Universitat Pompeu Fabra Barcelona

PREDICT-RF User Manual

Paula Mayo and Mariona Isern

Pompeu Fabra University, Barcelona

11 April 2023

Index

Introduction

PREDICT-RF is an algorithm capable of predicting the ligand-binding sites (LBSs) of a given protein. Determining the pocket of a protein and the residues involved in ligand interaction is a major basis in many biostructural fields such as drug design[1].
Moreover, resolving ligand-binding residues can be advantageous in computational prognoses of different physiological and pathological traits[2,3].

However, biological structures are difficult to obtain experimentally. Proof of that is the number of structures we can find in a structured-based repository such as the Protein Data Bank (PDB)[4], a well-known and used database. Compared to Uniprot, PDB had 203,084[5] entries in April 2023, 1,000 times less than Uniprot, which owned 245,871,724 sequence entries in February 2023[6]. Additionally, not all PDB entries present protein-ligand interactions, as well as not all modeled proteins. Thus, LBSs-predicting algorithms are a useful tool that complements experimentally obtained information.

PREDICT-RF is a machine learning-based (ML) algorithm written in Python3 that learns from a dataset of several computed residue characteristics of PDBs (obtained from PDBind[7,8]), and predicts the ligand binding residues by using Random Forest algorithm with undersampling balancing technique.

All useful information about PREDICT-RF is summarized in this manual. In here, the user will find how the algorithm works and what each function does in it, the requirements needed in order to execute PREDICT-RF and how to install it in a Unix-like computer, how to use it and execute it, and lastly, some useful examples.

Basis of the algorithm

Feature dataset construction

The dataset is constructed with structural and sequence information of each residue of the proteins, obtained from PDBind. In order to get all this information, a first algorithm extracts different characteristics:

1. Calculates the distance between residues and the ligand.
2. Extracts information about the positive charge, negative charge, isoelectric point value, polarity, and hydrophobicity for each residue from AAIndex database[9,10].
3. Extracts the secondary structure of each residue using DSSP software[11,12,13].
4. Calculates solvent-accessible surface areas (SASA) for each residue using the Shrake-Rupley algorithm[14].
5. Creates a position-specific scoring matrix (PSSM) using Psiblast[15,16].
6. Calculates entropy from the PSSM output file.
7. Creates a .csv file containing all this information.

Machine learning model

As the actual acquired data shows many fewer binding residues in comparison to non-binding residues, to effectively utilize the extracted features, undersampling balancing technique was applied. Lastly, a Random Forest classification algorithm was used to construct the model. It is also important to note that, as the classification prediction results in a high percentage of false positives, a threshold of 0.7 was used to balance the trade-off between false positives and false negatives predictions.

PREDICT-RF's workflow

1. Determine input type

In order to determine if it has to call the different functions once or as many times as PDB files inside a directory, it has to extract the protein name from the PDB file. Thus, it needs to check for a PDB file or a directory.

2. Getting chains

As PDBs are structured in chains containing residues (which may differ in different chains), PDBs are processed chain by chain. In order to do so, *get_chains* function is called. This function will extract a list with the file's chain, taking a PDB file as input.

Then, the following functions will be embedded inside a loop and will be called for each chain.

3. Features calculation

To obtain all features, *get_sequence_position* and *get_sequence* functions are called in order to get sequence and sequence position information, which will be used by other functions. Then, *residue_hydrophobicity*, *residue_polarity*, *residue_positive*, *residue_negative*, *residue_isoelectric_point*, *extract_secondary_structure*, *SASA*, and *pssm_calculation* functions will be executed.

These first five functions use a sequence as input and return a list with each residue's hydrophobicity, polarity, positive and negative charges, and isoelectric point extracted from the AAIndex1 database, respectively, using aaindex module.

Regarding the *extract_secondary_structure* function, it uses DSSP program in order to calculate secondary structures from a PDB file. However, secondary structure codes are modified to simplify information. DSSP codes '-', 'I', 'T', 'S' are replaced with 'C' (i.e. coil). Then, 'G' code is replaced by 'H' (i.e. helix), and 'B' is replaced by 'E' (i.e. extended, beta-strand).

With respect to the *SASA* function, it uses a PDB file as input and uses the SASA module. This, which is dependent on the biopython module, calculates the solvent-accessible surface area, by residue in this case, using the Shrake-Rupley function.

Finally, *pssm_calculation* is executed. This function uses a .fasta file and a database as input. The .fasta file is created during *get_sequence* execution, and the database is provided by the user in the command line. With this information, it executes Psi blast and takes the PSSM-containing file as output, from which entropy values are calculated following the next formula[17]:

$$SE = \Sigma Pij \cdot lnPij$$

4. Dataframe creation

The next step is to embed all these feature calculations into a dataframe, using pandas module. In this case, a dataframe for each chain has been created, so these are appended in a final dataframe containing information from every chain, that is passed to the prediction functions.

5. Prediction

After obtaining all the necessary information, prediction functions are called. In this case, *generate_prediction* function loads the model, creates a prediction, and returns a pandas dataframe with the ligand-binding residues that have been predicted, indicating their corresponding chain.

6. Output generation

Finally, *generate_output_from_prediction* function is executed. This function, at the same time, calls two other functions: *extract_required_data_from_prediction* and *generate_output_files*. The first one creates and returns two lists, one with each three-letter residue, its position, and its chain, and the other one with only the residue position and its chain.

The last function takes this information and creates two output files: a .txt file with the three-letter residue, its position, and its chain, and a .cmd file containing Chimera commands in order to visualize the predicted residues.

PREDICT-RF’s installation

PREDICT-RF requires some programs, modules, and packages in order to start running:

1. Python 3
2. Psi Blast
3. DSSP
4. Protein database such as SwissProt
5. Biopython
6. Numpy
7. Pandas
8. AAIndex
9. Scikit-Learn
10. tables.py
11. SASA.py
12. Model
13. Chimera, optional

1. Python 3
This program has been written using Python 3.10. Although no other Python versions have been tested, other versions may also work.

2, 3. Psi Blast and DSSP
Psi Blast and DSSP programs must be installed on your computer, taking into account that these must run in any folder and have read, write and execute permissions.

4. Protein database
In order to run psiblast, you need to have access to a protein database. However, the model was trained using SwissProt, so this is the recommended database.

