**Charles University**
**Faculty of Science**

Study programme: Bioinformatics

Branch of study: Bioinformatics


**Kateřina Břicháčková**

# Use of residue-level annotations for structural prediction of protein-ligand binding sites

# Využití anotací primární struktury pro strukturní predikci protein-ligand aktivních míst

Master thesis

Supervisor: RNDr. David Hoksza, Ph.D.

Prague, 2021

## Prohlášení

Prohlašuji, že jsem závěrečnou práci zpracovala samostatně a že jsem uvedla všechny použité informační zdroje a literaturu. Tato práce ani její podstatná část nebyla předložena k získání jiného nebo stejného akademického titulu.

V Praze, 18. 1. 2021 Kateřina Břicháčková

## Acknowledgement

The thesis is dedicated to Ondra Skácel - you are unforgettable. The biggest thanks go to my family, my partner David and dog Čenda.

I would like to thank to my supervisor, David Hoksza, for his guidance throughout writing this thesis, and for many helpful comments and advice. Many thanks to Radoslav Krivák and Petr Škoda for their kindliness and help with P2Rank program.

Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA LM2018140) provided within the program Projects of Large Research, Development and Innovations Infrastructures.

## Abstract

The number of experimentally resolved protein structures in the Protein Data Bank has been growing fast in the last 20 years, which motivates the development of many computational tools for protein-ligand binding sites prediction. Binding sites prediction from protein 3D structures has many important applications; it is a key step in the complex process of rational drug design, it helps to infer the side-effects of drugs, it provides insight into proteins biological functions and it is helpful in many other fields, such as protein-ligand docking and molecular dynamics. As far as we know, there has not been a study that would systematically investigate general properties of known ligand binding sites on a large scale. In this thesis, we examine these properties using existing experimental and predicted residue-level annotations of protein sequence and structure. We present an automated pipeline for statistical analysis of these annotations, based on hypothesis testing and effect size estimation. It is implemented in Python and it is easily extensible by user-defined annotations. The usage is demonstrated on 33 existing annotations and 4 different datasets. The practical significance of the results is tested with P2Rank prediction method. We are hopeful that the results as well as the pipeline could be eventually helpful for improving the performance of the existing binding sites predictors.

**Keywords:** binding sites, protein-ligand binding sites, binding sites prediction, P2Rank, residue-level annotations, 3D-based prediction, statistical analysis

# Abstrakt

V posledních 20 letech se počet experimentálních proteinových struktur v databázi Protein Data Bank rychle zvyšuje, což motivuje vývoj nástrojů pro predikci protein-ligand vazebných míst. Strukturní predikce vazebných míst má mnoho důležitých aplikací; je klíčovým krokem v komplexním procesu návrhu léčiv, pomáhá objevovat vedlejší účinky léčiv, umožňuje chápat biologické funkce proteinů a je využívá se i v mnoha jiných oborech, jako je protein-ligand docking nebo molekulová dynamika. Pokud je autorce známo, dosud nebyla provedena studie, která by systematicky zkoumala obecné vlastnosti známých vazebných míst na velkých datasetech. Tato práce se zaměřuje na analýzu těchto vlastností, s využitím existujících experimentálních i predikovaných anotací primární a terciální struktury proteinu. Je zde představena automatizovaná metoda pro statistickou analýzu těchto anotací, která je založena na testování hypotéz a odhadu velikosti účinku. Metoda je implementována v jazyce Python a lze ji jednoduše rozšířit o nové anotace definované uživatelem. Použití je demonstrováno na 33 existujících anotacích a čtyřech různých datasetech. Praktická významnost výsledků je otestována s pomocí metody P2Rank. Výsledky i samotná metoda pro statistickou analýzu by mohly posléze přispět ke zlepšení úspěšnosti existujících nástrojů pro predikci vazebných míst.

**Klíčová slova:** vazebná místa, protein-ligand vazebná místa, predikce vazebných míst, P2Rank, anotace primární struktury, 3D-based predikce, statistická analýza

# Contents


**1 Introduction** **3**

1.1 Thesis goals . . . . . 4

**2 Ligand binding sites prediction** **6**

2.1 Existing methods . . . . . 6

2.1.1 Geometry based approach . . . . . 6

2.1.2 Energy based approach . . . . . 7

2.1.3 Evolutionary approach . . . . . 7

2.1.4 Template based approach . . . . . 7

2.1.5 Machine learning approach . . . . . 7

2.1.6 Consensus approach . . . . . 8

2.2 Evaluation of success rates . . . . . 8

2.3 P2Rank . . . . . 9

**3 Methodology** **12**

3.1 Dataset file . . . . . 13

3.2 FASTA and PDB download . . . . . 14

3.3 Residue mappings . . . . . 14

3.4 Ligand binding sites . . . . . 15

3.5 Features . . . . . 15

3.5.1 UniProtKB . . . . . 17

3.5.2 PDBe-KB . . . . . 19

3.5.3 PDB . . . . . 21

3.5.4 FASTA . . . . . 23

3.5.5 Other resources . . . . . 24

3.5.6 User-defined features . . . . . 24

3.6 Statistical analysis . . . . . 25

3.6.1 Implementation . . . . . 26

3.6.2 Welch's test . . . . . 28

3.6.3 Chi-squared ($\chi^2$) test of independence . . . . . 29

3.6.4 Effect size . . . . . 31

3.7 P2Rank models training and evaluation . . . . . 32

**4 Evaluation and results** **34**

4.1 Datasets . . . . . 34

4.1.1 Ligands filtering . . . . . 35

4.2 Statistical analysis . . . . . 36

4.3 Statistical analysis - random sampling . . . . . 43
4.4 Discussion . . . . . 44
4.5 P2Rank models . . . . . 50
4.6 Experiment: comparison of conservation features . . . . . 55

**Conclusion** **59**

**List of Abbreviations** **60**

**Bibliography** **61**

**List of Figures** **72**

**List of Tables** **73**

**A Attachments** **74**
A.1 attachment1 . . . . . 74
A.2 attachment2 . . . . . 75

# 1. Introduction

Exploring the three-dimensional structure of a protein, as well as the quaternary structure of its complexes and interactions with other molecules, can help to understand the function of the protein. Thanks to ongoing efforts of structural biology, the number of experimentally resolved structures is growing rapidly. Within the last 20 years, the number of entries in the Protein Data Bank (PDB) [17] increased from 13 000 to 173 000. The number of structures released in 2020 is higher than the total number of PDB entries by the year 2000 [9]. This wealth of information motivates the development of various computational tools that can help to identify protein function.

Proteins can interact with other molecules, including nucleic acids, nucleotides, peptides, organic and inorganic compounds, metal ions or other proteins. These interactions have crucial role in many physiological processes and the proteins carry their function through the interactions [25]. Some interactions are unspecific, such as interactions with water and ions, other can be highly specific and have an important functional role. Binding to a protein binding site can be transient or persistent (e.g. metal ions). Binding often results in a change of conformation of the protein, causing essential changes in cellular function. Two important examples are enzyme-substrate complexes and receptor-ligands complexes which are crucial for signal transduction pathways [71].

The identification of protein's functionally important residues and defining binding sites locations is of a great importance. Nevertheless, many interactions have not been characterized experimentally and remain unknown. For this reason, plethora of computational methods for the prediction of binding sites and protein-protein interaction sites have been developed [71]. Various approaches have been proposed; they are described in Chapter 2 in more detail.

In this thesis, we focus on protein interactions with small compounds, generally referred to as *protein-ligand interactions*. For convenience we use terms 'ligand binding sites' or simply 'binding sites' to refer to the sites (set of interacting residues) on the protein structure where these small ligands bind. The interactions with small organic compounds are of particular interest, as they are essential for numerous cellular mechanisms, such as signalling or regulation of cell cycle [25].

Protein-ligand binding sites prediction has a very important application in rational drug design. The crucial part of the process that leads to a new drug design is to search for small drug-like molecules that are able to bind on particular proteins related to a disease [103]. Most of the currently used drugs are small

organic compounds [25]. The knowledge of binding sites is also important for prediction of off-target binding (molecule binds to a protein other than the primary drug target), possibly causing side-effects of a drug [99]. Prediction of binding sites has other applications in many fields, such as protein-ligand docking [83], inverse virtual screening [87], or molecular dynamics [39]. And finally, as mentioned before, protein-ligand binding gives valuable insight into understanding protein biological function.

## 1.1 Thesis goals

An interesting question is whether binding sites have some properties in common across different proteins and ligand types. There are some studies that explored the composition of binding sites on large scale [15, 41, 60], but those were focused on a few characteristics such as shape similarities or amino acid composition, rather than on the whole picture. Sequence and structural databases contain many valuable annotations that would be interesting to explore. As far as we know, there has not been a study that would take all suitable annotations of protein sequence and structure and perform an analysis to find out the general properties of know ligand binding sites. Knowing these properties could help to increase the success rates of existing binding sites predictors. In the thesis, we address this question and try to find the general properties of binding sites on large scale.

The aims of the thesis are following:

- To implement a pipeline for the statistical analysis of residue-level annotations. The pipeline should be designed to be easily extensible by new annotations defined by the user. It should be possible to run the statistical analysis with data supplied by the user, as well as to run the whole process, obtaining the data automatically.
- To use the pipeline to analyse chosen existing experimental and predicted residue-level annotations. Based on the results, to select the ligand binding sites properties which appear to have different values for binding and non-binding sites.
- To use existing method P2Rank [62] for prediction of ligand binding sites to test the practical significance of the results.

First, in Chapter 2 we describe existing approaches to protein-ligand binding sites prediction, introduce the P2Rank method in more detail and describe how the success rates are evaluated.

In Chapter 3 we present the analysis pipeline and its implementation, explain how the known binding sites (used as ground truth) were obtained, describe the annotations that were selected for experiments and introduce the statistical methods for evaluation of those annotations.

Finally, in Chapter 4 we go through the experiments and summarize and discuss the results. The descriptions of used datasets can be found there.

# 2. Ligand binding sites prediction

Great progress has been made in the field of the ligand binding sites prediction within the last 20 years. The existence of various projects and experiments, such as CASP (Critical Assessment of protein Structure Prediction) [64], CAMEO (Continuous Automated Model EvaluatiOn) [49] or CAFA (Critical Assessment of Function Annotation) [45], motivates the development of this field and provides standardized evaluation measures [102]. Plethora of tools based on different approaches has been developed, from simple geometry based algorithms to recently emerging deep learning based methods. Zhao *et al.* [102] published a review summarizing and categorizing more than 50 different methods.

The tools can vary in many aspects. Some of them are stand-alone command line tools running in fully automated manner, other are available as web-servers only, providing interactive interface. Some tools require some kind of pre-processing or supplying pre-calculated files, e.g. sequence alignments. The tools can differ in speed of prediction. In a survey comparing the running times of several methods published by Krivák and Hoksza [62], the differences of times required for prediction on a single protein were in order of magnitude. FPocket [48], the fastest of compared tools, can return prediction for one protein in 0.2 s, while getting prediction from the COACH web-server [100] would take 15 hours. The choice of a suitable tool depends on a use case: for processing large datasets, a fully automated and fast tool is required, whereas for inspecting several structures manually, when the speed is not the priority, a slower but more accurate method can be used.

In this Chapter, we summarize different strategies with specific examples of existing methods, we present P2Rank method in more detail and we explain how the performance of the predictors can be evaluated.

## 2.1 Existing methods

The existing methods for the protein-ligand binding sites prediction are based on various strategies and algorithmic approaches. It is possible to classify them into following caterogies; neverthless, keep in mind that many tools use a combination of the following approaches.

### 2.1.1 Geometry based approach

Most small ligands bind to concave cavities and pockets on the protein surface, as was observed in many studies of protein-ligand complexes in PDB [102]. The

geometric methods make use of this information and employ various techniques for calculating certain geometric measures from the protein structure. Some methods use additional properties such as polarity or charge, for instance FPocket [48] which calculates properties from the atoms in order to score each pocket. Other are purely geometrical, such as PocketPicker [97], a grid-based technique that uses a buriedness index to identify clusters of grid points where a ligand is likely to bind.

### 2.1.2 Energy based approach

Energy based methods predict the binding sites by calculating interaction energies between residues on the protein surface and a probe (for example methyl group in case of Q-SiteFinder [67]). The probe is placed on the grid points around the whole protein's surface and energetically most favourable locus is selected as putative binding site.

### 2.1.3 Evolutionary approach

These methods make use of conservation scores computed for each residue from the protein primary sequence. The conservation can be used to rescore the predicted sites (LIGSITE$^{CSC}$ [53]) or it can be used directly for the prediction (ConCavity [23]).

### 2.1.4 Template based approach

The template based methods make use of the fact that homologous proteins share similar folds and can bind ligands at similar positions. These methods rely on databases of protein-ligand complexes and use these complexes as templates to derive the predictions. Template based methods usually yield high confidence predictions; however, they are not able to discover novel binding sites [62]. There are two main types: structure template based methods (e.g. FINDSITE [90] which uses threading (fold recognition) algorithm) and sequence template based methods (e.g. S-SITE [100], using Needleman-Wunsch dynamic programming algorithm [78]).

### 2.1.5 Machine learning approach

Predicting protein-ligand binding sites can be seen as a classification problem. Many algorithm have been employed, for instance Suport Vector Machines (eFindSite [22]), Random Forests (P2Rank [62]), K-means clustering (ConCavity [23]),

multiple linear regression (SFCscore [91]) or artificial neural network (NNScore [40]). In the recent years, deep learning based prediction methods became popular in the field. Most of the methods use convolutional neural networks (CNN), e.g. DeepSite [56] or DeepCSeqSite [35].

### 2.1.6 Consensus approach

The consensus methods combine results of other existing methods and by this approach, they can achieve better results than the individual methods. For instance, MetaPocket 2.0 [52] combines eight different methods, taking top three sites from each of them. The resulting 24 sites, which can be spatially overlapping, are merged together by hierarchical clustering.

## 2.2 Evaluation of success rates

According to the way they represent prediction results, there are two types of predictors: *residue-centric* and *pocket-centric*. The residue-centric methods see the problem of binding sites prediction as binary classification. Each residue on the surface is classified as binding or non-binding. On the other hand, the pocket-centric methods do not classify individual residues; rather, they output a list of putative binding sites, usually represented as pocket centers, or sets of points around the protein surface, representing the shape of the binding site. The output list usually contains more predicted sites than the number of ligands and therefore, there are many false positives. Most methods rank the output list by the probability of being a true binding site, obtained by various ranking algorithms [62]. Ranking is important to prioritize the results, e.g. for visual inspection.

Residue-centric methods can be evaluated and compared by standard performance measures of binary classificators, such as accuracy, precision, recall, MCC (Matthews Correlation Coefficient) or AUC (Area Under the Curve) [69].

To evaluate the performance of pocket-centric ligand binding sites predictors, Chen *et al.* [25] proposed evaluation methodology with ligand-centric approach. The problem with previous protein-centric approach arises when a protein has more than one ligand. That approach requires only one correctly predicted binding site per protein to achieve 100% success rate, regardless of the number of ligands.

In the ligand-centric approach, to achieve 100% success rate we want a method to correctly identify binding sites for all ligands in the dataset. Every ligand is equally relevant for the final success rate. The output of a method is a ranked list

of putative sites. The list usually contains more sites than the number of ligands and therefore, many of them are false positives. For this reason, we take only the top-ranking sites into consideration for the evaluation. If all the predicted sites were considered, the most successful method would be the one returning so many binding sites they would cover the whole protein surface, which is certainly not desirable. The cutoffs Top-$n$ and Top-($n$+2) were used for the evaluation of P2Rank [62] ($n$ is the number of ligands). A binding site is considered correctly identified if at least one of the Top-$n$ or Top-($n$+2) binding sites passes a detection criterion (defined below). The success rate is then the number of correctly predicted sites divided by the total number of sites.

The position or shape of a predicted binding site does not have to match the true binding site precisely. Chen *et al.* [25] described several detection criteria to decide when the prediction is correct:

- $D_{CA}$ - a binding site is correctly predicted if the minimal distance between the center of the predicted site and any atom of the ligand is not greater than an arbitrary threshold D. The threshold D is usually 4 Å.
- $D_{CC}$ - a binding site is correctly predicted if the minimal distance between the center of the predicted site and the centrer of the ligand is not greater than D. This measure compensates for the size of the ligand (unlike $D_{CA}$, it does not give higher success rates to larger ligands).
- $O_{PL}$ - this measure expresses normalized spacial overlap between the predicted binding site and the ligand. It is defined as the volume of the intersection of the predicted site and the ligand, divided by the union of the two volumes. It is the most precise of the three measures, as it takes into account the spacial orientation. The disadvantage is that it can be computed only for the methods outputing the full set of grid points representing the site, instead of simply reporting the center of the site.

The first two criteria were used to evaluate the success rate of P2Rank method.

## 2.3 P2Rank

P2Rank [62] is a machine learning based ligand binding sites predictor developed by Radoslav Krivák and David Hoksza at Charles University. It is an open-source tool available as a stand-alone command line program or a web-server [55]. The command line tool is platform-independent, it is implemented in Java and Groovy and is very lightweight, as it does not depend on large structural or

sequence databases, or any other bioinformatics tools. It is especially suitable for large datasets, thanks to its speed which can be less than 1 s per protein.

P2Rank uses Random Forests classifier [20] to predict ligandability of specific regularly spaced points (*SAS points*, as described below) located around the surface of the protein, according to properties of their local neighborhoods. Random Forests is an ensemble learning method, combining a multitude of decision trees created by taking bootstrap samples from the training dataset and randomly selecting features for individual trees training.

The method takes a list of PDB files as input, and for each structure, outputs a ranked list of predicted binding sites. For each structure, the predictions are obtained following these steps: [61, 62]

1. Calculate protein's solvent-accessible surface [89] and generate a set of equally spaced points (called *SAS points*) along this surface. Each point will represent its local neighborhood. The default spacing is approximately 1.5 Å.

2. For each SAS point, calculate a feature vector based on its local chemical neighborhood. Some features are calculated directly on the SAS point (e.g. *protrusion*, a measure of point's 'buriedness'); other are projected from the nearby protein atoms, weighted by their distance. The features can be defined on the atomic level (e.g. B factor), or residue level (e.g. hydrophobicity). Altogether, the vector consists of 35 features.

3. The Random Forests classifier predicts ligandability scores for each SAS point, based on its feature vector. Instead of working with binary output (ligandable/nonligandable), the score ranges from 0 to 1 (1 = ligandable).

4. Filter out the SAS points with ligandability scores below certain threshold. Then apply single linkage clustering with 3 Å cutoff on the rest. Resulting clusters represent predicted binding sites. Only clusters with 3 or more points are considered.

5. Rank predicted binding sites to place the most promising ones to the top. The ranking score is defined as the sum of squared ligandability scores of all SAS points defining the site.

Figure 2.1 illustrates computed SAS points and predicted binding sites on an example structure.

