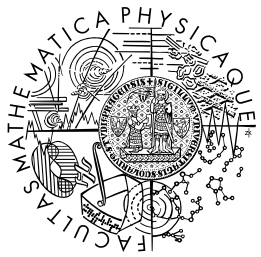


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

DOCTORAL THESIS

Mgr. Radoslav Krivák

Prediction of ligand binding sites from protein structure

Department of Software Engineering

Supervisor of the doctoral thesis: doc. RNDr. David Hoksza, Ph.D.

Study programme: Computer Science

Specialization: Software Systems

Prague 2023

I declare that I carried out this doctoral thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

First of all, I would like to thank David Hoksza for supervising my work, for providing constant support and inspiration and ultimately for introducing me to the exciting field of structural bioinformatics. I am also thankful to all my friends, collaborators and colleagues that helped in any way to improve my work, either by direct help or by providing helpful suggestions and inspiration.

Dedicated to my grandparents: Brigita, Sabína, Andrej and František.

Title: Prediction of ligand binding sites from protein structure

Author: Mgr. Radoslav Krivák

Department: Department of Software Engineering

Supervisor: doc. RNDr. David Hoksza, Ph.D., Department of Software Engineering

Abstract:

Ligand binding site prediction from protein structure is a fundamental problem in the field of structural bioinformatics that has many applications related to the elucidation of protein function and structure-based drug discovery. The first focus of this thesis was the application of machine learning to this and related problems. The second focus was the development of a practically usable tools based on our research. The machine learning based tools produced as a result of the work on this thesis include the pocket re-scoring method PRANK, a stand-alone ligand binding site prediction method P2Rank (together with its extended web interface PrankWeb) and the peptide binding prediction method P2Rank-Pept. We have shown that our methods outperformed available state-of-the-art tools while providing other benefits like prediction speed and stability. Furthermore, we have developed AHoJ, a flexible tool for the search and alignment of Apo-Holo protein pairs in the PDB. AHoJ that is ideal for creating Apo-Holo datasets which can in turn help to better evaluate binding site prediction methods in the future.

Keywords: Structural Bioinformatics, Protein-ligand binding sites, Machine learning

Contents

Part I

Commentary

Chapter 1

Introduction

1.1 Structure of the thesis

The thesis is structured in the following way. Part I: Commentary summarizes my work and contribution and puts it in the context while Part II: Publications contains the full text of ten peer-reviewed co-authored publications that constitute the core of the contribution and were published during my PhD study.

In Part II, some of the publications are introduced by a concise "Author's highlights". These are not necessarily summaries of the articles and are not meant to replace abstracts but rather highlight points that might be relevant to the readers of the thesis.

Instead of trying to be just a summary of included publications, the text of the thesis is intended to be an accompanying commentary to my work as a whole with some added value. The thesis contains some of my personal opinions and experiences and better explains the motivation behind some efforts and decisions. This includes some points that did not find a way to the original publications or could be only said with hindsight. Furthermore, while the included papers describe the software as it was when it was initially released, this thesis describes the software as it is now, with all accumulated improvements and changes.

1.2 Binding site prediction and related problems

Ligand binding site prediction is a fundamental problem in the field of computational biology that seeks to identify the location and shape of binding sites on protein structures that can interact with small molecules. This section contains a concise introduction to the problem and its context. The main goal is, however, to highlight inherent complications with the problem definition and bring up considerations that shaped the work presented in this thesis.

1.2.1 Motivation

Prediction of ligand binding sites from protein structure has many applications in elucidation of protein function [KJ14] and rational drug design [ZGWW12, PSM*10, TBNT16]. It has been employed in drug side-effects prediction [XXB11], fragment-based drug discovery [LEG16], docking prioritization [LJ06, FB15], structure based virtual screening [LSCZ14] and structure-based target prediction (or so called inverse virtual screening) [SBB*14]. Increasingly it is being used in large-scale structural studies that try to analyze and compare all known and putative binding sites on a genome-wide or PDB-wide level [DWH15, MBB16, MZF*17, SCS*17, BSSC18].

In practice, it is often the case that predicting ligand binding sites is not an end in itself but it represents only a step in a larger automated solution or pipeline. For instance, a druggability prediction server PockDrug-Server [HBG*15] relies on ligand binding site prediction internally. Similarly, allosteric site prediction tools Allosite [HLH*13] and AlloPred [HLH*13] both internally employ a ligand binding site prediction tool Fpocket [LGST09] as the first step of their workflows.

1.2.2 The problem statement

The problem of ligand binding site prediction from protein structure can be defined in the following way: given a protein structure, produce a list of putative binding sites and score/order them according to the likelihood of binding relevant ligands.

This definition is rather technical but still leads to several questions:

How can/should be predicted binding sites represented? It turns out that in whatever way possible and that the existing methods represent binding sites in various ways, which include but are not limited to: a set of protein surface atoms, a set of residues or a set of points around the surface of the protein (points on a regular 3D grid, alpha sphere centers or points on protein's solvent accessible surface). To evaluate a prediction method we need a binding site to be represented at least as a single point, i.e. center/centroid of a binding site.

Why it is important to score/order predicted binding sites? To meaningfully evaluate prediction methods and to determine their identification success rate it is necessary to consider only predicted sites with the highest score (e.g. Top-1/Top3 or better Top-n/Top-(n+2) where n is the number of known ligands on a given protein). If we were to consider all predicted pockets, an obviously useless method that would cover the whole surface of the protein with predicted binding sites would achieve 100% success rate.

Which types of ligands are relevant? This is often only implicitly defined by the datasets on which are particular methods trained and/or benchmarked. For a detailed discussion see Supplementary Materials to [KH18].

1.2.3 Related problems

Proteins can interact with a variety of binding partners: small molecule ligands, ions, peptides, other proteins and nucleic acids. For each type of binding partner, we can consider the problem of predicting its binding locations. Developing a prediction method for each of those molecular types presents distinct challenges and also offers specific clues that can be best utilized by specialized methods.

In contrast with the task of binding site prediction, there is a closely related task of binding residue prediction. Although the difference may seem only technical, it is important to distinguish between the two. The task of binding site prediction involves the prediction of binding sites as such, i.e. a binding site is considered an entity which shape and location (represented at least as a center point) needs to be determined. On the other hand, the task of binding residue prediction can be viewed as a task of labeling residues by a binary label (binding vs. non-binding), or by a binding probability score from the range of $[0, 1]$. One way to look at is that the task of binding residue prediction does not include the final step of clustering binding residues into binding sites.

An important variation of the problem is predicting binding residues from the sequence alone.

1.2.4 Existing methods and tools

Ligand binding site prediction methods have been in development for almost 40 years now (the first known method, to my knowledge, was published in 1985). During this time more than 50 different algorithms or improvements have been published.

Existing methods for ligand binding site prediction are based on a variety of algorithmic approaches. Traditionally, methods have been categorized based on their main algorithmic strategy into geometric, energetic, conservation based, template based, knowledge based and machine learning based. In reality, many of the existing tools are based on some combination of the mentioned approaches. Methods based on a consensus of results of other algorithms have also emerged.

More details on existing methods and tools can be found in numerous reviews and surveys [LJ06, HOH*10, PSM*10, LSZ10, CMGK11, FRH11, RBJ15, BS17, SLD*]. In the introduction to the paper [KH18] I have provided another comprehensive survey of existing tools with a focus on their practical usability. In it I have highlighted the importance of the categorization of the tools along several lines: template based / template-free methods, web servers / stand-alone tools, and residue-centric / pocket-centric methods and I have argued that there is a strong case for a new fast stand-alone user-friendly and template-free tool.

Studies that introduced existing methods reported relatively high prediction accuracy, usually on traditional small datasets. However, the results of the only independent benchmark [CMGK11] suggested that existing methods may not be as accurate as previously believed when applied to new datasets.

When I started working on the problem at the first sight it might seem that the field is crowded with tools available for researchers. However, after a closer survey [KH18] I found that only a few of the published methods were available as a stand-alone software that can be used locally (in contrast with web-based methods). Furthermore, most of those stand-alone tools were unnecessarily complicated to use (users were required to perform preprocessing tasks that could have been automated by the authors of the software). Even fewer of the tools were available as open-source software.

1.3 Goals

There is no reason to pretend that the work presented in this thesis was a liner process of first setting some fixed set of goals and then gradually accomplishing them. Indeed, what is included in the thesis is mostly only the work that led to in some way successful results. With that in mind, the following list is included here mainly to clarify my intentions and motivations and highlight the issues of existing tools I decided to focus on improving.

- Explore the possibility of improving existing ligand binding site prediction methods by replacing their scoring function.
- Develop a stand-alone ligand binding site prediction method based on machine learning. Although machine learning has been applied to the problem before and some studies have been published, their focus was on predicting binding residues rather than on predicting binding sites as such [KK09, QW00, CHG14].
- Produce command line tools that can be used locally and are easy to set up and use and therefore can be easily employed in larger bioinformatics pipelines.
- Produce intuitive web based tools with integrated visualizations that have documented REST APIs.
- Work towards a better evaluation of ligand binding site prediction methods on Apo-Holo datasets.

Chapter 2

Overview of the contribution

2.1 List of Publications

The following peer-reviewed publications and associated structural bioinformatics software constitute the core contribution presented in this thesis. Full texts of these publications (except [con19, con21]) including relevant supplementary materials are included in Part II.

[KH15a] KRIVÁK R., HOKSZA D.: **Improving protein-ligand binding site prediction accuracy by classification of inner pocket points using local features**. *Journal of Cheminformatics 7*, 1 (Apr 2015), 12. doi:10.1186/s13321-015-0059-5

[KH15b] KRIVÁK R., HOKSZA D.: **P2RANK: Knowledge-Based Ligand Binding Site Prediction Using Aggregated Local Features**. In *International Conference on Algorithms for Computational Biology* (2015), Springer, pp. 41–52. doi:10.1007/978-3-319-21233-3_4

[KH18] KRIVÁK R., HOKSZA D.: **P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure**. *Journal of cheminformatics 10*, 1 (2018), 39. doi:10.1186/s13321-018-0285-8

[KH7] KRIVÁK R., HOKSZA D., ŠKODA P.: **Improving quality of ligand-binding site prediction with Bayesian optimization**. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2017), pp. 2278–2279. doi:10.1109/BIBM.2017.8218024

[KJH18] KRIVÁK R., JENDELE L., HOKSZA D.: **Peptide-Binding Site Prediction From Protein Structure via Points on the Solvent Accessible Surface**. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics* (New York, NY, USA, 2018), BCB '18, Association for Computing Machinery, p. 645–650. doi:10.1145/3233547.3233708

[JKS*19] JENDELE L., KRIVAK R., SKODA P., NOVOTNY M., HOKSZA D.: **PrankWeb: a web server for ligand binding site prediction and visualization**. *Nucleic Acids Res. 47*, W1 (Jul 2019), W345–W349. doi:10.1093/nar/gkz424

[JSK*22] JAKUBEC D., SKODA P., KRIVAK R., NOVOTNY M., HOKSZA D.: **PrankWeb 3: accelerated ligand-binding site predictions for experimental and modelled protein structures**. *Nucleic Acids Research 50*, W1 (05 2022), W593–W597. doi:10.1093/nar/gkac389

[con19] CONSORTIUM P.-K.: **PDBe-KB: a community-driven resource for structural and functional annotations**. *Nucleic Acids Research 48*, D1 (10 2019), D344–D353. doi:10.1093/nar/gkz853

[con21] CONSORTIUM P.-K.: **PDBe-KB: collaboratively defining the biological context of structural data**. *Nucleic Acids Research 50*, D1 (11 2021), D534–D542. doi:10.1093/nar/gkab988

[FKHN22] FEIDAKIS C. P., KRIVAK R., HOKSZA D., NOVOTNY M.: **AHoJ: rapid, tailored search and retrieval of apo and holo protein structures for user-defined ligands**. *Bioinformatics 38*, 24 (10 2022), 5452–5453. doi:10.1093/bioinformatics/btac701

2.1.1 Autorship notes

In the publications where I am the first author [KH15a, KH15b, KH18, KH7, KJH18] I have contributed most of the research ideas and software development, performed the experiments and I have written most of the text of the manuscripts (all under the supervision and with consultation with my supervisor David Hoksza following his initial ideas about an aggregated representation of protein physico-chemical features).

In the publications related to the web interface (PrankWeb) [JKS*19, JSK*22] I have contributed some of the development, performed the experi-

ments [JKS*19] or helped with their design [JSK*22] and written parts of the manuscript [JKS*19].

Publications related to PDB-KB [con19, con21] were written by a consortium of authors and P2Rank is only one of the tools integrated with PDB-KB. I have helped to develop data transformation of P2Rank output to PDB-KB input format, contributed to the validator of PDB-KB input data and performed predictions on all proteins in the PDB.

In [FKHN22] I have developed the web interface and contributed to the development of the command line version of the software.

2.2 Summary of the contribution

This section summarizes the most important contributions of the work presented in this thesis. Most of the work was produced in cooperation with co-authors of respective publications.

A list of released bioinformatics software and practical/usable contributions follows.

1. We have developed PRANK, a machine learning based method that allows to re-score (re-rank) ligand binding sites predicted produced by other methods. Since it helps true binding sites to be ranked higher, it improves the applicability and usefulness of their predictions. PRANK is useful especially in combination with methods like Fpocket, which produce a large amount of predicted binding sites for each protein but do not always score true binding sites at the top. PRANK was made available as a free command line tool with source code available upon request. Later it became part of the P2Rank codebase and was released as open-source software.

2. We have developed P2Rank, a fully independent method for ligand binding site prediction based on machine learning. Although some machine learning based methods for a given problem were described in the literature before, to our knowledge P2Rank was the first pragmatically usable tool for ligand binding site prediction based on machine learning. P2Rank makes predictions by scoring and clustering points on the protein's solvent accessible surface. The ligandability score of individual points is determined by a Random Forest model trained on the dataset of known protein-ligand complexes. P2Rank

is released as open-source software (under MIT license) on GitHub (https://github.com/rdk/p2rank).

3. We have developed PrankWeb, a web application interface for P2Rank [JKS*19]. In addition to a standalone version of P2Rank, PrankWeb employs a custom-made conservation pipeline and improved prediction models trained using conservation as one of the features (i.e. descriptors). Unlike many similar tools at the time of the release, PrankWeb came with a documented REST API. The later version introduced the support for mmCIF format and prediction model specialized for AlphaFold structures [JSK*22]. PrankWeb is freely available at https://prankweb.cz/ and open-sourced (under Apache License 2.0) on GitHub (https://github.com/cusbg/prankweb).

4. We have integrated P2Rank/PrankWeb with EBI's Protein Data Bank in Europe – Knowledge Base (PDBe-KB), the new PDBe's major resource of integrated protein data [con19, con21]. PDB-KB now contains annotations based on P2Rank predictions precomputed for almost every protein in the PDB and it is being periodically updated with predictions on new proteins. PDBe-KB is available at https://pdbe-kb.org.

5. We have developed AHoJ, a highly-configurable tool for the search and alignment of Apo-Holo protein pairs in the PDB [FKHN22]. AHoJ is available as an open-source command line program and a web application that allows running searches for multiple queries at the same time (and thus produce Apo-Holo datasets) and includes integrated web-based visualization. The web application is freely available at http://apoholo.cz/ and the command line tool is open-sourced (under Apache License 2.0) on GitHub (https://github.com/cusbg/AHoJ-project).

6. I have developed FasterForest, a Java library that contains two highly optimized Random Forest implementations. These implementations represent mainly technical optimizations of the previous original open-source work [Sup13, Sel17] and require roughly 75% time and 50% space compared to the original implementations. The library was used during the development and optimization of our later methods [KJH18, JKS*19]. FasterForest library is available as open source under GNU GPL v2 (https://github.com/rdk/FasterForest).

The following list summarizes my research contributions, i.e. theoretically interesting results or novel contributions to the discussion in the field of binding site prediction.

1. P2Rank was the first machine learning based method related to a protein structure that internally used points on the solvent accessible surface of the protein instead of a typical approach of using points on a regular 3D grid.

2. In publication [KH18] I introduced some points that I believe were missing from the discussion in the field. These include the following: running times (i.e. speed) of prediction methods, we highlighted the difference between pocket-centric and residue-centric methods and respective evaluation methodologies, and included a discussion of the possibility of reaching Bayes optimal rate on inherently noisy datasets.

3. During the development of the prediction methods, I used the technique of Bayesian optimization [BCdF09] that allowed me to optimize several arbitrary parameters simultaneously.

4. We have developed and published the results of P2Rank-Pept, a method specialized for the prediction of peptide binding sites from protein structure. This demonstrated the applicability of our general approach to different related tasks P2Rank-Pept is a part of the P2Rank codebase, but up to this date it has not been released with a pre-trained model.

Chapter 3

Tools for ligand binding site prediction

3.1 PRANK: replacing the scoring function of existing methods

Most of the existing ligand binding site prediction methods find much more pockets on a given structure than there are actual true binding sites. At the same time, they employ a fairly simple ranking function leading to sub-optimal prediction results[1].

To address this problem, we introduced a novel machine learning-based pocket ranking algorithm called PRANK (Pocket RANKing) that can be used post-processing step which improves the performance of existing ligand binding site prediction methods. The outline of the algorithm is shown in Figure 3.1 and further described in Figure 3.2 which shows an internal pocket representation used by PRANK. A detailed description of the algorithm can be found in [KH15a].

Our benchmarks showed that our new scoring function considerably outperformed the native scoring functions of Fpocket [LGST09] and ConCavity [CLT*09] on all evaluated datasets. Furthermore, we showed that it outperformed two simpler scoring functions: PLB index, which is based on amino acid composition [SSKH07] and a simple ordering by pocket volume. Improvements in the prediction success rate achieved by PRANK

[1] measured as binding site prediction success rate considering Top-k predicted pockets with the highest score

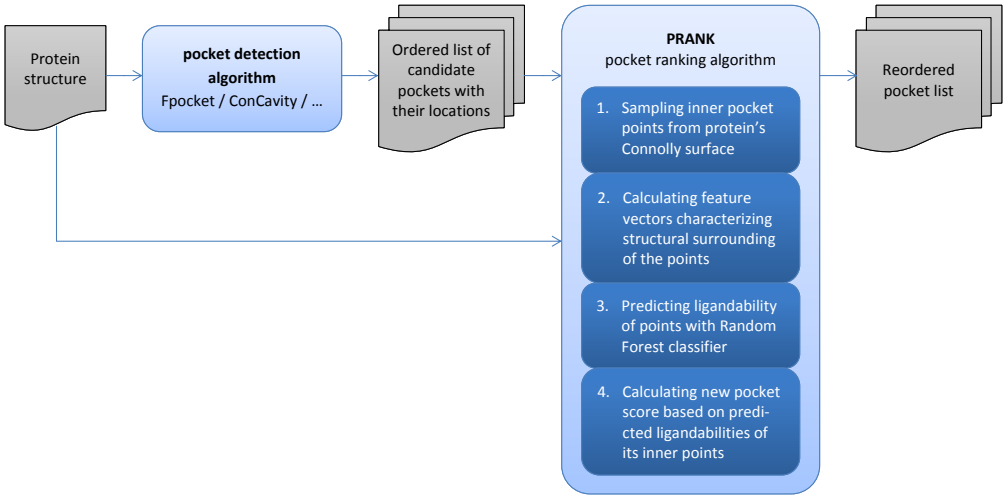


FIGURE 3.1: **Flowchart that outlines PRANK algorithm.**

when applied to Fpocket predictions can be seen in Figure 3.3.

PRANK takes a protein structure and the output of a third-party prediction method on the input and produces a list of re-scored and re-ranked pockets on the output. PRANK can currently process the output of the following methods: Fpocket , ConCavity , SiteHound [GS09], MetaPocket 2.0 [ZLL*11], LISE [XH12] and DeepSite [JDMR*17]. Furthermore, a clean internal API allows parsers for new methods to be easily implemented.

PRANK was originally developed and distributed as a set of scripts written in Groovy programming language and later integrated into the codebase and distribution of P2Rank as a standalone command line application running on Java Virtual Machine.

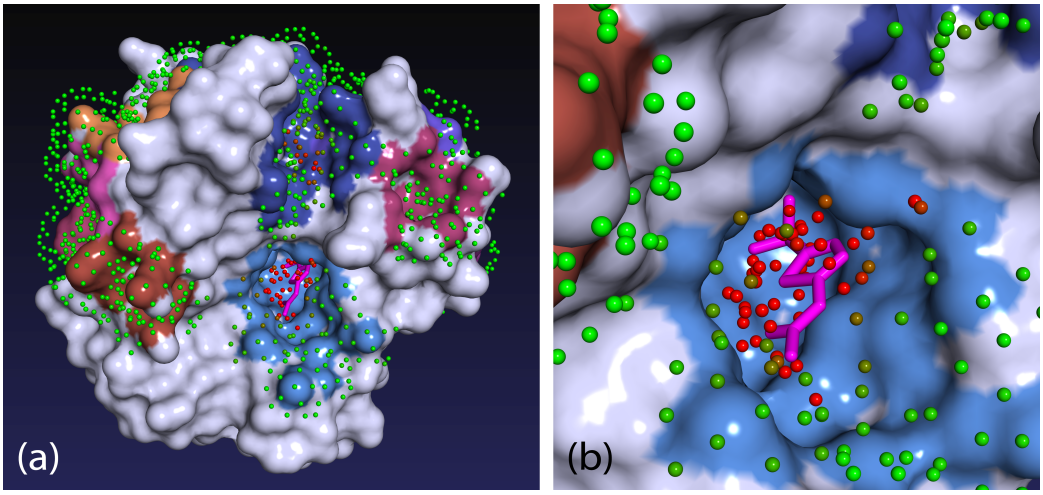


FIGURE 3.2: **PRANK: Visualization of inner pocket points. (a)** Displayed is the protein 1AZM bound to one ligand (magenta). Fpocket predicted 13 pockets that are depicted as colored areas on the protein surface. To rank these pockets, the protein was first covered with evenly spaced points on a solvent accessible surface (probe radius 1.6 Å) and only the points adjacent to one of the pockets were retained. The colour of the points reflects their ligandability (green = 0...red = 0.7) predicted by Random Forest classifier. PRANK algorithm rescores pockets according to the cumulative ligandability of their corresponding points (calculated as a sum of squares). Note that there are two clusters of ligandable (red) points in the picture, one located in the upper dark-blue pocket and the other in the light-blue pocket in the middle. The light-blue pocket, which is, in fact, the true binding site, contains more strongly ligandable points and therefore will be ranked higher. **(b)** Detailed view of the binding site with the ligand and the inner pocket points.

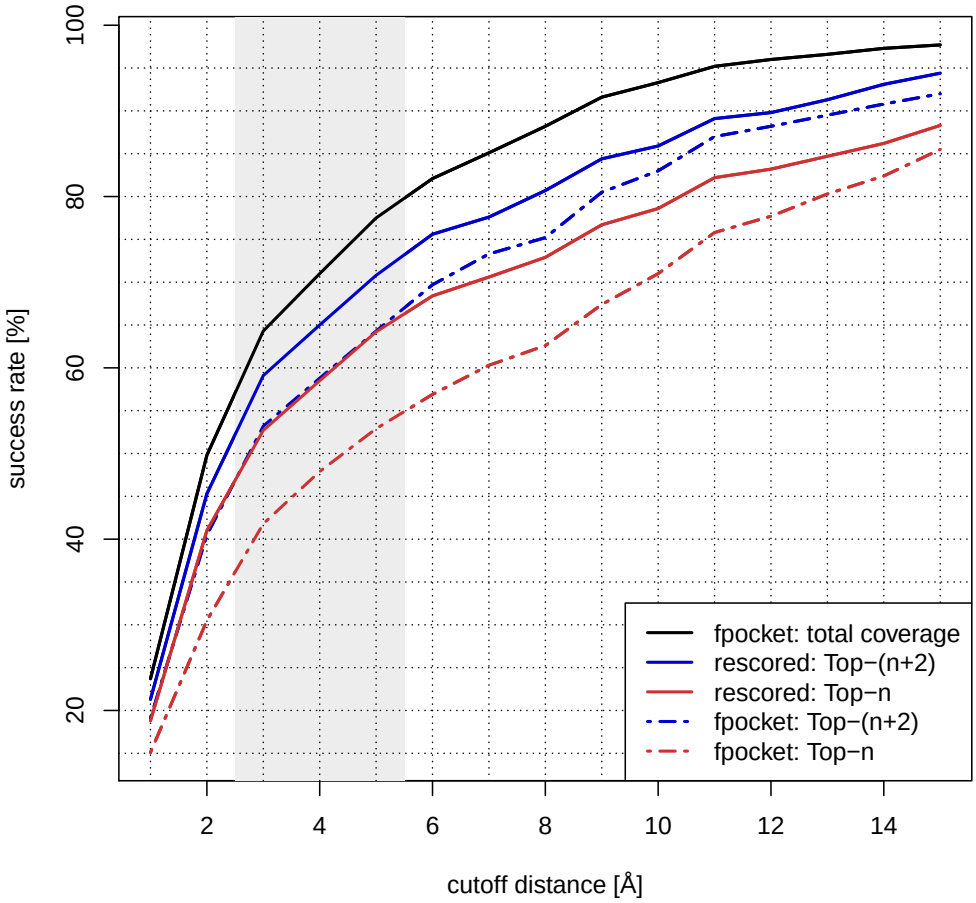


FIGURE 3.3: **PRANK: Results of rescoring Fpocket predictions on CHEN11 dataset.** Chart showing prediction success rates of Fpocket compared with results rescored by PRANK on CHEN11 dataset considering Top-n, Top-(n+2) and all pockets (total coverage). The success rate is measured by D_{CA} criterion for the range of integer cutoff distances (i.e. distance between the center of a predicted pocket and any atom of the ligand). Displayed results for rescored pockets are averaged from ten independent 5-fold cross-validation runs.

3.2 P2Rank: machine learning based method

Building on PRANK we have developed P2Rank a stand-alone independent ligand binding site prediction method. We have realized that relying on third-party methods for making predictions and then rescoring them is actually limiting and that our machine learning based approach can predict that the other methods are not able to identify at all. Compared to PRANK, P2Rank is looking at the whole surface of the protein. It covers it with points on a solvent accessible surface, predicts their ligandability and then clusters points with high ligandability into predicted binding sites. The working of the algorithm is illustrated in Figure 3.4 which shows an entire surface of the protein covered with points with predicted ligandability. A detailed description of the algorithm can be found in [KH18].

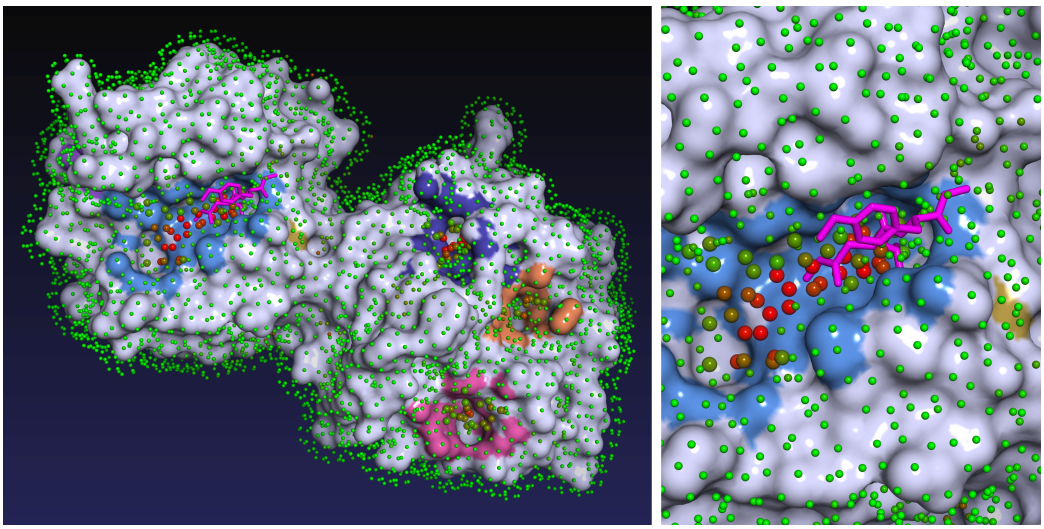


FIGURE 3.4: **P2Rank: Visualization of ligand binding sites predicted by for structure 1FBL.** Protein is covered by a layer of points lying on the Solvent Accessible Surface of the protein. Each point represents its local chemical neighborhood and is colored according to its predicted ligandability score (from 0=green to 1=red). Points with high ligandablity score are clustered to form predicted binding sites (marked by coloring adjacent protein surface). In this case, the largest predicted pocket (shown in the close-up) is indeed a correctly predicted true binding site that binds a known ligand (magenta). Visualization is based on a PyMOL script produced by P2Rank.

3.2.1 Features

This section contains a summary of the features and characteristics of the software from the point of view of a user and from the point of view of a new model/method developer. The current version of the software is described (P2Rank 2.4).

User facing features

- Ease of setup. P2Rank is distributed as a precompiled binary package with pre-trained prediction models that requires no compilation or installation. P2Rank does not depend on any third-party bioinformatics software and the only dependency is Java Virtual Machine.
- Ease of use. Given any protein structure, P2Rank is able to produce prediction by running a single command (i.e. no preprocessing steps or multiple-step procedures are needed). This is still quite rare among available methods.
- High prediction accuracy, especially when compared to methods that are comparably fast.
- PyMol visualisation. P2Rank optionally produces PyMol visualizations such as the one that can be seen in Figure 3.4.
- Optimized multi-threaded implementation. P2Rank is only one of two methods that need under one second to generate a prediction on a single protein of average size [KH18].
- Support for both PDB and mmCIF formats. P2Rank is one of the few existing ligand binding site prediction methods that are currently able to process mmCIF format and produce predictions on proteins of unlimited size as well as on AlphaFold models.
- Stability. Great care has been taken so that P2Rank finishes successfully (without crashing) on any valid PDB or mmCif input that contains protein structure. It is admittedly a moving target. P2Rank has been therefore evaluated by running it on the whole PDB and is regularly automatically run on new PDB entries. This stands in contrast with many available tools, some of which have a failure rate that can be as high as 20-80% (see supplementary materials to [KH18]).

- Interpretability. For each pocket and each residue, P2Rank produces a probability score, which is a number from the $[0, 1]$. Transformations from raw scores to probability scores are trained/fitted for each prediction model on a calibration dataset.

Features related to training new models and development of new methods

P2Rank can be also seen as a framework and a workbench for training new prediction models and developing new prediction methods. The following list summarizes the features that are relevant for advanced users/developers that want to do one of the following: train new models on specific datasets, develop methods for new prediction tasks, or develop new local prtoein descriptors and compare their contribution to predictive performance.

- Java API for predictions. P2Rank can be used as a library by the programs running on JVM.
- Training and evaluation of new models. P2Rank is able to train and evaluate new models on different dataset running single command.
- Configurability. P2Rank has more than 100 documented configurable parameters. Configuration can be stored in a config file and overridden in the command line.
- Different evaluation modes and metrics. P2Rank implements pocket-centric and also residue-centric evaluation and within them calculates various prediction performance metrics.
- Grid optimization with visualization. P2Rank implements an internal optimization loop for grid optimization based on a list of parameter values. If only one or two parameters are optimized at the same time P2Rank can produce bar charts or heatmaps for every calculated metric.
- Integration with external optimizers. P2Rank implements an internal optimization loop that can make use of third-party optimizers. Two optimizers that implement Bayesian optimization are currently integrated [SLA12, JG17].
- Easy development of new features/descriptors. P2Rank contains a clean internal API for the development of new features. New features

can be calculated either for protein atoms or residues (those are then projected onto solvent accessible surface points) or for solvent accessible surface points directly, depending on what comes most naturally.

- Ability to use externally calculated features/descriptors via CSV files which contain features calculated for every residue in the dataset.

3.2.2 Results

Results in Table 3.1 show that P2Rank clearly outperforms other evaluated tools in Top-n and Top-(n+2) categories on two datasets. P2Rank also achieves higher success rates than were possible to achieve just by re-scoring predictions of Fpocket using PRANK algorithm. Still, Fpocket+PRANK performed better than any of the other tools except for P2Rank. We have also evaluated the performance of a reduced version of P2Rank that uses only a single geometric feature (descriptor): protrusion. Surprisingly, even this simplified, purely geometric version of P2Rank slightly outperforms other tools in most cases (except for MetaPocket 2.0 in Top-(n+2) category).

TABLE 3.1: **P2Rank: Comparison of predictive performance on COACH420 and HOLO4K datasets.**

	COACH420		HOLO4K	
	Top-n	Top-(n+2)	Top-n	Top-(n+2)
Fpocket	56.4	68.9	52.4	63.1
Fpocket+PRANK[a]	63.6	76.5	62.0	71.0
SiteHound[†]	53.0	69.3	50.1	62.1
MetaPocket 2.0[†]	63.4	74.6	57.9	68.6
DeepSite[†]	56.4	63.4	45.6	48.2
P2Rank[protrusion][b]	64.2	73.0	59.3	67.7
P2Rank	**72.0**	**78.3**	**68.6**	**74.0**

The numbers represent identification success rate [%] measured by D_{CA} criterion (distance from pocket center to closest ligand atom) with 4 Å threshold considering only pockets ranked at the top of the list (n is the number of ligands in considered structure).
[†]These methods failed to produce predictions for some portion of input proteins. Here are displayed success rates calculated only based on subsets of proteins, on which they finished successfully. Detailed, pairwise comparison with P2Rank on the exact subsets can be found in the Supplementary Information of [KH18].
[a]predictions of Fpocket re-scored by PRANK algorithm
[b]reduced version of P2Rank that uses only single geometric feature: protrusion

3.3 PrankWeb: more than a web interface for P2Rank

We have developed PrankWeb, a web application for the prediction of ligand binding sites [JKS*19]. While PrankWeb uses P2Rank in the backend, it is not just a simple web interface for P2Rank. It additionally employs a custom-made conservation pipeline and improved prediction models trained using conservation as one of the features (i.e. descriptors). The new version [JSK*22] introduced the support for mmCIF format and prediction model specialized for AlphaFold structures [TAW*21].

Note: the pre-trained models that use conservation are included in the standalone command line distribution of P2Rank, but the conservation pipeline is not. To use these models in command line mode users can make use of PrankWeb's docker images.

3.3.1 Features

- PrankWeb is able to predict binding sites on experimental structures (PDB), AlphaFold models or any valid sructure uploaed by the user.
- Conservation pipeline. PrankWeb can calculate sequence conservation scores and employ this information in binding site prediction.
- Customizable web-based visualization of prediction results that integrates sequence and structural visualization. Visualization includes conservation score and AlphaFold score (pLDDT) if available.
- Precomputed predictions. We have computed the ligand binding site predictions for two components of the AlphaFold DB, the "model organism proteomes" and "Swiss-Prot", as well as for the whole PDB. For each database, AlphaFold DB and PDB, we computed the prediction with and without using conservation. Results precomputed for PDB are being automatically periodically updated by running predictions with the structures newly added to PDB. PrankWeb can serve the predictions on those structures to users instantaneously via its web interface. Moreover, precomputed predictions on individual databases are available for bulk download on PrankWeb's website.
- Documented REST API.

3.3.2 Results

Table 3.2 presents the evaluation of all new P2Rank models used for PrankWeb 3, as well as their comparison with the former models user by the original version or PrankWeb. It can be seen that the new Default models exceed the performance of the corresponding old models when evaluated on the representative HOLO4K dataset.

3.3.3 Implementation details

The original version of PrankWeb [JKS*19] was developed as a Java web application that was using P2Rank internally as a library via P2Rank's Java API. The advantage of this approach was that it avoided repeated JVM and model loading cost on each prediction run (which is measured in order of seconds).

TABLE 3.2: **PrankWeb: Results of four prediction models employed by PrankWeb 3** and comparison with two previously used models

	COACH420		HOLO4K	
	Top-n	Top-(n+2)	Top-n	Top-(n+2)
Default (old)	72.0	78.3	68.6	74.0
Default + conservation (old)	73.2	77.9	72.1	76.7
Default	71.6	76.8	72.7	78.0
Default + conservation	74.3	77.2	74.5	78.4
B-factor-free	71.2	77.5	72.1	77.2
B-factor-free + conservation	74.9	78.5	73.9	77.7

The numbers represent identification success rates (in %) measured using the DCA criterion utilizing a 4.0 Å threshold for the distance between the center of the predicted LBS and any ligand atom; only the n or (n+2), respectively, top-ranking predicted sites are considered in the evaluation, where n is the number of ligands in the respective 3D structure. Values for Default (old) and Default + conservation (old) represent results of old models used by the original version of PrankWeb. B-factor-free are used with AlphaFold predictions which utilize the B-factor field for confidence scores. Please note that old models were generated by the older version of P2Rank, which used older versions of BioJava and CDK. Using newer versions changed how certain PDB files are parsed, and an upgrade of the CDK library fixed a bug in the algorithm that generates SAS points. This, together with bug fixes in P2Rank itself, causes the scores for the Default (old) and Default models to differ.

With the new release, PrankWeb's architecture has been completely redesigned [JSK*22]. PrankWeb is now developed as a modern Python web application with modular architecture that strictly separates web-based user interface, data storage, and an execution component. Each component corresponds to a Docker image. Combined with docker-compose, it is easy to deploy and update PrankWeb instances, or using just the execution component run predictions on private data without exposing them to third-party servers. Each new prediction is now executed as a separate P2Rank process. This brings higher flexibility but also brings back JVM and model loading cost. This fact is now offset by faster startup times on newer JVMs and by the fact that predictions for many available structures are automatically precomputed by PrankWeb.

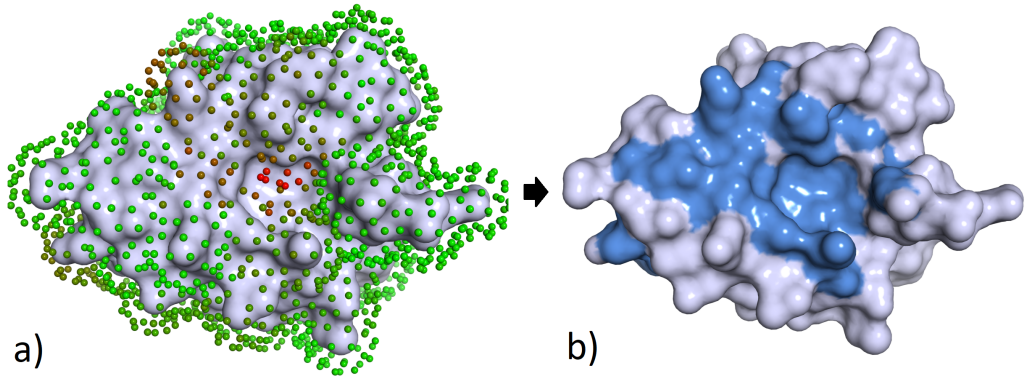


FIGURE 3.5: **Peptide-binding residue prediction based on points on the Solvent Accessible Surface**. **a)** Protein (3NFK/A) is covered in a layer of points lying on the solvent accessible surface. Each point represents its local chemical neighborhood and is described by a feature vector calculated from its surroundings. Points are colored according to the peptide-binding score ($\in$ [0,1]) predicted by a Random Forest classifier (*green=0/red=1*). **b)** Peptide-binding score of any given solvent exposed residue is based on the score of its adjacent points (radius of the cutoff and the form of aggregation function were subject to optimization). Residues with the score above a certain threshold are labeled as predicted positives (*blue*).

3.4 P2Rank-Pept: prediction of peptide binding sites

We have applied our approach to the task of peptide binding site prediction. Compared to P2Rank we had to develop and employ a variety of new features to achieve top performance. Among them were features related to protein geometry, secondary structure and sequence conservation. Figure 3.6 shows the outline of the algorithm i.e. the steps that P2Rank-Pept follows to predict peptide-binding residues using previously trained classification model. Prediction on a particular protein is further illustrated in Figure 3.5. P2Rank-Pept is a part of the P2Rank codebase, but up to this date it has not been released with a pre-trained model. Although we achieved predictive performance that was significantly higher than the competition, I was not convinced that the method is practically useful in its current state.

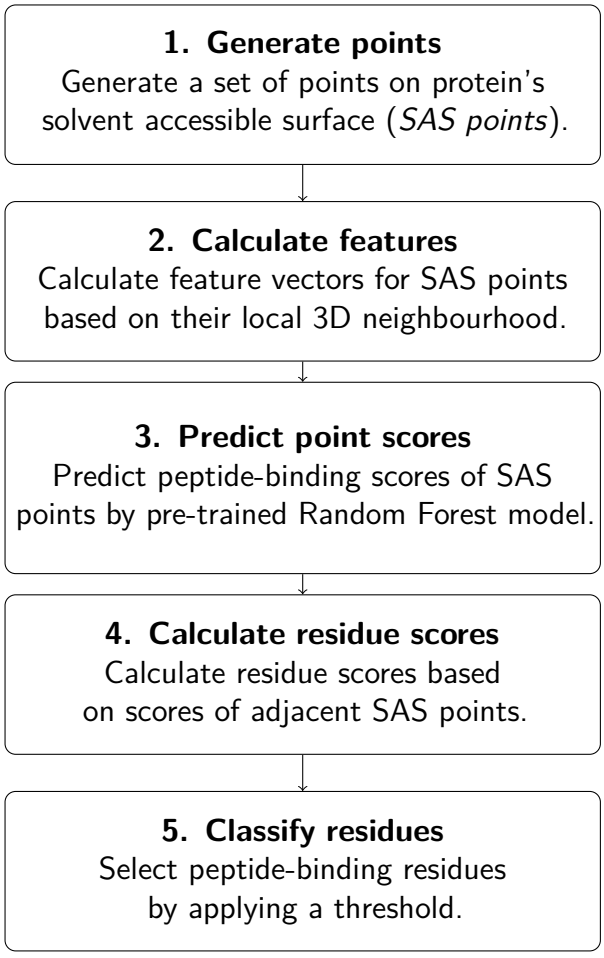


FIGURE 3.6: **P2Rank-Pept algorithm outline**

3.5 Integration with PDB-KB

We have integrated P2Rank/PrankWeb with EBI's Protein Data Bank in Europe – Knowledge Base (PDBe-KB), the new PDBe's major resource of integrated protein data [con19, con21]. PDB-KB now contains annotations based on P2Rank predictions precomputed for almost every protein in the PDB and it is being periodically updated with predictions on new proteins.

Chapter 4

Apo-Holo protein search

4.1 Introduction

Ligand-binding proteins exist in a bound (Holo) and an unbound (Apo) state. Structurally those states are almost always, to some extent, conformationally different due to the binding-induced conformational changes. For many proteins, both of these states can be found in the PDB, often in multiple entries.

This picture gets further complicated when we consider proteins that can bind multiple ligands on multiple binding sites (which is probably a majority of ligand-binding proteins). One particular protein with two binding sites can thus exist in a few different versions in the PDB: not binding any ligand, binding a ligand in one of the binding sites but not in the other, and binding ligands in both sites. The generally accepted definition is that a protein in the Apo state does not bind any ligands at all and Holo state covers the situations where it binds one or multiple ligands. However, when we talk about Apo-Holo protein pairs and their search, it is more useful to think about a pair of Apo-Holo structures with respect to: (a) a specific binding site, (b) a set of specific binding sites, (c) all known binding sites.

The Apo-Holo protein pairing is not readily available in the PDB and the consideration about multiple binding sites just illustrates one of the reasons. The process of Apo-Holo pairing is further complicated by sequence irregularities in the PDB, a consideration of which type of molecules should be considered as relevant ligands and a specific way how the binding site occupancy is determined (which is a process that necessarily involves some arbitrary thresholds). Apo-Holo protein pairing should thus not be seen as

a static link between PDB entries, but rather as a qualified search process, which results depend on a user query that can specify various arbitrary search options.

4.2 Motivation

Our motivation for developing Apo-Holo protein search tool was the need to create Apo-Holo datasets for better evaluation of binding site prediction methods. The general problem in the field of ligand binding site prediction (and arguably a shortcoming of my own work) is the fact that methods are typically being evaluated only on Holo datasets. Evaluating binding site prediction methods on Holo datasets means that the prediction method can "see" the protein structure as it is after the ligand-induced conformational changes. A prediction method can then use the information encoded in the conformational change in the Holo structure to predict a binding site that it would not be able to predict on the Apo structure. The consequence is that the reported results of success rates of binding site prediction methods can be overly optimistic and may not represent expected results when we apply them to Apo structures (which is almost always what we are looking for when running binding site prediction).

Many other bioinformatics tasks also require access to several conformations (preferably Apo and Holo) and can benefit from the existence of a flexible Apo-Holo search tool. These include observing the effects of ligand binding [BS08], exploring the specificity of a binding site [MSWN02], unveiling cryptic binding sites [CWR*16] and assessing the importance and consistency of water molecules [WDP*18].

4.3 Existing resources

Some resources to address the need of Apo-Holo protein pairing have been built previously. These can be divided into pre-calculated datasets or databases [LSG*10, CYF*12, DLOW07], and one search tool [MTNS11]. However, all the available resources seem to be either not actively updated or are not available at all at the time of writing.

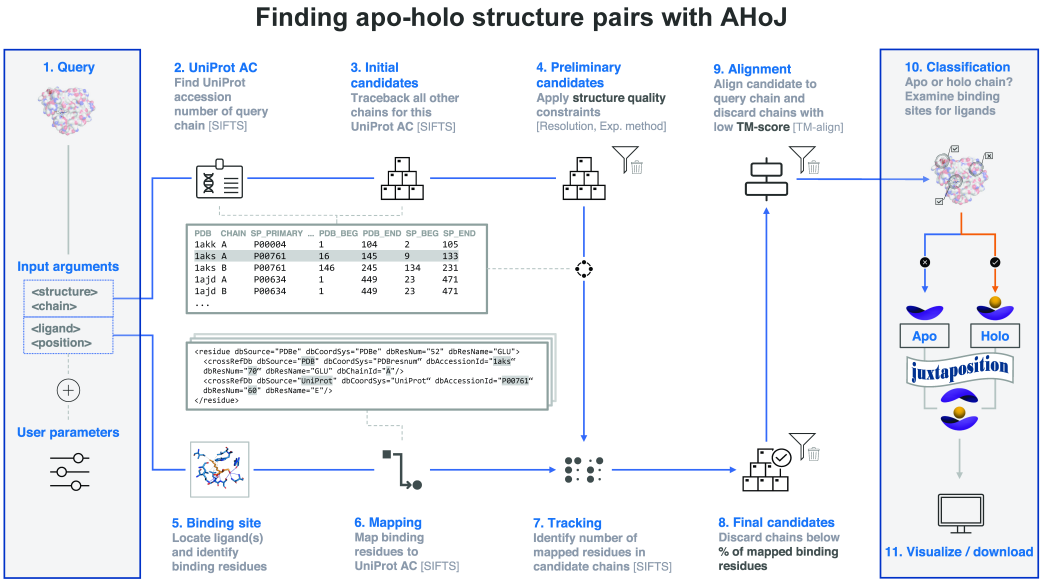


FIGURE 4.1: **Flowchart depicting the workflow in AHoJ**

4.4 Our solution

We have developed AHoJ, a command line tool and a web application that enables the user to conduct easy, fast and parameterizable searches for Apo-Holo structural pairs in the PDB against a query structure [FKHN22]. The user is allowed to specify one or more ligands or binding sites of interest as a part of a query, or can let the application detect the ligands instead. The query structure itself can be Holo or Apo and the result consists of two lists of found structures: those that are Apo with respect to specified binding sites and those that are Holo. All structures are furthermore aligned to the query structure and various metrics for each structure are calculated (including a sequence overlap with the query, RMSD and TM-score). The search process is illustrated in Figure 4.1.