5, 6, 7, 8, 9. Biopython, Numpy, Pandas, AAIndex, Scikit-learn (version 1.1.3)
These modules must be installed on your computer. Remember that these must be in the correct path so Python can find them. These can be easily installed using:

pip install <*module*>

For more information on how to install these programs check https://biopython.org/, https://numpy.org/install/, https://pandas.pydata.org/docs/getting_started/install.html, https://github.com/amckenna41/aaindex, or https://scikit-learn.org/stable/install.html.

10,11. tables.py and SASA.py
tables.py and SASA.py are two required modules. These files are located inside predict-rf.gz.tar folder.

12. Model
As PREDICT-RF is an ML-based algorithm, in order to make predictions, it requires a model that is passed to PREDICT-RF as ‘joblib_RandForest_undersampling_model.pkl’.

13. Chimera
Chimera is an optional program, it is not required in order to run PREDICT-RF. However, the program creates as output a .cmd file, which is passed to Chimera in order to visualize the LBS-forming residues. Be aware that Chimera can read .cmd files, but its newest version, ChimeraX cannot recognize .cmd extension.

Installation

In order to install PREDICT-RF, two downloadable .gz.tar files are available:

- predict-rf_self_installation.tar.gz:

A tar.gz. file containing predict-rf.py, predict-rf.yml and other necessary files. First, *untar* and *unzip* the folder, create the conda environment with predict-rf.yml file and activate the predict-rf environment. Then, call python and run predict-rf.py file.

```
tar -xvf predict-rf_self_installation.tar.gz
conda env create -f predict-rf.yml
conda activate project
python predict-rf.py -i INPUT -db DATABASE
```

It can be downloaded from: https://github.com/pmayog/PREDICT-RF.

Note: When creating project environment, check if “~/bin/miniconda3/envs/predict-rf” path exists in your computer. Otherwise, modify predict-rf.yml file and change the path.

- predict-rf.tar.gz:

A tar.gz file link containing a predict-rf executable file and all its dependencies. The user has to *untar* and *unzip* the file and write the path to the executable in order to call the program (untared predict-rf.tar.gz is named “dist”).
It is recommended to make an alias of the executable file.

```
tar -xvf predict-rf.tar.gz
alias predict-rf="path_to_/dist/predict-rf/predict-rf"
predict-rf -i INPUT -db DATABASE
```

It can be downloaded from:
https://drive.google.com/drive/folders/1D3g-f9G_Ji_eyrpA4cUR8V9SGsIVTRdN?usp=share_link.
Note: Remember that Psiblast and DSSP programs must run as commands (as dssp and psiblast commands).

Using PREDICT-RF

PREDICT-RF is an algorithm developed for Linux-based operating systems that run in the terminal with the following command and options:

```
predict-rf.py -i INPUT -db DATABASE [-e] [-k] [-v]
```

-i, --input: INPUT

INPUT must be a PDB file or a directory including at least one PDB file. If a non-PDB file is given as input, it will be skipped.

-db, --database: DATABASE

DATABASE must include the path to the database used when running PsiBlast. Database selection can affect Psi Blast output. If there is no PSSM output, try to change the database.

[-e], [--eval]

OPTIONAL. Specify the e-value used to run PsiBlast, 0.001 by default. This selection can affect Psi Blast output. If there is no PSSM output, try to change the threshold value.

[-k], [--keep]

OPTIONAL. Will keep files created by DSSP and psi blast, as well as other temporary files that will be commented later on (see Output section).

[-v], [--verbose]

OPTIONAL. Display in the terminal all information when running the program.

Output

After running PREDICT-RF, there will appear two types of files: a .txt file and a .cmd file. These will be named *protein*_binding_residues.txt and *protein*_binding_residues.cmd, respectively, where *protein* represents the name of the given PDB file.

1. *protein*_binding_residues.txt

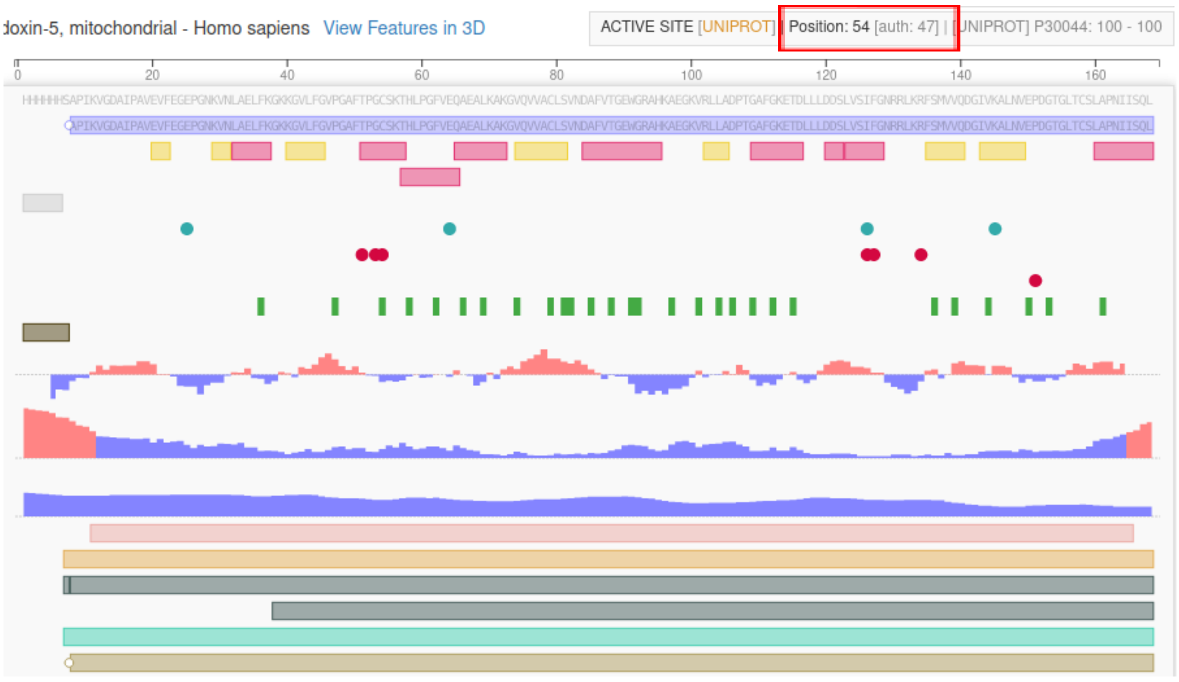
This file contains a list of all binding-related residues that have been predicted. It presents three columns per line representing:

Residue Position Chain

```
GLY   6 A
PRO  40 A
THR  44 A
CYS  47 A
SER  74 A
PHE  79 A
TRP  84 A
ASP 109 A
LEU 161 A
GLY   6 C
PRO  40 C
THR  44 C
CYS  47 C
```

Figure 1. 4mmm_binding_residues.txt visualization.