In theory, any classifier could be used in step 3. Random Forests algorithm was chosen for several reasons. First of all, it has great generalization ability. Random

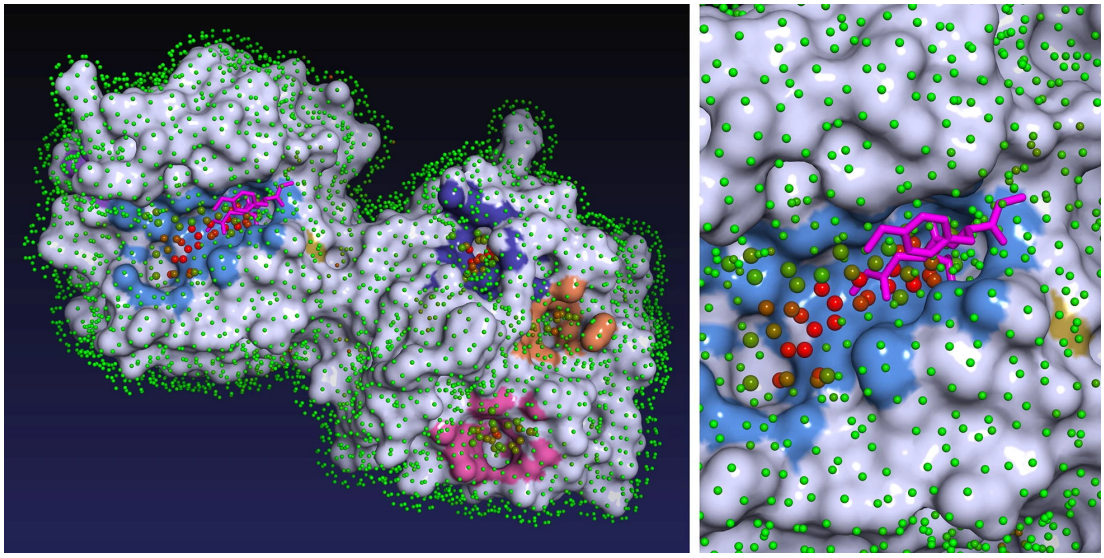


Figure 2.1: Visualization of SAS points for structure 1FBL. Each point is colored by its predited ligandability score (red points are the most ligandable, green points the least). Predicted binding sites are marked by coloured protein surface. Adapted from: *P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure* [62]

Forests are great when dealing with highly correlated variables [19] which is a very useful property for this application, as the feature vector contains a lot of related variables (e.g. various amino acid properties, such as hydrophobicity, polarity, aromaticity, charge etc.). It can cope with a large number of irrelevant variables, as well as binary and ordinal variables. Furthermore, it is relatively robust to outliers and noise, easily parallelized and simple to use, as it does not require prior scaling or other transformations of features. Also, it is faster than many other classifiers. And finally, it is able to report internal estimates of individual variable importances, which can give valuable insight into the problem [19, 20, 61].

P2Rank comes with a pre-trained optimized default model trained on Chen11 dataset [25], built with 100 trees, each grown with no depth limit using 6 features. In addition, a user can train and evaluate new models on his or her own datasets. This can be very useful for creating models specialized on certain types of proteins or ligands. The latest version of P2Rank also allows to add a custom feature to the feature vector without changing P2Rank source code. This possibility was employed for the experiments in this thesis. The feature values are supplied in CSV files (one file per structure) where one row describes one residue (or atom).

More details on setup, requirements, usage examples, models training and evaluation and adding new features can be found in the project's repository (`https://github.com/rdk/p2rank`) or in the tutorial available at `https://github.com/cusbg/p2rank-framework`.

# 3. Methodology

One of the main aims of the thesis was to develop a pipeline for statistical analysis of available protein structure annotations (hereinafter referred to as features), and to prepare this pipeline for adding user-defined features.

The process starts with downloading FASTA and PDB files for input proteins from databases. Residue-level mappings are downloaded as well, to allow cross-referencing the protein tertiary structure with the sequence annotations. After that, values for all features are computed or downloaded and assigned to each residue. Residues are labeled as binding or non-binding according to the ligands defined in the PDB file. As the next step, we perform statistical analysis of the features, using computed ligand binding sites labels and feature values. After examining the results, we can decide which features could be potentially interesting for the ligand binding sites prediction. Finally, we can train new P2Rank models with these new features and see if the performance has improved.

The analysis pipeline covers most steps of the process, from downloading the files from databases, to computing the statistical analysis of individual features. The only needed input is a dataset file with listed protein identifiers. Moreover, there are two scripts that further extend the pipeline and can be used to train and evaluate P2Rank models with new features. The structure of the pipeline is depicted in Diagram 3.1. The details about individual parts are described in this Chapter.

The pipeline is implemented in Python, utilizing several Python packages, such as BioPython [29], NumPy [44] or SciPy [47]. BioPython is an open-source collection of Python tools for computational biology and it was very useful in this work, especially for parsing PDB and FASTA files.

The pipeline comprises a set of Python scripts which are connected together by the main script `analysis_pipeline.py`. The main script should be used to run the pipeline. It defines the user API, parses and checks arguments, takes care of logging and runs individual parts of the pipeline. See `https://github.com/katebrich/LBS_analysis_pipeline` for more details about options, examples of usage, setup, requirements, input and output.

The features can be defined in 'config file'. It is a file in JSON format that lists names of features, their type (binary, categorical, ordinal or continuous) and a path to the class with implementation. Custom config file with user-defined features or with subsets of features can be created and passed as argument `-c new_config_path`.

TODO vic popsat usage? Nebo usage a setup v attachments? Příklady?

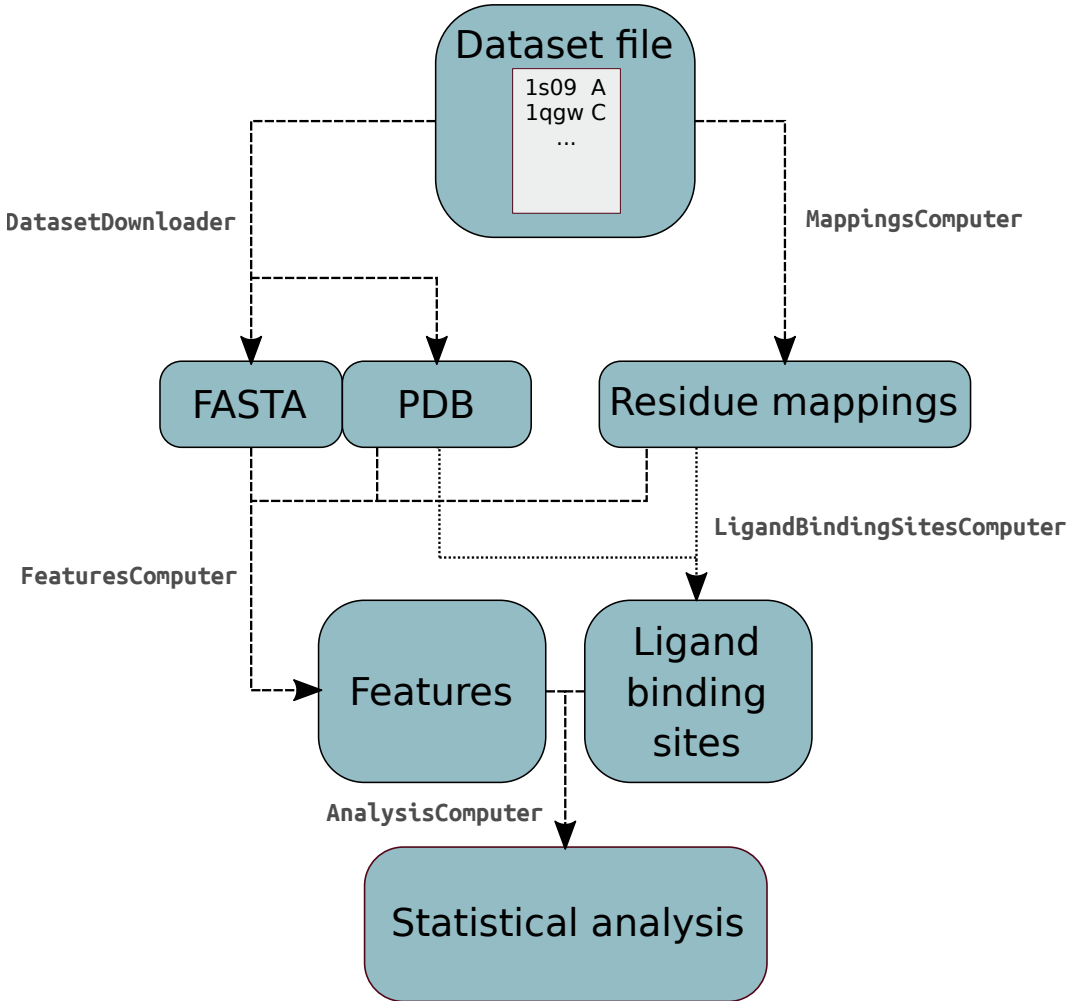



Figure 3.1: Diagram of the pipeline structure.

The information about versions of used software and databases can be found in Appendix TODO.

## 3.1 Dataset file

Dataset file is a mandatory input for the pipeline. It is a plain-text file with each row representing one structure. It has several columns separated by whitespace. The first two columns are mandatory and they contain PDB ID and chain ID (pipeline can only work with single-chain structures). The third column is optional and it can define a comma-separated list of ligands which will be used for the ligand binding sites computation.

## 3.2 FASTA and PDB download

For each structure, FASTA and PDB files are downloaded from PDBe, via Entry-based REST API [10], which is one of the possibilities to access large amouts of data about individual PDB entries programatically.

## 3.3 Residue mappings

The residue-level mappings are needed for cross-referencing the protein tertiary structure with the sequence annotations and UniProt records. The main reason is that the PDB entry may cover only a segment of the full-length protein and the segment does not have to be continuous.

Furthermore, the numbering of residues in the PDB file can differ from the primary sequence numbering. The residues in the PDB file are assigned identifiers by the author, in order to match the identification used in the publication. The identifier of the residue is composed of two parts. The first one is a residue number, and the second one, called 'insertion code', is usually a character and it is left empty for most residues. Typically, it is used to label insertions relative to the reference sequence. The author can assign the numbers how he or she desires; they do not have to start with one or zero, do not have to be in consecutive order and can even be negative.

For those reasons, the residue-level mappings are downloaded from PDBe REST API [10] early in the pipeline process, they are cached in files and can be used whenever sequence annotations need to be mapped on the structure described in the PDB file.

For the cross-referencing between protein structures and protein sequences in UniProtKB [94], we used UniProt segments mapping implemented in PDBe REST API. The implementation is based on SIFTS [37], a resource for the transfer of annotations between protein structure and sequence. The mappings are assigned by the SIFTS process with the UniProt sequence as reference and thus, the output is a set of segments that reflects discontinuities in the UniProt sequence.

There is a presumable issue with UniProt segments - sometimes the returned segment does not have the same length in UniProt coordinates and PDB coordinates. We reported it and it was recognized as a bug, but unfortunately has not been corrected before running the experiments in this work; however, it is very rare and it concerns only a few structures; those were removed from the datasets.

## 3.4 Ligand binding sites

Each residue is assigned a label *non-binding/binding* (0/1) according to positions of ligands in the PDB file. A residue is labeled as *binding* if it has at least one non-hydrogen surface atom within distance 4.0 Å of a non-hydrogen atom of any ligand. The distance 4.0 Å can be changed in the pipeline with `-l` argument. This ligand-based definition of binding sites was used in previous large-scale study exploring the composition of binding sites [60].

Only residues on the surface of the protein were taken into consideration in the analysis. The non-surface residues cannot be binding anyway, and excluding them decreases the imbalance between binding and non-binding residues counts. Furthermore, excluding the inner residues helps to reduce potential influence of difference of feature values in surface vs. non-surface residues. For example, inner residues tend to be more hydrophobic in general. Thus, binding sites could seem to be more hydrophilic than non-binding sites, but it would not be clear whether it is not simply the effect of being on the surface of the protein.

To decide which residues are located on the surface, solvent-accessible surface area of each residue was computed. We defined the surface residues as residues that have less than 5% of their surface accessible to the solvent. This cut-off was proposed by Miller *et al.* [75] and used in other studies[57, 60].

The solvent-accessible surface was computed using `Bio.PDB.SASA` module in BioPython [11]. It implements Shrake & Rupley algorithm [89] which uses a sphere of particular radius to probe the surface of the protein. It can be imagined as 'rolling a ball' along the surface (see Figure 3.2). The smaller the sphere radius, the more surface details it can detect. For this work, we used the default radius of 1.4 Å, which approximates the radius of a water molecule.

## 3.5 Features

This section describes all implemented features and provides information on how to add user-defined features. The features names and types are summarized in Table 3.1

The pipeline can run only with a subset of implemented features, by listing them and passing as argument (e.g. `-f hydropathy,aromaticity`). If argument `-f` is not stated, all features defined in the config file are computed.

The individual features are described in detail in following sections, categorized by the resource that was used for their retrieval.

| Name | Type | Source |
|---|---|---|
| PTM | binary | UniProtKB |
| lipidation | binary | UniProtKB |
| glycosylation | binary | UniProtKB |
| mod_res | binary | UniProtKB |
| disulfid | binary | UniProtKB |
| non_standard | binary | UniProtKB |
| sec_str | categorical | UniProtKB |
| helix | binary | UniProtKB |
| turn | binary | UniProtKB |
| strand | binary | UniProtKB |
| natural_variant | binary | UniProtKB |
| variation | binary | UniProtKB |
| compbias | binary | UniProtKB |
| pdbekb_conservation | ordinal | PDBe-KB |
| dynamine | continuous | PDBe-KB |
| efoldmine | continuous | PDBe-KB |
| depth | continuous | PDBe-KB |
| bfactor | continuous | PDBe-KB |
| exposure_CN | continuous | PDB |
| HSE_up | continuous | PDB |
| HSE_down | continuous | PDB |
| phi_angle | continuous | PDBe |
| psi_angle | continuous | PDBe |
| cis_peptide | binary | PDBe |
| aa | categorical | FASTA |
| hydropathy | ordinal | FASTA |
| mol_weight | ordinal | FASTA |
| polarity | categorical | FASTA |
| charge | binary | FASTA |
| aromaticity | binary | FASTA |
| H_bond_atoms | ordinal | FASTA |
| mobiDB | continuous | MobiDB |
| conservation | continuous | P2Rank |

Table 3.1: Summary of analysed features.

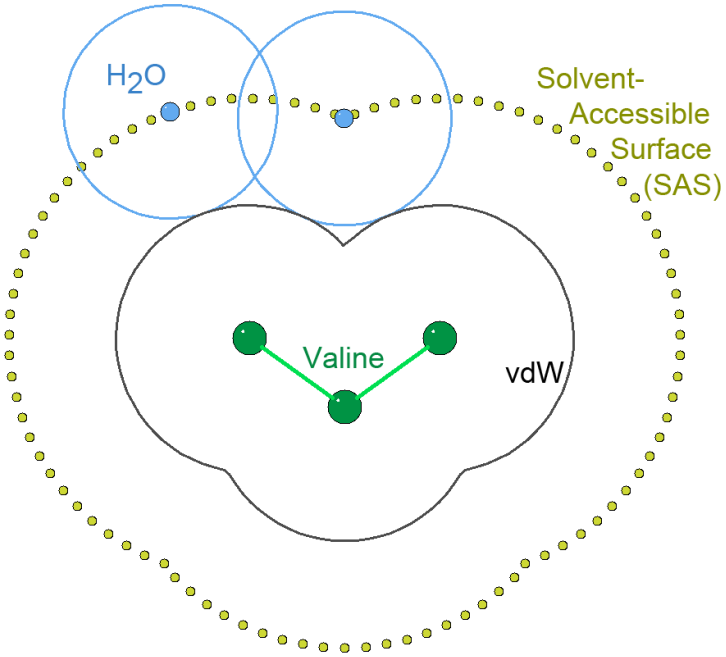



Figure 3.2: Illustration of the solvent accessible surface. It was created by rolling the probe (in blue) along the molecule surface and tracing the center of the probe. Retrieved 02-01-2020 from `https://commons.wikimedia.org/wiki/File:Surfacetype_Solvent-Accessible.png`

### 3.5.1 UniProtKB

The UniProt Knowledgebase (UniProtKB) is a large database of well-annotated protein sequence data. It tries to achieve the minimal redundancy of proteomes and it provides detailed, accurate and consistent annotations of the sequences [94].

Sequence annotations (called 'features') are available for every UniProtKB entry. They describe interesting sites and regions on the protein sequence and every feature has an associated description with available evidence, source and related publications. The features are arranged in a well-organized manner on the UniProt website [13], in so called 'Features viewer' with many overlapping tracks for different features. Nonetheless, for the purpose of this work, the best way to obtain the features was via the Proteins REST API [79]. It provides the interface to access the sequence annotation data as well as mapped variation data programmatically. The API is available at (`http://www.ebi.ac.uk/proteins/api/doc`).

Features are classified into eight categories which are further subdivided into types. For example, the category 'STRUCTURAL' comprises the types 'HELIX', 'TURN' and 'STRAND'.

The types and categories that were chosen as potentially relevant for ligand binding sites prediction are described below.

#### 3.5.1.1 PTM

Post-translational modifications are covalent chemical modifications of polypeptide chains after translation, usually modifying the functional group of the standard amino acids, or introducing a new group. They extend the set of the 20 standard amino acids and they can be important for the function of many proteins,as they can alter the interactions with other proteins, localization, activity, signal transduction, cell-cell interactions and other properties. Their enrichment in binding sites is very interesting to examine.

Three UniProtKB feature types were analysed: lipidation, glycosylation and type 'MOD_RES' which comprises phosphorylation, methylation, acetylation, amidation, formation of pyrrolidone carboxylic acid, isomerization, hydroxylation, sulfation, flavin-binding, cysteine oxidation and nitrosylation. Only experimentally determined modification sites are annotated, and they are further propagated to related orthologs when specific criteria are met [6].

Since lipidaton and glycosylation data were very sparse (e.g. there were only 15 lipidation sites in the whole holo4k dataset composed of 3973 proteins), the fourth feature called 'PTM' including all three types was added to the analysis.

#### 3.5.1.2 Disulfide bonds

Another type of post-translational modifications are disulfide bonds formed between two cysteine residues. Both intrachain and interchain bonds are annotated by UniProtKB. The disulfide bonds may be either experimentally determined or predicted (occuring in specific protein families) [4].

#### 3.5.1.3 Non-standard residues

Describes the occurence of non-standard amino acids (selenocysteine and pyrrolysine). There must be experimental evidence for this occurence; however, it can be propagated to close homologs [8].

#### 3.5.1.4 Secondary structure

This feature category annotates three types of secondary structures: helices, beta sheets and hydrogen-bonded turns. Residues not belonging to any of the classes are in a random-coil structure. The 'helix' class comprises alpha-helices, pi-helices and $3_{10}$ helices.

The secondary structure assignment is made by DSSP algorithm [58] based on the coordinate data sets extracted from the Protein Data Bank (PDB). They are neither predicted computationally, nor propagated to related species [12].

#### 3.5.1.5 Natural variant

This feature includes naturally occuring polymorphisms, variations between strains or RNA editing events [7].

