly” kernel, so it is not applicable here.

In order to train and to test the model, to avoid overfitting due to having an excess of non-binding residues or proteins with a different number of residues, residues were weighted by class (binding/non-binding) and protein, so all training proteins contributed equally to the model.

Finally, we tested the model on the test set obtaining the following results: accuracy = 0,87, precision = 0.26, recall = 0.23. We are aware that improvements in precision and recall should be made. However, we could not test more restrictive parameters, such as higher C and gamma values, due to computational limitations.

The next was creating the program itself. It takes a PDB file as an input, calculates the features of the protein, encoding and scaling them afterwards, and then predicts the binding residues. The binding residues are grouped into binding sites based on distance, which can be set by the user. We use a default maximum distance of 4 Å between a residue and any residue in a binding site for that residue to be part of it.

The results are printed in a text file containing the binding sites and the residues that conform them, as well as a PDB file that contains only the residues predicted as binding residues, which should be viewed in Chimera alongside the original PDB file to visualize the results.

2. TUTORIAL

SiteSeeker can be installed as a Python package. The following command installs the package and all of its dependencies:

```
$ pip install SiteSeeker-1.0.tar.gz
```

However, a limitation to be considered is that SiteSeeker requires DSSP to be installed in the system it is run in. It cannot be automatically installed like the Python dependencies, so users should make sure DSSP is available in their systems. The following commands can be used to install DSSP and create a link to it called dssp:

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

Upon installation, the package can be executed through the command line using the automatically generated alias SiteSeeker. It can also be imported as a Python module.

2.1. Command line.

SiteSeeker can be executed as a system utility using the following command:

```
SiteSeeker [-h] --input INPUT [--output OUTPUT] [--verbose] [--maxdist
MAXDIST] [--knownbind]
```

Flag	Meaning
-h, --help	Show the help message and exit
--input INPUT, -i INPUT*	Input file in PDB format (with extension .pdb)
--output OUTPUT, -o OUTPUT	Desired path and base filename of the output files (without the file extension)
--verbose, -v	Prints additional information during the execution
--maxdist MAXDIST, -d MAXDIST	Maximum distance between the closest residues in two binding sites for these to be considered separate binding sites (default: 4 Å)
--knownbind, -k	This can be used if the input PDB file contains the interaction with the ligand (with the flag HETATM) and the user would like to know the accuracy of SiteSeeker's prediction

*Features marked with * are compulsory.*

The program gives two outputs: a text file containing the binding sites and the residues of each binding site, and a PDB file with the coordinates of the binding residues that can be used to visualize the results in Chimera. We advise users to open the output PDB file and the PDB file of the protein of interest to visualize the results more clearly. If the name of the output files is not specified, they will be named pdb_code_SiteSeeker.txt and pdb_code_SiteSeeker.pdb by default, where pdb_code is the PDB code of the input protein.

It is worth noting that the installable tar file SiteSeeker-1.0.tar.gz, as well as the bin directory where the package is installed, will contain two main folders: SiteSeeker/ and Training/. SiteSeeke/ contains all the necessary modules and files to run the binding site prediction, and also this report. However, Training/ and its contents are not used at any point for the program to run, but rather contain the Python codes and the files that were used for the training of the model.

2.2. Python script.

Once installed, SiteSeeker can also be imported as a module and executed from a Python script. The code below shows how to do it:

```
import SiteSeeker.PredictBindingSites as SiteSeeker

# Main function of the SiteSeeker package: prediction of binding sites
SiteSeeker.SiteSeeker_predict('pdb_code.pdb', known_binding = False,
verbose = False, output = 'output_name', max_distance = 4)
```

The function SiteSeeker_predict performs the same analysis that is executed when using the command line version of the program, and the only compulsory parameter is the PDB file.

Moreover, more modules of the package can be imported in Python, such as SiteSeeker.residue_functions and SiteSeeker.aminoacid_info, both of which are used in the prediction of binding sites and contain, respectively, functions used for the extraction of features from a PDB file and dictionaries containing the aforementioned physicochemical properties of the different amino acid residues.