Both the command line tool and the web application can process multiple queries in one run and thus allow to easily create custom Apo-Holo datasets or allow researchers to work in a batch mode without any further programming. The web application allows downloading the results of individual queries or the results of all the queries in a job together. The command line tool produces PyMol visualization and the web application additionally contains an integrated Mol* [SBD*21] visualization of the results (see Figure 4.2). Both applications are freely available and the command line tool is open-sourced.

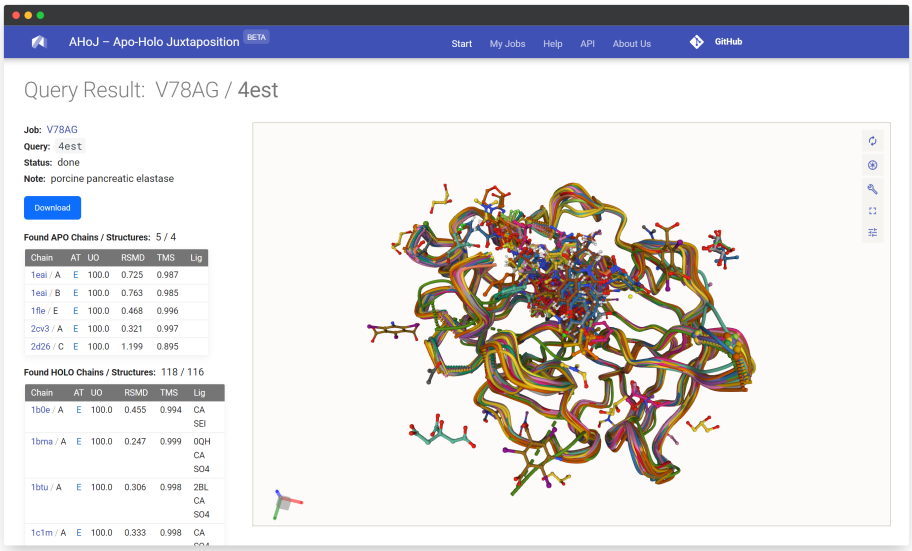


FIGURE 4.2: **AHoJ web application:** screenshot of a page that displays the result of a single search query.

Chapter 5

Conclusion

The main focus of my Ph.D. study was the application of machine learning to the problem of ligand binding site prediction from protein structure and related problems.

I have developed or contributed to the development of several novel methods which include the pocket re-scoring method PRANK, a stand-alone ligand binding site prediction method P2Rank (together with its extended web interface PrankWeb) and the peptide binding prediction method P2Rank-Pept.

The emphasis was always put also on producing pragmatically usable and user-friendly tools, not just on the publication of the methods. This seems to have been a largely successful approach which can be seen in the adoption data. To this date, a binary distribution of P2Rank has been downloaded more than 6500 times while PrankWeb is currently being used by more than 1300 unique users a month.

Furthermore, I have helped to develop AHoJ, a flexible tool for the search and alignment of Apo-Holo protein pairs in the PDB. The main motivation behind it was the need to create Apo-Holo datasets for better evaluation of binding site prediction methods. The existence of this tool will hopefully contribute to binding site prediction methods being again more commonly evaluated on Apo-Holo datasets.

Part II

Publications

Improving protein-ligand binding site prediction accuracy by classification of inner pocket points using local features

Reference

KRIVÁK R., HOKSZA D.: **Improving protein-ligand binding site prediction accuracy by classification of inner pocket points using local features**. *Journal of Cheminformatics 7*, 1 (Apr 2015), 12. doi:10.1186/s13321-015-0059-5

Author's highlights

We have developed PRANK, a machine learning based method that allows to re-score (re-rank) ligand binding sites predicted produced by other methods. Since it helps true binding sites to be ranked higher, it improves the applicability and usefulness of their predictions. PRANK was made available as a free command line tool with source code available upon request.

Note: in this paper we have used the term Connolly surface referring to the surface which would be more precisely described as solvent accessible surface.

Krivák and Hoksza *Journal of Cheminformatics* (2015) 7:12
DOI 10.1186/s13321-015-0059-5

Journal of Cheminformatics

RESEARCH ARTICLE Open Access

Improving protein-ligand binding site prediction accuracy by classification of inner pocket points using local features

Radoslav Krivák* and David Hoksza*

Abstract

Background: Protein-ligand binding site prediction from a 3D protein structure plays a pivotal role in rational drug design and can be helpful in drug side-effects prediction or elucidation of protein function. Embedded within the binding site detection problem is the problem of pocket ranking – how to score and sort candidate pockets so that the best scored predictions correspond to true ligand binding sites. Although there exist multiple pocket detection algorithms, they mostly employ a fairly simple ranking function leading to sub-optimal prediction results.

Results: We have developed a new pocket scoring approach (named PRANK) that prioritizes putative pockets according to their probability to bind a ligand. The method first carefully selects pocket points and labels them by physico-chemical characteristics of their local neighborhood. Random Forests classifier is subsequently applied to assign a ligandability score to each of the selected pocket point. The ligandability scores are finally merged into the resulting pocket score to be used for prioritization of the putative pockets. With the used of multiple datasets the experimental results demonstrate that the application of our method as a post-processing step greatly increases the quality of the prediction of Fpocket and ConCavity, two state of the art protein-ligand binding site prediction algorithms.

Conclusions: The positive experimental results show that our method can be used to improve the success rate, validity and applicability of existing protein-ligand binding site prediction tools. The method was implemented as a stand-alone program that currently contains support for Fpocket and Concavity out of the box, but is easily extendible to support other tools. PRANK is made freely available at http://siret.ms.mff.cuni.cz/prank.

Keywords: Ligand binding site; Protein pocket; Binding site prediction; Pocket score; Molecular recognition; Machine learning; Random forests

Background

Accurate prediction of ligand-binding sites, often simply called pockets, from a 3D protein structure plays a pivotal role in rational drug design [1,2] and can be helpful in drug side-effects prediction [3] and elucidation of protein function [4]. Ligand-binding sites are usually found in deep protein surface cavities, but it should be emphasized that not all binding sites are found in deep cavities. Although empirical studies show that the actual ligand-binding sites tend to coincide with the largest and deepest pocket on the protein's surface [5,6], there exist cases where ligands are found binding to rather exposed shallow clefts [7,8].

Plethora of pocket detection methods, that employ variety of different strategies, are currently available. These include purely geometric methods, energetic methods and methods that make use of evolutionary conservation (see below). All these methods take a protein structure as an input and produce an ordered list of putative pockets, which represent the locations on the protein surface where ligands are expected to bind. Not all reported pockets usually correspond to true binding sites, but it is expected that entries at the top of the ordered list correspond to regions with the highest probability of being a true binding site. Although it is not unusual for one protein to have more than one ligand-binding site, the

*Correspondence: krivak@ksi.mff.cuni.cz; hoksza@ksi.mff.cuni.cz
Department of Software Engineering, Charles University in Prague, Prague, Czech Republic

number of putative pockets predicted by pocket detection methods tends to be much higher than the number of actual known positives. The accuracy of a pocket prediction method is then evaluated by its ability to yield the true (experimentally confirmed) binding sites among the top-n putative pockets on its output (where n is usually taken to be 1, 3 or 5).

As the list of predicted pockets contains false positives, ordering of the pockets, i.e. pocket ranking, plays an important role and substantially contributes to the overall accuracy of the prediction method. More importantly, correct pocket ranking is of practical utility: it helps to prioritize subsequent efforts concerned with the predicted pockets, such as molecular docking or virtual screening.

While many ligand-binding site detection approaches employ complex and inventive algorithms to locate the pockets, the final ranking is often done by a simple method such as ordering by size or scoring pockets by a linear combination of few pocket descriptors. In the present study we are introducing a novel pocket ranking algorithm based on machine learning that can be used as a post-processing step after the application of a pocket prediction method and thus improve its accuracy. We demonstrate that applying this re-ordering step substantially improves identification success rates of two pocket prediction methods, Fpocket [9] and ConCavity [10], on several previously introduced datasets.

Pocket detection approaches

In the last few years, we have been able to observe increased interest in the field of pocket detection indicated by a number of recently published reviews [2,11,12], as well as by the influx of new detection methods. The pocket detection algorithms can be categorized based on the main strategy they adopt in the process of binding site identification. Those strategies and their representative methods shall be briefly reviewed in the following paragraphs.

Geometry based methods

The geometrical methods focus mainly on the algorithmic side of the problem of finding concave pockets and clefts on the surface of a 3D structure. Some methods are purely geometrical (LIGSITE [13], LIGSITE^{cs} [14], PocketPicker [5]), while others make use of additional physico-chemical information like polarity or charge (MOE SiteFinder [15], Fpocket [9]).

Energy based methods

The energy based methods build on the approximation of binding potentials or binding energies [16]. They place various probes on the grid points around the protein's surface and calculate interaction energies of those points with the use of underlying force field software. That results in higher computational demands of these methods [17]. Representative examples of the energy based methods include Q-SiteFinder [18], SiteHound [8], dPredGB [19] or the method by Morita et al. [20].

Evolutionary and threading based methods

The sequence-based evolutionary conservation approaches are based on the presumption that functionally important residues are preferentially conserved during the evolution because natural selection acts on function [21]. In LIGSITE^{csc} [14], a sequence conservation measure of neighboring residues was used to re-rank top-3 putative pockets calculated by LIGSITE^{cs}, which lead to an improved success rate (considering top-1 pocket). In ConCavity [10], unlike in LIGSITE^{csc}, the sequence conservation information is used not only to re-rank pockets, but it is also integrated directly into the pocket detection procedure. An example of an evolutionary based method which takes into account the structural information is FINDSITE [22,23]. It is based on the observation that even distantly homologous proteins usually have similar folds and bind ligands at similar locations. Thus at first ligand-bound structural templates are selected from the database of already known protein-ligand complexes by a threading (fold recognition) algorithm. The used threading algorithm is not based only on sequence similarity, but it also combines various scoring functions designed to match structurally related target/template pairs [24]. Found homologous structures are subsequently aligned with the target protein by a global structural alignment algorithm. Positions of ligands on superimposed template structures are then clustered into consensus binding sites.

Consensus methods

The consensus methods are essentially meta approaches combining results of other methods. The prominent example is MetaPocket [25]. The recently introduced updated version, MetaPocket 2.0 [26], aggregates predicted sites of 8 different algorithms (among them the aforementioned LIGSITE^{cs}, Q-SiteFinder, Fpocket and ConCavity) by taking top 3 sites from each method. The authors demonstrated that MetaPocket performed better than any of the individual methods alone.

Ranking algorithms

Given that every pocket identification algorithm is basically a heuristic it needs to incorporate a scoring function providing a measure of confidence in given prediction. A simple strategy for scoring putative pockets, one that is probably most commonly used, is ordering pockets by a single descriptor — like size (volume), pocket depth, surface area or the overall hydrophobicity. Another strategy for scoring pockets is to combine several pocket

descriptors. Fpocket, for example, uses a linear combination of 5 such descriptors which parameters were optimized on a training dataset. The same approach was also successfully applied in recent druggability prediction methods [27,28]. In ConCavity, the ranking procedure considers overall pocket evolutionary conservation score that is projected onto pocket grid probes. One study that focused solely on ranking of pockets previously found by other pocket detection algorithms introduced an approach based on amino acid composition and relative ligand binding propensities of different amino acids termed PLB index [29] (we compare our proposed method with PLB index in results section).

It has been suggested that pocket identification and pocket ranking are independent tasks and therefore should be evaluated separately [30].

It seems that pocket detection methods that have achieved the highest success rates in the aforementioned benchmark are those with more sophisticated ranking algorithms. It has also been suggested that the total coverage (i.e. identification success rate considering all predicted pockets without regard to the ordering) of many algorithms is actually close to 100% [30]. While our experiments do not support such a strong claim they, nevertheless, show that there is indeed a big difference between success rate with regards to top 1, top 3 binding sites and the total coverage. Therefore, there is room for improvement by introducing a more precise and sophisticated ranking algorithm that would rank the identified true pockets higher than the false ones.

Performance of existing methods

Considering that the goal of our method is to increase the performance of the existing state of the art methods we have to raise a question regarding their actual performance. It has been acknowledged that the field of ligand-binding site prediction lacks standardized and widely accepted benchmarking datasets and guidelines [30,31]. In the studies introducing the individual methods, their performance was usually compared to a couple of existing methods with (somewhat expectedly) favorable results, reporting success rates around 90% regarding the top 3 and 70% considering the top 1 predicted sites. The latest review [31] represents the first independent attempt to systematically assess the performance of the pocket detection methods, although only a limited set of 8 representative methods has been considered. It has challenged the previously reported high success rates of the pocket prediction programs. With the exception of FINDSITE, identification success rates of all methods on the new dataset were considerably lower than previously reported (closer to 50% rather than the often reported 70% for top 1 prediction). FINDSITE achieved clearly the best results, but only with the help of a comprehensive threading library that contained proteins highly similar to those from the benchmarking dataset. It was demonstrated that when those were removed from the library, success rates of FINDSITE dropped to the level of other methods [31].

Methods

We are introducing here a new pocket ranking method PRANK that can be used to increase the performance of existing pocket prediction methods. Thus the input of the method is a list of predicted putative pockets and its goal is to prioritize the list in such a way that the true pockets appear at the top of that list. PRANK is a machine learning method which is based on predicting ligandability of specific pocket points near the pocket surface. These points represent possible locations of contact atoms of a putative ligand. By aggregating predictions of those points PRANK outputs a score to be used for the re-ranking of the putative pockets. Thus, unlike previous studies that applied machine learning in the context of protein binding site prediction [32-37], we focused on the classification of inner pocket points rather than the classification of exposed amino acid residues or whole pockets. The following list outlines the PRANK method (see also Figure 1):

1. Sampling inner pocket points from Connolly surface of the protein.
2. Calculating feature descriptors of the sampled points based on their local chemical neighborhood.
    a) Computing property vectors of chosen protein's solvent exposed atoms.
    b) Projecting distance weighted properties of the adjacent protein atoms onto the sampled inner pocket points.
    c) Computing additional inner pocket points specific features.
3. Predicting ligandability of the sampled inner pocket points by random forests classifier using their feature vectors.
4. Aggregating predictions into the final pocket score.

Individual steps are described in greater detail in following sections. For the visualization of classified pocket points see Figure 2.

Pocket representation

To represent a pocket, PRANK first computes a set of its *inner points* by selecting evenly spaced points lying on the Connolly surface [38] that lie in the distance of at most 4 Å from the closest heavy pocket atom. This method of choosing points to represent a pocket is similar to the one used by Morita et al. [20], although we deliberately

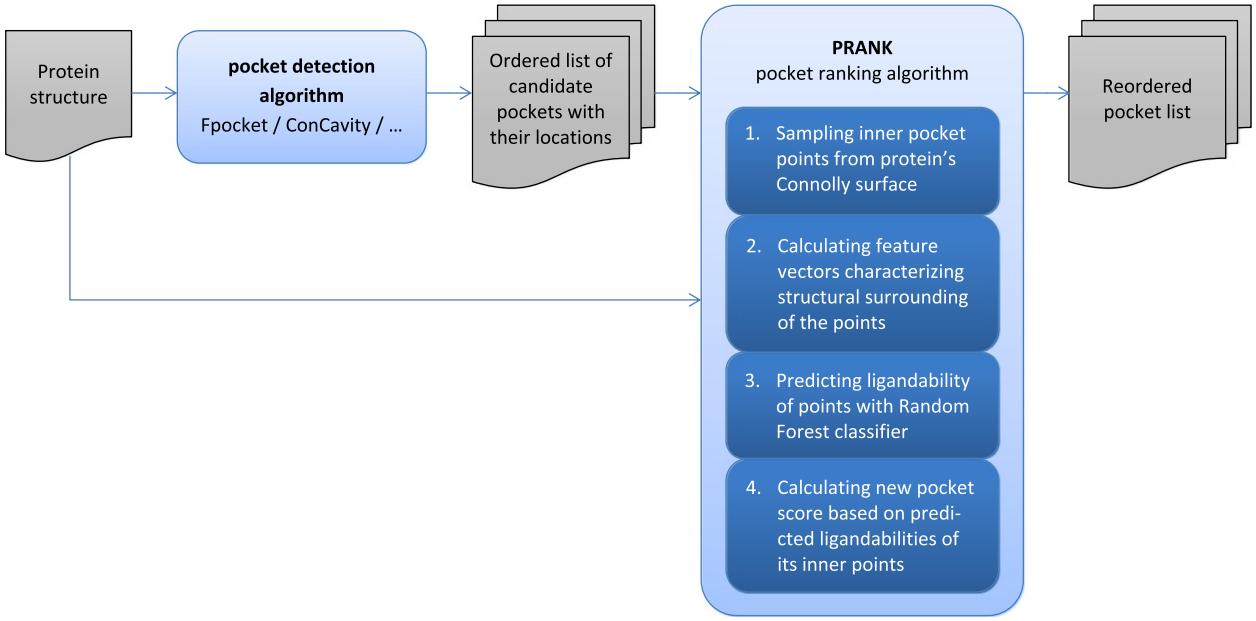


Figure 1 Flowchart of the PRANK pocket ranking approach.

use only one Connolly surface layer with optimized probe radius of 1.6 Å. Thus PRANK utilizes only points in a relatively short belt around the pocket surface as the bonding between ligand and protein takes place in this area.

Next, PRANK assigns a feature vector to each of the inner points. The feature vector is built in two steps: first, it calculates feature vectors for specific pocket atoms (*AFVs*) which are then aggregated into feature vectors of the inner points (*IFVs*).

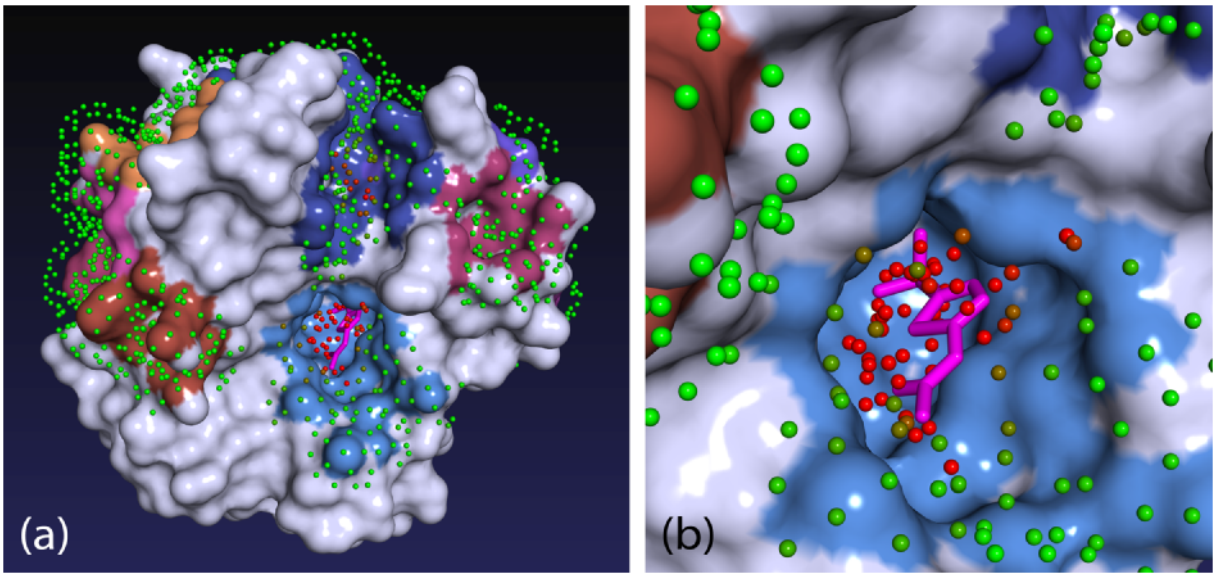


Figure 2 Visualization of inner pocket points. **(a)** Displayed is protein 1AZM from DT198 dataset bound to one ligand (magenta). Fpocket predicted 13 pockets that are depicted as colored areas on the protein surface. To rank these pockets, the protein was first covered with evenly spaced Connolly surface points (probe radius 1.6 Å) and only the points adjacent to one of the pockets were retained. Color of the points reflects their ligandability (green = 0…red = 0.7) predicted by Random Forest classifier. PRANK algorithm rescores pockets according to the cumulative ligandability of their corresponding points. Note that there are two clusters of ligandable points in the picture, one located in the upper dark-blue pocket and the other in the light-blue pocket in the middle. The light-blue pocket, which is in fact the true binding site, contains more ligandable points and therefore will be ranked higher. **(b)** Detailed view of the binding site with ligand and inner pocket points.

The AFVs are computed only for pocket atoms located in the atomic neighborhood of any inner point. The atomic neighborhood of point P is defined as:

$$\mathrm{A}(P) = \{\text{heavy solvent exposed protein atoms within 8 Å radius around } P\} \quad (1)$$

The features forming the AFVs include two types of features: residue level features and atomic level features. The residue level features are characteristics of residues inherited by their constituent atoms. Such features include, e.g., physico-chemical properties of standard amino acids or hydropathy index of amino acids [39]. The atomic levels features are specific to individual atoms meaning that different atoms within one amino acid can have different values of those features. Examples of such features are physico-chemical properties of individual amino acid atoms adopted from VolSite druggability prediction study [40] or statistical ligand-binding propensities of amino acid atoms [41] (see Additional file 1: Listings for the complete feature list).

To calculate the feature vector of an inner pocket point (IFV), the AFVs from its atomic neighborhood are aggregated using a simple aggregation function and concatenated with a vector of features computed specifically for that point from its local neighborhood. These inner point features include the number of H-bond donors and acceptors, B-factor of structure atoms or protrusion index [42] The following aggregation function is used to project the pocket atoms feature vectors onto the inner points:

$$\mathrm{IFV}(P) = \sum_{\mathrm{A}_i \in \mathrm{A}(P)} \mathrm{AFV}(\mathrm{A}_i) \cdot w(\mathrm{dist}(P, \mathrm{A}_i)) \quad || \quad \mathrm{FV}(P), \quad (2)$$

where FV is the vector of the inner points specific features and w is a distance weight function :

$$w(d) = 1 - d/8. \quad (3)$$

We evaluated several types of weight functions with different parameters (among them quadratic, Gaussian and sigmoid), but in the end we selected the present simple linear function which had produced the best results in the cross-validation experiments.

It also needs to be emphasized that all of the features included in the vectors are local, which means that they are calculated only based on the immediate spatial neighborhood of the points. No regard is taken to the shape and properties of the whole pocket or protein. Although the 8 Å cutoff radius by which we define chemical neighborhood can encompass considerable part of the whole pocket, immediate surrounding atoms have more influence thanks to the fact that we weight their contribution by distance (see Equation 3). Inner pocket points from different parts of the pocket can therefore have very different feature vectors. We propose that this locality has some positive impact on the generalization ability of the model.

One possible negative implication of considering only local features could be that local features are not sufficient to account for ligand binding quality of certain regions of protein surface since some ligand positions could be fixed by few relatively distant non-covalent bonds. However, our results show that in spite of that concern our local approach leads to practical improvements.

Classification-based ligandability prediction

Similarly to other studies that were trying to predict whether exposed residues of a protein are ligand binding or not, we used a machine learning approach to predict the ligandability of inner pocket points. The ligandability prediction is a binary classification problem for supervised learning. Training datasets of inner pocket points were generated as follows. For a given protein dataset with candidate pockets (e.g. CHEN11 dataset with Fpocket predictions) we merged all sampled inner pocket points and labeled as positive those located within 2.5 Å distance to any ligand atom. The resulting point datasets were highly imbalanced in terms of positives and negatives since most of the candidate pockets and their points were not true ligand binding sites (e.g. CHEN11-Fpocket dataset contained 451,104 negative and 30,166 positive points resulting in 15:1 ratio). Compensation techniques such as oversampling, undersampling and cost-sensitive learning are sometimes applied in such scenarios, but in our experiments they only led to notable degradation of the generalization ability of a trained classifier (i.e. performance on other datasets). The size of the point dataset depends on the density of the points sampled from the Connolly surface of a protein. The numerical algorithm that was employed to calculate the Connolly surface [43] is parametrized by an integer tessellation level. Our algorithm uses level 2 by default as higher levels increase the number of points geometrically but do not improve the results.

After preliminary experiments with several machine learning methods we decided to adopt Random Forests [44] as our predictive modelling tool of choice. Random Forests is an ensemble of trees created by using bootstrap samples of training data and random feature selection in tree induction [45]. In comparison with other machine learning approaches, Random Forests are characterized by an outstanding speed (both in learning and execution phase) and generalization ability [44]. Additionally, Random Forests is robust to the presence of a large number of irrelevant variables; it does not require their prior scaling [37] and can cope with complex interaction structures as well as highly correlated variables [46]. The

ability of Random Forests to handle correlated variable comes in handy in our case because for example features such as hydrophobicity and hydrophilicity are obviously related.

To report the performance of a classifier, three statistics are commonly reported: precision, recall (also called sensitivity) and Matthews Correlation Coefficient (MCC). MCC is often used to describe the performance of a binary classifier by a single number in scenarios with imbalanced datasets. In such scenarios the predictive accuracy is not an effective assessment index. MCC values range from +1 (perfect prediction), over 0 (random prediction) to −1 (inverse prediction). The performance statistics are calculated as shown below. TP, TN, FP and FP stand for true positive, true negative, false positive, and false negative predictions.

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{6}$$

Scoring function

As soon as the classifier is trained it can be used within the PRANK's scoring function to rescore the putative pockets. To do so we utilize the histogram of class probabilities returned by the random forests classifier for every sampled inner pocket point. Since our problem is binary (a point can either be seen as a pocket point or not) the histogram is an ordered pair $[P_0, P_1]$. The score is then the sum of predicted squared positive class probabilities of all inner pocket points:

$$\text{PScore} = \sum_i (P_1(V_i))^2 \tag{7}$$

Squaring the probabilities puts more emphasis on the points with probability closer to 1. Originally, we experimented with a mean probability based pocket score where *PScore* was divided by the number of inner points. However, we found that the employed cumulative score steadily gives better results. We attribute it to the fact that the size of a correctly predicted pocket can slightly deviate from the true pocket but it still should be recognized as a true pocket. In an oversized predicted pocket that contains in it a true binding site, dividing by the number of points would lead to the decrease of its score.

The higher the *PScore* of a putative pocket, the higher the probability of it being a true pocket. Thus the very last step involves reordering the putative pockets in the decreasing order of their *PScores*.

Optimization of parameters

Apart from the hyperparameters of the classifier, our method is parameterized by a number of additional parameters that influence various steps of the algorithm, from sampling inner pocket points to calculating and aggregating the features. Since many parameters have an impact on experiment running times and optimizing all parameters at once would be too costly, we optimized default values of those parameters by linear search, and in some cases by grid search (optimizing two parameters at once). Parameters were optimized with regard to the performance on CHEN11 dataset (see the datasets section) considering averaged results of repeated independent runs of 5-fold cross-validation. The optimized parameters included, for example, the probe radius of Connolly's surface (1.6 Å), ligand distance threshold to denote positive and negative points (2.5 Å) and the choice of the weight function in the inner points feature vector building step.

Implementation and efficiency

Our software is implemented in languages Groovy and Java with the help of machine learning framework Weka [47] and bioinformatical libraries BioJava [48] and The Chemistry Development Kit (CDK) [49]. Points on the Connolly's surface are calculated by a fast numerical algorithm [43] implemented in CDK.

Rescoring is implemented in a parallel fashion with configurable number of working threads and therefore can make use of all of the system's processor cores. In our experience, running times of our rescoring step were generally lower than the running times of the pocket prediction methods themselves, even on a single thread.

Experimental

Datasets

To show that application of PRANK is beneficial irrespective of the test set, we investigated its ability to increase the prediction accuracy on several diverse datasets. The following list briefly introduces those datasets.

- CHEN11 – This dataset includes 251 proteins and 476 ligands which were used to benchmark pocket detection methods in a recent comparative review [31]. It was designed with the intention to non-redundantly cover all SCOP families of ligand binding proteins from PDB. It can be considered as "hard" dataset as most methods performed rather poorly on this dataset.
- ASTEX – Astex Diverse set [50] is a collection of 85 proteins that was introduced as a benchmarking dataset for molecular docking methods.
- UB48 – UB48 [14] contains a set of 48 proteins in a bound and unbound state. It has been the most

widely used dataset for comparing pocket detection methods. Since it contains mainly small globular proteins with one stereotypical large binding site it can be seen as a rather "easy" dataset.
- DT198 – a dataset of 198 drug-target complexes [26].
- MP210 – a benchmarking dataset of 210 proteins in bound state introduced in the MetaPocket study [25].

For each dataset we generated predictions using two algorithms, Fpocket and ConCavity, which we use as model examples in our re-ranking experiments. Fpocket was used with its default parameters in version 1.0[a]. ConCavity can be run in two modes depending on whether it makes use of sequence conservation information or not. To execute it in the conservation mode it needs to be provided with pre-calculated residue scores. For this we were relying on the pre-computed sequence conservation files available online at the ConCavity website [51]. However, for several proteins from our datasets the conservation files were not available. For these proteins we executed ConCavity with the conservation option turned off. List of affected proteins is provided in Additional file 1: Listings. Except for the conservation switch, ConCavity was run with default parameters.

Table 1 shows statistics of individual datasets together with the average number of pockets predicted per protein by Fpocket and ConCavity. Evidently, Fpocket produces more putative pockets than ConCavity. This number alone, however, is not conclusive since incorrectly identified pockets can be included. However, the table also shows the total coverage (percentage of identified pockets) which is clearly in favor of Fpocket. Higher number of putative pockets and higher coverage makes Fpocket a better target of a re-ranking algorithm.

Evaluation methodology

To evaluate binding site predictions we followed the evaluation methodology introduced in [31]. Unlike previous studies, it uses the ligand-centric not protein-centric approach to calculate success rates. While the ligand-centric approach to evaluation, for a method to be 100% successful on a protein, we want it to identify every pocket on that protein for every relevant ligand in the dataset, the protein-centric approach only requires every protein to have at least one identified binding site. A pocket is considered successfully identified if at least one pocket (of all predicted pockets or from the top of the list) passes a chosen detection criterion (see below).

Furthermore, instead of reporting success rates for Top-1 or Top-3 predicted pockets, we report results for Top-n and Top-(n+2) cutoffs, where n is the number of known ligand-binding sites of the protein that includes evaluated binding site. This adjustment was made to accommodate for proteins with more than one known binding site (CHEN11 dataset, also introduced in [31] contains on average more than 2 binding sites per protein, see Table 1). Specifically, if a protein contains two binding sites, then Top-1 reporting is clearly insufficient in distinguishing methods which returned a correctly identified pocket in the first position of their result set but differ in the second position. For this reason, using the Top-n and Top-(n+2) cutoffs is more suitable for the ligand-centric evaluation approach.

Pocket detection criteria

Since a predicted pocket does not need to match the real pocket exactly, we need a criterion defining when the prediction is correct. When evaluating PRANK we adopted the following two criteria.

- D_{CA} is defined as the minimal distance between the center of the predicted pocket and any atom of the ligand. A binding site is then considered correctly predicted if D_{CA} is not greater than an arbitrary threshold, which is usually 4 A. It is the most commonly used detection criterion that has been utilized in virtually all previous studies.

Table 1 Datasets statistics

Dataset	Proteins	Ligands	#L	$\#P_{FP}$	$\#P_{CC}$	Cov_{FP}[%]	Cov_{CC}[%]	LS	PS_{FP}	PS_{CC}
CHEN11	251	476	1.90	12.41	1.75	71.0	52.3	26.9	38.9	51.0
ASTEX	85	143	1.68	21.58	2.25	81.1	65.7	23.2	41.9	56.9
DT198	198	192	0.97	18.57	2.19	80.2	65.6	20.8	41.2	53.7
MP210	210	288	1.37	14.50	1.99	78.8	68.2	22.8	40.0	50.9
B48	48	54	1.13	12.06	1.96	92.6	81.5	21.9	37.8	44.2
U48	48	54	1.13	11.40	1.79	88.9	77.8	21.9	38.0	46.8

Abbreviations: FP Fpocket, CC ConCavity.
#L: average number of ligands for one protein.
#P: average number of predicted pockets for one protein.
Cov: total coverage – success rate considering all predicted pockets (measured by D_{CA} criterion with 4 Å threshold).
LS: average number of heavy atoms in a relevant ligands (ligand size).
PS: average number of protein surface atoms that belong to a predicted pocket (pocket size).

- D_{CC} is defined as the distance between the center of the predicted pocket and the center of the ligand. It was introduced in the Findsite study [22] to compensate for the size of the ligand.

In several studies, criteria based on volume overlap of pocket and ligand were used in addition to the standard criteria. However, since our method does not change the shape of the predicted pockets, inclusion of a volume overlap based criterion would not influence the resulting pocket ordering. Therefore, we did not include any such a criterion into our evaluation.

Results and discussion

Results

To demonstrate the PRANK's ability to increase the quality of prediction of a pocket prediction method (Fpocket and ConCavity) we performed two types of tests. First, we used the CHEN11 dataset for cross-validation experiments and second, we trained our prediction model on the whole CHEN11 dataset and used this model to evaluate our method on the rest of the datasets. The same model is also distributed as the default model in our software package. The reason to train the final model on the CHEN11 dataset is its structural diversity and the fact that it was compiled to include all known ligands for given proteins. The cross-validation results show the viability of our modelling approach on a difficult dataset (CHEN11), and the evaluation of the final model on the remaining datasets attests the generalization ability and applicability of our software out of the box.

The results, including the performance statistics of the classifier, are summarized in Table 2. The *Top-n* column displays the success rate of the particular method (Fpocket or ConCavity) when PRANK is not involved, while the *Rescored* column shows the success rate when PRANK was utilized as a post-processing step. It should be emphasized that since PRANK's goal is not to discover any new pockets, the maximum achievable success rate is upper bounded by the total coverage of the native prediction method as displayed in the *All* column. In other words, the difference between *Top-n* and *All* represents the possible improvement margin, i.e., the highest nominal improvement in success rate for the *Top-n* cutoff that can be achieved by optimal reordering of the candidate pockets. Thus, the *Improvement* column shows the nominal improvement of PRANK while the *%possible* column shows the percentage of the possible improvement margin. Finally, the last three columns show the statistics related to the PRANK's underlying Random Forests classifier itself.

The results clearly show that the application of PRANK, using the D_{CA} pocket detection criterion with 4 A threshold, considerably outperformed the native ranking methods of Fpocket and ConCavity on all the evaluation datasets. In most of the cases more than 50% of the possible improvement (the *Rescored* column) was achieved. When translated into the absolute numbers, it means that in some cases using PRANK can boost the overall prediction performance of a method by up to 20% (the *Improvement* column) with respect to the absolute achievable maximum.

We also conducted experiments showing how PRANK behaves when the distance threshold in the D_{CA} pocket detection criterion varies. The results carried out on the CHEN11 dataset demonstrate that the improvement of PRANK is basically independent on the utilized threshold (see Figure 3). Finally, to explore the PRANK qualities in greater detail, Figure 4 displays the success rates tracking different distance thresholds and different Top-N cutoffs on the CHEN11-Fpocket dataset.

Furthermore, we compared performance of PRANK against two simpler pocket ranking methods: PLB index, which is based on amino acid composition [29], and simple ordering of pockets by volume that serves as a baseline. PLB index was originally developed to rescore pockets of MOE SiteFinder [15]. We have reimplemented the method and used it to rescore pockets found by Fpocket and ConCavity. The results of the comparison are summarized in Table 3. Using PRANK to rescore Fpocket outperforms both ranking methods on all datasets while for ConCavity predictions PRANK is outperformed only in individual cases by volume ranking on Astex dataset and PLB index on U(B)48 datasets. The improvement by application of PRANK is more significant when rescoring outputs of Fpocket than ConCavity. This can be attributed to the fact that ConCavity predicts, on average, less putative pockets than Fpocket (see Table 1). Having lower margin then allows even a simple method to yield relatively good performance since the possibility of error is lower as well. We can conclude that PRANK is better in prioritizing long lists of pockets that contain many false positives and therefore gives more stable results. All results are summarized in Additional file 2: Tables.

Although we believe that the overall performance or the PRANK method is good enough, the performance of the underlying prediction model itself can be considered less satisfactory (see the last three columns in Table 2). In few cases the classifier achieved precision of less than 0.5, which means that of all the predicted positives more than a half was predicted incorrectly. Despite of that, reordering pockets according to the new scores led to improvements. This is possible because even predictions deemed as false positives (not within a 2.5 A distance to the ligand) could actually be points from true pockets and contribute to their score. Secondly, because of the particular way we calculate the final pocket score (see

Table 2 Rescoring Fpocket and ConCavity predictions with PRANK: cross-validation results on CHEN11 dataset and the results of the final prediction model (trained on CHEN11-Fpocket) for all datasets

Dataset	Top-n [%]	Rescored [%]	All [%]	Δ	%possible*	P	R	MCC
Fpocket predictions								
CHEN11 (CV)**	47.9	58.8	71	+10.6	47.1	0.60	0.32	0.41
CHEN11***	47.9	67.9	71	+20	86.4	0.87	1.0	0.98
ASTEX	58	63.6	81.1	+5.6	24.2	0.56	0.41	0.46
DT198	37.5	56.2	80.2	+18.8	43.9	0.31	0.38	0.33
MP210	56.6	67.7	78.8	+11.1	50	0.58	0.42	0.47
B48	74.1	81.5	92.6	+7.4	40	0.58	0.45	0.49
U48	53.7	77.8	88.9	+24.1	68.4	0.55	0.36	0.42
ConCavity predictions								
CHEN11 (CV)**	47.9	50.7	52.3	+2.8	63.3	0.44	0.76	0.40
CHEN11***	47.9	52.3	52.3	+4.4	100	0.80	0.82	0.75
ASTEX	55.2	62.9	65.7	+7.7	73.3	0.60	0.55	0.46
DT198	45.8	61.5	65.6	+15.6	78.9	0.33	0.55	0.34
MP210	57.4	66.1	68.2	+8.7	80.6	0.63	0.53	0.49
B48	66.7	77.8	81.5	+11.1	75	0.61	0.53	0.47
U48	64.8	74.1	77.8	+9.3	71.4	0.58	0.46	0.43

Abbreviations: P precision, R recall, MCC Matthews correlation coefficient.
*percentage of improvement that was theoretically possible to obtain by reordering pockets [Δ / (All – Top-n)].
**cross-validation results.
***results where the test set was de facto the same as the training set for the Random Forest classifier (included here only for completeness).

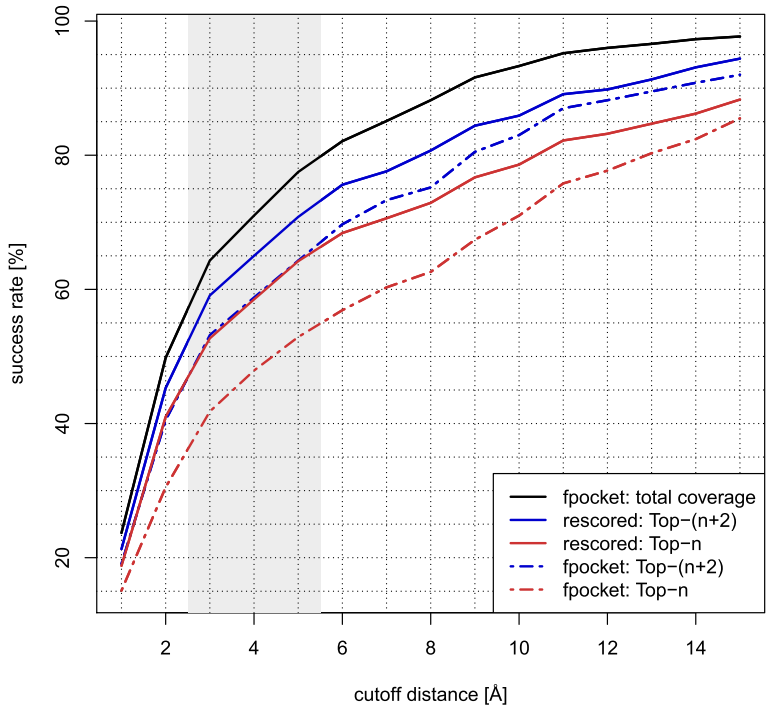


Figure 3 Rescoring Fpocket predictions on CHEN11 dataset. Success rates of Fpocket compared with results rescored by PRANK on CHEN11 dataset considering Top-n, Top-(n+2) and all pockets (total coverage). Identification success is measured by D_{CA} criterion for the range of integer cutoff distances. Displayed results for rescored pockets are averaged from ten independent 5-fold cross-validation runs.

	Fpocket				Rescored				Improvement			
	Top-n	Top-(n+2)	Top-(n+4)	All	Top-n	Top-(n+2)	Top-(n+4)	All	Top-n	Top-(n+2)	Top-(n+4)	All
D_{CA}(1Å)	15.1	19.1	21	23.7	18.8	21.3	22.6	23.7	3.7	2.1	1.6	0
D_{CA}(2Å)	30.5	40.5	44.5	49.8	41	45.3	47.5	49.8	10.5	4.7	3	0
D_{CA}(3Å)	41.8	53.2	57.8	64.3	52.7	59.1	61.3	64.3	10.9	6	3.6	0
D_{CA}(4Å)	47.9	58.8	63.7	71	58.5	65	67.6	71	10.6	6.2	3.9	0
D_{CA}(5Å)	52.9	64.3	69.5	77.5	64.2	70.8	73.7	77.5	11.3	6.5	4.2	0
D_{CA}(6Å)	56.9	69.7	74.6	82.1	68.4	75.6	78.8	82.1	11.4	5.9	4.2	0
D_{CA}(7Å)	60.3	73.3	77.5	85.1	70.6	77.6	81.1	85.1	10.3	4.3	3.6	0
D_{CA}(8Å)	62.6	75.2	81.3	88.2	72.9	80.7	84.3	88.2	10.3	5.5	3	0
D_{CA}(9Å)	67.4	80.5	85.1	91.6	76.7	84.4	87.9	91.6	9.3	3.9	2.8	0
D_{CA}(10Å)	71	83	86.8	93.3	78.6	85.9	89.5	93.3	7.5	3	2.8	0
D_{CA}(11Å)	75.8	87	89.9	95.2	82.2	89.1	91.7	95.2	6.4	2.1	1.8	0
D_{CA}(12Å)	77.7	88.2	91.2	96	83.2	89.8	92.2	96	5.4	1.6	1.1	0
D_{CA}(13Å)	80.3	89.5	92.6	96.6	84.7	91.3	93.6	96.6	4.5	1.8	0.9	0
D_{CA}(14Å)	82.4	90.8	93.7	97.3	86.2	93.1	95	97.3	3.9	2.3	1.3	0
D_{CA}(15Å)	85.5	92	94.5	97.7	88.3	94.4	96	97.7	2.8	2.4	1.4	0
D_{CC}(1Å)	2.3	2.7	2.9	2.9	2.6	2.9	2.9	2.9	0.3	0.2	0	0
D_{CC}(2Å)	8.2	10.9	11.6	12.8	11.4	12.2	12.6	12.8	3.2	1.3	1	0
D_{CC}(3Å)	17.2	22.3	23.7	26.1	22.5	24.6	25.2	26.1	5.3	2.3	1.4	0
D_{CC}(4Å)	25	32.6	34.9	38.9	32.6	36.2	37.4	38.9	7.6	3.7	2.5	0
D_{CC}(5Å)	32.8	41.8	45.6	51.3	42.4	47.4	49.3	51.3	9.6	5.6	3.7	0
D_{CC}(6Å)	37.8	49.2	52.9	60.1	49.2	55.2	57.6	60.1	11.4	6.1	4.7	0
D_{CC}(7Å)	44.3	56.3	60.5	68.3	55.3	62.5	65.1	68.3	10.9	6.2	4.6	0
D_{CC}(8Å)	49.6	62	66.4	74.4	60.6	68.4	71.1	74.4	11	6.4	4.7	0
D_{CC}(9Å)	54.6	65.8	70.8	79	64.4	72	75.3	79	9.7	6.2	4.5	0
D_{CC}(10Å)	55.9	67.9	73.1	81.9	66.3	74.1	77.7	81.9	10.4	6.2	4.6	0
D_{CC}(11Å)	59.7	72.3	77.9	85.9	69.9	78	82	85.9	10.2	5.7	4.1	0
D_{CC}(12Å)	64.7	76.7	81.3	89.1	72.7	80.2	84.5	89.1	8	3.5	3.2	0
D_{CC}(13Å)	67.6	80.5	84.5	91.2	75.3	82.7	86.3	91.2	7.6	2.2	1.9	0
D_{CC}(14Å)	71.4	83.8	87.2	93.1	79.1	86.1	89.1	93.1	7.7	2.3	1.9	0
D_{CC}(15Å)	75.6	88	91.4	96.2	82	90.1	93	96.2	6.4	2	1.7	0

Figure 4 Detailed results. Table and heatmap showing success rates [%] of Fpocket predictions for original and rescored output list of pockets together with the nominal improvements made by PRANK rescoring algorithm on CHEN11 dataset (measured by D_{CA} and D_{CC} criteria for different integer cutoff distances). For the D_{CA} criterion the biggest improvements were achieved around the meaningful 4-6 Å cutoff distances. Displayed results are averaged numbers from ten independent 5-fold cross-validation runs. Four columns in each group show success rates calculated considering progressively more predicted pockets ranked at the top (where *n* is the number of known ligand-binding sites of the protein that includes evaluated binding site). For protein with just one binding site they correspond to Top-1, Top-3 and Top-5 cutoffs that were commonly used to report results in previous ligand-binding site prediction studies.

Equation 7), even the predictions labeled as negative (having P_1 probability lower than 0.5) contribute to the score to some extent.

Discussion

Methods based on evolutionary conservation (such as ConCavity and LIGSITE^{csc}) are biased towards binding sites with biological ligands (meaning ligands that have their biological function i.e 'are supposed to bind there') and therefore can possibly ignore pockets that are not evolutionary conserved but still ligandable with respect to their physico-chemical properties. Those are perhaps the most interesting pockets because among them we can find novel binding sites for which synthetic ligands can be designed. Our method, on the other hand, is based only on local geometric and physico-chemical features of points near protein surface and therefore, we believe, not prone to such bias.

It can be argued that since our model is trained on a particular dataset, it is biased towards binding sites in this dataset. This is inherently a possible issue of all methods that are based on machine learning from examples. However, we believe that by training a classifier to predict ligandability of pocket points (that represent *local* chemical neighborhood rather than the whole pocket) we provided a way for sufficient generalization and therefore ability to correctly predict ligandability of novel sites.

While our rescoring method leads to significant improvements of the final success rates of binding site predictions, performance of the classifier itself is less satisfactory (see Table 2). Here, we will try to outline possible reasons. Several indicators point to the fact that the training data we are dealing with in the classification phase are very noisy.