#### 3.5.1.6 Variation

*Variation service* is a utility that can retrieve variation data from UniProtKB. The variants are either extracted from the scientific literature and manually reviewed, or mapped from large scale studies, such as 1000 Genomes [33], COSMIC [46], ClinVar [66] or ExAC [59]. The Proteins REST API provides various options for variants retrieval, such as to filter by the consequence type, associated disease name, cross reference database type (e.g. ClinVar) or by the source type [79].

#### 3.5.1.7 Compositional bias

The regions of compositional bias are parts of the polypeptide chain where some of the amino acids are over-represented, not following the standard frequencies. The regions can be enriched in one or more different amino acids [2].

### 3.5.2 PDBe-KB

PDBe-KB (Protein Data Bank in Europe - Knowledge Base) is managed by the PDBe team at the European Bioinformatics Institute. It is a collaborative resource that aims to bring together the annotations from various sources and to show the macromolecular structures in broader biological context.

One drawback of PDB is that every page represents only one entry that is based on a single experiment. There may be several PDB entries for the full-length protein, each covering only a segment of it. Nevertheless, the entries for the same protein are not interconnected. PDBe-KB has developed the *aggregated views of proteins*, displaying an overview of all the data related to the full-length protein defined by the UniProtKB accession.

The structures from the PDB are extensively used by scientific software and other resources. There exist many valuable annotations, such as ligand binding sites, post-translational modification sites, molecular channels or effects of mutations, that are created outside of the PDB. The problem is that the data

is fragmented and therefore it would require immense effort of a researcher to collect and make use of all available data for a structure of interest.

The aggregated views of proteins integrates the annotations from *PDBe-KB partners*, collaborating scientific software developers. It facilitates the retrieval of these annotations with a uniform data access mechanism (via FTP or REST API). The project is called 'FunPDBe'. A common data exchange scheme was defined to facilitate the transfer of data. [32]

The use of PDBe-KB was difficult because of the lack of documentation and a few bugs that were encountered during this work (some of them corrected by now after pointing them out). However, it is understandable since it was launched only two years ago and the constant improvements are done since then.

#### 3.5.2.1 Conservation

PDBe-KB provides pre-calculated residue-level conservation scores, obtained by a pipeline using HMMER and Skylign web servers that was described by Jakubec et al. [54].

The values of the score are integers ranging from 0 to 9, with 9 being the most conserved. Since scores higher than 4 were very sparse and the feature would not meet the assumptions of the Chi-squared test, the scores 4 and higher were merged into one category (4). This does not deteriorate the prediction nor the hypothesis test, as vast majority (over 95%) of non-binding residues were scored 1 and lower.

#### 3.5.2.2 DynaMine

DynaMine [26] was developed by the Bio2Byte group [1] and it is one of the PDBe-KB partner resources. It provides the annotations of the backbone dynamics predicted only from the FASTA sequence. DynaMine predicts backbone flexibility at the residue-level, using a linear regression model trained on a large dataset of curated NMR chemical shifts extracted from the Biological Magnetic Resonance Data Bank [95]. The predictor estimates the value of the 'order parameter' ($S^2$) which is related to the rotational freedom of the N-H bond vector of the backbone. The values range from 0 (highly dynamic) to 1 (complete order).

#### 3.5.2.3 EFoldMine

EFoldMine [81] tool comes from the same group as DynaMine. It is a predictor of the early folding regions of proteins. It makes predictions at the residue-level derived only from the FASTA sequence. Internally it uses dynamics predictions and secondary structure propensities as features and the linear regression model

is trained on data from NMR pulsed labelling experiments. Unfortunatelly, the early stages of protein folding are not understood very well so far and experimental data is very difficult to obtain. The predictor was trained on the dataset of only 30 proteins and its performance is quite poor.

#### 3.5.2.4 Depth

Depth [93] is a webserver that can measure residue burial within the protein. It is able to find small cavities in proteins and could be used as a ligand-binding sites predictor as such. The residue depth values are computed from the input PDB file.

The algorithm places input 3D structure in the box of model water, each residue with at least two hydration shells around itself. The water molecules in cavities are removed: the algorithm removes the water molecule if there are less than a given number of water molecules in its spherical volume of given size. The minimum number of neighbouring molecules and the spherical volume can be defined by the user. The removal is iterated until there are no more cavity waters. Residue depth is then computed as the distance to the closest water molecule.

### 3.5.3 PDB

The following features can be computed or obtained directly from the PDB file.

#### 3.5.3.1 B factor

The B factor, also called the Debye-Waller factor or the temperature factor, describes "the attenuation of X-ray or neutron scattering caused by thermal motion" [92]. It can be used to identify and interpret flexibility of proteins, supposing that high B factors are indicators of higher flexibility, whereas atoms with low B factors generally belong to the well-ordered parts of the structure. B factors can be also view as indicators of the relative vibrational motion of atoms in a protein [92].

The values can be obtained directly from the PDB files: each ATOM record of a X-ray structure (except for hydrogens) deposited in PDB contains B factor value for the atom. B factor for a residue was computed be averaging B factors of all its atoms.

#### 3.5.3.2 Contact number exposure

Contact number (CN) is a simple solvent exposure measure that can be computed directly from the 3D structure. The CN value for residue is number of C$\alpha$ atoms within a sphere of chosen radius around the C$\alpha$ of that residue [80].

The implementation in BioPython module `Bio.PDB.HSExposure` was used for computation, with default sphere radius 12 Å.

#### 3.5.3.3 Half sphere exposure

Half sphere exposure (HSE) is a solvent exposure measure introduced by Hamelryck (2005) [50]. The CN sphere (defined above) around the C$\alpha$ atom is split in two halves by the plane perpendicular to the C$\alpha$-C$\beta$ vector, going through the C$\alpha$, as illustrated in Figure 3.3. Two different measures are obtained, HSE-up, which is number of C$\alpha$ in 'upper' half sphere (containing C$\beta$), and HSE-down, number of C$\alpha$ in the opposite sphere.

Class `HSExposureCB` from BioPython module `Bio.PDB.HSExposure` with default sphere radius 12 Å was used.

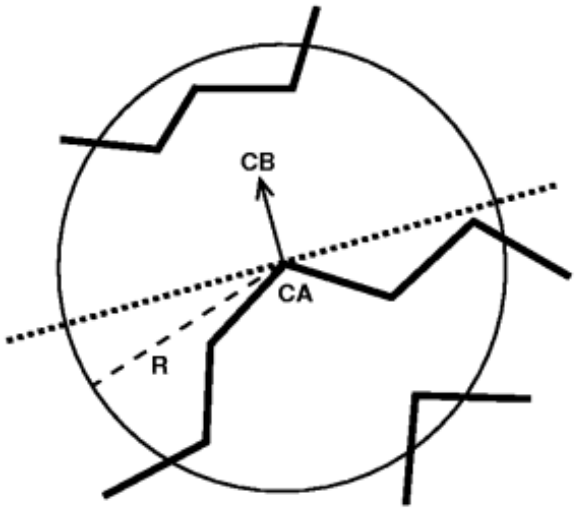



Figure 3.3: Half sphere exposure. Retrieved 02-01-2020 from `https://en.wikipedia.org/wiki/File:HSECa.png`

#### 3.5.3.4 Phi and psi angles

Three dihedral angles of a popypeptide backbone phi ($\phi$), psi ($\psi$) and omega ($\omega$) are depicted on the Figure 3.4. While the $\omega$ angle is restricted due to the planar character of the peptide bond, the $\phi$ and $\psi$ angles have high rotational freedom around the N-C$\alpha$ ($\phi$ torsion) or C$\alpha$-C ($\psi$ torsion) bonds. The Ramachandran

plot provides good visualization of the whole $\phi$
$\psi$ space [82]. The angles sizes can be computed directly from the PDBe; for the purpose of this work, they were obtained from PDBe via REST API.

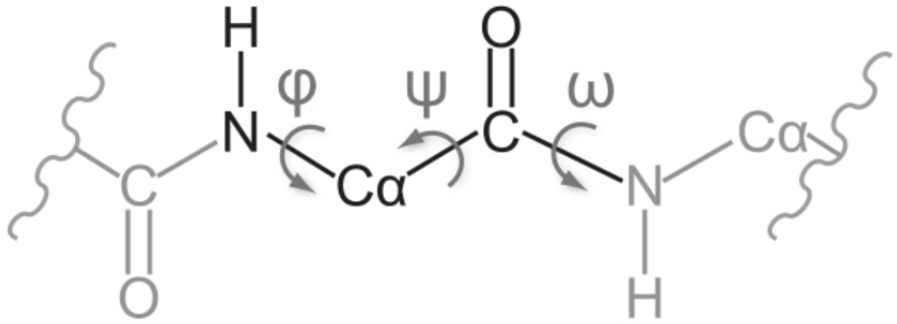



Figure 3.4: Polypeptide torsion angles phi, psi and omega. Retrieved 02-01-2020 from `https://www.researchgate.net/figure/Backbone-torsion-angles-of-a-prototypical-amino-acid-building-block-embedded-i fig2_284713304`

#### 3.5.3.5 Cis peptide

The majority of protein bonds is found with torsion angle $\omega$ close to 180°, in so-called *trans* conformation. The *cis* isomer, having $\omega$ close to 0°, is rather rare. The *cis-trans* isomeration is involved in some biological processes, such as protein folding or membrane binding [34].

The residues with *trans* bond are obtained from PDBe via REST API.

### 3.5.4 FASTA

There are features that can be derived directly from the FASTA sequence. Every amino acid is assigned a value and the feature values are obtained according to the FASTA file. These features are:

- **Amino acid** - Categorical feature which is simply the amino acid letter.
- **Hydropathy** - The values of hydropathy index proposed by J. Kyte and R. F. Doolittle [65]. It takes into consideration hydrophilic and hydrophobic properties of the 20 amino acid side chains. It is based on experimental observations derived from the literature. It ranges from -4.5 (Arg) to 4.5 (Ile) and the larger the number is, the more hydrophobic the amino acid.
- **Molecular weight** - Residue mass in Daltons.

- **Polarity** - Classification of amino acids according to the side chain - categories Polar, Nonpolar and Polar uncharged.
- **Charge** - Binary feature indicating whether the side group is charged in physiological pH.
- **Aromaticity** - Binary feature labeling residues that contain aromatic ring.
- **Hydrogen bond atoms** - Number of atoms of the side chain that are either hydrogen donor or hydrogen acceptor.

The biochemical properties of amino acids (all features above except hydropathy) were obtained from TODO citovat biochemie voet

### 3.5.5 Other resources

#### 3.5.5.1 MobiDB

MobiDB is a database of protein disorder and mobility annotations. It provides annotations and predictions for intrinsically disordered (ID) proteins. MobiDB-lite is a method for highly specific predictions of long (at least 20 residues) disorders. It is a consensus-based prediction, combining results of eight different predictors [77]. It has been integrated into the MobiDB and the web server provides programmatic access to retrieve single entries via REST API [5].

#### 3.5.5.2 P2Rank

The sequence conservation scores for the feature `conservation` are computed through Conservation pipeline [3] implemented for P2Rank. The conservation scores are computed from multiple sequence alignment (MSA) using the Jensen-Shannon divergence model [**?** ]. The pipeline searches for sequences similar to the query sequence in several databases. The details of the implementation were described by Jendele *et al.* [55]. The pipeline runs on local computer and needs to have SwissProt, UniRef90 and TrEMBL databases downloaded locally.

### 3.5.6 User-defined features

There are two ways to run the statistical analysis with user-defined features. The first, more time-demanding way is to compute all requited inputs (i.e. feature values and ligand binding sites labels) outside of the pipeline, and then run only the analysis task (with argument `-tasks A`). More details about the usage and a few examples are described in the README file in the GitHub repository of the project (`https://github.com/katebrich/LBS_analysis_pipeline`).

Another, more straightforward possibility is to implement the new feature directly inside the pipeline. Two steps need to be made:

- to add the feature to the config file. The class with feature implementation is loaded dynamically according to the feature name, for the easier definition of new features,
- to implement class with method `get_values` which computes the feature values and returns them in the required format.

## 3.6 Statistical analysis

To find the features which are possibly important for prediction of protein-ligand binding sites, statistical analysis has the crucial role. This section describes the method that was used to analyse the statistical significance of the features and to distinguish the ones that stand out in the known protein-ligand binding sites.

In this work, the problem is seen as a hypothesis testing problem. Two populations will be compared: we take values of a feature for all the residues across all the proteins in the dataset and then compare the values associated with the binding residues and non-binding residues.

The *null hypothesis* and the *alternative hypothesis*, denoted by $H_0$ and $H_1$, respectively, will be tested:

- $H_0$ - The feature values in binding sites do not significantly differ from the values in non-binding sites.
- $H_1$ - There is a significant difference of feature values in binding sites and non-binding sites.

To decide which one of two complementary hypotheses is true, we employ a suitable *hypothesis test*. Welch's test and Chi-squared test of indepence, both described below in more detail, will be used according to the feature type (binary, categorical or continuous).

As one may expect, the tests are not error-proof and a mistake can be made in the decision of whether to accept or reject the null hypothesis. There are two types of errors in hypothesis testing, commonly known as *Type I error* and *Type II error*. The test has made a Type I error if it incorrectly rejects a true null hypothesis. If, on the other hand, a null hypothesis is accepted and it is not true, a Type II error has been made. Both situations are depicted in the Table 3.2. The ideal test would have both error probabilities equal to zero. Nevertheless, in

| | | Prediction: Accept $H_0$ | Prediction: Reject $H_0$ |
|---|---|---|---|
| Truth | $H_0$ | Correct (true positive) | **Type I error** (false positive) |
| | $H_1$ | **Type II error** (false negative) | Correct (true negative) |

Table 3.2: Type I and II Error in hypothesis testing.

most cases it is not possible to make both error probabilities arbitrarily small for a fixed sample size [24].

To control statistical significance of the result, we define a *significance level*, a constant denoted by $\alpha$. It represents the probability of making a Type I error, in other words, the probability that the study rejects the null hypothesis when it is true. The typical choices in practice are $\alpha = 0.01$, 0.05 or 0.10 [24]. One should be aware that by fixing the significance level of the test, the experimenter is controlling only the Type I error probabilities. The probability of the Type II error is subject to factors such as the accuracy and completeness of the data and most importantly, the true effect size [88]. Our choice will be $\alpha = 0.05$.

The *P value* is reported as a result of the statistical test. The P value is the probability that, under the assumption the null hypothesis is true, we observe the same or greater difference between groups. Smaller values of $p(X)$ give stronger evidence for rejecting the null hypothesis. The null hypothesis is rejected when $p(X) \leq \alpha$. P value gives an idea of how strongly the data contradict the null hypothesis; furthermore, it allows other researchers to make a decision according to the significance level of their choice [14, 43, 88].

### 3.6.1 Implementation

The following sections rationalize the choice of Welch's and chi-squared test. The imlementation of these tests in module `scipy.stats` from the SciPy Python library [47] was used in the pipeline; namely `ttest_ind` with `equal_var=False` to perform Welch's test, and `chi2_contingency` for chi-squared test of independence.

By default, the pipeline computes the analysis for all the residues across all

proteins in dataset. Neverthless, random sampling (without repetition) can be performed by specifying the sample size with argument `-s`. It is also possible to run more iterations of random sampling (argument `-i`). In this case, mean P values will be reported in summary. Individual P values from all iterations will be reported in separate files for each feature. Another possibility is to balance the number of binding and non-binding sites with argument `-b`. Same number of binding and non-binding residues will be sampled in that case.

The significance level is 0.05 by default and can be changed with `-a` argument.

The output of the whole analysis pipeline are folders with results for each feature, as well as several summary files:

- **p_values_means.csv** - Averaged P values obtained from all iterations.
- **p_values.csv** - List of P values from all iterations for all features. P values for individual features are also included in the results folder for each feature.
- **cohens_d.csv** - Effect sizes for continuous features (as described below).
- **cohens_w.csv** - Effect sizes for binary, ordinal and categorical features.
- **p_vals_perc.csv** - Summary of how many percent of iterations had P value below given significance level $\alpha$ for each feature.
- **means_difference.csv** - Summary only for continuous features. Lists the differences of means and variances in both populations (all binding vs. all non-binding).
- **binding_ratios.csv** - binding/non-binding sites ratios for all rows in the dataset.
- **errors.txt** - Lists features that ended up with an error. This is often caused by lack of data (e.g. the sample size is bigger than number of rows or the data for a categorical feature are too sparse to meet the assumptions of Chi-squared test). Detailed information about the errors can be found in the log file.

The results folder for each feature contain file `pairs.txt` with paired ligand binding sites values and feature values, detailed information about all iterations, and various histograms and plots (according to the type of feature).

### 3.6.2 Welch's test

Welch's unequal variances t-test, or Welch's test in short, is a two-sample hypothesis test used to decide whether two populations have different central tendencies (means or medians). The decision is made based on the samples from the two populations. It is a more robust alteration of the widely-used Student's t-test [98].

Both Student's and Welch's t-test assume that the two examined populations follow a normal distribution [98]. Nevertheless, when testing for the equality of means of 'large enough samples', the normality assumption can be violated thanks to the large sample theory and the Central Limit Theorem [43]. It has been shown in previous studies that for large samples, the statistical significance level is protected not only for normally distributed data, but also for many non-normal distributions; moreover, in case of Welch's test, this is true even for unequal variances [72, 106, 107]. According to Lehmann and Romano [43], the Type II error is also relatively insensitive to non-normality. Many articles and textbooks mention that when the sample sizes are small, nonparametric tests (i.e. tests that do not assume a specific distribution) such as the Mann-Whitney test [73] should be considered as an alternative to t-tests. However, t-tests become superior when sample sizes increase [72, 104]. The simulations made by Lumley *et al.* [72] show that 'sufficiently large sample size' means under 100 in most cases. Even for extremely non-normal data, the sufficient size is at most 500. This suggests that the choice of Welch's test is legitimate for this work.

The problem of the Student's t-test is that it performs badly when the variances of the two compared populations are unequal. Both Type I and Type II errors are negatively affected by violation of the equal variances assumption. The unequal variances can be less problematic if sample sizes are similar, but in practice, that is not always the case [85].

Unlike Student's t-test, Welch's test does not assume equal variances of the populations. It performs well when the samples have unequal variances; furthermore, it can be used even when the samples have unequal sizes [38].

Some researchers tend to pre-test for variance equality by a preliminary test of variances (such as Levene's [68] or Brown-Forsythe test [21]) and then choose whether to use Student's or Welch's t-test. However, although this approach persists in some textbooks and software packages, it is not recommended by statisticians. As a preliminary test itself is subject to Type I and II errors, this two-stage procedure would not protect the significance level and could lead to incorrect decisions. One should be aware of the fact that even if the test suggested that the samples variances are nearly equal, it would not mean that

the whole population variances could not differ to a larger extent [105]. Some researchers may try to make the significance level of a preliminary test more strict, so that they could be more confident about the choice of the subsequent test; however, as the significance level decreases, the performance of the compound test paradoxically gets worse. According to Zimmerman [105], "a higher Type I error rate of the preliminary test actually improves the performance of the compound test" [105]. This suggests that using the preliminary test is not correct in principle.