2.3. Example.

Executing SiteSeeker to predict the binding sites of 2XIR:

```
$ SiteSeeker -i pdb_files/2XIR.pdb -o results/2XIR -v
Reading info from  pdb_files/2XIR.pdb
Extracting features from input file...
Predicting binding sites...
Obtaining results...
6 binding sites were predicted. Results can be found in
```

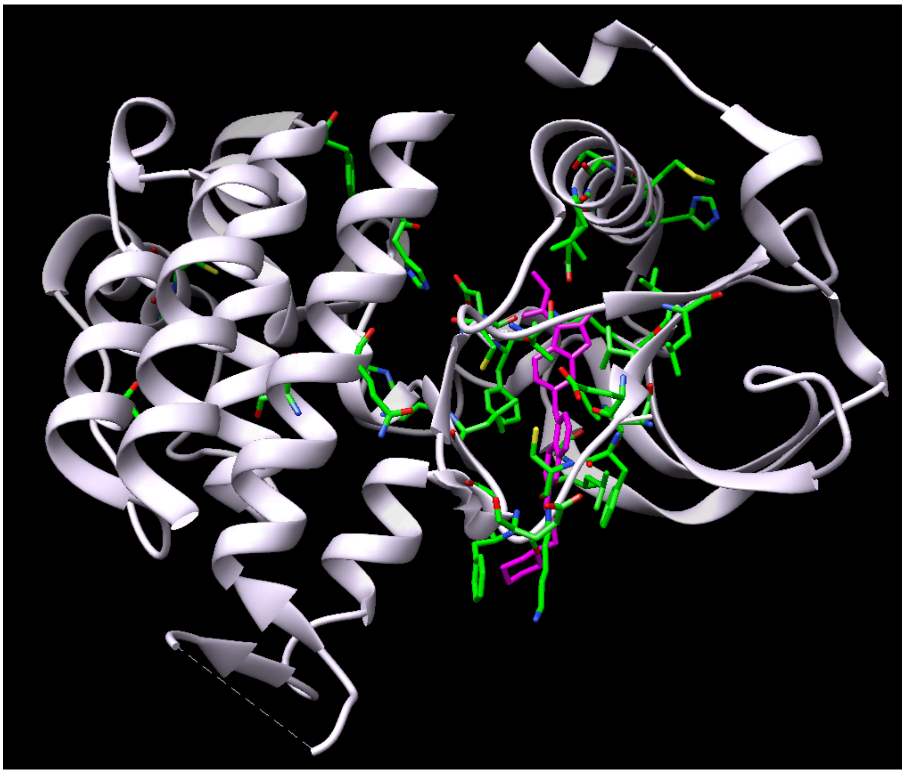
```
results/2XIR.txt, and the binding residues can be visualized with the
file results/2XIR.pdb using Chimera.
```

Two result files are generated: results/2XIR.txt and results/2XIR.pdb. The txt file contains a summary of each predicted binding site and the residues that form it.

To visualize the results, the output PDB file should be opened in Chimera along with the original PDB:

```
$ chimera results/2XIR.pdb pdb_files/2XIR.pdb
```

We recommend the user to change the colors according their preference. The following figure is an example of what can be observed:



In pink is the ligand of the protein (which, in this case is known; when using the program to predict the binding sites of a protein without its ligand's presence, it would not be there). In white is the whole 2XIR protein, while the residues in green are the predicted binding residues (the results/2XIR.pdb file).

Now we will do the same, but assuming that we have information about the ligand of the protein in our pdb file (as HETATM atoms). In this case, the prediction is also performed, but it can be evaluated as well.

```
$ SiteSeeker -i pdb_files/2XIR.pdb -o results/2XIR -vk
Reading info from  pdb_files/2XIR.pdb
```

```
Extracting features from input file...
Predicting binding sites...
Obtaining results...
6 binding sites were predicted. Results can be found in
results/2XIR.txt, and the binding residues can be visualized with the
file results/2XIR.pdb using Chimera.
Accuracy: 0.9527
Precision: 0.5758
Recall: 1.0
```

