Imperial College London

Computing Group Project

MSc Bioinformatics and Theoretical Systems Biology

Department of Life Science

March 2021

Prediction of ligand binding and the impact of missense variants via docking

Authors

Charles Harris (CH)

Sofia Kapsiani (SK)

Ananporn Supataragul (AS)

Oscar Wright (OW)

Supervisor

Prof. Michael Sternberg

Abstract

Written by SK, AS

Template-based modelling methods, such as Phyre2, attempt to predict the protein structure of a query sequence based on homologous proteins with experimentally solved structures. Protein models generated by Phyre2 can be further enhanced by identifying ligand-binding sites, which provide critical information on the biological process of the protein. Several modelling techniques such SWISS-MODEL and COACH-D can generate ligand-protein complexes, however, these approaches do not indicate how close the predicted ligand is to the true biological ligand.

Here we present the LignadTemplateFinder, an automated approach for identifying protein complexes within the PDBe-KB database with biologically-relevant ligands bound. We have established an in-house cross-referencing between the UniProtKB, PDBe-KB and ChEBI databases to determine whether the ligand bound to a protein structure is the biological ligand or a closely related molecule, through the concept of chemical similarity. Ligand coordinates can be manually extracted from the PDB templates and inherited into Phyre2 predicted structures. Our approach was tested on 1,000 protein sequences and PDB templates with ligand information were obtained for 71.1% of the entries.

Additionally, we have developed a separate automated approach to explore how missense variants can impact protein-ligand interactions by using homology-based docking. Our pipeline successfully overcomes the inability to analyse modelled structures in existing methods, eliminates the manual pre-processing steps of docking and provides further details on the docking results. A dataset of missense variants with known impacts on protein structure was used to assess the pipeline. 19 out of 21 docking results are consistent with the missense impacts on binding information in the dataset. Limitations of the pipeline were identified which could be addressed in the future.

Contents

1 Introduction **5**

1.1 Background . . . 7

1.1.1 Template-based structure prediction . . . 7

1.1.2 Databases of sequence, protein structure and ligands . . . 7

1.1.3 Chemical similarity . . . 8

1.1.4 Ligand docking . . . 10

1.2 Related work . . . 11

2 LigandTemplateFinder **13**

2.1 Methods . . . 13

2.1.1 LigandTemplateFinder overview . . . 13

2.1.2 Mapping between UniProt, PDBe and ChEBI . . . 14

2.1.3 UniProt binding sites . . . 16

2.1.4 Similarity search algorithm . . . 17

2.1.5 Identification of the true sequence ligands in case of no mappings . . . 17

2.1.6 Identification of valid templates with correct ligands bound . . . 18

2.1.7 Inheritance of ligand coordinates . . . 19

2.1.8 LigandTemplateFinder input . . . 19

2.1.9 Evaluation of our approach . . . 20

2.1.10 Evaluation of similarity search . . . 21

2.2 Results . . . 22

2.2.1 LigandTemplateFinder output example . . . 22

2.2.2 Ability to identify templates with known ligands bound . . . 24

2.2.3 Ligand frequency in UniProt entries . . . 25

2.2.4 Evaluation of chemical similarity algorithm . . . 27

2.2.5 Modelling large ligands . . . 29

2.2.6 Modelling template with analogue . . . 30

2.2.7 Ligand transfer analysis . . . 32

2.2.8 Efficiency of database cross-referencing . . . 33

3 Predicting the impact of missense variants on ligand binding via homology-based docking **34**

3.1 Methods . . . 34

3.1.1 Programs and Python modules . . . 34

3.1.2 Pipeline inputs . . . 36

3.1.3 Missense PDB file generation . . . 36

3.1.4 File preparations and homology-based ligand docking . . . 37

3.1.5 Analysis of docked ligand . . . 39

3.1.6 Protein-ligand interactions visualization . . . 40
3.1.7 Pipeline benchmarking . . . 40
3.2 Results . . . 43
3.2.1 Pipeline usage . . . 43
3.2.2 Benchmarking results . . . 43
3.2.3 Analysis 1: Comparing predicted WT docking to crystalised WT docking . . . 45
3.2.4 Analysis 2: Comparing predicted missense docking to crystalised missense docking 45
3.2.5 Analysis 3: Comparing predicted WT docking to predicted missense docking . . . 45
3.2.6 Case study: Testing pipeline on missense variants which are known to affect the ligand-binding affinity . . . 46

4 Discussion **49**
4.1 LigandTemplateFinder . . . 49
4.1.1 Enhancing ligand modelling . . . 49
4.1.2 Problems with binding site identification . . . 50
4.1.3 Improvements in PDB ligand mapping . . . 51
4.1.4 Improvements in UniProt ligand mapping . . . 52
4.1.5 Limitations in chemical similarity searches . . . 53
4.2 Docking and missense variant prediction . . . 54
4.2.1 Limitations and future work . . . 54

5 Conclusion **56**

6 Appendix **62**

Abbreviations

API Application Programming Interface

CAA Acyl-Coenzyme A

FAD Flavin Adenine Dinucleotide

PDB Protein Data Bank

RMSD Root-Mean-Square Deviation

SNP Single Nucleotide Polymorphism

WT Wild Type

1 Introduction

Written by AS

Understanding the effects of missense variants is essential in many areas of biology. Not only can these variants alter the functions of proteins but they can also impact the binding of ligands to proteins. Particularly in the field associated with disease, where interactions between protein and ligand or drug are critical, a slight alteration in the interactions could lead to a drop in drug efficiency. There are several strategies available that report the effects of missense variants on protein stability (Ittisoponpisan et al. 2019). However, these strategies lack assessment of missense variants effects on protein-ligand interactions and the ability to analyse modelled structure.

Written by CH

Template-based modelling has become a standard tool used by researchers to predict the three-dimensional structure of proteins with many programs being widely available (Biasini et al. 2014, Kelley et al. 2015). However, the ability to predict structures with the correct biological ligand bound has been significantly less explored by the community (Michino et al. 2009). Furthermore, most current methods blindly assume that if there is a ligand bound into a homologous template then that is the true biological ligand (Biasini et al. 2014).

According to the two main interpretations of ligand binding into macromolecules, conformational selection (Boehr et al. 2009) and induced fit (Koshland Jr 1958), the bound form will have an alternative conformation to the unbound form (or ensembles of unbound forms). Therefore, when modeling ligands into predicted structures it is vitally important that one selects a template that was solved with the true ligand bound (or a closely related) analogue (Csermely et al. 2010, Brylinski & Skolnick 2009).

Here, we present two separate pipelines for the purpose of modelling ligands into predicted structures. The first is LigandTemplateFinder which, given either a UniProt ID or sequence file as input, will find the biological ligands associated with that protein and suggest a suitable template from the PDB that can be used in a templated-based modelling program of choice. To achieve this, we have established an in-house reference between the UniProt, PDBe and ChEBI databases to identify structural templates with the biological ligand bound through chemoinformatics methods originally developed for drug discovery.

Written by AS

The second is a pipeline to predict the effects of missense variants on ligand binding via docking. This pipeline analyses the changes in protein-ligand interactions predicted from docking a ligand into normal and missense proteins, which indicates the impact of the missense variant(s). It is possible for the user to input a 3D coordinate file of a protein structure downloaded from PDB database or

predicted by homology modelling software, for instance Phyre2 (Kelley et al. 2015). The target ligand from the selected template will be docked into the query protein structure by homology-based docking with AutoDock Vina (Vina) (Trott & Olson 2009). While both pipelines are currently independent from one another, in future they could be integrated into a program such as Phyre2 to enable automatic modelling of ligands in predicted models.

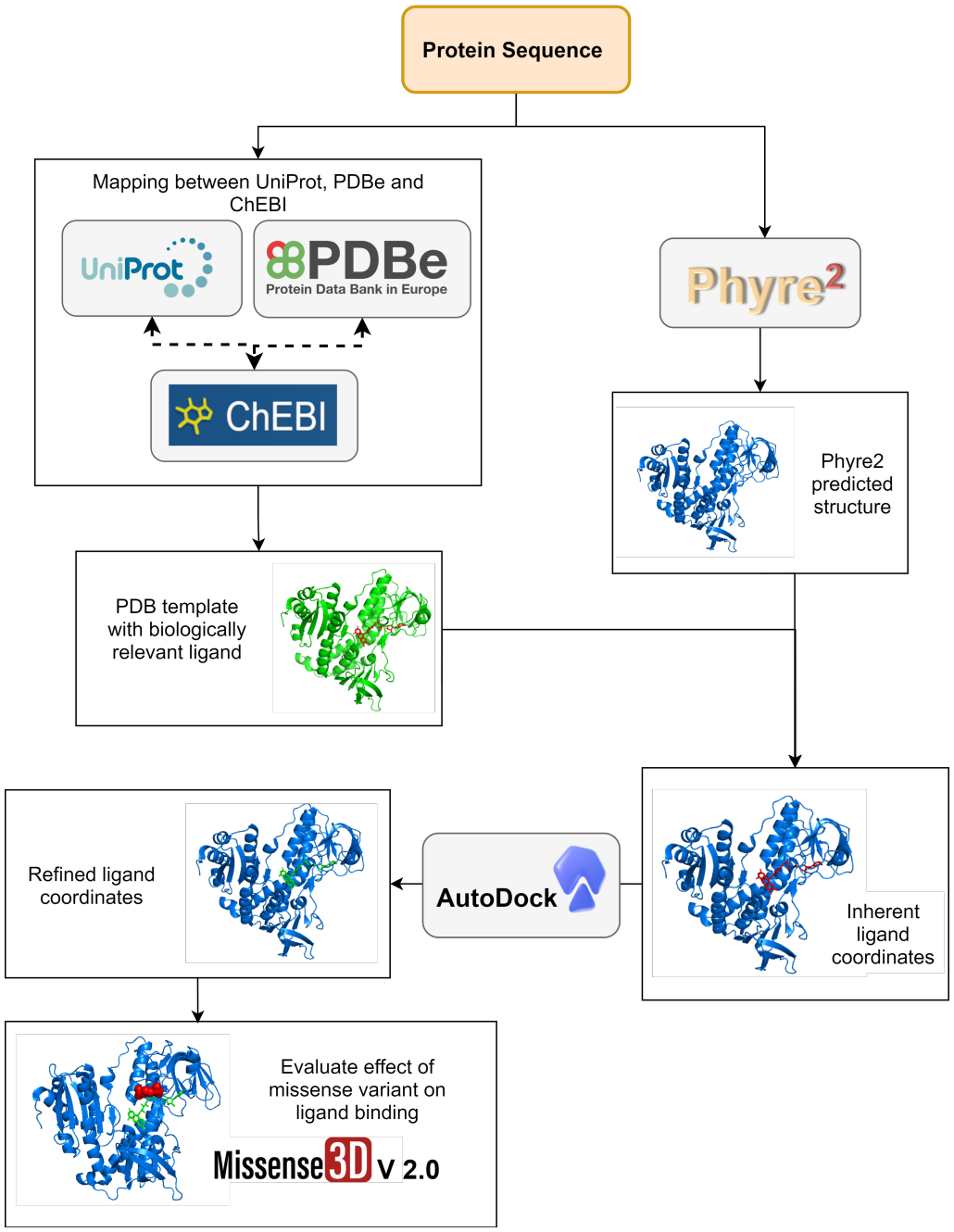


Figure 1: *Figure by SK and CH*: Overview of the eventual aims for our project. Adapted from Prof. Sternberg's research grant: "Enhancing the Phyre protein modelling resource: prediction of ligand binding and the impact of missense variants"

1.1 Background

1.1.1 Template-based structure prediction

Written by CH, SK

Tempalte-based modelling (also known as comparative and homology modelling) relies on the principle that two proteins with highly similar amino acid sequences will share similar three-dimensional (3D) structures. Template-based modelling employs evolutionary related proteins with an experimentally resolved structure (template) to predict the 3D structure of a query protein (target) (Biasini et al. 2014). A typical approach to template-based modelling involves: (i) searching for templates with known 3D structures, (ii) selecting structures to be used as templates, (iii) alignment of target and template sequences, (iv) building model for target sequence from alignment information and (v) model evaluation (Fiser 2010).

On the other hand, template-free modelling techniques enable protein structure prediction by relying only on the sequence (or more commonly multiple aligned sequences). Whilst there has been considerable advancement in the field of template free structure prediction using Deep Learning (Greener et al. 2019, Yang et al. 2020), notably AlphaFold2 (Jumper et al. 2020), these methods are currently unable to consider modelling ligands docked to proteins. Conversely, there is a wealth of ligand pose information in the PDB that can be exploited using template-based methods.

Phyre2 is a state-of-the-art automated approach for the prediction and analysis of protein structure developed by Kelley et al. (2015). It is one of the most widely used template-based modelling tools for protein structure prediction with over 1,000 protein submissions per day. Specifically, Phyre2 builds a hidden Markov model (HMM) of the query sequence based homologous sequences identified through a PSI-BLAST search. The HMM constructed is screened against a database of HHM with known structures using HMM-HMM comparison (Kelley et al. 2015).

1.1.2 Databases of sequence, protein structure and ligands

Written by SK

Chemical Entities of Biological Interest (ChEBI, www.ebi.ac.uk/chebi/) is a database of "small" chemical compounds, which at the time of writing consists of 58,829 entities (Release 197, 1st of March 2021) (Hastings et al. 2016). ChEBI also incorporates ontology information, with two major sub-ontologies; the chemical entry ontology, which classifies the compounds based on their chemical structure, and the role ontology, reflecting the chemical and/or biological activities of the entry (Hastings et al. 2016). ChEBI currently provides links to approximately 30 million UniProt entries, which includes primarily chemical participants of enzyme catalysed-reactions and cofactors.

UniProt Knowledgebase (UniProtKB, www.uniprot.org) is a database of protein sequences and functional annotation, containing approximately 190 million eukaryotic, bacterial, archaeal and viral sequences (*UniProt: The universal protein knowledgebase in 2021* 2021). UniProtKB includes manually

curated UniProtKB/Swiss-Prot entries and unreviewed UniProtKB/TrEMBL entries, which are automatically annotated (*UniProt: The universal protein knowledgebase in 2021* 2021). Functional information provided by UniProtKB includes description of the protein's general function, catalytic activity, cofactors, binding sites, pathways, gene ontology and more. The catalytic activity subsection provides the chemical reactions catalysed by enzymes, using the Rhea knowledgebase of biochemical reactions(*UniProt: The universal protein knowledgebase in 2021* 2021, Morgat et al. 2020). Specifically, Rhea employs the ChEBI database to provides information on biochemical reactions, participating molecules and their chemical structures (*UniProt: The universal protein knowledgebase in 2021* 2021). The cofactor subsection of UniProtKB provides information on substances required for catalytic activity enzymes, such as metal cations, and links to the associated ChEBI identifiers. Compounds such as ATP, NAD, FAD that are substrates of the enzyme-catalysed reactions are found in the catalytic activity subsection rather than in cofactors (*UniProt: The universal protein knowledgebase in 2021* 2021).