Welch's test should be used whenever the researcher is not sure that the variances are truly equal. Ruxton [85] even suggests the routine use of Welch's test. When the sample sizes and variances are equal, both tests perform similarly. When dealing with unequal variances and unequal sample sizes, Welch's test is more robust than Student's t-test and the Type I error rate does not deviate far from the nominal value [38]. Hence, Welch's test can be applied without any significant disadvantages to Student's t-test.

For all the reasons stated above, Welch's test seems to be the best choice for the purpose of this study. It has the best combination of performance and ease of use, the calculation is straightforward and it is available in commonly used statistics packages. This test will be used for continuous features.

### 3.6.3 Chi-squared ($\chi^2$) test of independence

A different kind of tests will be needed for the analysis of categorical and binary features. In this section, the $\chi^2$ test will be compared to another well-known test for the analysis of data in contingency tables, the Fisher's exact test.

A *contigency table* is a table displayed in a form of a matrix where cells represent a frequency distribution of samples in the categories. An example of a contingency table can be seen in Table 3.3. The sums of frequencies in rows and columns are called *marginal totals*.

| | Aromatic residue | Non-aromatic residue | Total |
|---|---|---|---|
| Binding sites | 1016 | 4654 | 5670 |
| Non-binding sites | 4829 | 44545 | 49374 |
| Total | 5845 | 49199 | 55044 |

Table 3.3: A $2 \times 2$ contingency table for binary feature `aromaticity` computed on dataset Chen11.

The null hypothesis assumes independence of the groups; in our case, the assumption is that there is no difference in the proportions of the analysed feature between binding sites and non-binding sites.

Fisher's exact test belongs to a class of so-called *exact tests*; it means that the P value is calculated accurately, not approximately, as is the case of many tests including Welch's test and $\chi^2$ test. Fisher's test is mostly used for $2 \times 2$ contigency tables, although the principle of the computation can be extended to a general $m \times n$ table [74]. The principle of the test lies in computing the probability of obtaining a table that is more or equally extreme in the departure from the null hypothesis than the analysed table and has identical marginal totals [18].

Chi-squared test if independence is able to decide whether the difference between the observed frequencies and the 'expected frequencies' is statistically significant. The expected frequencies are computed for every cell using this formula:

$$\frac{row\ total \times column\ total}{grand\ total}$$

It can be imagined as the average frequencies we would get in the long run with the same marginal totals, assuming the null hypothesis is true (i.e. there is no association between groups). The result of the test tells how likely are we to observe given data under the assumption of the true null hypothesis [18].

The biggest difference between the two mentioned tests is that the chi-squared test is based on a aproximation approach; therefore, it needs a 'large enough' sample. W. G. Cochran (1952, 1954) proposed a set of recommendations about the minimum expectations to be used in $\chi^2$ tests and about the choice between Fisher's test and $\chi^2$ test:

- **The 2 x 2 table** - Fisher's exact test should be used whenever the sample size is smaller than 20, or when the sample size is smaller than 40 and if the expected frequency in at least one cell is less than 5. For sample sizes bigger than 40, always use chi-squared test [27, 28].
- **More than 1 degree of freedom** - Chi-squared test can be used when at most 20% of cells have the expected frequency less than 5 and no cell have the expected frequencies less than 1 [28].

These recommendations are presented in several textbooks and articles as a rule of thumb [63] and recommended to be used in practice.

As the sample sizes in this work are very large, the number of binding and non-binding sites is unbalanced, and the data for some features can be sparse, chi-squared test should be better choice for both binary and categorical features.

### 3.6.4 Effect size

In most cases, the main purpose of research is to estimate actual effects that exist in the real world. Effect size can be understood as "the degree to which the phenomenon is present in the population" [30]. In other words, it is the sensitivity of the dependent variable to changes in the independent variable. Effect size complements statistical hypothesis testing and can also help with planning of the sample size [31].

It is considered good practice to report the effect size in addition to statistical tests to provide an objective measure of importance of the results. While statistical significance reflects likeliness of our results, the effect size indicates the practical importance of our findings. For very large samples, even the miniscule effects can become statistically significant, due to the increased power of the statistical test; this is called the *P-value problem* [70].

The P-value itself does not have an objective meaning and is not an unambiguous measure of evidence. The sample size hugely influences the significance, and relying only on the P-value can lead to acceptance of the hypothesis of no practical significance. Despite that, this appears to be a common practice. Lin et al. [70] reviewed articles in two leading Information System (IS) journals and reported that 50% of recent papers with sample sizes over 10,000 were relying on low P-values.

The critical difference between effect size indices and test statistics is that the effect size is not affected by the size of the sample.

There are many measures of effect size, including odds ratio, relative risk or various correlation coefficients (e.g. Pearson's, Spearman's) [42]. For the purpose of this work, we will use two effect size indices proposed by Cohen (1992) [31] and commonly known as Cohen's $d$ and Cohen's $w$.

#### 3.6.4.1 Cohen's *d*

Cohen's $d$ is one of the most used effect size measures for continuous data and can be used for all tests of the difference between means of independent samples (e.g. t-tests). It is defined as the difference of the means divided by the pooled standard deviation of the two groups A and B: [30]

$$d = \frac{\overline{X}_A - \overline{X}_B}{s}, \tag{3.1}$$

where $\overline{X}_A$ and $\overline{X}_B$ are the means of the two groups and $s$ is the pooled

standard deviation for two independent samples defined as:

$$s = \sqrt{\frac{\sum \left(X_A - \overline{X}_A\right)^2 + \sum \left(X_B - \overline{X}_B\right)^2}{n_A + n_B - 2}}. \tag{3.2}$$

Cohen [30] also suggests following interpretation of magnitudes:

| Cohen's $d$ | Effect size |
|---|---|
| 0.2 | Small |
| 0.5 | Medium |
| 0.8 | Large |

These cutoffs, defined for most effect size measures, provide a good basis for interpreting the effect sizes; however, they tend to be misused and remain controversial practice [42]. Even Cohen himself warned about "many dangers" emerging from the use of such arbitrary categories. Therefore, these three categories can serve as a guideline for interpretation, but should not be used and trusted blindly.

#### 3.6.4.2 Cohen's *w*

Cohen's $w$ is an effect size index used for chi-squared tests. It is defined as

$$w = \sqrt{\sum_{i=1}^{k} \frac{\left(P_{1i} - P_{0i}\right)^2}{P_{0i}}}, \tag{3.3}$$

where $k$ is the number of cells and $P_{0i}$ and $P_{1i}$ are proportions in cell $i$ under the null and alternative hypothesis [30].

The interpretation of magnitudes is following:

| Cohen's $w$ | Effect size |
|---|---|
| 0.1 | Small |
| 0.3 | Medium |
| 0.5 | Large |

## 3.7 P2Rank models training and evaluation

Two scripts further extend the analysis pipeline and can be used to train P2Rank models with obtained data. Two dataset files are needed as input - one for training and another for evaluation. The process is following:

- for both datasets, run the whole pipeline twice:

  - in the first run, download data, compute mappings, ligand binding sites, features and analysis with default parameters
  - in the second run, recompute the analysis with random sampling (sample size 500, 1000 iterations)

- convert both dataset files to the format accepted by P2Rank
- create .csv files with custom features using previously computed feature values
- train and evaluate new P2Rank model on given datasets with custom features

It is possible either to train one model with all given features at once with script `pipeline_P2Rank_allFeatures.sh`, or to train one model per feature with `pipeline_P2Rank_oneFeature.sh`

# 4. Evaluation and results

## 4.1 Datasets

The choice of datasets of protein-ligand complexes used for statistical analysis and P2Rank model training and evaluation was strongly inspired by the datasets described in the P2Rank article [62]. All structures were re-downloaded directly from PDBe, according to their PDB ID (four-character alphanumeric identifier) and chain ID (one-character identifier) used in the original datasets. It was not possible to take the original datasets as they were, since the structures were not up-to-date and the annotations downloaded from the databases (e.g. feature values) could not be mapped properly.

Furthermore, the re-downloaded datasets were filtered: Obsolete structures were replaced with their current entries, structures that did not have a corresponding UniProt record were removed, as well as structures with the incorrect segments mapping due to the bug in PDBe (mentioned in section TODO-ODKAZ). The pipeline can only work with single-chain structures, and the structures from *holo4k* (see below) and a few structures from *joined* were multi-chain; thus, only one chain was chosen from each such structure.

The resulting datasets were named identically with the original datasets:

- **chen11** - a smaller non-redundant dataset that was originally designed for a comparative study of ligand binding sites predictors [25]. It comprises at most one representative chain for every SCOP family [76] to ensure the minimal sequence similarity and maximal variability in tertiary structure. The original dataset covers 6 structural classes, 148 protein folds, 184 superfamilies and 251 families [25]; after re-downloading and filtering, the numbers are slightly smaller. Although this dataset is rather small, it covers wide range of non-homologous proteins. Therefore, it serves as a good training dataset (P2Rank default model was trained on this dataset as well).

- **coach420** - a dataset that was originally taken from a benchmark study [84] and used in other studies [62, 100]. The non-redundant dataset harbors a mix of natural and drug-like ligand molecules.

- **joined** - a larger dataset created by merging smaller datasets from previous studies. It comprises a set of drug-target complexes extracted from DrugBank, DrugPort and PDB DT198[101], a benchmark set for the validation of protein-ligand docking performance [51], and a dataset with bound and

unbound structures used for evaluation of a ligand binding sites predictor [53].

- **holo4k** - a large set of protein-ligand complexes used in a large-scale evaluation of four binding sites predictors [86].

### 4.1.1 Ligands filtering

The downloaded PDB files contain more ligands per structure, and not all of them are of interest for drug design and other applications. These non-relevant ligands can be ions, peptides, small molecules such as solvents, buffers, detergents and salts that are merely artifacts, and other specific types of ligands.

For each dataset, three variants were created by different filters of the relevant ligands:

- **No filter** - Only water molecules were filtered out.
- **P2Rank filter** - Relevant ligands were obtained according to the rules used by P2Rank software [62]. These rules are:
  - the ligand has at least 5 atoms
  - at least one atom of the ligand is in distance 4 Å from any protein atom
  - the center of the mass of the ligand is not farther than 5.5 Å from the closest protein atom
  - the name of ligand PDB group is not any of following: HOH, DOD, WAT, NAG, MAN, UNK, GLC, ABA, MPD, GOL, SO4, PO4
- **MOAD filter** - Biologically relevant ligands according to the Binding MOAD [16] database. It contains manually curated crystalography protein-ligand complexes with validated biologically relevant ligands. Only structures obtained by X-ray crystalography with resolution higher than 2.5 Å have entry in Binding MOAD; other structures were removed from the datasets.

The structures without any remaining ligands after applying the P2Rank or MOAD filters were removed.

The summary of datasets properties can be seen in Table 4.1. The more strict the filter is, the lower the Binding/Non-binding ratio; nevertheless, the information obtained from the relevant binding sites should be more valuable. As we can see, MOAD filter is more strict and filteres out more ligands than P2Rank filter.

| Dataset | Proteins | Ligands | Lig./Pro. | Binding | Non-bind. | B/N ratio |
|---|---|---|---|---|---|---|
| chen11 | 241 | 1039 | 4.3112 | 5670 | 49374 | 0.1148 |
| chen11_filter_p2rank | 223 | 401 | 1.7982 | 4590 | 47073 | 0.0975 |
| chen11_filter_MOAD | 178 | 266 | 1.4944 | 3032 | 39006 | 0.0777 |
| coach420 | 417 | 841 | 2.0168 | 5988 | 80575 | 0.0743 |
| coach420_filter_p2rank | 369 | 427 | 1.1572 | 5247 | 71498 | 0.0734 |
| coach420_filter_MOAD | 258 | 291 | 1.1279 | 3688 | 48485 | 0.0761 |
| joined | 527 | 1522 | 2.888 | 8260 | 108337 | 0.0762 |
| joined_filter_p2rank | 446 | 585 | 1.3117 | 6492 | 97158 | 0.0668 |
| joined_filter_MOAD | 348 | 417 | 1.1983 | 4614 | 72363 | 0.0638 |
| holo4k | 3973 | 10391 | 2.6154 | 69866 | 790091 | 0.0884 |
| holo4k_filter_p2rank | 3842 | 5049 | 1.3142 | 62483 | 784885 | 0.0796 |
| holo4k_filter_MOAD | 3308 | 4023 | 1.2161 | 50834 | 679918 | 0.0748 |

Table 4.1: Summary of dataset properties with and without ligands filtering. *chen11* dataset has the highest average number of ligands per protein, but when the ligands are filtered, the number is comparable to the other datasets. It indicates that *chen11* has the highest ratio of biologically irrelevant ligands.

## 4.2 Statistical analysis

The statistical analysis of ligand binding sites properties was performed using the analysis pipeline described in Chapter 3 with default parameters. The results were collected for all the datasets, including the versions with filtered ligands.

The results for the datasets with different ligands filters does not seem to differ widely and does not reveal any additional informations. For that reason and for better clarity of the text, following features analysis and plots will be shown only for the datasets with P2Rank ligands filter. The results obtained from the analysis pipeline for all the datasets with and without filters are included in Attachments. TODO odkaz

The P-values computed by the hypothesis tests are shown in Table 4.2 and discussed below.

Three artificial features were added for comparison and to check the validity of the tool:

- `lbs` - Ligand binding sites labels (0/1). Should have the best performance of all the features, the P-value should be zero.
- `random_binary` - Random binary numbers. Should not be significant.
- `random_cont` - Random continuous feature with values from uniform distribution from 0 to 10. Should not be significant.

Some features had to be excluded from the analysis since the data were very sparse and the assumptions of the hypothesis tests would not be met. For example, there were only 15 lipidation sites in the whole *holo4k* dataset containing 857,635 residues. The excluded features are: `lipidation`, `glycosylation`, `non_standard` and `compbias`.

The `conservation` feature was computed only for the three smaller datasets and was omitted for *holo4k*. The computational time would be very high, as it takes 15-30 minutes on average per structure, and the dataset contains almost four thousand proteins. Nevertheless, the comparison on the other three datasets should be sufficient. Furthermore, the computation ended with error for some structures, as there were not enough sequences found by BLAST (the required number was set to 30).

The problem with feature `variation` was that the data were missing for many structures (around 3/4) as downloading via REST API resulted in *404 Not Found* error. Data were not available on the UniProt website either. This might be caused by lack of variation data from large-scale studies for some organisms. UniProt helpdesk was contacted to help to explain the issue, but, unfortunately, the question was left without answer. Nevertheless, the feature was analysed on the subset of structures where the data were available.

For some features downloaded from databases, such as `depth` or `dynamine`, there were missing data for a few structures as well. These cases were not very frequent and they most likely could not affect the analysis, so they were omitted.

As we can see from the results in Table 4.2, most features appear to be statistically significant, having the P-value below the significance level $\alpha = 0.05$. The results for the test features `lbs`, `random_binary` and `random_cont` seem to be valid. Nevertheless, when looking at the histograms and plots, some results are not as expected. Let's take a look at the histogram depicted in Figure 4.1: the distribution of `dynamine` values does not seem significantly different in binding and non-binding sites. Note that for better comparison of binding and non-binding sites (since their ratio is very unbalanced), the density is computed with respect to the number of binding or non-binding sites; the value in the histogram bin can be understood as conditional probability of getting that value when having a binding/non-binding residue.

One conspicuous thing about the Table 4.2 is that, in general, the P-values are getting smaller as the dataset size grows (the datasets in the table are sorted from the smallest on the left to the largest on the right). This is referred to as the *P-value problem*. For very large samples, the statistical power of hypothesis tests is higher, and causes P-value going to zero. When dealing with large samples, even the miniscule effects can become statistically significant. The test can detect

| | chen11 | coach420 | joined | holo4k |
|---|---|---|---|---|
| **lbs (test)** | 0 | 0 | 0 | 0 |
| **pdbekb_conservation** | 0 | 0 | 0 | 0 |
| **conservation** | 0 | 0 | 0 | X |
| **HSE_up** | 0 | 0 | 0 | 0 |
| **exposure_CN** | 1.43E-278 | 0 | 0 | 0 |
| **depth** | 1.26E-241 | 1.38E-251 | 0 | 0 |
| **HSE_down** | 8.20E-158 | 1.99E-244 | 0 | 0 |
| **bfactor** | 3.79E-197 | 1.12E-143 | 0 | 0 |
| **mol_weight** | 4.70E-143 | 7.15E-136 | 1.16E-245 | 0 |
| **aa** | 3.96E-143 | 1.80E-137 | 7.54E-245 | 0 |
| **hydropathy** | 2.83E-139 | 4.26E-137 | 1.40E-244 | 0 |
| **aromaticity** | 1.17E-78 | 1.56E-57 | 3.05E-111 | 0 |
| **H_bond_atoms** | 6.57E-53 | 2.85E-49 | 8.00E-109 | 0 |
| **sec_str** | 1.44E-17 | 1.34E-52 | 4.66E-44 | 0 |
| **polarity** | 5.99E-10 | 6.97E-25 | 5.31E-37 | 0 |
| **charged** | 8.28E-11 | 2.45E-25 | 9.96E-29 | 0 |
| **strand** | 4.24E-18 | 9.51E-35 | 1.74E-44 | 4.40E-279 |
| **helix** | 1.45E-07 | 1.01E-35 | 3.23E-16 | 1.21E-268 |
| **disulfid** | 0.353 | 4.29E-07 | 0.1728 | 2.16E-94 |
| **mod_res** | 0.362 | 0.004458 | 0.000132 | 5.95E-55 |
| **mobiDB** | 1.22E-06 | 0.0004838 | 7.91E-13 | 9.73E-54 |
| **cis_peptide** | 0.1914 | 9.85E-05 | 2.63E-05 | 1.06E-48 |
| **natural_variant** | 0.7951 | 0.000333 | 1.19E-05 | 7.46E-45 |
| **PTM** | 0.8162 | 0.04154 | 0.007383 | 1.83E-38 |
| **phi_angle** | 1.91E-05 | 2.85E-05 | 6.08E-08 | 6.68E-37 |
| **psi_angle** | 0.02216 | 1.79E-15 | 0.002092 | 2.58E-20 |
| **efoldmine** | 0.0001874 | 0.001525 | 7.54E-23 | 2.51E-14 |
| **dynamine** | 0.007162 | 0.01094 | 1.26E-07 | 0.0003708 |
| **variation** | 0.1864 | 0.2724 | 0.01176 | 0.004204 |
| **turn** | 0.6574 | 0.00167 | 0.5149 | 0.03571 |
| **random_cont (test)** | 0.8255 | 0.9688 | 0.9748 | 0.6919 |
| **random_binary (test)** | 0.04112 | 0.1669 | 0.1722 | 0.7793 |

Table 4.2: P-values returned by hypothesis tests for individual features for all four datasets (with P2Rank ligands filtering). Features are sorted according to the values for *holo4k*. Values highlighted with red colour are higher that the significance level $\alpha = 0.05$.