3. ANALYSIS OF EXAMPLES

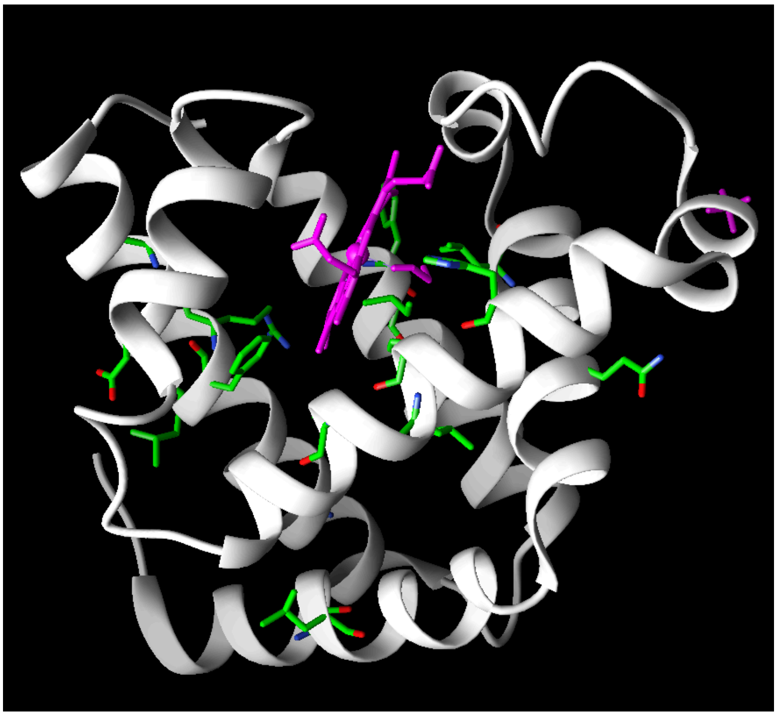
We selected 5 proteins that were in no way involved in the training or evaluation process of SiteSeeker to analyze as examples. Still, they are proteins with known binding of a ligand, in order to test these results.

3.1. 106M

For this protein, a total of 5 binding sites were predicted. The accuracy, precision and recall of the protein were 0.82, 0.31, and 0.23 respectively. If we open the protein in Chimera, we can see that it predicts part of real the binding site. However, there are a series of false positives that need to be taken into account.

In this table, we can see all the binding sites and residues. In dark blue, we have marked the one that we believe belongs to the binding site.

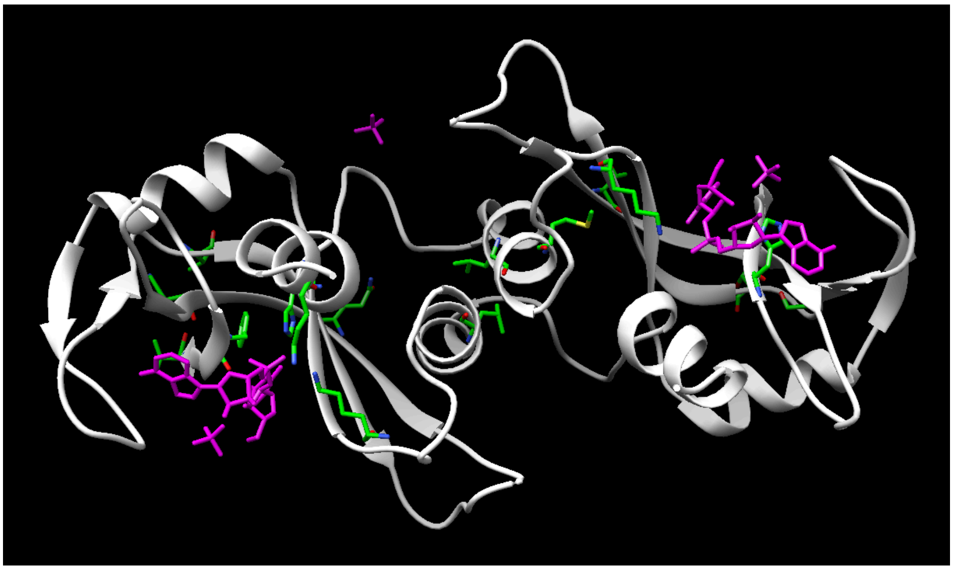
Binding site 1	Binding site 2	Binding site 3	Binding site 4	Binding site 5
A LEU 9 A LEU 11	A GLN 26	A LEU 32 A HIS 64 A PHE 68 A LEU 69 A LEU 72 A PHE 106 A ILE 107	A GLY 129	A LEU 137 A PHE 138 A ARG 139 A ASP 141 A ALA 143