Note: It must be taken into consideration that positions are an author's reference and may differ from PDB's standardized positions. It can be easily seen in PDB's sequence section:


ɔoxin-5, mitochondrial - Homo sapiens View Features in 3D

ACTIVE SITE [UNIPROT] | Position: 54 [auth: 47] | [UNIPROT] P30044: 100 - 100

0
20
40
60
80
100
120
140
160
HHHHHHSAPIKVGDAIPAVEVFEGEPGNKVNLAELFKGKKGVLFGVPGAFTPGCSKTHLPGFVEQAEALKAKGVQVVACLSVNDAFVTGEWGRAHKAEGKVRLLADPTGAFGKETDLLLDDSLVSIFGNRRLKRFSMVVQDGIVKALNVEPDGTGLTCSLAPNIISQL
APIKVGDAIPAVEVFEGEPGNKVNLAELFKGKKGVLFGVPGAFTPGCSKTHLPGFVEQAEALKAKGVQVVACLSVNDAFVTGEWGRAHKAEGKVRLLADPTGAFGKETDLLLDDSLVSIFGNRRLKRFSMVVQDGIVKALNVEPDGTGLTCSLAPNIISQL

Figure 2. 4mmm PDB sequence section.

2. *protein*_binding_residue.cmd

This file represents a set of commands executable by Chimera. It can be opened with the following command:

chimera *protein*_binding_residues.cmd

The commands included in the file can be resumed as follows:

1. Open PDB
2. Delete solvent
3. Show surface
4. Color surface in white
5. Color predicted residues in red
6. Label predicted residues

```
open pdb/4mmm.pdb
del solvent
surf
color white,s
color red,s :6.A
rlabel :6.A
color red,s :40.A
rlabel :40.A
color red,s :44.A
rlabel :44.A
color red,s :47.A
rlabel :47.A
color red,s :74.A
rlabel :74.A
color red,s :79.A
rlabel :79.A
```

Figure 3. 4mmm_binding_residues.cmd visualization.

As a result, we can obtain something similar to the following image:

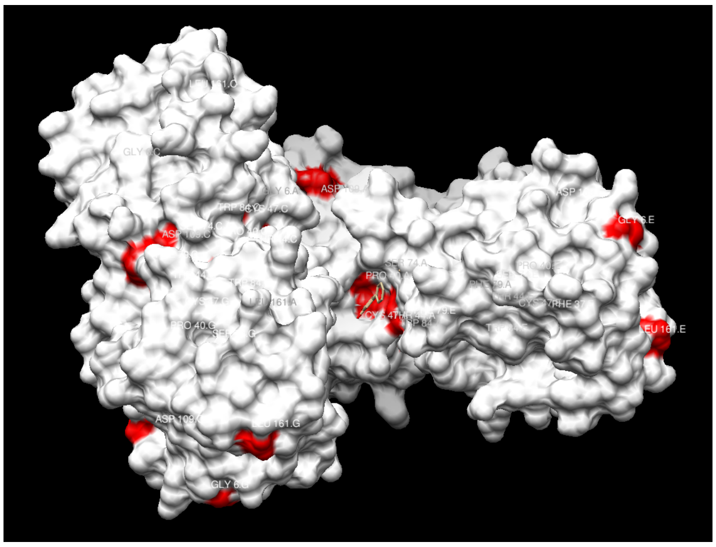


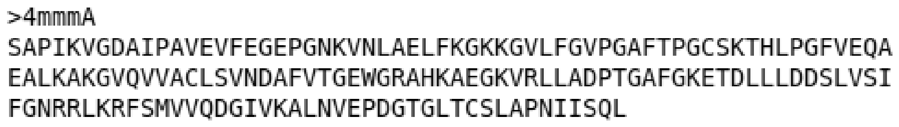
Figure 4. 4mmm_binding_residues.cmd visualization in Chimera.

Note: .cmd files are only supported by classic Chimera software but not by its newest version ChimeraX.

Moreover, if --keep or -k option is used, other files will be created:

1. *proteinchain*.fa

A fasta file for each chain given a protein (e.g. 4mmmA.fa, 4mmmC.fa, 4mmmE.fa, 4mmmG.fa):