*`variation` is computed only on the small subsets of proteins for which the data were available in databases.

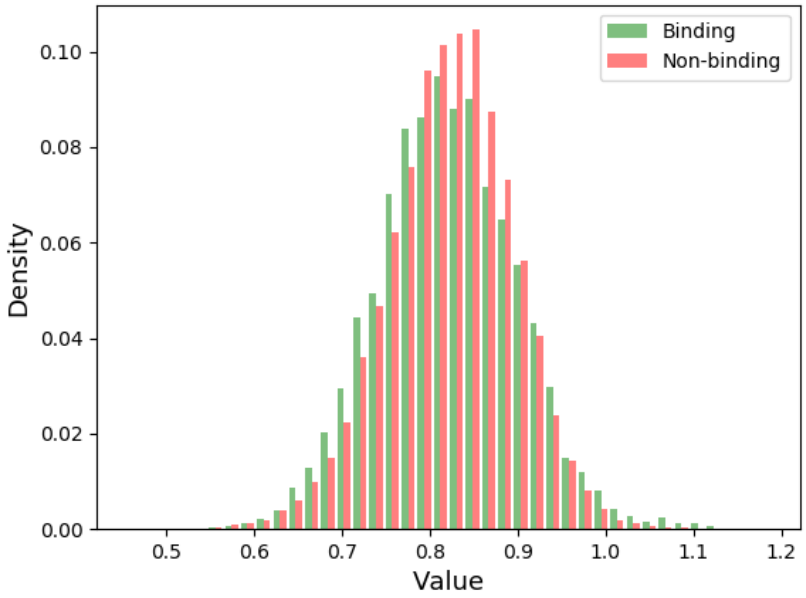
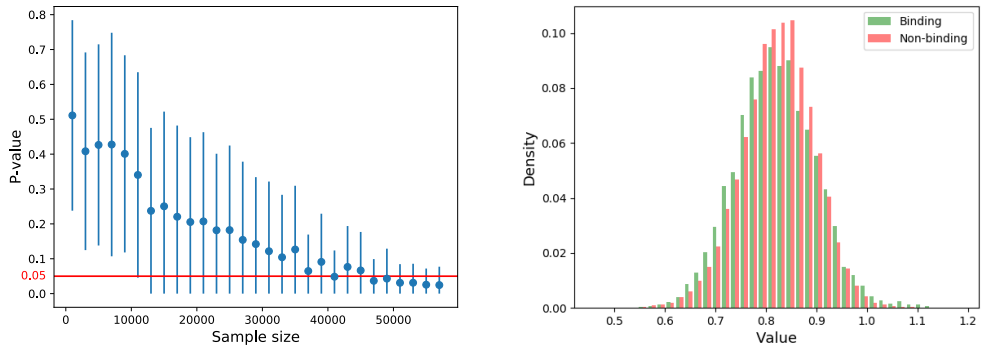



Figure 4.1: Histogram for feature `dynamine` computed on *holo4k* dataset. Density on the y-axis is computed with respect to the number of binding or non-binding sites. Difference in means: 0.0012.

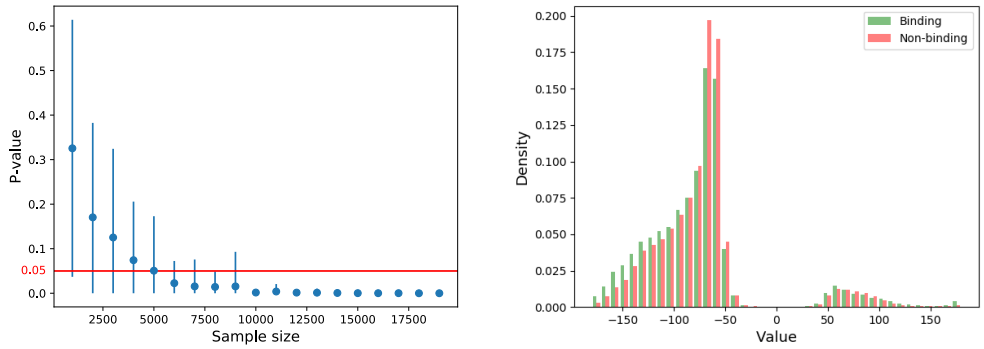
subtler and more complex effects, which can be advantageous in some cases, but also misleading. It all depends on the purpose of the statistical testing. The question we should ask is not whether the results are statistically significant (which there almost always is for large samples), but whether they are interesting for our research [70].

Let's see the P-value problem demonstrated on our data. Figure 4.2 shows different speeds of P-value deflation for chosen features. The computation of the plots was inspired by 'Monte Carlo CPS Charts' described by Lin *et al.* [70]. For feature `exposure_CN`, sample size of 50 is sufficient to get P-values below significance level 0.05. We can clearly see from the histogram that the values in binding and non-binding sites differ, so this is an expected result. On the contrary, the values of feature `dynamine` does not seem to differ significantly for binding and non-binding sites, and yet, if the sample size is large enough, we get statistically significant result.

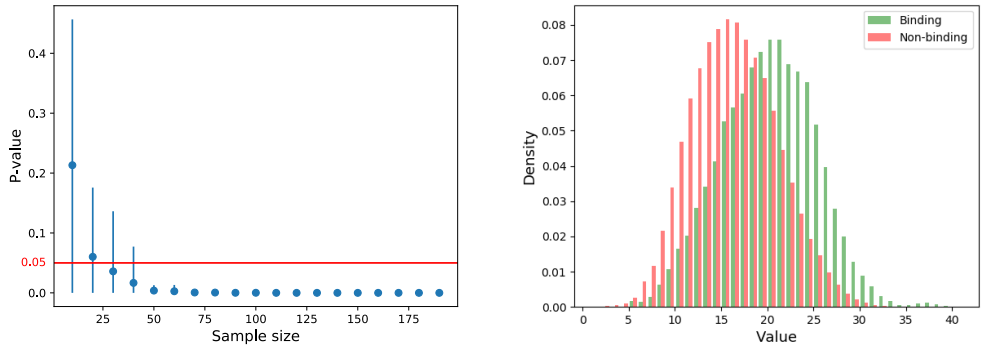
To complement the hypothesis tests and to provide an objective measure of importance of the results, we report two effect size measures, as described in Section 3.6.4: Cohen's $d$ for continuous features and Cohen's $w$ for the rest. The results can be found in Table 4.3 and Table 4.4. Looking at the effect sizes, it seems that many low P-values reported in Table 4.2 are most likely mere artifacts of the large sample sizes. Only a few features seem to have any practical




(a) `dynamine`




(b) `phi_angle`




(c) `exposure_CN`

Figure 4.2: Different speed of P-value deflation demonstrated on three features. The P-value decreases with increasing sample size. The results were obtained from 100 iterations of random sampling with given sample size, taking the same number of binding and non-binding residues. Mean P-values and standard deviations are displayed (the error bars are cut so they are not negative). Computed on dataset *holo4k* with P2Rank ligands filter. The red line represents significance level $\alpha = 0.05$.

| | chen11 | coach420 | joined | holo4k |
|---|---|---|---|---|
| **conservation** | 0.7726 | 0.9009 | 0.8083 | X |
| **exposure_CN** | 0.6161 | 0.7419 | 0.8181 | 0.839 |
| **depth** | 0.6666 | 0.6308 | 0.8569 | 0.8255 |
| **HSE_up** | 0.5772 | 0.6512 | 0.7513 | 0.7373 |
| **HSE_down** | 0.4424 | 0.5236 | 0.5874 | 0.5895 |
| **bfactor** | 0.3959 | 0.3207 | 0.3872 | 0.4382 |
| **phi_angle** | 0.06842 | 0.06729 | 0.07535 | 0.0592 |
| **mobiDB** | 0.06575 | 0.04492 | 0.07718 | 0.05498 |
| **psi_angle** | 0.03685 | 0.1209 | 0.04151 | 0.04111 |
| **efoldmine** | 0.06121 | 0.0491 | 0.1455 | 0.03436 |
| **dynamine** | 0.04341 | 0.04098 | 0.07485 | 0.01685 |
| **random_cont (test)** | 0.003419 | 0.00056 | 0.0004042 | 0.001647 |

Table 4.3: Values of Cohen's $d$ for continuous features. Features are sorted according to the values for *holo4k*.

significance. The individual features will be discussed below.

Another noticeable thing about Table 4.2 is that the results for some features vary across datasets. Let's take a look at features `disulfid` and `turn`, for example. The P-value is very high for datasets *chen11* and *joined*; contrarily, it is low for *coach420* and *holo4k*. In this case it is not true that the P-value would decrease with the increasing sample size. This leads to a question of how the datasets are composed, and whether they are representative samples from the whole population of proteins. Taken into consideration the way how the datasets were assembled, it is likely that some bias was introduced. The question is whether taking the whole PDB database would help to solve this issue. There probably would be the problem with redundancy of data, as close homologs and overlapping PDB entries would be included. Furthermore, the database itself is most likely a biased sample of the real world of proteins, as the tertiary structure is yet to be discovered for many of them. And most importantly, this approach would be computationally very demanding.

| | chen11 | coach420 | joined | holo4k |
|---|---|---|---|---|
| **lbs (test)** | 0.9999 | 0.9999 | 0.9999 | 1 |
| **pdbekb_conservation** | 0.341 | 0.3266 | 0.3149 | 0.349 |
| **aa** | 0.1191 | 0.09592 | 0.108 | 0.1127 |
| **mol_weight** | 0.1188 | 0.09515 | 0.108 | 0.1124 |
| **hydropathy** | 0.1167 | 0.09502 | 0.1074 | 0.1113 |
| **H_bond_atoms** | 0.07016 | 0.05561 | 0.07041 | 0.07102 |
| **aromaticity** | 0.08261 | 0.05771 | 0.06961 | 0.06377 |
| **sec_str** | 0.03977 | 0.05644 | 0.04458 | 0.04623 |
| **polarity** | 0.02867 | 0.03807 | 0.04015 | 0.04574 |
| **charged** | 0.02858 | 0.03755 | 0.03454 | 0.04437 |
| **strand** | 0.03819 | 0.04444 | 0.04363 | 0.03883 |
| **helix** | 0.02316 | 0.04509 | 0.02546 | 0.03809 |
| **disulfid** | 0.00409 | 0.01827 | 0.004251 | 0.02242 |
| **mod_res** | 0.004014 | 0.01028 | 0.01192 | 0.01698 |
| **cis_peptide** | 0.005721 | 0.01404 | 0.01304 | 0.01592 |
| **natural_variant** | 0.001144 | 0.01297 | 0.01366 | 0.01528 |
| **PTM** | 0.001024 | 0.007366 | 0.008354 | 0.01411 |
| **variation** | 0.01132 | 0.008884 | 0.01283 | 0.005528 |
| **turn** | 0.001953 | 0.01136 | 0.00203 | 0.002284 |
| **random_binary (test)** | 0.008985 | 0.00499 | 0.00424 | 0.0003044 |

Table 4.4: Values of Cohen's $w$ for binary, ordinal and categorical features. Features are sorted according to the values for *holo4k*.

## 4.3 Statistical analysis - random sampling

To make use of our large datasets, one different approach was implemented. Dataset *mix* was created by merging all four datasets together, removing a few duplicates. Random sampling without replacement was applied on this dataset, in each iteration taking a sample of 500 binding and 500 non-binding sites without replacement. 1000 iterations were executed and mean effect sizes were reported. The resulting effect sizes are shown in Table 4.5 and Table 4.6.

This approach has the advantage of removing possible dependencies of nearby residues. For many features, the values of individual residues are not independent of each other. For instance, a helix or a beta sheet always covers several consecutive residues. The features obtained from the structure, such as different measures of burriedness (e.g. `exposure_CN`, `depth`) can have similar values for spacially near residues. And the features computed from the FASTA sequence, such as `dynamine` or `mobiDB`, most likely have some dependencies as well, from the principle of their computation. When taking a small sample of 500 residues, the chance of these dependencies affecting the analysis is minimal.

The sample size of 500 was chosen for two reasons: firstly, validity of the Central Limit Theorem needs to be assured, as described in Section 3.6.2. Lumley *et al.* [72] demonstrated that 500 is a sufficiently large sample even for extremely non-normal data. And secondly, the minimum sample size assuring the Central Limit Theorem validity should be chosen, to avoid the P-value problem. Smaller sample size would probably be sufficient for the Central Limit Theorem, as 500 is a very safe estimation. Nevertheless, the sample size could not be much smaller anyhow, since the data for some categorial features would be very sparse. Even with the sample size of 500, features `PTM`, `mod_res`, `natural_variant`, `disulfid` and `cis_peptide` needed to be excluded from the analysis, as there was not sufficient number of positives in this smaller sample.

| | mean | standard deviation |
|---|---|---|
| **conservation** | 0.8812 | 0.06919 |
| **exposure_CN** | 0.7829 | 0.06753 |
| **HSE_up** | 0.7511 | 0.06866 |
| **depth** | 0.6927 | 0.06815 |
| **HSE_down** | 0.5649 | 0.06453 |
| **bfactor** | 0.4744 | 0.06165 |
| **phi_angle** | 0.07136 | 0.05012 |
| **mobiDB** | 0.07066 | 0.04874 |
| **psi_angle** | 0.06029 | 0.045 |
| **efoldmine** | 0.05961 | 0.04477 |
| **dynamine** | 0.05236 | 0.04086 |
| **random_cont (test)** | 0.04968 | 0.03807 |

Table 4.5: Mean effect sizes (cohen's $d$) computed from 1000 iterations of random sampling with sample size 500. Computed on dataset *mix* (all 4 datasets merged together) with P2Rank ligands filter. Features are sorted according to the effect size.

## 4.4 Discussion

Let's discuss the results from the two previous sections in more detail.

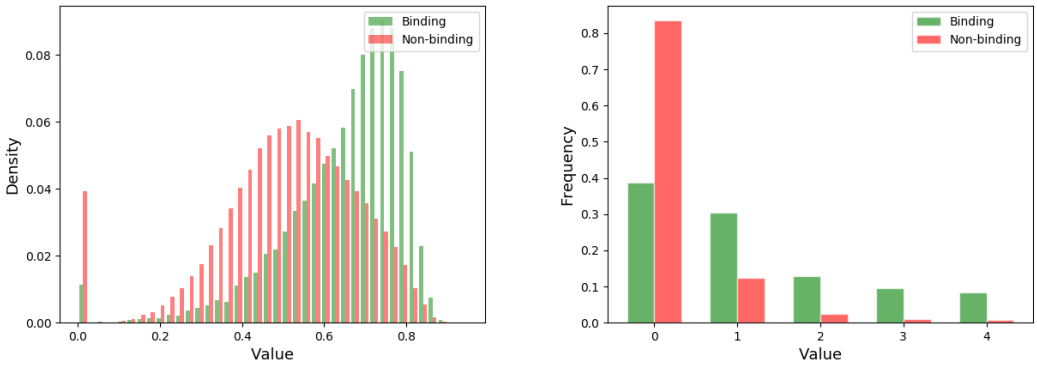
The results for both conservation features `conservation` and `pdbekb_conservation` turned out as expected. Sequence conservation has been used previously in many approaches for protein-ligand binding sites prediction and its importance for the prediction has been repeatedly demonstrated [36, 53, 55, 96]. The higher values of conservation for binding residues are clearly visible from the Figure 4.3. The effect sizes are the highest from all the examined features and it seems that the sequence conservation might be even more correlated with binding sites positions than the locations of cavities on the protein surface.

Features `HSE_up`, `HSE_down`, `exposure_CN` and `depth` are closely related to the 'buriedness' of the residue. Similarly as conservation, this feature was expected to be important for ligand binding sites recognition. Many binding sites are shaped as cavities, or concave pockets, on the surface of the 3D structure. The geometrical methods, such as LIGSITE [53] or PocketPicker [97], as well as other approaches (including P2Rank), make use of this property (as explained in Section 2.1. The histograms for these continuous features are depicted in Figure 4.4. The effect sizes for all four features indicate quite large effect.

The analysis reveals that binding sites have slightly lower B factor values on

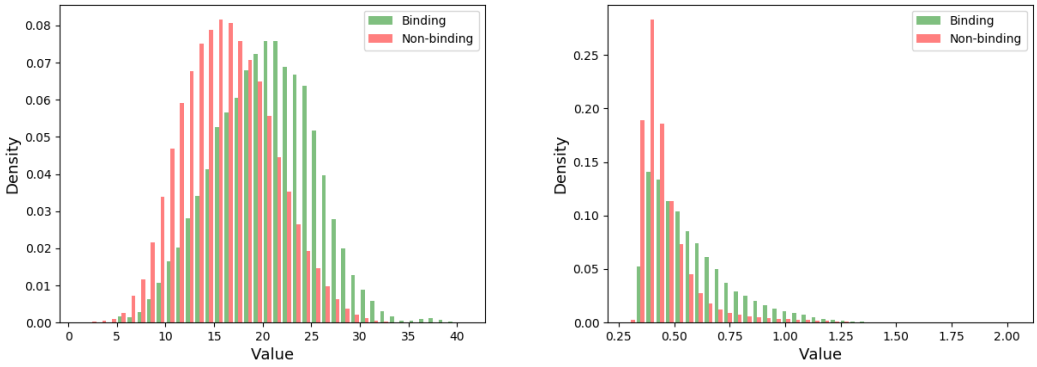
| | mean | standard deviation |
|---|---|---|
| **lbs (test)** | 0.998 | 2.22E-16 |
| **pdbekb_conservation** | 0.4742 | 0.02518 |
| **aa** | 0.2443 | 0.02677 |
| **mol_weight** | 0.2414 | 0.0259 |
| **hydropathy** | 0.2336 | 0.02724 |
| **H_bond_atoms** | 0.1411 | 0.02893 |
| **aromaticity** | 0.1067 | 0.03028 |
| **sec_str** | 0.09798 | 0.02917 |
| **polarity** | 0.09157 | 0.0296 |
| **charged** | 0.08189 | 0.032 |
| **strand** | 0.07289 | 0.03048 |
| **helix** | 0.07052 | 0.02998 |
| **random_binary (test)** | 0.02375 | 0.01976 |
| **variation** | 0.02298 | 0.01929 |
| **turn** | 0.02023 | 0.01797 |

Table 4.6: Mean effect sizes (cohen's $w$) computed from 1000 iterations of random sampling with sample size 500. Computed on dataset *mix* (all 4 datasets merged together) with P2Rank ligands filter. Features are sorted according to the effect size. *`variation` is computed only on the subsets of proteins for which the data were available in databases.