The Protein Data Bank in Europe-Knowledge Base (PDBe-KB, https://pdbe-kb.org) is a resource for macromolecule structural data with functional and structural annotations (Varadi et al. 2020). The annotations of the macromolecule structures are literature extracted, manually curated and/or predicted computationally. (Varadi et al. 2020). Each entry contains information related the structure's function and biology, ligand environments, experiments, validation, citations to primary publication(s) and cross-references to associated UniProtKB identifiers. As of 2019, there were 150,000 PDBe structures linked to over 47,500 unique UniProtKB entries with about 12,000 PDBe structures added every year (Varadi et al. 2020).

Some features such as the instant coordinates of ligands bound to the PDB structures are only available RCSB PDB database (Berman 2000). Whilst the actually coordinates in RCSB PDB and the PDBe are the same, both databases offer different metadata and mappings for each structure. In the cases where data from RCSB PDB is used instead it is referred to accordingly.

1.1.3 Chemical similarity

Written by SK

A central principle in chemoinformatics, referred to as the "chemical similarity principle", states that compounds with similar structures will exhibit similar biological activities. Quantifying chemical similarity between a known reference compound and a query, enables the prediction of molecular behaviour from structurally related compounds. Chemical similarity is most commonly calculated by employing molecular fingerprints, which are a digital representations of a compound's structure. Several different types of fingerprints exist, each reflecting different aspects of a molecule (Cereto-Massagué et al. 2015). Interestingly, an increase in the complexity of the fingerprints seldomly results in a similarity performance gain and even though three-dimentional (3D) fingerprints exist, two-dimensional fingerprints (2D) are currently the standard method for chemical similarity searching (Muegge & Mukherjee 2016, Cereto-Massagué et al. 2015). The three most predominant approaches to 2D fingerprints are substructure keys, topological and circular fingerprints.

Substructure keys-based fingerprints are encoded in a binary vector of a set number of bits, where "1" represents the presence and "0" the absence of a predefined structural group from a list of structural keys, as shown in Figure 2. Molecular ACCess System (MACCS) keys with 166-bit length are among the most widely used structural keys fingerprints (Durant et al. 2002, Skinnider et al. 2017). Even though these are relatively short bit strings, they include most of the interesting molecular features explored in chemoinformatics (Cereto-Massagué et al. 2015).

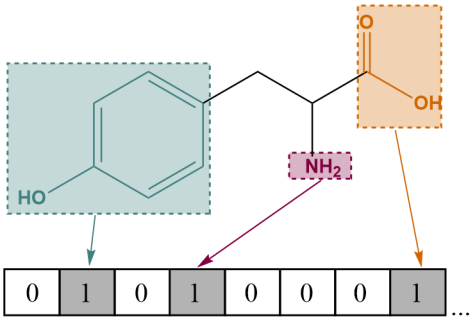


Figure 2: Hypothetical structural key-based fingerprint for tyrosine, redrawn from Cereto-Massagué et al. (2015) with ChemDraw™.

On the other hand, topological and circular fingerprints can capture more rare molecular fragments. However, as these fingerprints are generated using a hashing algorithm, they suffer from bit collisions, where one bit may correspond to more than one molecular fragment, as shown in Figure 3. A bit cannot be traced back to a specific structural group or feature and therefore are not suitable for substructure searching (Cereto-Massagué et al. 2015).

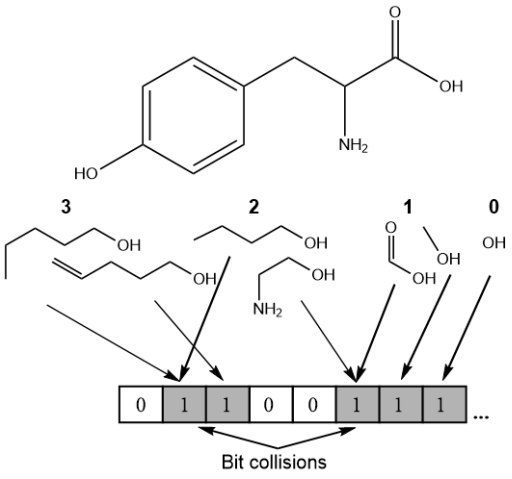


Figure 3: Hypothetical topological fingerprints for tyrosine, generated by following a path up to a given number of bonds and hashing these paths to produce molecular fragments. A similar approach is used for circular fingerprints, however, rather than following paths in the molecule, these fingerprints look at the environment of each atom within a certain radius. The figure was redrawn from Cereto-Massagué et al. (2015) using ChemDraw™.

In chemoinformatics, the *defacto* standard for measuring chemical similarity with 2D molecular fingerprints is the Tanimoto coefficient (please refer to methods section "Similarity search algorithm") (Cereto-Massagué et al. 2015, Rácz et al. 2018). Several free online software is available for chemical similarity searches using molecular fingerprints, some of the major databases enabling similarity searches are summarised below:

- PubChem: enables similarity search between chemical structures in PubChem database using the PubChem fingerprint (a structural keys-based fingerprint of 881-bits length). Chemical similarity is measured using the Tanimoto coefficient (Kim et al. 2021, Bolton et al. 2008).
- ChEBI: finds other compounds within the ChEBI database that resembles query structure using Tanimoto similarity and OrChem Fingerprints (approximately 800-bits long combining both structural keys-based and hashed fingerprints) (Hastings et al. 2016, Rijnbeek & Steinbeck 2009).
- ChEMBL: returns compounds similar to query with at least 85% Tanimoto similarity using circular fingerprints implemented in FPSim2 python module (https://chembl.github.io/FPSim2/) (Gaulton et al. 2017).
- ZINC Database: performs similarity search to identify close analogues within the molecules in ZINC database using a combination of four fingerprints: sFP (substructure fingerprints), ECFPs (circular fingerprints), MQNs (molecular quantum numbers) and SMIfp (SMILES fingerprints). Chemical similarity is quantified with the city-block (also known as Manhattan) distance (Sterling & Irwin 2015, Awale & Reymond 2014).

1.1.4 Ligand docking

Written by AS

Protein-ligand binding is driven by the noncovalent interactions between ligand and nearby protein residues including van der Waals contacts and hydrogen bonds. The driving forces can be quantified by the change in Gibbs free energy (delta G) of the system. A negative delta G indicates that the bound protein-ligand complex is more stable than free protein and ligand, therefore, the protein-ligand binding process can occur spontaneously (Du et al. 2016).

Ligand binding modes and affinities can be predicted by using structure-based computational approaches which are faster and cheaper than using experimental techniques. One of the computational approaches is protein-ligand docking which employs molecular dynamics simulations to identify the most favourable conformations of the protein-ligand complex and the corresponding binding affinities. This method has been widely used in research for over a decade, especially in the field of drug discovery. As a result, several protein-ligand docking software tools have been developed and are available free of charge (Du et al. 2016), such as AutoDock (Morris et al. 2009). Each software utilises a search algorithm for ligand conformations and scoring function to estimate the binding affinity.

AutoDock Vina ((Trott & Olson 2009)), a new separate version of AutoDock, is an automated docking tool used to predict the best binding modes of the ligand in a specific area (grid box) of the protein and provides the binding affinity of each ligand conformation. AutoDock Vina implements a flexible-ligand algorithm which treats the protein structure as a rigid object and considers only the flexibility of the ligand. The scoring function of Vina uses a combination knowledge-based and empirical approaches which calculate the delta G of binding from the delta G of attraction, repulsion, hydrogen bonding, hydrophobic interactions, and ligand torsion.

The grid maps for each atom type in the empty region of the specified search space which is not occupied by the protein atoms are pre-calculated. Then ligand conformations are searched stochastically by gradient-based local search in the grid box. The transformation of ligand in the searching process includes translation of ligand position in X, Y and Z axis, rotation of ligand molecule and movement of ligand rotatable bonds.

Vina provides a pool of predicted ligand conformations instead of presenting only the best binding mode. This is because the software is not perfect, and the ligand conformation generated by docking is just a prediction not the actual ligand conformation. Therefore, the best binding mode is not always the true answer.

1.2 Related work

Written by CH

There are many approaches currently in use that aim to predict ligand binding poses or ligand binding sites (Govindaraj & Brylinski 2018, Zhao et al. 2020). SWISS-MODEL (Biasini et al. 2014) is a popular template-based structure prediction server that has the ability to predict and model biological ligands into a 3D structure. The program first identifies the biological ligands associated with a protein by a conservative homology transfer approach so small molecules that are observed in the templates of its manually curated template library. All ligands in the library are categorised as: "(i) relevant, non-covalently bound ligands, (ii) covalent modifcations or (iii) non-functional binders (e.g. buffer or solvent)". Non-covalently bound ligands will be considered for the model if the binding site residues are highly conserved and there are no steric clashes (Biasini et al. 2014). COACH-D (Wu et al. 2018) takes a similar approach but refines the ligand binding poses using AutoDock Vina (Trott & Olson 2009).

3DLigandSite (Wass et al. 2010) is similar to our approach in that it relies on the superimposition of ligands from homologous structures to predict binding residues for a query sequence or structure. Only ligands from the top 25 hits as predicted by MAMMOTH (Ortiz et al. 2002) are considered. Inherited ligands are clustered and the largest cluster is used to find the general area of the binding site. FunFOLD2 (Roche et al. 2013) relies on similar principles but can only predict one binding site per target. Whilst both the program do return the clusters for the ligands in the cluster, it is often more useful for biologists to have one high confidence pose returned.

Crucially, in the above methods there are no automatic checks that the ligands found in the templates are the biological ligands. This makes it very challenging for a life science user to identify which template

is most suitable for modelling the true ligand and potentially leading to false positives. This is particularly true when the ligand in the PDB is not the true biological ligand.

Furthermore, with the exception of COACH-D (Wu et al. 2018) there is no attempt at refining the ligand poses in the query binding sites which may have different properties to those in the templates (SWISS-MODEL rejects ligands from templates that cause a steric clash in the predicted models all together (Biasini et al. 2014)). This is done to be conservative but means that any conformation rearrangement of the binding sites of the query protein results in ligand pose information being lost.

None of the methods mentioned provide a all-in-one pipeline for the prediction of ligand binding (using high quality ligand mappings) and the effect of missense variants on ligand binding starting with only a sequence. In this work, we present a proof of concept for a future program that can predict ligand binding using high quality sequence annotations and then predict the impact of missense mutations.

2 LigandTemplateFinder

2.1 Methods

2.1.1 LigandTemplateFinder overview

Written by SK, work by CH and SK

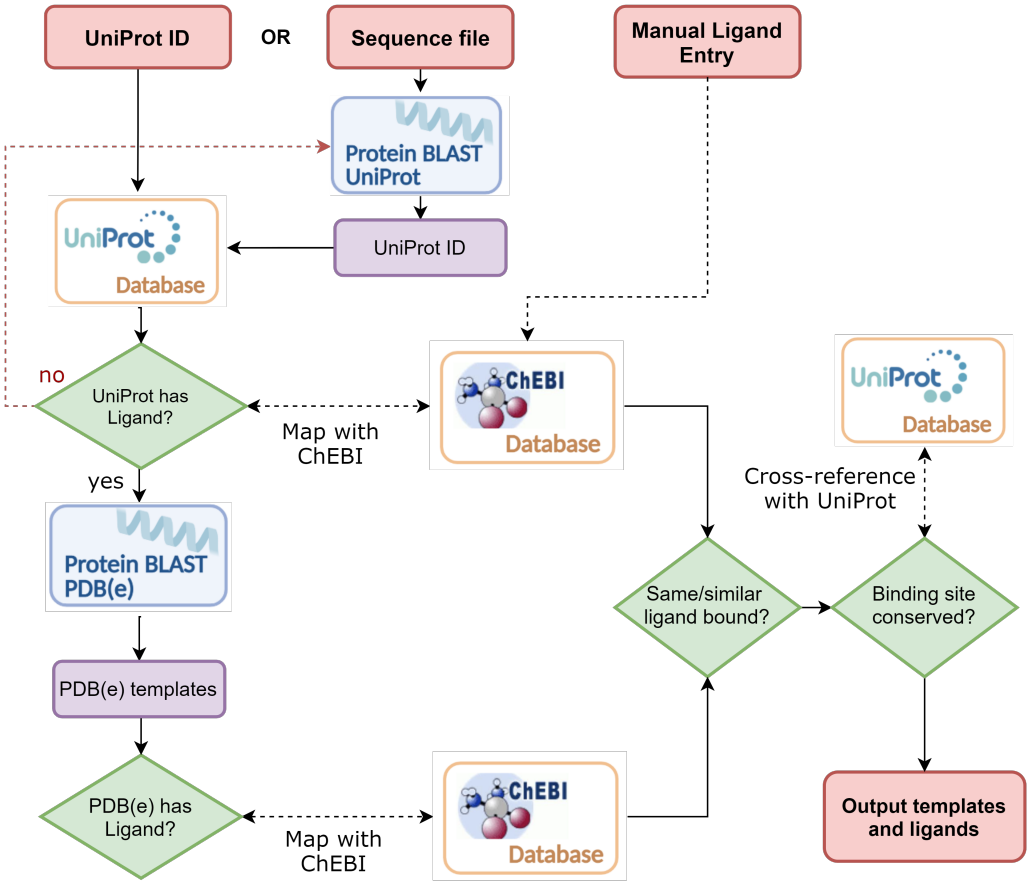


Figure 4: *Figure by SK and CH*: True ligands that are associated with either a UniProt or query sequence are obtained with mappings to the ChEBI database. Alternatively, the user can specify ligands known to be associated with the query. Valid template candidates are found by BLAST searching the query against the PDB. If either the true ligand or analogue is bound to the template structure and at least 3 binding residues are conserved in the query then the template is proposed.

Crystal structures often have ligands bound to their catalytic site, providing valuable information on the protein's biochemical processes. However, in several protein structures, the ligand-binding site is occupied by an inhibitor or synthetic analogue rather than the biological ligand (Biasini et al. 2014). Herein, we propose a novel approach, the LigandTemplateFinder, for employing the links between UniProt, PDBe and ChEBI to identify PDBe templates containing the true biological ligand using the concept of chemical similarity (*UniProt: The universal protein knowledgebase in 2021* 2021, Varadi et al. 2020, Hastings et al. 2016). A diagrammatic overview of the LigandTemplateFinder pipeline is shown in Figure 4.