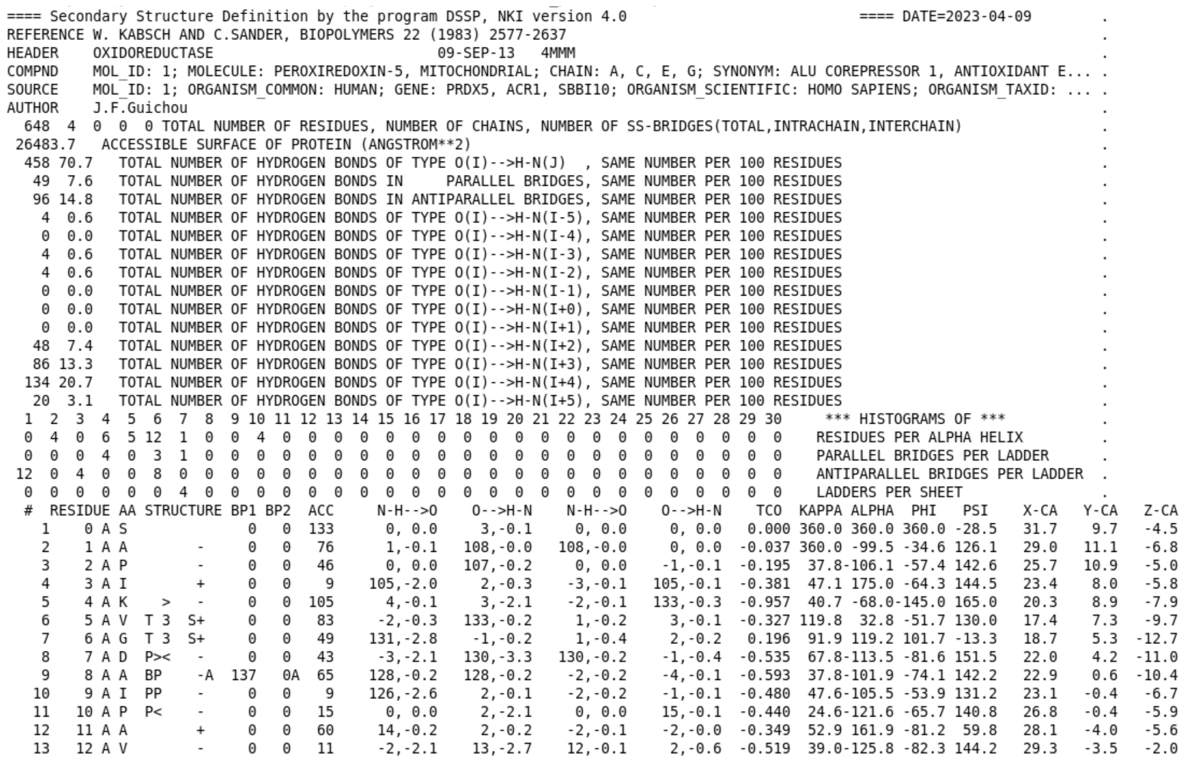


```
>4mmmA
SAPIKVGDAIPAVEVFEGEPGNKVNLAELFKGKKGVLFGVPGAFTPGCSKTHLPGFVEQA
EALKAKGVQVVACLSVNDAFVTGEWGRAHKAEGKVRLLADPTGAFGKETDLLLDDSLVSI
FGNRRLKRFSMVVQDGIVKALNVEPDGTGLTCSLAPNIISQL
```

Figure 5. 4mmmA.fa visualization.

2. *protein*_protein.dssp

A .dssp file for each PDB file, containing secondary structures prediction per residue:

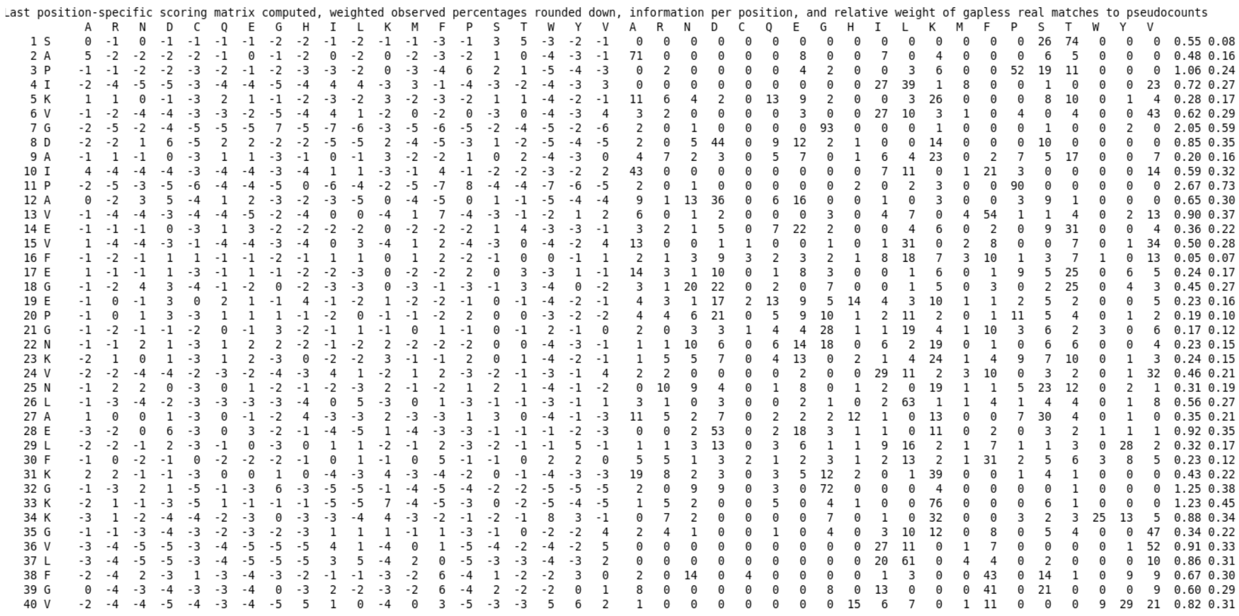


```
==== Secondary Structure Definition by the program DSSP, NKI version 4.0                    ==== DATE=2023-04-09        .
REFERENCE W. KABSCH AND C.SANDER, BIOPOLYMERS 22 (1983) 2577-2637                                                      .
HEADER    OXIDOREDUCTASE                          09-SEP-13   4MMM                                                      .
COMPND    MOL_ID: 1; MOLECULE: PEROXIREDOXIN-5, MITOCHONDRIAL; CHAIN: A, C, E, G; SYNONYM: ALU COREPRESSOR 1, ANTIOXIDANT E... .
SOURCE    MOL_ID: 1; ORGANISM_COMMON: HUMAN; GENE: PRDX5, ACR1, SBBI10; ORGANISM_SCIENTIFIC: HOMO SAPIENS; ORGANISM_TAXID: ... .
AUTHOR    J.F.Guichou                                                                                                   .
  648  4  0  0  0 TOTAL NUMBER OF RESIDUES, NUMBER OF CHAINS, NUMBER OF SS-BRIDGES(TOTAL,INTRACHAIN,INTERCHAIN)                .
 26483.7   ACCESSIBLE SURFACE OF PROTEIN (ANGSTROM**2)                                                                   .
  458 70.7   TOTAL NUMBER OF HYDROGEN BONDS OF TYPE O(I)-->H-N(J)  , SAME NUMBER PER 100 RESIDUES                          .
   49  7.6   TOTAL NUMBER OF HYDROGEN BONDS IN     PARALLEL BRIDGES, SAME NUMBER PER 100 RESIDUES                          .
   96 14.8   TOTAL NUMBER OF HYDROGEN BONDS IN ANTIPARALLEL BRIDGES, SAME NUMBER PER 100 RESIDUES                          .
    4  0.6   TOTAL NUMBER OF HYDROGEN BONDS OF TYPE O(I)-->H-N(I-5), SAME NUMBER PER 100 RESIDUES                          .
    0  0.0   TOTAL NUMBER OF HYDROGEN BONDS OF TYPE O(I)-->H-N(I-4), SAME NUMBER PER 100 RESIDUES                          .
    4  0.6   TOTAL NUMBER OF HYDROGEN BONDS OF TYPE O(I)-->H-N(I-3), SAME NUMBER PER 100 RESIDUES                          .
    4  0.6   TOTAL NUMBER OF HYDROGEN BONDS OF TYPE O(I)-->H-N(I-2), SAME NUMBER PER 100 RESIDUES                          .
    0  0.0   TOTAL NUMBER OF HYDROGEN BONDS OF TYPE O(I)-->H-N(I-1), SAME NUMBER PER 100 RESIDUES                          .
    0  0.0   TOTAL NUMBER OF HYDROGEN BONDS OF TYPE O(I)-->H-N(I+0), SAME NUMBER PER 100 RESIDUES                          .
    0  0.0   TOTAL NUMBER OF HYDROGEN BONDS OF TYPE O(I)-->H-N(I+1), SAME NUMBER PER 100 RESIDUES                          .
   48  7.4   TOTAL NUMBER OF HYDROGEN BONDS OF TYPE O(I)-->H-N(I+2), SAME NUMBER PER 100 RESIDUES                          .
   86 13.3   TOTAL NUMBER OF HYDROGEN BONDS OF TYPE O(I)-->H-N(I+3), SAME NUMBER PER 100 RESIDUES                          .
  134 20.7   TOTAL NUMBER OF HYDROGEN BONDS OF TYPE O(I)-->H-N(I+4), SAME NUMBER PER 100 RESIDUES                          .
   20  3.1   TOTAL NUMBER OF HYDROGEN BONDS OF TYPE O(I)-->H-N(I+5), SAME NUMBER PER 100 RESIDUES                          .
  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30     *** HISTOGRAMS OF ***           .
  0  4  0  6  5 12  1  0  0  4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    RESIDUES PER ALPHA HELIX         .
  0  0  0  4  0  3  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    PARALLEL BRIDGES PER LADDER      .
 12  0  4  0  0  8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    ANTIPARALLEL BRIDGES PER LADDER  .
  0  0  0  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    LADDERS PER SHEET                .
  #  RESIDUE AA STRUCTURE BP1 BP2  ACC     N-H-->O    O-->H-N    N-H-->O    O-->H-N    TCO  KAPPA ALPHA  PHI   PSI    X-CA   Y-CA   Z-CA
    1    0 A S              0   0  133      0, 0.0     3,-0.1     0, 0.0     0, 0.0   0.000 360.0 360.0 360.0 -28.5   31.7    9.7   -4.5
    2    1 A A        -     0   0   76      1,-0.1   108,-0.0   108,-0.0     0, 0.0  -0.037 360.0 -99.5 -34.6 126.1   29.0   11.1   -6.8
    3    2 A P        -     0   0   46      0, 0.0   107,-0.2     0, 0.0    -1,-0.1  -0.195  37.8-106.1 -57.4 142.6   25.7   10.9   -5.0
    4    3 A I        +     0   0    9    105,-2.0     2,-0.3    -3,-0.1   105,-0.1  -0.381  47.1 175.0 -64.3 144.5   23.4    8.0   -5.8
    5    4 A K    >   -     0   0  105      4,-0.1     3,-2.1    -2,-0.1   133,-0.3  -0.957  40.7 -68.0-145.0 165.0   20.3    8.9   -7.9
    6    5 A V  T 3  S+     0   0   83     -2,-0.3   133,-0.2     1,-0.2     3,-0.1  -0.327 119.8  32.8 -51.7 130.0   17.4    7.3   -9.7
    7    6 A G  T 3  S+     0   0   49    131,-2.8    -1,-0.2     1,-0.4     2,-0.2   0.196  91.9 119.2 101.7 -13.3   18.7    5.3  -12.7
    8    7 A D  P><   -     0   0   43     -3,-2.1   130,-3.3   130,-0.2    -1,-0.4  -0.535  67.8-113.5 -81.6 151.5   22.0    4.2  -11.0
    9    8 A A  BP    -A  137   0A  65    128,-0.2   128,-0.2    -2,-0.2    -4,-0.1  -0.593  37.8-101.9 -74.1 142.2   22.9    0.6  -10.4
   10    9 A I  PP    -     0   0    9    126,-2.6     2,-0.1    -2,-0.2    -1,-0.1  -0.480  47.6-105.5 -53.9 131.2   23.1   -0.4   -6.7
   11   10 A P  P<    -     0   0   15      0, 0.0     2,-2.1     0, 0.0    15,-0.1  -0.440  24.6-121.6 -65.7 140.8   26.8   -0.4   -5.9
   12   11 A A        +     0   0   60     14,-0.2     2,-0.2    -2,-0.1    -2,-0.0  -0.349  52.9 161.9 -81.2  59.8   28.1   -4.0   -5.6
   13   12 A V        -     0   0   11     -2,-2.1    13,-2.7    12,-0.1     2,-0.6  -0.519  39.0-125.8 -82.3 144.2   29.3   -3.5   -2.0
```

Figure 6. 4mmm.dssp visualization.