This can be due to two main reasons: one is related to the feature extraction and the other, more fundamental,

Table 3 PRANK vs. simpler rescoring methods

Dataset	Top-n [%]	All [%]	PRANK [%]	Δ PRANK	PLB [%]	Δ PLB	VOL [%]	Δ VOL
Fpocket predictions								
CHEN11	47.9	71	58.8**	+10.6	49.8	+1.9	34.5	-13.4
ASTEX	58	81.1	63.6	+5.6	56.6	-1.4	32.2	-25.9
DT198	37.5	80.2	56.2	+18.8	43.2	+5.7	19.3	-18.2
MP210	56.6	78.8	67.7	+11.1	54.5	-2.1	30.6	-26
B48	74.1	92.6	81.5	+7.4	72.2	-1.9	42.6	-31.5
U48	53.7	88.9	77.8	+24.1	66.7	+13	31.5	-22.2
ConCavity predictions								
CHEN11	47.9	52.3	50.7**	+2.8	50.4	+2.5	50.2	+2.3
ASTEX	55.2	65.7	62.9	+7.7	62.9	+7.7	63.6	+8.4
DT198	45.8	65.6	61.5	+15.6	56.8	+10.9	59.4	+13.5
MP210	57.4	68.2	66.1	+8.7	64.9	+7.3	64.6	+6.9
B48	66.7	81.5	77.8	+11.1	79.6	+13	75.9	+9.3
U48	64.8	77.8	74.1	+9.3	75.9	+11.1	70.4	+5.6

PLB - rescoring by the Propensity for Ligand Binding index based on amino acid composition of pockets [29].
VOL - rescoring by approximate volume.
** cross-validation results.
The number presented for rescoring methods (columns: PRANK,PLB,VOL) is the success rate considering Top-n predicted pockets measured by D_{CA} criterion with 4 Å threshold.

has to do with completeness (or rather incompleteness) of the available experimental data.

Regarding the feature extraction, it is possible that (a) our feature set is not comprehensive enough and/or (b) we somehow dilute our feature vectors in the aggregation step mixing positives and negatives. While we cannot rule out the possibility that either could be the case, it is practically impossible to prove such a conclusion.

As for the available experimental data, on the other hand, it is easy to see how their inherent incompleteness could be contributing to the noisiness of our datasets. If we establish some region on protein's surface as a true ligand-binding site, this—by definition—means that there is an experimentally confirmed 3D structure complex available and thus there exists a ligand which binds at exactly that place. All positives in our datasets are therefore correctly labeled.

What about negatives? Negatives, in our case, are practically represented by everything else or more precisely all other points within the putative pockets. Hence, we can ask the following question: If a point near the protein surface is labeled as negative, does that mean that no ligand could bind at that place (because of its unfavorable physico-chemical properties), or do we simply not have a crystal structure where such event happens? We have no means of giving a definite answer to this question, but we suppose that some pockets are labeled as negatives incorrectly because of the inherent lack of complete experimental data (complete in a sense of confirming/ruling out binding with all possible ligands).

The dataset that was used to train our final classification model (CHEN11) had been constructed in a way that made the presence of false negatives less likely by including all known PDB ligands for the proteins present in the dataset. It is possible that it would prove better to work with much more narrowly defined negatives, that is, to take our negatives only from the putative pockets for which no ligand has been found despite a deliberate effort. However, this approach would have its own problems since examples of such cases are quite rare [30,52] and although they exist, they do not cover all structural diversity of whole PDB the way CHEN11 dataset does. Moreover, there are known cases when a ligand has been found for pockets that were previously deemed unliganda-ble [53]. Another source of more reliable negatives could be proteins deemed unligandable by physical fragment screens [54]. Nonetheless, as it could be quite interesting to see the effect it would have on the performance of our method, we shall leave it for the future research.

Conclusion

We introduced PRANK, a novel method to be used as a post processing step to any pocket identification method providing a rescoring mechanism to prioritize the predicted putative pockets. Since pocket prediction tools output many false positive results, a subsequent prioritization step can greatly boost the performance of such tools. PRANK is based on machine-learning providing the ability to predict ligandability of specific pocket points. The predictions are combined into a score for a given

putative pocket which is then used in the re-ranking phase. As demonstrated on multiple datasets using the examples of Fpocket and ConCavity, the method consistently increases the performance of the pocket detection methods by correct prioritization of the putative sites. PRANK is distributed as a freely available tool currently capable to work with the outputs of Fpocket and ConCavity, but it can be easily adapted to process an output from basically any pocket prediction tool. We believe that we have addressed a previously neglected problem of pocket scoring and thus the introduced method and the accompanying software present a valuable addition to the array of publicly available cheminformatics tools. PRANK is freely available at http://siret.ms.mff.cuni.cz/prank.

Endnote

[a] Although version 2.0 of Fpocket in its beta was available, we decided to use the version 1.0 since it consistently yielded better results.

Additional files

Additional file 1: Listings. Document that contains supplementary listings: (1) the complete list of properties of feature vectors used to represent inner points and (2) the lists of proteins by dataset for which ConCavity was run with the conservation mode switched off.

Additional file 2: Tables. Excel file that contains data used to produce tables and figures in this article.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
Both authors proposed the ranking algorithm based on pocket representation conceived by DH. RK proposed machine learning approach, designed and implemented the algorithm and performed the experiments. Manuscript was written by RK and DH. Both authors read and approved the final manuscript.

Acknowledgements
This work was supported by the Czech Science Foundation (GA CR) project 14-29032P.

Received: 29 September 2014 Accepted: 24 February 2015
Published online: 01 April 2015

References

1. Zheng X, Gan L, Wang E, Wang J. Pocket-based drug design: Exploring pocket space. AAPS J. 2013;15(1):228–41.
2. Pérot S, Sperandio O, Miteva M, Camproux A, Villoutreix B. Druggable pockets and binding site centric chemical space: a paradigm shift in drug discovery. Drug Discovery Today. 2010;15(15-16):656–67.
3. Xie L, Xie L, Bourne PE. Structure-based systems biology for analyzing off-target binding. Curr Opin Struct Biol. 2011;21(2):189–99.
4. Konc J, Janežič D. Binding site comparison for function prediction and pharmaceutical discovery. Curr Opin Struct Biol. 2014;25:34–9.
5. Weisel M, Proschak E, Schneider G. Pocketpicker: analysis of ligand binding-sites with shape descriptors. Chem Cent J. 2007;1(1):7.
6. Sotriffer C, Klebe G. Identification and mapping of small-molecule binding sites in proteins: computational tools for structure-based drug design. Il Farmaco. 2002;57(3):243–51.
7. Nisius B, Sha F, Gohlke H. Structure-based computational analysis of protein binding sites for function and druggability prediction. J Biotechnol. 2012;159(3):123–34.
8. Ghersi D, Sanchez R. EasyMIFS and SiteHound: a toolkit for the identification of ligand-binding sites in protein structures. Bioinf (Oxford, England). 2009;25(23):3185–6.
9. Le Guilloux V, Schmidtke P, Tuffery P. Fpocket: An open source platform for ligand pocket detection. BMC Bioinf. 2009;10(1):168.
10. Capra JA, Laskowski RA, Thornton JM, Singh M, Funkhouser TA. Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3d structure. PLoS Comput Biol. 2009;5(12):1000585.
11. Henrich S, Outi S, Huang B, Rippmann F, Cruciani G, Wade R. Computational approaches to identifying and characterizing protein binding sites for ligand design. J Mol Recognit: JMR. 2010;23(2):209–19.
12. Leis S, Schneider S, Zacharias M. In silico prediction of binding sites on proteins. Curr Med Chem. 2010;17(15):1550–62.
13. Hendlich M, Rippmann F, Barnickel G. LIGSITE: automatic and efficient detection of potential small molecule-binding sites in proteins. J Mol Graphics Modell. 1997;15(6):359–63389.
14. Huang B, Schroeder M. Ligsitecsc: predicting ligand binding sites using the connolly surface and degree of conservation. BMC Struct Biol. 2006;6(1):19.
15. Labute P, Santavy M. Locating Binding Sites in Protein Structures. (Online; accessed 2013-07-16). http://www.chemcomp.com/journal/sitefind.htm Accessed 2013-07-16.
16. Hajduk PJ, Huth JR, Tse C. Predicting protein druggability. Drug Discovery Today. 2005;10(23-24):1675–82.
17. Schmidtke P, Axel B, Luque F, Barril X. MDpocket: open-source cavity detection and characterization on molecular dynamics trajectories. Bioinf (Oxford, England). 2011;27(23):3276–85.
18. Laurie A, Jackson R. Q-SiteFinder: an energy-based method for the prediction of protein-ligand binding sites. Bioinf (Oxford, England). 2005;21(9):1908–16.
19. Schneider S, Zacharias M. Combining geometric pocket detection and desolvation properties to detect putative ligand binding sites on proteins. J Struct Biol. 2012;180(3):546–50.
20. Morita M, Nakamura S, Shimizu K. Highly accurate method for ligand-binding site prediction in unbound state (apo) protein structures. Proteins. 2008;73(2):468–79.
21. Roy A, Zhang Y. Recognizing protein-ligand binding sites by global structural alignment and local geometry refinement. Struct (London, England:1993). 2012;20(6):987–97.
22. Brylinski M, Skolnick J. A threading-based method (FINDSITE) for ligand-binding site prediction and functional annotation. Proc Nat Acad Sci USA. 2008;105(1):129–34.
23. Skolnick J, Brylinski M. FINDSITE: a combined evolution/structure-based approach to protein function prediction. Briefings Bioinf. 2009;10(4):378–91.
24. Skolnick J, Kihara D, Zhang Y. Development and large scale benchmark testing of the PROSPECTOR_3 threading algorithm. Proteins. 2004;56(3):502–18.
25. Huang B. MetaPocket: a meta approach to improve protein ligand binding site prediction. Omics: J integrative Biol. 2009;13(4):325–30.
26. Zhang Z, Li Y, Lin B, Schroeder M, Huang B. Identification of cavities on protein surface using multiple computational approaches for drug binding site prediction. Bioinf (Oxford, England). 2011;27(15):2083–8.
27. Schmidtke P, Barril X. Understanding and predicting druggability. a high-throughput method for detection of drug binding sites. J Med Chem. 2010;53(15):5858–67.
28. Krasowski A, Muthas D, Sarkar A, Schmitt S, Brenk R. Drugpred: a structure-based approach to predict protein druggability developed using an extensive nonredundant data set. J Chem Inf Model. 2011;51(11):2829–42.
29. Soga S, Shirai H, Kobori M, Hirayama N. Use of amino acid composition to predict ligand-binding sites. J Chem Inf Model. 2007;47(2):400–6. PMID: 17243757.
30. Schmidtke P. Protein-ligand binding sites Identification, characterization and interrelations. PhD thesis, University of Barcelona (September 2011).
31. Chen K, Mizianty M, Gao J, Kurgan L. A critical comparative assessment of predictions of protein-binding sites for biologically relevant organic compounds. Struct (London, England: 1993). 2011;19(5):613–21.

32. Fariselli P, Pazos F, Valencia A, Casadio R. Prediction of protein–protein interaction sites in heterocomplexes with neural networks. Eur J Biochemistry/FEBS. 2002;269(5):1356–61.
33. Bordner AJ. Predicting small ligand binding sites in proteins using backbone structure. Bioinf (Oxford, England). 2008;24(24):2865–71.
34. Sikic M, Tomic S, Vlahovicek K. Prediction of protein-protein interaction sites in sequences and 3d structures by random forests. PLoS Computational Biol. 2009;5(1):1000278.
35. Zhou H-X, Shan Y. Prediction of protein interaction sites from sequence profile and residue neighbor list. Proteins: Struct Funct Bioinf. 2001;44(3): 336–43.
36. Xiong Y, Xia J, Zhang W, Liu J. Exploiting a reduced set of weighted average features to improve prediction of dna-binding residues from 3d structures. PloS one. 2011;6(12):28440.
37. Nayal M, Honig B. On the nature of cavities on protein surfaces: application to the identification of drug-binding sites. Proteins. 2006;63(4):892–906.
38. Connolly M. Solvent-accessible surfaces of proteins and nucleic acids. Science. 1983;221(4612):709–13.
39. Kyte J, Doolittle RF. A simple method for displaying the hydropathic character of a protein. Journal of Molecular Biology. 1982;157(1):105–32.
40. Desaphy J, Azdimousa K, Kellenberger E, Rognan D. Comparison and druggability prediction of protein-ligand binding sites from pharmacophore-annotated cavity shapes. J Chem Inf Model. 2012;52(8): 2287–99.
41. Khazanov NA, Carlson HA. Exploring the composition of protein-ligand binding sites on a large scale. PLoS Comput Biol. 2013;9(11):1003321.
42. Pintar A, Carugo O, Pongor S. Cx, an algorithm that identifies protruding atoms in proteins. Bioinformatics. 2002;18(7):980–4.
43. Eisenhaber F, Lijnzaad P, Argos P, Sander C, Scharf M. The double cubic lattice method: Efficient approaches to numerical integration of surface area and volume and to dot surface contouring of molecular assemblies. Journal of Computational Chemistry. 1995;16(3):273–84.
44. Breiman L. Random forests. Machine Learning. 2001;45(1):5–32.
45. Svetnik V, Liaw A, Tong C, Culberson JC, Sheridan RP, Feuston BP. Random forest: a classification and regression tool for compound classification and qsar modeling. Journal of chemical information and computer sciences. 2003;43(6):1947–58.
46. Boulesteix A-L, Janitza S, Kruppa J, K-nig IR. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. Wiley Interdisciplinary Rev: Data Min Knowledge Discovery. 2012;2(6):493–507.
47. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The weka data mining software: an update. ACM SIGKDD Explorations Newsletter. 2009;11(1):10–8.
48. Prlic A, Yates A, Bliven SE, Rose PW, Jacobsen J, Troshin PV, et al. Biojava: an open-source framework for bioinformatics in 2012. Bioinf (Oxford, England). 2012;28(20):2693–5.
49. Steinbeck C, Han Y, Kuhn S, Horlacher O, Luttmann E, Willighagen E. The chemistry development kit (cdk): An open-source java library for chemo- and bioinformatics. J Chem Inf Comput Sci. 2003;43(2):493–500. PMID: 12653513.
50. Hartshorn M, Verdonk M, Chessari G, Brewerton S, Mooij W, Mortenson P, et al. Diverse, high-quality test set for the validation of protein-ligand docking performance. J Med Chem. 2007;50(4):726–41.
51. ConCavity Website. http://compbio.cs.princeton.edu/concavity/.
52. Hajduk PJ, Huth JR, Fesik SW. Druggability indices for protein targets derived from nmr-based screening data. J Med Chem. 2005;48(7):2518–25.
53. Filippakopoulos P, Qi J, Picaud S, Shen Y, Smith WB, Fedorov O, et al. Selective inhibition of bet bromodomains. Nature. 2010;468(7327): 1067–73.
54. Hajduk PJ. Sar by nmr: putting the pieces together. Mol Interventions. 2006;6(5):266–72.

Additional Listings File

Supplementary information to the article

Improving ligand-binding site prediction accuracy by classification of inner pocket points using local features

Radoslav Krivák and David Hoksza

1. Complete list of features

Complete list of properties of feature vectors used to represent inner pocket points.

Feature name	description
`hydrophobic`	binary attribute, 1 for hydrophobic residues
`hydrophilic`	binary attribute, 1 for hydrophilic residues
`hydrophatyIndex`	side-chain hydrophaty index with values in range ⟨−4.5,4.5⟩ [1]
`aliphatic`	binary attribute, 1 for aliphatic residues
`aromatic`	binary attribute, 1 for aromatic residues
`sulfur`	binary attribute, 1 for residues containing sulfur
`hydroxyl`	binary attribute, 1 for hydroxyl group containing residues
`basic`	binary attribute, 1 for basic residues
`acidic`	binary attribute, 1 for acidic residues
`amide`	binary attribute, 1 for amide group containing residues
`posCharge`	binary attribute, 1 for positively charged residues
`negCharge`	binary attribute, 1 for negatively charged residues
`hBondDonor`	binary attribute, 1 for H-bond donor containing residues
`hBondAcceptor`	binary attribute, 1 for H-bond acceptor containing residues
`hBondDonorAcceptor`	binary attribute, 1 for residues that have H-bond donor AND acceptor
`polar`	binary attribute, 1 for polar residues
`ionizable`	binary attribute, 1 for ionizable residues
`atoms`	absolute number of protein exposed atoms (within 8 Å radius of the point)
`atomDensity`	number of protein exposed atoms weighted by distance
`atomC`	number of carbon atoms
`atomO`	number of oxygen atoms
`atomN`	number of nitrogen atoms
`hDonorAtoms`	number of H-bond donor atoms

hAcceptorAtoms	number of H-bond acceptor atoms
vsAromatic	VolSite atomic level features [2]
vsCation	~
vsAnion	~
vsHydrophobic	~
vsAcceptor	~
vsDonor	~
ap5sasaValids	Ligand binding propensity for biologically valid ligands [3]
ap5sasaInvalids	Ligand binding propensity for biologically invalid ligands [3]
protrusion	Protein surface protrusion inspired by [4] calculated simply as number of all protein atoms (not just exposed) within 10 Å radius of the point
bfactor	B-factor number of the atom from pdb file

References:

1. Kyte, J., Doolittle, R.F.: A simple method for displaying the hydropathic character of a protein. Journal of Molecular Biology 157(1), 105–132 (1982).
2. Desaphy, J., Azdimousa, K., Kellenberger, E., Rognan, D.: Comparison and druggability prediction of protein-ligand binding sites from pharmacophore-annotated cavity shapes. Journal of chemical information and modeling 52(8), 2287–2299 (2012).
3. Khazanov, N.A., Carlson, H.A.: Exploring the composition of protein-ligand binding sites on a large scale. PLoS computational biology 9(11), 1003321 (2013).
4. Alessandro Pintar, Oliviero Carugo, and Sándor Pongor: CX, an algorithm that identifies protruding atoms in proteins. Bioinformatics (2002) 18 (7): 980-984

2. ConCavity no-conservation proteins

Lists of proteins (by dataset) for which ConCavity was run with the conservation mode switched off. Those are the proteins for which pre-computed sequence conservation files were not available or ConCavity failed to produce any results in conservation mode even if they were.

CHEN11 - 16/251	a.003.001.004_1m6zb.pdb a.138.001.003_1qdbb.pdb b.042.002.001_2zqnb.pdb b.085.007.002_2g46a.pdb b.089.001.001_1iiya.pdb c.001.013.001_1p7tb.pdb c.002.001.003_2g82c.pdb c.025.001.004_1ja1b.pdb c.062.001.001_3bqmb.pdb c.065.001.001_2blnb.pdb c.087.001.001_1m5rb.pdb

	c.087.001.010_2c1xa.pdb d.001.001.004_1bvic.pdb d.019.001.001_2akrc.pdb d.110.006.001_1p0zg.pdb e.003.001.001_2hdub.pdb
ASTEX - 2/85	1hnn.pdb 1oyt.pdb
UB48 - 5/96	1dwd.pdb 1hxf.pdb 1ida.pdb 1pso.pdb 3gch.pdb
DT198 - 19/198	1cea_A.pdb 1fj8_A.pdb 1lxf_C.pdb 1pk2_A.pdb 1q8y_B.pdb 1y4l_B.pdb 2cft_A.pdb 2xh1_A.pdb 2xhd_A.pdb 2zt7_A.pdb 3d90_A.pdb 3gmz_A.pdb 3h6t_A.pdb 3ii0_A.pdb 3inj_A.pdb 3iyt_A.pdb 3k4v_A.pdb 3kvv_A.pdb 3l6b_A.pdb
MP210 - 4/210	1ac0.pdb 1b6n.pdb 2er0.pdb 3gch.pdb

P2RANK: Knowledge-Based Ligand Binding Site Prediction Using Aggregated Local Features

Reference

KRIVÁK R., HOKSZA D.: **P2RANK: Knowledge-Based Ligand Binding Site Prediction Using Aggregated Local Features**. In *International Conference on Algorithms for Computational Biology* (2015), Springer, pp. 41–52. doi:10.1007/978-3-319-21233-3_4

Author's highlights

Building on PRANK method we have deceloped P2Rank: a method for prediction of ligand binding sites. This conference article contains the cleanest exposition of P2Rank algorithm itself.

Note: in this paper we have used the term Connolly surface referring to the surface which would be more precisely described as solvent accessible surface.

P2RANK: knowledge-based ligand binding site prediction using aggregated local features

Radoslav Krivák and David Hoksza

[1] Charles University in Prague, FMP, Department of software engineering, Malostranské nám. 25, 118 00, Prague, Czech Republic
krivak@ksi.mff.cuni.cz
[2] Charles University in Prague, FMP, Department of software engineering, Malostranské nám. 25, 118 00, Prague, Czech Republic
hoksza@ksi.mff.cuni.cz,
WWW home page: http://siret.cz/hoksza

Abstract. The knowledge of protein-ligand binding sites is vital prerequisite for any structure-based virtual screening campaign. If no prior knowledge about binding sites is available, the ligand-binding site prediction methods are the only way to obtain the necessary information. Here we introduce P2RANK, a novel machine learning-based method for prediction of ligand binding sites from protein structure. P2RANK uses Random Forests learner to infer ligandability of local chemical neighborhoods near the protein surface which are represented by specific near-surface points and described by aggregating physico-chemical features projected on those points from neighboring protein atoms. The points with high predicted ligandability are clustered and ranked to obtain the resulting list of binding site predictions. The new method was compared with a state-of-the-art binding site prediction method Fpocket on three representative datasets. The results show that P2RANK outperforms Fpocket by 10 to 20 percentage points on all the datasets. Moreover, since P2RANK does not rely on any external software for computation of various complex features, such as sequence conservation scores or binding energies, it represents an ideal tool for inclusion into future structural bioinformatics pipelines.

Keywords: ligand-binding site prediction, protein structure, molecular recognition, machine learning, random forest

1 Introduction

1.1 Motivation

Prediction of ligand binding sites from protein structure has many applications, ranging from use in rational drug design [30, 44], drug side-effects prediction [42] to elucidation of protein function [18]. Of special interest is the application in structure based virtual screening (SBVS) pipelines. In most types of SBVS, docking algorithms are used to predict possible ligand-binding interactions. It is

recommended to focus docking to a protein cavity of interest to limit the search space of possible conformations. In the cases where there is no a priori information with regard to which protein regions to focus on (e.g. confirmed active sties), it may be necessary to perform blind docking which scans the whole protein surface. Compared to local docking it is generally less accurate and significantly more time consuming, which limits the size of compound libraries that is possible to screen [34]. Alternatively, ligand binding site prediction can be employed in such scenarios to generate and prioritize the locations on which to center subsequent docking procedure [23]. In a similar manner, binding site prediction could also be of great use in a related task of structure-based target prediction (or so called inverse virtual screening) [37]. As a result of the structural genomic efforts [28], many protein structures still lack functional annotation and even if it is present, it may not be complete. We believe that accurate ligand binding site prediction methods (in combination with validation via docking) can help to discover new and potentially useful allosteric binding sites.

1.2 Existing methods

Many different ligand binding site prediction methods based on various strategies have been already developed. The first dedicated method was proposed in 1992 [26] and the recent increase of interest in the field, presumably due to rapid increase in number of available protein structures, is indicated by the number of recently published reviews [6, 13, 23, 25, 30]. Several categories of methods (or rather distinctive approaches) have been recognized. We present them together with their representative examples, although in reality the actual methods may use a combination of those approaches:

- **Geometrical methods.** Methods focused mainly on the algorithmic side of the problem of finding concave pockets and clefts on the surface of a 3D structure [12,15,41], some of them incorporating additional physico-chemical information like polarity or charge [21,24].
- **Energetic methods.** Methods that build on the approximations of free energy potentials by force fields, placing probes around the protein surface and calculating binding energies [1,10,22,27,36].
- **Evolutionary methods.** Algorithms that make use of sequence conservation estimates (functional residues are more evolutionary conserved) [5,15], or protein threading (fold recognition from sequence) to find set of evolutionary related structures and determine their common binding sites [4,38].
- **Knowledge based methods.** Methods that try to capture and exploit the knowledge about protein-ligand binding that is implicitly stored in sequence and structural databases by means of statistical inference [34]. Although several residue-centric studies focused on classification of ligand binding residues have been published [7, 16, 32], there are not many examples of complete prediction methods using this approach that would produce putative binding sites as such [33].

– **Consensus methods.** Those are meta approaches that combine the results of other methods [6, 14, 43].

In studies that introduced respective methods relatively high identification success rates have been reported (usually more than 70% considering only the first predicted binding site). However, the results of the only independent benchmarking study [6] suggest that accuracy of many of the methods may not be as good as previously believed (closer to 50%). It showed that there is still a need for more accurate methods and thus an opportunity for improvement by examining new approaches to binding site prediction. On a practical side, only few of the methods are available for download as ready to use free software packages.

2 Materials and methods

2.1 Method outline

In this paper we are introducing a novel method for prediction of ligand binding sites form a protein structure. Method is named P2RANK and it represents an evolutionary improvement of our pocket ranking method PRANK [19] which could only be used to reorder output of other pocket prediction methods. By not relying on other methods to generate putative binding site locations and thus making it a full-fledged method that can generate predictions itself has led to a marked improvements in identification success rates.

The method takes a PDB structure as an input and outputs a ranked list of predicted ligand binding sites defined by a set of points The following list outlines the proposed method:

1. Generating a set of regularly spaced points lying on a protein's Connolly surface (referred to as *Connolly points*).
2. Calculating feature descriptors of Connolly points based on their local chemical neighborhood:
   a) computing property vectors for protein's solvent exposed atoms,
   b) projecting distance weighted properties of the adjacent protein atoms onto Connolly points,
   c) computing additional features describing Connolly point neighborhood.
3. Predicting ligandability score of Connolly points by Random Forest classifier.
4. Clustering points with high ligandability score and thus forming pocket predictions.
5. Ranking predicted pockets by cumulative ligandability score of their points.

Individual steps are described in greater detail in following paragraphs. For visualization of Connolly points see Figure 1. One possible way how to look at our protein surface representation is that protein solvent exposed atoms produce potential fields for every feature (e.g. hydrophobicity, aromaticity, ...), and Connolly points are sampling values from those fields at places near the protein surface, which are likely to harbor potential ligand atoms.

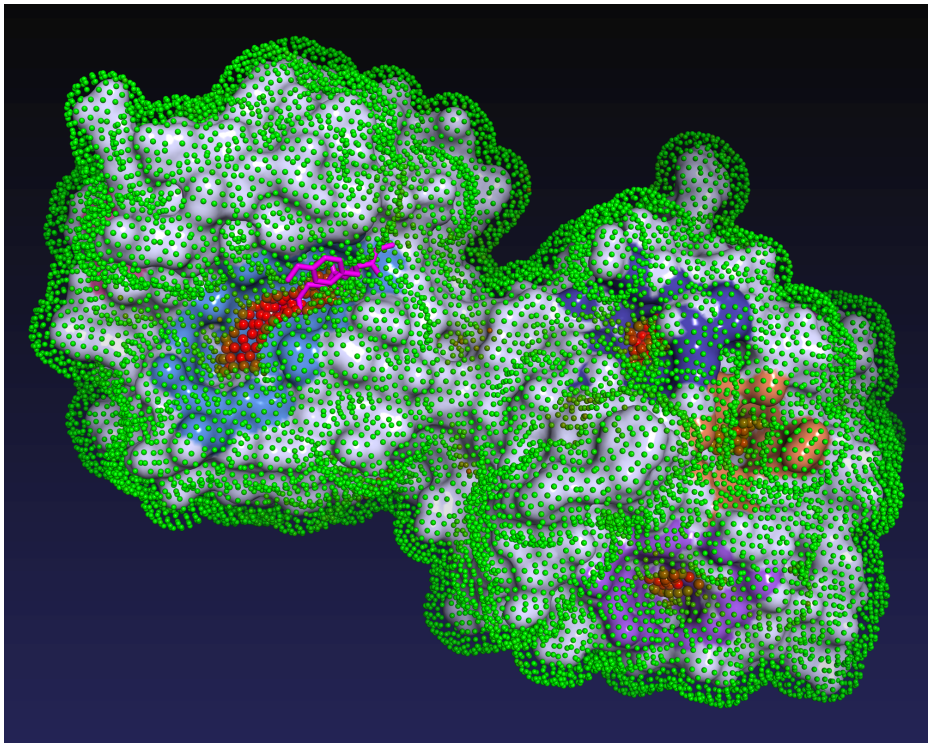


Fig. 1: **Connolly points**. Protein (1FBL) is covered in a layer of points lying on a Connolly surface. Each point represents its local chemical neighborhood and is colored according to its predicted ligandability score (*green=0/red=0.7*). Points deemed highly ligandable by a threshold (*displayed slightly enlarged*) are clustered to form predicted pockets (*highlighted by coloring adjacent protein surface by different colors*). In this case, the largest predicted pocket (*shown on the left*) is indeed a correctly identified true binding site that binds a ligand (*magenta*). Visible are three other smaller predicted pockets, or rather hotspots (*shown on the right*). Cumulative ligandability score of their respective points is lower, therefore they will be ranked lower than the true pocket on the resulting list of predicted binding sites.

Connolly points (1.) Position of Connolly points is generated by a fast numerical algorithm [9] implemented in CDK library [39]. The algorithm produces a set of a more or less regularly spaced points lying on a Connolly surface of the protein. Solvent radius used is 1.6 Å (this value as well as other arbitrary parameters was optimized, see Results section). The density of the points depend on an integer parameter tessellation level with default value 2 that produces points with approximately 1.5 Å spacing.

Feature representation (2.) For each Connolly point a feature vector (CFV) that describes its local physico-chemical neighborhood is calculated. Before calculating CFVs, each solvent exposed heavy atom of the protein is assigned atomic feature vector (AFV) that describes given atom. CFV of a given Connolly point is then calculated by aggregating AFVs of neighboring atoms and adding additional Connolly point features (XFV), i.e. extra features that are not defined for atoms.

Atomic neighborhood of Connolly point P is defined as:

$$\mathrm{A}(P) = \{\text{solvent exposed heavy protein atoms within } r\text{=6 Å radius around } P\} \quad (1)$$

The following aggregation function is used to project AFVs onto the Connolly points and calculate CFV for point P:

$$\mathrm{CFV}(P) = \sum_{\mathrm{A}_i \in \mathrm{A}(P)} \mathrm{AFV}\,(\mathrm{A}_i) \cdot w(\mathrm{dist}(P, \mathrm{A}_i)) \quad || \quad \mathrm{XFV}(P), \quad (2)$$

where $||$ is the operator of concatenation, XFV is a vector of additional features specific to Connolly points and w is a distance weight function:

$$w(d) = 1 - d/6. \quad (3)$$

AFV that describes protein atoms consists of two types of features: residue level features (inherited by all atoms of a given residue) and atomic level features. Residue level features include e.g. physico-chemical properties of standard amino acids or hydropathy index of amino acids [20]. Examples of atomic features are pharmacophore-related labels of atoms adopted from VolSite druggability prediction study [8] or statistical ligand-binding propensities of amino acid atoms [17]. Most of the features are table features defined either for 20 standard amino acids or their atom types (ALA.CA, ALA.CB,...). Exception to this is temperature factor, taken directly from PDB file, which can be different for each atom. Extra Connolly point features (XFV), which are not defined for atoms, include the number of neighboring H-bond donors and acceptors and protrusion index [31]. Protrusion is defined as a density of a protein atoms around the point and is calculated using larger neighborhood cutoff radius (10 Å). Altogether, CFV consists of 34 features. For their complete list and description we refer the reader to [19].

Classification (3.) Machine learning approach was used to classify Connolly points as ligandable/unligandable from their feature vectors. In general, output of a binary classifier is a number between 0 and 1 that represents certainty of a trained model that the classified instance belongs to the particular class (here class_1=ligandable). Commonly, a threshold optimizing certain metric is chosen and applied to produce binary output. In our case we decided to work directly with output score (rather than binary output) which we refer to as a predicted ligandability score (LS).

In theory any classification algorithm can be employed at this stage. After preliminary experiments with several machine learning methods we decided to adopt Random Forests [3] as our predictive modelling tool of choice. Random Forests is an ensemble of trees created by using bootstrap samples of training data and random feature selection in tree induction [40]. In comparison with other machine learning approaches, Random Forests are characterized by an outstanding speed (both in learning and execution phase) and generalization ability [3]. Additionally, Random Forests is robust to the presence of a large number of irrelevant variables; it does not require their prior scaling [29] and can cope with complex interaction structures as well as highly correlated variables [2].

Random Forests algorithm has 3 basic hyperparameters: number of trees, maximum tree depth and a number of random features used to construct each tree. To train the final model we used Random Forest with 100 trees of unlimited depth, each tree built considering 6 features. Model is trained on a dataset of ligandable and unligandable points that come from PDB structures with known ligand positions. To train our final model which we distribute with our software we used protein/ligand complexes from CHEN11 dataset (see Datasets section).

Clustering (4.) To prepare putative binding site predictions we first filter out Connolly points that have ligandability score lower than give threshold (default t=0.35) and apply single linkage clustering procedure on the rest (default cutoff distance d=3 Å). Predicted pocket is then defined by the set of Connolly points in a cluster. For each pocket we compute associated set of protein solvent exposed atoms that form putative ligand binding surface patch. We include into the output all pockets that are defined by 3 or more Connolly points. This is rather low threshold, which results to many small predicted pockets that are most probably not true binding sites. However, this was a deliberate choice as thanks to an efficient ranking algorithm those small pockets will always be ranked at the bottom of the list. Nevertheless, those small clusters might be still interesting for visual inspection (possibly forming hotspots for protein-protein interactions or peptide binding).

Ranking (5.) Each pocket is assigned a score calculated as the sum of squared ligandability scores of all of the Connolly points P_i that define the pocket:

$$\text{PScore} = \sum_i \big(\text{LS}(P_i)\big)^2 \tag{4}$$

Squaring of the ligandability scores puts more emphasis on the points with ligandability score closer to 1 (i.e. points that were classified as ligandable with higher certainty). Score defined in such way will roughly order pockets by size but will favor smaller pockets with strongly ligandable points before larger pockets with weakly ligandable points. The very last step of the algorithm involves reordering the putative pockets in the decreasing order of their *PScores*.

Ranking of the predicted pockets is important for prioritization of subsequent efforts, e.g. docking or visual inspection. Pocket ranking is also pivotal in the

context of evaluation and comparison of different ligand-binding site prediction methods, where only pockets with highest ranks are considered (usually Top-1 and Top-3). If it was not so, then the simplistic and obviously useless method that returns many binding sites covering all of the protein surface (most of them false positives) would achieve 100% identification success rate.

2.2 Datasets

For the purpose of training an evaluation of our method we have used three datasets:

- **CHEN11** – dataset introduced in benchmarking study [6]. A non-redundant dataset constructed in a way so that each SCOP family has one typical representative.
- **JOINED** – consists of structures from several smaller datasets used in previous studies (48bound/unbound structures [15], ASTEX [11], 198 drug targets [43], 210 bound proteins [14]) joined into one larger dataset.
- **HOLO4K** – large dataset of protein-ligand complexes currently available in PDB based on the list published in [35].

Details of the datasets are presented in Table 1.

Table 1: Datasets.

Dataset	Proteins	Ligands	avg. ligands	avg. lig. atoms	avg. prot. atoms
CHEN11	251	374	1.49	26.9	1836
JOINED	589	689	1.17	22.5	2400
HOLO4K	4543	11511	2.54	22.4	3888

3 Experimental evaluation

3.1 Evaluation measures

To evaluate predictive performance of our method and compare it with Fpocket we have used methodology based on ligand-centric counting and D_{CA} (distance between the center of the pocket and any ligand atom) pocket identification criterion with 4 Å threshold. Ligand-centric counting means, that for every relevant ligand in the dataset, its binding site must be correctly predicted for a method to achieve 100% identification success rate. Connected to this is the use of Top-n and Top-(n+2) rank cutoffs where n is the number of ligands in a protein structure where evaluated ligand comes from (for proteins with only

one ligand this corresponds to usual Top-1 and Top-3 cutoffs). This evaluation methodology is the same as used in benchmarking study [6].

Because of the great differences in evaluation protocols used to produce results of previously published methods, we are of the opinion that the only way how to accurately compare ligand binding site prediction methods is to preform experiments and compare methods side by side using the same methodology (as opposed to using results taken from literature even if experiments are performed on the same dataset). Aforementioned differences in protocols include: different identification criteria (D_{CA}/D_{CC} (center-center distance)/variously defined volume overlap criteria), different counting strategies (ligand-centric/protein-centric), different valid ligand selection and different rank cutoffs considered (Top-1/Top-3/Top-4/Top-n). Experimentally comparing ligand binding site prediction methods is complicated and lengthy effort involving many technical hurdles. Nevertheless, instead of reporting here the results of our method side-by-side with results taken from literature we compare it thoroughly on large datasets with Fpocket and our previous ranking method (which improves results of Fpocket by reordering its output). The significance of our results with respect to other methods can be inferred by comparing our presented results with mentioned independent benchmarking study which includes Fpocket [6].

3.2 Results

We have evaluated our method on 3 different datasets and compared its predictive performance with a well-known Fpocket method. Results are summarized in Table 2. Our method comes with a pre-trained classification model that was trained on the CHEN11 dataset. On JOINED and HOLO4K datasets we report results achieved using this default model and on CHEN11 dataset averaged results from 10 runs of 5-fold cross-validation. Success rates (percentage of correctly predicted binding sites) are reported for Top-n and Top-(n+2) cutoffs from the top of the ranked list.

Table 2: Results: the numbers represent identification success rate [%] measured by D_{CA} criterion with 4 Å threshold considering only pockets ranked at the top of the list (n is the number of ligands in considered structure). *average results of 10 independent 5-fold cross-validation runs.

	Top-n			Top-(n+2)		
Dataset	Fpocket	PRANK	P2RANK	Fpocket	PRANK	P2RANK
CHEN11	47.8	58.6*	**59.2***	61.5	**68.1***	65.9*
JOINED	51.1	64.7	**71.6**	68.9	76.1	**78.7**
HOLO4K	45.2	53.6	**63.9**	55.1	61.6	**69.8**

It is apparent that P2RANK outperforms Fpockets on all dataset by a large margin. The difference is most visible comparing results for Top-1 cutoffs. For a difficult CHEN11 dataset a difference amounts to more than 10 percent in nominal terms and almost 20 for other datasets. Detailed results on HOLO4K dataset are compared in Figure 2.

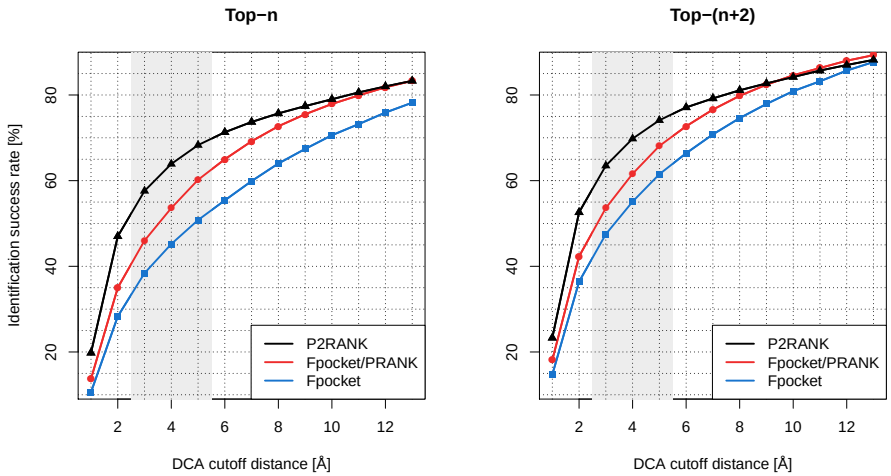


Fig. 2: Results on HOLO4K dataset for different D_{CA} cutoff distances.

3.3 Optimization and tradeoffs

Extensive optimization of practically all arbitrary parameters of the algorithm (distance cutoffs, thresholds, ...) was performed to establish optimal default values. Parameters were optimized with respect to the success rate achieved on the JOINED dataset. This was done to avoid bias towards CHEN11 dataset so the cross-validation results on the CHEN11 could be compared with benchmarking study which used the same evaluation criteria [6]. We have also refrained from tweaking the parameters with respect to HOLO4K dataset, so that the results on this dataset present unbiased estimate of the algorithm's true identification success rate on unknown input.

Several parameters present a tradeoff between the time and space complexity of the algorithm and its accuracy. Among those is the number of trees in Random Forest model and tessellation level influencing density of the generated Connolly points. Ultimately we have decided to use 100 trees (using 10× more trees leads only to marginal improvements) and tessellation level of 2 (using higher levels leads to some improvements but also unproportionally longer running times).

4 Conclusion

In the present paper we have proposed P2RANK, a novel method for ligand-binding site prediction based on classification of points lying on a protein's Connolly surface. Each point represents potential location of a binding ligand atom and is described by a feature vector generated from its spatial neighborhood. Ligandability score is predicted for each point by a Random Forests classifier and points with higher score are clustered forming predicted pockets. Pockets are then ranked according to the cumulative ligandability score of their points.

To our knowledge this is a first time a machine learning approach was used in such a manner for ligand binding site prediction. Methods that applied machine learning in this context focused on classification of ligand-binding residues i.e. were residue-centric. Unfortunately, most of those residue-centric studies were focused on a successful classification of residues themselves and not on predicting ligand binding sites as such.

We showed on several datasets that P2RANK significantly improves identification success over the state of the art method Fpocket, while still being reasonably fast to be used on large datasets. Like Fpocket, P2RANK is a stand-alone program that is ready to be used as is, without depending on any external data or programs or secondary inputs such as pre-computed sequence conservation scores, forcefield calculations or threading template libraries. We believe that it is a viable method with a potential to become useful part of structural bioinformatics toolkit.

Acknowledgments

This work was supported by the Czech Science Foundation grant 14-29032P and by project SVV-2015-260222.

References

1. J. An, M. Totrov, and R. Abagyan. Pocketome via comprehensive identification and classification of ligand binding envelopes. *Molecular & Cellular Proteomics*, 4(6):75261, 2005.
2. A.-L. Boulesteix, S. Janitza, J. Kruppa, and I. R. K-nig. 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.
3. L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
4. M. Brylinski and J. Skolnick. A threading-based method (FINDSITE) for ligand-binding site prediction and functional annotation. *Proceedings of the National Academy of Sciences of the United States of America*, 105(1):129–134, 2008.
5. J. A. Capra, R. A. Laskowski, J. M. Thornton, M. Singh, and T. A. Funkhouser. Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3d structure. *PLoS Comput Biol*, 5(12):e1000585, 12 2009.

6. K. Chen, M. Mizianty, J. Gao, and L. Kurgan. A critical comparative assessment of predictions of protein-binding sites for biologically relevant organic compounds. *Structure (London, England : 1993)*, 19(5):613–621, 2011.
7. P. Chen, J. Z. Huang, and X. Gao. Ligandrfs: random forest ensemble to identify ligand-binding residues from sequence information alone. *BMC bioinformatics*, 15 Suppl 15:S4, Jan 2014.
8. J. Desaphy, K. Azdimousa, E. Kellenberger, and D. Rognan. Comparison and druggability prediction of protein-ligand binding sites from pharmacophore-annotated cavity shapes. *J. Chem. Inf. Model.*, 52(8):2287–2299, 2012.
9. F. Eisenhaber, P. Lijnzaad, P. Argos, C. Sander, and M. Scharf. The double cubic lattice method: Efficient approaches to numerical integration of surface area and volume and to dot surface contouring of molecular assemblies. *Journal of Computational Chemistry*, 16(3):273–284, 1995.
10. D. Ghersi and R. Sanchez. EasyMIFS and SiteHound: a toolkit for the identification of ligand-binding sites in protein structures. *Bioinformatics (Oxford, England)*, 25(23):3185–3186, 2009.
11. M. Hartshorn, M. Verdonk, G. Chessari, S. Brewerton, W. Mooij, P. Mortenson, and C. Murray. Diverse, high-quality test set for the validation of protein-ligand docking performance. *Journal of medicinal chemistry*, 50(4):726–741, 2007.
12. M. Hendlich, F. Rippmann, and G. Barnickel. LIGSITE: automatic and efficient detection of potential small molecule-binding sites in proteins. *Journal of molecular graphics & modelling*, 15(6):359–63, 389, 1997.
13. S. Henrich, S. Outi, B. Huang, F. Rippmann, G. Cruciani, and R. Wade. Computational approaches to identifying and characterizing protein binding sites for ligand design. *Journal of molecular recognition : JMR*, 23(2):209–219, 2010.
14. B. Huang. MetaPocket: a meta approach to improve protein ligand binding site prediction. *Omics : a journal of integrative biology*, 13(4):325–330, 2009.
15. B. Huang and M. Schroeder. Ligsitecsc: predicting ligand binding sites using the connolly surface and degree of conservation. *BMC Structural Biology*, 6(1):19, 2006.
16. C. Kauffman and G. Karypis. Librus: combined machine learning and homology information for sequence-based ligand-binding residue prediction. *Bioinformatics (Oxford, England)*, 25(23):3099107, Dec 2009.
17. N. A. Khazanov and H. A. Carlson. Exploring the composition of protein-ligand binding sites on a large scale. *PLoS computational biology*, 9(11):e1003321, Nov 2013.
18. J. Konc and D. Janei. Binding site comparison for function prediction and pharmaceutical discovery. *Current opinion in structural biology*, 25:34–9, Apr 2014.
19. R. Krivak and D. Hoksza. Improving protein-ligand binding site prediction accuracy by classification of inner pocket points using local features. *Journal of Cheminformatics*, 7(1):12, 2015.
20. J. Kyte and R. F. Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, 157(1):105 – 132, 1982.
21. P. Labute and M. Santavy. Locating binding sites in protein structures. `http://www.chemcomp.com/journal/sitefind.htm`, 2001. (Online; accessed 2013-07-16).
22. A. Laurie and R. Jackson. Q-SiteFinder: an energy-based method for the prediction of protein-ligand binding sites. *Bioinformatics (Oxford, England)*, 21(9):1908–1916, 2005.
23. A. Laurie and R. Jackson. Methods for the prediction of protein-ligand binding sites for structure-based drug design and virtual ligand screening. *Current protein & peptide science*, 7(5):395406, 2006.