3.2. 11BA

For this protein, a total of 10 binding sites were predicted. The accuracy, precision and recall of the protein were 0.84, 0.38, and 0.17 respectively. In Chimera, we can see that the number of false positives is slightly lower than the previous case, but still considerable. The binding sites that we consider that are good predictions are marked in purple and green. However, we think that binding site 2 and binding site 10 should be joined together, as well as binding site 5 and binding site 6. The problem is that, for these actual binding sites, many residues were not predicted as binding.

Binding site 1	Binding site 2	Binding site 3	Binding site 4	Binding site 5
A LEU 28 B LEU 28 B MET 30 B LYS 41	A LYS 41	A THR 78 A HIS 105 A VAL 124	A GLN 101	A PHE 120 B HIS 12
Binding site 6	Binding site 7	Binding site 8	Binding site 9	Binding site 10
B LYS 7	B SER 75	B THR 78	B THR 100	B HIS 119

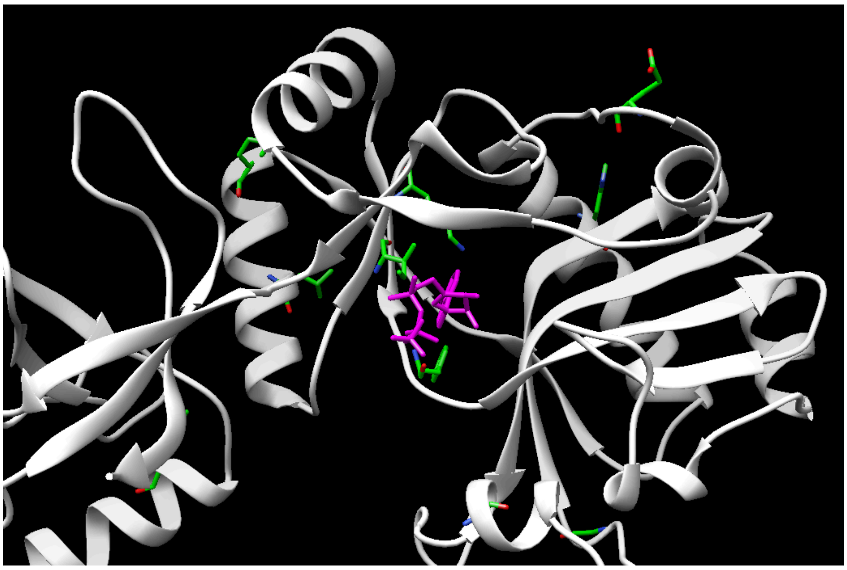


3.3. 1A0I

For this protein, a total of 9 binding sites were predicted. The accuracy, precision and recall of the protein were 0.94, 0.3, and 0.21,respectively.

Looking at the protein in Chimera, we see that there are, again, a number of false positives that should not be ignored. There are 2 predicted binding sites that could constitute a real binding site, although some of the binding residues are missing.

Binding site 1	Binding site 2	Binding site 3	Binding site 4	Binding site 5
A TYR 35	A GLY 104	A GLY 142	A GLU 161	A HIS 173
Binding site 6	Binding site 7	Binding site 8	Binding site 9	
A MET 201	A LEU 207	A ILE 220 A LYS 222	A MET 288	