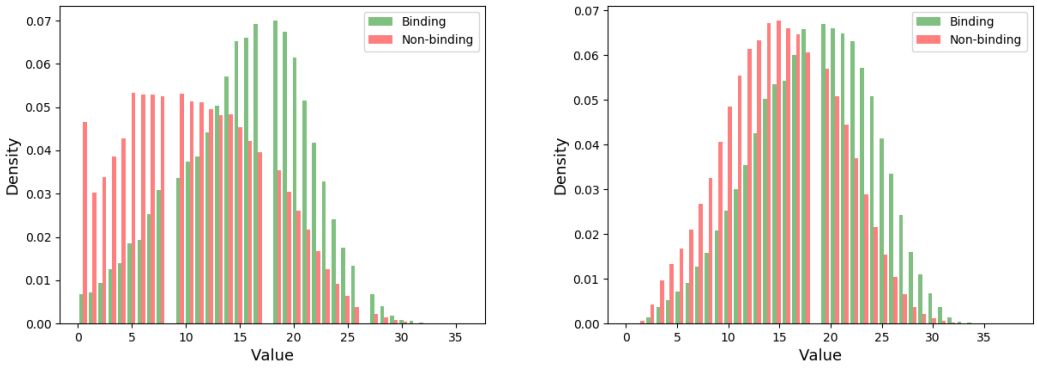



(a) `conservation` (b) `pdbekb_conservation`

Figure 4.3: Higher values of conservation for binding residues demonstrated on two features: (a) continuous feature computed by the P2Rank conservation pipeline, and (b) ordinal feature downloaded from PDBe-KB database.




(a) `exposure_CN` (b) `depth`




(c) `HSE_up` (d) `HSE_down`

Figure 4.4: The features related to the residue 'buriedness' have higher values in binding sites.

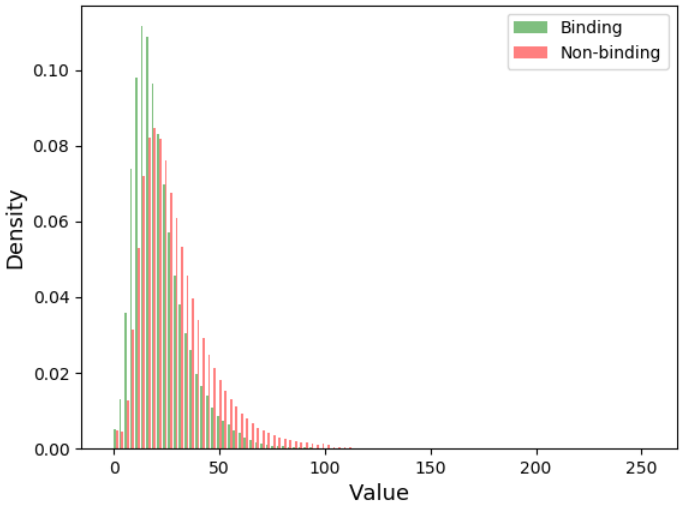
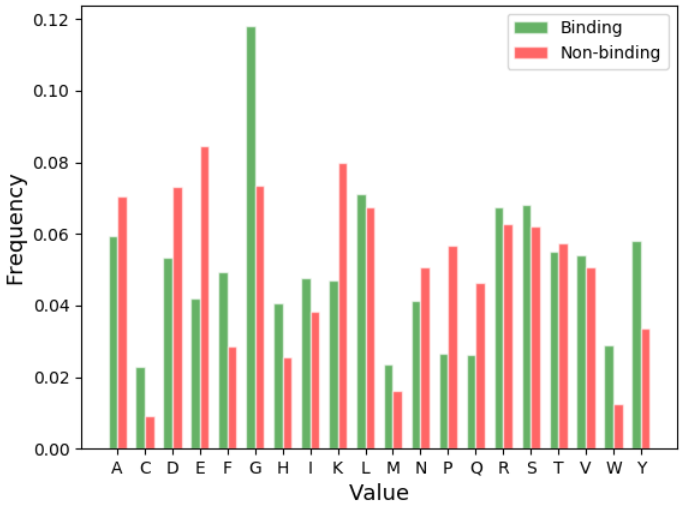



Figure 4.5: `bfactor`: Binding sites seem to have lower B factor values, which could indicate bigger rigidity of binding sites.

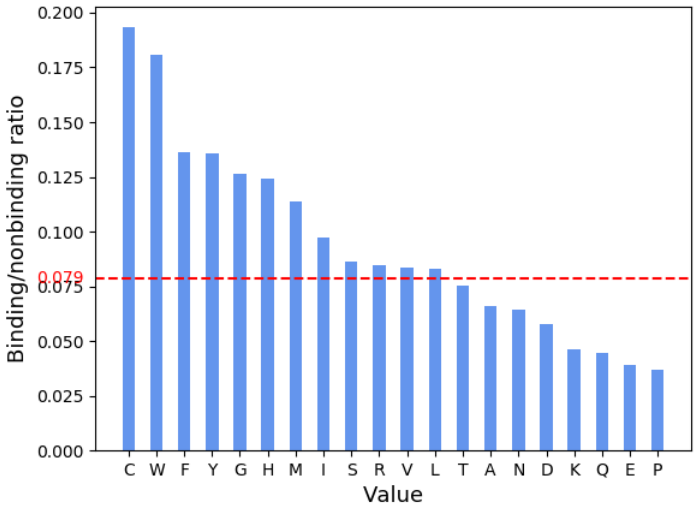
average (see Figure 4.5). This could indicate that binding sites are more well-ordered in general, whereas non-binding sites might have higher flexibility.

Binding and non-binding sites seem to have different residue composition. Let's take a look at Figure 4.6. Cys, Trp, Phe, Tyr, Gly, His, Met and Ile all have high binding/non-binding ratios, and thus, are more likely to occur in binding sites. On the other hand, Pro, Glu, Gln, Lys and Asp disfavour binding sites. Arg, Val, Ser and Leu are very frequent in binding sites; however, they have the ratios similar to the total binding/non-binding ratio, as they are very frequent on the whole protein surface, not only in binding sites. This result is in accordance with a large-scale study that explored the composition of protein-ligand binding sites [60]. This is an interesting result; nevertheless, the effect size is smaller than for the previously discussed features and it is not clear whether the tendency of some amino acids to appear in the binding sites with higher frequency could be used for the prediction.

The molecular weight and hydropathy seem to have similar effect as the amino acid composition. This result is most probably caused by the fact that the values of the molecular weight and the hydropathy index correlate with the amino acid labels almost completely. To explain it more clearly, `mol_weight` is an ordinal feature with 19 categories, where one category means one value of molecular weight for one amino acid (only leucine and isoleucine have the same molecular weight and thus fall into the same category). It means that computing this feature is almost equivalent to computing feature `aa`, only with different labels for categories and two amino acids merged into one category. The feature `hydropathy`




(a) Frequencies of individual amino acids in binding and non-binding sites.




(b) Comparison of binding/non-binding ratios, computed as occurrences of the AA in binding sites divided by its occurrences in non-binding sites. The red line marks the total binding/non-binding ratio (total number of binding sites divided by total number of non-binding sites). High ratio means that the AA favours binding sites, and on the contrary, the low ratio indicates the tendency to occur in non-binding sites.

Figure 4.6: Feature `aa`.

is similar, only with 17 categories. The rest of amino acid-based features does not have this characteristic, as they are either binary (`charged`, `aromaticity`) or with small number of categories (`polarity`, `H_bond_atoms`).

The rest of the features does not seem to be very important for binding sites differetiation, as the effects are very small or negligible. `aromaticity` and `H_bond_atoms` features look the most promising of the rest and could be given a try. None of properties such as torsion angles, secondary structures, disorder regions, polarity and charge of a residue, early folding regions or regions with increased backbone dynamics appear to have significantly different value within the binding sites. This means that unfortunately, the analysis has not revealed any novel features that would help to increase the protein-ligand binding sites prediction.

| | $D_{CA}$ [4Å] | | $D_{CC}$ [4Å] | |
|---|---|---|---|---|
| | **Top-$n$** | **Top-($n$+2)** | **Top-$n$** | **Top-($n$+2)** |
| **baseline model** | 73.6 | 78.0 | 47.2 | 49.6 |
| **P2Rank + csv features** | 75.1 | 77.5 | 47.5 | 50.2 |
| **csv features** | 70.3 | 74.8 | 39.5 | 42.9 |

Table 4.7: Comparison of the performance of models with different sets of features: P2Rank default features (baseline model), csv features and both sets together.

## 4.5 P2Rank models

The features examined in the previous sectio were used to train new P2Rank models and to find out their practical significance. All models were trained with the same parameters as the default P2Rank model (100 trees, each grown with no depth limit using 6 features). The models were trained on *chen11* dataset and evaluated on *coach420* dataset. The training was done with parameter `loop=10` which means that the model was trained 10 times, every time with different random seed, and the performances of the 10 resulting models were averaged at the end. This is important for reducing the influence of the random behaviour of the classifier.

The features obtained by the analysis pipeline are called 'csv features' in the following section, in accordance with the terminology used by P2Rank for user-defined features in csv files.

A couple of features were left out from the following experiments. The feature `variation` had missing values for around 3/4 structures, as mentioned in the previous section, and the comparison with the rest of the features would not make much sense. Categorical feature `aa` was replaced by 20 binary features representing individual amino acids (`aa_HIS`, `aa_SER` etc.). Similarly, categorical feature `sec_str` was left out and represented by features `helix`, `strand` and `turn`.

The performance of the baseline model (trained with all default P2Rank features and no csv feature) was compared with the model trained with all csv features (test features `lbs`, `random_binary` and `random_cont` were left out of course), and the model trained with both P2Rank features and csv features. The results are summarized in Table 4.7. Two metrics are reported: $D_{CA}$ and $D_{CC}$, as described in Section 2.2, with distance threshold 4Å. We used Top-$n$ and Top-($n$+2) rank cutoffs ($n$ is the number of relevant ligands).

The baseline model performs a little better in the Top-$n$ category than the default P2Rank model described in the P2Rank article [62], which achieved the

success rate of 72%. This can be caused by slightly different datasets (as described in Section 4.1), or, to some degree, by the random behaviour of the classifier.

When all csv features are added to the baseline model, the performance increases by only 1.5 %. The reason probably is that many P2Rank features are identical or very correlated with the new csv features. P2Rank already uses B factor, amino acid properties, buriedness and other properties for training, and csv features evidently does not contribute with much new information. Moreover, csv features are mutually correlated, as well as P2Rank features. These highly correlated features, as well as having so many irrelevant features, cannot be beneficial for the prediction.

The performance of the model trained with csv features only is inferior to the baseline model, but surprisingly, the difference is only 3.3 %.

We can see that the performances measured in the $D_{CC}$ metric are very low. This is unfortunatelly true for various tools. The benchmark study published by Chen *et al.* [25] compared 10 predictors and in the Top-$n$ category with cutoff 4Å, the top-performing predictor (FINDSITE [90]) had 57% success rate; the rest of the methods scored at most 28%. Furthermore, FINDSITE is a template-based method, relying on the availability of known protein-ligand complexes similar to the query protein; moreover, the 57 % success rate was achieved with the similarity threshold set to 1.

Let's take a look at how the individual csv features help to improve the performance, by adding only one at a time. Table 4.8 summarizes the results of training one model per feature, with one csv feature plus all P2Rank default features. Only models with features `pdbekb_conservation` and `conservation` are clearly superior to the baseline model. There are a few features that help to increase the performance of the baseline model, but only by tenths of percent; this difference is too small to proclaim the results significant. The rest of the features do not help to improve P2Rank performance, probably due to the correlations and recurrence of the similar features that were already included in P2Rank.

No csv features (except for two conservation features) seem to provide new information to the P2Rank classifier. This way, it is not possible to find out whether the statistical analysis produced meaningful results.

We can get a kind of comparison of individual csv features thanks to the feature importances obtained by the Random Forests algorithm. Feature importances (also called 'variable importances') indicate how much is every feature important for the prediction. The importances listed in Table 4.9 were obtained from training the model with all csv features and no P2Rank features (the same model as in Table 4.7. This comparison can outline the real importances to some degree, but keep in mind that it is only indicative. The importance measures

| | $D_{CA}$ [4Å] | | $D_{CC}$ [4Å] | |
|---|---|---|---|---|
| | **Top-$n$** | **Top-($n$+2)** | **Top-$n$** | **Top-($n$+2)** |
| **lbs (test)** | 90.2 | 90.5 | 78.1 | 78.6 |
| **pdbekb_conservation** | 78.9 | 81.6 | 48.9 | 51 |
| **conservation** | 76.8 | 79.8 | 47.7 | 50.4 |
| **psi_angle** | 74.3 | 78.4 | 47.1 | 49.5 |
| **aa_ASN** | 74.1 | 78.7 | 46.9 | 49.6 |
| **HSE_down** | 74.1 | 78.6 | 47 | 49.4 |
| **efoldmine** | 74.1 | 78 | 46.5 | 48.8 |
| **aa_GLU** | 73.9 | 78.3 | 47.1 | 49.6 |
| **aa_PHE** | 73.9 | 78.2 | 47.5 | 49.9 |
| **aa_SER** | 73.8 | 78.5 | 47.3 | 49.9 |
| **aa_ILE** | 73.8 | 78.4 | 47 | 49.4 |
| **aa_ASP** | 73.8 | 78.4 | 46.7 | 49.3 |
| **turn** | 73.8 | 78.3 | 47.2 | 49.4 |
| **aa_GLY** | 73.8 | 78.2 | 47.3 | 49.9 |
| **phi_angle** | 73.8 | 78 | 47.1 | 49.4 |
| **aa_TYR** | 73.7 | 78.4 | 47 | 49.5 |
| **helix** | 73.6 | 78.6 | 47.1 | 49.5 |
| **aa_VAL** | 73.6 | 78.4 | 47.4 | 49.8 |
| **aa_LEU** | 73.6 | 78.4 | 46.9 | 49.4 |
| **aa_ARG** | 73.6 | 78.2 | 47.1 | 49.6 |
| **strand** | 73.6 | 78.2 | 46.8 | 49.3 |
| **HSE_up** | 73.6 | 78.2 | 46.2 | 48.6 |
| **aromaticity** | 73.6 | 78.1 | 46.9 | 49 |
| **dynamine** | 73.6 | 78.1 | 46.9 | 49.3 |
| **aa_MET** | 73.6 | 78.1 | 46.8 | 49 |
| **random_binary (test)** | 73.6 | 78.1 | 46.7 | 49.1 |
| **aa_THR** | 73.6 | 78 | 47.1 | 49.4 |
| **mobiDB** | 73.6 | 78 | 47 | 49.4 |
| **mol_weight** | 73.5 | 78.2 | 46.9 | 49.5 |
| **H_bond_atoms** | 73.5 | 78.1 | 47.2 | 49.8 |
| **aa_CYS** | 73.5 | 78.1 | 47.1 | 49.7 |
| **charged** | 73.5 | 78.1 | 46.9 | 49.2 |
| **aa_HIS** | 73.5 | 78 | 46.8 | 49.3 |
| **exposure_CN** | 73.5 | 78 | 46.6 | 48.8 |
| **random_cont (test)** | 73.5 | 77.9 | 46.5 | 48.8 |
| **aa_LYS** | 73.4 | 78.3 | 46.6 | 49.4 |
| **aa_PRO** | 73.3 | 78.3 | 46.7 | 49.3 |
| **aa_GLN** | 73.3 | 78 | 47 | 49.5 |
| **aa_ALA** | 73.3 | 77.9 | 47.2 | 49.6 |
| **hydropathy** | 73.3 | 77.8 | 46.5 | 49.1 |
| **depth** | 73.2 | 78.4 | 46.2 | 48.9 |
| **aa_TRP** | 73.1 | 77.9 | 46.6 | 48.9 |
| **bfactor** | 73 | 77.5 | 47.4 | 49.8 |

Table 4.8: Performances of models trained with individual csv features used together with all P2Rank features. The features are sorted according to the performances in the first column.

can be affected by correlations of the features [**?** ] and can be biased when the features have different number of categories or different scales of measurement (continuous, nominal etc.) [**?** ]. Unfortunately, this experiment has all of these characteristics.

The feature importances roughly correspond with the results of the statistical analysis. Both conservation features take place on the top, as well as four features related to the 'burriedness'. `bfactor` feature is also among the top scoring features. We observed only smaller effect size for `aromaticity` feature, but it seems that it can provide new information to the classifier. Feature `helix` appears to be more important than we would expect based on the statistical analysis; however, it is hard to tell what is the reason for that. It is possible that even if the effect is small, it can still be useful for the classifier. It could be caused by the issues with correlation and number of categories, as described above, or simply by pure luck - a small perturbation of training data or input parameters can cause different features selection.

TODO dalsi modely

TODO zminit, ze tam muze mit efekt to, jak P2Rank mapuje hodnoty na sas points + proc lbs vychazi tak malo

TODO nejak to shrnout, jestli teda pipelina pomohla a jak, ze muze slouzit pro hrubou predstavu, ale tech faktoru, co to ovlivnuji, je hodne, jak v te analyye, tak pri trenovani stromu. Ze konzervovanost pomohla, ale nic moc jineho nepomohlo, protoze vsechny ty dobre featury uz tam v nejake forme byly predtim.

| feature | importance |
|---|---|
| pdbekb_conservation | 0.008592 |
| HSE_up | 0.006906 |
| conservation | 0.004021 |
| depth | 0.003474 |
| HSE_down | 0.002467 |
| exposure_CN | 0.002011 |
| aromaticity | 0.001627 |
| bfactor | 0.000891 |
| helix | 0.000807 |
| aa_HIS | 0.000783 |
| hydropathy | 0.000706 |
| efoldmine | 0.000675 |
| dynamine | 0.000661 |
| strand | 0.000645 |
| mobiDB | 0.000578 |
| charged | 0.000564 |
| mol_weight | 0.000533 |
| aa_PHE | 0.000516 |
| phi_angle | 0.000457 |
| aa_GLY | 0.000454 |
| psi_angle | 0.000433 |
| H_bond_atoms | 0.00042 |
| aa_CYS | 0.000389 |
| aa_LEU | 0.000363 |
| aa_TYR | 0.000305 |
| aa_GLU | 0.000221 |
| aa_PRO | 0.000221 |
| aa_SER | 0.000211 |
| aa_ILE | 0.000209 |
| aa_ARG | 0.000209 |
| aa_MET | 0.000196 |
| aa_LYS | 0.000194 |
| aa_ASP | 0.000138 |
| aa_VAL | 0.000137 |
| aa_TRP | 0.000121 |
| aa_THR | 0.000116 |
| aa_ALA | 0.000107 |
| aa_ASN | 0.000102 |
| aa_GLN | 9.1E-05 |
| turn | 9E-05 |

Table 4.9: Importances of csv features returned by the Random Forests algorithm. The model was trained with all csv features and without P2Rank features.

## 4.6 Experiment: comparison of conservation features

In this section, we show the usage of the analysis pipeline on a practical example.

Apart from the default model, P2Rank is currently distributed with a pre-trained conservation-aware model. The conservation scores are computed by the conservation pipeline mentioned in Section 3.5.5.2. Including conservation feature increases P2Rank performance by 1.3 % (measured in the $D_{CA}$ metrics for Top-$n$ category), as reported by Jendele *et al.* [55].

The disadvantage of this conservation feature is that it slows down the prediction dramatically. The computation of conservation scores can take several minutes for one protein; in our experiments (on a single 2,30 GHz CPU core), the cases when the computation took more than 30 minutes were not exceptional. Without the conservation feature, P2Rank is able to output predictions within a couple of seconds [62].