24. V. Le Guilloux, P. Schmidtke, and P. Tuffery. Fpocket: An open source platform for ligand pocket detection. *BMC Bioinformatics*, 10(1):168, 2009.
25. S. Leis, S. Schneider, and M. Zacharias. In silico prediction of binding sites on proteins. *Current medicinal chemistry*, 17(15):1550–1562, 2010.
26. D. G. Levitt and L. J. Banaszak. Pocket: A computer graphies method for identifying and displaying protein cavities and their surrounding amino acids. *Journal of Molecular Graphics*, 10(4):229 – 234, 1992.
27. M. Morita, S. Nakamura, and K. Shimizu. Highly accurate method for ligand-binding site prediction in unbound state (apo) protein structures. *Proteins*, 73(2):468–79, Nov 2008.
28. R. Nair, J. Liu, T.-T. Soong, T. Acton, J. Everett, A. Kouranov, A. Fiser, A. Godzik, L. Jaroszewski, C. Orengo, and et al. Structural genomics is the largest contributor of novel structural leverage. *Journal of Structural and Functional Genomics*, 10(2):18191, 2009.
29. M. Nayal and B. Honig. On the nature of cavities on protein surfaces: application to the identification of drug-binding sites. *Proteins*, 63(4):892–906, Jun 2006.
30. S. Pérot, O. Sperandio, M. Miteva, A. Camproux, and B. Villoutreix. Druggable pockets and binding site centric chemical space: a paradigm shift in drug discovery. *Drug discovery today*, 15(15-16):656–667, 2010.
31. A. Pintar, O. Carugo, and S. Pongor. Cx, an algorithm that identifies protruding atoms in proteins. *Bioinformatics*, 18(7):980–984, 2002.
32. Z. Qiu, C. Qin, M. Jiu, and X. Wang. A simple iterative method to optimize protein-ligand-binding residue prediction. *Journal of theoretical biology*, 317:21923, Jan 2013.
33. Z. Qiu and X. Wang. Improved prediction of protein ligand-binding sites using random forests. *Protein and Peptide Letters*, 18(12):1212–1218, 2011-12-01T00:00:00.
34. D. Rognan. *Docking Methods for Virtual Screening: Principles and Recent Advances*, pages 153–176. Wiley-VCH Verlag GmbH & Co. KGaA, 2011.
35. P. Schmidtke, C. Souaille, F. Estienne, N. Baurin, and R. Kroemer. Large-scale comparison of four binding site detection algorithms. *Journal of chemical information and modeling*, 50(12):2191200, 2010.
36. S. Schneider and M. Zacharias. Combining geometric pocket detection and desolvation properties to detect putative ligand binding sites on proteins. *Journal of structural biology*, 180(3):546–50, Dec 2012.
37. K. Schomburg, S. Bietz, H. Briem, A. Henzler, S. Urbaczek, and M. Rarey. Facing the challenges of structure-based target prediction by inverse virtual screening. *Journal of chemical information and modeling*, 54(6):167686, 2014.
38. J. Skolnick and M. Brylinski. FINDSITE: a combined evolution/structure-based approach to protein function prediction. *Briefings in bioinformatics*, 10(4):378–391, 2009.
39. C. Steinbeck, Y. Han, S. Kuhn, O. Horlacher, E. Luttmann, and E. Willighagen. The chemistry development kit (cdk): An open-source java library for chemo- and bioinformatics. *Journal of Chemical Information and Computer Sciences*, 43(2):493–500, 2003. PMID: 12653513.
40. V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston. Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43(6):1947–58, Jan 2003.
41. M. Weisel, E. Proschak, and G. Schneider. Pocketpicker: analysis of ligand binding-sites with shape descriptors. *Chemistry Central Journal*, 1(1):7, 2007.

42. L. Xie, L. Xie, and P. E. Bourne. Structure-based systems biology for analyzing off-target binding. *Current opinion in structural biology*, 21(2):189–99, Apr 2011.
43. Z. Zhang, Y. Li, B. Lin, M. Schroeder, and B. Huang. Identification of cavities on protein surface using multiple computational approaches for drug binding site prediction. *Bioinformatics (Oxford, England)*, 27(15):2083–2088, 2011.
44. X. Zheng, L. Gan, E. Wang, and J. Wang. Pocket-based drug design: Exploring pocket space. *The AAPS Journal*, 2012.

P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure

Reference

KRIVÁK R., HOKSZA D.: **P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure**. *Journal of cheminformatics 10*, 1 (2018), 39. doi:10.1186/s13321-018-0285-8

Author's highlights

An expanded version of the previous conference contribution that introduced P2Rank as a freely available open-source tool. P2Rank was extensively tested against 5 other state-of-the-art methods. Notable is a longer introduction discussing the state of the field of LBS prediction methods.

The supplementary material contains a discussion about which ligands are considered biologically relevant. Furthermore, there is a detailed pairwise comparison with every other method from our evaluation.

Krivák and Hoksza *J Cheminform* (2018) 10:39
https://doi.org/10.1186/s13321-018-0285-8

Journal of Cheminformatics

SOFTWARE

Open Access

P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure

Radoslav Krivák* and David Hoksza*

Abstract

Background: Ligand binding site prediction from protein structure has many applications related to elucidation of protein function and structure based drug discovery. It often represents only one step of many in complex computational drug design efforts. Although many methods have been published to date, only few of them are suitable for use in automated pipelines or for processing large datasets. These use cases require stability and speed, which disqualifies many of the recently introduced tools that are either template based or available only as web servers.

Results: We present P2Rank, a stand-alone template-free tool for prediction of ligand binding sites based on machine learning. It is based on prediction of ligandability of local chemical neighbourhoods that are centered on points placed on the solvent accessible surface of a protein. We show that P2Rank outperforms several existing tools, which include two widely used stand-alone tools (Fpocket, SiteHound), a comprehensive consensus based tool (MetaPocket 2.0), and a recent deep learning based method (DeepSite). P2Rank belongs to the fastest available tools (requires under 1 s for prediction on one protein), with additional advantage of multi-threaded implementation.

Conclusions: P2Rank is a new open source software package for ligand binding site prediction from protein structure. It is available as a user-friendly stand-alone command line program and a Java library. P2Rank has a lightweight installation and does not depend on other bioinformatics tools or large structural or sequence databases. Thanks to its speed and ability to make fully automated predictions, it is particularly well suited for processing large datasets or as a component of scalable structural bioinformatics pipelines.

Keywords: Ligand binding sites, Protein pockets, Binding site prediction, Protein surface descriptors, Machine learning, Random forests

Background

Motivation

Prediction of ligand binding sites (LBS, or simply pockets[1]) from protein structure has many applications in elucidation of protein function [1] and rational drug design [2–4]. It has been employed in drug side-effects prediction [5], fragment-based drug discovery [6], docking prioritization [7, 8], structure based virtual screening [9] and structure-based target prediction (or so called inverse virtual screening) [10]. Increasingly, LBS prediction is being used in large-scale structural studies that try to analyze and compare all known and putative binding sites on a genome-wide level [11–15]. In practice, it is often the case that predicting ligand binding sites is not an end in itself but it represents only a step in larger automated solution or pipeline. For example, druggability prediction server PockDrug-Server [16] relies on LBS prediction internally. Similarly, allosteric site prediction tools Allosite [17] and AlloPred [17] both employ pocket prediction tool Fpocket [18] as the first step of their algorithms.

*Correspondence: krivak@ksi.mff.cuni.cz; hoksza@ksi.mff.cuni.cz
Department of Software Engineering, Charles University, Prague, Czech Republic

[1] We use the term 'pocket' liberally as a convenient one word synonym for 'ligand binding site', although not all ligand binding sites are necessarily located in concave pockets.

In the rest of this section we will summarize existing methods and available tools. We will introduce categorization along several lines:

(1) web servers/stand-alone tools,
(2) template based/template-free methods,
(3) residue-centric/pocket-centric prediction,

and we will discuss strengths and weaknesses of tools in these categories. We will also discuss an overlooked aspect of the speed of available tools. We will try to convey that there is a strong case for new fast stand-alone user-friendly tool that is not based on search in a large template library of known protein-ligand complexes.

Existing approaches

Existing methods for LBS prediction are based on variety of algorithmic approaches. Traditionally, methods have been categorized based on their main algorithmic strategy into geometric, energetic, conservation based, template based (the last two also sometimes referred to as evolutionary) and machine learning/knowledge based. In reality, many of the state-of-the-art tools are based on some combination of the mentioned approaches. Methods based on consensus of results of other algorithms have also emerged. Table 1 lists available tools for LBS prediction from protein structure introduced since 2009 (to cover most recent and still widely used methods). In the following paragraphs we will introduce the tools that we have used to comparatively evaluate the performance of P2Rank. More details on existing approaches, including older ones, can be found in numerous reviews and surveys [3, 7, 19–25].

Fpocket is a fast geometric stand-alone tool based on filtering and clustering of alpha spheres found by way of Voronoi tessellation [18]. It has been one of the most widely used methods in recent years, especially in large scale applications. Fpocket typically produces relatively high number of predicted pockets for one protein. Among them, Fpocket finds most of the known binding sites, but they are not always ranked at the top. To address this problem, we have previously developed a method called PRANK [26] that is able to re-score binding site predicted by Fpocket and thus improve relevance of its results (i.e. improve identification success rate among Top-n pockets). Usage simplicity of Fpocket together with its computational efficiency contribute to the fact that it remains a popular choice for LBS prediction, as

Table 1 Availability of existing tools for ligand binding site prediction from protein structure introduced since 2009

Name	Year	Type	Web server	Stand-alone	Fully automated†	Source Code
SiteMap [35]	2009	Geometric	–	Yes	Yes	–
Fpocket [18]	2009	Geometric	Yes	Yes	Yes	Yes
SiteHound [28]	2009	Energetic	Yes	Yes	Yes	Yes
ConCavity [36]	2009	Conservation	Yes	Yes	–	Yes
3DLigandSite [37]	2010	Template	Yes	–	–	–
POCASA [38]	2010	Geometric	Yes	–	–	–
DoGSite [39]	2010	Geometric	Yes	–	–	–
MetaPocket 2.0 [27]	2011	consensus	Yes	–	–	–
MSPocket [81]	2011	Geometric	–	Yes	Yes	Yes
FTSite [40]	2012	Energetic	Yes	–	–	–
LISE [41]	2012	Knowledge/conservation	Yes	Yes	–	–
COFACTOR [42]	2012	Template	Yes	Yes	Yes	–
COACH [43]	2013	Template††	Yes	Yes	Yes	–
G-LoSA [44]	2013	Template	–	Yes	–	Yes
eFindSite [45]	2013	Template	Yes	Yes	–	Yes
GalaxySite [46]	2014	Template/docking	Yes	–	–	–
LIBRA [47]	2015	Template	Yes	Yes	–	–
P2Rank (this work)	2015*	Machine learning	–**	Yes	Yes	Yes
bSiteFinder [48]	2016	Template	Yes	–	–	–
ISMBLab-LIG [32]	2016	Machine learning	Yes	–	–	–
DeepSite [33]	2017	Machine learning	Yes	–	–	–

† Applies to stand-alone versions

†† Consensus of template based methods: TM-SITE, S-SITE and COFACTOR (also FINDSITE and ConCavity in web version)

*Algorithm introduced in conference proceedings [49]

**In development

can be illustrated by its employment in recent large-scale structural studies [11–15]. Overall good user experience with Fpocket in contrast with other available methods has been an inspiration for designing our tool.

MetaPocket 2.0 is a prominent example of a consensus based method [27]. It aggregates results produced by 8 different previously published algorithms by taking top 3 sites predicted by each method. It was shown to perform better that any single one of those individual methods. MetaPocket 2.0 is only available as a web server.

SiteHound is one of the latest energetic methods, and the latest one with stand-alone version [28]. It works by placing a probe on a grid points around a protein surface and calculating interaction energies with the help of underlying force field software. It is available as a web server and as a fully automated stand-alone tool.

Fpocket, SiteHound and MetaPocket 2.0 belong to the most cited and widely used template-free methods introduced in the last decade.

The tool presented in this article is based on machine learning from examples. As a main approach, machine learning has been under-utilized among published methods. Although some studies that applied machine learning to the problem have been published, their focus was mainly on classification of binding residues rather than on predicting binding sites as such [29–31]. Machine learning has been also employed to solve partial tasks in complex eFindSite and COACH methods. Tools based primarily on machine learning have been introduced only very recently [32, 33] (with notable earlier exception [34]). The latest one of them is DeepSite, a method based on multi-layer (for different atom types) voxelized representation of 3D space and deep convolutional neural networks. It is available only as a web server, but it is reasonably fast and has usable, although undocumented web API.

Studies that introduced existing methods reported relatively high identification success rates, usually on traditional small datasets. However, the results of the only independent benchmark [21] suggest that existing methods may not be as accurate as previously believed when applied to new datasets. It showed that there is still a need for more accurate methods, and that nominally high results reported by the authors of respective methods may not be always indicative of their true performance on unseen proteins.

Stand-alone tools versus web servers

Relatively many methods for LBS prediction have been published to date, and it may seem that the field is crowded with tools available for researchers. However, after closer survey (see Table 1) we found that only few of the published methods are available as a stand-alone software that can be used locally (in contrast to web-only methods), and most of those that are are unnecessarily complicated to use (i.e. users are required to perform preprocessing tasks that could have been automated by the authors of the software). Even fewer of them are available as open source software.

The recent trend has been to make methods available only as a web server. Contrary to that, we believe that there is still a strong case for stand-alone tools. Online methods with a web interface have many advantages including usage simplicity, visual presentation and the fact that they are ready to be used without installation. They are best suited for use cases when researchers want to manually examine one or a small number of proteins. However, for many other use cases, such as those that involve processing of large datasets, tools need to be used in automated mode. Web-only tools are intended for interactive use and unfortunately, as a rule, do not provide stable and documented APIs. Thus, the only way how to use those tools in automated mode is to write patchy web scraping scripts that upload proteins and parse the result pages, which format is not well defined and can change without notice. This approach is far from ideal since it leads to fragile implementations and potentially irreproducible results. Another consideration when using web-only tools is a lack of control over employed computational resources and consequently over speed, stability and availability. Locally executable tools are therefore more suitable in many use cases such as batch processing of large datasets, or in cases where LBS prediction is needed as a stable part of a larger software solution or pipeline.

We believe that from the user perspective, predicting LBS with a stand-alone tool should be as simple as running a single command. With notable exception of Fpocket (`fpocket -f protein.pdb`), SiteHound and COACH, this is rarely the case. All other methods we examined were not able to produce predictions in fully automated manner, and required a manual multi-step procedure for either generating secondary data or data preprocessing of some sort. For example, methods based on sequence conservation like ConCavity or LigsiteCSC [50] ask user to calculate or download sequence conservation scores for a given protein first. Similarly, some template based methods like eFindSite (and also LISE) require pre-calculated sequence alignments as an input (in addition to other preprocessing steps).

Such requirements pose additional work to users and sometimes put them in front of decisions that they may not be ready to make (e.g. what is the best way to calculate conservation scores or which algorithm/database

should be used to generate alignments). Tools that are not fully automated thus pose unnecessary usability barriers that can hinder their widespread adoption.

Template based versus template-free methods

A substantial effort in the recent decade has been devoted to the development of template based methods, which exploit the general tendency of certain protein families to bind ligands at similar locations [45]. From earlier methods like ProFunc [51] and FINDSITE [52, 53] to the recent, more complex methods, their defining feature is that they all rely on a large databases of know protein-ligand complexes. This template database typically consists of a substantial portion of all protein-ligand complexes in the PDB. The difference between methods is in the sophistication by which they search in their template library and then align and aggregate results to form predictions. This search is usually done in a sequential manner, which accounts for the fact that they are typically much slower than template-free methods.

Template based methods belong to the most successful and practically useful of currently available methods. This is because for any unannotated protein, regardless of the use case, we would probably like to know the answer to the question: Are there any known examples of confirmed binding sites on related proteins? Template based methods can give (to some extent) definitive answer to this question, which can be very informative either way. They are able to produce high confidence predictions (especially when closely related proteins are found) supported by examples from the template library.

However, apart from slow speed, template based methods have a fundamental theoretical limitation. Since they are all based on search in a template library, by definition, they are unable to predict truly novel sites that have no analogues in their template library (more precisely: in the template database there is no related protein that has a known binding site at a similar location). Template-free methods, on the other hand, rely on intrinsic local properties of protein surface patches or 3D chemical neighourhoods. As such, they can at least potentially predict truly novel binding sites. Whether this limitation will become more or less relevant in the future is an open question. On the one hand, the number of experimentally solved structures grows steadily. Consequently, template databases will improve their coverage of the space of all possible binding sites with time. On the other hand, advances in ab initio protein modeling [54], de novo protein design [55, 56], directed in silico protein evolution [57] and the fact that LBS prediction is being applied to MD trajectories [58] will offer ever more opportunities for novel binding sites to occur.

Table 2 Prediction speed

Method	Time[†]
COACH (web server)	15 h (self reported estimate)
eFindSite (web server)	6.9 ± 0 h
COACH (stand-alone)	6.4 ± 2 h
GalaxySite (web server)	2 h (self reported estimate)
3DLigandSite (web server)	1–3 h (self reported estimate)
ISMBLab-LIG (web server)	71 ± 2 min
FTSite (web server)	39 ± 3 min
LISE (web server)	39 ± 0.1 min
MetaPocket 2.0 (web server)	2.8 ± 0.4 min
DeepSite (web server)	38 ± 0.03 s
SiteHound (stand-alone)	12 ± 0.5 s
P2Rank (stand-alone)	6.8 ± 0.2 s (cold start*)
	0.9 s (in larger dataset*)
Fpocket (stand-alone)	0.2 ± 0.01 s

[†] Average time required for LBS prediction on a single protein. Displayed is self reported estimate or a result of our test on a small dataset of 5 proteins á ~ 2500 atoms. Stand-alone tools were tested on a single 3.7 GHz CPU core. For web servers the wall time from submitting a job to receiving the result was measured.

*Difference is due to JVM initialization and model loading cost

Another concern related to template based methods is how to meaningfully compare their performance to template-free methods. It is obvious that the query protein structure (for which we want to predict LBS) should be excluded from the template library during evaluation, otherwise the problem is reduced to a simple search. What, then, about very close homologs? To achieve realistic results, authors of eFindSite suggest [45] using sequence identity threshold $t = 40\%$ (35% in earlier work [52]) and excluding templates with higher sequence identity to the query protein when doing benchmarking predictions. This seems reasonable, albeit any particular choice of threshold t is inevitably arbitrary. For any method other than eFindSite we can find a particular value T for which it will perform roughly the same as eFindSite at $t = T$.

For those reasons we see the two categories of methods as complementary and ideally used in combination where possible; template based methods for their ability to give potentially very high confidence predictions, and template-free methods for the ability to potentially predict truly novel binding sites.

Prediction speed

Discussion about running times of existing methods has been largely missing in published studies and reviews. See Table 2 for our survey of running times of several web based and stand-alone tools. As it turns out, the

differences between times required for prediction by individual methods can be in orders of magnitude.

But is the speed of prediction even relevant? For use cases involving only a few proteins probably not; after all, it is worth to wait for potentially better predictions. There are use cases, however, for which high computational requirements might be prohibitive. Those include genome-wide structural studies and prediction on trajectories from MD simulations. For illustration, predictions for 40,000 proteins by a stand-alone version of COACH method would take roughly 30 years on a single CPU core (whereas here introduced P2Rank would need only under 12 h).

Residue-centric versus pocket-centric perspective

Available tools differ also in the way they represent prediction results. Most of the methods produce a ranked list of pockets, which are usually represented as a pocket center and/or as a set of points in the empty space around the protein surface that characterize the shape of the pocket. These could be regularly spaced grid points (most of the methods), alpha sphere centers (Fpocket) or points on a solvent accessible surface (P2Rank). These *pocket-centric* methods are typically evaluated and compared in terms of the identification success rate considering Top-*k* pockets from the ranked list of predicted binding sites (where *k* is usually 1, 3 or 5).

A subset of published methods is focused primarily on predicting ligand binding residues. Many of those methods do not produce a ranked list of binding sites as such, nor do they pinpoint their locations and shapes. Those *residue-centric* methods look at the problem of LBS prediction as to the problem of binary classification of solvent exposed residues to binding and non-binding. This is also the way how they are evaluated and compared, usually in terms of standard binary classification metrics: MCC, AUC or F-measure. This point of view originated with earlier methods for LBS prediction directly from sequence. It is also prevalent as a main evaluation methodology among methods that compete in CASP [59] and CAMEO [60] competitions, where prediction of ligand binding residues on homology models is one of the disciplines.

This residue-centric view represents not only a different way of looking at the problem, but also a different and in some cases conflicting objective. Methods that are optimized to achieve the best results in binding residue prediction will not necessarily be best at ranked pocket prediction and vice versa. To illustrate where are those objectives misaligned, consider the following case: a method predicts a large binding site centered around a small known ligand, such that predicted pocket defines three times larger protein surface than is the contact surface defined by this known ligand (similar situation can be seen in Fig. 1). How should be this prediction evaluated? From the pocket-centric point of view, it is considered a successful prediction and therefore a net positive. From the residue classification point of view, this adds around twice as much false positives than true positives (2/3 of predicted residues are not contact residues with known ligand) to the confusion matrix, and that will have mostly negative impact toward chosen performance metric. Ligand binding site is a fuzzy concept, even more so is the notion of its exact borders. It is not unreasonable to assume that considered binding site could harbor a larger ligand [61] (perhaps a superstructure of the known small one). It may be objected that this just means that residue-centric view favours more precise predictions. However, by the same token, a residue-centric evaluation methodology will favour spatially precise prediction of one larger binding site over few correct smaller ones.

We believe that pocket-centric point of view better represents a common sense associated with LBS prediction, and as an evaluation methodology awards those methods that fail to predict the least amount of potentially interesting binding sites. In this context, P2Rank is a pocket-centric method.

Other limitations and advantages of available tools

Available tools have other practical and theoretical limitations. For instance COACH web server limits 3 jobs per user (IP address) and ISMBLab-LIG and eFindSite web servers asks for entering captcha-like code with every prediction request. Some methods are able to predict LBS only on single-chain proteins or they work with single-chain structures internally (this is true for most of the template based methods). This could be a usability inconvenience as preprocessing step of splitting structures by chains is needed first. More importantly, it means that those tools will not be able to predict potential binding sites that emerge around places, where chains connect in multimers and biological assemblies.

It should be acknowledged that some tools offer functionality that goes beyond simple LBS prediction from structure. Some tools are able to perform prediction just from sequence by automatically building a homology model first (GalaxySite, 3DLigandSite, FunFold [23]). Another useful function of some methods is the ability to suggest possible binding ligands (GalaxySite and template based methods). Other tools are able to directly predict druggability of predicted pockets (Fpocket, DogSite) or predict transient pockets in molecular simulation trajectories [62, 63]. That being said, in the present work we assess other tools only by their ability to predict LBS from structure.

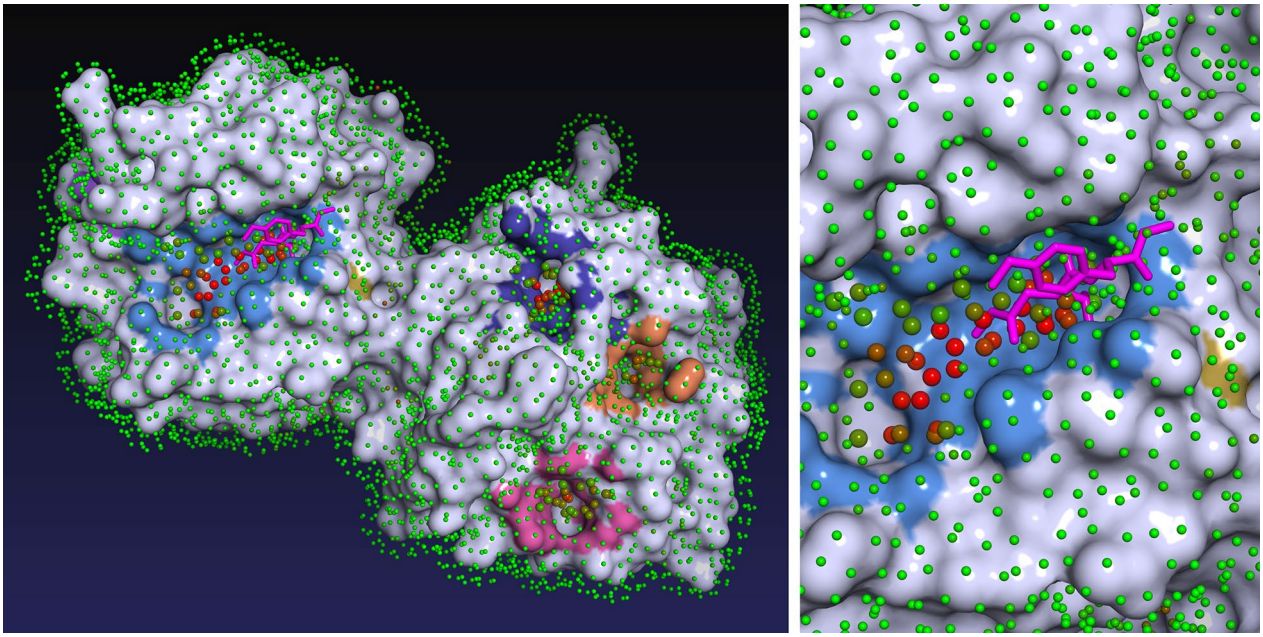


Fig. 1 Visualization of ligand binding sites predicted by P2Rank for structure 1FBL. Protein is covered in a layer of points lying on the Solvent Accessible Surface of the protein. Each point represents its local chemical neighborhood and is colored according to its predicted ligandability score (from 0 = green to 1 = red). Points with high ligandablity score are clustered to form predicted binding sites (marked by coloring adjacent protein surface). In this case, the largest predicted pocket (shown in the close-up) is indeed a correctly predicted true binding site that binds a known ligand (magenta). Visualization is based on a PyMOL script produced by P2Rank

Implementation and usage

P2Rank is a command line program written in Groovy and Java distributed as a binary package that requires no dependencies except Java Runtime Environment. It is lightweight in the sense that (unlike many alternative stand-alone tools) it specifically does not depend on other bioinformatics tools or large structural or sequence databases that would need to be installed on a local machine. It is platform independent (to the extent Java is) and has been tested on Linux and Windows.

Input is a PDB file or a dataset file that contains a list of PDB files. P2Rank is able to automatically produce predictions for any PDB file (single or multi chained) by running a single command (`prank predict -f protein.pdb`). No preprocessing steps on part of the user are needed. For each input protein, P2Rank produces an output CSV file which contains an ordered list of predicted pockets and their scores. Pockets are characterized by coordinates of their centers, by a list of solvent exposed protein atoms and by a list of amino acid residues that constitute the binding site. PDB file with labeled SAS points (which form a primary internal representation of predicted pockets) can be also produced. The program can optionally generate a PyMOL [64] script that produces 3D visualizations such as the one shown in Fig. 1. In addition to that, P2Rank allows to easily train and evaluate new models on custom datasets and then use them for predictions. This approach can be used to create models that are specialized for specific types of proteins or ligands.

P2Rank has an efficient well optimized implementation: required running time averages to less than 1 s for a protein of ~2500 atoms on a single 3.7 GHz CPU core. On multi-core machines datasets can be processed in parallel with a configurable number of working threads. Memory footprint is around 1GB but grows only slowly with additional working threads. Additionally, P2Rank has a clean internal Java API and apart from being used as a command line tool it can be easily employed as a library for LBS prediction by programs running on JVM.

Results and discussion

Results

We have extensively evaluated prediction performance of P2Rank and compared it against several widely used and state-of-the art methods. Those include geometric Fpocket, energetic SiteHound, consensus based MetaPocket 2.0 and deep learning based DeepSite. In the comparison we focused mainly on tools that P2Rank directly competes with: that is template-free stand-alone fully automated tools that are freely available.

Table 3 Comparison of predictive performance on COACH420 and HOLO4K datasets

	COACH420		HOLO4K	
	Top-n	**Top-(n+2)**	**Top-n**	**Top-(n+2)**
Fpocket	56.4	68.9	52.4	63.1
Fpocket+PRANK[a]	63.6	76.5	62.0	71.0
SiteHound[†]	53.0	69.3	50.1	62.1
MetaPocket 2.0[†]	63.4	74.6	57.9	68.6
DeepSite[†]	56.4	63.4	45.6	48.2
P2Rank[protrusion][b]	64.2	73.0	59.3	67.7
P2Rank	*72.0*	*78.3*	*68.6*	*74.0*

The numbers represent identification success rate [%] measured by DCCcriterion (distance from pocket center to closest ligand atom) with 4 Å threshold considering only pockets ranked at the top of the list (n is the number of ligands in considered structure)

[†] These methods failed to produce predictions for some portion of input proteins. Here we display success rates calculated only based on subsets of proteins, on which they finished successfully. Detailed, pairwise comparison with P2Rank on the exact subsets can be found in the Additional file 1.

[a] Predictions of Fpocket re-scored by PRANK algorithm (which is included in P2Rank software package)

[b] Reduced version of P2Rank that uses only single geometric feature: protrusion

It should be noted that our prediction model was trained on CHEN11 dataset and some arbitrary parameters of the algorithm were tweaked with respect to the performance on JOINED dataset (see "Datasets" section). We want to emphasize that only results on CHEN420 and HOLO4K datasets represent an unbiased estimate of P2Rank's performance.

Results in Table 3 show that P2Rank clearly outperforms other tools in Top-n and Top-(n+2) categories on both datasets. P2Rank also achieves higher success rates that were possible to achieve just by re-scoring predictions of Fpocket by PRANK algorithm (PRANK is part of P2Rank software package and works on similar principles). Still, Fpocket+PRANK performed better than any of the other tools with the exception of P2Rank.

We have also evaluated performance of a reduced version of P2Rank that uses only single geometric feature (descriptor): protrusion. Surprisingly, even this simplified, purely geometric version of P2Rank slightly outperforms other tools in most cases (with the exception of MetaPocket 2.0 in Top-(n+2) category).

Some of the evaluated tools failed to produce predictions on some portion of inputs. Since we wanted to compare the viability of the methods, not just robustness of their implementations, we considered success rates only on subsets of original datasets on which given tools finished successfully and produced predictions. Detailed, pairwise breakdown of the results is included in Additional file 1.

Table 4 Average number of predicted binding sites

	COACH420	HOLO4K
avg. protein atoms	2179	3908
avg. true sites	1.2	2.4
Fpocket	14.6	27
SiteHound	66.2	99.5
MetaPocket 2.0	6.3	6.4
DeepSite	3.2	2.8
P2Rank	6.3	12.6

Displayed is the average total number of binding sites predicted per protein by each method on a given dataset

Furthermore, we have compared prediction speed with aforementioned and several additional tools. Results in Table 2 show that P2Rank is faster than other tools with the exception of Fpocket.

Differences in average total number of predicted sites are shown in Table 4. The table also shows that HOLO4K dataset contains larger proteins with more binding sites than COACH420. This is due to the fact that HOLO4K contains mainly multimers and COACH420 only single-chain proteins. Interestingly, Fpocket and P2Rank seem to scale the number of predicted sites with protein size, while MetaPocket 2.0 and DeepSite do not. SiteHound produced significantly more small pockets that other tools.

Discussion

DeepSite is the only other machine learning based method in our benchmark and we shall discuss how it relates to our method and offer possible explanation for its lower performance. Predictive model of DeepSite is deep convolutional neural network trained on a large dataset of 7622 structures derived from sc-PDB [65] database. DeepSite is based on learning from relatively large instance representations (i.e. model input; 8×16^3 sliding box) and a large dataset, whereas P2Rank is based on smaller representations (1D feature vector) and smaller training dataset. Voxelized representation used by DeepSite, in related works also referred to as *atomic grid* [66, 67], is closer to the raw structural data (atomic coordinates and types) and as such it holds more information. It potentially allows trained model to capture more interactions than our feature based representation. In the light of our results, however, we suspect that even larger training datasets may be needed for such voxelized representations to perform well. Another possible reason for relatively poor performance of DeepSite in our benchmark may be that our respective training sets come form different distributions, more specifically the fact that the relevant ligands (and therefore binding sites) are defined

differently. More work is needed to compare respective approaches, ideally using the same training and test datasets and evaluation methodology. This discussion only highlights prevalent and recognized [21, 26, 68] problem of the field: the lack of standardized protocols and benchmarks.

Another general problem in the field is the over-reliance on the ground truth as defined by known protein-ligand complexes from PDB. It is naive to assume that in our datasets all possible binding sites are demarked by bound ligands. That is to say that many locations labeled as negatives (non-binding sites) in the datasets may be binding sites yet to be discovered, or they are already known, but the particular ligand binding is captured in a different PDB entry. Due to protein flexibility and allosteric effects, in some cases it may not even be possible for a protein to bind two ligands at two known binging sites at the same time. We conjecture that between 1/3 and 1/2 of true ligand binding sites are not demarked by ligands in structures directly taken from the PDB. This is particularly problematic for machine learning and knowledge based methods which use such datasets for training their models or constructing their knowledge bases. From their perspective it means that training datasets are extremely noisy.

There is no perfect solution, but the best effort to mitigate this issue we have encountered is expressed in the way CHEN11 dataset was constructed. For all proteins in this dataset, close homologs were found in the PDB, aligned with them and ligands from homologs were superimposed to those structures. Consequently, it is less likely that CHEN11 dataset contains unmarked true binding sites (although some risk that some of those additional binding sites are false is introduced). We believe that this dataset serves as a better source for the ground truth than raw structures taken directly from PDB (therefore we use it as a training set despite its relatively small size). The way this dataset was constructed is akin to the working of template based methods, and we believe that, in a similar way, template based methods can help to construct better training datasets in the future (by adding very high confidence predictions based on close homologs as binding sites).

Furthermore, when such noisy datasets are used for evaluation (of all, not just machine learning based methods), there is a theoretical performance limit that can be achieved even by an optimal predictor (i.e. predictor that achieves Bayes optimal rate). Even optimal predictor would sometimes predict (on top of the ranked list) fundamentally true binding site that is not correctly labeled in the evaluation dataset, with the effect that a 100% success rate would not be achieved on this protein and consequently on the dataset. For this reason we are suspicious when we see reported success rates that are unrealistically high, say close to or above 95% in Top-1/Top-n category (which seem to be above optimal achievable rate on noisy datasets). This can be indicative of a data leakage (in machine learning and knowledge based methods) or overfitting on a given dataset (i.e. dataset was used to optimize parameters during development) or, in case of template based methods, of the fact that the query protein was not removed from the template library during evaluation (as we have seen in some recent papers). We believe that if some method seem to achieve such high success rates, especially on small datasets, it may not be indicative of its true performance and researchers should check for mentioned pitfalls and try to evaluate it on larger datasets. More research is, however, needed to support our conjecture and to provide better estimates.

In the introduction, we have argued that template based methods are not able to predict truly novel sites (with respect to their template library), implying that our method should be better in this regard. A question that can be raised here is, that since our method is based on machine learning from examples, whether that means that it is also only as good as is the training set, and therefore subject to similar limitations as template based methods. The answer is yes, to some extent this is true for any machine learning based method. However, the premise of our method is that the model is not learning to remember particular binding sites, but rather learns what makes local neighbourhoods around the protein surface intrinsically ligandable. Algorithm should then be able to apply this learned generalized knowledge to predict novel sites. But this is exactly what can be illustrated by the performance of our method on a large dataset like HOLO4K.

The unique feature of our method is that we predict ligandability of points on a solvent accessible surface. Other related machine learning approaches were focused on predicting ligandability of residues, solvent exposed atoms or points on a regular grid. In our preliminary experiments, focusing on grid points or atoms led to significantly worse results. We mention it as this insight might be helpful for authors of related methods in the future.

Future work

One limitation of our tool is that it does not produce exact shapes and volumes of predicted binding sites. For each predicted pocket, P2Rank can produce a set of its SAS points that somewhat define its shape, but they are not regularly spaced in 3D. This is something we would like to address in the future versions of the software, and improve it to produce volumetrically exactly defined,

geometrically feasible binding sites. As a consequence, in our evaluation we did not use volumetric overlap identification criteria sometimes employed in other studies [18, 33]. It is possible that other methods produce predictions with more accurate shapes (where those binding sites are found in the first place). However, given the large margin with which P2Rank outperformed other compared methods, it is very unlikely that the conclusions of the benchmark would be different using volumetric criteria.

Currently, P2Rank still does not use all available information that is possible to derive from protein structure. Sequence conservation and energetic calculations (using different probes) could be used to further enrich the feature vector. Our present research is also focused on applying rotation invariant geometric 3D descriptors as well as more powerful machine learning methods to the problem.

Materials and methods

P2Rank algorithm

The P2Rank algorithm (which principles we introduced previously in [49]) is based on classification of points evenly spread on protein's Solvent Accessible Surface (referred to as *SAS points*). These points represent local spherical 3D neighbourhoods that are centered on them. At the same time, they can be seen as potential locations of contact atoms of potential ligands. Initially, SAS points are described by a vector of physico-chemical, geometric and statistical features calculated from its local geometric neighbourhood. Consecutively, a predicted ligandability score is assigned to each SAS point by a machine learning based model. Finally, the points with high predicted ligandability score are clustered to form predicted ligand binding sites (see Fig. 1).

To generate predictions for a given protein using a pre-trained classification model P2Rank follows these instructions:

1. Generate a set of regularly spaced points lying on a protein's Solvent Accessible Surface (*SAS points*). Positions of the points are calculated by a fast numerical algorithm [69] implemented in CDK library [70].
2. Calculate feature descriptors of SAS points based on their local chemical neighborhood:
   (a) compute property vectors for protein's solvent exposed atoms,
   (b) project distance weighted properties of nearby protein atoms onto SAS points (6Å neighbourhood is considered, $w(d) = 1 - d/6$),
   (c) compute additional features describing SAS points' neighborhoods and assign them directly to SAS points.
3. Predict ligandability score of SAS points by Random Forest classifier.
4. Cluster points with high ligandability score and thus form pocket predictions (single-linkage clustering with 3Å cut-off).
5. Rank predicted pockets by cumulative ligandability score of their points (sum of squared ligandability scores of all points in the cluster).

Initial step of our approach relates our method to the energetic method by Morita et al. [71], where points on a solvent accessible surface were used to discretize space around the protein (in contrast with a typical approach of using points on a regular grid).

Feature vector that represents SAS points and their neighbourhoods contains 35 numerical features, some of which were inspired by other studies [72–76]. For the complete list of features and analysis of their importance, see Additional file 1. The single most important feature turned out to be a geometric feature termed protrusion. It is defined simply as a number of protein atoms within a sphere of 10 Å around a SAS point, and as such can be seen as a proxy for point's "buriedness". In the "Results" section we show that even a simplified version of the algorithm, based only on this feature alone, seem to outperform many of the other methods.

P2Rank is distributed with a pre-trained model based on Random Forests algorithm that was trained on a relatively small but diverse CHEN11 dataset (see "Datasets" section). Various arbitrary parameters of the algorithm (cut-offs, thresholds, protrusion radius, etc.) and hyperparameters of Random Forest were optimized with respect to the performance on JOINED dataset. The final default model has 200 trees, each grown with no depth limit using 6 features.

Datasets

To train and evaluate P2Rank we were working with following datasets of protein-ligand complexes:

- **CHEN11**—a dataset of 251 proteins harboring 476 ligands introduced in LBS prediction benchmarking study [21]. A non-redundant dataset designed in a way so that every SCOP family [77] has at most one typical representative and to minimize the number of unannotated binding sites (by superimposing ligands from very close homologs). As such it serves as a good source for the ground truth and we employ it as a training set. See [21] for the details on how it was constructed.
- **JOINED**—consists of structures from several smaller datasets used in previous studies (B48/U48, B210,

DT198, ASTEX) joined into one larger dataset. We use it as a development set (i.e. validation set).

  - **B48/U48**—Datasets that contain a set of 48 proteins in a bound and unbound state [50].
  - **B210**—a benchmarking dataset of 210 proteins in bound state [50].
  - **DT198**—a dataset of 198 drug-target complexes [27].
  - **ASTEX**—Astex Diverse set [78] is a collection of 85 proteins that was introduced as a benchmarking dataset for molecular docking methods.
- **COACH420**—consists of 420 single chain structures that contain a mix of drug targets and naturally occurring ligands (we have taken COACH test set [42, 43] and removed proteins contained in CHEN11 and JOINED).
- **HOLO4K**—large dataset of protein-ligand complexes based on the list published in [79]. Contains larger multi-chain structures downloaded directly from PDB. Disjunct with CHEN11 and JOINED.

Evaluation methodology

To evaluate predictive performance of P2Rank and compare it with other methods we have used methodology based on ligand-centric counting and DCC(distance between the center of the pocket and any ligand atom) pocket identification criterion with 4 Å threshold. Binding sites are defined by ligands present in evaluation datasets. Every structure in a dataset can have more than one relevant ligand (see below) and for every relevant ligand, its binding site must be correctly predicted for a method to achieve 100% identification success rate on the given dataset. Every relevant ligand contributes with equal weight toward the final success rate. The output of prediction methods is a ranked list of several putative binding sites, but during evaluation only those ranked at the top are considered. We use Top-n and Top-(n+2) rank cutoffs where n is the number of relevant ligands in the evaluated target protein structure (for proteins with only one ligand this corresponds to the usual Top-1 and Top-3 cutoffs). This evaluation methodology is the same as the one that was used in independent benchmarking study [21]. P2Rank is focused on predicting binding sites for biologically relevant ligands and PDB files in considered datasets often contain ligands (or HET groups) that are not relevant. To determine which ligands are relevant we use a custom filter and alternatively the binding MOAD [80] database. For more details on how we determine which ligands are relevant, see Additional file 1.

Conclusion

We have presented P2Rank, a novel machine learning based tool for prediction of ligand binding sites from protein structure. We have shown that P2Rank outperforms several alternative tools on two large datasets and that it belongs to the fastest available tools. P2Rank is able to work directly with multi-chain structures and thus find potential binding sites that consist of residues from multiple chains. Among other advantages is the fact that P2Rank works out of the box, as it does not depend on other bioinformatics tools or databases. Unlike many alternative stand-alone tools, P2Rank is able to make fully automated predictions from the command line (no manual preprocessing steps are needed).

P2Rank is, therefore, well suited to be used as a stable component in structural bioinformatics pipelines, where fast and accurate prediction is required. We believe that P2Rank should be particularly beneficial for predicting novel allosteric sites, for which template based methods would generally be less effective. P2Rank is available as an open source command line tool and a Java library.

Availability and requirements

- Project name: P2Rank
- Project home page: http://siret.ms.mff.cuni.cz/p2rank
- Operating system(s): Platform independent
- Programming language: Groovy, Java
- Other requirements: JRE 8 or higher (Java 1.8)
- Source code: http://github.com/rdk/p2rank
- License: MIT

Additional file

Additional file 1.

Abbreviations

LBS: ligand binding site(s); MD: molecular dynamics; PDB: the Protein Data Bank; SAS: solvent accessible surface; MCC: Matthews correlation coefficient; AUC: area under the ROC curve (receiver operating characteristic); JVM: Java virtual machine.

Authors' contributions

DH suggested pocket representation based on aggregated local features. RK introduced a machine learning approach, designed and implemented the algorithm and preformed the experiments. DH supervised the project. Both authors contributed to the writing of the paper. Both authors read and approved the final manuscript.

Acknowledgements

We would like to thank Michal Brylinski (author of eFindSite) for helping us to clarify our view on template based methods.

Competing interests
The authors declare that they have no competing interests.

Availability of data and materials
Datasets used in our benchmarks (together with predictions produced by all evaluated methods) are available at http://github.com/rdk/p2rank-datasets.

Funding
This work was supported by the project SVV 260451 and by the Grant Agency of Charles University (project Nr. 1556217).