To start with, the user inputs a UniProt identifier (ID) or a protein FASTA file. An in-house mapping between UniProt and ChEBI has been established to retrieve the ChEBI IDs of the ligands and cofactors associated with the UniProt entry. In the case where the sequence is provided in a FASTA format, a protein BLAST (BLASTP) search against the entries of UniProtKB is performed to identify homologous UniProt entries. If ligands can not be retrieved from the UniProt sequence, the BLASTP search is repeated to identify a UniProt entry with ligand information. Additionally, the user has the option to manually provide the ChEBI ID of a ligand of interest that may not be present in our mapping.

In the next stage of our pipleine, a BLASTP search against the entries of the PDBe database is performed to identify homologous PDBe templates with bound ligands. The PDBe REST API is employed to extract the ChEBI IDs for the chemical components in the PDBe structures. A PDBe template is considered by our pipeline only if it contains at least three conserved residues in the query-template alignment.

Finally, the similarity between the UniProt and PDBe ligands is calculated using chemical information extracted from ChEBI, to identify whether the ligand bound to the PDBe structure is the true biological ligand or a structurally related molecule.

2.1.2 Mapping between UniProt, PDBe and ChEBI

Written by SK, work by SK

The in-house cross-referencing linking UniProt to ChEBI was established using Python (version 3.7.9) in the Oracle Linux Server (release 7.4). The mapping database between UniProt and ChEBI (version 2021_01) was downloaded from the European Bioinformatics Institute's (EMBL-EBI; http://www.ebi.ac.uk) public file transfer protocol (FTP) folder at ftp://ftp.ebi.ac.uk/pub/contrib/UniProtKB/for_chebi/ in a text (.txt) format.

Each entry in the mapping database contained a UniProt ID and a single ChEBI ID corresponding to a molecule that was either related to the catalytic activity of the enzyme or a cofactor. The database consisted of 146,254,023 entries with a total of 22,798,807 unique UniProt IDs and 8,212 unique ChEBI IDs. The key statistics for the database are shown in Figure 5 .

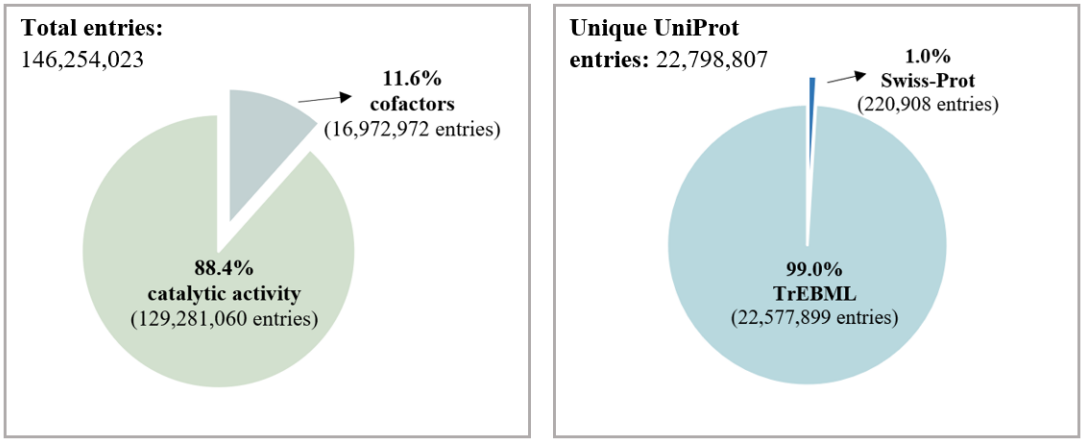


Figure 5: Key statistics for UniProt to ChEBI mapping database, which contained a total of 146,254,023 entries with 22,798,807 unique UniProt IDs.

The database was converted into a Python dictionary, in which the keys were the unique UniProt IDs and the values were a list of every ChEBI ID associated with the respective sequence. This split into 118,227 dictionaries based on the first four characters of the UniProt IDs and saved as pickle files in the Linux environment. This approach of using a hash map format enabled retrieving the ChEBI IDs associated with the enzymatic activity of a UniProt sequence within milliseconds to maximum 5.4 seconds (tested on 2,000 UniProt IDs).)

The chemical components and binding site residues in the PDB structures are retrieved through the PDBe REST Application Programming Interface (API) (Varadi et al. 2020) (*work done by CH*). The mapping between ChEBI and PDB chemical components is obtained through the compound mapping function of the API, which utilises the PDB Chemical Components Dictionary to cross-reference between the PDB and ChEBI (Westbrook et al. 2015). Approximately 5,005 mappings between PDBe and ChEBI are currently available.

As water molecules are used as solvents in PDB structures and due to the uncertainty in resolving the position of hydrogen ions the following molecules, water (CHEBI: 15377), hydron (CHEBI:15378), proton (CHEBI:24636) and dihydrogen (CHEBI: 18276), were excluded from the list of ligands and cofactors (Tyzack et al. 2018).

For a given sequence, the true biological ligand will be contained within UniProt "catalytic activity" section. Due to inconsistencies in the terminology of a "cofactor" between UniProt and PDBe, where a molecule such as ATP is indicated as a cofactor in the PDBe structure but is listed under the "catalytic activity" section of UniProt, we have established a list of organic cofactor-like molecules referred to as "catalytic activity/cofactors". This included molecules such as ATP (CHEBI:15422), NADH (CHEBI:57945), FAD (CHEBI:16238) and compounds with a 100% ChEBI similarity score to them, obtained using the BioService python package (ChEBI module) (Cokelaer et al. 2013). A full list of the molecules under the section of "catalytic activity/cofactors" can be find in the Appendix Table 3.

Chemical components of PDBe structures matched with compounds from our UniProt/ChEBI mapping will be returned as under one of the following categories:

- "catalytic activity": PDBe ligands matched with molecules under the UniProt "catalytic activity" section. These will be also referred to as "UniProt ligands".
- "cofactors": PDBe chemical components matched with substances under the UniProt "cofactor" section.
- "catalytic activity/cofactors": molecules bound to PDBe structures match with compounds in the UniProt "catalytic activity" section, which could potentially be cofactors.

Written by CH, work by CH

BLASTP searches against the PDB was conducted by the manual creation of a local BLAST database. The database can be easily updated using information from https://www.rcsb.org/downloads/fasta. BLASTP searches against the UniProt database was performed using an in-house database that is updated regularly.

2.1.3 UniProt binding sites

Written by SK, work by SK

Binding site information was extracted from UniProt using the BioServices Python package (UniProt Module) (Cokelaer et al. 2013). The location of binding sites on the protein sequence and the description of the interaction made, including the identity of the compound involved in the interaction where available, are retrieved from UniProt's "Function" section. Specifically, information from the following subsections is extracted:

- "Active site": position of single amino acids directly involved in the catalytic activity of the enzyme,
- "Binding Site": location of a residue interacting with another chemical entry such as a "physiological ligand",
- "Metal binding": binding sites for metal ions, and
- "Nucleotide binding": region of protein sequence where nucleotide phosphates bind (https://www.uniprot.org/help/f (*UniProt: The universal protein knowledgebase in 2021* 2021).

UniProt binding sites are provided for (i) the input UniProt ID and (ii) for each of the PDBe templates returned from the BLASTP search against the PDBe database. The latter is an optional feature of our pipeline. It is achieved by employing the Structure Integration with Function, Taxonomy and Sequence (SIFTS) module of the PDBe REST API to map between PDBe structures and UniProt (Velankar et al. 2012, Dana et al. 2019). This enables the user to cross-reference between the binding site provided our pipeline's query-template alignment, retrieved using PDBe REST API *(work done by CH)*, and the site information from the UniProt sequence(s) associated with that PDBe template.

2.1.4 Similarity search algorithm

Written by SK, work by SK

Chemical similarity was calculated using the Tanimoto coffecient with MACCS keys substructural fingerprints of 166-bits. The Tanimoto coefficient is calculated as shown in equation 1, where c corresponds to the number of common bits set to "1", a is the total number of bits in molecule A and b represents the total number of bits in molecule B.

$$Tc = \frac{c}{a + b - c} \tag{1}$$

Chemical similarity calculations were performed in Python RDKit environment (Landrum 2016). Chemical similarity is calculated between the molecules associated with the query UniProt entry and the chemical components of the PDBe templates returned from the BLASTP search, to identify whether any of the ligands bound in the PDB structures are the biological ligand or a close analogue.

The mapping database between UniProt and ChEBI is employed to extract the ChEBI IDs associated with the query UniProt entry. The PDBe REST API Compound Mapping is employed to obtain the ChEBI IDs of the ligands bound to PDB structures. The BioServices Python package is employed to programmatically access the ChEBI web services and retrieve the ChEBI ASCII Name and the isomeric simplified molecular-input line-entry system (SMILES) strings from the ligands ChEBI IDs (Cokelaer et al. 2013). Specifically, SMILES strings are chemical notations to reperesent compounds and reactions in a compact format.

The RDKit Python module is deployed to generate MACCS keys substractural fingerprint representations of 166-bit length from the ligands SMILES strings. The Tanimoto similarity between the molecular fingerprints of each UniProt and PDBe ligand is calculated and UniProt/PDBe ligand pairs with a similarity score above a given threshold are returned. The default setting is 0.70 Tanimoto similarity, which can otherwise be defined by the user.

2.1.5 Identification of the true sequence ligands in case of no mappings

Written by CH, work by CH

The true biological ligands associated with a query sequence are determined in 3 possible ways, as is seen a the top of Figure 4. (i) The query sequence is a UniProt identifier that has experimental mappings to ligands in the ChEBI, (ii) the user is searching with an unannotated sequence file/the query UniProt ID is not associated with any ligands so ligands are transferred by homology after a BLASTP search of the UniProt database or (iii) the user can manually add ChEBI IDs to the sequence if known ligands are not present in our mappings. The last case is useful for when the true ligand is known but not present in our mappings (for example heme ligands associated with haemoglobins).

In the case that the user wishes to predict a structure on a un-annotated sequence, or the query

UniProt ID does not have any ChEBI mapping, our program will try and perform ligand transfer by sequence homology. We perform a BLASTP search against the UniProt database to find sequence homologues for which we have experimental ligand mappings to the ChEBI. If the user has searched for a sequence within in the UniProt database (i.e. the percentage identity is 100%) then all the ligands are transferred. Otherwise, the catalytic ligands are inferred by taking the consensus ligands from the top 3 hits from the BLAST search as long as they are all above 50% identity. The same if performed with cofactors but instead a threshold of 90% is used (see Results for justifications).

To test our hypothesis, further analysis was performed to establish the reliability of ligand transfer by homology. For this, we took inspiration from a similar analysis by Devos & Valencia (2000) which looked at enzyme function. 1,000 UniProt sequences were randomly selected and a BLASTP search was performed against the UniProt database to find homologues. All duplicate sequence-sequence pairs were removed and then sorted by percentage identity. The false and positive rates for the success of transferring the ligands between the query and hits was calculated for all percentage identities in bins of 10%. A successful transfer in this case was defined as the two sequences sharing at least one ChEBI ID between them from their catalytic ligands.

Homologous members of the same protein family will tend to share many ubiquitous ligands, even at low percentage identity. For example, ATPases might act on different substrates but will all be associated with ATP. Therefore, this analysis will return misleading positive hits for sequences of low homology, even though the biological substrate of interested might not be correctly identified.

Therefore, we repeated the analysis but also ignored ligands from a list of "common" ligands that we wanted to ignore for the sake of assessing our methods ability to identify the meaningful substrates. The full list is provided in Appendix Table 3 and includes ligands that are common across many families of proteins such as ATP, NADP and water as well as ChEBI entries of little value for ligand docking.

2.1.6 Identification of valid templates with correct ligands bound

Written by CH, work by CH

Identification of homologous templates with suitable ligands bound starts with a BLASTP search of the query sequence against all protein sequences in the PDB. Any structure with a BLASTP E-value greater than 0.001 or a resolution more than 2.5 Å were not considered. These two parameters are optional and can be changed by the user to be more or less conservative.

Before we can return a template we need to ensure that the binding site residues for the ligand we are suggesting is conserved in the query sequence. The binding site residues that coordinate with the ligand in the template structure are first retrieved from the PDBe. Using the query-template alignment generated from our BLAST search of the PDB, we cross-reference the residues in the alignment to ensure that the query actually aligns with the binding site and that the residues are sufficiently conserved. Our program uses a user defined cut-off value (defaulted to a conservative value of 3, as is seen in Biasini et al. (2014)) of the number of conserved residues between the query and the template before we suggest a ligand/binding site hit.

Provided that a binding site is deemed valid and returned, the program displays an alignment of just the binding site residues in both the query and template sequence along with position numbers (both in the sequence and according to the PDB file) and a match column to aid visual inspection (See results). A list of binding site residues that do not align with the query are returned, this includes residues that are out of the alignment and those on different chains. We also display a list all of the interacting partners that mediate the binding of the ligand but are not standard amino acids that can be mapped to the sequence (e.g. non-standard amino acids in the template sequence and water molecules).

All candidate templates, and binding sites within, are defined as custom Template and BindingSite python objects respectively. All information generated during analysis is stored within these objects allowing the user to easily interrogate and analyse their results as desired. We have created an easy to use print function that will display all the suggested templates (and information mentioned above) in the terminal that we would recommend for most users (See results). A Least Recently Used (LRU) caching procedure (Jelenković & Radovanović 2004) was used extensively throughout our program (e.g. cacheing recently used PDBe to ChEBI mappings, SMILES and molecular fingerprints) which greatly increased the throughput (especially when you consider the same ligands are usually found in templates of close homology).

2.1.7 Inheritance of ligand coordinates

Written by CH

Once a homology model has been predicted using a template we have identified, we can be begin to model the ligand into the predicted structure. This has not been automated at this stage as the process is highly depending on the scenario and the fact that Phyre2 cannot be automatically queried via command line.

Depending on the program used, some template libraries do not retain the coordinate system found in the original PDB file. If the template and predicted structure follow the same coordinate system, then an instance MOL2 file containing the ligand coordinates found in the template structure can be inherited into the predicted model. This file can either be manually retrieved from the template page on the PDB or via a Python script. Furthermore, it is recommended to keep the protein and ligand coordinates in separate files for the purposes of docking.

If the structure prediction program changes the coordinate system for the protein (e.g. Phyre2) then the whole template PDB file has to be loaded into PyMol, then predicted structure is superimposed onto the corresponding domain from the template structure.

If desired (e.g. in the case that simply inheriting the coordinates causes a steric clash), ligand poses can be refined using AutoDock Vina (*performed by AS and OW* - see docking section).

2.1.8 LigandTemplateFinder input

Written by SK, work by CH and SK

LigandTemplateFinder takes as an input either a UniProt ID or the query amino acid sequence in a FASTA format. The documentation for the parameters of LigandTemplateFinder is presented in Table 1.