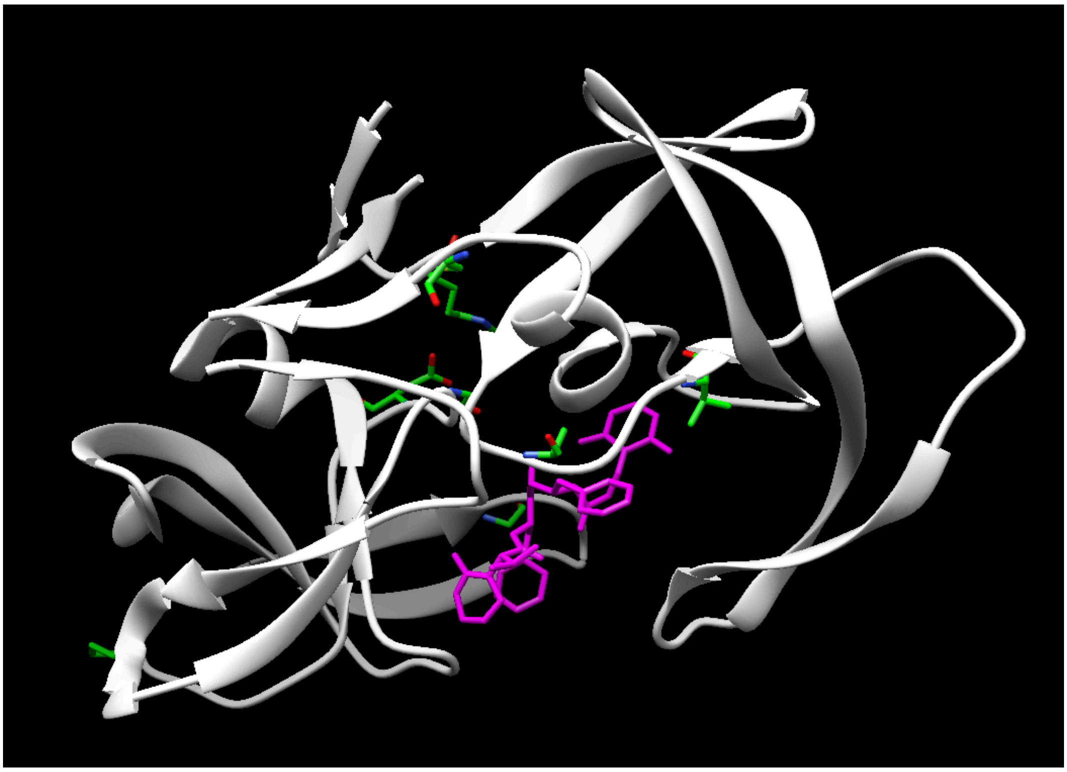


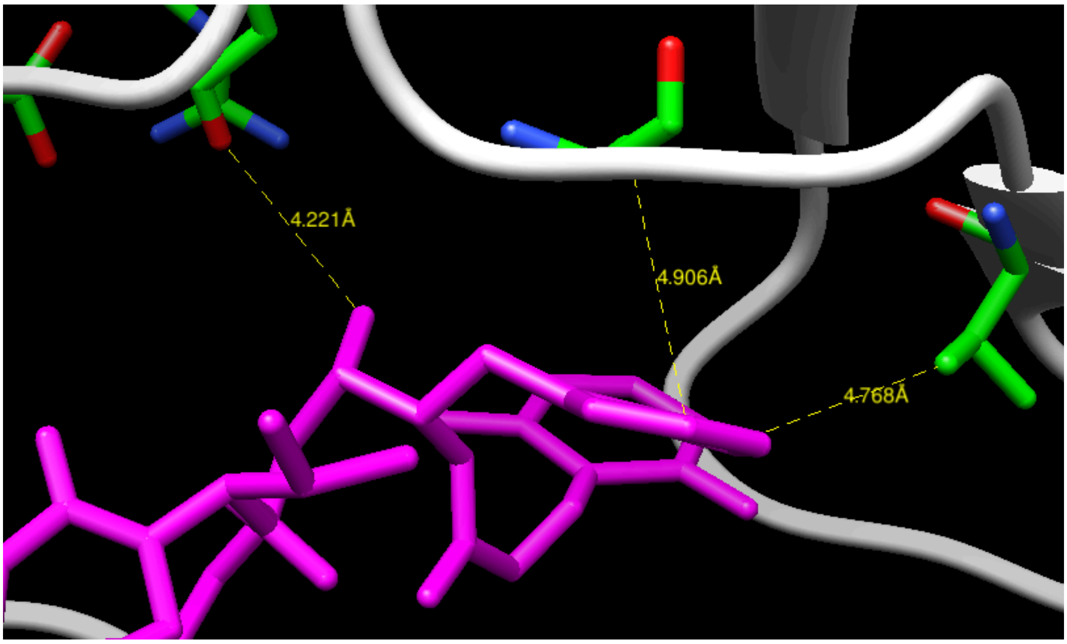
3.4. 1RV7

For this protein, a total of 7 binding sites were predicted. The accuracy, precision and recall of the protein were 0.9, 0.13, and 0.08, respectively. Looking at Chimera, at a first glance we would think that there are almost no correctly predicted binding sites. However, if we calculate the distance between some of the predicted binding residues and the ligand, we see that it is slightly higher than 4 Å. Other articles, such as Chen et al. (2013) state that a binding site can be separated by up to 5 Å from its ligand, so here the problem could be that we decided to choose a threshold of 4 Å, which can lead to an underestimation of the precision, since actual binding sites are being labeled as non-binding sites.

The predicted binding sites that could be part of the real binding site are colored in a darker blue:

Binding site 1	Binding site 2	Binding site 3	Binding site 4
A ARG 8 B ASP 29	A ALA 28	A VAL 32	A CYS 95
Binding site 5	Binding site 6	Binding site 7	
B GLY 27	B PRO 39	B GLY 48	