Publisher's Note

Received: 12 November 2017 Accepted: 29 June 2018
Published online: 14 August 2018

References

1. Konc J, Janežič D (2014) Binding site comparison for function prediction and pharmaceutical discovery. Curr Opin Struct Biol 25:34–9
2. Zheng X, Gan L, Wang E, Wang J (2013) Pocket-based drug design: exploring pocket space. AAPS J 15:228–241
3. Pérot S, Sperandio O, Miteva M, Camproux A, Villoutreix B (2010) Druggable pockets and binding site centric chemical space: a paradigm shift in drug discovery. Drug Discov Today 15(15–16):656–667
4. Tibaut T, Borišek J, Novič M, Turk D (2016) Comparison of in silico tools for binding site prediction applied for structure-based design of autolysin inhibitors. SAR QSAR Environ Res 27(7):573–587 (PMID: 27686112)
5. Xie L, Xie L, Bourne PE (2011) Structure-based systems biology for analyzing off-target binding. Curr Opin Struct Biol 21(2):189–99
6. Grove Laurie E, Sandor Vajda DK (2016) Computational methods to support fragment-based drug discovery. In: Fagerberg J, Mowery DC, Nelson RR (eds) Fragment-based drug discovery: lessons and outlook. Wiley, Weinheim, pp 197–222 (Chap. 9)
7. Laurie A, Jackson R (2006) Methods for the prediction of protein-ligand binding sites for structure-based drug design and virtual ligand screening. Curr Protein Peptide Sci 7(5):395–406
8. Feinstein WP, Brylinski M (2015) Calculating an optimal box size for ligand docking and virtual screening against experimental and predicted binding pockets. J Cheminform 7(1):1–10
9. Lionta E, Spyrou G, Cournia DKV (2014) Zoe: structure-based virtual screening for drug discovery: principles, applications and recent advances. Curr Top Med Chem 14(16):1923–1938
10. Schomburg K, Bietz S, Briem H, Henzler A, Urbaczek S, Rarey M (2014) Facing the challenges of structure-based target prediction by inverse virtual screening. J Chem Inf Model 54(6):1676–86
11. Degac J, Winter U, Helms V (2015) Graph-based clustering of predicted ligand-binding pockets on protein surfaces. J Chem Inf Model 55(9):1944–1952 (PMID: 26325445)
12. Meyers J, Brown N, Blagg J (2016) Mapping the 3D structures of small molecule binding sites. J Cheminform 8(1):70
13. Monzon AM, Zea DJ, Fornasari MS, Saldaño TE, Fernandez-Alberti S, Tosatto SCE, Parisi G (2017) Conformational diversity analysis reveals three functional mechanisms in proteins. PLOS Comput Biol 13(2):1–18
14. Shen Q, Cheng F, Song H, Lu W, Zhao J, An X, Liu M, Chen G, Zhao Z, Zhang J (2017) Proteome-scale investigation of protein allosteric regulation perturbed by somatic mutations in 7000 cancer genomes. Am J Hum Genet 100(1):5–20
15. Bhagavat R, Sankar S, Srinivasan N, Chandra N (2018) An augmented pocketome: detection and analysis of small-molecule binding pockets in proteins of known 3D structure. Structure 26(3):499–5122
16. Hussein H, Borrel A, Geneix C, Petitjean M, Regad L, Camproux A (2015) PockDrug-Server: a new web server for predicting pocket druggability on holo and apo proteins. Nucleic Acids Res 43(W1):436–442
17. Huang W, Lu S, Huang Z, Liu X, Mou L, Luo Y, Zhao Y, Liu Y, Chen Z, Hou T, Zhang J (2013) Allosite: a method for predicting allosteric sites. Bioinformatics 29(18):2357–2359
18. Le Guilloux V, Schmidtke P, Tuffery P (2009) Fpocket: an open source platform for ligand pocket detection. BMC Bioinform 10(1):168
19. Henrich S, Outi S, Huang B, Rippmann F, Cruciani G, Wade R (2010) Computational approaches to identifying and characterizing protein binding sites for ligand design. J Mol Recognit JMR 23(2):209–219
20. Leis S, Schneider S, Zacharias M (2010) In silico prediction of binding sites on proteins. Curr Med Chem 17(15):1550–1562
21. Chen K, Mizianty M, Gao J, Kurgan L (2011) A critical comparative assessment of predictions of protein-binding sites for biologically relevant organic compounds. Structure (London, England : 1993) 19(5):613–621
22. Fauman EB, Rai BK, Huang ES (2011) Structure-based druggability assessment-identifying suitable targets for small molecule therapeutics. Curr Opin Chem Biol 15(4):463–468 (Next Generation Therapeutics)
23. Roche DB, Brackenridge DA, McGuffin LJ (2015) Proteins and their interacting partners: an introduction to protein-ligand binding site prediction methods. Int J Mol Sci 16(12):29829–29842
24. Broomhead NK, Soliman ME (2017) Can we rely on computational predictions to correctly identify ligand binding sites on novel protein drug targets? Assessment of binding site prediction methods and a protocol for validation of predicted binding sites. Cell Biochem Biophys 75(1):15–23
25. Simões T, Lopes D, Dias S, Fernandes F, Pereira J, Jorge J, Bajaj C, Gomes A (2017) Geometric detection algorithms for cavities on protein surfaces in molecular graphics: a survey. In: Computer graphics forum
26. Krivak R, Hoksza D (2015) Improving protein-ligand binding site prediction accuracy by classification of inner pocket points using local features. J Cheminform 7(1):12
27. Zhang Z, Li Y, Lin B, Schroeder M, Huang B (2011) Identification of cavities on protein surface using multiple computational approaches for drug binding site prediction. Bioinformatics (Oxford, England) 27(15):2083–2088
28. Ghersi D, Sanchez R (2009) EasyMIFS and SiteHound: a toolkit for the identification of ligand-binding sites in protein structures. Bioinformatics (Oxford, England) 25(23):3185–3186
29. Kauffman C, Karypis G (2009) Librus: combined machine learning and homology information for sequence-based ligand-binding residue prediction. Bioinformatics (Oxford, England) 25(23):3099–107
30. Qiu Z, Wang X (2011) Improved prediction of protein ligand-binding sites using random forests. Protein Peptide Lett 18(12):1212–1218
31. Chen P, Huang JZ, Gao X (2014) Ligandrfs: random forest ensemble to identify ligand-binding residues from sequence information alone. BMC Bioinform 15(Suppl 15):4
32. Jian JW, Elumalai P, Pitti T, Wu CY, Tsai KC, Chang JY, Peng HP, Yang AS (2016) Predicting ligand binding sites on protein surfaces by 3-Dimensional probability density distributions of interacting atoms. PLoS ONE 11(8):0160315
33. Jiménez J, Doerr S, Martínez-Rosell G, Rose AS, De Fabritiis G (2017) Deepsite: protein-binding site predictor using 3D-convolutional neural networks. Bioinformatics 33(19):3036–3042
34. Nayal M, Honig B (2006) On the nature of cavities on protein surfaces: application to the identification of drug-binding sites. Proteins 63(4):892–906
35. Halgren TA (2009) Identifying and characterizing binding sites and assessing druggability. J Chem Inf Model 49(2):377–389 (PMID: 19154148)
36. Capra JA, Laskowski RA, Thornton JM, Singh M, Funkhouser TA (2009) Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3D structure. PLoS Comput Biol 5(12):1000585
37. Wass MN, Kelley LA, Sternberg MJ (2017) 3DLigandSite: predicting ligand-binding sites using similar structures. Nucleic Acids Res 38(Web Server issue):469–73
38. Yu J, Zhou Y, Tanaka I, Yao M (2010) Roll: a new algorithm for the detection of protein pockets and cavities with a rolling probe sphere. Bioinformatics 26(1):46–52
39. Volkamer A, Griewel A, Grombacher T, Rarey M (2010) Analyzing the topology of active sites: on the prediction of pockets and subpockets. J Chem Inf Model 50(11):2041–52
40. Ngan CH, Hall DR, Zerbe B, Grove LE, Kozakov D, Vajda S (2012) FTSite: high accuracy detection of ligand binding sites on unbound protein structures. Bioinformatics 28(2):286–7
41. Xie Z, Hwang M (2012) Ligand-binding site prediction using ligand-interacting and binding site-enriched protein triangles. Bioinformatics 28(12):1579–1585

42. Roy A, Yang J, Zhang Y (2012) Cofactor: an accurate comparative algorithm for structure-based protein function annotation. Nucleic Acids Res 40(W1):471–477
43. Yang J, Roy A, Zhang Y (2013) Protein-ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment. Bioinformatics 29(20):2588–2595
44. Lee HS, Im W (2013) Ligand binding site detection by local structure alignment and its performance complementarity. J Chem Inf Model 53(9):2462–2470 (PMID: 23957286)
45. Brylinski M, Feinstein WP (2013) eFindSite: improved prediction of ligand binding sites in protein models using meta-threading, machine learning and auxiliary ligands. J Comput Aided Mol Des 27(6):551–567
46. Heo L, Shin W, Lee M, Seok C (2014) GalaxySite: ligand-binding-site prediction by using molecular docking. Nucleic Acids Res 42(W1):210–214
47. Viet Hung L, Caprari S, Bizai M, Toti D, Polticelli F (2015) Libra: ligand binding site recognition application. Bioinformatics 31(24):4020–4022
48. Gao J, Zhang Q, Liu M, Zhu L, Wu D, Cao Z, Zhu R (2016) bSiteFinder, an improved protein-binding sites prediction server based on structural alignment: more accurate and less time-consuming. J Cheminform 8(1):38
49. Krivák R, Hoksza D (2015) In: Dediu A-H, Hernández-Quiroz F, Martín-Vide C, Rosenblueth AD (eds) P2RANK: knowledge-based ligand binding site prediction using aggregated local features. Springer, Cham, pp 41–52
50. Huang B, Schroeder M (2006) Ligsitecsc: predicting ligand binding sites using the connolly surface and degree of conservation. BMC Struct Biol 6(1):19
51. Laskowski RA, Watson JD, Thornton JM (2005) Profunc: a server for predicting protein function from 3D structure. Nucleic Acids Res 33:89–93
52. Brylinski M, Skolnick J (2008) A threading-based method (FINDSITE) for ligand-binding site prediction and functional annotation. Proc Natl Acad Sci USA 105(1):129–134
53. Skolnick J, Brylinski M (2009) FINDSITE: a combined evolution/structure-based approach to protein function prediction. Briefings Bioinform 10(4):378–391
54. Lee J, Freddolino PL, Zhang Y (2017) In: Rigden DJ (ed) Ab initio protein structure prediction. Springer, Dordrecht, pp 3–35
55. Karanicolas J, Corn J et al (2011) A de novo protein binding pair by computational design and directed evolution. Mol Cell 42(2):250–260
56. Damborsky J, Brezovsky J (2014) Computational tools for designing and engineering enzymes. Curr Opin Chem Biol 19(Supplement C):8–16 (Biocatalysis and biotransformation Bioinorganic chemistry)
57. Wang M, Zhao H (2016) In: Stoddard BL (ed) Combined and iterative use of computational design and directed evolution for protein–ligand binding design. Springer, New York, pp 139–153
58. Di Pietro O, Juárez-Jiménez J, Muñoz-Torrero D, Laughton CA, Luque FJ (2017) Unveiling a novel transient druggable pocket in bace-1 through molecular simulations: conformational analysis and binding mode of multisite inhibitors. PLOS ONE 12(5):1–22
59. Gallo Cassarino T, Bordoli L, Schwede T (2014) Assessment of ligand binding site predictions in CASP10. Proteins Struct Funct Bioinform 82:154–163
60. Haas J, Roth S, Arnold K, Kiefer F, Schmidt T, Bordoli L, Schwede T (2013) The protein model portal-a comprehensive resource for protein structure and model information. Database 2013:031
61. Ma B, Shatsky M, Wolfson HJ, Nussinov R (2002) Multiple diverse ligands binding at a single protein site: a matter of pre-existing populations. Protein Sci 11(2):184–197
62. Schmidtke P, Axel B, Luque F, Barril X (2011) MDpocket: open-source cavity detection and characterization on molecular dynamics trajectories. Bioinformatics (Oxford, England) 27(23):3276–3285
63. Stank A, Kokh DB, Horn M, Sizikova E, Neil R, Panecka J, Richter S, Wade RC (2017) Trapp webserver: predicting protein binding site flexibility and detecting transient binding pockets. Nucleic Acids Res 45(W1):325–330
64. Schrödinger LLC (2015) The PyMOL molecular graphics system, version 1.8
65. Desaphy J, Bret G, Rognan D, Kellenberger E (2015) sc-PDB: a 3D-database of ligandable binding sites-10 years on. Nucleic Acids Res 43(D1):399–404
66. Ragoza M, Hochuli J, Idrobo E, Sunseri J, Koes DR (2017) Protein-ligand scoring with convolutional neural networks. J Chem Inf Model 57(4):942–957 (PMID: 28368587)
67. Ragoza M, Turner L, Koes DR (2017) Ligand pose optimization with atomic grid-based convolutional neural networks. ArXiv e-prints
68. Schmidtke P (2011) Protein-ligand binding sites. Identification, characterization and interrelations. Ph.D. thesis, University of Barcelona
69. Eisenhaber F, Lijnzaad P, Argos P, Sander C, Scharf M (1995) The double cubic lattice method: Efficient approaches to numerical integration of surface area and volume and to dot surface contouring of molecular assemblies. J Comput Chem 16(3):273–284
70. Steinbeck C, Han Y, Kuhn S, Horlacher O, Luttmann E, Willighagen E (2003) The chemistry development kit (CDK): An open-source Java library for chemo- and bioinformatics. J Chem Inf Comput Sci 43(2):493–500 (PMID: 12653513)
71. Morita M, Nakamura S, Shimizu K (2008) Highly accurate method for ligand-binding site prediction in unbound state (apo) protein structures. Proteins 73(2):468–79
72. Kyte J, Doolittle RF (1982) A simple method for displaying the hydropathic character of a protein. J Mol Biol 157(1):105–132
73. Desaphy J, Azdimousa K, Kellenberger E, Rognan D (2012) Comparison and druggability prediction of protein-ligand binding sites from pharmacophore-annotated cavity shapes. J Chem Inf Model 52(8):2287–2299
74. Kapcha LH, Rossky PJ (2014) A simple atomic-level hydrophobicity scale reveals protein interfacial structure. J Mol Biol 426(2):484–498
75. Khazanov NA, Carlson HA (2013) Exploring the composition of protein-ligand binding sites on a large scale. PLoS Comput Biol 9(11):1003321
76. Pintar A, Carugo O, Pongor S (2002) Cx, an algorithm that identifies protruding atoms in proteins. Bioinformatics 18(7):980–984
77. Murzin AG, Brenner SE, Hubbard T, Chothia C (1995) Scop: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 247(4):536–540
78. Hartshorn M, Verdonk M, Chessari G, Brewerton S, Mooij W, Mortenson P, Murray C (2007) Diverse, high-quality test set for the validation of protein-ligand docking performance. J Med Chem 50(4):726–741
79. Schmidtke P, Souaille C, Estienne F, Baurin N, Kroemer R (2010) Large-scale comparison of four binding site detection algorithms. J Chem Inf Model 50(12):2191–200
80. Hu L, Benson ML, Smith RD, Lerner MG, Carlson HA (2005) Binding moad (mother of all databases). Proteins Struct Funct Bioinform 60(3):333–340
81. Zhu H, Pisabarro MT (2011) MSPocket: an orientation-independent algorithm for the detection of ligand binding pockets. Bioinformatics 27(3):351–358

P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure

Radoslav Krivák and David Hoksza

1 RELEVANT LIGANDS

P2Rank is focused on predicting binding sites for biologically relevant ligands. PDB files in considered datasets often contain more than one such ligand of interest. PDB files also contain a variety of other HET groups like solvents, salt and misplaced groups (which are not in contact with the protein). Instead of declaring only one ligand as relevant for every file in a dataset (as was often done in other ligand binding site prediction studies), we determine relevant ligands by a filter.
Ligands that are considered relevant must comply to these conditions:

- number of ligand atoms is greater or equal than 5
- distance from any atom of the ligand to the closest protein atom is at least 4Å (to remove "floating" ligands)
- distance form the center of the mass of the ligand to the closest protein atom is not greater than 5.5Å (to remove ligands that "stick out")
- name of the PDB group is not on the list of ignored groups:
`(HOH, DOD, WAT, NAG, MAN, UNK, GLC, ABA, MPD, GOL, SO4, PO4)`

Choosing relevant ligands in exactly this particular way is admittedly arbitrary. In order to make sure our results are robust with respect to the particular way relevant ligands are determined, we have created a versions of JOINED and HOLO4K datasets where relevant ligands are determined in a different way. Binding MOAD [2] release 2013, a database of biologically relevant ligands in PDB, was used to determine relevant ligands in resulting datasets JOINED(Mlig) and HOLO4K(Mlig). PDB files that have no entry in MOAD were removed from the new datasets. It has to be noted that the notion of biologically relevant ligand does not have a widely accepted definition. There are other databases that purportedly collect only biologically relevant ligand interactions from the PDB (e.g. BioLiP [8], PDBbind [7]) that use different criteria for accepting particular ligand as biologically relevant (with MOAD being the strictest of them, not accepting any small ions for example). For the discussion see [8]. We believe that predicting binding sites for ions, peptides and other specific types of binding partners would be better served by specialized methods.

2 ADDITIONAL RESULTS

2.1 Collecting Predictions

P2Rank All reported results correspond to P2Rank v2.0 with default parameters.

Fpocket Stand-alone version of Fpocket v1.0 with default parameters was used (code downloaded from SourceForge repository). Version 2.0RC1 was available at the time but it seemed to be producing consistently worse results.

SiteHound Stand-alone Linux version of SiteHound was downloaded from SiteHound website (version label: January 12, 2010). Command used to generate predictions: `ls *.pdb | xargs -i python ../auto.py -i  -p CMET -k` (executed in directory with pdb files). Default probe and parameters were used.

MetaPocket 2.0 Predictions were obtained from MetaPocket 2.0 web server by web scraping python script in Fall 2017 using default parameters.

DeepSite Predictions were obtained from DeepSite web server by web scraping python script in Fall 2017 using default parameters.

LISE We also made an effort to compare our method with LISE, which is the latest template-free method with a stand-alone version. However, we found that stand-alone version of LISE failed on ∼50% of inputs, mainly due to file

parsing errors. Moreover, on the rest of inputs it exhibited very poor identification success rates (<20%), indicative of some other technical problem. Ultimately, we have decided not to compare results of LISE and P2Rank side by side.

2.2 Detailed Results

Table 1 shows comparison with Fpocket and PRANK, including results on train and validation datasets. Table 2 shows pairwise comparison of P2Rank with SiteHound, MetaPocket 2.0 and DeepSite on exact subsets on which those methods finished successfully and produced predictions.

(Mlig) datasets Tables 1 and 2 also show results on (Mlig) version of the datasets, where relevant ligands were determined in a different way (see Relevant Ligands). Results on (Mlig) datasets tell the same story. In the absolute sense, numbers are higher on HOLO4K(Mlig), which has approx. by 1/3 less relevant binding sites to be predicted than HOLO4K. Nevertheless, P2Rank outperforms other methods with similar margins, especially in Top-n category. Similar margins achieved on those datasets show that our results are robust with respect to the particular way relevant ligands are defined.

Note on DeepSite Presented results of DeepSite on HOLO4K do not represent completely unbiased estimation of its performance. DeepSite is trained on a large dataset which contains some of the proteins that are also included in our test set (733 proteins from HOLO4K), although possibly not on all of the chains.

2.3 Different feature sets

To assess contributions of some features, we have evaluated results of P2Rank with different, reduced, sets of features (Table 3). We would like to note that parameter optimization and final model selection was done with respect to the results on JOINED dataset.

Note on atomic propensity features Atom type propensity features (`apRawValids`,`apRawInvalids`) are based on tables that were calculated from large subset of all protein-ligand complexes from PDB. It is possible that among those complexes were some structures from our test sets. An issue can be raised, that in an absolute sense this may constitute a data leakage; that is to say that there is a possibility that the results reported on those test sets may be biased, as they were achieved with the help of features that were derived also using some structures from those test sets. Practically speaking, contribution of any single protein to numbers in these propensity tables is probably below rounding error. Nevertheless, to avoid possibility of basing our conclusions on biased results, we have evaluated performance of reduced feature set without these propensity features ([full−propensities] in Table 3). Table 3 shows that with respect to the results on COACH420 and HOLO4K, contribution of those features is minimal at best, and on HOLO4K the average success rates without using those features are actually better than results reported in the paper for default P2Rank model. Even if we reported results without using those features, the conclusions of our benchmark and comparison of methods would not change.

Table 1. Comparison with Fpocket and PRANK. Results on CHEN11 (training set) and JOINED (development set) are not representative and are included here only for completeness. In datasets labeled as *Mlig*, relevant ligands (and therefore binding sites that are expected to be predicted) were determined in a different way (see Relevant Ligands).

			Top-n			Top-(n+2)		
Dataset	proteins	ligands	Fpocket	PRANK*	P2Rank	Fpocket	PRANK*	P2Rank
CHEN11	251	476	47.1	**58.2**†	57.9†	57.6	**64.5**†	63.9†
JOINED	537	626	53.8	68.2	**74.4**	72.4	80.0	**80.2**
COACH420	420	511	56.4	63.6	**72.0**	68.9	76.5	**78.3**
HOLO4K	4009	9584	52.4	62.0	**68.6**	63.1	71.0	**74.0**
COACH420(Mlig)	300	378	57.4	64.0	**71.2**	70.4	**76.5**	**76.5**
HOLO4K(Mlig)	3448	6886	56.9	68.3	**73.7**	70.3	79.6	**80.9**

The numbers represent identification success rate [%] measured by D_{CA} criterion (distance from pocket center to closest ligand atom) with 4 Å threshold considering only pockets ranked at the top of the list (n is the number of ligands in considered structure).
*predictions of Fpocket re-scored by PRANK algorithm (which is included in P2Rank software package)
†average results of 10 independent 5-fold cross-validation runs

Table 2. Comparison with SiteHound, MetaPocket 2.0 and DeepSite. Exact pairwise comparison on subsets of the datasets on which compared methods finished successfully. Datasets JOINED/* and HOLO4K/* are subsets of JOINED and HOLO4K on which respective methods finished successfully and produced predictions (SH=SiteHound, MP=MetaPocket2, DS=DeepSite). Similarly for (Mlig) datasets. In datasets labeled as *Mlig*, relevant ligands (and therefore binding sites that are expected to be predicted) were determined in a different way (see Relevant Ligands).

Dataset	proteins	ligands	Top-n		Top-(n+2)	
			SiteHound	P2Rank	SiteHound	P2Rank
COACH420/SH	284	345	53.0	**72.8**	69.3	**77.1**
HOLO4K/SH	2878	6826	50.1	**68.8**	62.1	**74.3**
COACH420(Mlig)/SH	203	257	51.0	**70.4**	67.7	**75.1**
HOLO4K(Mlig)/SH	2470	4843	53.1	**74.0**	67.8	**81.3**
			MetaPocket 2.0	P2Rank	MetaPocket 2.0	P2Rank
COACH420/MP	417	508	63.4	**72.2**	74.6	**78.1**
HOLO4K/MP	2575	5021	57.9	**72.4**	68.6	**77.7**
COACH420(Mlig)	300	378	62.2	**71.2**	73.3	**76.5**
HOLO4K(Mlig)/MP	2202	3706	62.3	**78.3**	75.2	**84.6**
			DeepSite	P2Rank	DeepSite	P2Rank
COACH420	420	511	56.4	**72.0**	63.4	**78.3**
HOLO4K/DS	3991	9557	45.6	**68.6**	48.2	**74.0**
COACH420(Mlig)	300	378	54.5	**71.2**	61.6	**76.5**
HOLO4K(Mlig)/DS	3430	6861	50.8	**73.7**	54.4	**80.8**

The numbers represent identification success rate [%] measured by D_{CA} criterion (distance from pocket center to closest ligand atom) with 4 Å threshold considering only pockets ranked at the top of the list (n is the number of ligands in considered structure).

Table 3. Predictive performance of different feature sets. The numbers represent identification success rate [%] measured by D_{CA} criterion (distance from pocket center to closest ligand atom) with 4 Å threshold considering only pockets ranked at the top of the list (n is the number of ligands in considered structure). In rows representing feature sets each number is an average results of 10 train/eval runs.

	JOINED		COACH420		HOLO4K	
	Top-n	Top-(n+2)	Top-n	Top-(n+2)	Top-n	Top-(n+2)
[protrusion][a]	62.8	73.4	64.2	73.0	59.3	67.7
[full−protrusion][b]	64.3	75.9	60.5	71.8	68.2	75.9
[full−propensities][c]	73.9	80.5	71.6	77.9	69.1	74.7
[full][d]	74.0	80.2	71.4	78.1	70.1	75.4
P2Rank (default model)[e]	74.4	80.2	72.0	78.3	68.6	74.0

[a]reduced set of features that includes only one feature: protrusion
[b]reduced set of features that does not include protrusion
[c]reduced set of features that does not include atomic propensity features (see "ap*" features)
[d]full set of features
[e]Default pre-trained model of P2Rank (with full set of features). Note that numbers are slightly different from [full] since this row represents the results of a particular pre-selected model (the default model P2Rank is distributed with), while [full] row contains averages of 10 runs. Model selection was done based on performance on JOINED.

3 FEATURES

Features that are used to describe accessible surface points are listed in Table 4.

Table 4. - Complete list of features that are used to describe solvent accessible surface (SAS) points. *Type: a...values are assigned to protein solvent exposed atoms and then projected onto SAS points p...values are assigned directly to SAS points **source: values are determined by Amino Acid Type table / Atom Type table / given in PDB file / calculated on the spot

Feature name	T*	source**	description
`hydrophobic`	a	AA tab.	binary attribute, 1 for hydrophobic residues
`hydrophilic`	a	AA tab.	binary attribute, 1 for hydrophilic residues
`hydrophatyIndex`	a	AA tab.	side-chain hydropathy index with values in range $\langle -4.5, 4.5 \rangle$ [5]
`aliphatic`	a	AA tab.	binary attribute, 1 for aliphatic residues
`aromatic`	a	AA tab.	binary attribute, 1 for aromatic residues
`sulfur`	a	AA tab.	binary attribute, 1 for residues containing sulfur
`hydroxyl`	a	AA tab.	binary attribute, 1 for hydroxyl group containing residues
`basic`	a	AA tab.	binary attribute, 1 for basic residues
`acidic`	a	AA tab.	binary attribute, 1 for acidic residues
`amide`	a	AA tab.	binary attribute, 1 for amide group containing residues
`posCharge`	a	AA tab.	binary attribute, 1 for positively charged residues
`negCharge`	a	AA tab.	binary attribute, 1 for negatively charged residues
`hBondDonor`	a	AA tab.	binary attribute, 1 for H-bond donor containing residues
`hBondAcceptor`	a	AA tab.	binary attribute, 1 for H-bond acceptor containing residues
`hBondDonorAcceptor`	a	AA tab.	binary attribute, 1 for residues that have H-bond donor AND acceptor
`polar`	a	AA tab.	binary attribute, 1 for polar residues
`ionizable`	a	AA tab.	binary attribute, 1 for ionizable residues
`vsAromatic`	a	AT tab.	VolSite atomic level features [1]
`vsCation`	a	AT tab.	
`vsAnion`	a	AT tab.	
`vsHydrophobic`	a	AT tab.	
`vsAcceptor`	a	AT tab.	
`vsDonor`	a	AT tab.	
`atomicHydrophobicity`	a	AT tab.	Atom type hydrophobicity scale [3]
`apRawValids`	a	AT tab.	Ligand binding propensity for biologically valid ligands [4]
`apRawInvalids`	a	AT tab.	Ligand binding propensity for biologically invalid ligands [4]
`bfactor`	a	given	B-factor number of the atom from pdb file
`atoms`	p	calc.	absolute number of protein exposed atoms in the neighbourhood (within 6 Å radius of the point)
`atomDensity`	p	calc.	number of protein exposed atoms weighted by distance
`atomC`	p	calc.	number of carbon atoms in the neighbourhood
`atomO`	p	calc.	number of oxygen atoms in the neighbourhood
`atomN`	p	calc.	number of nitrogen atoms in the neighbourhood
`hDonorAtoms`	p	calc.	number of H-bond donor atoms in the neighbourhood
`hAcceptorAtoms`	p	calc.	number of H-bond acceptor atoms in the neighbourhood
`protrusion`	p	calc.	Protein surface protrusion inspired by [6] calculated simply as number of all protein atoms (not just exposed) within 10 Å radius of the point

3.1 Feature Importances

Table 5 contains calculated feature importances.

Table 5. Feature Importances.

feature	importance
protrusion	0.084528
bfactor	0.013888
apRawInvalids	0.011785
vsAromatic	0.010165
apRawValids	0.009403
atomO	0.009275
hydrophobic	0.008630
hydrophilic	0.007643
vsAcceptor	0.006244
vsHydrophobic	0.005273
atoms	0.005188
aromatic	0.004433
atomN	0.004236
hydrophatyIndex	0.004232
atomC	0.003687
vsDonor	0.003451
aliphatic	0.003350
atomicHydrophobicity	0.002663
hBondDonorAcceptor	0.002650
hDonorAtoms	0.002626
atomDensity	0.002549
polar	0.002402
ionizable	0.002142
hAcceptorAtoms	0.001904
hBondAcceptor	0.001705
sulfur	0.001621
negCharge	0.001538
acidic	0.001504
basic	0.001467
hydroxyl	0.001328
vsAnion	0.001072
hBondDonor	0.001059
posCharge	0.001021
vsCation	0.000832
amide	0.000831

Feature importances calculated by Random Forest algorithm on CHEN11 dataset. Avg. of 10 runs.

REFERENCES

[1]J. Desaphy, K. Azdimousa, E. Kellenberger, and D. Rognan. Comparison and druggability prediction of protein-ligand binding sites from pharmacophore-annotated cavity shapes. *Journal of chemical information and modeling*, 52(8):2287–2299, 2012.

[2]L. Hu, M. L. Benson, R. D. Smith, M. G. Lerner, and H. A. Carlson. Binding moad (mother of all databases). *Proteins: Structure, Function, and Bioinformatics*, 60(3):333–340, 2005.

[3]L. H. Kapcha and P. J. Rossky. A simple atomic-level hydrophobicity scale reveals protein interfacial structure. *Journal of Molecular Biology*, 426(2):484 – 498, 2014.

[4]N. A. Khazanov and H. A. Carlson. Exploring the composition of protein-ligand binding sites on a large scale. *PLoS computational biology*, 9(11):e1003321, Nov 2013.

[5]J. Kyte and R. F. Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, 157(1):105 – 132, 1982.

[6]A. Pintar, O. Carugo, and S. Pongor. Cx, an algorithm that identifies protruding atoms in proteins. *Bioinformatics*, 18(7):980–984, 2002.

[7]R. Wang, X. Fang, Y. Lu, and S. Wang.

[8]J. Yang, A. Roy, and Y. Zhang. Biolip: a semi-manually curated database for biologically relevant ligandprotein interactions. *Nucleic Acids Research*, 41(D1):D1096–D1103, 2013.

Improving quality of ligand-binding site prediction with Bayesian optimization

Reference

KRIVÁK R., HOKSZA D., ŠKODA P.: **Improving quality of ligand-binding site prediction with Bayesian optimization**. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2017), pp. 2278–2279. doi:10.1109/BIBM.2017.8218024

Author's highlights

Extended conference abstract summarising several updates to the algorithm. Most notable of them is the implementation of a framework for using Bayesian optimization to optimize several arbitrary parameters of the algorithm at the same time.

Improving quality of ligand-binding site prediction with Bayesian optimization

Radoslav Krivák
Faculty of Mathematics and Physics
Charles University in Prague
Prague, Czech Republic
Email: krivak@ksi.mff.cuni.cz

David Hoksza
Faculty of Mathematics and Physics
Charles University in Prague
Prague, Czech Republic
Email: hoksza@ksi.mff.cuni.cz

Petr Škoda
Faculty of Mathematics and Physics
Charles University in Prague
Prague, Czech Republic
Email: skoda@ksi.mff.cuni.cz

***Abstract*—Ligand binding site prediction from protein structure plays an important role in various complex rational drug design efforts. Its applications include drug side effects prediction, docking prioritization in inverse virtual screening and elucidation of protein function in genome wide structural studies. Currently available tools have limitations that disqualify them from many possible use cases. In general they are either fast and relatively inaccurate (e.g. purely geometric methods) or accurate but too slow for large scale applications (e.g. methods that rely on a large template libraries of known protein-ligand complexes). P2Rank is a recently intorduced machine learning based method that have already exhibited speeds comparable to fastest geometric methods while providing much higher identification success rates. Here we present an improved version that brings speed-up as well as higher quality predictions. A leap in predictive performance was achieved thanks to the technique of Bayesian optimization, which allowed simultaneous optimization of numerous arbitrary parameters of the algorithm. We have evaluated our method with respect to various performance and prediction quality criteria and compared it to other state of the art methods, as well as to it's previous version, with encouraging results.**

Keywords*—*Ligand binding site prediction; protein surface; machine learning; Random Forests; Bayesian optimization;

I. Introduction

Ligand binding site prediction from protein structure has many applications related to rational drug design. It can find employment in various tasks such as drug side-effects prediction, docking prioritization, structure based virtual screening and structure-based target prediction. Increasingly it can be seen applied in genome-wide structural studies that try to analyze and compare all known and putative binding sites. Many of those use cases imply the need for fast standalone tool that can be used as a stable part of larger pipeline. This disqualifies many currently available tools that are available only as web servers and/or are simply too slow.

Existing methods for ligand binding site prediction are based on variety of algorithmic approaches that involve protein geometry, energetical calculations, sequence conservation or search in a template library of known protein-ligand complexes. Methods based on consensus of other algorithms and on machine learning have also emerged. With a bit of simplification it can be said that existing methods are either fast and relatively inaccurate (e.g. purely geometric methods) or accurate but too slow for large-scale applications (e.g. methods that rely on a large template libraries).

II. P2Rank algorithm

Previously we have developed P2Rank [1], a machine learning based algorithm and command line tool for fast and accurate ligand bindng site prediction. P2Rank is based on classification of local geometrical neighborhoods represented by the points lying on protein's solvent accessible surface. Each point is represented by a vector of properties that describe local geometry and physico-chemical properties that are derived mainly from neighboring protein atoms. Random Forest classifier is trained on a dataset of known protein ligand complexes and then used to predict ligandability score of each point. Finally, the points with ligandability score that is higher than certain threshold are clustered into predicted sites, which are then scored and ranked. The speed is comparable to the fastest geometric algorithms like Fpocket (around 1 second for prediction one protein on average single core CPU).

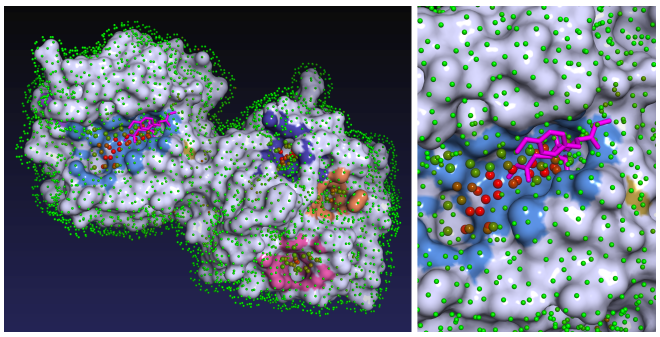


Fig. 1. P2Rank in action: predicted ligand binding sites for structure 1FBL

Improvements in presented version include more sophisticated clustering and scoring, new geometrical descriptors, better dealing with class imbalances and new faster implementation of Random Forest. However, as in many other complex algorithms, practically every step (feature aggregation, training, prediction, clustering, scoring, ...) has several arbitrary parameters or thresholds, and mentioned improvements only add to that. Default value of each of those parameters needs to be determined in order for algorithm to achieve optimal performance. This can be done by iterative manual tuning and grid optimization but only to a certain point. Parameters can exhibit complex nonlinear interdependencies and need to be optimized together. Finding a global optimum for more than a dozen of interdependent parameters becomes practically impossible with this approach.

An example of a 'clique' of few parameters that are interdependent with respect to predictive performance consists of: (a) density of points on accessible surface (i.e. instances which we classify), (b) cutoff distance around a ligand that define ligandable vs. unligandable points in the training set, (c) final target class weight ratio (for weighting instances during training), (d) threshold of predicted ligandabilty score that define ligandable points in prediction phase, (e) minimal size of a cluster, (f) clustering distance. Trying to optimize those parameters indvidually or two at a time by grid optimization would not be very useful, as they are all related to class imbalance and distribution of predicted ligandability scores and need to be optmized simultaneously.

III. Bayesian optimization

Bayesian optimization is a general sequential strategy for optimization of expensive black-box functions. It works by building a model of objective function, based on it deciding which parameter assignment should be tried next and iteratively updating the model. The next sampled location in parameter space is determined by an acquisition function, which represents a trade-off between exploration (where the objective function is very uncertain) and exploitation (where the objective function is expected to be high) [2]. Perhaps the most commonly used model for modeling objective function in Bayesian optimization is the Gaussian process (GP), as it is implemented in the Spearmint package [3] we have used.

In recent years Bayesian optimization has become prominent especially as a tool to optimize hyperparameters of machine learning models. We have applied it to optimize not just model hyperparameters, but all critical parameters of the algorithm that can influence predictive performance and quality of predictions. Those include all forementioned interdependent parameters as well as parameters related to feature extraction such as neighborhood radius (i.e. size of the surface patch represented by a single point).

This approach to optimization can be too powerful which can lead to certain pittfalls, namely easy overfitting to a particular development dataset or to a narrowly defined performance metric. We discuss how to avoid or mediate those issues.

IV. Results

Optimization runs yielded some surprising parameter assignments that could have been hardly selected by manual tuning, but which nevertheless led to performance improvements. Results of the final model and parameter assignment (chosen based on development set performance) are shown in Table 1. We have compared our method with the previous version as well as with few state-of-the-art methods, namely geometric algorithm Fpocket [4], consensus based MetaPocket 2.0 [5] and deep learning based DeepSite [6]. Displayed are identification success rates (in per cent) according to several criteria and an overlap based prediction quality metric. New version (P2Rank 2.1) shows improvements in all metrics over the original one (P2Rank 2.0) as well as over other methods.

TABLE I. Results on the test set (4009 proteins)

method	DCA	DCC	DSO	avg. overlap ratio
DeepSite	45.6	31.5	n/a	n/a
Fpocket	52.4	38.8	41.7	0.33
MetaPocket2	56.6	43.4	n/a	n/a
P2Rank 2.0	68.6	52.0	67.9	0.49
P2Rank 2.1	71.4	56.6	71.0	0.53

Identification criteria:
DCA: distance between the center of the pocket to the closest ligand atom with 4 Å threshold.
DCC: distance between the center of the pocket to the center of the ligand with 5 Å threshold.
DSO: Discretized Surface Overlap with a threshold of 20%. Binding site is correctly identified if the intersection of points covered by the ligand and predicted pocket is not smaller than 20% of their union.
In all cases considered is only Top-n predicted pockets where n is the number of ligands in considered query structure.

V. Conclusions

P2Rank is fast and accurate ligand binding site prediction algorithm based on machine learning. Here we present P2Rank 2.1, a faster and more sophsticated version. Increasing the complexity of the algorithm did not, however, automatically lead to improvement in predictive performance. Many additional arbitrary parameters were introduced and due to complex interparameter dependencies it became hard to see if implemented improvements are actually helping. Ultimately, we have used the technique of bayessian optmization to find optimal values of those arbitrary parameters, which lead to final performance improvements.

This optimization approach is becoming commonplace in machine learning literature (for hyperparameter optimization), but we believe that it can be useful in development of any algorithm that involve a number arbitrary perameters (which is true for most of algorithms in sequence and structural bioinformatics).

Acknowledgment

This work was supported by the Grant Agency of Charles University [project Nr. 1556217 and Nr. 174615]. This work was also supported by project SVV 260451.

References

[1] R. Krivák and D. Hoksza, *P2RANK: Knowledge-Based Ligand Binding Site Prediction Using Aggregated Local Features*. Cham: Springer International Publishing, 2015, pp. 41–52.

[2] E. Brochu, V. M. Cora, and N. de Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *CoRR*, vol. abs/1012.2599, 2009.

[3] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 2951–2959.

[4] V. Le Guilloux, P. Schmidtke, and P. Tuffery, "Fpocket: An open source platform for ligand pocket detection," *BMC Bioinformatics*, vol. 10, no. 1, p. 168, 2009. [Online]. Available: http://www.biomedcentral.com/1471-2105/10/168

[5] Z. Zhang, Y. Li, B. Lin, M. Schroeder, and B. Huang, "Identification of cavities on protein surface using multiple computational approaches for drug binding site prediction." *Bioinformatics (Oxford, England)*, vol. 27, no. 15, pp. 2083–2088, 2011.

[6] J. Jimnez, S. Doerr, G. Martnez-Rosell, A. S. Rose, and G. De Fabritiis, "Deepsite: protein-binding site predictor using 3d-convolutional neural networks," *Bioinformatics*, vol. 33, no. 19, pp. 3036–3042, 2017.

Peptide-Binding Site Prediction From Protein Structure via Points on the Solvent Accessible Surface

Reference

KRIVÁK R., JENDELE L., HOKSZA D.: **Peptide-Binding Site Prediction From Protein Structure via Points on the Solvent Accessible Surface**. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics* (New York, NY, USA, 2018), BCB '18, Association for Computing Machinery, p. 645–650. doi:10.1145/3233547.3233708

Author's highlights

P2Rank was modified for the prediction of peptide binding Sites. Apart from working with different datasets the method had to be redesigned to include a residue-centric prediction mode. To achieve performance better than other existing methods several new descriptors were introduced, including geometrical descriptors and sequence conservation score.

Peptide-Binding Site Prediction From Protein Structure via points on the Solvent Accessible Surface

Radoslav Krivák
Faculty of Mathematics and Physics, Charles University
Prague, Czech Republic
krivak@ksi.mff.cuni.cz

Lukáš Jendele
Faculty of Mathematics and Physics, Charles University
Prague, Czech Republic
jendele@ksi.mff.cuni.cz

David Hoksza
Faculty of Mathematics and Physics, Charles University
Prague, Czech Republic
hoksza@ksi.mff.cuni.cz

ABSTRACT

Protein-peptide binding interactions play an important role in cellular regulation and are functionally important in many diseases. If no prior knowledge of the location of a binding site is available, prediction may be needed as a starting point for further modeling or docking. Existing approaches to prediction either require a sequence of the peptide to be already known or offer an unsatisfactory predictive performance. **Here we propose P2Rank-Pept, a new machine learning based method for prediction of peptide-binding sites from protein structure. We show that our method significantly outperforms other evaluated methods, including the most recent structure based prediction method SPRINT-Str published last year (AUC:** $0.85 > 0.78$**).** P2Rank-Pept utilizes local structural and sequence information, including evolutionary conservation, and builds a prediction model based on a Random Forest classifier. The novelty of our approach lies in using points on the solvent accessible surface as a unit of classification (as opposed to the typical approach of focusing on amino acid residues), and in the application of the robust technique of Bayesian optimization to systematically optimize arbitrary parameters of the algorithm. Our results assert that P2Rank software package is a viable framework for developing top-performing binding-site prediction methods for different types of binding partners.

CCS CONCEPTS

• **Applied computing** → **Molecular structural biology**; *Bioinformatics*; *Computational proteomics*; • **Computing methodologies** → *Supervised learning*; *Classification and regression trees*;

KEYWORDS

peptides, protein-peptide binding, binding site prediction, protein structure, machine learning, random forests

ACM Reference Format:
Radoslav Krivák, Lukáš Jendele, and David Hoksza. 2018. Peptide-Binding Site Prediction From Protein Structure via points on the Solvent Accessible Surface. In *9th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (ACM-BCB'18), August 29-September 1, 2018, Washington, DC, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3233547.3233708

ACM-BCB'18, August 29-September 1, 2018, Washington, DC, USA

ACM ISBN 978-1-4503-5794-4/18/08…$15.00
https://doi.org/10.1145/3233547.3233708

1 INTRODUCTION

Motivation. Peptide-mediated interactions belong to the most important molecular interactions that involve proteins. Accurate prediction of a peptide binding site can provide a useful starting point for peptide binder modeling and docking. This is important particularly for peptides, because blind peptide docking is available only by a subset of methods and it is very expensive due to many degrees of freedom. Avoiding global blind docking allows intensifying the search to relevant sites. Ab initio versions [17, 22] of Rosetta FlexPepDock [1] can predict ideal peptide chain and its conformation, but they assume prior approximate knowledge of the peptide-binding site. This approach in particular [8, 15], and rational design of peptide-based modulators in general has recently became rapidly growing avenue for targeting protein-protein interactions [6, 20, 33].

The problem is defined in the following way: given any unannotated protein structure, predict which residues are peptide-binding (and additionally cluster them into peptide-binding sites). Variations of the problem include predicting peptide-binding residues from the sequence alone and predicting a binding site for a particular given peptide sequence. Technically, the problem lies between protein-ligand binding site and protein-protein interface prediction and shares the challenging aspects of both. Like in the case of protein-ligand binding, it is a problem with very high class imbalance. At the same time, like in protein-protein interactions, and unlike in protein-ligand binding, the location of binding sites is not as highly correlated with deep concave geometrical pockets.

Related work. Earlier peptide binding site prediction methods were focused on specific protein types. Many of available general methods require a peptide sequence to be already known [23, 29, 30]. Binding site prediction when specific peptide sequence is unknown is possible only by a handful of published methods. Among them is PeptiMap [15], an advanced method based on fragment mapping. Two other energetic methods have limited availability. ACCLUSTER [5] has an ab intio version, however is only available as a web server that delivers results by e-mail which did not seem to work at the time of writing. FoldX [31] is available for download, however, FoldX is a commercial tool and the protocol for predicting peptide sites is not documented.

Machine learning has been applied to prediction of various kinds of binding sites [19], but only few of the methods were developed specifically for peptides. The only methods that use machine learning and structural information are recently published SPRINT-Str [28] and an overlooked, but conceptually interesting MultiVORFFIP [24, 25]. PBRpredict [10], another promising sequence-based predictor appeared only very recently.

In this paper, we introduce P2Rank-pept, a novel method for prediction of peptide binding sites from protein structure. P2Rank-pept belongs to the methods that do not rely on prior knowledge of the particular peptide sequence. The fact that aforementioned methods were published does not always mean that they are available and ready to be used. Most of the tools are available only as web servers, which is not ideal for constructing reliable pipelines (for illustration why: during the writing of this paper, two of the online tools were momentarily not working). Our goal is therefore to develop a stand-alone and open source tool, as we believe this is the best way how to provide value for the community.

2 METHODS

2.1 P2Rank-Pept Algorithm

Conceptually, the working of the algorithm is centered on points on the solvent accessible surface of a protein. Those points represent a local chemical 3D neighbourhoods that are centered on them. At the same time, they can be seen as locations of contact atoms of potential binding peptides. First, we try to classify those points, and only then we classify residues based the points adjacent to them. This basic principle of the algorithm was successfully applied in our previous work on protein-ligand binding site prediction [12, 13], however, it was never used for prediction of peptide-binding sites. Most of the other prediction methods focus directly on classification of residues. Intuitively, we believe it is more natural to classify portions of empty space around the protein than to classify residues. The empty space near the protein surface could be either occupied by a ligand (peptide) atom or not, whereas a residue can be peptide-binding only partially (perhaps only with a single contact atom). Focusing on SAS points thus allows more precise, less noisy labeling.

Figure 1 shows the outline of the algorithm i.e. the steps that P2Rank-Pept follows to predict peptide-binding residues using pre-trained classification model. Prediction on a particular protein is further illustrated in Figure 2. Details of individual steps are described in the following paragraphs.

Step 1: Generate points. Set of regularly-spaced points on the Solvent Accessible Surface of the protein (SAS points) is calculated by a fast numerical algorithm [9] implemented in CDK library [32]. Alternatively, we could have used points placed on a regular 3D grid to discretize the empty space around the protein. However, that would be less efficient as more points would be needed to cover whole surface of a protein with similar effective density. More importantly, another advantage of using SAS points is that they are roughly equidistant to closest protein atoms. This fact helps to make distance-weighed features that are projected from neighboring protein atoms to be more discriminative. The probe radius ρ is one of the configurable parameters (1.8 Å by default).

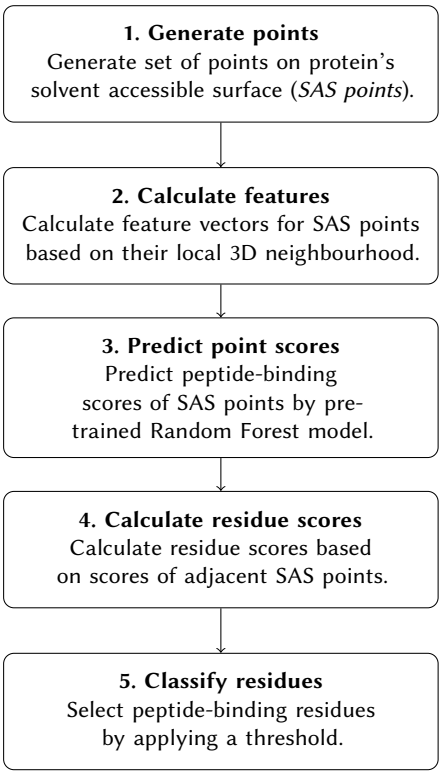


Figure 1: P2Rank-Pept algorithm outline

Step 2: Calculate features. We consider three basic types of features: atom-related, residue-related and SAS point-related according to which entity are those features most naturally assigned to. For instance, the b-factor from PDB file is assigned primarily to protein atoms, whereas evolutionary conservation scores are calculated primarily for residues. Some geometrical features, such as surface protrusion, are calculated directly for SAS points. Thus, many of the features are assigned to protein atoms or residues first, and only then they are projected onto SAS points. In other words, a portion of SAS point feature vector is aggregated from neighboring atoms and residues. The notion and a particular process of aggregating these local features is central to the working of the algorithm.

Atom-related feature are aggregated in the following way: the value of the SAS feature is the average of distance weighted ($w(d) = 1 - d/10$) atomic feature values using atoms in the spherical neighbourhood of the SAS point (radius is another parameter, 10 Å by default). Aggregation of residue-related features is done in one of two ways: (**1**) calculating an average value (or sum) from a set of neighboring residues, or (**2**) taking a value from the nearest residue. For some residue-related features, both methods are applied at the same time as it helped to increase the performance.

Step 3: Predict point scores. Pre-trained Random Forest assigns a classification score from the interval $[0, 1]$ to each SAS point (sometimes incorrectly referred to as predicted probability). This score represents a confidence of the classifier that a point is at a

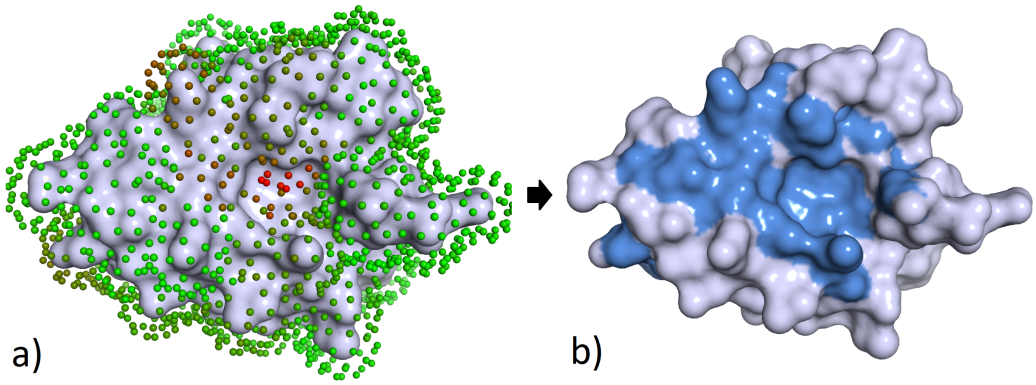


Figure 2: Peptide-binding residue prediction based on points on the Solvent Accessible Surface. a) Protein (3NFK/A) is covered in a layer of points lying on the solvent accessible surface. Each point represents its local chemical neighborhood and is described by a feature vector calculated from its surroundings. Points are colored according to the peptide-binding score ($\in$ [0,1]) predicted by a Random Forest classifier (*green=0/red=1*). b) Peptide-binding score of any given solvent exposed residue is based on the score of its adjacent points (radius of the cutoff and the form of aggregation function were subject to optimization). Residues with the score above a certain threshold are labeled as positives (*blue*).

peptide-binding location. We do not work with binary output of the classifier but directly with the classification score.