Table 1: LigandTemplateFinder parameters

Parameter	**Description**
uniprot_id	UniProt identifier of query sequence (*default = None*)
query_file	Query amino acid sequence in FASTA format (*default = None*)
e_value_thres	E-value (Expect value) threshold for protein BLAST search against the PDBe dataset (*default = 1e-3*)
resolution_thres	Resolution threshold for protein BLAST search against the PDBe dataset (*default = 2.5*)
similarity_thres	Threshold for Tanimoto similarity between reference ligands in UniProt and ligands bound to PDBe templates returned from the protein BLAST search (*default* $\geq$ *0.70*)
site_conservation_thres	Minimum number of conserved residues in query-template alignment for PDBe template to be returned (*default = 3*)
uniprot_binding	When set to True will return binding site information (Active site, Metal binding, Binding site and Nucleotide binding) from the UniProtKB Function section (*default = False*)
manual_ligands	Allows user to add manually the ChEBI IDs of ligands if not present in our mapping (*default = None*)
max_templates	Maximum number of PDBe templates returned from protein BLAST search (*default = 100*)
mode	When set to 1 will look only for one PDB template with ligand per query (*default = 0*)
verbose	Controls verbosity when performing computations (*default = False*)

2.1.9 Evaluation of our approach

Written by CH, work by CH

In some examples, we predicted the coordinates of know structures with ligands bound, excluding the real structure from the database of possible templates. Predicting the poses of real targets is the best way we can assess the reliability of our method in the absence of entering a real blind trail like CASP.

At various points in our evaluation we use a randomly selected dataset of 1,000 UniProt sequences for which we have ChEBI mappings (*selected by SK*). This dataset was used to do large scale evaluations of our pipeline.

2.1.10 Evaluation of similarity search

Written by SK, work by SK

To evaluate the reliability of our pipeline's similarity search algorithm (Tanimoto similarity with MACCS keys substructural fingerprints) its performance was compared to the established similarity search provided by ChEBI (Tanimoto similarity with OrChem fingerprints). Specifically, ChEBI IDs were extracted for every chemical component in the PDBe database and every molecule in "catalytic activity" section of UniProt available in our in-house mapping database between UniProt and ChEBI. A random subset of 1,000 chemical compounds from UniProt was extracted and the ChEBI similarity search was employed to identify PDBe ligands with 90-100% Tanimoto similarity to UniProt compounds. The chemical similarity between these UniProt/PDBe compound pairs was recalculated with our pipeline's similarity algorithm. This process repeated for UniProt/PDBe pairs with 50-60% and 20-30% ChEBI similarity score.

2.2 Results

2.2.1 LigandTemplateFinder output example

Written by CH, SK, work by CH, SK

An example output of our program with query UniProt ID of P0AGB0 is displayed in Figure 6. The selection of the best template is often highly dependent on the intended modelling outcome of the experiment and is to an extend subjective. This is why programs like Phyre2 and SWISS-MODEL return a list of suggested templates that can be used (Kelley et al. 2015, Biasini et al. 2014). We have followed a similar approach and provided all the information on the templates hits to the user for them to make their decision.

For each template hit we display the UniProt binding sites associated with the query sequence, the template PDB code and chain identifier, the e-value returned from BLAST, the resolution of the template structure, and the percent identity and alignment length between the query and template.

For each conserved binding site for which we have managed to match a chemical compound from UniProt to, we display the site ID, a site description and the ligand type (please refer to the "Mapping between UniProt, PDBe and ChEBI" methods section). Additionally, we are providing the compound name and ChEBI ID associated with the molecules retrieved from both the query sequence and the template structure as well as the similarity calculated between the UniProt compound and that found in the PDB template. We also display a vertical alignment of just the key binding residues to allow the user to assess the conservation of the binding site.

As an optional feature of our pipeline, the user may choose to display the UniProt binding sites associated with the specific PDB template hit in order to cross-reference with the information provided by our pipeline's vertical alignment.

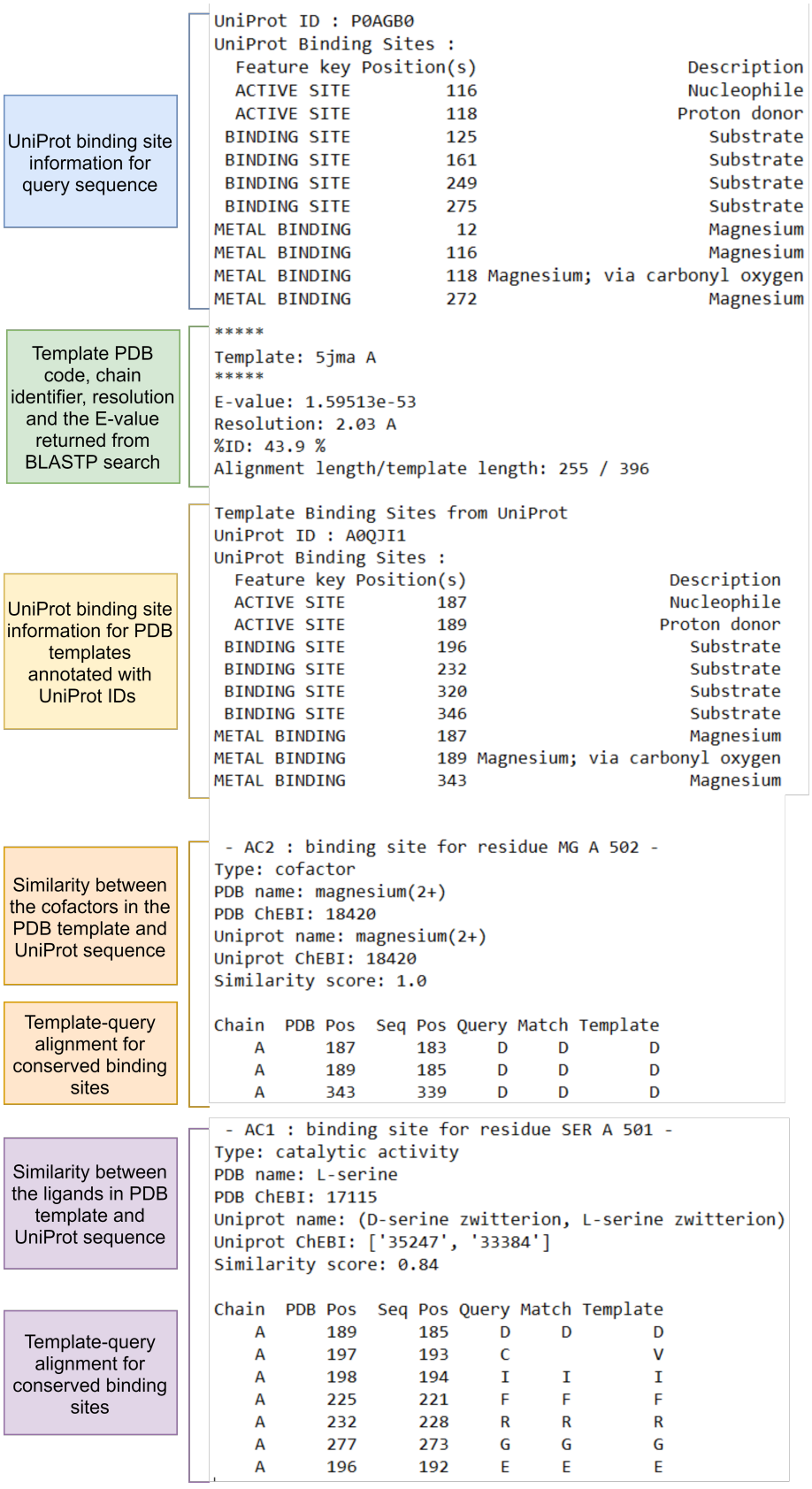


Figure 6: *Figure by SK*. LigandTemplateFinger output example for query sequence with UniProt ID of P0AGB0

2.2.2 Ability to identify templates with known ligands bound

Written by CH, work by CH

A large analysis was performed on 1,000 randomly selected UniProt proteins to test the coverage of our approach (Figure 7). Using our default parameters, we were able to identify valid templates which had starting ligand coordinates for 71.1% of queries. The number of templates returned per query had a mean of 47.5 and median of 11 (Figure 7a). Figure 7b showed that the Tanimoto coefficients between the true biological ligands and those found in the templates fit into two distinct peaks. The first in the range of 0.90-1.00 and the second being 0.80-0.85 Tanimoto similarity. This experiment, which analysed and compared the ligands in over 194,017 template candidates, was performed in under 12 hours on a single machine.

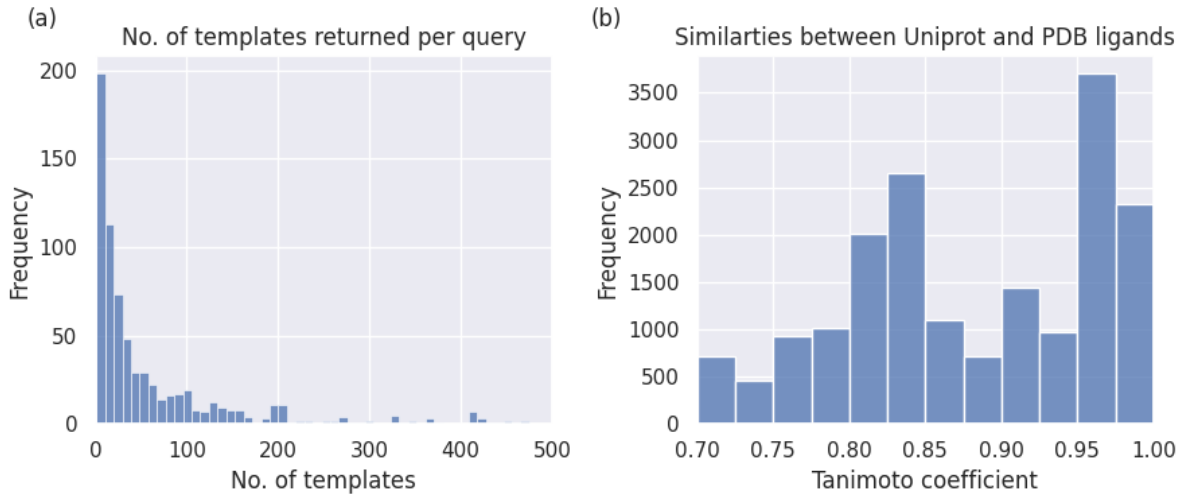


Figure 7: Results from running LigandTemplateFinder on 1,000 randomly selected UniProt sequences. (a) Frequency distribution of the number of valid templates returned per query, all queries returning 0 templates removed from graph (289 instances). (b) Frequency distribution of the Tanimoto coefficients between ligands associated with a query sequence versus those found in the template structure.

Figure 8 shows an example binding site (query Q51945, template 2D4V) which our method identified as being sufficiently conserved. However, 3 of the 8 coordinating residues were incorrectly predicted as being unconcerned when in fact all the residues were, demonstrating that our approach of identifying conserved binding sites might suffer from false negatives.

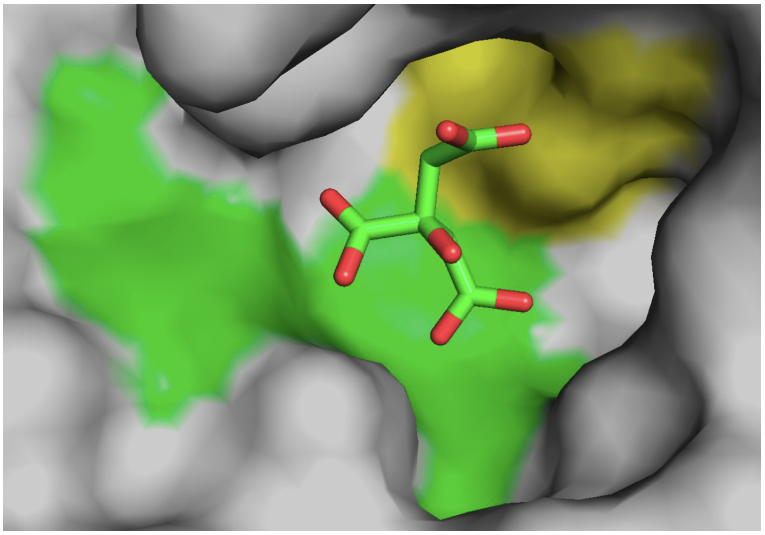


Figure 8: Automatic identification of a preserved binding site in a homology-model. Query sequence: Q51945. Template: 2D4V. Green: Binding residues are identified as conserved in query structure. Yellow: Residues incorrectly predicted as being unconserved in the model (see Discussion).

2.2.3 Ligand frequency in UniProt entries

Written by SK, work by SK

The ChEBI IDs of the molecules under the "catalytic activity" and "cofactor" section of UniProt were extracted from our in-house mapping between ChEBI and UniProt. A total of 146,254,032 ChEBI IDs were retrieved corresponding to 8,212 unique IDs. The frequency of each ChEBI ID within the "catalytic activity" and "cofactor" sections was counted and the top thirty most represented chemical compounds within each section are displayed in Figure 9 and Figure 10, respectively.

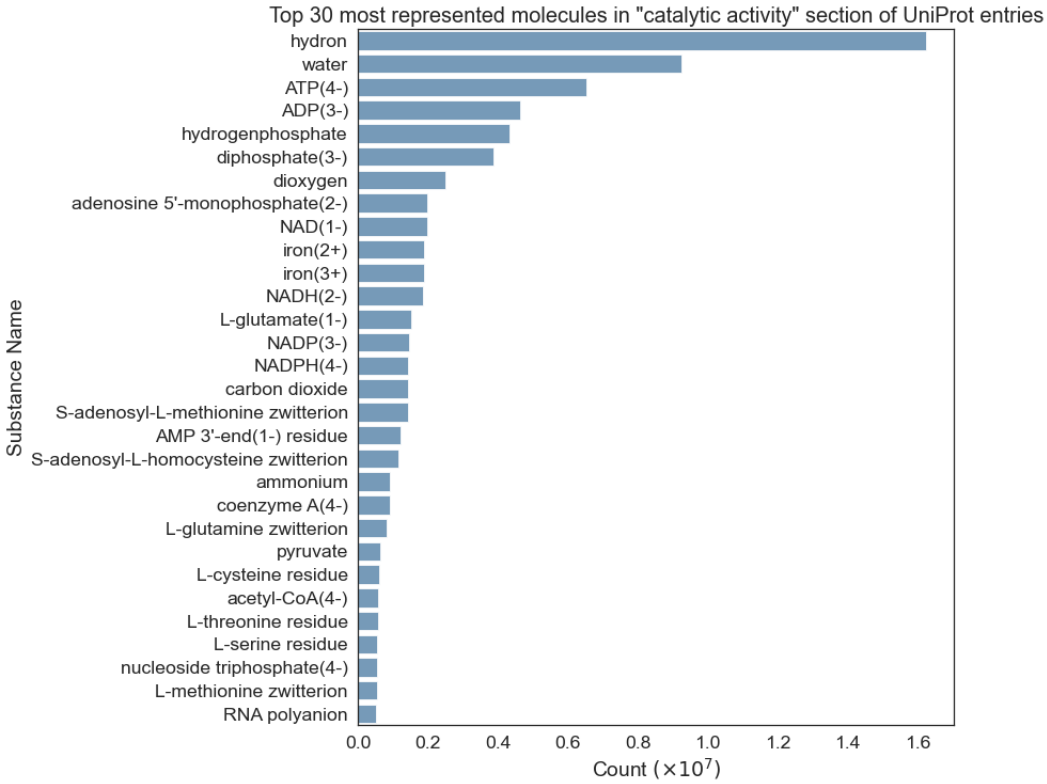


Figure 9: Top 30 most frequent molecules in the "catalytic activity" section of our in-house UniProt to ChEBI mapping. These range from 16,241,068 hydron ions to 511,258 RNA polyanions. The total number of unique molecules in this section is 8,163.