3. *proteinchain*.pssm

A .pssm file for each chain given a protein (e.g. 4mmmA.pssm, 4mmmC.pssm, 4mmmE.pssm, 4mmmG.pssm) containing two matrices:



```
Last position-specific scoring matrix computed, weighted observed percentages rounded down, information per position, and relative weight of gapless real matches to pseudocounts
         A   R   N   D   C   Q   E   G   H   I   L   K   M   F   P   S   T   W   Y   V   A   R   N   D   C   Q   E   G   H   I   L   K   M   F   P   S   T   W   Y   V
   1 S   0  -1   0  -1  -1  -1  -1  -2  -2  -1  -2  -1  -1  -3  -1   3   5  -3  -2  -1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  26  74   0   0   0  0.55 0.08
   2 A   5  -2  -2  -2  -2  -1   0  -1  -2   0  -2   0  -2  -3  -2   1   0  -4  -3  -1  71   0   0   0   0   0   8   0   0   7   0   4   0   0   0   6   5   0   0   0  0.48 0.16
   3 P  -1  -1  -2  -2  -3  -2  -1  -2  -3  -3  -2   0  -3  -4   6   2   1  -5  -4  -3   0   2   0   0   0   0   4   2   0   0   3   6   0   0  52  19  11   0   0   0  1.06 0.24
   4 I  -2  -4  -5  -5  -3  -4  -4  -5  -4   4   4  -3   3  -1  -4  -3  -2  -4  -3   3   0   0   0   0   0   0   0   0   0  27  39   1   8   0   0   1   0   0   0  23  0.72 0.27
   5 K   1   1   0  -1  -3   2   1  -1  -2  -3  -2   3  -2  -3  -2   1   1  -4  -2  -1  11   6   4   2   0  13   9   2   0   0   3  26   0   0   0   8  10   0   1   4  0.28 0.17
   6 V  -1  -2  -4  -4  -3  -3  -2  -5  -4   4   1  -2   0  -2   0  -3   0  -4  -3   4   3   2   0   0   0   0   3   0   0  27  10   3   1   0   4   0   4   0   0  43  0.62 0.29
   7 G  -2  -5  -2  -4  -5  -5  -5   7  -5  -7  -6  -3  -5  -6  -5  -2  -4  -5  -2  -6   2   0   1   0   0   0   0  93   0   0   0   1   0   0   0   1   0   0   2   0  2.05 0.59
   8 D  -2  -2   1   6  -5   2   2  -2  -2  -5  -5   2  -4  -5  -3   1  -2  -5  -4  -5   2   0   5  44   0   9  12   2   1   0   0  14   0   0   0  10   0   0   0   0  0.85 0.35
   9 A  -1   1  -1   0  -3   1   1  -3  -1   0  -1   3  -2  -2   1   0   2  -4  -3   0   4   7   2   3   0   5   7   0   1   6   4  23   0   2   7   5  17   0   0   7  0.20 0.16
  10 I   4  -4  -4  -4  -3  -4  -4  -3  -4   1   1  -3  -1   4  -1  -2  -2  -3  -2   2  43   0   0   0   0   0   0   0   0   7  11   0   1  21   3   0   0   0   0  14  0.59 0.32
  11 P  -2  -5  -3  -5  -6  -4  -4  -5   0  -6  -4  -2  -5  -7   8  -4  -4  -7  -6  -5   2   0   1   0   0   0   0   0   2   0   2   3   0   0  90   0   0   0   0   0  2.67 0.73
  12 A   0  -2   3   5  -4   1   2  -3  -2  -3  -5   0  -4  -5   0   1  -1  -5  -4  -4   9   1  13  36   0   6  16   0   0   1   0   3   0   0   3   9   1   0   0   0  0.65 0.30
  13 V  -1  -4  -4  -3  -4  -4  -5  -2  -4   0   0  -4   1   7  -4  -3  -1  -2   1   2   6   0   1   2   0   0   0   3   0   4   7   0   4  54   1   1   4   0   2  13  0.90 0.37
  14 E  -1  -1  -1   0  -3   1   3  -2  -2  -2  -2   0  -2  -2  -2   1   4  -3  -3  -1   3   2   1   5   0   7  22   2   0   0   4   6   0   2   0   9  31   0   0   4  0.36 0.22
  15 V   1  -4  -4  -3  -1  -4  -4  -3  -4   0   3  -4   1   2  -4  -3   0  -4  -2   4  13   0   0   1   1   0   0   1   0   1  31   0   2   8   0   0   7   0   1  34  0.50 0.28
  16 F  -1  -2  -1   1   1  -1  -1  -2  -1   1   1   0   1   2  -2  -1   0   0  -1   1   2   1   3   9   3   2   3   2   1   8  18   7   3  10   1   3   7   1   0  13  0.05 0.07
  17 E   1  -1  -1   1  -3  -1   1  -1  -2  -2  -3   0  -2  -2   2   0   3  -3   1  -1  14   3   1  10   0   1   8   3   0   0   1   6   0   1   9   5  25   0   6   5  0.24 0.17
  18 G  -1  -2   4   3  -4  -1  -2   0  -2  -3  -3   0  -3  -1  -3  -1   3  -4   0  -2   3   1  20  22   0   2   0   7   0   0   1   5   0   3   0   2  25   0   4   3  0.45 0.27
  19 E  -1   0  -1   3   0   2   1  -1   4  -1  -2   1  -2  -2  -1   0  -1  -4  -2  -1   4   3   1  17   2  13   9   5  14   4   3  10   1   1   2   5   2   0   0   5  0.23 0.16
  20 P  -1   0   1   3  -3   1   1   1  -1  -2   0  -1  -1  -2   2   0   0  -3  -2  -2   4   4   6  21   0   5   9  10   1   2  11   2   0   1  11   5   4   0   1   2  0.19 0.10
  21 G  -1  -2  -1  -1  -2   0  -1   3  -2  -1   1  -1   0   1  -1   0  -1   2  -1   0   2   0   3   3   1   4   4  28   1   1  19   4   1  10   3   6   2   3   0   6  0.17 0.12
  22 N  -1  -1   2   1  -3   1   2   2  -2  -1  -2   2  -2  -2  -2   0   0  -4  -3  -1   1   1  10   6   0   6  14  18   0   6   2  19   0   1   0   6   6   0   0   4  0.23 0.15
  23 K  -2   1   0   1  -3   1   2  -3   0  -2  -2   3  -1  -1   2   0   1  -4  -2  -1   1   5   5   7   0   4  13   0   2   1   4  24   1   4   9   7  10   0   1   3  0.24 0.15
  24 V  -2  -2  -4  -4  -2  -3  -2  -4  -3   4   1  -2   1   2  -3  -2  -1  -3  -1   4   2   2   0   0   0   0   2   0   0  29  11   2   3  10   0   3   2   0   1  32  0.46 0.21
  25 N  -1   2   2   0  -3   0   1  -2  -1  -2  -3   2  -1  -2   1   2   1  -4  -1  -2   0  10   9   4   0   1   8   0   1   2   0  19   1   1   5  23  12   0   2   1  0.31 0.19
  26 L  -1  -3  -4  -2  -3  -3  -3  -3  -4   0   5  -3   0   1  -3  -1  -1  -3  -1   1   3   1   0   3   0   0   2   1   0   2  63   1   1   4   1   4   4   0   1   8  0.56 0.27
  27 A   1   0   0   1  -3   0  -1  -2   4  -3  -3   2  -3  -3   1   3   0  -4  -1  -3  11   5   2   7   0   2   2   2  12   1   0  13   0   0   7  30   4   0   1   0  0.35 0.21
  28 E  -3  -2   0   6  -3   0   3  -2  -1  -4  -5   1  -4  -3  -3  -1  -1  -1  -2  -3   0   0   2  53   0   2  18   3   1   1   0  11   0   2   0   3   2   1   1   1  0.92 0.35
  29 L  -2  -2  -1   2  -3  -1   0  -3   0   1   1  -2  -1   2  -3  -2  -1  -1   5  -1   1   1   3  13   0   3   6   1   1   9  16   2   1   7   1   1   3   0  28   2  0.32 0.17
  30 F  -1   0  -2  -1   0  -2  -2  -2  -1   0   1  -1   0   5  -1  -1   0   2   2   0   5   5   1   3   2   1   2   3   1   2  13   2   1  31   2   5   6   3   8   5  0.23 0.12
  31 K   2   2  -1  -1  -3   0   0   1   0  -4  -3   4  -3  -4  -2   0  -1  -4  -3  -3  19   8   2   3   0   3   5  12   2   0   1  39   0   0   1   4   1   0   0   0  0.43 0.22
  32 G  -1  -3   2   1  -5  -1  -3   6  -3  -5  -5  -1  -4  -5  -4  -2  -2  -5  -5  -5   2   0   9   9   0   3   0  72   0   0   0   4   0   0   0   0   1   0   0   0  1.25 0.38
  33 K  -2   1  -1  -3  -5   1  -1  -1  -1  -5  -5   7  -4  -5  -3   0  -2  -5  -4  -5   1   5   2   0   0   5   0   4   1   0   0  76   0   0   0   6   1   0   0   0  1.23 0.45
  34 K  -3   1  -2  -4  -4  -2  -3   0  -3  -3  -4   4  -3  -2  -1  -2  -1   8   3  -1   0   7   2   0   0   0   0   7   0   1   0  32   0   0   3   2   3  25  13   5  0.88 0.34
  35 G  -1  -1  -3  -4  -3  -2  -3  -2  -3   1   1   1  -1   1  -3  -1   0  -2  -2   4   2   4   1   0   0   1   0   4   0   3  10  12   0   8   0   5   4   0   0  47  0.34 0.22
  36 V  -3  -4  -5  -5  -3  -4  -5  -5  -5   4   1  -4   0   1  -5  -4  -2  -4  -2   5   0   0   0   0   0   0   0   0   0  27  11   0   1   7   0   0   0   0   1  52  0.91 0.33
  37 L  -3  -4  -5  -5  -3  -4  -5  -5  -5   3   5  -4   2   0  -5  -3  -3  -4  -3   2   0   0   0   0   0   0   0   0   0  20  61   0   4   4   0   2   0   0   0  10  0.86 0.31
  38 F  -2  -4   2  -3   1  -3  -4  -3  -2  -1  -1  -3  -2   6  -4   1  -2  -2   3   0   2   0  14   0   4   0   0   0   0   1   3   0   0  43   0  14   1   0   9   9  0.67 0.30
  39 G   0  -4  -3  -4  -3  -3  -4   0  -3   2  -2  -3  -2   6  -4   2  -2  -2   0   1   8   0   0   0   0   0   0   8   0  13   0   0   0  41   0  21   0   0   0   9  0.60 0.29
  40 V  -2  -4  -4  -5  -4  -3  -4  -5   5   1   0  -4   0   3  -5  -3  -3   5   6   2   1   0   0   0   0   0   0   0  15   6   7   0   1  11   0   0   0   9  29  21  0.82 0.31
```