There exist other tools that are able to compute the conservation scores. The question is, are they as good as the conservation currently used by P2Rank?

The impact of P2Rank conservation was compared with another conservation tool, called INTAA-conservation [**?** ]. Similarly as P2Rank conservation pipeline, this tool uses multiple sequence alignment to calculate the conservation scores, but should be significantly faster. By default, UniProtKB/Swiss-Prot sequence database [**?** ] is used.

Although the conservation feature downloaded from PDBe-KB database was analysed previously, we include it into this comparison as well.

Thus, we compare two continuous features `INTAA_conservation` and `conservation` (P2Rank conservation), and one ordinal feature `pdbekb_conservation`. The last two are the same features as described in the previous sections.

The comparison of the conservation features was made on two datasets: *chen11* and *coach420*. Some structures were excluded from the original datasets because it was not possible to obtain the alignments, most likely because there was not sufficient number of matches in the sequence database and thus it was not possible to create the multiple sequence alignment. For *chen11*, 16 structures were excluded because of the missing data for `conservation` and 3 structures because of `INTAA_conservation` feature. Values for `pdbekb_conservation` downloaded from PDBe-KB database were available for all the structures in *chen11* and were missing only for two structures in *coach420*.

The whole procedure of getting the results and training the models is described in Attachment TODO odkaz.

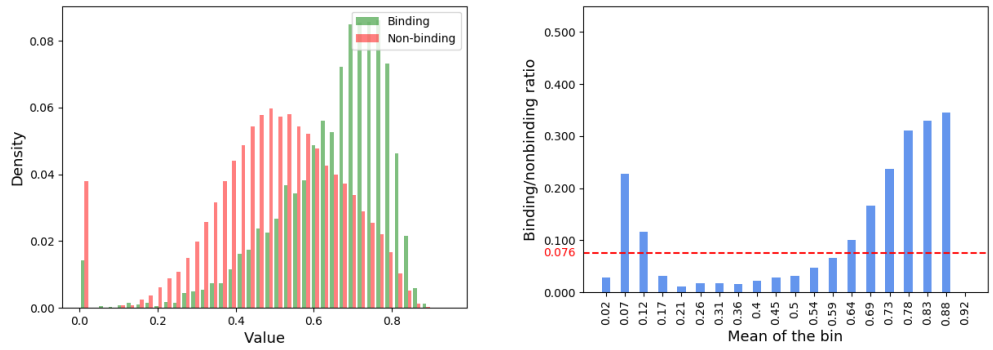
| | | conservation | INTAA_cons. | pdbekb_cons. |
|---|---|---|---|---|
| **P-value (whole sample)** | **chen11** | 0 | 0 | 0 |
| | **coach420** | 0 | 0 | 0 |
| **effect size* (whole sample)** | **chen11** | 0.7089 | 0.6411 | 0.3212 |
| | **coach420** | 0.8248 | 0.6122 | 0.3 |
| **mean P-value (sample 500)** | **chen11** | 3.06E-20 | 1.34E-11 | 3.79E-22 |
| | **coach420** | 8.07E-28 | 1.45E-12 | 1.54E-27 |
| **mean effect s.* (sample 500)** | **chen11** | 0.7387 | 0.6217 | 0.3828 |
| | **coach420** | 0.8693 | 0.5978 | 0.4295 |

Table 4.10: Results of the statistical analysis of three conservation features. The features were compares on two datasets (*chen11* and *coach420*) and the analysis was performed twice: with all the data rows, and 1000 iterations of random sampling with sample size 500, taking the same number of binding and non-binding sites. Keep in mind that the mean P-values cannot be interpreted in the original meaning of P-value and are stated here only to get the idea.
* The effect sizes for continuous features (`conservation` and `INTAA_conservation`) are measured as Cohen's $d$ value, whereas Cohen's $w$ is stated for `pdbekb_conservation`. Keep in mind that the two measures are scaled differently and the values cannot be compared directly with each other.

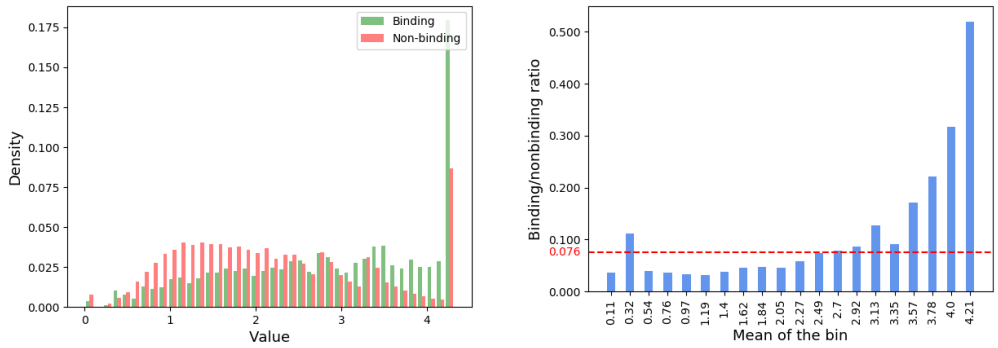
First, let's look at the results of the hypothesis testing, summarized in Table 4.10. The effect sizes as well as P-values seem to be very significant, inicating that all three features should be important for the prediction of binding sites. `conservation` seems to be the better one of the two continuous features. The effect size of the `pdbekb_conservation` cannot be compared with the two other features directly, since it is measured on a different scale. However, based on the interpretation tables described in Section 3.6.4, all three features have similar 'medium to large' effect sizes.

The visual comparison obtained from the bigger dataset `coach420` are depicted in Figure 4.7. It is hard to tell which feature should be more valuable for the prediction. `INTAA_conservation` data are more spread out, having bigger variance, which is probably the reason why the effect size is little lower than for `conservation`. On the other hand, when looking at the binding-non-binding ratios, `INTAA_conservation` should separate the data similarly as `conservation`. Based on the binding/non-binding ratios, `pdbekb_conservation`, could be the best option, but the differences are too small to draw any conclusions.

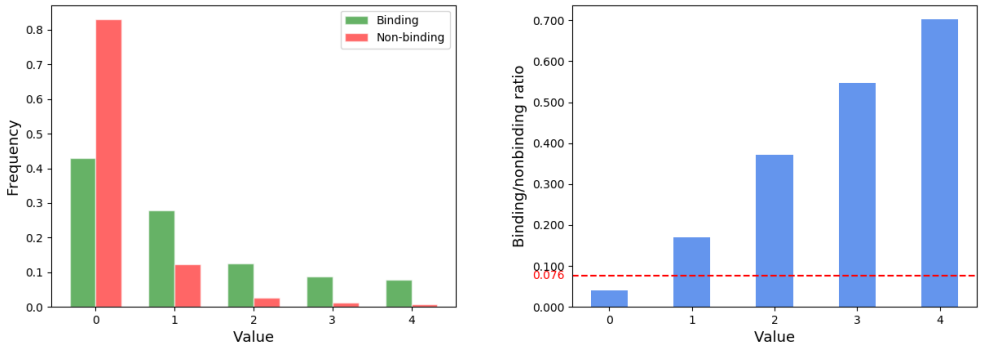
The results of the statistical analysis indicate that all three conservation features should have an impact on the binding sites prediction. The features will probably perform similarly, feature `conservation` with the biggest effect size could be slightly better. The computation of conservation scores in P2Rank




(a) `conservation`




(b) `INTAA_conservation`




(c) `pdbekb_conservation`

Figure 4.7: Visual comparison of three conservation features, obtained from dataset *coach420*. Binding/non-binding ratios are plotted on the right. The continuous data were divided into equally sized bins and for each bin, the ratio was computed as the number of binding residues having the conservation score in the corresponding bin, divided by the number of non-binding residues with the conservation score in that bin. For the ordinal feature `pdbekb_conservation`, these are simply the ratios of binding/non-binding residues for each category.

could be probably replaced with much faster INTAA-conservation tool, with a potential slight decrease of performance.

To test the correctness of the results, we trained three P2Rank models, each with a different conservation feature. The results are shown in Table **??**. All three features improved the performance of the baseline model in both metrics. That is in accordance with the statistical analysis. Neverthless, `conservation` feature seems to perform the worst. `INTAA_conservation` is slightly worse in $D_{CA}$ metric in Top-$n$ category, but is the best one in the rest of the compared categories.

| | $D_{CA}$ [4Å] | | $D_{CC}$ [4Å] | |
|---|---|---|---|---|
| | **Top-$n$** | **Top-($n$+2)** | **Top-$n$** | **Top-($n$+2)** |
| **baseline model** | 73.6 | 78.0 | 47.2 | 49.6 |
| **conservation** | 77.1 | 80.1 | 48.2 | 50.7 |
| **INTAA_conservation** | 77.6 | 81.7 | 49.1 | 52.2 |
| **pdbekb_conservation** | 78.4 | 81.3 | 48.8 | 51.1 |

Table 4.11: Performance of P2Rank models with different conservation features. The models were trained on *chen11* dataset and evaluated on *coach420* dataset.

This shows that the pipeline can be helpful for distinguishing useful features from unuseful and can suggest a set of promising features to try out; nevertheless, the results are only indicative and should not be interpreted blindly. In this case, the statistical analysis was not powerful enough to predict the small differences in the performances, but was able to indicate the importance of sequence conservation for the prediction.

To sum up, P2Rank conservation can probably be replaced with INTAA-conservation tool with possibly improved or similar performance. The advantage would be bigger speed and the need of only one sequential database (currently used tool needs three different databases downloaded locally). Another possibility is to use the pre-computed conservation scores in PDBe-KB database and clear away the necessity to install the databases locally.

# Conclusion

Conclusion.

# List of Abbreviations

| | |
|---|---|
| AA | Amino acid |
| atd | a tak dale |

# Bibliography


[1] Bio2byte group. URL https://bio2byte.be/info/. Accessed: 20.11.2020.

[2] URL https://www.uniprot.org/help/compbias. Accessed: 15.11.2020.

[3] Sequence conservation pipeline. URL https://github.com/cusbg/sequence-conservation. Accessed: 5.12.2020.

[4] URL https://www.uniprot.org/help/disulfid. Accessed: 15.11.2020.

[5] Mobidb api documentation. URL https://mobidb.bio.unipd.it/help/apidoc. Accessed: 2.12.2020.

[6] Todo. URL https://www.uniprot.org/help/mod_res. Accessed: 15.11.2020.

[7] URL https://www.uniprot.org/help/variant.

[8] URL https://www.uniprot.org/help/non_std. Accessed: 15.11.2020.

[9] Rcsb pdb: Pdb statistics, . URL https://www.rcsb.org/stats/growth/growth-released-structures. Accessed: 3.1.2021.

[10] Pdbe rest api, . URL https://www.ebi.ac.uk/pdbe/pdbe-rest-api. Accessed: 5.11.2020.

[11] Biopython api documentation. URL https://biopython.org/docs/dev/api/Bio.PDB.SASA.html. Accessed: 10.11.2020.

[12] URL https://www.uniprot.org/docs/userman.htm. Accessed: 15.12.2020.

[13] Uniprot website. URL https://www.uniprot.org/. Accessed: 16.12.2020.

[14] Chittaranjan Andrade. The p value and statistical significance: Misunderstandings, explanations, challenges, and alternatives. *Indian Journal of Psychological Medicine*, 41(3):210–215, 2019. doi: 10.4103/ijpsym.ijpsym_193_19.

[15] Gail J. Bartlett, Craig T. Porter, Neera Borkakoti, and Janet M. Thornton. Analysis of catalytic residues in enzyme active sites. *Journal of Molecular Biology*, 324(1):105–121, 2002. doi: 10.1016/s0022-2836(02)01036-7.

[16] M. L. Benson, R. D. Smith, N. A. Khazanov, B. Dimcheff, J. Beaver, P. Dresslar, J. Nerothin, and H. A. Carlson. Binding MOAD, a high-quality protein ligand database. *Nucleic Acids Research*, 36, 2007. doi: 10.1093/nar/gkm911.

[17] H. M. Berman, J. Westbrook, and Z. Feng. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 2000. doi: 10.1093/nar/28.1.235.

[18] Martin Bland. *An introduction to medical statistics*. Oxford University Press, 1987. ISBN 0192615025.

[19] Anne-Laure Boulesteix, Silke Janitza, Jochen Kruppa, and Inke R. König. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6):493–507, 2012. doi: 10.1002/widm.1072.

[20] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/a:1010933404324.

[21] Morton B. Brown and Alan B. Forsythe. Robust tests for the equality of variances. *Journal of the American Statistical Association*, 69(346):364–367, 1974. doi: 10.1080/01621459.1974.10482955.

[22] Michal Brylinski and Wei P. Feinstein. eFindSite: Improved prediction of ligand binding sites in protein models using meta-threading, machine learning and auxiliary ligands. *Journal of Computer-Aided Molecular Design*, 27 (6):551–567, 2013. doi: 10.1007/s10822-013-9663-5.

[23] John A. Capra, Roman A. Laskowski, Janet M. Thornton, Mona Singh, and Thomas A. Funkhouser. Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3d structure. *PLoS Computational Biology*, 5(12), 2009. doi: 10.1371/journal.pcbi.1000585.

[24] George Casella and Roger L. Berger. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002.

[25] Ke Chen, Marcin J. Mizianty, Jianzhao Gao, and Lukasz Kurgan. A critical comparative assessment of predictions of protein-binding sites for biologically relevant organic compounds. *Structure (London, England : 1993)*, 19: 613–621, 2011. doi: 10.1016/j.str.2011.02.015.

[26] Elisa Cilia, Rita Pancsa, Peter Tompa, Tom Lenaerts, and Wim F. Vranken. From protein sequence to dynamics and disorder with DynaMine. *Nature communications*, 4:2741, 2013. doi: 10.1038/ncomms3741.

[27] William G. Cochran. The chi2 test of goodness of fit. *The Annals of Mathematical Statistics*, 23(3):315–345, 1952. doi: 10.1214/aoms/1177729380.

[28] William G. Cochran. The combination of estimates from different experiments. *Biometrics*, 10(1):101, 1954. doi: 10.2307/3001666.

[29] Peter J. A. Cock, Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J. L. de Hoon. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics (Oxford, England)*, 25:1422–1423, 2009. doi: 10.1093/bioinformatics/btp163.

[30] Jacob Cohen. *Statistical power analysis for the behavioral sciences*. Academic Press, New York, 1977. ISBN 0121790606.

[31] Jacob Cohen. A power primer. *Psychological Bulletin*, 112(1):155–159, 1992. doi: 10.1037/0033-2909.112.1.155.

[32] PDBe-KB consortium. PDBe-KB: a community-driven resource for structural and functional annotations. *Nucleic acids research*, 48:D344–D353, 2020. doi: 10.1093/nar/gkz853.

[33] The 1000 Genomes Project Consortium. A global reference for human genetic variation. *Nature*, 526(7571):68–74, 2015. doi: 10.1038/nature15393.

[34] Pierrick Craveur, Agnel Praveen Joseph, Pierre Poulain, Alexandre G. de Brevern, and Joseph Rebehmed. Cis-trans isomerization of omega dihedrals in proteins. *Amino acids*, 45:279–289, 2013. doi: 10.1007/s00726-013-1511-3.

[35] Yifeng Cui, Qiwen Dong, Daocheng Hong, and Xikun Wang. Predicting protein-ligand binding residues with deep convolutional neural networks. *BMC Bioinformatics*, 20(1), 2019. doi: 10.1186/s12859-019-2672-1.

[36] Tianli Dai, Qi Liu, Jun Gao, Zhiwei Cao, and Ruixin Zhu. A new protein-ligand binding sites prediction method based on the integration of protein sequence conservation information. *BMC bioinformatics*, 12 Suppl 14:S9, 2011. doi: 10.1186/1471-2105-12-S14-S9.

[37] Jose M. Dana, Aleksandras Gutmanas, Nidhi Tyagi, Guoying Qi, Claire O'Donovan, Maria Martin, and Sameer Velankar. SIFTS: updated structure integration with function, taxonomy and sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins. *Nucleic Acids Research*, 47(D1):D482–D489, 2018. doi: 10.1093/nar/gky1114.

[38] B. Derrick and P. White. Why welch's test is type i error robust. *The Quantitative Methods for Psychology*, 12(1):30–38, 2016. doi: 10.20982/tqmp.12.1.p030.

[39] Jacob D. Durrant and J. Andrew McCammon. Molecular dynamics simulations and drug discovery. *BMC Biology*, 9(1), 2011. doi: 10.1186/1741-7007-9-71.

[40] Jacob D. Durrant and J. Andrew McCammon. NNScore 2.0: A neural-network receptor–ligand scoring function. *Journal of Chemical Information and Modeling*, 51(11):2897–2903, 2011. doi: 10.1021/ci2003889.

[41] Christiane Ehrt, Tobias Brinkjost, and Oliver Koch. Binding site characterization - similarity, promiscuity, and druggability. *MedChemComm*, 10 (7):1145–1159, 2019. doi: 10.1039/c9md00102f.

[42] Paul D. Ellis. *The Essential Guide to Effect Sizes*. Cambridge University Press, 2014. ISBN 0521194237.

[43] Joseph P. Romano Erich L. Lehmann. *Testing Statistical Hypotheses*. Springer New York, 2008. ISBN 0387988645.

[44] Harris et al. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. doi: 10.1038/s41586-020-2649-2.

[45] Radivojac et al. A large-scale evaluation of computational protein function prediction. *Nature Methods*, 10(3):221–227, 2013. doi: 10.1038/nmeth.2340.

[46] Tate et al. COSMIC: the catalogue of somatic mutations in cancer. *Nucleic Acids Research*, 47(D1):D941–D947, 2018. doi: 10.1093/nar/gky1015.

[47] Virtanen et al. SciPy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[48] Vincent Le Guilloux, Peter Schmidtke, and Pierre Tuffery. Fpocket: An open source platform for ligand pocket detection. *BMC Bioinformatics*, 10 (1):168, 2009. doi: 10.1186/1471-2105-10-168.

[49] Jürgen Haas, Alessandro Barbato, Dario Behringer, Gabriel Studer, Steven Roth, Martino Bertoni, Khaled Mostaguir, Rafal Gumienny, and Torsten Schwede. Continuous automated model EvaluatiOn (CAMEO) complementing the critical assessment of structure prediction in CASP12. *Proteins: Structure, Function, and Bioinformatics*, 86:387–398, 2017. doi: 10.1002/prot.25431.

[50] Thomas Hamelryck. An amino acid has two sides: a new 2D measure provides a different view of solvent exposure. *Proteins*, 59:38–48, 2005. doi: 10.1002/prot.20379.

[51] Michael J. Hartshorn, Marcel L. Verdonk, Gianni Chessari, Suzanne C. Brewerton, Wijnand T. M. Mooij, Paul N. Mortenson, and Christopher W. Murray. Diverse, high-quality test set for the validation of protein-ligand docking performance. *Journal of medicinal chemistry*, 50:726–741, 2007. doi: 10.1021/jm061277y.