The most represented compounds within the "catalytic activity" section of UniProt are the hydrogen ion and water. As mentioned in the ""Mapping between UniProt, PDBe and ChEBI"" methods section, these compound are excluded from our matching between UniProt molecules and ligands bound to PDB structures. Compounds such as ATP, ADP, NAD etc are participants of the reaction catalysed by the enzyme, however, they often act as cofactors, rather than being the true biological ligands. Therefore, these compounds are returned as "catalytic activity/cofactors" by our pipeline, indicating that the molecules are listed in the "catalytic activity" section of UniProt, but could potentially be cofactors (please refer to Appendix Table 3). For example, in the case of the enzyme Tartrate dehydrogenase/decarboxylase (UniProt ID: Q51945), where NAD is listed under the "catalytic activity" section, however, the solved protein structure (PDB ID: 3flk) confirms that NAD is actually a cofactor.

Additionally, several substances are reported in their ionic or zwitterionic form (such as S-adenosyl-L-methionine zwitterion), which is due to the compound information being extracted from chemical reactions and the PDB structures will usually contain the molecular form of the compounds. Finally, entries such as "L-cysteine residue" reflect amino acid residues in the protein that are modified during catalysis.

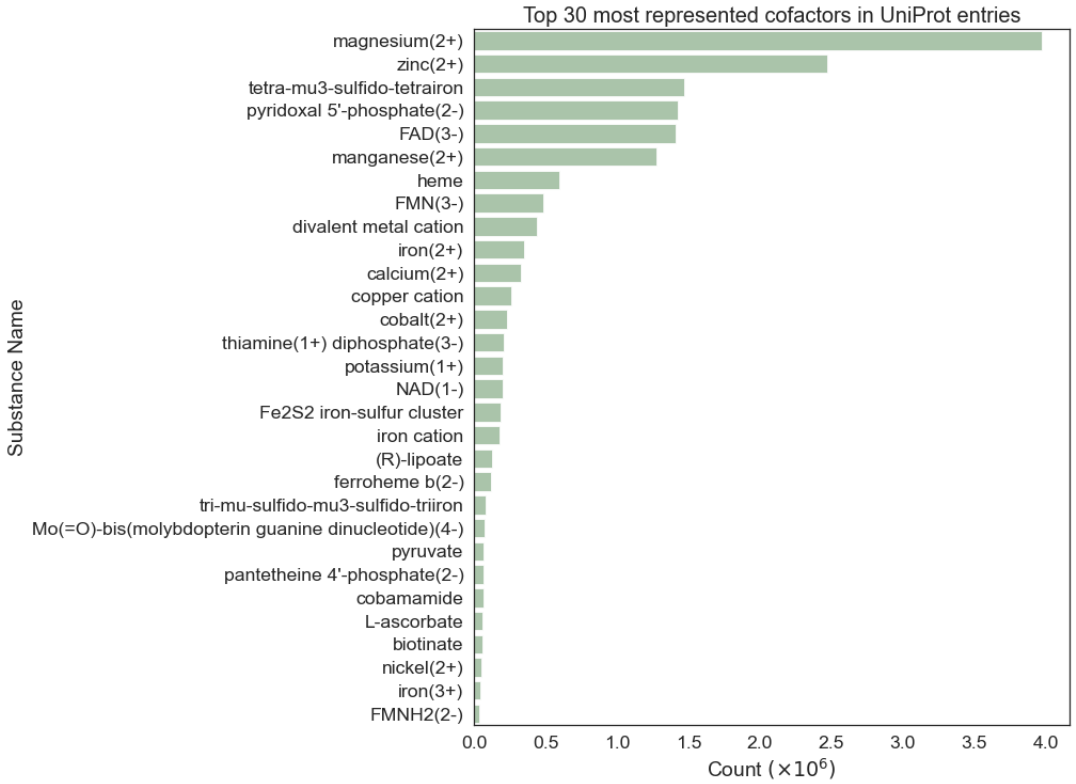


Figure 10: Top 30 most frequent substances within the "cofactor" section of our in-house cross-referencing between UniProt and ChEBI, which range from 3,967,480 magnesium cations to 38,794 $FMNH_2$ dianions. The number of unique substances within the "cofactor" section is 115.

A large majority of the substances within the "cofactor" section are metal ions such as magnesium, zinc and manganese cations. Compounds such as FAD and NAD will usually be listed under the "cofactor" section if they are necessary for catalysis and are not substances of the biochemical reactions.

2.2.4 Evaluation of chemical similarity algorithm

Written by SK, work by SK

The results of the evaluation of our pipeline's chemical similarity search (Tanimoto similarity with MACCS keys substructural fingerprints) against the similarity search provided by ChEBI (Tanimoto similarity with OrChem fingerprints) are shown in Figure 11. A total of 1,327, 733 and 107 UniProt/PDBe pairs were extracted with 90-100%, 50-60% and 20-30% ChEBI similarity, respectively.

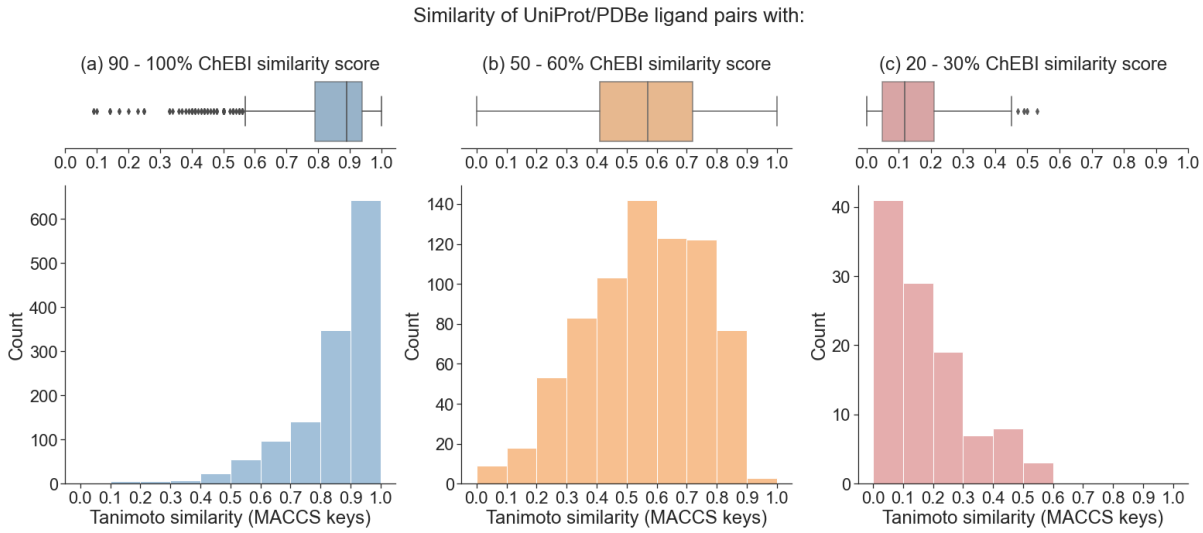


Figure 11: Boxplots and histograms of the Tanimoto scores calculated with MACCS keys substructural fingerprints for UniProt/PDB(e) ligand pairs with (a) 90-100% (in blue), (b) 50-60% (in orange) and (c) 20-30% (in red) ChEBI Tanimoto similarity. The histogram plots show the distribution of the Tanimoto scores calculated with MACCS keys fingerprints for each ChEBI similarity threshold. The whiskers of the boxplots indicate the minimum and maximum Tanimoto score (MACCS keys fingerprints) and the box shows the 25th, median and 75th percentile. The median Tanimoto score (MACCS keys fingerprints) was 89%($\pm$ 14.8 stdev), 57% ($\pm$ 19.5 stdev) , 12% ($\pm$ 14.2 stdev) for 100%, 50-60% and 20-30% ChEBI Tanimoto similarity, respectively

The results indicate that even though the same similarity measure, Tanimoto similarity, is used by the two algorithms, the choice of molecular fingerprints can critically impact the performance of each similarity search. Some key cases identified where our algorithm performs differently from the ChEBI similarity search are illustrated in Figure 12.

Further analysis, indicated that our pipeline's similarity measure will distinguish between different oxidation states of the same compound, whereas this is often overseen by the ChEBI similarity search. This is illustrated in Figure 12 (a) where ATP and ionized ATP (ATP^{4-}) has a ChEBI similarity score of 100%, while, the chemical similarity calculated by our pipeline is 94%. Additionally, our analysis suggests that smaller molecules will receive lower Tanimoto scores by our pipeline. An example is illustrated in Figure 12 (b), where our pipeline's similarity score between *D*-alanine and *D*-alanine zwitterion is only 69%.

Another interesting case is that of *L*-glutamine zwitterion and N^5-methyl-*L*-glutamine zwitterion which have a ChEBI Tanimoto coefficient of 100%, while our pipeline's similarly score is 83%, as shown in Figure 12 (c). The two compounds are predicted as analogues by the Zinc database (Sterling & Irwin 2015). In this instance, our similarity algorithm would be more efficient in distinguishing that N^5-methyl-*L*-glutamine zwitterion is not the actual biological ligand but a possible analogue of *L*-glutamine zwitterion. Finally, Figure 12 (d) shows an example of a possible outlier, where 2,3-dihydroxybenzoyl 5'-adenylate and 5'-xanthylate^{2-} received a high Tanimoto score, 87%, by our pipeline. The relatively

high similarity score is likely due to similar functional groups shared by the two compounds.

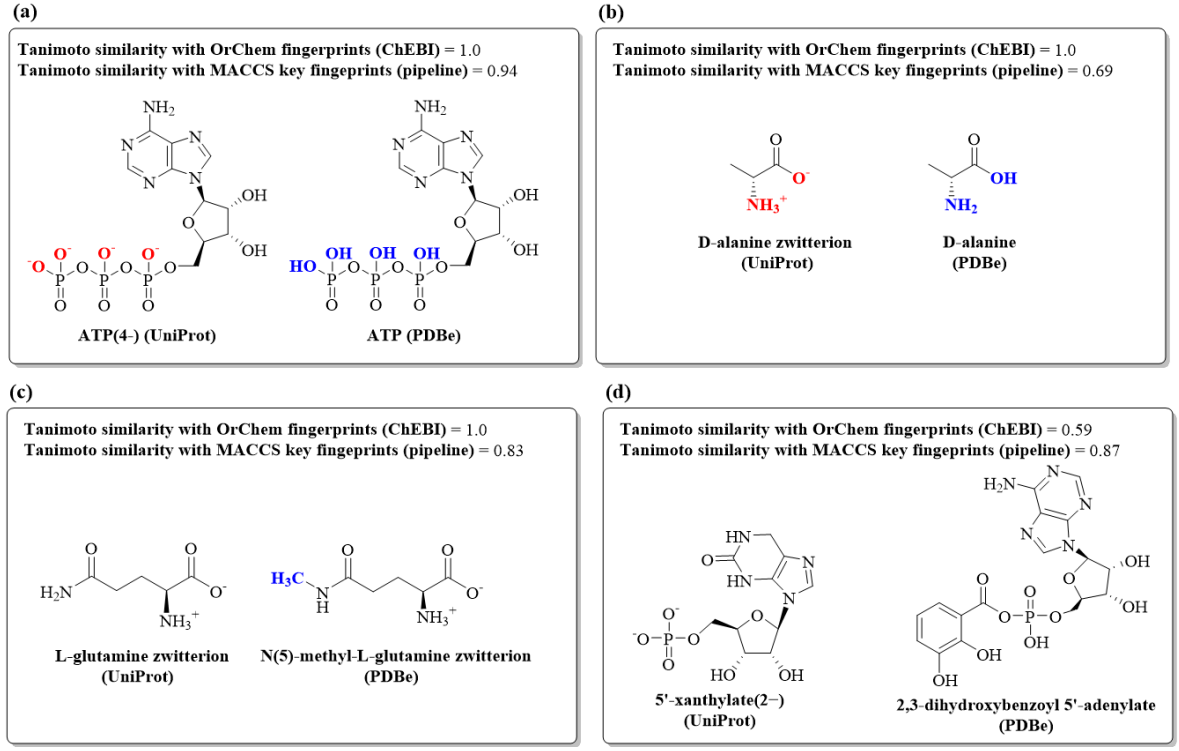


Figure 12: Key cases where our pipeline's similarity search (Tanimoto similarity with MACCS keys substructural fingerprints) performs differently from ChEBI's (Tanimoto similarity with OrChem fingerprints). Tanimoto similarity between (a) ATP (CHEBI:15422) and ATP^{4-} (CHEBI:30616), (b) *D*-alanine (CHEBI:15570) and *D*-alanine zwitterion (CHEBI:57416), (c) *L*-glutamine zwitterion (CHEBI:58359) and N^5-methyl-*L*-glutamine zwitterion (CHEBI:17592), and (d) 5'-xanthylate^{2-} (CHEBI:57464) and 2,3-dihydroxybenzoyl 5'-adenylate (CHEBI:15572), calculated by ChEBI's and our pipeline's similarity search.

2.2.5 Modelling large ligands

Written by CH, work by CH

As a demonstration of our pipeline's ability to model complex ligands, even before the use of docking, we modeled an peroxisomal short-chain specific Acyl-Coenzyme A (CAA) oxidase (Q96329) for which there is an experimental structure with ligands bound (2IX5 (Mackenzie et al. 2006)) to use as ground truth (Figure 13). CAA oxidases are enzymes that catalyse the breakdown of fatty acids in β-oxidation. Fatty acids are presented to the enzyme in the form of acyl-CoA thioesters and this family of enzymes are specific to fatty acids of 4-6 carbons in length. Flavin Adenine Dinucleotide (FAD) is required as a cofactor. In the first half-reaction, the cofactor is reduced to FADH- whilst the substrate is oxidised. Finally, FADH- is reoxidised by molecular oxygen in the final step (Mackenzie et al. 2006).