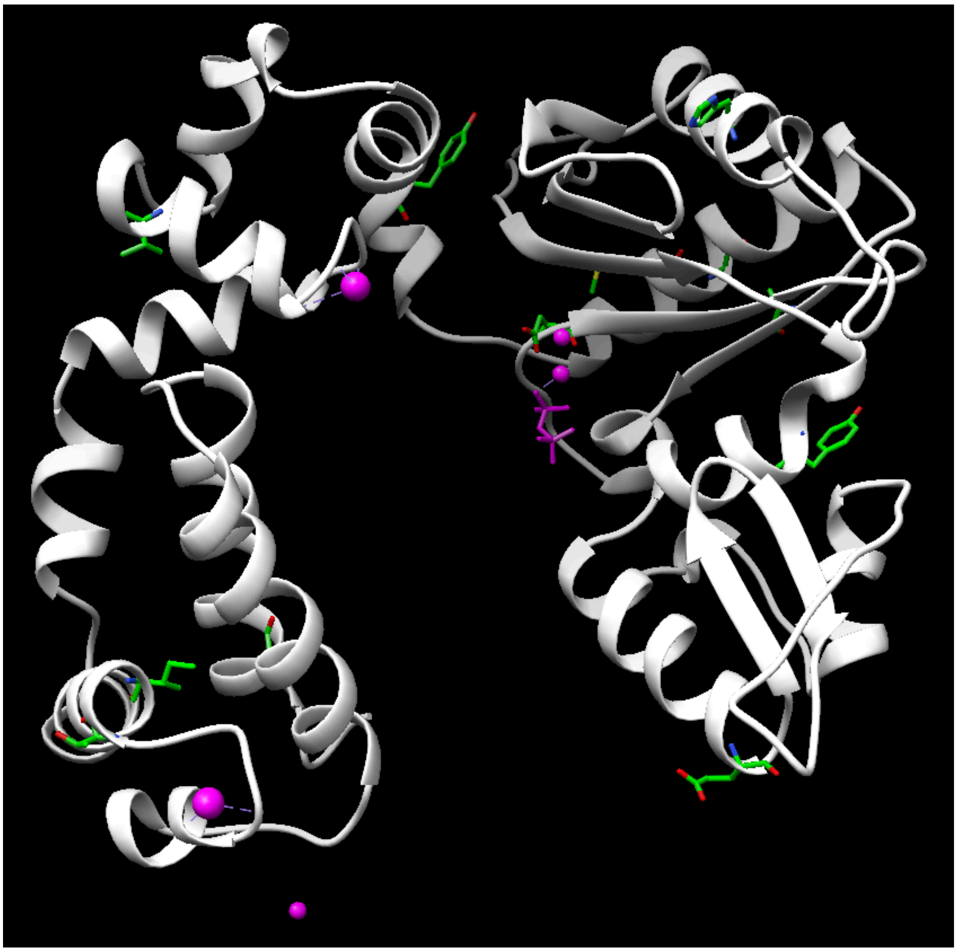




3.5. 9ICV

For this protein, a total of 10 binding sites were predicted. The accuracy, precision and recall of the protein were 0.9, 0.08, and 0.05, respectively. Looking at Chimera, we see that only one of the predicted binding sites could actually be considered part of a real binding site. The number of false positives is high, which lowers the precision. This could be due to our program having a lower performance with single-atom ligands (which are predominant in this case).

Binding site 1	Binding site 2	Binding site 3	Binding site 4	Binding site 5
A THR 16	A THR 67 A ILE 69	A VAL 115	A TYR 142	A MET 158 A ASP 160
Binding site 6	Binding site 7	Binding site 8	Binding site 9	Binding site 10
A ALA 175	A ASP 190	A HIS 212	A TYR 266	A GLU 288



4. CONCLUSION

All in all, the program is capable of predicting some of the residues that conform the real binding sites, but fails to accurately determine all the binding residues, and also to precisely profile the real binding sites. This is mainly due to a very high rate of false positives and false negatives, and also depends on the maximum distance set to ensemble the different binding sites. The user can adjust this last parameter according to their preferences.

The high amount of false positives was to be expected because, even if the rate of false positives was low, non-binding residues greatly outnumber binding residues in any given protein.

Mainly to reduce false negatives, some adjustments could be made to the machine learning model. On one hand, it could be interesting to try other features to see if they perform better than the ones that we selected. On the other hand, as we have mentioned before, changing some parameters such as the C and the gamma to make them more strict could be a way to increase precision, but also a potential cause of overfitting. Unfortunately, we could not test this possibility because we lacked the computational power necessary to do so, so we can not assess whether an improvement in precision would have been made.