[52] Bingding Huang. MetaPocket: A meta approach to improve protein ligand binding site prediction. *OMICS: A Journal of Integrative Biology*, 13(4): 325–330, 2009. doi: 10.1089/omi.2009.0045.

[53] Bingding Huang and Michael Schroeder. LIGSITEcsc: predicting ligand binding sites using the connolly surface and degree of conservation. *BMC structural biology*, 6:19, 2006. doi: 10.1186/1472-6807-6-19.

[54] David Jakubec, Jiří Vondrášek, and Robert D. Finn. 3DPatch: fast 3D structure visualization with residue conservation. *Bioinformatics (Oxford, England)*, 35:332–334, 2019. doi: 10.1093/bioinformatics/bty464.

[55] Lukas Jendele, Radoslav Krivak, Petr Skoda, Marian Novotny, and David Hoksza. PrankWeb: a web server for ligand binding site prediction and visualization. *Nucleic acids research*, 47:W345–W349, 2019. doi: 10.1093/nar/gkz424.

[56] J. Jiménez, S. Doerr, G. Martínez-Rosell, A. S. Rose, and G. De Fabritiis. DeepSite: protein-binding site predictor using 3d-convolutional neural networks. *Bioinformatics*, 33(19):3036–3042, 2017. doi: 10.1093/bioinformatics/btx350.

[57] S. Jones and J. M. Thornton. Analysis of protein-protein interaction sites using surface patches. *Journal of molecular biology*, 272:121–132, 1997. doi: 10.1006/jmbi.1997.1234.

[58] W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22:2577–2637, 1983. doi: 10.1002/bip.360221211.

[59] Konrad J. Karczewski, Ben Weisburd, Brett Thomas, Matthew Solomonson, Douglas M. Ruderfer, David Kavanagh, Tymor Hamamsy, Monkol Lek, Kaitlin E. Samocha, Beryl B. Cummings, Daniel Birnbaum, Mark J. Daly, and Daniel G. MacArthur and. The ExAC browser: displaying reference data information from over 60 000 exomes. *Nucleic Acids Research*, 45(D1): D840–D845, no 2016. doi: 10.1093/nar/gkw971.

[60] Nickolay A. Khazanov and Heather A. Carlson. Exploring the composition of protein-ligand binding sites on a large scale. *PLoS computational biology*, 9:e1003321, 2013. doi: 10.1371/journal.pcbi.1003321.

[61] Radoslav Krivák and David Hoksza. P2RANK: Knowledge-based ligand binding site prediction using aggregated local features. In *Algorithms for Computational Biology*, pages 41–52. Springer International Publishing, 2015. doi: 10.1007/978-3-319-21233-3_4.

[62] Radoslav Krivák and David Hoksza. P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *Journal of cheminformatics*, 10:39, 2018. doi: 10.1186/s13321-018-0285-8.

[63] P. M. Kroonenberg and Albert Verbeek. The tale of cochran's rule: My contingency table has so many expected values smaller than 5, what am i to do? *The American Statistician*, 72(2):175–183, 2018. doi: 10.1080/00031305.2017.1286260.

[64] Andriy Kryshtafovych, Torsten Schwede, Maya Topf, Krzysztof Fidelis, and John Moult. Critical assessment of methods of protein structure prediction (CASP)—round XIII. *Proteins: Structure, Function, and Bioinformatics*, 87(12):1011–1020, 2019. doi: 10.1002/prot.25823.

[65] J. Kyte and R. F. Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of molecular biology*, 157:105–132, 1982. doi: 10.1016/0022-2836(82)90515-0.

[66] Melissa J. Landrum, Jennifer M. Lee, Mark Benson, Garth R. Brown, Chen Chao, Shanmuga Chitipiralla, Baoshan Gu, Jennifer Hart, Douglas Hoffman, Wonhee Jang, Karen Karapetyan, Kenneth Katz, Chunlei Liu, Zenith

Maddipatla, Adriana Malheiro, Kurt McDaniel, Michael Ovetsky, George Riley, George Zhou, J Bradley Holmes, Brandi L. Kattman, and Donna R. Maglott. ClinVar: improving access to variant interpretations and supporting evidence. *Nucleic Acids Research*, 46(D1):D1062–D1067, 2017. doi: 10.1093/nar/gkx1153.

[67] A. T. R. Laurie and R. M. Jackson. Q-SiteFinder: an energy-based method for the prediction of protein-ligand binding sites. *Bioinformatics*, 21(9): 1908–1916, 2005. doi: 10.1093/bioinformatics/bti315.

[68] Howard Levene. Robust tests for equality of variances. In *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, pages 278–292. Stanford University Press.

[69] Jake Lever, Martin Krzywinski, and Naomi Altman. Classification evaluation. *Nature Methods*, 13(8):603–604, 2016. doi: 10.1038/nmeth.3945.

[70] Mingfeng Lin, Henry C. Lucas, and Galit Shmueli. Too big to fail: Large samples and thep-value problem. *Information Systems Research*, 24(4): 906–917, 2013. doi: 10.1287/isre.2013.0480.

[71] Gonzalo Lopez, Iakes Ezkurdia, and Michael L. Tress. Assessment of ligand binding residue predictions in CASP8. *Proteins: Structure, Function, and Bioinformatics*, 77(S9):138–146, 2009. doi: 10.1002/prot.22557.

[72] Thomas Lumley, Paula Diehr, Scott Emerson, and Lu Chen. The importance of the normality assumption in large public health data sets. *Annual Review of Public Health*, 23(1):151–169, 2002. doi: 10.1146/annurev.publhealth.23.100901.140546.

[73] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50–60, 1947. doi: 10.1214/aoms/1177730491.

[74] Cyrus R. Mehta and Nitin R. Patel. A network algorithm for performing fisher's exact test in r $\times$ c contingency tables. *Journal of the American Statistical Association*, 78(382):427, 1983. doi: 10.2307/2288652.

[75] S. Miller, J. Janin, A. M. Lesk, and C. Chothia. Interior and surface of monomeric proteins. *Journal of molecular biology*, 196:641–656, 1987. doi: 10.1016/0022-2836(87)90038-6.

[76] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247:536–540, 1995. doi: 10.1006/jmbi.1995.0159.

[77] Marco Necci, Damiano Piovesan, Zsuzsanna Dosztányi, and Silvio C. E. Tosatto. MobiDB-lite: fast and highly specific consensus prediction of intrinsic disorder in proteins. *Bioinformatics (Oxford, England)*, 33:1402–1404, 2017. doi: 10.1093/bioinformatics/btx015.

[78] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970. doi: 10.1016/0022-2836(70)90057-4.

[79] A. Nightingale, R. Antunes, E. Alpi, B. Bursteinas, L. Gonzales, W. Liu, J. Luo, G. Qi, E. Turner, and M. Martin. The proteins api: accessing key integrated protein and genome information. *Nucleic acids research*, 45: W539–W544, 2017. doi: 10.1093/nar/gkx237.

[80] Martin Paluszewski and Pawel Winter. Protein decoy generation using branch and bound with efficient bounding. In *Lecture Notes in Computer Science*, pages 382–393. Springer Berlin Heidelberg. doi: 10.1007/978-3-540-87361-7_32.

[81] Daniele Raimondi, Gabriele Orlando, Rita Pancsa, Taushif Khan, and Wim F. Vranken. Exploring the sequence-based prediction of folding initiation sites in proteins. *Scientific reports*, 7:8826, 2017. doi: 10.1038/s41598-017-08366-3.

[82] G. N. Ramachandran, C. Ramakrishnan, and V. Sasisekharan. Stereochemistry of polypeptide chain configurations. *Journal of molecular biology*, 7: 95–99, 1963. doi: 10.1016/s0022-2836(63)80023-6.

[83] Didier Rognan. Docking methods for virtual screening: Principles and recent advances. In *Methods and Principles in Medicinal Chemistry*, pages 153–176. Wiley-VCH Verlag GmbH & Co. KGaA, 2011. doi: 10.1002/9783527633326.ch6.

[84] Ambrish Roy, Jianyi Yang, and Yang Zhang. COFACTOR: an accurate comparative algorithm for structure-based protein function annotation. *Nucleic acids research*, 40:W471–W477, 2012. doi: 10.1093/nar/gks372.

[85] Graeme D. Ruxton. The unequal variance t-test is an underused alternative to student's t-test and the mann–whitney u test. *Behavioral Ecology*, 17 (4):688–690, may 2006. doi: 10.1093/beheco/ark016.

[86] Peter Schmidtke, Catherine Souaille, Frédéric Estienne, Nicolas Baurin, and Romano T. Kroemer. Large-scale comparison of four binding site detection algorithms. *Journal of chemical information and modeling*, 50:2191–2200, 2010. doi: 10.1021/ci1000289.

[87] Karen T. Schomburg, Stefan Bietz, Hans Briem, Angela M. Henzler, Sascha Urbaczek, and Matthias Rarey. Facing the challenges of structure-based target prediction by inverse virtual screening. *Journal of Chemical Information and Modeling*, 54(6):1676–1686, 2014. doi: 10.1021/ci500130e.

[88] Pak C. Sham and Shaun M. Purcell. Statistical power and significance testing in large-scale genetic studies. *Nature Reviews Genetics*, 15(5):335–346, apr 2014. doi: 10.1038/nrg3706.

[89] A. Shrake and J. A. Rupley. Environment and exposure to solvent of protein atoms. lysozyme and insulin. *Journal of molecular biology*, 79:351–371, 1973. doi: 10.1016/0022-2836(73)90011-9.

[90] J. Skolnick and M. Brylinski. FINDSITE: a combined evolution/structure-based approach to protein function prediction. *Briefings in Bioinformatics*, 10(4):378–391, 2009. doi: 10.1093/bib/bbp017.

[91] Christoph A. Sotriffer, Paul Sanschagrin, Hans Matter, and Gerhard Klebe. SFCscore: Scoring functions for affinity prediction of protein-ligand complexes. *Proteins: Structure, Function, and Bioinformatics*, 73(2):395–419, 2008. doi: 10.1002/prot.22058.

[92] Zhoutong Sun, Qian Liu, Ge Qu, Yan Feng, and Manfred T. Reetz. Utility of B-Factors in protein science: Interpreting rigidity, flexibility, and internal motion and engineering thermostability. *Chemical reviews*, 119:1626–1665, 2019. doi: 10.1021/acs.chemrev.8b00290.

[93] Kuan Pern Tan, Raghavan Varadarajan, and M. S. Madhusudhan. DEPTH: a web server to compute depth and predict small-molecule binding cavities in proteins. *Nucleic acids research*, 39:W242–W248, 2011. doi: 10.1093/nar/gkr356.

[94] The UniProt Consortium. UniProt: the universal protein knowledgebase. *Nucleic acids research*, 45:D158–D169, 2017. doi: 10.1093/nar/gkw1099.

[95] Eldon L. Ulrich, Hideo Akutsu, Jurgen F. Doreleijers, Yoko Harano, Yannis E. Ioannidis, Jundong Lin, Miron Livny, Steve Mading, Dimitri Maziuk, Zachary Miller, Eiichi Nakatani, Christopher F. Schulte, David E. Tolmie, R. Kent Wenger, Hongyang Yao, and John L. Markley. BioMagResBank. *Nucleic acids research*, 36:D402–D408, 2008. doi: 10.1093/nar/gkm957.

[96] Mark N. Wass and Michael J. E. Sternberg. Prediction of ligand binding sites using homologous structures and conservation at CASP8. *Proteins*, 77 Suppl 9:147–151, 2009. doi: 10.1002/prot.22513.

[97] Martin Weisel, Ewgenij Proschak, and Gisbert Schneider. PocketPicker: analysis of ligand binding-sites with shape descriptors. *Chemistry Central Journal*, 1(1), 2007. doi: 10.1186/1752-153x-1-7.

[98] B. L. Welsch. THE GENERALIZATION OF 'STUDENT's' PROBLEM WHEN SEVERAL DIFFERENT POPULATION VARLANCES ARE INVOLVED. *Biometrika*, 34(1-2):28–35, 1947. doi: 10.1093/biomet/34.1-2.28.

[99] Lei Xie, Li Xie, and Philip E. Bourne. Structure-based systems biology for analyzing off-target binding. *Current Opinion in Structural Biology*, 21(2): 189–199, 2011. doi: 10.1016/j.sbi.2011.01.004.

[100] Jianyi Yang, Ambrish Roy, and Yang Zhang. Protein-ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment. *Bioinformatics (Oxford, England)*, 29:2588–2595, 2013. doi: 10.1093/bioinformatics/btt447.

[101] Zengming Zhang, Yu Li, Biaoyang Lin, Michael Schroeder, and Bingding Huang. Identification of cavities on protein surface using multiple computational approaches for drug binding site prediction. *Bioinformatics (Oxford, England)*, 27:2083–2088, 2011. doi: 10.1093/bioinformatics/btr331.

[102] Jingtian Zhao, Yang Cao, and Le Zhang. Exploring the computational methods for protein-ligand binding site prediction. *Computational and Structural Biotechnology Journal*, 18:417–426, 2020. doi: 10.1016/j.csbj.2020.02.008.

[103] Xiliang Zheng, LinFeng Gan, Erkang Wang, and Jin Wang. Pocket-based drug design: Exploring pocket space. *The AAPS Journal*, 15(1):228–241, 2012. doi: 10.1208/s12248-012-9426-6.

[104] Donald W. Zimmerman. Invalidation of parametric and nonparametric statistical tests by concurrent violation of two assumptions. *The Journal of Experimental Education*, 67(1):55–68, 1998. doi: 10.1080/00220979809598344.

[105] Donald W. Zimmerman. A note on preliminary tests of equality of variances. *British Journal of Mathematical and Statistical Psychology*, 57(1): 173–181, 2004. doi: 10.1348/000711004849222.

[106] Donald W. Zimmerman and Bruno D. Zumbo. Rank transformations and the power of the student t test and welch t' test for non-normal populations with unequal variances. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, 47(3):523–539, 1993. doi: 10.1037/h0078850.

[107] Bruno D. Zumbo and Daniel Coulombe. Investigation of the robust rank-order test for non-normal populations with unequal variances: The case of reaction time. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, 51(2):139–150, 1997. doi: 10.1037/1196-1961.51.2.139.

# List of Figures


2.1 Visualization of SAS points for structure 1FBL. Each point is colored by its predited ligandability score (red points are the most ligandable, green points the least). Predicted binding sites are marked by coloured protein surface. Adapted from: *P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure* [62] . . . . . 11

3.1 Diagram of the pipeline structure. . . . . 13
3.2 Illustration of the solvent accessible surface. It was created by rolling the probe (in blue) along the molecule surface and tracing the center of the probe. Retrieved 02-01-2020 from `https://commons.wikimedia.org/wiki/File:Surfacetype_Solvent-Accessible.png` . . . . . 17
3.3 Half sphere exposure. Retrieved 02-01-2020 from `https://en.wikipedia.org/wiki/File:HSECa.png` . . . . . 22
3.4 Polypeptide torsion angles phi, psi and omega. Retrieved 02-01-2020 from `https://www.researchgate.net/figure/Backbone-torsion-angles-of-fig2_284713304` . . . . . 23

4.1 Histogram for feature `dynamine` . . . . . 39
4.2 P-value deflation . . . . . 40
4.3 Conservation features . . . . . 45
4.4 The features related to the residue 'buriedness' have higher values in binding sites. . . . . 46
4.5 `bfactor`: Binding sites seem to have lower B factor values, which could indicate bigger rigidity of binding sites. . . . . 47
4.6 Feature `aa`. . . . . 48
4.7 Visual comparison of three conservation features, obtained from dataset *coach420*. Binding/non-binding ratios are plotted on the right. The continuous data were divided into equally sized bins and for each bin, the ratio was computed as the number of binding residues having the conservation score in the corresponding bin, divided by the number of non-binding residues with the conservation score in that bin. For the ordinal feature `pdbekb_conservation`, these are simply the ratios of binding/non-binding residues for each category. . . . . 57

# List of Tables


3.1 Summary of analysed features. . . . . 16
3.2 Type I and II Error in hypothesis testing. . . . . 26
3.3 A 2×2 contingency table for binary feature `aromaticity` computed on dataset Chen11. . . . . 29

4.1 Summary of dataset properties with and without ligands filtering 36
4.2 P-values returned by hypothesis tests for individual features . . . 38
4.3 Values of Cohen's $d$ . . . . . 41
4.4 Values of Cohen's $w$ . . . . . 42
4.5 Mean effect sizes (cohen's $d$) . . . . . 44
4.6 Mean effect sizes (cohen's $w$) . . . . . 45
4.7 Comparison of the performance of models with different sets of features: P2Rank default features (baseline model), csv features and both sets together. . . . . 50
4.8 Performances of models trained with individual csv features used together with all P2Rank features. The features are sorted according to the performances in the first column. . . . . 52
4.9 Importances of csv features returned by the Random Forests algorithm. The model was trained with all csv features and without P2Rank features. . . . . 54
4.10 Results of the statistical analysis of three conservation features. The features were compares on two datasets (*chen11* and *coach420*) and the analysis was performed twice: with all the data rows, and 1000 iterations of random sampling with sample size 500, taking the same number of binding and non-binding sites. Keep in mind that the mean P-values cannot be interpreted in the original meaning of P-value and are stated here only to get the idea. * The effect sizes for continuous features (`conservation` and `INTAA_conservation`) are measured as Cohen's $d$ value, whereas Cohen's $w$ is stated for `pdbekb_conservation`. Keep in mind that the two measures are scaled differently and the values cannot be compared directly with each other. . . . . 56
4.11 Performance of P2Rank models with different conservation features 58

# A. Attachments

## A.1 attachment1

TODO

The procedure was following:

- We installed INTAA-conservation tool and downloaded Swiss-Prot database, as described in the project repository (`https://github.com/davidjakubec/INTAA-conservation`). Conservation scores for all structures in datasets *chen11* and *coach420* were calculated.
- Similarly, calculate conservation scores using P2Rank conservation pipeline. The usage is described at `https://github.com/cusbg/sequence-conservation`.
- New dataset files `chen11_conservation.txt` and `coach420_conservation.txt` were created.
- We created new config file `config_conservation.json` with definitions of three conservation features. The file is included in Attachment TODO.
- We implemented new feature `INTAA_conservation` in class `Features.Custom.INTAACons` (TODO odkaz na attachments).
- Since conservation scores for `conservation` and `INTAA_conservation` were computed locally and the computation was not included directly in the pipeline, these features were implemented so that they read and parse the files obtained by INTAA-conservation tool and supply values for individual residues to the pipeline. For this reason, before running the pipeline we must create the output folder (named as the dataset file) and place the pre-computed files into the subfolders named 'conservation' and 'INTAA_conservation'.
- The following command was run to download the PDB files, compute the features, perform the statistical analysis and train and evaluate three P2Rank models with each conservation feature: `bash scripts/pipeline_P2Rank_oneFea -t data/datasets/chen11_conservation.txt -e data/datasets/coach420_conserv -c scripts/source/config_conservation.json -l 10 -m 4 -f conservation,INTA`
- TODO recompute analysis sample 500

## A.2 attachment2

blabla