Our method identified 1JQI (Battaile et al. 2002) as being suitable template for modelling (only 29.9% sequence identity to query) and a structure was predicted using Phyre2. Our predicted structure

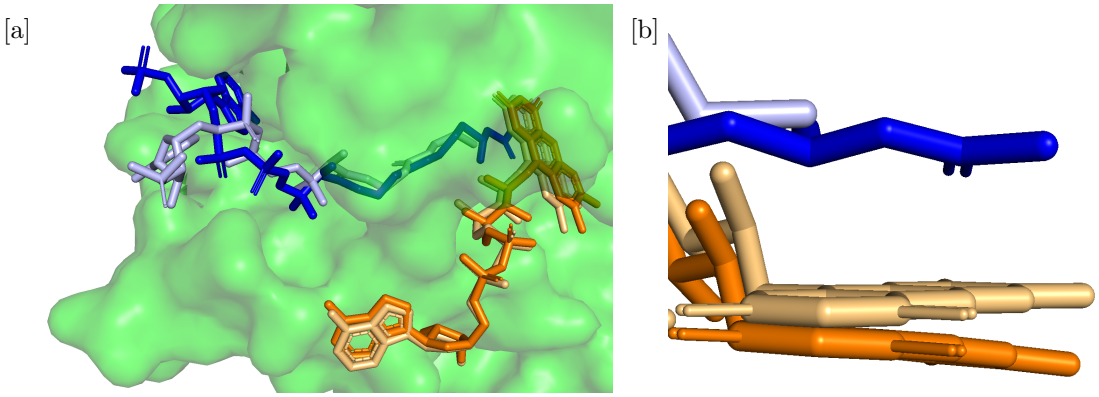


Figure 13: Ligands CAA and FAD in predicted model of Acyl-coenzyme A oxidase 4 (no refinement used). Green: predicted structure. Dark blue: predicted ligand pose of CAA. Light blue: experimental ground truth of CAA pose (2IX5). Orange: predicted pose of FAD. Light orange: experimental ground truth of FAD pose (2IX5). (a) Ligands CAA and FAD modelled into predicted structure (green). (b) Proximity in binding pocket demonstrates ability to model β-oxidation mechanism.

was superimposed onto the template structure and then the ligand coordinates were transferred to our model. The real experimental structure was also superimposed onto the template to obtain a ground truth for the ligand poses (Figure 13a). The pose of FAD was predicted very well against the ground truth. The predicted and experimental CAA ligands are anchored to the enzyme in the same position (relative to their pantothenate groups) but penetrate the active site in differing amounts due to their varying fatty acid R groups (Figure 13b). The 3-phosphoadenosine groups in the CAA ligands take differing conformations. Regardless, a clear oxidation mechanism can be identified by the proximity of the modeled ligands in a binding pocket within the predicted structure.

2.2.6 Modelling template with analogue

Written by SK, work by SK

To evaluate our pipeline's ability to identify templates with analogues bound, we modelled the enzyme tartrate dehydrogenase from *Pseudomonas putida* (*P. putid*), with the UniProt ID of Q51945, for which the crystal structure has been solved to 2Åcomplexed with an intermediate analogue (Malik & Viola 2010).

Tartrate dehydrogenase has the ability to catalyse three distinct NAD- dependent reactions(Tipton & Peisach 1990). Specifically, tartrate dehydrogenase can catalyse (i) the oxidation of (+)-tartrate to oxaloglycolate, (ii) the decarboxylation of *meso*-tartrate to *D*-glycerate and (iii) the oxidative decarboxylation of D-malate to CO^2 and pyruvate (Tipton & Peisach 1990). The enzyme's ability to catalyse different reactions is due to the substrates undergoing the same initial catalytic steps, but the intermediates dissociating from the enzyme at different stages of the catalytic cycle, yielding different final products (Malik & Viola 2010). Tartrate dehydrogenase is a homodimer, with the active site being

constructed of residues from both subunits (Malik & Viola 2010). Moreover, the enzymatic activity of tartrate dehydrogenase requires divalent and monovalent cations such as Mn^{2+} and K^{+} (Tipton & Peisach 1990).

The solved protein structure of *P.putida* tartrate dehydrogenase (PDB ID: 3FLK) contains the co-factor NADH, a metal ion and the intermediate analogue oxalate, which resembles the intermediates formed during the catalytic reaction (Malik & Viola 2010).

Even though the solved protein structure contained an analogue of the intermediates rather than the substrates of the catalytic reaction, our pipeline was still able to capture some similarity between oxalate and one of the final products, oxaloglycolate. Specifically, oxalate and oxaloglycolate had a Tanimoto similarity (calculated with MACCS Keys substructural fingerprints) of 77% and the similarity map between the two compounds is shown in Figure 14 .

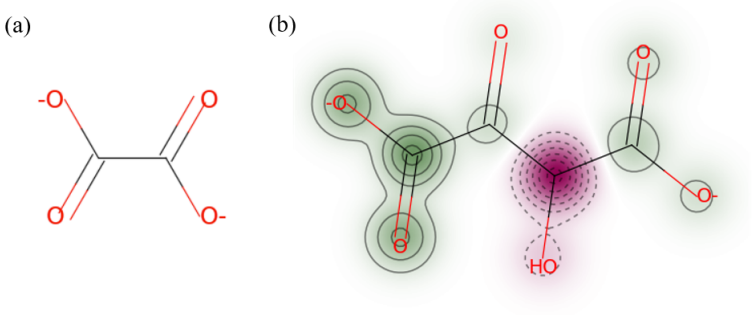


Figure 14: (a) chemical structure of oxalate (b) chemical structure of oxaloglycolate showing in green areas of high similarity to oxalate chemical structure and in red colour dissimilar areas, generated in the Python RDKit environment

Therefore, the 3FLK protein structure was returned by our pipeline within the top hits, with 99.5% sequence identity to the query and an E-value of 0.0. The coordinates from oxalate bound to 3FLK were extracted and inherited into the Phyre2 predicted model. Oxalate was replaced through superposition with oxaloglycolate and *D*-malate, followed by refinement using AutoDock Vina (*Work done by OW*), as shown in Figure 15. The binding affinities were -4.7 and -5.2 (kcal/mol) for oxaloglycolate and *D*-malate, respectively. As D-malate is a reactant for one of the reactions catalysed by tartrate dehydrogenase is was expected to have a higher affinity for the target than oxaloglycolate which is a product (Malik & Viola 2010).

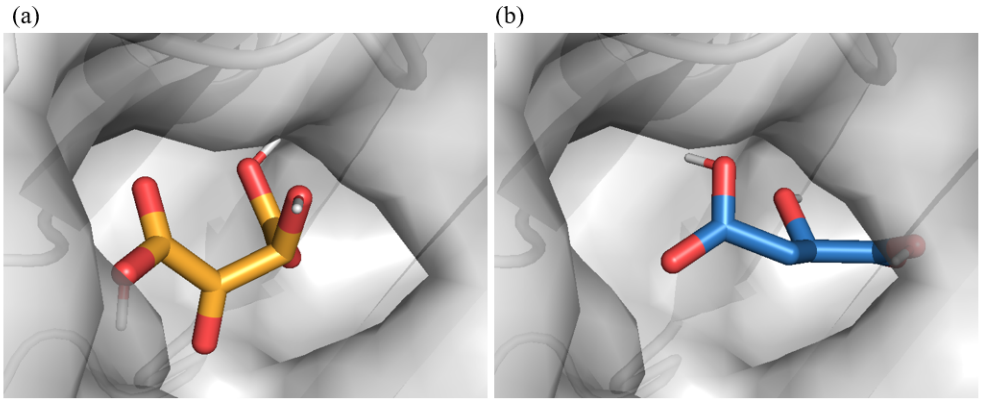


Figure 15: Figure by SK, (a) oxaloglycolate and (b) *D*-malate interactions with binding site of Phyre2 predicted model, refined using AutoDock Vina (*Work done by OW*)

2.2.7 Ligand transfer analysis

Written by CH, work by CH

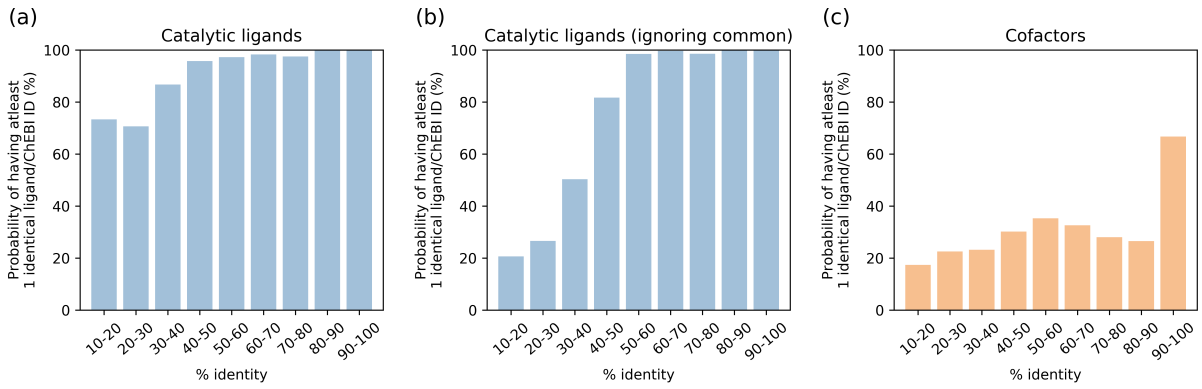


Figure 16: Ligand transfer by sequence homology analysis. Here a "positive transfer" is defined as the query and hit sequence sharing *at least one* ChEBI identifier. Transfer analysis of (a) all catalytic ligands, (b) all catalytic ligands minus those deemed to be highly common (see Appendix Table 3) and (c) all cofactors.

The results of the ligand transfer by sequence homology is shown in Figure 16. Our results show that the catalytic ligands associated with a protein sequence can reliably be transferred with nearly 100% success even at 50% identity (Figure 16a). This is still the case even when the analysis is altered to ignore ligands expected to be present in two sequences of low homology (Figure 16b) and thus gives an estimate in our ability to predict the biological substrate of interest. The same analysis repeated on cofactors did not yield such definitive results (Figure 16c), with only 66.7% of cofactors being successfully transferred at 90-100% sequence identity and only 26.6% at 80-90% identity.

2.2.8 Efficiency of database cross-referencing

Written by SK, work by CH and SK

The established in-house referencing between UniProt to ChEBI can retrieve compound information from 8.89 x 10^{-5} to a maximum of 5.41 seconds. The total space usage of the original mapping database downloaded from the EBI was 11 GigaBits which has been decreased to 1.7 GigaBits and 1.2 GigaBits for the "catalytic activity" and "cofactor" section, respectively by conversion to dictionaries. Nevertheless, our in-house mapping database will need to be updated when new versions are released (the Python script for generating our mapping can be provided upon request).

Additionally, the local BLAST databases (PDBe and UniProt) also require regular updating (later is already automated). Reviving chemical compound names and SMILES strings is enabled by accessing the ChEBI API. Similarly, the PDBe API which fetches the ligands and the binding sites will not need to be updated.

3 Predicting the impact of missense variants on ligand binding via homology-based docking

3.1 Methods

Written by AS, Work done by AS, OW

A Missense variant is a single nucleotide substitution in a gene which alters the encoded amino acid. Some missense variants can have a wide range of effects on the protein while others might have very little or no effect on the protein. These variations depend on the differences in amino acids properties between normal and mutated residues. Both structural and functional changes are caused by missense variants including protein stability, enzyme activity and binding (Bhattacharya et al. 2017). Several methods have been developed using a variety of approaches to predict these impacts of missense variants on protein (Ittisoponpisan et al. 2019). However, none of the available techniques report the impact of missense variants on ligand binding.

This being the case, a Python pipeline was developed to predict the effects of missense variants on ligand binding via docking. The pipeline was designed to be executed on the Linux server to generate a batch script containing lines of commands to execute other python scripts and programs, details of which are described below. The overview of the pipeline is illustrated in Figure 17.

The main process of the pipeline is predicting protein-ligand interactions given a 3D coordinate file of a query protein and a template protein coordinate file from the PDB database. The interactions are predicted by using a homology-based docking approach. Coordinates of the target ligand bound inside the template structure are inherited into the query protein structure. The ligand conformation is then refined by AutoDock Vina. Protein-ligand interactions of docked ligand conformations are presented in a Microsoft Excel Spreadsheets file and a PyMOL session file.

In order to predict the effects of missense variants on ligand binding, the pipeline first predicts interactions between the query protein and target ligand. Secondly, a mutant structure is generated from the query protein according to the missense mutation inputted. Thirdly, interactions between the target ligand and the mutant protein are predicted. The two protein-ligand interaction outputs can be compared to identify the differences; this highlights the effect of the missense variant(s) on ligand binding.