Step 4: Calculate residue scores. Each solvent exposed residue is assigned a score that is calculated from scores of the SAS points in its vicinity. Residues that are not solvent exposed are assigned score of 0. Residue score is calculated by the formula

$$\text{residue\_score}(R) = \sum_{P_i \in \text{NP}(R)} \text{predicted\_score}(P_i)^{\beta}, \qquad (1)$$

where R is the residue, NP(R) is the set of SAS points that are located within 3 Å of any heavy atom of the residue. Exponent β is an arbitrary parameter that can put more weight either to the lower or to the higher predicted scores and as such was a subject to optimization.

Previous formula gives us score from the interval $[0, \infty)$. To transform the score to $[0, 1]$ interval we use the formula

$$\text{normalized\_score}(R) = \frac{2}{\pi} \arctan\big(\text{residue\_score}(R)\big). \qquad (2)$$

Step 5: Classify residues. Residue is designated as peptide-binding if residue_score(R) $> \tau$, a classification threshold that is another parameter of the algorithm.

2.2 Features

Compared to P2Rank, our earlier ligand-binding site prediction algorithm, we had to develop and employ a variety of new features to achieve the top performance. Among them were features related to homology-based sequence conservation.

Employed features describe or are related to:

- **chemistry and physics** e.g. hydrophobicity of residues, presence of hydrogen donors and acceptors, …
- **geometry** e.g. related to position and orientation of surrounding residues and atoms, surface protrusion, …
- **secondary structure** e.g. type of secondary structure, length and position in the continuous segment, …
- **sequence patterns** including peptide binding propensities of sequence duplets and triplets calculated from the Train set,
- **evolutionary conservation** including sequence conservation score of residues and surface patches.

Altogether, more than 70 features were employed in the final model. Some of the features were inspired by or adopted from other studies [7, 11, 14, 21]. See Table 1 for the list of most important features.

Conservation scores were calculated using Jensen-Shannon divergence [4] on alignments of sequences from SwissProt and UniRef90 (found by psiblast search with e-value=1e5 and one iteration).

2.3 Aggregation of features and scores

At several points in the algorithm we need to solve a recurring problem: how to aggregate a set (of variable size) of numbers into a single representative number. This problem can be seen in aggregating of atomic-level features (Step 2) and in calculating a residue score (Step 4). Using an average may seem to be an obvious answer, but in some cases a good argument can be made for using a simple summation. Being aware that our intuition may not correspond to the way that allows to extract the strongest signal from the set, we had decided to make no prior assumptions. Instead, we tried to find the optimal aggregation function for each case by optimization. We have expressed an aggregation function in the following generic form:

$$\text{aggregated\_number} = \frac{1}{n^{\alpha}} \sum_{i=1}^{n} \text{value}_i\,, \ \alpha \in [0, 1] \qquad (3)$$

Table 1: Most important features

feature	importance
conserv_atomic[1]	0.0186
bfactor	0.0183
protrusion[2]	0.0173
chem.hydrophobic[3]	0.0166
RAx[4]	0.0157
conserv_sas[1]	0.0150
ss_atomic.extended[5]	0.0140

Gini importances of 7 most important features calculated by Random Forest algorithm on the TR1070 dataset.
[1]conservation score related features
[2]protein surface protrusion [21]
(calculated simply as a number of protein atoms within a sphere)
[3]binary hydrophobicity of a residue
[4]statistical ligand binding propensity of a residue
[5]secondary structure - extended element

As α goes from 0 to 1, the value of aggregation function moves from the sum to the average. This exponent was optimized for each case separately. We believe that using even more generic form of the aggregation function could lead to another performance improvements in the future.

2.4 Prediction Model

Random Forest as a model of choice. In theory, any machine learning algorithm / classification model that can output classification score from the interval $[0, 1]$ can be used at this stage. Following our prior work, we employ Random Forest [2] as a prediction model of choice, as it showed the best performance at similar tasks. Random Forest is able to deal with raw (not normalized), correlated and heterogeneous features, which allows for rapid experimentation. Moreover, our experiments showed that it seems to provide more meaningful predicted score distributions for highly noisy data than alternative approaches like SVM and neural networks. This ability is important because (**a**) we believe that datasets that are used here and in related binding site prediction studies are in general very noisy, and (**b**) we work directly with predicted scores, not binary labels. The assumption that datasets are noisy is inherent to the problem, as we suppose many of the sites on proteins are labeled as negatives (non-binding) incorrectly—they are true binding sites yet to be discovered. Each newly designed peptide inhibitor of another protein-protein interaction only supports this conjecture. The second reason why datasets could be noisy, now in the sense of separability and overlap in the feature vector space, is that a particular set of features that we use may not be discriminative enough.

Training. Points within $\delta = 2.6$ Å of any peptide heavy atom were labeled as positives. This resulted to a class imbalance of ~4:96 on the Train set, which was dealt with by combination of sub-sampling and weighted classification. Ratios of sub-sampling, class weight multiplier and the distance δ were subject to optimization.

Hyper-parameters. Hyper-parameters of the Random Forest algorithm were systematically optimized, and the final model has 500 trees (more did not lead to noticeable improvements) constructed with no depth limit, each using $\sqrt{n}$ features and a bag size of 55%.

2.5 Optimization of parameters

Need for joint optimization. Extensive optimization of most of the arbitrary parameters of the algorithm was performed to assign optimal default values. Those parameters include various distance cut-offs and thresholds used during training, feature extraction and prediction. Few examples of these parameters were mentioned before: (**a**) probe radius ρ for calculating solvent accessible surface, (**b**) score exponent β from the Equation 1, (**c**) a residue score threshold τ, (**d**) hyper-parameters of the Random Forest algorithm, (**e**) balancing ratios for dealing with the class imbalance. Altogether, there are more than two dozens of such parameters that have a direct influence on the performance of the algorithm. Due to the number and interdependence of the parameters, approaches of manual tuning, grid optimization or random search would not be feasible in this case.

Bayesian optimization. We have used the technique of Bayesian optimization [3, 26] to jointly optimize multiple interdependent parameters at once. Bayesian optimization is a general sequential strategy for optimization of black-box functions. This technique lands itself well for this problem, since it generally allows reaching close-to-optimal solutions in a relatively short number of steps, i.e. function evaluations. In our case, the function evaluation is an expensive end-to-end training and evaluation of the whole algorithm on large datasets.

Technical setup and avoiding overfitting. The parameters were optimized with respect to the MCC metric on the D200 set (while training on the D870 set), and some parameters were additionally optimized with respect to the repeated cross-validation results on the whole TR1070 set. Ideally, all optimization would be done with respect to the average of repeated cross-validation results on the whole TR1070 set. However, we have resorted to the Train set split for the lack of time and computational resources. Independent test set TS125 was not used during parameter optimization, as this would lead to overfitting and biased results.

2.6 Datasets

We were working with the Train and Test datasets introduced in SPRINT-Str study [28]. The Train dataset contains 1070 and the Test set 125 peptide-binding complexes, where each complex consist of one protein chain and one peptide chain. Protein residues were labeled positive (peptide-binding) and negative (non-binding) by the authors of the datasets using 3.5 Å distance threshold between

Table 2: Datasets

Dataset	proteins	residues	positives
Train (TR1070)	1070	252677	5.7%
Dev-Train (D870)	870	203168	5.7%
Dev-Test (D200)	200	49509	5.4%
Test (TS125)	125	30870	5.5%

Table 3: Results on the independent test set TS125

Method	MCC	AUC	F-measure	ACC	SEN	SPE
P2Rank-Pept (this work)†	**0.346**±.003	**0.851**±.000	**0.383**±0.002	0.922	0.440	0.950
P2Rank-Pept[−conservation]†	0.341±.003	0.838±.000	0.380±0.003	0.919	0.442	0.948
SPRINT-Str [28]	0.293	0.782	0.309	0.941	0.24	0.98
Multi-VORFFIP [25]	0.212	0.778	0.225	0.826	0.506	0.845
PeptiMap [15]	0.27	0.63	0.294	0.92	0.32	0.95
SPRINT-Seq [27]	0.20	0.68	0.221	0.92	0.21	0.96
PepSite [29]	0.20	0.61	0.219	0.929	0.18	0.97
PinUp [18]	0.13	0.58	0.18	0.88	0.24	0.91
VisGrid [16]	0.15	0.63	0.19	0.89	0.24	0.928

†averages of 10 train/eval runs trained on TR1070 and evaluated on TS125

any heavy atom of the peptide and any heavy atom of the residue. Several proteins were removed from the original Train set due to mismatch between chains in labeling file and chains in the structure PDB file. The independent Test set (T125), is exactly the same as the one used in the aforementioned study. We have also randomly split the Train set to Dev-Train set (D870) and Dev-Test set (D200) that were used during method development and optimization. Notable is the high class imbalance of positive and negative residues (Table 2).

3 RESULTS AND DISCUSSION

We have evaluated the performance of our method on the independent test set (T125) and compared it to the results of other methods that were published in [28]. The same evaluation methodology and the same datasets were used. Additionally, we have gathered the results of Multi-VORFFIP [25], which was not evaluated in previous study. It should be noted that Multi-VORFFIP is also a machine learning method, and there is a possibility it was trained on some proteins from the Test set. SPRINT-Str, on the other hand, was trained on roughly the same dataset as our method.

P2Rank-Pept outperforms other methods. Results in Table 3 show that P2Rank-Pept clearly outperforms other methods in terms of MCC, AUC and F-measure by large margins. It falls short behind SPRINT-Str in Specificity (SPE) and overall Accuracy. This is due to shifted precision-recall balance toward recall (i.e. predicting more false positives and less false negatives). However, only MCC, AUC and F-measure are metrics that are relevant in problems with high class imbalance, such as this problem.

Sequence conservation has little effect. We have further evaluated the performance of the version of the algorithm that does not use sequence conservation at all (see P2Rank-Pept[−conservation] in Table 3 and [−c] in Table 4). Interestingly, although conservation tops the list of the most important features (Table 1), it seems that it does not provide much more additional predicting value to the remaining feature set. This is rather surprising, as other methods, notably SPRINT-Str, found conservation to be crucial to achieving their performance. A version of P2Rank-Pept without conservation is desirable, as this feature is the most expensive to calculate, and requires installing rather elaborate pipeline to calculate the sequence conservation scores. Thus, P2Rank-Pept[−conservation] is a more pragmatic version, as it seem to offer almost the same performance, while being by orders of magnitudes faster and more ready to use.

Limitations and future work. Although our results are encouraging, the evaluation it this paper is subject to some limitations. The main one, as we see it, is that the performance was evaluated and compared to other methods only by a residue-centric methodology. This was done mainly so we could compare the proposed method to other methods, which performance was reported in this way. Most of the methods do not even allow for other types of evaluation, as they only produce residue labels and not binding sites as such. However, we believe that binary classification metrics focusing on residues may not always tell the whole story. For the practical purposes, it may be more important to assess how many binding sites were completely missed, and how many were predicted at least partially. We plan to extend on that and thoroughly evaluate our method using other metrics, such as site identification success rate and peptide coverage.

Table 4: Results using different feature sets

	Train†		Test*	
Feature set	AUC	MCC	AUC	MCC
[full]	.848±.001	.323±.003	.851±.000	.346±.003
[−c]	.840±.001	.322±.003	.838±.000	.341±.003
[−pb]	.843±.001	.315±.002	.846±.001	.330±.003
[−c,pb]	.835±.001	.308±.002	.829±.001	.317±.003
[c]	.772±.001	.204±.002	.781±.001	.220±.002
[pb]	.700±.001	.126±.001	.712±.001	.134±.001
[c,pb]	.759±.001	.204±.001	.777±.001	.230±.002

c: conservation related features, pb: protrusion and b-factor
†averages of 10 independent 5-fold cross-validation runs on TR1070
*averages of 10 train/eval runs trained on TR1070 and evaluated on TS125

4 CONCLUSION

The main contributions of this work can be summarized as follows. We have introduced P2Rank-Pept, a novel approach to prediction of protein-peptide binding sites from structure. We have evaluated available tools for protein-peptide binding site prediction and showed that P2Rank-pept offers significantly better performance. The novelty of our approach lies in focusing on points on the protein solvent accessible surface instead of residues as a units of prediction. We have applied the technique of Bayesian optimization to simultaneously optimize various arbitrary parameters of the algorithm on the development dataset. This approach of systematic optimization is still rarely seen for methods of this type and the choice of values of arbitrary parameters is often left without discussion. Our results assert that open source software package P2Rank can be used as a framework for developing structural prediction methods. Proposed algorithm will be released as an open source stand-alone tool (github.com/rdk/p2rank), and we believe it can contribute to designing new peptide based protein-protein interaction inhibitors.

ACKNOWLEDGMENTS

This work was supported by the project SVV 260451 and by the Grant Agency of Charles University [project Nr. 1556217].

REFERENCES

[1] Raveh Barak, London Nir, and Schueler-Furman Ora. [n. d.]. Sub-angstrom modeling of complexes between flexible peptides and globular proteins. *Proteins: Structure, Function, and Bioinformatics* 78, 9 ([n. d.]), 2029–2040. https://doi.org/10.1002/prot.22716

[2] Leo Breiman. 2001. Random forests. *Machine Learning* 45 (2001). https://doi.org/10.1023/A:1010933404324

[3] Eric Brochu, Vlad M. Cora, and Nando de Freitas. 2009. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. *CoRR* abs/1012.2599 (2009).

[4] John A Capra and Mona Singh. 2007. Predicting functionally important residues from sequence conservation. *Bioinformatics* 23, 15 (2007), 1875–1882.

[5] Yan Chengfei and Zou Xiaoqin. [n. d.]. Predicting peptide binding sites on protein surfaces by clustering chemical interactions. *Journal of Computational Chemistry* 36, 1 ([n. d.]), 49–61. https://doi.org/10.1002/jcc.23771

[6] Anna D Cunningham, Nir Qvit, and Daria Mochly-Rosen. 2017. Peptides and peptidomimetics as regulators of protein-protein interactions. *Current Opinion in Structural Biology* 44 (2017), 59 – 66. https://doi.org/10.1016/j.sbi.2016.12.009 Carbohydrates: A feast of structural glycobiology - Sequences and topology: Computational studies of protein-protein interactions.

[7] Jérémy Desaphy, Karima Azdimousa, Esther Kellenberger, and Didier Rognan. 2012. Comparison and druggability prediction of protein-ligand binding sites from pharmacophore-annotated cavity shapes. *J. Chem. Inf. Model.* 52, 8 (2012), 2287–2299.

[8] David J Diller, Jon Swanson, Alexander S Bayden, Mark Jarosinski, and Joseph Audie. 2015. Rational, computer-enabled peptide drug design: principles, methods, applications and future directions. *Future Medicinal Chemistry* 7, 16 (2015), 2173–2193. https://doi.org/10.4155/fmc.15.142 PMID: 26510691.

[9] Frank Eisenhaber, Philip Lijnzaad, Patrick Argos, Chris Sander, and Michael Scharf. 1995. The double cubic lattice method: Efficient approaches to numerical integration of surface area and volume and to dot surface contouring of molecular assemblies. *Journal of Computational Chemistry* 16, 3 (1995), 273–284.

[10] Sumaiya Iqbal and Tamjidul Hoque. 2018. PBRpredict-Suite: A Suite of Models to Predict Peptide Recognition Domain Residues from Protein Sequence. *Bioinformatics* (2018), bty352. https://doi.org/10.1093/bioinformatics/bty352

[11] Lauren H. Kapcha and Peter J. Rossky. 2014. A Simple Atomic-Level Hydrophobicity Scale Reveals Protein Interfacial Structure. *Journal of Molecular Biology* 426, 2 (2014), 484 – 498. https://doi.org/10.1016/j.jmb.2013.09.039

[12] Radoslav Krivák and David Hoksza. 2015. Improving protein-ligand binding site prediction accuracy by classification of inner pocket points using local features. *Journal of Cheminformatics* 7, 1 (2015), 12. https://doi.org/10.1186/s13321-015-0059-5

[13] Radoslav Krivák and David Hoksza. 2015. P2RANK: Knowledge-Based Ligand Binding Site Prediction Using Aggregated Local Features. In *Algorithms for Computational Biology*, Adrian-Horia Dediu, Francisco Hernández-Quiroz, Carlos Martín-Vide, and David A. Rosenblueth (Eds.). Springer International Publishing, Cham, 41–52.

[14] Jack Kyte and Russell F. Doolittle. 1982. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology* 157, 1 (1982), 105 – 132. http://www.sciencedirect.com/science/article/pii/0022283682905150

[15] Assaf Lavi, Chi Ho Ngan, Dana Movshovitz-Attias, Tanggis Bohnuud, Christine Yueh, Dmitri Beglov, Ora Schueler-Furman, and Dima Kozakov. [n. d.]. Detection of peptide-binding sites on protein surfaces: The first step toward the modeling and targeting of peptide-mediated interactions. *Proteins: Structure, Function, and Bioinformatics* 81, 12 ([n. d.]), 2096–2105. https://doi.org/10.1002/prot.24422

[16] Bin Li, Srinivasan Turuvekere, Manish Agrawal, David La, Karthik Ramani, and Daisuke Kihara. [n. d.]. Characterization of local geometry of protein surfaces with the visibility criterion. *Proteins: Structure, Function, and Bioinformatics* 71, 2 ([n. d.]), 670–683. https://doi.org/10.1002/prot.21732 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.21732

[17] Haiou Li, Liyao Lu, Rong Chen, Lijun Quan, Xiaoyan Xia, and Qiang Lü. 2014. PaFlexPepDock: Parallel Ab-Initio Docking of Peptides onto Their Receptors with Full Flexibility Based on Rosetta. *PLOS ONE* 9, 5 (05 2014), 1–13. https://doi.org/10.1371/journal.pone.0094769

[18] Shide Liang, Chi Zhang, Song Liu, and Yaoqi Zhou. 2006. Protein binding site prediction using an empirical scoring function. *Nucleic Acids Research* 34, 13 (2006), 3698–3707. https://doi.org/10.1093/nar/gkl454

[19] Angelica Nakagawa Lima, Eric Allison Philot, Gustavo Henrique Goulart Trossini, Luis Paulo Barbour Scott, Vinícius Goncalves Maltarollo, and Kathia Maria Honorio. 2016. Use of machine learning approaches for novel drug discovery. *Expert Opinion on Drug Discovery* 11, 3 (2016), 225–239. https://doi.org/10.1517/17460441.2016.1146250 PMID: 26814169.

[20] Ashley E. Modell, Sarah L. Blosser, and Paramjit S. Arora. 2016. Systematic Targeting of Protein-Protein Interactions. *Trends in Pharmacological Sciences* 37, 8 (2016), 702 – 713. https://doi.org/10.1016/j.tips.2016.05.008

[21] Alessandro Pintar, Oliviero Carugo, and Sándor Pongor. 2002. CX, an algorithm that identifies protruding atoms in proteins. *Bioinformatics* 18, 7 (2002), 980–984. https://doi.org/10.1093/bioinformatics/18.7.980

[22] Barak Raveh, Nir London, Lior Zimmerman, and Ora Schueler-Furman. 2011. Rosetta FlexPepDock ab-initio: Simultaneous Folding, Docking and Refinement of Peptides onto Their Receptors. *PLOS ONE* 6, 4 (04 2011), 1–10. https://doi.org/10.1371/journal.pone.0018934

[23] Adrien Saladin, Julien Rey, Pierre Thévenet, Martin Zacharias, Gautier Moroy, and Pierre Tufféry. 2014. PEP-SiteFinder: a tool for the blind identification of peptide binding sites on protein surfaces. *Nucleic Acids Research* 42, W1 (2014), W221–W226. https://doi.org/10.1093/nar/gku404

[24] Joan Segura, Pamela F. Jones, and Narcis Fernandez-Fuentes. 2011. Improving the prediction of protein binding sites by combining heterogeneous data and Voronoi diagrams. *BMC Bioinformatics* 12, 1 (23 Aug 2011), 352. https://doi.org/10.1186/1471-2105-12-352

[25] Joan Segura, Pamela F. Jones, and Narcis Fernandez-Fuentes. 2012. A holistic in silico approach to predict functional sites in protein structures. *Bioinformatics* 28, 14 (2012), 1845–1850. https://doi.org/10.1093/bioinformatics/bts269

[26] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems 25*. 2951–2959.

[27] Ghazaleh Taherzadeh, Yuedong Yang, Tuo Zhang, Alan Wee-Chung Liew, and Yaoqi Zhou. [n. d.]. Sequence-based prediction of proteinâĂŞpeptide binding sites using support vector machine. *Journal of Computational Chemistry* 37, 13 ([n. d.]), 1223–1229. https://doi.org/10.1002/jcc.24314

[28] Ghazaleh Taherzadeh, Yaoqi Zhou, Alan Wee-Chung Liew, and Yuedong Yang. 2018. Structure-based prediction of protein- peptide binding regions using Random Forest. *Bioinformatics* 34, 3 (2018), 477–484. https://doi.org/10.1093/bioinformatics/btx614

[29] Leonardo G. Trabuco, Stefano Lise, Evangelia Petsalaki, and Robert B. Russell. 2012. PepSite: prediction of peptide-binding sites from protein surfaces. *Nucleic Acids Research* 40, W1 (2012), W423–W427. https://doi.org/10.1093/nar/gks398

[30] Leonardo G. Trabuco, Stefano Lise, Evangelia Petsalaki, and Robert B. Russell. 2012. PepSite: prediction of peptide-binding sites from protein surfaces. *Nucleic Acids Research* 40, W1 (2012), W423–W427. https://doi.org/10.1093/nar/gks398

[31] Erik Verschueren, Peter Vanhee, Frederic Rousseau, Joost Schymkowitz, and Luis Serrano. 2013. Protein-Peptide Complex Prediction through Fragment Interaction Patterns. *Structure* 21, 5 (2013), 789 – 797. https://doi.org/10.1016/j.str.2013.02.023

[32] Egon L. Willighagen, John W. Mayfield, Jonathan Alvarsson, Arvid Berg, Lars Carlsson, Nina Jeliazkova, Stefan Kuhn, Tomáš Pluskal, Miquel Rojas-Chertó, Ola Spjuth, Gilleain Torrance, Chris T. Evelo, Rajarshi Guha, and Christoph Steinbeck. 2017. The Chemistry Development Kit (CDK) v2.0: atom typing, depiction, molecular formulas, and substructure searching. *Journal of Cheminformatics* 9, 1 (06 Jun 2017), 33. https://doi.org/10.1186/s13321-017-0220-4

[33] Paulina Wójcik and Łukasz Berlicki. 2016. Peptide-based inhibitors of protein-protein interactions. *Bioorganic & Medicinal Chemistry Letters* 26, 3 (2016), 707 – 713. https://doi.org/10.1016/j.bmcl.2015.12.084

PrankWeb: a web server for ligand binding site prediction and visualization

Reference

JENDELE L., KRIVAK R., SKODA P., NOVOTNY M., HOKSZA D.: **PrankWeb: a web server for ligand binding site prediction and visualization**. *Nucleic Acids Res. 47*, W1 (Jul 2019), W345–W349. doi:10.1093/nar/gkz424

Author's highlights

We have developed easy to use web interface for P2Rank with web based visualization, the ability to download the results and documented REST API. A custom pipeline for calculating sequence conservation scores was developed as part of the project and a new default model for P2Rank was trained (the one using sequence conservation among features). The performance of the model using conservation was compared to the model without conservation with the result that conservation contributes to a slightly better prediction success rate and results in producing a lower number of more relevant pockets. At the same time P2Rank introduced Java API which allowed it to be used as a library by programs running on JVM.

Published online 22 May 2019 *Nucleic Acids Research, 2019, Vol. 47, Web Server issue* ***W345–W349***
doi: 10.1093/nar/gkz424

PrankWeb: a web server for ligand binding site prediction and visualization

Lukas Jendele[1], Radoslav Krivak[1], Petr Skoda[1], Marian Novotny[2] and David Hoksza[1,3,*]

[1]Department of Software Engineering, Faculty of Mathematics and Physics, Charles University, Czech Republic, [2]Department of Cell Biology, Faculty of Science, Charles University, Czech Republic and [3]Luxembourg Centre for Systems Biomedicine, University of Luxembourg, Luxembourg

Received March 18, 2019; Revised April 27, 2019; Editorial Decision May 03, 2019; Accepted May 09, 2019

ABSTRACT

PrankWeb is an online resource providing an interface to P2Rank, a state-of-the-art method for ligand binding site prediction. P2Rank is a template-free machine learning method based on the prediction of local chemical neighborhood ligandability centered on points placed on a solvent-accessible protein surface. Points with a high ligandability score are then clustered to form the resulting ligand binding sites. In addition, PrankWeb provides a web interface enabling users to easily carry out the prediction and visually inspect the predicted binding sites via an integrated sequence-structure view. Moreover, PrankWeb can determine sequence conservation for the input molecule and use this in both the prediction and result visualization steps. Alongside its online visualization options, PrankWeb also offers the possibility of exporting the results as a PyMOL script for offline visualization. The web frontend communicates with the server side via a REST API. In high-throughput scenarios, therefore, users can utilize the server API directly, bypassing the need for a web-based frontend or installation of the P2Rank application. PrankWeb is available at http://prankweb.cz/, while the web application source code and the P2Rank method can be accessed at https://github.com/jendelel/PrankWebApp and https://github.com/rdk/p2rank, respectively.

INTRODUCTION

The field of structural biology has recently experienced enormous progress in all aspects of structural determination and, as a result, 3D structures of proteins are becoming increasingly available. Indeed, structural genomics consortia are now able to solve protein structures with no known function (1), the information acquired from 3D coordinates for such proteins being used to annotate the proteins. An important clue for predicting protein function is the identification of ligands or small molecules that can bind to the protein. Ligands and other small molecules can either be determined directly within the protein's 3D structure or a 3D structure of the protein can be used to predict ligand binding sites, and thus help to annotate the protein.

A range of protein ligand binding site prediction approaches have been developed over recent years, including a number that are provided as a web service (Table 1). Fpocket (2), SiteHound (3), ConCavity (4), POCASA (5), MetaPocket 2.0 (6), FTSite (7) and bSiteFinder (8) all support online visualization using Jmol (9), a Java-based molecular structure viewer. Due to known security risks, however, Java applets are no longer supported in modern web browsers and these websites can now be considered outdated. A simple solution to the Jmol issue is to use JSmol (10), a JavaScript replacement for Jmol. This is the avenue taken by 3DLigandSite (11), COFACTOR (12,13), COACH (14) ISMBLAB-LIG (15) and LIBRA (16). Though JSmol supports complex visualization options, it suffers from performance issues due to inefficiencies introduced when migrating Jmol code from Java to JavaScript. Fpocket uses OpenAstex (17), another Java based visualizer; however, this project suffers from the same problems as Jmol and now appears to have been discontinued as we were unable to find an active resource. Relatively few of the web servers support visualization via modern WebGL-based viewers, such as LiteMol (18), NGL (19,20) and PV (21). As an example, NGL supports visualizations in DoGSite (22) and DeepSite (23); however, while it is possible to view 3D structures in NGL, the DeepSite and DoGSite websites lack the option to customize protein, ligand and binding site visualizations. Similarly, GalaxySite (24) only offers minimal 3D cartoon visualization of the protein and its ligands via the PV viewer. In response to this situation, we recently developed P2Rank (25), a state-of-the-art method for protein ligand binding site prediction. Here, we describe PrankWeb, an online web server providing an interactive interface for the P2Rank method.

*To whom correspondence should be addressed. Tel: +420 951 554 406; Email: hoksza@ksi.mff.cuni.cz
Present address: Lukas Jendele, Department of Computer Science, ETH Zurich, Switzerland.

Table 1. Availability of web-based tools for structure-based ligand binding site prediction introduced since 2009

Name	Year	Type	Stand-alone	Online Visualization	Offline visualization	Source code
SiteHound (3)	2009	Energetic	Yes	Jmol	PyMOL[b], Chimera[b]	Yes
ConCavity (4)	2009	Conservation	Yes	Jmol	PyMOL	Yes
Fpocket (2)	2010	Geometric	Yes	Jmol, OpenAstex	PyMOL, VMD	Yes
3DLigandSite (11)	2010	Template	—	JSmol	PyMOL	—
POCASA (5)	2010	Geometric	—	Jmol	—	—
DoGSite (22)	2010	Geometric	—	NGL	—	—
MetaPocket 2.0 (6)	2011	Consensus	—	Jmol	PyMOL	—
FTSite (7)	2012	Energetic	—	Jmol, static	PyMOL	—
COFACTOR(12,13)	2012, 2017	Template	Yes	JSmol	—	—
COACH (14)	2013	Template	Yes	JSmol	—	—
eFindSite (27)[a]	2014	Template	Yes	—	PyMOL, VMD, Chimera	Yes
GalaxySite (24)	2014	Template/docking	—	PV, static	—	—
bSiteFinder (8)	2016	Template	—	Jmol	—	—
ISMBLab-LIG (15)	2016	Machine learning	—	JSmol & sequence	—	—
LIBRA-WA (16)	2017	Template	Yes	JSmol	—	—
DeepSite (23)	2017	Machine learning	—	NGL	—	—
PrankWeb (P2Rank)	this work	Machine learning	Yes	LiteMol & Proteal	PyMOL	Yes

[a]In the process of setting up a new interface.
[b]Only data files provided.

PrankWeb serves as an intuitive tool for ligand binding site prediction and its immediate visual analysis by displaying the prediction as a combination of the protein's 3D structure, its sequence and a list of binding pockets. It allows users to display protein ligand binding sites and conservation as structural and sequence views and to customize the visualization style. As PrankWeb's visualization is based on LiteMol and Protael (26), it runs on all modern browsers with no additional plugins.

MATERIALS AND METHODS

P2Rank

P2Rank (25), the backend of PrankWeb, is a template-free, machine learning-based method for ligand binding site prediction employing random forests (28) to predict ligandability of points on the solvent accessible surface of a protein. These points represent potential locations of binding ligand contact atoms and are described by a feature vector calculated from the local geometric neighbourhood. The feature vector consists of physico-chemical and geometric properties calculated from the surrounding atoms and residues (e.g. hydrophobicity, aromaticity or surface protrusion). PrankWeb also introduces a new model that includes information derived from residue sequence evolutionary conservation scores (see Supplementary Information for computation of conservation scores). Points with high predicted ligandability are clustered and ranked according to a ranking function based on the cumulative score of the cluster.

P2Rank is able to use different pre-trained models with varying feature vectors. PrankWeb exposes two such models, the default P2Rank model (without conservation) and a new model that uses conservation information (P2Rank+Conservation). Both models were trained on a relatively small but diverse dataset of protein ligand complexes (25,29).

As a template-free method, P2Rank does not share the limitations of template-based methods that are unable to predict truly novel sites with no analogues in their template libraries of known protein–ligand complexes. As such, P2Rank should be particularly beneficial for predicting novel allosteric sites for which template-based methods are generally less effective (25). Another advantage of P2Rank is its ability to work directly with multi-chain structures and predict binding sites formed near the chain interfaces.

We compared the predictive performance of P2Rank with several competing algorithms using two datasets: COACH420 (14), which contains 420 single-chain complexes, and HOLO4K (25), which contains 4009 multi-chain structures (see Table 2). The default model used by PrankWeb (P2Rank+Conservation) clearly outperformed the other methods, as did the original P2Rank model (without conservation) in most cases. Many of the methods listed in Table 1 are hard to compare using larger datasets as, unlike PrankWeb, they do not expose REST APIs; consequently, batch processing is hindered by slow running times, with results only being deliverable by email or captcha. For a description of the evaluation methodology and more detailed results, see the Supplementary Material. Possible reasons why P2Rank requires less training data and performs better than methods based on more modern machine learning approaches (e.g. DeepSite) are discussed in (25).

Prediction speeds varied greatly between tools, ranging from under one second (Fpocket, P2Rank) to >10 h (COACH) for prediction on one average sized protein (2500 atoms). We have previously shown that P2Rank (without conservation) is the second fastest of the tools presently available (25). While PrankWeb provides little overhead to prediction speed, use of the model with conservation may take a few minutes if conservation scores need to be calculated from scratch (see Conservation pipeline section in the Supplementary Material).

Web server

PrankWeb allows users to predict and visualize the protein ligand binding sites and contrast these with both highly conserved areas and actual ligand binding sites.

Table 2. Benchmark on COACH420 and HOLO4K datasets

	COACH420		HOLO4K	
	Top-*n*	Top-(*n*+2)	Top-*n*	Top-(*n*+2)
Fpocket 1.0	56.4	68.9	52.4	63.1
Fpocket 3.1	42.9	56.9	54.9	64.3
SiteHound[a]	53.0	69.3	50.1	62.1
MetaPocket 2.0[a]	63.4	74.6	57.9	68.6
DeepSite[a]	56.4	63.4	45.6	48.2
P2Rank	72.0	**78.3**	68.6	74.0
P2Rank+Cons.[b]	**73.2**	77.9	**72.1**	**76.7**

Comparing identification success rate [%] measured by the DCA criterion (distance from pocket center to closest ligand atom) with 4 Å threshold considering only pockets ranked at the top of the list (*n* is the number of ligands in the considered structure).
[a]Failed to produce predictions for some of the input proteins. Here we display calculated success rates based only on those protein subsets for which the corresponding method was finished successfully.
[b]P2Rank with conservation (the default prediction model of PrankWeb).

To carry out the prediction, users can either upload a PDB file or provide a PDB ID, in which case PrankWeb will download and store the corresponding PDB file from the PDB database (30). In addition to selecting what protein to analyze, users can also specify whether evolutionary conservation should be included in the prediction process, which in turn determines which of the two pre-trained models will be used.

Conservation scores are calculated using the Jensen-Divergence method (31) from a multiple sequence alignment (MSA) file, which can come from three sources: (i) users can specify their own alignment file, (ii) if a protein's PDB code is provided, PrankWeb uses MSA from the HSSP (32) database or (iii) where no MSA is provided and no MSA is found in HSSP, the MSA is computed using PrankWeb's own conservation pipeline, which utilizes UniProt (33), PSI-Blast (34), MUSCLE (35) and CD-HIT (36). This process is depicted in Figure 2 and described in detail in the Supplementary Material.

After specification of the input, the submitted data is sent via a REST API to the server, which then starts the prediction pipeline. The user is provided with a URL address from which progress of the prediction process can be tracked and results inspected once the process finishes.

On the results page, PrankWeb utilizes LiteMol for visualization of 3D structural information and Protael for sequence visualization. Figure 1 displays the predicted binding sites of dasatinib (a drug used for treatment of chronic myelogenous leukemia) bound to the kinase domain of human LCK (PDB ID 3AD5). The sequence and structure plugins are synchronized so that the user can easily locate a sequence position in the structure and *vice versa*. The sequence view comprises predicted pockets, computed conservation and binding sites (if present in the PDB file). The side panel displays information about the identified pockets and a toolbar allowing the user to (i) download all inputs and calculated results, (ii) share the results page link or (iii) switch between visualization modes. PrankWeb comes with three predefined 3D model renderings (protein surface, cartoon and atoms) and the predicted binding sites and conservation scores are color coded. Conservation is displayed in grayscale (darker denoting more conserved residues) and binding sites are color-highlighted. When the conservation score is not available, the protein surface is white. If conservation analysis is chosen, the user can contrast the positions of putative active sites with conservation scores of the respective positions. In cases where the preset modes do not

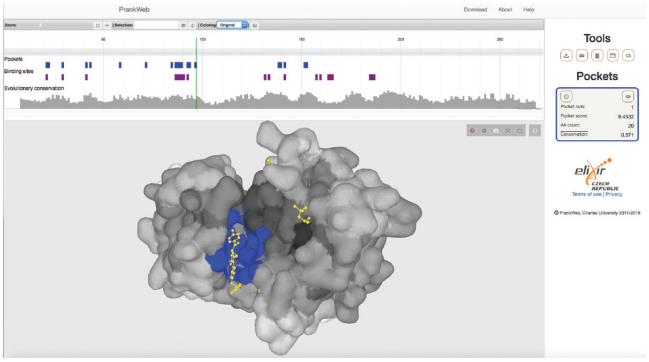


Figure 1. An example of PrankWeb output. The figure shows a predicted ligand binding site (blue colour) on the surface of human Lck kinase (3AD5) The actual ligand binding pose of dasatinib is shown in yellow. The second small molecule in the figure is dimethyl sulfoxide. The figure also shows a sequence view of the protein with binding sites and conservation scores indicated (top panel). The right panel shows a summary of the binding sites and provides tools to modify the view or to download the results.

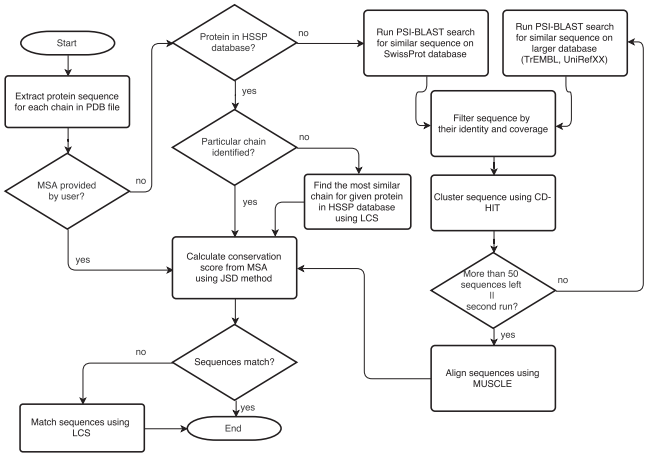


Figure 2. Flow diagram illustrating conservation loading workflow and conservation pipeline.

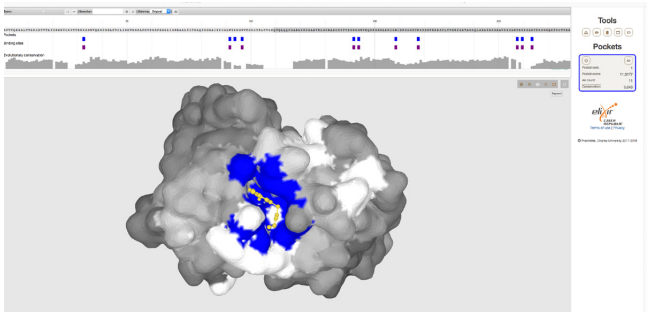


Figure 3. Prediction of a 'difficult' pocket. The authors of the FTSite method describe three structures for which their method failed. This figure shows a PrankWeb prediction for one of these, the structure of mouse immunoglobulin (1a6w). The prediction is indicated by the blue colour and the actual ligand is in yellow.

suffice, one can completely customize the 3D visualization using LiteMol's advanced user interface or the PyMOL visualization script for offline inspection.

PrankWeb consists of a Java backend, REST API and a Typescript frontend, the backend being based on the WildFly (37) web server and the P2Rank application, while the frontend uses the Protael, LiteMol and Bootstrap.js libraries to provide an interactive user interface on top of the REST API. All source code is available under Apache License 2.0 at GitHub (https://github.com/jendelel/PrankWebApp). The GitHub website also includes documentation for developers on how to use our REST API and how to deploy their own version of the server.

DISCUSSION

PrankWeb has been shown to provide correct predictions, even in cases where other methods have failed. Nghan et al. (7) mentions three cases (i.e. the glucose/galactose receptor (1GCG, 1GCA), purine nucleoside phosphorylase (1ULA,1ULB) and mouse FV antibody fragment (1A6U,1A6W)) where their FTSite method was unable to identify a ligand binding site with their best ranked prediction. PrankWeb, on the other hand, correctly identified the binding site as best ranked in both apo and holo structure in all three cases. Figure 3 shows the predicted ligand binding site of the holo structure (1GCA) on the interface of two imunoglobulin subunits, together with the experimentally solved structure of 4-HYDROXY-5-IODO-3-NITROPHENYLACETYL-EPSILON-AMINOCAPROIC ACID ANION (NIP). The 3D structure of NIP appears in the PDB just once, however, which makes it difficult to train its binding.

It should be noted that the current version of PrankWeb is aimed at discovering the binding sites of small biological ligands. None of the models employed by PrankWeb has been trained on other ligand types, such as metallic ion ligands or peptides. Such tasks would be better served by models trained on specialized datasets. We plan to build on our current work by including such models into PrankWeb in the future (38).

CONCLUSION

Here, we present PrankWeb, a new web interface for P2Rank, a state-of-the-art ligand binding prediction method. PrankWeb allows users to quickly carry out predictions and visually inspect the results. PrankWeb also contains a pipeline for computation of conservation scores, which are included in the ligand binding site prediction and the results of structure-sequence visualization. PrankWeb not only provides a user-friendly interface it also serves as a REST API, enabling developers to use PrankWeb as a service. Both PrankWeb and P2Rank are open sourced on GitHub and freely available.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We greatly appreciate access given to computing and storage facilities owned by parties and projects contributing to the MetaCentrum National Grid Infrastructure, as provided under the programme 'Projects of Large Research, Development, and Innovation Infrastructures' (CESNET LM2015042).

FUNDING

This work was supported by the ELIXIR CZ Research Infrastructure Project [MEYS Grant LM2015047] and by the Grant Agency of Charles University [1556217].
Conflict of interest statement. None declared.

REFERENCES

1. Skolnick,J., Fetrow,J.S. and Kolinski,A. (2000) Structural genomics and its importance for gene function analysis. *Nat. Biotechnol.*, **18**, 283.
2. Schmidtke,P., Le Guilloux,V., Maupetit,J. and Tufféry,P. (2010) Fpocket: online tools for protein ensemble pocket detection and tracking. *Nucleic Acids Res.*, **38**, W582–W589.
3. Hernandez,M., Ghersi,D. and Sanchez,R. (2009) SITEHOUND-web: a server for ligand binding site identification in protein structures. *Nucleic Acids Res.*, **37**, W413–W416.
4. Capra,J.A., Laskowski,R.A., Thornton,J.M., Singh,M. and Funkhouser,T.A. (2009) Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3D structure. *PLoS Comput. Biol.*, **5**, e1000585.
5. Yu,J., Zhou,Y., Tanaka,I. and Yao,M. (2010) Roll: a new algorithm for the detection of protein pockets and cavities with a rolling probe sphere. *Bioinformatics*, **26**, 46–52.
6. Zhang,Z., Li,Y., Lin,B., Schroeder,M. and Huang,B. (2011) Identification of cavities on protein surface using multiple computational approaches for drug binding site prediction. *Bioinformatics*, **27**, 2083.
7. Ngan,C.-H., Hall,D.R., Zerbe,B.S., Grove,L.E., Kozakov,D. and Vajda,S. (2012) FTSite: high accuracy detection of ligand binding sites on unbound protein structures. *Bioinformatics*, **28**, 286–287.
8. Gao,J., Zhang,Q., Liu,M., Zhu,L., Wu,D., Cao,Z. and Zhu,R. (2016) bSiteFinder, an improved protein-binding sites prediction server based on structural alignment: more accurate and less time-consuming. *J. Cheminf.*, **8**, 38.
9. Tully,S.P., Stitt,T.M., Caldwell,R.D., Hardock,B.J., Hanson,R.M. and Maslak,P. (2013) Interactive web-based pointillist visualization of hydrogenic orbitals using jmol. *J. Chem. Educ.*, **90**, 129–131.

10. Hanson,R.M., Prilusky,J., Renjian,Z., Nakane,T. and Sussman,J.L. (2013) JSmol and the next-generation web-based representation of 3 D molecular structure as applied to proteopedia. *Isr. J. Chem.*, **53**, 207–216.
11. Wass,M.N., Kelley,L.A. and Sternberg,M.J. (2010) 3DLigandSite: predicting ligand-binding sites using similar structures. *Nucleic Acids Res.*, **38**, W469–W473.
12. Roy,A., Yang,J. and Zhang,Y. (2012) COFACTOR: an accurate comparative algorithm for structure-based protein function annotation. *Nucleic Acids Res.*, **40**, W471–W477.
13. Zhang,C., Freddolino,P.L. and Zhang,Y. (2017) COFACTOR: improved protein function prediction by combining structure, sequence and protein–protein interaction information. *Nucleic Acids Res.*, **45**, W291–W299.
14. Yang,J., Roy,A. and Zhang,Y. (2013) Protein-ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment. *Bioinformatics*, **29**, 2588–2595.
15. Jian,J.-W., Elumalai,P., Pitti,T., Wu,C.Y., Tsai,K.-C., Chang,J.-Y., Peng,H.-P. and Yang,A.-S. (2016) Predicting ligand binding sites on protein surfaces by 3-dimensional probability density distributions of interacting atoms. *PLoS One*, **11**, e0160315.
16. Toti,D., Viet Hung,L., Tortosa,V., Brandi,V. and Polticelli,F. (2017) LIBRA-WA: a web application for ligand binding site detection and protein function recognition. *Bioinformatics*, **34**, 878–880.
17. Hartshorn,M.J. (2002) AstexViewer TM†: a visualisation aid for structure-based drug design. *J. Comput. Aid. Mol. Des.*, **16**, 871–881.
18. Sehnal,D., Deshpande,M., Vareková,R.S., Mir,S., Berka,K., Midlik,A., Pravda,L., Velankar,S. and Koča,J. (2017) LiteMol suite: interactive web-based visualization of large-scale macromolecular structure data. *Nat. Methods*, **14**, 1121–1122.
19. Rose,A.S. and Hildebrand,P.W. (2015) NGL viewer: a web application for molecular visualization. *Nucleic Acids Res.*, **43**, W576.
20. Rose,A.S., Bradley,A.R., Valasatava,Y., Duarte,J.M., Prlić,A. and Rose,P.W. (2016) Web-based molecular graphics for large complexes. In: *Proc. 21st Int. Conf. Web3D Technology*. ACM, NY, pp. 185–186.
21. Biasini,M. (2015) *pv: v1.8.1*. https://biasmv.github.io/pv/.
22. Volkamer,A., Griewel,A., Grombacher,T. and Rarey,M. (2010) Analyzing the topology of active sites: On the prediction of pockets and subpockets. *J. Chem. Inf. Model.*, **50**, 2041–2052.
23. Jiménez,J., Doerr,S., Martínez-Rosell,G., Rose,A.S. and Fabritiis,G.D. (2017) DeepSite: protein-binding site predictor using 3D-convolutional neural networks. *Bioinformatics*, **33**, 3036–3042.
24. Heo,L., Shin,W.-H., Lee,M.S. and Seok,C. (2014) GalaxySite: ligand-binding-site prediction by using molecular docking. *Nucleic Acids Res.*, **42**, W210–W214.
25. Krivák,R. and Hoksza,D. (2018) P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *J. Cheminf.*, **10**, 39.
26. Sedova,M., Jaroszewski,L. and Godzik,A. (2016) Protael: protein data visualization library for the web. *Bioinformatics*, **32**, 602–604.
27. Feinstein,W.P. and Brylinski,M. (2014) eFindSite: Enhanced Fingerprint-Based virtual screening against predicted ligand binding sites in protein models. *Mol. Inf.*, **33**, 135–150.
28. Ho,T.K. (1995) Random decision forests. In: *Proc. 3rd Int. Conf. Document Analysis and Recognition*. IEEE, Vol. **1**, pp. 278–282.
29. Chen,K., Mizianty,M., Gao,J. and Kurgan,L. (2011) A critical comparative assessment of predictions of protein-binding sites for biologically relevant organic compounds. *Structure*, **19**, 613–621.
30. Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) The protein data bank. *Nucleic Acids Res.*, **28**, 235–242.
31. Capra,J.A. and Singh,M. (2007) Predicting functionally important residues from sequence conservation. *Bioinformatics*, **23**, 1875–1882.
32. Joosten,R.P., te Beek,T.A., Krieger,E., Hekkelman,M.L., Hooft,R.W., Schneider,R., Sander,C. and Vriend,G. (2011) A series of PDB related databases for everyday needs. *Nucleic Acids Res.*, **39**, D411.
33. The UniProt Consortium (2018) UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Res.*, **47**, D506–D515.
34. Altschul,S.F., Madden,T.L., Schäffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.
35. Edgar,R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.*, **32**, 1792.
36. Li,W. and Godzik,A. (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, **22**, 1658.
37. WildFly Homepage · WildFly. http://wildfly.org/.
38. Krivák,R., Jendele,L. and Hoksza,D. (2018) Peptide-Binding site prediction from protein structure via points on the solvent accessible surface. In: *Proc. 2019 ACM Int. Conf. Bioinformatics*. ACM, pp. 645–650.