Figure 7. 4mmmA.pssm visualization.

Examples

Single PDB file without optional arguments

In this example, we will use ‘6uh0.pdb’ as the input file, which contains the carbonic anhydrase 2 protein in complex with SB4-202[18].

We will run the following command:

```
python predict-rf.py -i 6uh0.pdb -db ~/Documents/swissprot
```

In this case, as we are not using other options, we will not have output files from Psi Blast, DSSP, and other temporary files, and it will not show any message in the terminal except warnings.

After running this, we will get two files:

- 6uh0_binding_residues.txt
- 6uh0_binding_residues.cmd

Directory with PDB and other files without optional arguments

In this case, we will use as input a directory called ‘random’, in which there are three PDB files[19,20,21], five Python scripts, and one text file.

We will run the following command:

```
python predict-rf.py -i random/ -db ~/Documents/swissprot
```

Although -v option is not introduced, the terminal will display a message to warn the user that some files will be skipped because are not PDB files:

```
Skipping file. Input must contain PDB file/s.
```

As a result of this, output files will just correspond to PDB files:

- 5fck_binding_residues.cmd
- 5fck_binding_residues.txt
- 6upj_binding_residues.cmd
- 6upj_binding_residues.txt
- 7std_binding_residues.cmd
- 7std_binding_residues.txt

Single PDB file without Psiblast output, using -k option

Now, we will use a PDB file with no Psiblast result. To do so, we will use ‘5d1d.pdb’[22]:

```
python predict-rf.py -i 5d1d.pdb -db ~/Documents/swissprot -k
```

In this situation, ‘5d1d.pdb’ contains four chains (A, B, C, D), but only chains A and B obtain a PSSM matrix when running Psiblast. As a result, the terminal will show a message and chains will be skipped:

```
Empty PSSM...
Skipping chain...
```

Thus, the output files (both .txt and .cmd) will just contain chains A and B, as chains C and D are not passed to prediction functions. In this case, however, as -k option has been used, there will also appear other files:

- Fasta file for each chain: 5d1dA.fa, 5d1dB.fa, 5d1dC.fa and 5d1dD.fa
- DSSP file per PDB file: 5d1d_protein.dssp
- Psiblast PSSM file for each successful chain: 5d1dA_sprot.pssm and 5d1dB_sprot.pssm

Note: PSSM missing output can be due to database or threshold selection, try to modify -db and/or -e options.