3.1.1 Programs and Python modules

Written by AS

Programs used in the pipeline are as follows: Autodock Vina version 1.1.2 (Trott & Olson 2009) was used for ligand docking and refinement, python scripts from AutoDockTools MGLTools version 1.5.6 (Morris et al. 2009) were used for file preparation and results processing in Autodock Vina and PyMOL version 2.4.1 (Schrödinger, LLC 2020) was used for protein structure mutation and protein-

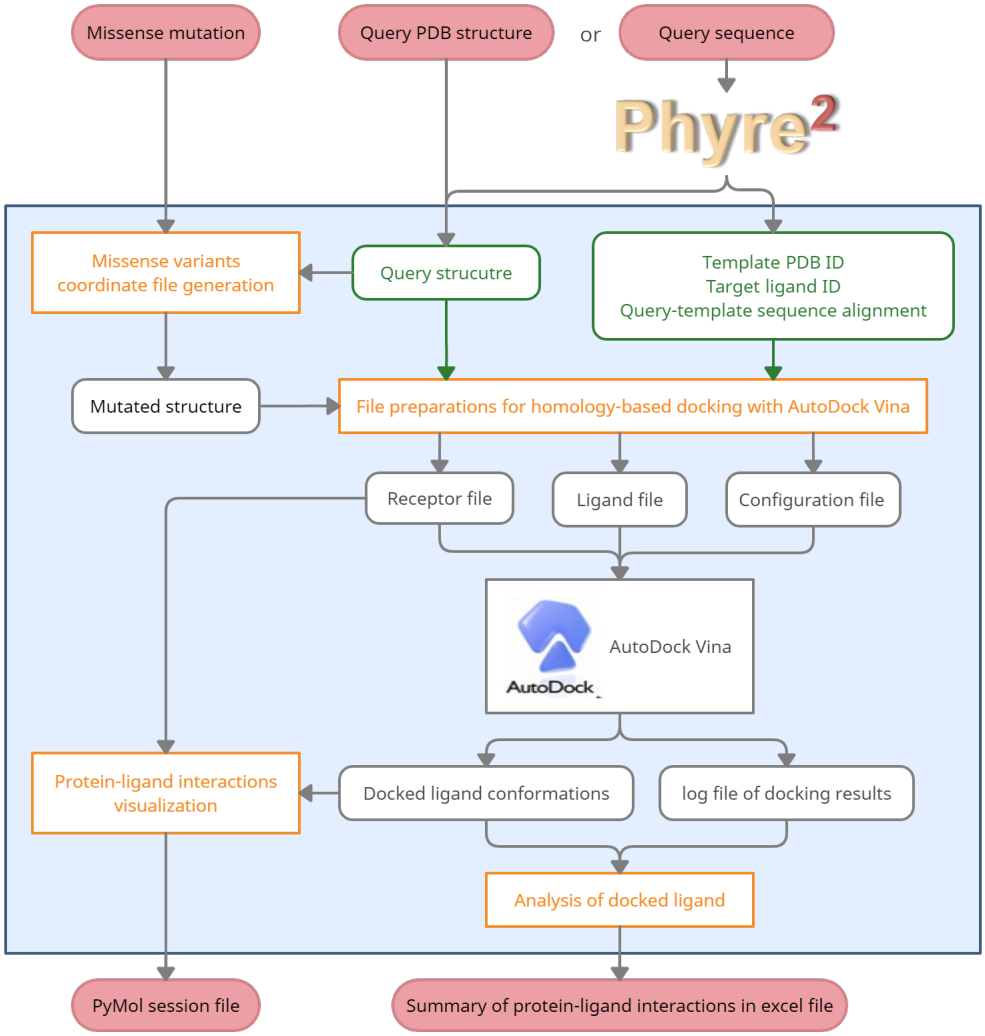


Figure 17: Flowchart of the procedures for predicting the impact of missense variants on ligand binding via docking, showing the pipeline developed as blue box. Red boxes represent the inputs and outputs of the pipeline. Green boxes are inputs of the pipeline that could be obtained from Phyre2, orange boxes are the developed Python scripts and black boxes are the intermediate files.

ligand interaction visualization. 3D coordinates files of query protein structures were modelled using Phyre2 (Kelley et al. 2015).

Scripts in the pipeline were developed and executed in Python version 3.8.5. Built-in Python modules used in the pipeline are argparse (used by all developed Python scripts to take command line arguments as inputs), datetime, os, sys and re. Other Python modules used and their version are as follow: Requests 2.24.0, Biopython 1.78 (Cock et al. 2009, Hamelryck & Manderick 2003), Pandas 1.1.3 (pandas development team 2020), 0.4.0, Numpy 1.19.2, Mendeleev 0.6.1 and 'Get raw distances' (Python module for PyMOL written by Takanori Nakane and Thomas Holder, downloaded from PyMOLWiki: https://pymolwiki.org/index.php/Get_raw_distances).

3.1.2 Pipeline inputs

Written by AS

The pipeline requires four mandatory parameters, namely, PDBID, LIGANDID, QUERYPDB and ALIGNMENT. QUERYPDB is the 3D structure file of the query protein in Protein Data Bank (.pdb) format which could be downloaded from the RCSB PDB database (Berman 2000) or could be a modelled structure predicted by a program such as Phyre2 (Kelley et al. 2015). LIGANDID is the id of the target ligand bound in the template protein structure which will be docked into the query protein. PDBID is the PDB id of the template protein structure with the desired ligand. The template protein can be selected from one of the templates provided by Phyre2 which is homologous to the query protein. ALIGNMENT is a FASTA file of the query and template amino acid sequence pairwise alignment which can also be retrieved from Phyre2.

There is one optional parameter for the pipeline called MUTATIONS. This parameter consists of two arguments, first is a keyword 'same' or 'diff' which refers to generating a mutant structure with all the inputted mutations or creating a mutant structure for each mutation. The second argument is mutations to be generated which should be denoted by the amino acid three letter code and their residue number. For example, 'ALA54PRO' indicates that alanine at position 54 in the query structure will be mutated to proline in the mutant structure. In the case where no mutations parameter is inputted, the pipeline could be used for homology-based ligand docking and prediction of protein-ligand interactions between the query protein and target ligand.

3.1.3 Missense PDB file generation

Written by OW, work by OW

A problem when docking a protein and ligand is that mutations in proteins can affect the efficacy of the binding. Missense variants cause changes to a single amino acid in a protein sequence. Often they have no effect on a protein at all, especially when the change is conservative, meaning the resulting amino acid has similar biochemical properties. There are cases however when a missense variant is non-conservative which can cause a change in the structure and function of the protein.

In the field of drug design, it is crucial to study the effects that missense variants may have on the ability of a small molecule to bind to a target. A single amino acid change may be all its takes to drastically change ligand binding capabilities or to result in off target effects.

We have developed a feature in our pipeline allowing users to specify mutations in the query protein sequence. They can then dock the ligand into the mutated protein to predict the effect that this mutation may have on the binding and conformation of the protein-ligand complex. This allows a direct comparison of binding of ligand to wild-type (WT) verses mutant proteins.

To achieve these mutations, we have developed a script that uses the PyMOL API. PyMOL is a molecular graphics software designed for the generation of high-quality molecular graphics images

(Schrödinger, LLC 2020). PyMOL can also edit PDB files with its mutagenesis feature and we use this to generate missense variants in our query protein. This is as simple as selecting what amino acid you want to change and the amino acid you want to change it to. There will be multiple conformations the new amino acid can take, each with a probability of existence. Our script takes the most likely conformation predicted by PyMOL; this is the conformation with the least steric clashes with surrounding residues. The script takes an input of a PDB file as well as mutations to be made and outputs a mutated PDB file. If the user specifies a residue that is not in the protein the script will throw an error. It is an optional feature that allows users to analyse the effect of single or multiple missense variants.

3.1.4 File preparations and homology-based ligand docking

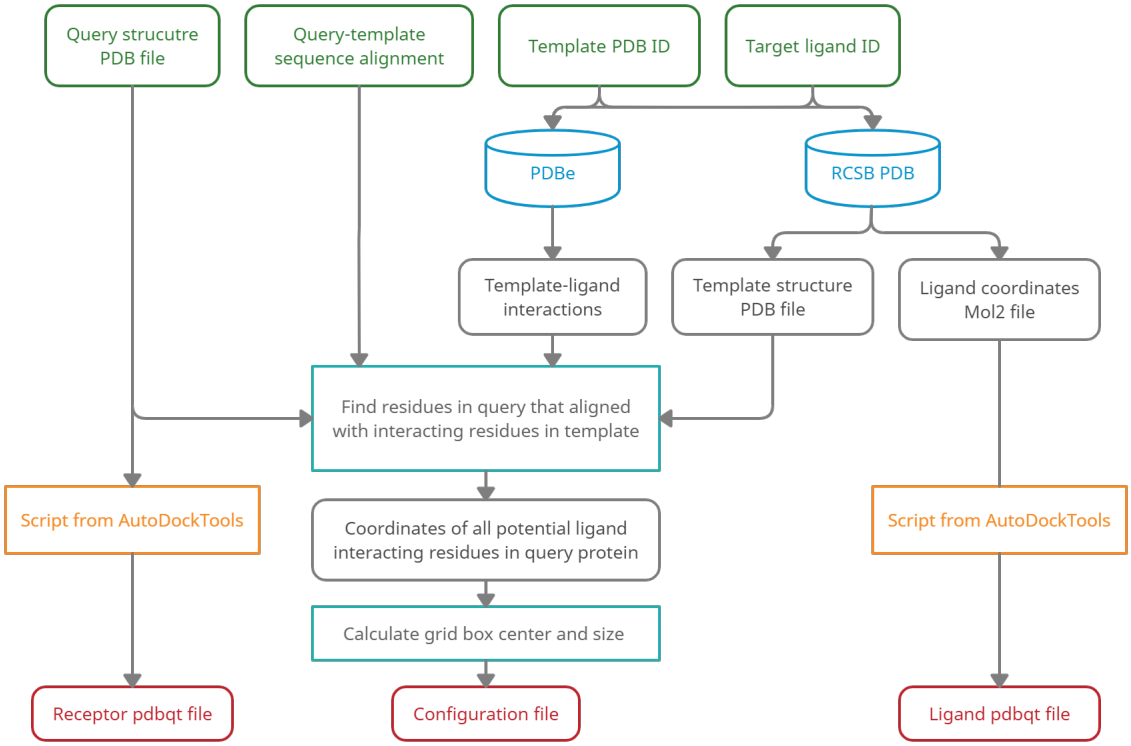
Written by AS

Protein-ligand binding is driven by the noncovalent interactions between ligand and nearby protein residues such as van der Waals interactions and hydrogen bonds. Ligand binding can be predicted by structure-based computational approaches using several protein-ligand docking methods ((Du et al. 2016)). In this project, an automated docking tool called AutoDock Vina ((Trott & Olson 2009)) is used to predict the best binding modes of the ligand in specific search area of protein and provides binding affinity of each ligand conformation.

Defining grid box centre and size are critical steps for ligand docking due to the fact that AutoDock Vina will only dock the ligand inside a specified search space. While it is possible to set the search space of AutoDock Vina to cover the whole protein structure, the docking process would require a longer execution time and would be computationally intensive. Therefore, the search space should not be too big but should cover the potential ligand-binding site. The potential ligand binding site can be inherited from the template protein structure by utilising the homology-based ligand docking principle based on the assumption that homologous proteins have similar ligand-binding sites.

This is why a python script called *pre_vina.py* was developed to automatically define the grid box centre and size from coordinates of query protein residues that aligned with ligand interacting residues in the template. This script is made up of two sections, the first section downloads all the necessary files and the second section creates a configuration file (*config.txt*) for AutoDock Vina. The workflow of file preparations for homology-based docking with AutoDock Vina including the *pre_vina* script is shown in Figure 18 and starts by passing all four required parameters of the pipeline into the *pre_vina.py* script.

The first section of the script downloads all the necessary files including template structure PDB file, ligand coordinates Mol2 file and template-ligand interactions file. First of all, the 3D coordinate file of the template protein structure is downloaded from the PDB database using retrieve_pdb_file function in PDB section of Biopython module (Hamelryck & Manderick 2003). Then PDBe REST API (Varadi et al. 2020) is used to obtain a list of bound ligands in the template PDB and their information (https://www.ebi.ac.uk/pdbe/graph-api/pdb/ligand_monomers/:pdbId). The existence of the target ligand in the list is checked, if the target ligand could not be found in the list then the script will be terminated and show a message that the target ligand not found in template PDB. In typical cases



where the target ligand is found, chain ID and author residue number of template protein that the ligand bound to are extracted. These two parameters are inserted into the url query of PDBe REST API to retrieve template-ligand interactions information in JSON format (https://www.ebi.ac.uk/pdbe/graph-api/pdb/bound_ligand_interactions/pdbId/:chain/:seqId). Significant interaction details are extracted and stored as a tab-separated values file. Each interaction record contains (i) protein residue name as three letter amino acid code, (ii) protein residue number, (iii) atom in the protein residue that interacts with ligand, (iv) atom in the ligand that interacts with the protein, and (v) type of interactions such as 'hbond' for hydrogen bond and 'vdw' for Van der Waals interaction. Subsequently, the instance coordinate file of the target ligand is downloaded from the RCSB PDB database using http/https protocols according to the 'Small molecule files' section of File Download Services in RCSB PDB (https://models.rcsb.org/v1/:pdbId/ligand?auth _seq_id=seqId&encoding=mol2).

The ligand structure file is downloaded as a Mol2 file instead of a regular PDB file due to the fact that it contains additional information on the types of bonds between ligand atoms. This is a beneficial feature for determining which bonds in the ligand are rotatable in the docking process. Even though both the PDBe and RCSB PDB are a member of Worldwide Protein Data Bank (Berman et al. 2007) that ensure consistency of PDB data, each database provides distinct advanced tools and services for effective data access, visualization and analysis. Hence, both databases were used in the pipeline, PDBe for downloading template-ligand interactions and RCSB PDB for downloading the Mol2 file of the ligand.

The second section of the script utilizes files inputted to the pipeline and files downloaded in the first section to create a configuration file that contains command line arguments for running AutoDock Vina. Only the required arguments are stated, including receptor file name, ligand file name and search

space (X, Y and Z coordinates of the centre and size in the X, Y and Z dimension in Angstroms). Defining the search space process begins with opening the PDB file of query and template protein by using PDBParser in Bio.PDB and parsing a FASTA file of query-template sequence alignment with SeqIO from Biopython. Then amino acids sequence of each protein was built from the PDB structure and indexed to the sequence in the alignment. This is necessary as the query and template alignment might not cover the whole protein and there are gaps in the alignment. Therefore, the residue number in the PDB structure of each residue in the alignment is recorded. By linking residue numbers of query protein with those of template protein, it is possible to predict the potential active site residues in query protein which are residues aligned to ligand interacting residues in the template. Coordinates of the CA (C-alpha) atoms in potential active site residues are then extracted from the query PDB file. From these coordinates, maximum and minimum values of each axis (X, Y and Z) are identified and set as the axis limits. The midpoint of all three axes is defined as the centre of the search space. The size of the search space is calculated from the difference between maximum and minimum values of each axis with an addition of 1 Angstrom to take into account the rotation of residues.

After *pre_vina* script created a configuration file, receptor and ligand files are converted into PDBQT format, required by AutoDock Vina. PDBQT file has a similar file format to PDB file with further information on partial charge and atom types. Two python scripts from Utilities24 directory of AutoDockTools, namely *prepare_ligand4.py* and *prepare_receptor4.py*, are used to create PDBQT files from the inputted 3D structure files of query protein (PDB file), with an optional parameter "-A hydrogens" to add hydrogens to the structure, and ligand (Mol2 file downloaded in the first section of *pre_vina.py*) respectively. The two python scripts are executed using pythonsh, a version of python 2 with all required modules from AutoDock. The two PDBQT files generated are then used by AutoDock Vina with parameters in the configuration file to dock the target ligand into the specific search space in query protein.

3.1.5 Analysis of docked ligand

Written by OW

Vina uses an empirical scoring function to measure the probability of a ligand docking with a certain conformation; it considers steric interactions, atomic distance, hydrogen bonds and hydrophobic interactions. A stochastic global optimization algorithm is used to minimise this scoring function within the search space and several of the most likely ligand conformations are found. Generally, if all top ligand conformations are clustered in the same space it suggests this is the true binding pocket; however, due to the stochastic nature of this search, the global minimum may have not been found.