Published online

Nucleic Acids Research, 0000, Vol. 00, No. 00 ***1–3***
doi:10.1093/nar/gkn000

Supplementary material for PrankWeb: a web server for ligand binding site prediction and visualization

Lukas Jendele [1†], Radoslav Krivak [1], Petr Skoda [1], Marian Novotny [2] and David Hoksza [1,3*]

[1]Department of Software Engineering, Faculty of Mathematics and Physics, Charles University, Czech Republic
[2]Department of Cell Biology, Faculty of Science, Charles University, Czech Republic
[3]Luxembourg Centre for Systems Biomedicine, University of Luxembourg, Luxembourg

CONSERVATION PIPELINE

Conservation scores for PrankWeb are computed from multiple sequence alignment (MSA). MSA for a particular sequence can be acquired from an HSSP database (1) or calculated from a set of sequences using bioinformatics tools. Moreover, PrankWeb also allows users to upload their own MSA for each chain.

If the HSSP database contains the protein of interest and no chain for the particular ID was found, PrankWeb takes the chain with the longest common subsequence. In case the protein is not present in HSSP and the user did not provide the MSA for that protein, homology pipeline is invoked to obtain an MSA. The main idea of the pipeline (inspired by ConSurfDB (2)) is based on querying databases for similar sequences to the input sequence. The decision making process for calculating conservation scores is illustrated in Figure 2 of the main article. It takes a protein sequence in FASTA format as input and outputs a tab-separated file with conservation scores, which is the result of the Jensen-Shannon divergence method for calculating the conservation scores from multiple sequence alignment. (3)

The pipeline proceeds as follows:

1. SwissProt is queried for similar protein sequences using PSI-BLAST (4) with e-value=10^{-5}. ConSurfDB uses the same e-value.

2. The sequences that are too similar or too different than our query sequence are filtered out.

3. Then CD-HIT (5) is run with default parameters to cluster the sequences and outputs a non-redundant representative sequence list.

4. If less than 50 sequences are left, we repeat the steps 1–3 on, the larger database, UniRef90 (6).

5. Sequences are aligned using MUSCLE (7).

6. At this point, we have a multiple sequence alignment and can calculate the conservation score using the Jensen-Shannon divergence method (3).

EVALUATION METHODOLOGY

To evaluate predictive performance of PrankWeb we have used the same methodology that was used in original P2Rank article (8). It is based on ligand-centric counting and the DCA (distance between the center of the pocket and any ligand atom) pocket identification criterion with 4 Å threshold. Ground-truth binding sites are defined by ligands present in evaluation datasets. Every structure in a dataset can contain more than one relevant ligand (see below) and for every relevant ligand, its binding site must be correctly predicted for a method to achieve 100% identification success rate on the given dataset. Every relevant ligand contributes with equal weight toward the final success rate. The output of prediction methods is a ranked list of several putative binding sites, but during evaluation only those ranked at the top are considered. We use Top-n and Top-(n+2) rank cutoffs where for every evaluated protein structure n is the number of relevant ligands in this structure (i.e. for proteins that have only one ligand this corresponds to the usual Top-1 and Top-3 cutoffs and for proteins with 2 ligands to Top-2 and Top-4 cutoffs). This evaluation methodology is the same as the one that was used in the only independent benchmark of ligand binding site prediction algorithms to date (9).

Relevant Ligands

P2Rank is focused on predicting binding sites for biologically relevant ligands and PDB files in considered datasets often contain ligands (i.e. HET groups) that are not relevant. To determine which ligands in benchmark datasets are relevant we use a custom filter and alternatively the binding MOAD (10) database.

In addition to biologically relevant ligands, PDB files contain a variety of other HET groups like solvents, salt and misplaced groups (that are not in contact with the protein). Instead of declaring only one ligand as relevant for every file in a dataset (as was done in other ligand binding site prediction studies), we determine relevant ligands by a filter. Ligands that are considered relevant must comply to these conditions:

- Number of ligand atoms is greater or equal than 5.

*Correspondence should be addressed to D. Hoksza. Tel: +420 951 554 406; Email: hoksza@ksi.mff.cuni.cz
† Current address: Lukas Jendele, Department of Computer Science, ETH Zurich, Switzerland.

Table 1. Benchmark on COACH420, COACH420(Mlig), HOLO4K and HOLO4K(Mlig) datasets.

	COACH420		COACH420(Mlig)		HOLO4K		HOLO4K(Mlig)	
	Top-n	Top-(n+2)	Top-n	Top-(n+2)	Top-n	Top-(n+2)	Top-n	Top-(n+2)
Fpocket 1.0	56.4	68.9	57.4	70.4	52.4	63.1	56.9	70.3
Fpocket 3.1	42.9	56.9	43.1	56.3	54.9	64.3	57.4	69.1
SiteHound*	53.0	69.3	51.0	67.7	50.1	62.1	53.1	67.8
MetaPocket 2.0*	63.4	74.6	62.2	73.3	57.9	68.6	62.3	75.2
DeepSite*	56.4	63.4	54.5	61.6	45.6	48.2	50.8	54.4
P2Rank	72.0	**78.3**	**71.2**	**76.5**	68.6	74.0	73.7	80.9
P2Rank+Conservation†	**73.2**	77.9	70.9	75.1	**72.1**	**76.7**	**77.2**	**83.3**

Comparing identification success rate [%] measured by the DCA criterion (distance from pocket center to closest ligand atom) with 4 Å threshold considering only pockets ranked at the top of the list (n is the number of ligands in the considered structure).
*Failed to produce predictions for some of the input proteins. Here we display success rates calculated only based on subsets of proteins, on which corresponding methods finished successfully. Detailed, pairwise comparison with P2Rank on the exact subsets can be found in the Supplementary Information of P2Rank article (8).
† P2Rank with conservation (the default prediction model of PrankWeb)

- Distance from any atom of the ligand to the closest protein atom is at least 4 Å (to remove "floating" HET groups present in some structures).
- Distance form the center of the mass of the ligand to the closest protein atom is not greater than 5.5 Å (to remove ligands that "stick out").
- Name of the PDB group is not on the list of ignored groups:
  `(HOH, DOD, WAT, NAG, MAN, UNK, GLC, ABA, MPD, GOL, SO4, PO4).`

Choosing relevant ligands in this particular way is admittedly arbitrary. In order to make sure our results are robust with respect to the exact way relevant ligands are determined, we have created a versions of COACH420 and HOLO4K datasets where relevant ligands are determined in a different way. Binding MOAD (10) release 2013, a database of biologically relevant ligands in PDB, was used to determine relevant ligands in resulting datasets COACH420(Mlig) and HOLO4K(Mlig). PDB files that have no entry in MOAD were removed from the new datasets.

It should be noted that the notion of a biologically relevant ligand does not have a widely accepted definition. There are other databases that purportedly collect only biologically relevant ligand interactions from the PDB (e.g. BioLiP (11), PDBbind (12)) that use different criteria for accepting particular ligand as biologically relevant (with MOAD being the strictest of them, for example, by not accepting any small ions). For a discussion on the caveats of determining biologically relevant ligands see (11).

Datasets

All datasets used to train and optimize our models and produce presented results are available on GitHub http://github.com/rdk/p2rank-datasets and described in detail in P2Rank paper (8).

P2Rank was trained on the CHEN11 dataset (both models employed by PrankWeb: with and without conservation) and various parameters of the algorithm were optimized with respect to the results on the JOINED dataset (8), that was used as a development/validation dataset. For future benchmarks we note that results on proteins from those datasets would not represent an unbiased estimate of P2Rank's performance.

ADDITIONAL RESULTS

Table 1 is an extended version of the results table from the main article which includes results on *(Mlig) versions of datasets where relevant ligands were determined differently (see Relevant Ligands section). It shows that our results are robust with respect to the particular way relevant ligands are determined. New P2Rank model with conservation seems to perform slightly worse on COACH420 dataset but substantially better on larger HOLO4K dataset. Table 2 shows average numbers of predicted sites for each method. P2Rank+Conservation in general predicts fewer but more relevant sites than the original P2Rank model.

The results were taken from (8) and we performed new benchmark experiments for Fpocket 3.1 and P2Rank+Conservation. Results of Fpocket 3.1 correspond to the 3.1.2 version downloaded and compiled from GitHub (https://github.com/Discngine/fpocket), run with default parameters.

Table 2. Number of predicted binding sites and dataset statistics.

	COACH420	HOLO4K
Proteins	420	4009
Avg. protein atoms	2179	3908
Avg. ligands	1.2	2.4
Fpocket 1.0	14.6	27.0
Fpocket 3.1	13.9	16.0
SiteHound	66.2	99.5
MetaPocket 2.0	6.3	6.4
DeepSite	3.2	2.8
P2Rank	6.3	12.6
P2Rank+Conservation	3.4	7.7

Displayed is the average total number of binding sites predicted per protein by each method on a given dataset.

REFERENCES

1. Joosten, R. P., te Beek, T. A., Krieger, E., Hekkelman, M. L., Hooft, R. W., Schneider, R., Sander, C., and Vriend, G. (2011) A series of PDB related databases for everyday needs. Nucleic Acids Research, **39**(suppl_1), D411 [PubMed:21071423] [PubMed Central:PMC3013697] [doi:10.1093/nar/gkq1105].
2. Ashkenazy, H., Erez, E., Martz, E., Pupko, T., and Ben-Tal, N. (2010) ConSurf 2010: calculating evolutionary conservation in sequence and structure of proteins and nucleic acids. Nucleic Acids Research, **38**(Web Server), W529–W533 [PubMed:20478830] [PubMed Central:PMC2896094] [doi:10.1093/nar/gkq399].
3. Capra, J. A. and Singh, M. (2007) Predicting functionally important residues from sequence conservation. Bioinformatics, **23**(15), 1875–1882 [PubMed:17519246] [doi:10.1093/bioinformatics/btm270].
4. Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.. Nucleic Acids Res., **25**, 3389–3402 [PubMed:9254694] [PubMed Central:PMC146917] [doi:10.1093/nar/25.17.3389].
5. Li, W. and Godzik, A. (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics, **22**(13), 1658 [PubMed:16731699] [doi:10.1093/bioinformatics/btl158].
6. Suzek, B. E., Huang, H., McGarvey, P., Mazumder, R., and Wu, C. H. (2007) UniRef: comprehensive and non-redundant UniProt reference clusters. Bioinformatics, **23**(10), 1282–1288 [PubMed:17379688] [doi:10.1093/bioinformatics/btm098].
7. Edgar, R. C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res., **32**(5), 1792 [PubMed:15034147] [PubMed Central:PMC390337] [doi:10.1093/nar/gkh340].
8. Krivák, R. and Hoksza, D. (2018) P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. J. Cheminf., **10**(1), 39 [PubMed:30109435] [PubMed Central:PMC6091426] [doi:10.1186/s13321-018-0285-8].
9. Chen, K., Mizianty, M., Gao, J., and Kurgan, L. (2011) A critical comparative assessment of predictions of protein-binding sites for biologically relevant organic compounds.. Structure, **19**(5), 613–621 [PubMed:21565696] [doi:10.1016/j.str.2011.02.015].
10. Hu, L., Benson, M. L., Smith, R. D., Lerner, M. G., and Carlson, H. A. (2005) Binding MOAD (Mother Of All Databases). Proteins, **60**(3), 333–340 [PubMed:15971202] [doi:10.1002/prot.20512].
11. Yang, J., Roy, A., and Zhang, Y. (2013) BioLiP: a semi-manually curated database for biologically relevant ligandprotein interactions. Nucleic Acids Res., **41**(D1), D1096–D1103 [PubMed:23087378] [PubMed Central:PMC3531193] [doi:10.1093/nar/gks966].
12. Wang, R., Fang, X., Lu, Y., and Wang, S. (2004) The PDBbind database: Collection of binding affinities for protein-ligand complexes with known three-dimensional structures. J. Med. Chem., **47**(12), 2977–2980 [PubMed:15163179] [doi:10.1021/jm030580l].

PrankWeb 3: accelerated ligand-binding site predictions for experimental and modelled protein structures

Reference

JAKUBEC D., SKODA P., KRIVAK R., NOVOTNY M., HOKSZA D.: **PrankWeb 3: accelerated ligand-binding site predictions for experimental and modelled protein structures**. *Nucleic Acids Research 50*, W1 (05 2022), W593–W597. doi:10.1093/nar/gkac389

Author's highlights

Complete rewrite of PrankWeb as a modern modular Python web application. The conservation pipeline was completely redesigned. The new version is more efficient and consistent with regard to the required time for calculation for any single sequence. Four new prediction models were trained.

Published online 24 May 2022 *Nucleic Acids Research, 2022, Vol. 50, Web Server issue* ***W593–W597***
https://doi.org/10.1093/nar/gkac389

PrankWeb 3: accelerated ligand-binding site predictions for experimental and modelled protein structures

David Jakubec[1,†], Petr Skoda[1,†], Radoslav Krivak[1], Marian Novotny[2] and David Hoksza[1,*]

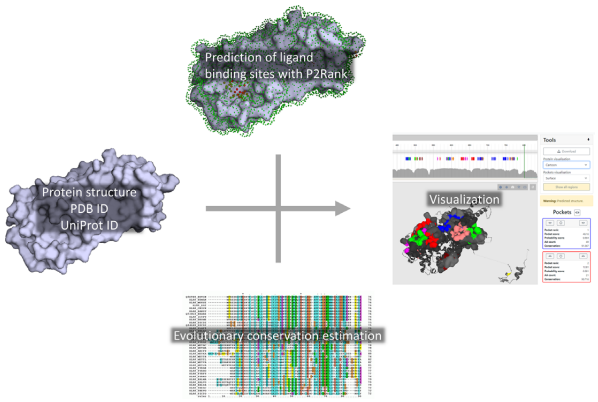
[1]Department of Software Engineering, Faculty of Mathematics and Physics, Charles University, Czech Republic and [2]Department of Cell Biology, Faculty of Science, Charles University, Czech Republic

Received March 25, 2022; Revised April 15, 2022; Editorial Decision April 27, 2022; Accepted May 06, 2022

ABSTRACT

Knowledge of protein–ligand binding sites (LBSs) enables research ranging from protein function annotation to structure-based drug design. To this end, we have previously developed a stand-alone tool, P2Rank, and the web server PrankWeb (https://prankweb.cz/) for fast and accurate LBS prediction. Here, we present significant enhancements to PrankWeb. First, a new, more accurate evolutionary conservation estimation pipeline based on the UniRef50 sequence database and the HMMER3 package is introduced. Second, PrankWeb now allows users to enter UniProt ID to carry out LBS predictions in situations where no experimental structure is available by utilizing the AlphaFold model database. Additionally, a range of minor improvements has been implemented. These include the ability to deploy PrankWeb and P2Rank as Docker containers, support for the mmCIF file format, improved public REST API access, or the ability to batch download the LBS predictions for the whole PDB archive and parts of the AlphaFold database.

GRAPHICAL ABSTRACT



INTRODUCTION

Interactions of proteins with other molecules drive biological processes at the molecular level. One specific class of such interactions are protein–small molecule (ligand) interactions; identifying the sites and roles of these interactions is crucial for the elucidation of the molecular mechanisms of enzymes, regulation of protein oligomerization, or designing new drugs (e.g., in case drug resistance has occurred) (1,2). In these applications, precise knowledge of the protein's ligand-binding sites (LBSs) is required. As experimental identification of LBSs is time-consuming and expensive, computational methods have been developed to facilitate LBS identification from the protein three-dimensional (3D) structure. These methods can be broadly categorized as geometric, energetic, evolution-based, and knowledge- or machine learning (ML)-based. Many of the existing methods combine the aforementioned approaches, which is also the case of the P2Rank method (3) developed in our group. P2Rank assigns structural, physico-chemical, and evolutionary features to points on a mesh covering the protein surface and builds an ML model over this representation. The model is used to detect ligandable points, which

*To whom correspondence should be addressed. Tel: +420 951 554 227; Email: david.hoksza@matfyz.cuni.cz
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

are then clustered to obtain a list of surface patches corresponding to the predicted LBSs. The approach has achieved state-of-the-art performance and is still on par or outperforming newer deep learning methods (4).

The lack of broadly accessible online resources has historically hindered access to the LBS prediction methods. To this end, we have developed PrankWeb (5), an online tool encapsulating the P2Rank approach. PrankWeb has allowed its users to enter a 3D structure as a Protein Data Bank (6) (PDB) file or using a PDB identifier, carried out evolutionary conservation analysis, predicted the LBSs using P2Rank, and enabled visual examination of the results. This paper introduces PrankWeb 3, an improved version of the resource.

A limiting aspect of the structure-based LBS prediction approaches is the necessity of having the protein 3D structure determined. Although the number of resolved protein structures keeps increasing, it is still far behind the number of known protein sequences (7). However, recent advances in protein structure prediction, namely the introduction of the AlphaFold 2 method (8) and the AlphaFold Protein Structure Database (AlphaFold DB) (9), have opened the door for the application of structure-based approaches also toward proteins for which only the sequence is known. This development has motivated one of the major improvements in PrankWeb 3: the adoption of the AlphaFold DB, allowing PrankWeb users to enter a UniProt accession number as the input. This change significantly increases the number of proteins to which PrankWeb is applicable (section *Predicted structures*). Another significant improvement is the replacement of the former evolutionary conservation estimation pipeline with a faster, more consistent version (section *Evolutionary conservation calculation pipeline*). The last major change has been the refactoring of the PrankWeb application resulting in a modular architecture with strictly separated components. Such architecture enables easy utilization of the application or its parts (such as the conservation calculation pipeline) to advanced users via Docker containers (section *Other improvements*). A detailed description of the changes follows.

EVOLUTIONARY CONSERVATION CALCULATION PIPELINE

Evolutionary conservation (EC) has been identified as a powerful indicator of functionally significant regions of protein structures; for this reason, it has been utilized as an optional feature capable of improving the default P2Rank predictions. Previous versions of PrankWeb utilized a series of sequence databases to construct a multiple sequence alignment (MSA) of sequences similar to the given query, and subsequently quantified the EC of its individual columns using Jensen–Shannon divergence (10). This approach possessed two major drawbacks. First, the use of fallback sequence databases for the construction of an MSA of sufficient size resulted in discontinuities in the conservation scores as the number of sequences in the MSA exceeded the threshold. A single P2Rank model was thus unable to account for the different sequence distributions (and, therefore, conservation scores) intrinsic to the individual sequence databases. Second, and more importantly, the previous EC calculation pipeline could take several hours to complete, severely impacting the user's experience with PrankWeb.

Table 1. The runtimes of the new EC calculation pipeline (in seconds) measured on the datasets used for the training (CHEN11), validation (JOINED), and testing (COACH420 and HOLO4K) of P2Rank models. The computations were performed on a desktop computer running Ubuntu 20.04, HMMER v3.3.2, and using the i7-3770K processor. The numbers in parentheses indicate the number of polypeptide chains in the respective datasets. See the original P2Rank publication (3) for a detailed description of the datasets

	CHEN11 (251)	JOINED (643)	COACH420 (420)	HOLO4K (8588)
Runtime (50th percentile; s)	107	109	108	127
Runtime (95th percentile; s)	139	244	193	324

Starting with PrankWeb 3, the former EC calculation pipeline has been replaced with a simpler, faster, and more consistent one inspired by the recent Amino Acid Interactions web server v2.0 (11). The new pipeline operates as follows. First, polypeptide chain sequences are extracted from the input file using P2Rank. The *phmmer* tool from the HMMER software package (http://hmmer.org/) is then used to identify and align similar sequences for each respective query; UniRef50 Release 2021_03 (12) is used as the single target sequence database. Up to 1000 sequences are then randomly selected from each MSA to form the respective sample MSAs; weights are assigned to the individual sequences constituting the sample MSAs using the Gerstein/Sonnhammer/Chothia algorithm (13) implemented in the *esl-weight* miniapp included with the HMMER software. Finally, per-column information content (i.e. conservation score) and gap character frequency values are calculated using the *esl-alistat* miniapp, taking the individual sequence weights into account; positions containing the gap character in $>50\,\%$ of sequences are masked to appear as possessing no conservation at all. The pipeline utilizes a fixed seed value for any random selection, making the output deterministic for a given query.

Table 1 shows the runtimes of the new EC calculation pipeline measured on the datasets used for the training, validation, and testing of P2Rank models. It can be seen that for 50% of queries, the EC calculation pipeline (which constitutes most of the time required for PrankWeb predictions) finishes in about 2 min, while nearly all queries finish within 5 min. In comparison, for the previous EC conservation pipeline on the CHEN11 dataset, the median of runtimes was 275 s (4.6 min) while 95th percentile was 854 s (14.2 min).

The adoption of the new EC calculation pipeline necessitated the preparation of a new EC-aware P2Rank model. Table 2 presents the evaluation of all the new P2Rank models prepared for PrankWeb 3, as well as their comparison with the former models; it can be seen that the new Default models exceed the performance of the corresponding old models when evaluated on the representative HOLO4K dataset.

Table 2. Identification success rates (in %) measured using the DCA criterion utilizing a 4.0 Å threshold for the distance between the center of the predicted LBS and any ligand atom; only the *n* or (*n* + 2), respectively, top-ranking predicted LBSs are considered in the evaluation, where *n* is the number of ligands in the respective 3D structure. Values for Default (old) and Default + conservation (old) are taken from the original PrankWeb publication (5) and are shown only for comparison, as these models are no longer used. B-factor-free are used with AlphaFold predictions which utilize the B-factor field for confidence scores. Please note that old models were generated by the older version of P2Rank, which used older versions of BioJava and CDK. Using newer versions changed how certain PDB files are parsed, and an upgrade of the CDK library fixed a bug in the algorithm that generates SAS points. This, together with bug fixes in P2Rank itself, causes the scores for the Default (old) and Default models to differ

	COACH420		HOLO4K	
	Top-*n*	Top-(*n* + 2)	Top-*n*	Top-(*n* + 2)
Default (old)	72.0	78.3	68.6	74.0
Default + conservation (old)	73.2	77.9	72.1	76.7
Default	71.6	76.8	72.7	78.0
Default + conservation	74.3	77.2	74.5	78.4
B-factor-free	71.2	77.5	72.1	77.2
B-factor-free + conservation	74.9	78.5	73.9	77.7

PREDICTED STRUCTURES

The AlphaFold DB (9) is a freely and openly accessible resource housing 3D structure models for a selection of biomedically significant proteins predicted using AlphaFold 2 (8). In PrankWeb 3, we have precomputed the P2Rank LBS predictions for two components of the AlphaFold DB—the 'model organism proteomes' and 'Swiss-Prot'—totalling over 800 000 proteins. As the AlphaFold 3D structure models utilize the *B*-factor fields of the structure files to store the per-residue confidence scores, computing these LBS predictions necessitated the preparation of two additional, *B*-factor field-agnostic P2Rank models (Table 2); it can be seen that the performance of these on the representative HOLO4K dataset (consisting of experimentally resolved 3D structures) is only marginally worse compared to the models utilizing *B*-factor as a feature.

To show how PrankWeb can be used to predict and visualize binding sites for predicted structures, we chose a protein from the G protein-coupled receptors (GPCR) family. The GPCR family is not only the largest protein family (with over 800 members), but also a family with >160 validated drug targets. GPCRs are membrane proteins and as such have represented a major challenge for structural biology. Advances in cryoEM methodology have brought a revolution in our understanding of intricate differences among GPCR proteins with more than 450 structures of over 80 proteins (14) solved so far, but many proteins indicated in human disease are still without an experimentally solved structure. The availability of high-quality 3D structure models in the AlphaFold DB, however, massively expands the number of proteins that can be investigated with PrankWeb. We used PrankWeb to show predicted binding sites on the AlphaFold model of succinate receptor 1 (uniprot code Q9BXA5), a protein suspected as a major player in the development of kidney hypertension and possibly also metabolic syndrome and thus potential drug target (15) without known experimentally solved 3D structure. The structure submission interface of PrankWeb has been extended to enable fetching predicted structures from the AlphaFold DB via the UniProt accession. After the accession is entered, the structure is downloaded from the AlphaFold DB (if not cached) and binding sites are predicted with P2Rank. Once the results are available, they are visualized in the PrankWeb interface. For AlphaFold predictions, the structure is color-coded by the confidence score. Moreover, PrankWeb enables visualization of only high-confidence regions (pLDDT > 70).

The results for the succinate receptor 1 are shown in Figure 1. Figure 1 A displays the best predicted pocket in blue on top. As the experimental structure with, or even without a ligand, is not known, the predicted structure was aligned using PyMOL with the structure of a closely related P2Y12 receptor (PDB ID 4NTJ (16)). The structural alignment (Figure 1B) shows that the best predicted succinate receptor binding pocket is different from ligand binding pocket of P2Y12 receptor as expected due to different properties and size of these ligands, although we can not be completely sure that the predicted binding site is correct as there is no experimentally solved structure of this receptor. This shows that using AlphaFold models for prediction of binding sites provides information that can not be extracted from experimentally solved structures of closely related proteins.

OTHER IMPROVEMENTS

Additional updates focus on improving the user experience and usability. The updates range from small quality of life improvements to complete redesign of the PrankWeb architecture.

The most noticeable change is in the results visualization page (Figure 2). First, the user can now select a visualization mode for the inspected protein and the predicted binding sites. The modes available are surface, cartoon, and balls and sticks. Second, when a pocket prediction is carried out on a predicted structure, the user can hide low-confident regions, i.e. regions with pLDDT score <70. Finally, the protein surface is colored by conservation score for the experimental structures, and by residue-level confidence scores for the predicted structures.

Another addition to the results visualization page is the pocket's probability score. By default, the pockets are sorted using the P2Rank's raw pocket score. However, as this value is not bound, it is hard to interpret by a user. To tackle this we added the pocket's probability score that has a clearly defined maximum value and thus should provide easier interpretation to a user. The pocket probability score is calculated as a monotonous transformation of a raw pocket score to the interval [0,1]. The transformation is calibrated for each model on the HOLO4K dataset in such a way that the probability score represents a ratio of true binding sites among all predicted sites with a comparable raw score.

We have also updated the HTTP-based API to v2, indicating breaking changes. The core idea was to shift the API closer to the REST ideas. The change allows users to easily create new prediction tasks for custom structures using POST. GET requests can be used to retrieve prediction

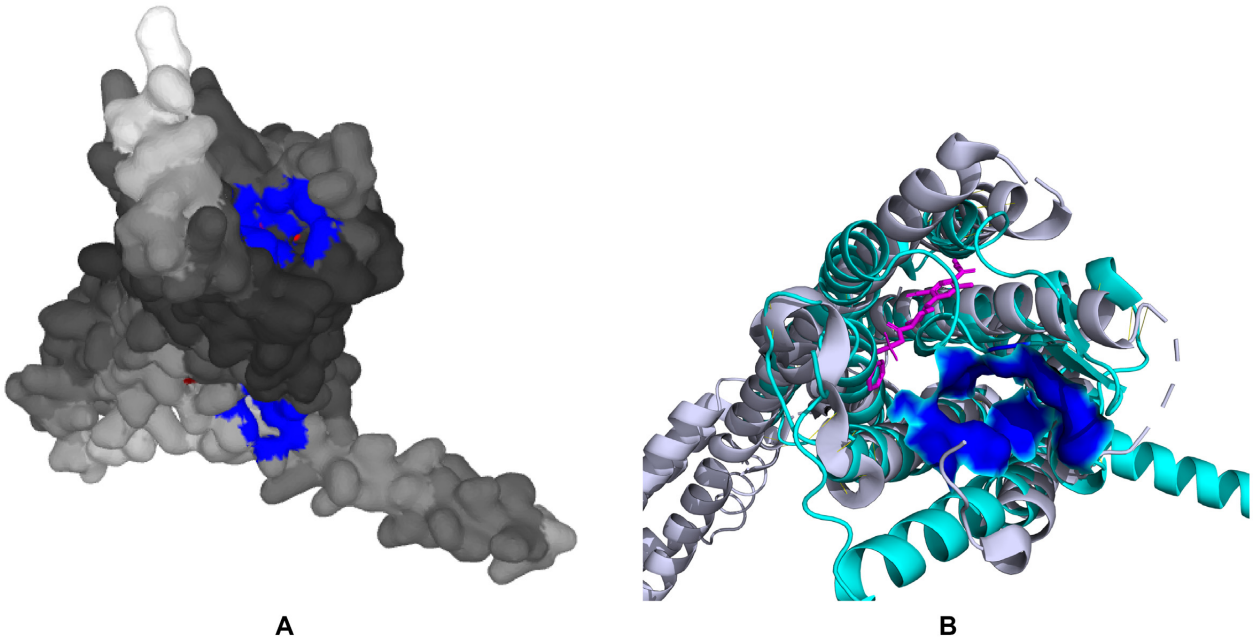


Figure 1. P2Rank prediction on an AlphaFold model of human succinate receptor (Q9BXA5). (**A**) Visualization of the pockets from PrankWeb (available at https://prankweb.cz/analyze?database=v3-alphafold&code=Q9BXA5). The main pocket is in blue on the top of the structure. The structure is colored-coded by AlphaFold confidence (darker being more confident). (**B**) The predicted succinate receptor structure (in cyan) is aligned with closely related P2Y receptor (in grey, PDB ID 4NTJ) and its ligand (in magenta). The best binding pocket predicted for succinate receptor is shown in blue and is clearly outside of the binding pocket of P2Y receptor (visualized with PyMOL, http://www.pymol.org/pymol).

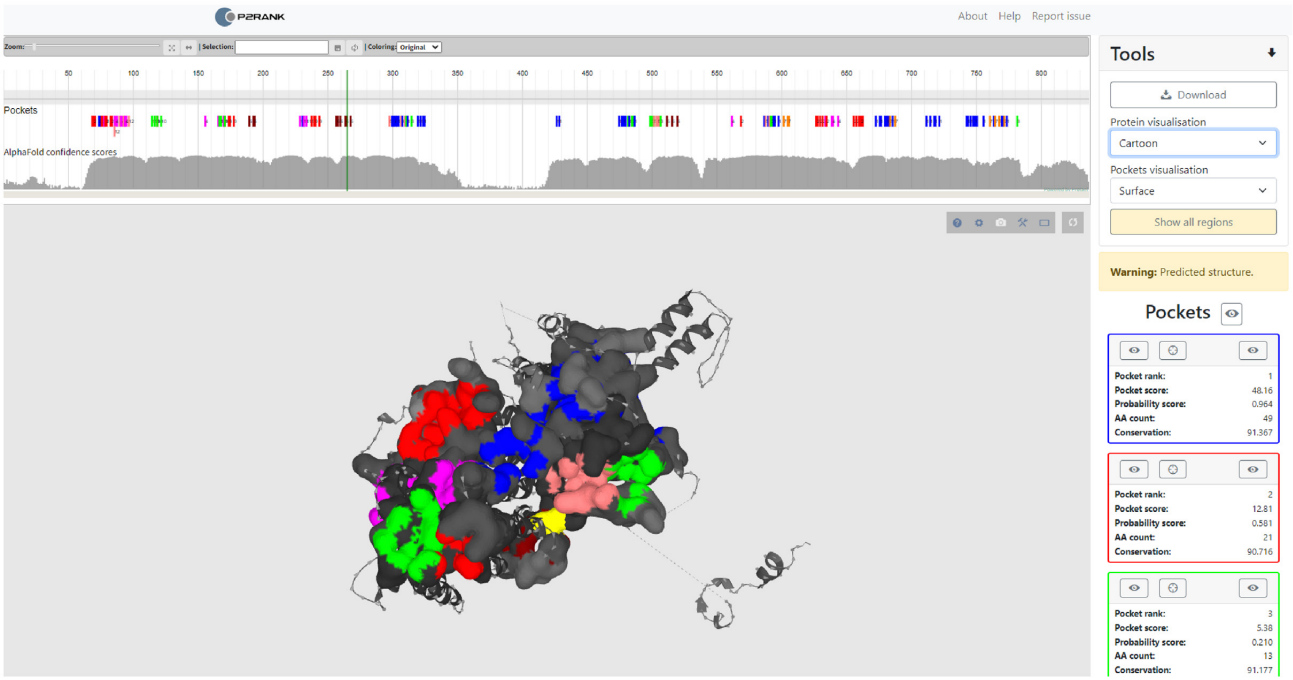


Figure 2. PrankWeb results visualization page. The view shows predicted LBSs on the AlphaFold model of the human striatin-interacting protein (Q5VSL9), available at https://prankweb.cz/analyze?database=v3-alphafold&code=Q5VSL9. Pockets are displayed using surface visualization while the rest of hte structure is shown as cartoon. Different putative pockets are distinguished by color. The parts of the structure which are not part of any pocket are color-coded by tha AlphaFold confidence score, with darker regions being more confident. Finally, the visualization shows only high-confident parts of the structure (pLDDT score > 70) which are connected by dotted lines. Switching between full structure and confident regions only can be controlled by the user.

status, log, structure or prediction archive. The prediction archive can be also downloaded from the user interface and contains visualizations of the protein in PyMOL, parameters used to run P2rank, prediction log file and information about the predicted pockets in the CSV format. In addition, the archive can contain conservation scores if the user has chosen to use conservation in the prediction.

We also added links to the pre-computed predictions described in the section *Evolutionary conservation calculation pipeline*. Users can thus download all predictions computed for PDB and AlphaFold. For each database, we provide predictions computed with and without the use of conservation. The archive has similar content to the archive for a single prediction, the main difference is in the structure as the archives house multiple predictions.

Another modification in PrankWeb 3 is added support for the mmCIF format as the structure definition format. This was necessary as the PDB format has been deprecated due to its limitations.

Finally, under the hood, PrankWeb's architecture has been completely redesigned. The new modular architecture strictly separates web-based user interface, data storage, and an execution component. The execution component is responsible for running the predictions from start to end. Starting with a protein file or UniProt ID, it will compute conservation and produce pocket predictions. Each component corresponds to a Docker image. Combined with docker-compose, it is easy to deploy and update PrankWeb instances. Thanks to the modular architecture, users can deploy only the execution component, using Docker, on their hardware. As a result, it is possible to run predictions on private data without exposing them to third-party servers. Another advantage is that such deployment allows users to run as many predictions as their computation resources allow. On the other hand, we are aware that not every user has the capacity to run the predictions on a large scale database such as PDB and parts of the AlphaFold.

DATA AVAILABILITY

The PrankWeb web server is publicly available at https://prankweb.cz/. The source codes are available at https://github.com/cusbg/p2rank-framework.

ACKNOWLEDGEMENTS

Computational resources were supplied by the project 'e-Infrastruktura CZ' (e-INFRA CZ LM2018140) supported by the Ministry of Education, Youth and Sports of the Czech Republic. This work was also carried out with the support of the Charles University grant SVV-260588 and the ELIXIR CZ Research Infrastructure (ID LM2018131, MEYS CR), including access to the computational resources.

FUNDING

Funding for open access charge: ELIXIR CZ Research Infrastructure (ID LM2018131, MEYS CR).
Conflict of interest statement. None declared.

REFERENCES

1. Konc,J., Lešnik,S. and Janežič,D. (2015) Modeling enzyme-ligand binding in drug discovery. *J. Cheminform.*, **7**, 48.
2. Imamura,A., Okada,T., Mase,H., Otani,T., Kobayashi,T., Tamura,M., Kubata,B.K., Inoue,K., Rambo,R.P., Uchiyama,S. *et al.* (2020) Allosteric regulation accompanied by oligomeric state changes of Trypanosoma brucei GMP reductase through cystathionine-β-synthase domain. *Nat. Commun.*, **11**, 1837.
3. Krivák,R. and Hoksza,D. (2018) P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *J. Cheminf.*, **10**, 39.
4. Mylonas,S.K., Axenopoulos,A. and Daras,P. (2021) DeepSurf: a surface-based deep learning approach for the prediction of ligand binding sites on proteins. *Bioinformatics*, **37**, 1681–1690.
5. Jendele,L., Krivak,R., Skoda,P., Novotny,M. and Hoksza,D. (2019) PrankWeb: a web server for ligand binding site prediction and visualization. *Nucleic Acids Res.*, **47**, W345–W349.
6. Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) The protein data bank. *Nucleic Acids Res.*, **28**, 235–242.
7. Mitchell,A.L., Almeida,A., Beracochea,M., Boland,M., Burgin,J., Cochrane,G., Crusoe,M.R., Kale,V., Potter,S.C., Richardson,L.J. *et al.* (2020) MGnify: the microbiome analysis resource in 2020. *Nucleic Acids Res.*, **48**, D570–D578.
8. Jumper,J., Evans,R., Pritzel,A., Green,T., Figurnov,M., Ronneberger,O., Tunyasuvunakool,K., Bates,R., Žídek,A., Potapenko,A. *et al.* (2021) Highly accurate protein structure prediction with AlphaFold. *Nature*, **596**, 583–589.
9. Varadi,M., Anyango,S., Deshpande,M., Nair,S., Natassia,C., Yordanova,G., Yuan,D., Stroe,O., Wood,G., Laydon,A. *et al.* (2022) AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Res.*, **50**, D439–D444.
10. Capra,J.A. and Singh,M. (2007) Predicting functionally important residues from sequence conservation. *Bioinformatics*, **23**, 1875–1882.
11. Vymětal,J., Jakubec,D., Galgonek,J. and Vondrášek,J. (2021) Amino Acid Interactions (INTAA) web server v2.0: a single service for computation of energetics and conservation in biomolecular 3D structures. *Nucleic Acids Res.*, **49**, W15–W20.
12. Suzek,B.E., Wang,Y., Huang,H., McGarvey,P.B. and Wu,C.H. (2015) UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, **31**, 926–932.
13. Gerstein,M., Sonnhammer,E.L. and Chothia,C. (1994) Volume changes in protein evolution. *J. Mol. Biol.*, **236**, 1067–1078.
14. Yang,D., Zhou,Q., Labroska,V., Qin,S., Darbalaei,S., Wu,Y., Yuliantie,E., Xie,L., Tao,H., Cheng,J. *et al.* (2021) G protein-coupled receptors: structure- and function-based drug discovery. *Signal Transduct. Target Ther.*, **6**, 7.
15. Ariza,A.C., Deen,P.M. and Robben,J.H. (2012) The succinate receptor as a novel therapeutic target for oxidative and metabolic stress-related conditions. *Front. Endocrinol. (Lausanne)*, **3**, 22.
16. Zhang,K., Zhang,J., Gao,Z.G., Zhang,D., Zhu,L., Han,G.W., Moss,S.M., Paoletta,S., Kiselev,E., Lu,W. *et al.* (2014) Structure of the human P2Y12 receptor in complex with an antithrombotic drug. *Nature*, **509**, 115–118.

PDBe-KB: a community-driven resource for structural and functional annotations

Reference

CONSORTIUM P.-K.: **PDBe-KB: a community-driven resource for structural and functional annotations**. *Nucleic Acids Research 48*, D1 (10 2019), D344–D353. doi:10.1093/nar/gkz853

Abstract

The Protein Data Bank in Europe-Knowledge Base (PDBe-KB, https://pdbe-kb.org) is a community-driven, collaborative resource for literature-derived, manually curated and computationally predicted structural and functional annotations of macromolecular structure data, contained in the Protein Data Bank (PDB). The goal of PDBe-KB is two-fold: (i) to increase the visibility and reduce the fragmentation of annotations contributed by specialist data resources, and to make these data more findable, accessible, interoperable and reusable (FAIR) and (ii) to place macromolecular structure data in their biological context, thus facilitating their use by the broader scientific community in fundamental and applied research. Here, we describe the guidelines of this collaborative effort, the current status of contributed data, and the PDBe-KB infrastructure, which includes the data exchange format, the deposition system for added value annotations, the distributable database containing the assembled data, and programmatic access endpoints. We also describe a series of novel web-pages—the PDBe-KB aggregated views of structure data—which combine information on macromolecular structures from many PDB entries. We have recently

released the first set of pages in this series, which provide an overview of available structural and functional information for a protein of interest, referenced by a UniProtKB accession.

PDBe-KB: collaboratively defining the biological context of structural data

Reference

CONSORTIUM P.-K.: **PDBe-KB: collaboratively defining the biological context of structural data**. *Nucleic Acids Research 50*, D1 (11 2021), D534–D542. doi:10.1093/nar/gkab988

Abstract

The Protein Data Bank in Europe – Knowledge Base (PDBe-KB, https://pdbe-kb.org) is an open collaboration between world-leading specialist data resources contributing functional and biophysical annotations derived from or relevant to the Protein Data Bank (PDB). The goal of PDBe-KB is to place macromolecular structure data in their biological context by developing standardised data exchange formats and integrating functional annotations from the contributing partner resources into a knowledge graph that can provide valuable biological insights. Since we described PDBe-KB in 2019, there have been significant improvements in the variety of available annotation data sets and user functionality. Here, we provide an overview of the consortium, highlighting the addition of annotations such as predicted covalent binders, phosphorylation sites, effects of mutations on the protein structure and energetic local frustration. In addition, we describe a library of reusable web-based visualisation components and introduce new features such as a bulk download data service and a novel superposition service that generates clusters of superposed protein chains weekly for the whole PDB archive.

AHoJ: rapid, tailored search and retrieval of apo and holo protein structures for user-defined ligands

Reference

FEIDAKIS C. P., KRIVAK R., HOKSZA D., NOVOTNY M.: **AHoJ: rapid, tailored search and retrieval of apo and holo protein structures for user-defined ligands**. *Bioinformatics 38*, 24 (10 2022), 5452–5453. doi:10.1093/bioinformatics/btac701

Author's highlights

We have developed AHoJ, a highly-configurable tool for the search and alignment of Apo-Holo protein pairs in the PDB. AHoJ is available as an open-source command line program and a web application that allows running searches for multiple queries at the same time (and thus produce Apo-Holo datasets) and includes integrated web-based visualization.

Bioinformatics, 38(24), 2022, 5452–5453
https://doi.org/10.1093/bioinformatics/btac701
Advance Access Publication Date: 25 October 2022
Applications Note

OXFORD

Structural bioinformatics

AHoJ: rapid, tailored search and retrieval of apo and holo protein structures for user-defined ligands

Christos P. Feidakis[1], Radoslav Krivak[2], David Hoksza[2] and Marian Novotny [1,*]

[1]Department of Cell Biology, Faculty of Science, Charles University, Prague 12843, Czech Republic and [2]Department of Software Engineering, Faculty of Mathematics and Physics, Charles University, Prague 12116, Czech Republic

*To whom correspondence should be addressed.

Associate Editor: Lenore Cowen

Received on July 5, 2022; revised on September 29, 2022; editorial decision on October 19, 2022; accepted on October 24, 2022

Abstract

Summary: Understanding the mechanism of action of a protein or designing better ligands for it, often requires access to a bound (holo) and an unbound (apo) state of the protein. Resources for the quick and easy retrieval of such conformations are severely limited. Apo–Holo Juxtaposition (AHoJ), is a web application for retrieving apo–holo structure pairs for user-defined ligands. Given a query structure and one or more user-specified ligands, it retrieves all other structures of the same protein that feature the same binding site(s), aligns them, and examines the superimposed binding sites to determine whether each structure is apo or holo, in reference to the query. The resulting superimposed datasets of apo–holo pairs can be visualized and downloaded for further analysis. AHoJ accepts multiple input queries, allowing the creation of customized apo–holo datasets.

Availability and implementation: Freely available for non-commercial use at http://apoholo.cz. Source code available at https://github.com/cusbg/AHoJ-project.

Contact: marian@natur.cuni.cz

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The study of protein–ligand interactions constitutes a prominent field in structural biology. Observing the effects of ligand binding (Brylinski and Skolnick, 2008), or exploring the specificity of a binding site (Ma *et al.*, 2002), involve studying several protein–ligand interactions. Unveiling cryptic binding sites (Cimermancic *et al.*, 2016), assessing the importance and consistency of water molecules (Wlodawer *et al.*, 2018), or transcending the technical limitations of rigid body docking with ensemble docking methodologies (Amaro *et al.*, 2018), also require access to several conformations (preferably apo and holo).

A number of datasets and tools have been built to address this need. ComSin (Lobanov *et al.*, 2010) comprised a database of apo and holo protein pairs which exhibit significant shifts in their levels of intrinsic disorder upon complex formation. AH-DB (Chang *et al.*, 2012) expanded this scope by including small ligands in its repertoire of apo–holo pairs. The BUDDY-system (Morita *et al.*, 2011) provided a more flexible solution where the user could specify the ligand of interest, and the application would try to pair up the provided holo structure with an apo counterpart. At the time of writing, none of these servers are available. A recent work in preprint (APObind—unpublished data) aims to complement an existing database of protein–ligand complexes, by pairing up the holo complexes with their apo counterparts. LigASite (Dessailly *et al.*, 2008) is a more dated yet surviving resource that features pairs of apo and holo structures for 550 proteins. In both cases however, the ligand cannot be specified by the user.

The available resources appear to be restricted, and in some cases non-existent. The ability to define a ligand, and therefore a binding site, that will guide the search for apo and holo structures is missing altogether. This can be particularly useful as proteins often bind several ligands, and even within the same protein, different structures can bind different ligands in the same or in different binding sites. Therefore, finding pairs of apo and holo structures for a given target structure, requires specifying one or more ligands of interest. A methodology that defines the relevant ligands according to a fixed assumption (i.e. automatically), can restrict a user who wants to focus on a ligand that is deemed irrelevant, or narrow down the search to a single ligand when more bind the same structure. Ultimately, when an application forcefully decides upon the relevance of a ligand, it strips the user of this choice and it is also confronted with the non-trivial matter of biological relevance (Capitani *et al.*, 2016).