Directory with PDB files, using -v and -e options

In this example, we are using a folder called ‘pdb’ that just contains PDB files[23,24,25]. We will also use -v and -e options with a threshold of 0.0001, so we will execute:

```
python predict-rf.py -i pdb -db ~/Documents/swissprot -e 0.0001 -v
```

Using -v option will inform us on what the program is doing, and the terminal will show the following information:

```
Starting with: 1sbi
Getting chains...
Done
Getting sequence positions...
Done
```

```
Getting sequence...
Done
Getting residues hydrophobicity...
Done
Getting residues polarity...
Done
Getting residues positive charge...
Done
Getting residues negative charge...
Done
Getting residues isoelectric point...
Done
Getting residues secondary structure…
Done
Getting residues solvent accesible surface area...
Done
Getting PSSM...
Warning: [psiblast] Query_1 1sbi: Composition-based score adjustment conditioned on sequence
properties and unconditional composition-based score adjustment is not supported with PSSMs,
resetting to default value of standard composition-based statistics
Done
```

Note: Psiblast warning will always appear (when using -v option and without -v option). Other warnings can also appear.

References

1. Zhao J, Cao Y, Zhang L. Exploring the computational methods for protein-ligand binding site prediction. *Comput Struct Biotechnol J*. 2020; 18: 417-426.

2. Yang C, Chen EA, Zhang Y. Protein–Ligand Docking in the Machine-Learning Era. *Molecules*. 2022; 27(14): 4568.

3. Jayadeepa RM, Sharma S. Computational models for 5αR inhibitors for treatment of prostate cancer: review of previous works and screening of natural inhibitors of 5αR2. *Curr Comput Aided Drug Des*. 2011; 7(4): 231-237.

4. RCSB. Protein Data Bank [Internet]. PDB; 2023. Available from: https://www.rcsb.org/.

5. RCSB. PDB Statistics: Overall Growth of Released Structures Per Year [Internet]. PDB; 2023. Available from: https://www.rcsb.org/stats/growth/growth-released-structures.

6. Ebi-EMBL. Current Release Statistics [Internet]. Ebi-EMBL; 2023. Available from: https://www.ebi.ac.uk/uniprot/TrEMBLstats.

7. Wang R, Fang X, Lu Y, Yang CY, Wang S. The PDBbind database: methodologies and updates. *J Med Chem*. 2005; 48(12): 4111–4119.

8. Wang S, University of Michigan. PDB bind [Internet]. Fudan University; 2023. Available from: http://www.pdbbind.org.cn/.

9. McKenna A. aaindex [Internet]. McKenna A; 2021. Available from: https://github.com/amckenna41/aaindex.

10. Kawashima S, Ogata H, Kanehisa M. AAindex: Amino Acid Index Database. *Nucleic Acids Res*. 1999;27(1): 368-369.

11. Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*. 1983; 22(12): 2577-2637.

12. Touw WG, Baakman C, Black J, Beek T, Krieger E, Joosten RP, Vriend G. A series of PDB-related databases for everyday needs. Nucleic Acids Res. 2015; 43: D364-D368.

13. Kabsch W, Sander C. DSSP [Internet]. Available from: https://swift.cmbi.umcn.nl/gv/dssp/.

14. Rodrigues J. Biopython [Internet]. Rodrigues J.; 2020. Available from: https://github.com/biopython/biopython/blob/master/Bio/PDB/SASA.py.

15. Altschul SF, Madden TL, Schäffer AA. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*. 1997; 25: 3389–3402.

16. National Center for Biotechnology Information. Blast [Internet]. NCBI; 2023. Available from: https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE=Proteins&PROGRAM=blastp&RUN_PSIBLAST=on.

17. Altschul SF, Gertz EM, Agarwala R, Schaffer AA, Yu Y. PSI-BLAST pseudocounts and the minimum description length principle. *Nucleic Acids Res*. 2009; 37(3): 815–824.

18. Bua S, Lomelino C, Murray AB, Osman SM, ALOthman ZA, Bozdag M, Abdel-Aziz HA, et al. "A Sweet Combination": Developing Saccharin and Acesulfame K Structures for Selectively Targeting the Tumor-Associated Carbonic Anhydrases IX and XII. *J Med Chem*. 2020; 63: 321-333.

19. Maibaum J, Liao SM, Vulpetti A, Ostermann N, Randl S, Rudisser S, Lorthiois E, et al. Small-molecule factor D inhibitors targeting the alternative complement pathway. Nat Chem Biol. 2016; 12: 1105-1110.

20. Romines KR, Watenpaugh KD, Tomich PK, Howe WJ, Morris JK, Lovasz KD, Mulichak AM, et al. Use of medium-sized cycloalkyl rings to enhance secondary binding: discovery of a new class of human immunodeficiency virus (HIV) protease inhibitors. *J Med Chem*. 1995; 38: 1884-1891.

21. Wawrzak Z, Sandalova T, Steffens JJ, Basarab GS, Lundqvist T, Lindqvist Y, Jordan DB. High-resolution structures of scytalone dehydratase-inhibitor complexes crystallized at physiological pH. *Proteins*. 1999; 35: 425-439.

22. Decroos C, Christianson NH, Gullett LE, Bowman CM, Christianson KE, Deardorff MA, Christianson DW. Biochemical and Structural Characterization of HDAC8 Mutants Associated with Cornelia de Lange Syndrome Spectrum Disorders. *Biochemistry*. 2015; 54: 6501-6513.

23. Kidd RD, Yennawar HP, Sears P, Wong CH, Farber GK. A weak calcium binding site in subtilisin BPN has a dramatic effect on protein stability. *J Am Chem Soc*. 1996; 118: 1645-1650.

24. Aguirre C, Brink TT, Guichou JF, Cala O, Krimm I. Comparing Binding Modes of Analogous Fragments Using NMR in Fragment-Based Drug Design: Application to PRDX5. *PLoS One.* 2014; 9: e102300-e102300.

25. James LC, Roversi P, Tawfik D. Antibody Multispecificity Mediated by Conformational Diversity. *Science.* 2003; 299: 1362.