To sum up, this program is a good start to predict protein-ligand binding sites, but further improvements should be made in order to guarantee a proper performance.

5. BIBLIOGRAPHY

1. Chen, P., Huang, J. Z., & Gao, X. (2014). LigandRFs: random forest ensemble to identify ligand-binding residues from sequence information alone. *BMC bioinformatics*, *15 Suppl 15*(Suppl 15), S4. https://doi.org/10.1186/1471-2105-15-S15-S4
2. Khazanov NA, Carlson HA. Exploring the composition of protein-ligand binding sites on a large scale. PLoS Comput Biol. 2013;9(11):e1003321. doi: 10.1371/journal.pcbi.1003321. Epub 2013 Nov 21. PMID: 24277997; PMCID: PMC3836696.
3. McGreig, J. E., Uri, H., Antczak, M., Sternberg, M. J. E., Michaelis, M., & Wass, M. N. (2022). 3DLigandSite: structure-based prediction of protein-ligand binding sites. *Nucleic acids research*, *50*(W1), W13–W20. https://doi.org/10.1093/nar/gkac250
4. Meiler, J., Müller, M., Zeidler, A., Schmäschke, F. (2001). Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks. *Molecular modeling annual, 7,* 360-369. https://link.springer.com/article/10.1007/s008940100038
5. Yamaguchi, S., Nakashima, H., Moriwaki, Y., Terada, T., & Shimizu, K. (2022). Prediction of protein mononucleotide binding sites using AlphaFold2 and machine learning. *Computational biology and chemistry*, *100*, 107744. https://doi.org/10.1016/j.compbiolchem.2022.107744
6. Wang, K., Gao, J., Shen, S., Tuszynski, J. A., Ruan, J., & Hu, G. (2013). An accurate method for prediction of protein-ligand binding site on protein surface using SVM and statistical depth function. *BioMed research international*, *2013*, 409658. https://doi.org/10.1155/2013/409658
7. Wong, G. Y., Leung, F. H., & Ling, S. H. (2013). Predicting protein-ligand binding site using support vector machine with protein properties. *IEEE/ACM transactions on computational biology and bioinformatics*, *10*(6), 1517–1529. https://doi.org/10.1109/TCBB.2013.126