Here, we present a web application that enables the user to conduct easy and fast parameterizable searches for apo and holo structure pairs against a target structure, by specifying one or more ligands of interest in this target structure, or letting the application detect the ligands

instead. By tracking the binding site of the user-defined ligand across structures, it can construct a repertoire of ligands that bind the same site and enable studies on binding-site specificity.

2 Materials and methods

AHoJ starts the search by spatially marking the user-defined ligand(s) and identifying their binding residues with PyMOL. Ligands are typically confined to non-protein chemical moieties, however in AHoJ, the concept of ligand can be extended to include water molecules and modified or non-standard residues (e.g. phosphorylated residues or D-residues) as points of interest or candidate ligands (see Supplementary Information for details).

It then compiles a list of candidate structure chains by (i) detecting the UniProt accession number (AC) (UniProt: the universal protein knowledgebase, 2017) of each query chain and (ii) retrieving all other chains that belong to the same UniProt AC. At the same time, it maps the binding residues of the query ligands onto the UniProt sequence by using the residue-level mappings from SIFTS (Dana *et al.*, 2019), and cross-examines each candidate chain to determine how many of the mapped binding residues are present. If a minimum percentage of binding residues is detected, the chain is considered a successful candidate and it is aligned onto the query chain with TM-align (Zhang and Skolnick, 2005). The user can adjust these parameters (see Supplementary Information for details). The candidate's area around the superimposed query ligand is examined for ligands, and the results are saved along with the aligned chains. This process is repeated for all candidate chains and each one is listed as holo or apo respective to the presence or absence of ligands in the defined binding site(s). The detected ligands along with metrics for the similarity between candidate and query, presence of binding residues and alignment scores, are reported for each apo and holo chain. The overall workflow is depicted in Supplementary Figure S1. Results are visualized in the browser and can be downloaded locally and loaded into PyMOL through an included script.

Acknowledgements

We thank the reviewers for taking the time to review the manuscript and providing valuable feedback.

Funding

This work was supported by the Grant Agency of Charles University [Project No. 1038120] and the ELIXIR CZ Research Infrastructure [ID LM2018131, MEYS CR].

Conflict of Interest: none declared.

References

Amaro,R.E. *et al.* (2018) Ensemble docking in drug discovery. *Biophys. J.*, **114**, 2271–2278.

Brylinski,M. and Skolnick,J. (2008) What is the relationship between the global structures of apo and holo proteins? *Proteins*, **70**, 363–377.

Capitani,G. *et al.* (2016) Understanding the fabric of protein crystals: computational classification of biological interfaces and crystal contacts. *Bioinformatics*, **32**, 481–489.

Chang,D.T.-H. *et al.* (2012) AH-DB: collecting protein structure pairs before and after binding. *Nucleic Acids Res.*, **40**, D472–D478.

Cimermancic,P. *et al.* (2016) CryptoSite: expanding the druggable proteome by characterization and prediction of cryptic binding sites. *J. Mol. Biol.*, **428**, 709–719.

Dana,J.M. *et al.* (2019) SIFTS: updated structure integration with function, taxonomy and sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins. *Nucleic Acids Res.*, **47**, D482–D489.

Dessailly,B.H. *et al.* (2008) LigASite—a database of biologically relevant binding sites in proteins with known apo-structures. *Nucleic Acids Res.*, **36**, D667–D673.

Lobanov,M. *et al.* (2010) ComSin: database of protein structures in bound (complex) and unbound (single) states in relation to their intrinsic disorder. *Nucleic Acids Res.*, **38**, D283–D287.

Ma,B. *et al.* (2002) Multiple diverse ligands binding at a single protein site: a matter of pre-existing populations. *Protein Sci.*, **11**, 184–197.

Morita,M. *et al.* (2011) BUDDY-system: a web site for constructing a dataset of protein pairs between ligand-bound and unbound states. *BMC Res. Notes*, **4**, 143.

Schiebel,J. *et al.* (2018) Intriguing role of water in protein-ligand binding studied by neutron crystallography on trypsin complexes. *Nat. Commun.*, **9**, 3559.

The UniProt Consortium (2017) UniProt: the universal protein knowledgebase. *Nucleic Acids Res.*, **45**, D158–D169.

Wlodawer,A. *et al.* (2018) Detect, correct, retract: how to manage incorrect structural models. *FEBS J.*, **285**, 444–466.

Zhang,Y. and Skolnick,J. (2005) TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.*, **33**, 2302–2309.

Supplementary Information

Detailed methodology

AHoJ uses the UniProt accession number (AC) of a protein to build the original pool of candidates, and then leverages residue-level mappings between each PDB structure and its corresponding UniProt sequence (which are precalculated for every structure in SIFTS files), to measure their sequence overlap with the query sequence, and also to map the query binding site(s) across the candidate structures (Figure S1). These, along with an additional set of metrics, are used to measure the biological similarity between the candidate and the query structures. Some of these metrics are informative, meant to indicate the quality of the match (between candidate and query) and help users decide which resulting structures to use, and others are used as thresholds to filter out candidates that are deemed unsuitable for an apo or holo classification. The metrics are described herein according to their order of appearance in the application pipeline, along with the relevant user variables where applicable.

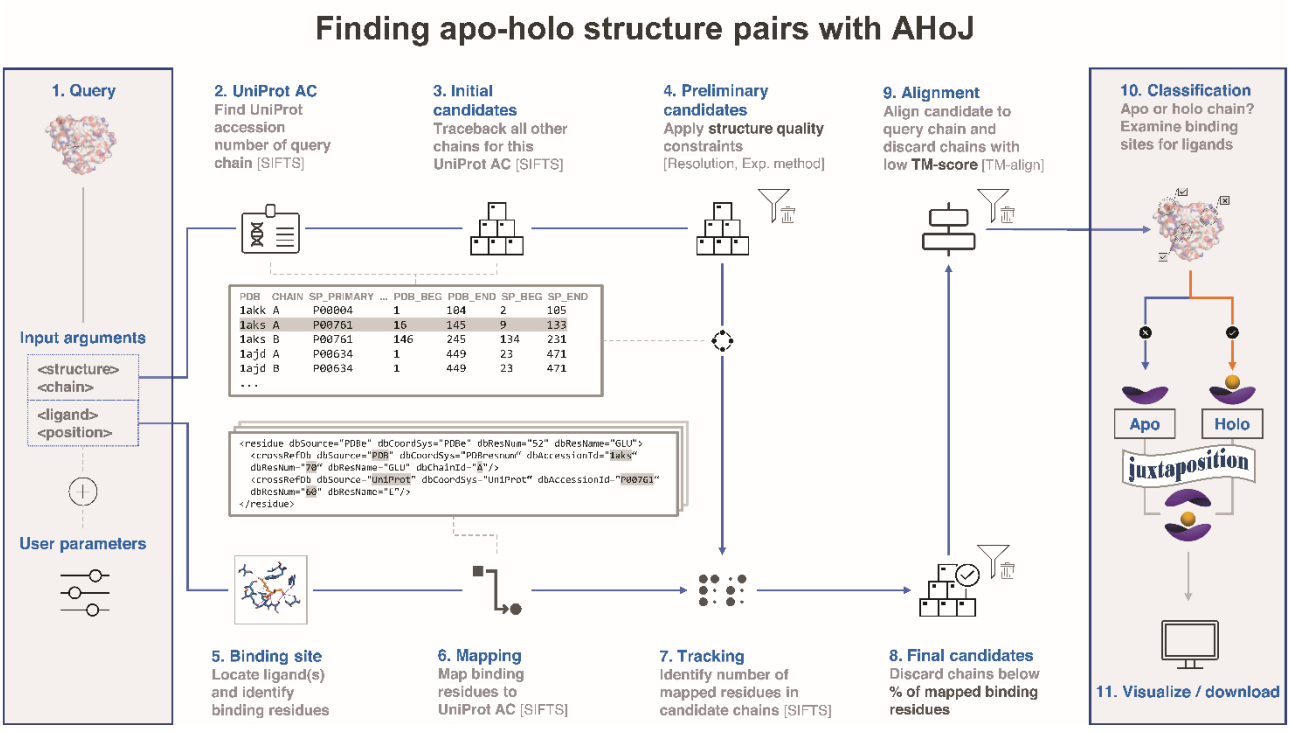


Figure S1. Flowchart depicting the workflow in AHoJ.

Ligand definition

AHoJ's architecture is primarily based on allowing users to define the ligands they deem relevant in each search. It was originally designed with single ligands in mind, with the ligand entity being confined to the size and naming convention of the “residue” group in the PDB and excluding non-heteroatom, standard, protein-coding residues. The original scope was extended to include multiple ligands in the same search, but the ligand entity remains confined to the “residue” group.
Multiple ligands can thus be specified per search, albeit they are considered as separate ligand entities. For example, oligosaccharides consist of multiple “single-residue” entities and can thus only be specified as multiple ligands. Standard, protein-coding residues, cannot be currently considered as ligands in AHoJ (e.g., peptides), but are used instead to detect the ligand(s) of interest they bind.
In the case of oligosaccharides, these consist of several heteroatom “residue” entities which can be searched indirectly (i.e., separately) in AHoJ, but not as a single “multi-residue” ligand. Multiple ligands can be used together in a single search query (example below), allowing for the indirect search of larger ligands; but this should be used carefully, as unlike single ligands, multiple ligands cannot be specified by their index positions in the structure, so AHoJ will detect any and all of these ligands in the specified chains (there could be more than the user-intended one).
An example of an oligosaccharide would be GLC-GAL-BGC in structure 3k0v. Here, this oligosaccharide could be searched indirectly by the query “`3k0v E GLC,GAL,BGC`”, where the three oligosaccharide components are considered separately but the search is successful as they are unique in this structure and chain. If any of these three components or “residues” had multiple occurrences in the structure and chain mentioned in the user query, these too would be detected and considered during the search, possibly reducing the resulting number of apo structures because of the additional constraints imposed. A possible workaround to this, would be to use a binding residue in the query (instead of the ligand(s) themselves), in this case “`3k0v A TYR 660`”. In this search, AHoJ would detect any heteroatom ligands in the proximity of the mentioned residue (TYR 660 in chain A), according to the user option “Ligand scanning radius”. In this example however, the default scanning radius would not suffice to detect all three sugar components, and it would have to be extended to 5.5 Angstroms to do so. We generally recommend caution when increasing the scanning radius, as this also affects the detection of ligands when searching for apo and holo structures later.

UniProt sequence overlap (mapping structures onto the UniProt sequence)

A key informative metric measures the percentage of overlap between the query structure chain and the prospective candidate chain. After all candidate structures for a given UniProt AC are retrieved (for a given query chain), each one is compared to the query sequence in terms of its overall coverage according to the start and end residue-level mappings that are available in the SIFTS files. This metric does not emerge from a pairwise alignment between candidate and query chain and does not refer to a sequence identity score; it is rather a comparison of the observed residues in each chain (i.e., present in the actual structures) that correspond to the same protein. The result is a percentage between 0 and 100, that reflects the percentage of amino acids in a given query structure chain, that are present in the given candidate structure chain. This metric is informative and it is not used as a cut-off threshold for filtering out candidate chains in a typical query where the query chain is holo, except for cases where the query chain is apo (and the default filtering by mapped binding residues cannot be applied). In such cases the user can specify a percentage as a minimum threshold (default is “0” (%)).

Note that this metric has directionality in the sense that the percentage of sequence overlap is computed from the perspective of the query chain and is therefore subject to length bias that may arise from comparing sequences of different sizes, much like in a typical sequence alignment. For example, a long query chain would be less likely to have candidates with a high percentage of sequence overlap, while a shorter query chain would be more likely to have candidates with a high percentage of sequence overlap. This constitutes a basic incentive to avoid reliance on sequence overlap – and not use it as a candidate eligibility criterion when possible. AHoJ circumvents this in the case of holo query chains, by mapping the binding site(s) residues between query and candidate chains instead, whose presence or absence may be irrelevant to the overall sequence overlap (see “Mapping the binding site” for details).

Structure quality and experimental method

In AHoJ the user can specify a minimum resolution threshold which is applicable to structures that are resolved by scattering methods, in order to discard structures of unwanted resolution. Additionally, it is possible to exclude NMR structures or only consider X-ray crystallography structures. Note that in the latter case, structures resolved by hybrid methods including X-ray crystallography (e.g., electron paramagnetic resonance and neutron diffraction), will be excluded. These variables are used as thresholds for discarding structures. The R-free of the structures is reported in the results when available.

Mapping the binding site

When the query is a holo structure, AHoJ marks the position of the defined ligand(s) and identifies the binding residues (ligands need to be heteroatom entities according to the PDB file, see “Ligand detection” and “Notion of extended ligand” below for more information about ligand eligibility). It then looks for the presence of these binding residues in the candidate structures, to determine whether the candidate is suitable for an apo or holo assessment. This operation is performed by mapping the PDB numbering of the binding residues onto the UniProt sequence numbering and then cross-referencing these positions with the candidate structures, to determine if the residues are present or absent in the actual structure. This is performed by parsing the SIFTS files with the residue-level mappings of a given structure. The metric is used as a cut-off threshold to discard candidates that do not feature any of the binding residues of the query binding sites. A minimum cut-off of “1” (%) is set by default (user adjustable) and it is applied as the minimum percentage of binding residues that have to be present in the candidate chain out of the total binding residues in the query chain, for the chain to be classified as apo or holo. In the case of an apo query structure that does not bind any ligands and thus does not have a designated binding site, this metric is not applied. In such cases, the first metric (UniProt sequence overlap) is applied as a cut-off.

Alignment

The candidate chains that score above the previous threshold, are subsequently aligned to the query chain with TM-align (Zhang and Skolnick, 2005). This step also serves as a cut-off point, by specifying a minimum TM-score between the candidate and query chains (default = 0.5 (minimum TM-score), user adjustable). For each TM-align, two TM-scores are generated, each normalized by the length of the two aligned chains, which gives rise to inherent directionality -or length bias- depending on which chain is first and which is second. AHoJ captures both TM-scores generated in every alignment and applies the minimum TM-score threshold (default = 0.5, user-adjustable) to the highest one, to avoid discarding candidate chains that score poorly on account of their low overall coverage against the query chain. The RMSD is also reported in the results as an informative metric.

Ligand detection

Successfully aligned candidates are assessed for ligands in the superimposed positions of the query ligands. Any heteroatom of the candidate structure that is positioned within a set radius (default = 4.5 Angstrom, user-adjustable) from the superposition of the specified (or auto-detected) query ligand atoms, is considered a ligand. By default, water molecules, modified residues and D-residues

are ignored (user-adjustable). The user can also specify whether AHoJ should consider any detected ligand (within the above conditions) or restrict the search to the same ligand that was specified in the user query. This parameter is turned off by default, so that any detected ligand is considered in this step. If at least one ligand atom is detected within this scanning radius, the candidate structure is classified as holo, otherwise, as apo. In the presence of multiple defined binding sites in the query chain, if at least one of them is occupied by a ligand in the candidate chain, the chain is characterized as holo. The PDB names of the detected ligands are featured in the results for each candidate chain, and their positions (chain and PDB position index) are included in a separate CSV file with ligand information.

The notion of extended ligand

AHoJ was designed around the premise that non-protein chemical moieties are the main point of interest in protein–ligand interaction research. Under this premise, it accepts any heteroatom as a ligand, that can be specified by its 1-3 PDB character code as a ligand name, and optionally also by its PDB position index in the structure.

Besides chemical compounds and ions, there is established evidence that water molecules hold a key role in understanding protein interactions (Schiebel et al., 2018). Furthermore, correctly assigning water molecules in the electron density maps of X-ray crystallographic structures, can be challenging, and has resulted in miss-annotations between water molecules and metal ions in deposited structures (Wlodawer et al., 2018). AHoJ allows users to define a water molecule as a ligand, and search for water molecules -or other ligands- in candidate structures in that particular superposition, in the same way that it would with a ligand, with the difference of changing internally the radius for scanning the candidate chain around the superposition of the query ligand from the default value of 4.5 to 2.5 Angstrom.

Another category of molecules that undoubtedly escape the definition of a ligand but are also important in understanding protein structure and function, are post translationally modified residues (e.g. phosphorylated residues). AHoJ allows users to specify such residue in a given structure, and search for apo and holo structures that lack or possess the specified modified residue in that particular superposition. Under the same principle, D-forms of amino acids can also be specified. Water molecules, modified residues and D-residues can be specified as input ligands through the user query or as candidate ligands (i.e., detectable entities affecting the apo or holo status of a candidate chain) through the respective parameters (*--water_as_ligand, --nonstd_rsds_as_lig, --d_aa_as_lig*).

Usage

AHoJ works on the principle that users have a structure of interest and a point of interest on that structure (i.e., ligand, modified residue, water molecule) that they want to compare -in terms of the presence or absence of this point of interest- to the other structures of the same protein. The use-case can thus vary according to the user's input (type of point of interest) and the parameters, but the main objective is to perform comparisons for a given point of interest across different structures of the same protein. To accommodate this versatility in different types of points of interest, AHoJ offers a set of options through user-adjustable parameters and a text query format (single line input) that can accept 1 to 4 arguments.

Query format

The maximum arguments within the single line input are of this form:
<pdb_code> <chains> <ligand_name> <ligand_position>

- pdb_code: This is the 4-character code of a PDB protein structure (case-insensitive). This argument is obligatory and only 1 PDB code can be input per line. (e.g., "1a73" or "3fav" or "3FAV"). If it is the only argument (i.e., because the user does not know the ligand that binds to the structure), it will trigger automatic detection of ligands in the structure.
- chains: A single chain or multiple chains separated by commas (without whitespace), or "!" in the case of ligand-binding-only chains, or "*" in the case of all chains (i.e. "A" or "A,C,D" or "!" or "*"). This argument is case-sensitive and it is obligatory if the user intends to provide any argument after that (i.e. ligands or position).
- ligand_name: This argument is case-insensitive. A single ligand, multiple ligands separated by commas (without whitespace), or no ligands can be input per line (e.g., "HEM" or "hem" or "ATP" or "ZN" or "HEM,ATP,ZN") or "*" for the automatic detection of all ligands in the specified chain(s). Besides specifying the ligand directly by its name (and optionally, its position), the user can also specify a residue that binds the ligand (e.g., "HIS") and AHoJ will detect the ligand (as long as it is within 4.5 Angstroms of the residue). This approach however can lead to the selection of more than one ligand if they are within this radius from the specified residue. This argument is non-obligatory, if omitted or specified as "*", AHoJ will automatically detect the ligands in the structure. If there are no ligands in the query structure, it will be characterized as apo and the search for candidates will continue. A water molecule can also be specified as a ligand (e.g., "HOH") but in such cases, its position must be specified as well. Note: when specifying the position argument, the user can only specify one ligand per query.

- ligand_position: This argument is an integer (e.g., “260” or “1”). It refers to the PDB index of the previously specified ligand, binding residue or water molecule. When this argument is specified, only one ligand or residue can be specified in the previous argument.

The primary mode of search in AHoJ, starts with a holo (bound) state. The most straightforward case is specifying a ligand as a point of interest. In such case, the ligand can be specified in the text query, by its 1-3 character PDB naming convention and also with its PDB index position in the amino acid sequence (this avoids considering all ligands of the same name that bind the same chain).

Examples

Example of a user query:
consider ZN ligand in position 201 in chain A of PDB code 1a73
'1a73 A ZN 201'
The application will fetch the structure 1a73 and look for zinc+2 (ZN) ligand in chain A and position 201 of the sequence to validate the input. If ZN is found in chain A and position 201 of 1a73 (1a73A), it will retrieve all other known chains that belong to the same protein with 1a73A, align them with 1a73A and look for ZN (and also other ligands) at the superimposed binding site of ZN in 1a73A. If it finds protein chains with ZN, it will list them as HOLO, if the superimposed site is empty of ligands, the chain will be listed as APO. If another ligand is detected on that site instead of ZN, the chain will be listed as APO or HOLO, depending on the value of *--lig_free_sites* parameter (if the user wants APO with no other ligands there, it will be listed as HOLO, and if the user allows other ligands in this binding site, it will be listed as APO).
Example of an alternative query that leads to the same result with the previous example:
consider ligands near residue HIS134 in chain A of 1a73 (the detected ligand will be ZN 201 in chain A)
'1a73 A HIS 134'

More examples of user queries

consider ZN ligands in chains A and B of 1a73
'1a73 A,B ZN'
consider ZN ligands in all chains of 1a73
'1a73 ALL ZN' or '1a73 * ZN'
find and consider all ligands in all chains of 1a73
'1a73'

find and consider all ligands in chain A of 1a73

'1a73 A'

consider ZN and MG ligands in chain A of 1a73

'1a73 A ZN,MG'

consider ZN ligands in all chains of 3fav

'3fav ZN'

Multiple queries

Besides single queries, AHoJ also accepts multiple queries at once and processes them in batch mode. Queries are separated by line breaks, and one query is entered per line. The results for each query are saved in a separate folder and all of them are packed and downloaded in a single file. This can be useful for building datasets of apo and holo structures or simply processing multiple queries at once.

Example of a multiple query with comments for every single query (characters after "#" are ignored and can be used as comments):

1a73 A,B ZN # consider ZN ligands in chains A and B of 1a73

1a73 ALL ZN # consider ZN ligands in all chains of 1a73

1a73 # find and consider all ligands in all chains of 1a73

1a73 A # find and consider all ligands in chain A of 1a73

1a73 A ZN,MG # consider ZN and MG ligands in chain A of 1a73

3fav ALL ZN # consider ZN ligands in all chains of 3fav

1DB1 # vitamin D3 study

4est # porcine pancreatic elastase

3CQV # reverb beta - all chains, all ligands

3CQV A # reverb beta - chain A (in this case same effect)

3CQV A HEM # reverb beta - ligand HEM (in this case same effect)

Results

The results are visualized in the browser through Mol* and they can be downloaded as a zip file after a run has completed.

Files

In a successful run, AHoJ should generate the following files:

i) PDB structure files (cif.gz format) for the query structure (whole structure) and the successfully processed apo and holo candidate chains, aligned to the respective query chain(s).

Note: a given candidate chain could be a match for more than one query chains, and could thus appear more than once, in each case aligned to the respective query chain.
ii) 1 or 2 CSV files with the successfully processed candidate chains for apo and holo chains respectively [results_apo.csv, results_holo.csv]. These CSV files contain the following information for each found chain: **query_chain, apo_chain, Resolution, R-free, %UniProt_overlap, Mapped_bndg_rsds, %Mapped_bndg_rsds, RMSD, TM_score, iTM_score, ligands**
iii) 1 CSV file with the positions of the relevant ligands that were detected in both query and resulting candidate structures. This file is needed to load ligand selections when loading the results into the PyMOL with the included script.
Note: The ligands listed in the files refer to the ligands that were detected in the superimposed positions of the specified query ligands, thus they might not include ligands that bind elsewhere in the candidate chains. If the CSV file for apo chains includes ligands (which seems contradicting), it indicates that the user set the parameter *--lig_free_sites* to 0 (OFF), and thus any other ligands besides the query ligand were detected in the superimposed binding sites of candidate structures but ignored.
iii) 1 CSV file with information of the ligand positions for both query and candidate structures [ligands.csv]. This file is important for reference purposes and also if the user wants to reconstruct the PyMOL session with annotations locally.
iv) Console log file with information from the standard output [console.log]. This file can be used for reference and for better understanding the mechanism of action of AHoJ.
v) A PyMOL script file for loading the results into a PyMOL session [load_results_into_PyMOL.pml]. This is useful for viewing the results locally on the user's computer. The script has to be opened through PyMOL. The resulting session can then be saved by the user as a PyMOL session (.pse).

Visualization

i) The web application allows the visualization of the results in the browser with the molstar (Mol*) viewer. Web application: https://github.com/rdk/AHoJ-webapp
ii) The results can also be downloaded and visualized locally by loading the PyMOL script that is included in the results folder through PyMOL [load_results_into_PyMOL.pml]. The script has to be loaded from within the results folder. After downloading and unpacking the results into a folder, start a new PyMOL session and open the .pml file through it. A PyMOL installation is needed for this to work (Incentive or Open-Source)

Parameters

Basic

--res_threshold : resolution threshold [default = 3.8]
Floating point number that represents angstroms and is applied as a cutoff point when assessing candidate structures that are resolved by scattering methods (X-ray crystallography, electron microscopy, neutron diffraction). It applies at the highest resolution value, when this is available in the PDB structure file. It can take any value, suggested min/max = 1.5/8. Condition is <=
--include_nmr : include NMR structures [default = 1]
0 or 1. When set to 1 (ON), NMR structures are considered as candidates. In the case of multiple states for a certain structure, the first one is considered.
--xray_only : x-ray structures only [default = 0]
0 or 1. When set to 1 (ON), only X-ray structures are considered. This overrides the NMR setting.
--lig_free_sites : ligand-free sites [default = 1]
0 or 1. When set to 1 (ON), it does not tolerate any ligands (in addition to the user-specified one(s)) in the superimposed binding sites of the candidate apo-proteins. When set to 0 (OFF), it tolerates ligands other than the user-specified one(s) in the same superimposed binding site(s). If the user wants to find apo structures that don't bind any ligands in the superimposed binding site(s) of the query ligand(s), they should set this value to 1 (default).

Advanced

--bndgrsds_threshold : binding residues threshold [default = 1.0, min/max = 1/100]
Floating point number that represents a percentage (%) and is applied as a minimum cut-off upon the percentage of the number of successfully mapped binding residues in the candidate chain out of the total number of binding residues in the query chain. The binding residues are mapped between query and candidate by converting PDB to UniProt numbering. "1%" translates to at least 1% percent of the query residues being present in the candidate structure, for the structure to be considered as apo or holo.
--save_apo : save aligned Apo chains [default = 1]
0 or 1. When set to 1 (ON), saves the structure files of the aligned APO chains (mmCIF). Disabling this is only recommended in multiple queries if visualizations are not needed (reduces download size). This setting does not affect the search for apo or holo chains or the final result reports.
--save_holo : save aligned Holo chains [default = 1]
0 or 1. When set to 1 (ON), saves the structure files of the aligned HOLO chains (mmCIF). Disabling this is only recommended in multiple queries if visualizations are not needed (reduces

download size). This setting does not affect the search for apo or holo chains or the final result reports.

--overlap_threshold : sequence overlap threshold [default = 0, min/max = 0/100]

Floating point number that represents a percentage (%) and is applied as a cutoff point when comparing the sequence overlap between the query and the candidate chain. It applies to the percentage of sequence overlap between query and candidate chains, and it is calculated from the query's perspective according to the UniProt residue numbering. If set to 100 (%), it means that the candidate chain has to completely cover the query chain. It can be longer than the query, but not shorter.

Note: "100" guarantees complete coverage, but it is the strictest setting. If the user wants a more lenient filtration, they can lower the value, or even set it to 0 and rely on the template-modeling score (TM-score) by using the default value (0.5) or setting their own TM-score cutoff with the "--min_tmscore" parameter.

--lig_scan_radius : ligand scanning radius [default = 4.0]

Floating point number that represents angstroms and is applied as a scanning radius when looking for ligands in the candidate structures. This scanning radius is applied on the positions of the atoms of the superimposed query ligands to the aligned candidate structure, to scan for ligands. The resulting scanning space is a "carved" surface that has the shape of the query ligand, extended outward by the given radius. If the candidate structure binds ligands outside of this superimposed area, they will be ignored, and the candidate will be characterised as an apo-protein.

--min_tmscore : minimum TM-score [default = 0.5, min/max = 0/1]

Floating point number that is applied as a minimum accepted template-modeling score between the query and the candidate chain. Value 1 indicates a perfect match, values higher than 0.5 generally assume the same fold in SCOP/CATH.

--water_as_ligand : [default = 0]

0 or 1. When set to 1 (ON), allows the detection of water molecules (i.e., 'HOH') as ligands in the superposition of the query ligand(s) in the candidate chains. If this setting is enabled and at least one water molecule is detected within the scanning radius, that would warrant a holo classification for the candidate chain. When a water molecule is defined in the user query, this setting is automatically enabled.

--nonstd_rsds_as_lig : non-standard residues as ligands [default = 0]

0 or 1. When set to 1 (ON), allows the detection of non-standard -or modified- residues (e.g., 'TPO', 'SEP') as ligands in the superposition of the query ligand(s) in the candidate chains. If this setting is enabled and at least one modified residue is detected within the scanning radius, that would warrant

a holo classification for the candidate chain. When a modified residue is defined in the user query, this setting is automatically enabled.
Note: The current list of non-standard residues includes the following residue names: 'SEP TPO PSU MSE MSO 1MA 2MG 5MC 5MU 7MG H2U M2G OMC OMG PSU YG PYG PYL SEC PHA'.
--d_aa_as_lig : D-amino acids as ligands [default = 0]
0 or 1. When set to 1 (ON), allows the detection of D-residues (e.g., 'DAL', 'DSN') as ligands in the superposition of the query ligand(s) in the candidate chains. If this setting is enabled and at least one D-residue is detected within the scanning radius, that would warrant a holo classification for the candidate chain. When a D-residue is defined in the user query, this setting is automatically enabled.
Note: The current list of D-residues includes the following residue names: 'DAL DAR DSG DAS DCY DGN DGL DHI DIL DLE DLY MED DPN DPR DSN DTH DTR DTY DVA'.

Bibliography

[BCdF09] BROCHU E., CORA V. M., DE FREITAS N.: **A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning**. *CoRR abs/1012.2599* (2009).

[BS08] BRYLINSKI M., SKOLNICK J.: **What is the relationship between the global structures of apo and holo proteins?** *Proteins: Structure, Function, and Bioinformatics 70*, 2 (2008), 363–377. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.21510, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.21510, doi:https://doi.org/10.1002/prot.21510.

[BS17] BROOMHEAD N. K., SOLIMAN M. E.: **Can We Rely on Computational Predictions To Correctly Identify Ligand Binding Sites on Novel Protein Drug Targets? Assessment of Binding Site Prediction Methods and a Protocol for Validation of Predicted Binding Sites**. *Cell Biochemistry and Biophysics 75*, 1 (Mar 2017), 15–23. URL: https://doi.org/10.1007/s12013-016-0769-y, doi:10.1007/s12013-016-0769-y.

[BSSC18] BHAGAVAT R., SANKAR S., SRINIVASAN N., CHANDRA N.: **An Augmented Pocketome: Detection and Analysis of Small-Molecule Binding Pockets in Proteins of Known 3D Structure**. *Structure 26*, 3 (2018), 499 – 512.e2. doi:https://doi.org/10.1016/j.str.2018.02.001.

[CHG14] CHEN P., HUANG J. Z., GAO X.: **LigandRFs: random forest ensemble to identify ligand-binding residues from sequence information alone.** *BMC bioinformatics 15 Suppl 15* (Jan 2014), S4. doi:10.1186/1471-2105-15-S15-S4.

[CLT*09] CAPRA J. A., LASKOWSKI R. A., THORNTON J. M., SINGH M., FUNKHOUSER T. A.: **Predicting Protein Ligand Binding Sites by Combining Evolutionary Sequence Conservation and 3D Structure**. *PLoS Comput Biol 5*, 12 (12 2009), e1000585.

[CMGK11] CHEN K., MIZIANTY M., GAO J., KURGAN L.: **A critical comparative assessment of predictions of protein-binding sites for biologically relevant organic compounds.** *Structure (London, England : 1993) 19*, 5 (2011), 613–621. URL: http://dx.doi.org/10.1016/j.str.2011.02.015, doi:10.1016/j.str.2011.02.015.

[con19] CONSORTIUM P.-K.: **PDBe-KB: a community-driven resource for structural and functional annotations**. *Nucleic Acids Research 48*, D1 (10 2019), D344–D353. doi:10.1093/nar/gkz853.

[con21] CONSORTIUM P.-K.: **PDBe-KB: collaboratively defining the biological context of structural data**. *Nucleic Acids Research 50*, D1 (11 2021), D534–D542. doi:10.1093/nar/gkab988.

[CWR*16] CIMERMANCIC P., WEINKAM P., RETTENMAIER T. J., BICHMANN L., KEEDY D. A., WOLDEYES R. A., SCHNEIDMAN-DUHOVNY D., DEMERDASH O. N., MITCHELL J. C., WELLS J. A., ET AL.: **CryptoSite: expanding the druggable proteome by characterization and prediction of cryptic binding sites**. *Journal of molecular biology 428*, 4 (2016), 709–719.

[CYF*12] CHANG D. T.-H., YAO T.-J., FAN C.-Y., CHIANG C.-Y., BAI Y.-H.: **AH-DB: collecting protein structure pairs before and after binding**. *Nucleic acids research 40*, D1 (2012), D472–D478.

[DLOW07] DESSAILLY B. H., LENSINK M. F., ORENGO C. A., WODAK S. J.: **LigASite—a database of biologically relevant binding sites in proteins with known apo-structures**. *Nucleic acids research 36*, suppl_1 (2007), D667–D673.

[DWH15] DEGAC J., WINTER U., HELMS V.: **Graph-Based Clustering of Predicted Ligand-Binding Pockets on Protein Surfaces**. *Journal of Chemical Information and Modeling 55*, 9 (2015), 1944–1952. PMID: 26325445. doi:10.1021/acs.jcim.5b00045.

[FB15] FEINSTEIN W. P., BRYLINSKI M.: **Calculating an optimal box size for ligand docking and virtual screening against experimental and predicted binding pockets**. *Journal of Cheminformatics 7*, 1 (2015), 1–10. URL: http://dx.doi.org/10.1186/s13321-015-0067-5, doi:10.1186/s13321-015-0067-5.

[FKHN22] FEIDAKIS C. P., KRIVAK R., HOKSZA D., NOVOTNY M.: **AHoJ: rapid, tailored search and retrieval of apo and holo protein structures for user-defined ligands**. *Bioinformatics 38*, 24 (10 2022), 5452–5453. doi:10.1093/bioinformatics/btac701.

[FRH11] FAUMAN E. B., RAI B. K., HUANG E. S.: **Structure-based druggability assessment—identifying suitable targets for small molecule therapeutics**. *Current Opinion in Chemical Biology 15*, 4 (2011), 463 – 468. Next Generation Therapeutics. doi:https://doi.org/10.1016/j.cbpa.2011.05.020.

[GS09] GHERSI D., SANCHEZ R.: **EasyMIFS and SiteHound: a toolkit for the identification of ligand-binding sites in protein structures.** *Bioinformatics (Oxford, England) 25*, 23 (2009), 3185–3186. URL: http://dx.doi.org/10.1093/bioinformatics/btp562, doi:10.1093/bioinformatics/btp562.

[HBG*15] HUSSEIN H., BORREL A., GENEIX C., PETITJEAN M., REGAD L., CAMPROUX A.: **PockDrug-Server: a new web server for predicting pocket druggability on holo and apo proteins**. W436–W442. doi:10.1093/nar/gkv462.

[HLH*13] HUANG W., LU S., HUANG Z., LIU X., MOU L., LUO Y., ZHAO Y., LIU Y., CHEN Z., HOU T., ZHANG J.: **Allosite: a method for predicting allosteric sites**. *Bioinformatics 29*, 18 (2013), 2357–2359. doi:10.1093/bioinformatics/btt399.

[HOH*10] HENRICH S., OUTI S., HUANG B., RIPPMANN F., CRUCIANI G., WADE R.: **Computational approaches to identifying and characterizing protein binding sites for ligand design.** *Journal of molecular recognition : JMR 23*, 2 (2010), 209–219. URL: http://dx.doi.org/10.1002/jmr.984, doi:10.1002/jmr.984.

[JDMR*17] JIMÉNEZ J., DOERR S., MARTÍNEZ-ROSELL G., ROSE A. S., DE FABRITIIS G.: **DeepSite: protein-binding site predictor**

using 3D-convolutional neural networks. *Bioinformatics 33*, 19 (2017), 3036–3042. doi:10.1093/bioinformatics/btx350.

[JG17] JIMÉNEZ J., GINEBRA J.: **pyGPGO: Bayesian Optimization for Python**. *Journal of Open Source Software 2*, 19 (2017), 431. URL: https://doi.org/10.21105/joss.00431, doi:10.21105/joss.00431.

[JKS*19] JENDELE L., KRIVAK R., SKODA P., NOVOTNY M., HOKSZA D.: **PrankWeb: a web server for ligand binding site prediction and visualization**. *Nucleic Acids Res. 47*, W1 (Jul 2019), W345–W349. doi:10.1093/nar/gkz424.

[JSK*22] JAKUBEC D., SKODA P., KRIVAK R., NOVOTNY M., HOKSZA D.: **PrankWeb 3: accelerated ligand-binding site predictions for experimental and modelled protein structures**. *Nucleic Acids Research 50*, W1 (05 2022), W593–W597. doi:10.1093/nar/gkac389.

[KH15a] KRIVÁK R., HOKSZA D.: **Improving protein-ligand binding site prediction accuracy by classification of inner pocket points using local features**. *Journal of Cheminformatics 7*, 1 (Apr 2015), 12. doi:10.1186/s13321-015-0059-5.

[KH15b] KRIVÁK R., HOKSZA D.: **P2RANK: Knowledge-Based Ligand Binding Site Prediction Using Aggregated Local Features**. In *International Conference on Algorithms for Computational Biology* (2015), Springer, pp. 41–52. doi:10.1007/978-3-319-21233-3_4.

[KH18] KRIVÁK R., HOKSZA D.: **P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure**. *Journal of cheminformatics 10*, 1 (2018), 39. doi:10.1186/s13321-018-0285-8.

[KH7] KRIVÁK R., HOKSZA D., ŠKODA P.: **Improving quality of ligand-binding site prediction with Bayesian optimization**. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2017), pp. 2278–2279. doi:10.1109/BIBM.2017.8218024.

[KJ14] KONC J., JANEŽIČ D.: **Binding site comparison for function prediction and pharmaceutical discovery.** *Current opinion in*

structural biology 25 (Apr 2014), 34–9. doi:10.1016/j.sbi.2013.11.012.

[KJH18] KRIVÁK R., JENDELE L., HOKSZA D.: **Peptide-Binding Site Prediction From Protein Structure via Points on the Solvent Accessible Surface**. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics* (New York, NY, USA, 2018), BCB '18, Association for Computing Machinery, p. 645–650. doi:10.1145/3233547.3233708.

[KK09] KAUFFMAN C., KARYPIS G.: **LIBRUS: combined machine learning and homology information for sequence-based ligand-binding residue prediction.** *Bioinformatics (Oxford, England) 25*, 23 (Dec 2009), 3099–107. URL: http://bioinformatics.oxfordjournals.org/cgi/pmidlookup?view=long&pmid=19786483, doi:10.1093/bioinformatics/btp561.

[LEG16] LAURIE E. GROVE SANDOR VAJDA D. K.: **Computational Methods to Support Fragment-based Drug Discovery**. In *Fragment-based Drug Discovery: Lessons and Outlook*, Fagerberg J., Mowery D. C., Nelson R. R., (Eds.). Wiley, Weinheim, 2016, ch. 9, pp. 197–222.

[LGST09] LE GUILLOUX V., SCHMIDTKE P., TUFFERY P.: **Fpocket: an open source platform for ligand pocket detection.** *BMC bioinformatics 10* (2009). URL: http://dx.doi.org/10.1186/1471-2105-10-168, doi:10.1186/1471-2105-10-168.

[LJ06] LAURIE A., JACKSON R.: **Methods for the prediction of protein-ligand binding sites for structure-based drug design and virtual ligand screening.** *Current protein & peptide science 7*, 5 (2006), 395–406.

[LSCZ14] LIONTA E., SPYROU G., COURNIA D. K. V., ZOE: **Structure-Based Virtual Screening for Drug Discovery: Principles, Applications and Recent Advances**. *Current Topics in Medicinal Chemistry 14*, 16 (2014), 1923–1938. URL: http://www.eurekaselect.com/node/124979/article.

[LSG*10] LOBANOV M. Y., SHOEMAKER B. A., GARBUZYNSKIY S. O., FONG J. H., PANCHENKO A. R., GALZITSKAYA O. V.: **ComSin:**

database of protein structures in bound (complex) and unbound (single) states in relation to their intrinsic disorder. *Nucleic Acids Research 38*, suppl_1 (2010), D283–D287.

[LSZ10] LEIS S., SCHNEIDER S., ZACHARIAS M.: **In silico prediction of binding sites on proteins.** *Current medicinal chemistry 17*, 15 (2010), 1550–1562.

[MBB16] MEYERS J., BROWN N., BLAGG J.: **Mapping the 3D structures of small molecule binding sites**. *Journal of Cheminformatics 8*, 1 (2016), 70. doi:10.1186/s13321-016-0180-0.

[MSWN02] MA B., SHATSKY M., WOLFSON H. J., NUSSINOV R.: **Multiple diverse ligands binding at a single protein site: a matter of pre-existing populations**. *Protein science 11*, 2 (2002), 184–197.

[MTNS11] MORITA M., TERADA T., NAKAMURA S., SHIMIZU K.: **BUDDY-system: A web site for constructing a dataset of protein pairs between ligand-bound and unbound states**. *BMC Research Notes 4* (2011), 1–4.

[MZF*17] MONZON A. M., ZEA D. J., FORNASARI M. S., SALDAÑO T. E., FERNANDEZ-ALBERTI S., TOSATTO S. C. E., PARISI G.: **Conformational diversity analysis reveals three functional mechanisms in proteins**. *PLOS Computational Biology 13*, 2 (02 2017), 1–18. URL: https://doi.org/10.1371/journal.pcbi.1005398, doi:10.1371/journal.pcbi.1005398.

[PSM*10] PÉROT S., SPERANDIO O., MITEVA M., CAMPROUX A., VILLOUTREIX B.: **Druggable pockets and binding site centric chemical space: a paradigm shift in drug discovery.** *Drug discovery today 15*, 15-16 (2010), 656–667. URL: http://dx.doi.org/10.1016/j.drudis.2010.05.015, doi:10.1016/j.drudis.2010.05.015.

[QW00] QIU Z., WANG X.: **Improved Prediction of Protein Ligand-Binding Sites Using Random Forests**. *Protein and Peptide Letters 18*, 12 (2011-12-01T00:00:00), 1212–1218. URL: http://www.ingentaconnect.com/content/ben/ppl/2011/00000018/00000012/art00005, doi:doi:10.2174/092986611797642788.

[RBJ15] ROCHE D. B., BRACKENRIDGE D. A., J M. L.: **Proteins and Their Interacting Partners: An Introduction to Protein-Ligand Binding Site Prediction Methods.** *Int J Mol Sci 16*, 12 (2015), 29829–42. doi:10.3390/ijms161226202.

[SBB*14] SCHOMBURG K., BIETZ S., BRIEM H., HENZLER A., URBACZEK S., RAREY M.: **Facing the challenges of structure-based target prediction by inverse virtual screening.** *Journal of chemical information and modeling 54*, 6 (2014), 1676–86. doi:10.1021/ci500130e.

[SBD*21] SEHNAL D., BITTRICH S., DESHPANDE M., SVOBODOVÁ R., BERKA K., BAZGIER V., VELANKAR S., BURLEY S. K., KOČA J., ROSE A. S.: **Mol* Viewer: modern web app for 3D visualization and analysis of large biomolecular structures**. *Nucleic Acids Research 49*, W1 (05 2021), W431–W437. URL: https://doi.org/10.1093/nar/gkab314, arXiv:https://academic.oup.com/nar/article-pdf/49/W1/W431/38842088/gkab314.pdf, doi:10.1093/nar/gkab314.

[SCS*17] SHEN Q., CHENG F., SONG H., LU W., ZHAO J., AN X., LIU M., CHEN G., ZHAO Z., ZHANG J.: **Proteome-Scale Investigation of Protein Allosteric Regulation Perturbed by Somatic Mutations in 7,000 Cancer Genomes**. *The American Journal of Human Genetics 100*, 1 (2017), 5 – 20. URL: http://www.sciencedirect.com/science/article/pii/S0002929716304013, doi:https://doi.org/10.1016/j.ajhg.2016.09.020.

[Sel17] SELLÉS J. P.: **FastRandomForest 2.0**, 2017. URL: https://github.com/GenomeDataScience/FastRandomForest.

[SLA12] SNOEK J., LAROCHELLE H., ADAMS R. P.: **Practical Bayesian Optimization of Machine Learning Algorithms**. In *Advances in Neural Information Processing Systems 25*. 2012, pp. 2951–2959.

[SLD*] SIMÕES T., LOPES D., DIAS S., FERNANDES F., PEREIRA J., JORGE J., BAJAJ C., GOMES A.: **Geometric Detection Algorithms for Cavities on Protein Surfaces in Molecular Graphics: A Survey**. *Computer Graphics Forum*. doi:10.1111/cgf.13158.

[SSKH07] SOGA S., SHIRAI H., KOBORI M., HIRAYAMA N.: **Use of Amino Acid Composition to Predict Ligand-Binding Sites**. *Journal of Chemical Information and Modeling 47*, 2 (2007), 400–406. PMID: 17243757. doi:10.1021/ci6002202.

[Sup13] SUPEK F.: **FastRandomForest**, 2013. URL: https://code.google.com/archive/p/fast-random-forest/.

[TAW*21] TUNYASUVUNAKOOL K., ADLER J., WU Z., GREEN T., ZIELINSKI M., ŽÍDEK A., BRIDGLAND A., COWIE A., MEYER C., LAYDON A., ET AL.: **Highly accurate protein structure prediction for the human proteome**. *Nature 596*, 7873 (2021), 590–596.

[TBNT16] TIBAUT T., BORIŠEK J., NOVIČ M., TURK D.: **Comparison of in silico tools for binding site prediction applied for structure-based design of autolysin inhibitors**. *SAR and QSAR in Environmental Research 27*, 7 (2016), 573–587. PMID: 27686112. doi:10.1080/1062936X.2016.1217271.

[WDP*18] WLODAWER A., DAUTER Z., POREBSKI P. J., MINOR W., STANFIELD R., JASKOLSKI M., POZHARSKI E., WEICHENBERGER C. X., RUPP B.: **Detect, correct, retract: How to manage incorrect structural models**. *The FEBS journal 285*, 3 (2018), 444–466.

[XH12] XIE Z., HWANG M.: **Ligand-binding site prediction using ligand-interacting and binding site-enriched protein triangles**. *Bioinformatics 28*, 12 (2012), 1579–1585. doi:10.1093/bioinformatics/bts182.

[XXB11] XIE L., XIE L., BOURNE P. E.: **Structure-based systems biology for analyzing off-target binding.** *Current opinion in structural biology 21*, 2 (Apr 2011), 189–99. doi:10.1016/j.sbi.2011.01.004.

[ZGWW12] ZHENG X., GAN L., WANG E., WANG J.: **Pocket-Based Drug Design: Exploring Pocket Space**. *The AAPS Journal* (2012). doi:10.1208/s12248-012-9426-6.

[ZLL*11] ZHANG Z., LI Y., LIN B., SCHROEDER M., HUANG B.: **Identification of cavities on protein surface using multiple computational approaches for drug binding site prediction.** *Bioinformatics (Oxford, England) 27*, 15 (2011), 2083–2088. URL: http://dx.doi.org/10.1093/bioinformatics/btr331, doi:10.1093/bioinformatics/btr331.

List of Figures

List of Tables