The docking of a protein and ligand can be measured using binding affinity. This is a measure of the strength of the interaction in kcal/mol and relates to the energy required for the protein and ligand to dissociate. Vina ranks predicted ligand conformations by their binding affinity to the protein. The standard output from Vina is a single (.pdbqt) file of ligand conformations and a text file for binding affinity values. This output somewhat lacks interpretability, so we have added some post-vina processing

steps to our pipeline to generate more detailed and meaningful results.

The first processing step starts with the *process_VinaResult.py* script which is built into AutoDockTools. This script splits the ligand output file into separate files for each ligand conformation. From these separate files, our pipeline will analyse each conformation individually and store information in an excel file using a script called *analyse_vina_results.py*. Included in this excel file is a breakdown of the docking for each ligand conformation which includes the information listed below:

- Ligand Binding Affinity
- Cluster
- Residues in protein that interact with ligand through Van Der Waals forces
- Residues in protein that hydrogen bond with ligand

This output provides a simple, readable overview of the docking results and allows a comparison between top ligand conformations predicted by Vina.

3.1.6 Protein-ligand interactions visualization

Written by AS

Protein-ligand interactions of the AutoDock Vina results are visualized with Pymol. A python script in the pipeline called visualize.py was developed to create a Pymol session file (.pse) for showing protein-ligand interactions of each pair of protein and ligand. Only the residues in the query protein that are able to form a hydrogen bond with the ligand and residues within 3.6 Angstroms from ligand which represent Van der Waals interacting residues are shown. Hydrogen bonds between protein and ligand predicted by Pymol are also displayed as yellow dash lines. Each model of ligand conformation from AutoDock Vina is presented as one state of the ligand which can be accessed by using the arrow keys. An example of Pymol session file generated is shown in Figure 19 in the results section.

3.1.7 Pipeline benchmarking

Written by AS

Accuracy of the developed pipeline was assessed by using the pipeline to predict the effect of known missense variants on protein-ligand interactions of known proteins where the impact of missense variants have already been researched. Benchmarking of the pipeline was performed on a supplemental dataset of a research article on the impact of genetic variation on three dimensional structure and function of proteins Bhattacharya et al. (2017) which was downloaded from https://github.com/rcsb/SnpsInPdb. The dataset includes various information of missense variants as follow: PDB ID of the variant structure, single nucleotide polymorphism (SNP) ID of the mutation, PDB ID of the wild-type structure (only for a record which has wild-type structure available in PDB), mutated residue number in mutant structure, location of the mutation in the protein (surface or buried, located in loop, alpha helix or beta sheet),

frequency of the mutation, the impact of the mutation on disease risk and protein functional consequence of the SNP (enzyme activity, aggregates, stability, binding/dissociation, rearrangement).

The dataset was filtered to obtain a dataset that contains only records with all the information required to run our pipeline. Five filtration criteria applied to the dataset are described below, records that fit into at least one of the criteria were removed. Even though the original dataset contains a total of 374 records, the filtered dataset only consist of 26 records.

- (i) No wild-type PDB ID available: the wild-type PDB structure is a required detail because it will be used as a query protein structure of the pipeline.

- (ii) No amino acid mutation information of SNP could be found in the database: a python script was developed to append variant annotation retrieved from MyVariant.info API, a REST web service to query/retrieve variant annotation data, aggregated from many popular data resources, for instance, dbNSFP, ClinVar, CADD and UniProt (Xin et al. 2016) using get function in Requests python module to the dataset. For each record, SNP ID was queried in MyVariant.info API with 'dbnsfp' in fields parameter to limit the fields returned to only variant annotation from dbNSFP database, a database of all potential non-synonymous single-nucleotide variants (nsSNVs) in the human genome functional prediction and annotation (Liu et al. 2011). Amino acid alteration of the SNP was extracted from 'aa' nested field in 'dnsfp' field, which indicates reference and altered amino acids. Despite that, no mutated residue number was provided in the 'aa' field. For this reason, HGVS standard nomenclature of predicted SNP consequences on the protein level, for example 'p.A308S', from the 'hgvsp' nested field in 'dbnsfp' was also extracted to the dataset. It is possible for one SNP to correspond to several single amino acid changes while this amino acid alteration annotation might not be available in the database for some SNPs.

- (iii) Missense mutations of the SNP fetched from the database could not be identified in the wild-type and mutant PDB structures. Mutated residue stated in the dataset was manually examined in both wild-type and mutant PDB structures to verify that the amino acids mutation retrieved from database actually occurred. In cases where more than one mutations were found for one SNP, this manual curation was also performed to determine which mutation could change the wild-type to mutant protein.

- (iv) Wild-type and mutant PDB structures have no ligand in common. Since the aim of this benchmarking process is to assess the efficiency of the pipeline in predicting the impact of missense variants on interactions between protein and target ligand, it is essential for the wild-type and mutant protein structure to share at least one ligand which would be the target protein. A python script was written to obtain the list of shared ligands between wild-type and mutant protein. First, a list of ligands bound to each protein were retrieved from PDBe Graph API of PDBe REST API (Varadi et al. 2020) using get function in Requests python module to search for ligand monomers of each PDB id. The common ligands found in both list were then appended to the dataset as shared ligand.

- (v) Only ions such as MG, HG, ZN, CU, CA was the only shared ligand between wild-type and mutant protein.

After the filtration, each record in the dataset contains two additional information, including amino acid mutation of SNP (e.g. A>T) and shared ligands between wild-type and mutant PDB structure. From the 26 records in the filtered dataset, there were 13 unique wild-type proteins given that some records have the same wild-type proteins but different SNP ID. FASTA file containing amino acids sequences of the 13 wild-type proteins were inputted into batch processing mode of phyre2. The predicted structure of each wild-type was downloaded and used as a query protein structure in the pipeline. The template protein of each query was selected from several templates suggested by phyre2.

Written by OW, work by OW

We created a python script (*phyre_parser.py*) to retrieve information from either a single or batch Phyre results page using web scraping. By inputting the URL of the results page into this command-line script, a text file is returned that for each Phyre job contains (i) the job description (ii) the PDBID of each template (iii) the names of ligands that bind to each template. This information was used to find a template that binds to the ligand of interest; from which ligand coordinates were inherited for docking into the predicted structure.

Written by AS

As an alternative approach to manually executing the pipeline for every record in the dataset, all required parameters were combined into one CSV (Comma Separated Value) file. This file was then inputted into a python script call batch_docking that created a batch script (batch_docking_cmd.bat) with all the commands to execute the pipeline for each record. In addition, a Linux command to create a log file from terminal output was also added to the source command to execute the generated batch script (source batch_docking_cmd.bat— tee -a batch_docking_log.txt). This log file was used for discovering types and causes of errors to aid the debugging process. The starting and ending time of the execution were stated at the start and end of the log file, respectively, to show the total time taken which might be useful for running time approximation of next batch run.

Several analyses were perform on the pipeline benchmarking results, details are described in the result section. The quality of each result is determined by the root-mean-square deviation (RMSD) calculated from comparing the protein-ligand complex results with known structure from the dataset.

3.2 Results

3.2.1 Pipeline usage

Written by OW, Work done by AS, OW

To manually dock a single protein and ligand with Vina, one would have to convert files into the correct format, specify a search space using AutoDockTools and copy this information into a configuration file before running the program. For a competent user of Autodock this would take at least 10 minutes and for a first-time user considerably longer. Our docking pipeline successfully eliminates all manual pre-processing steps and so the run time is only limited by the docking itself. Typically, we find the average run time for a single docking is around 20 seconds with an extra 10 seconds added on for every mutation to be analysed.

As well as being faster, our pipeline gives more detailed and readable results than the standard output of Vina, saving users time on the analysis of docking results. An example PyMOL file produced by our pipeline is shown below (Figure 19).

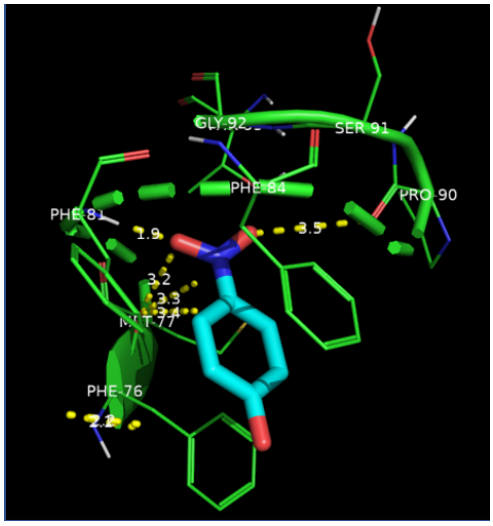


Figure 19: An example of a PyMOL session file produced by the pipeline. Here we see NPO (blue) docked to 1LS6 (green). Yellow dashed lines show hydrogen bonds. Protein residues and hydrogen bond lengths are labelled in white.

3.2.2 Benchmarking results

Written by OW, Work done by AS, OW

We performed three analyses on the benchmarking data set to assess the accuracy of our pipeline. The proteins used in benchmarking as well as the results for the analyses are shown in **Table 20**.

Table 2: Benchmarking dataset and pipeline results. Each row represents a protein for which a WT and missense structures are known.

Query PDB ID	**Template PDB ID**	**Ligand**	**Mutation**	**Mutant PDB ID**	**Binding/ Dissociation**	**Analysis 1 RMSD**	**Analysis 2 RMSD**	**Analysis 3 RMSD**	**Ligand binding affinity (WT protein)**	**Ligand binding affinity (Mutant protein)**
1LFG	1ce2	CO3	ALA10THR	1B0L	0	0.2084	0.347	0.013	-3.1	-3.1
1H0C	6pd1	GOL	GLY170ARG	1J04	0	14.454	12.977	0.007	-3	-3
2AOT	1jqe	SAH	THR105ILE	1JQE	0	1.065	1.09	0.111	-8.6	-8.7
1UCN	1zs6	ADP	SER120GLY	2HVE	0	8.117	8.364	0.027	-8.4	-8.4
2CE2	1x1r	GDP	LYS117ARG	2QUZ	1	5.24	8.423	8.239	-10.5	-7
3BWM	2cl5	SAM	VAL107MET	3BWY	0	0.483	0.461	0.033	-9.1	-9
2CVD	1iyi	GSH	VAL186ILE	3EE2	0	2.73	2.045	2.347	-1.9	-2
3K7G	3m1n	SO4	ASP85GLU	3K7J	0	17.258	17.385	0.703	42.6	43.3
2YD0	3se6	NAG	GLY301ASP	3MDJ	0	22.797	21.95	4.893	38	37.4
2YD0	3se6	NAG	LYS483ARG	3MDJ	0	22.797	21.899	4.923	38	37.3
2YD0	3se6	NAG	GLN685GLU	3MDJ	0	22.797	21.88	4.934	38	37.4
2YD0	3mdj	BES	GLY301ASP	3MDJ	0	7.441	7.97	1.707	-4.2	-5.4
2YD0	3mdj	BES	LYS483ARG	3MDJ	0	7.441	7.998	1.735	-4.2	-5.2
2YD0	3mdj	BES	GLN685GLU	3MDJ	0	7.441	3.494	7.71	-4.2	-3.5
3QE2	6j7a	FAD	VAL430GLU	3QFC	1	15.329	14.462	5.247	1.5	-0.6
3QE2	6j7a	FMN	VAL430GLU	3QFC	1	17.017	17.091	4.521	0	-2.7
3QE2	2bpo	NAP	VAL430GLU	3QFC	1	6.38	6.331	8.754	-8.5	-8.4
1LS6	3u3o	A3P	VAL243ILE	3QVU	0	0.526	0.541	0.034	-12.4	-12.4
1LS6	2zvp	NPO	VAL243ILE	3QVU	0	4.393	8.375	0.124	-6.5	-6.5
3TG4	3mek	SAM	GLY165GLU	3S7B	0	1.522	1.582	0.217	-8.6	-8.7
4JBS	3se6	NAG	LYS392ASN	3SE6	1	11.248	11.518	1.889	52.1	48

3.2.3 Analysis 1: Comparing predicted WT docking to crystalised WT docking

For the first analysis we compared the ligands docked into Phyre-predicted protein structures of the WT proteins with the actual known structures downloaded from the PDB to see if we could replicate the correct docking conformations.

We took the protein sequences of the WT proteins and predicted their structures with Phyre. For each Phyre job we chose a template that bound to the ligand of interest to inherit ligand coordinates from. We then ran our pipeline to dock the ligand and predicted protein structure. As Phyre was predicting the structure of proteins with known structures, it would provide the actual protein structure as a template for the predicted structure. We purposely avoided using the actual protein as a template for ligand coordinate inheritance because we wanted to see how well dockings could be predicted using templates with more remote homology. Also, we envisage under normal circumstances there will not be a template with a 100% sequence identity, otherwise the docking is redundant.

To assess our predicted docking, we downloaded the coordinate file of the known protein-ligand complex from the PDB. In PyMOL we aligned the protein from our predicted docking with the actual protein downloaded from the PDB. We calculated RMSD between the predicted and actual ligand conformations using the PyMOL function *rms_cur*. This function computes the RMSD between two atom selections without performing any fitting and only using matching atoms in both ligands for the fit. If ligands displayed symmetry, then we calculated RMSD based on atom type to ensure results were the same no matter the orientation of the ligand. Out of the 17 dockings performed, 11 ligands bound in the correct binding pocket, 5 of which had a RMSD of less than 2 when compared to the actual ligand conformation.

3.2.4 Analysis 2: Comparing predicted missense docking to crystalised missense docking

We next compared ligands docked into mutated predicted structures with known mutant structures of the protein-liagnd complexes downloaded from PDB. From the Phyre-predicted structures, our pipeline generated missense mutations in the proteins that corresponded to those in the benchmarking dataset. Using the same process as with Analysis 1, we aligned predicted mutant structures with known mutant structures downloaded from the PDB in PyMOL and calculated RMSD between the ligands.

Out of the 21 dockings performed, 13 bound in the correct binding pocket, 5 of which had a RMSD of less than 2 when compared to the actual ligand conformation.

3.2.5 Analysis 3: Comparing predicted WT docking to predicted missense docking

The third analysis we undertook was to compare ligand binding in predicted WT structures and predicted mutant structures. The benchmarking dataset contained information on whether the missense mutation alters the binding/dissociation of the ligand. We calculate RMSD between ligands in WT vs mutant structures and compared binding affinity values to see if our ligand binding/dissociation was changed as expected.

Out of the 21 dockings performed, 5 were expected to show a change in binding/dissociation when the missense mutation was applied. We found that in 4 out of these 5 cases, the ligand changes binding

pose so that the RMSD between ligand in the WT vs mutant structure is more than 2 and binding affinity changes by more than 1 kcal/mol.

Of the 16 remaining dockings in which the mutation is known not to change binding/dissociation of the ligand; 15 have a similar binding affinity (within 1 kcal/mol) between ligands in WT and mutant structures. This is shown in figure 20.

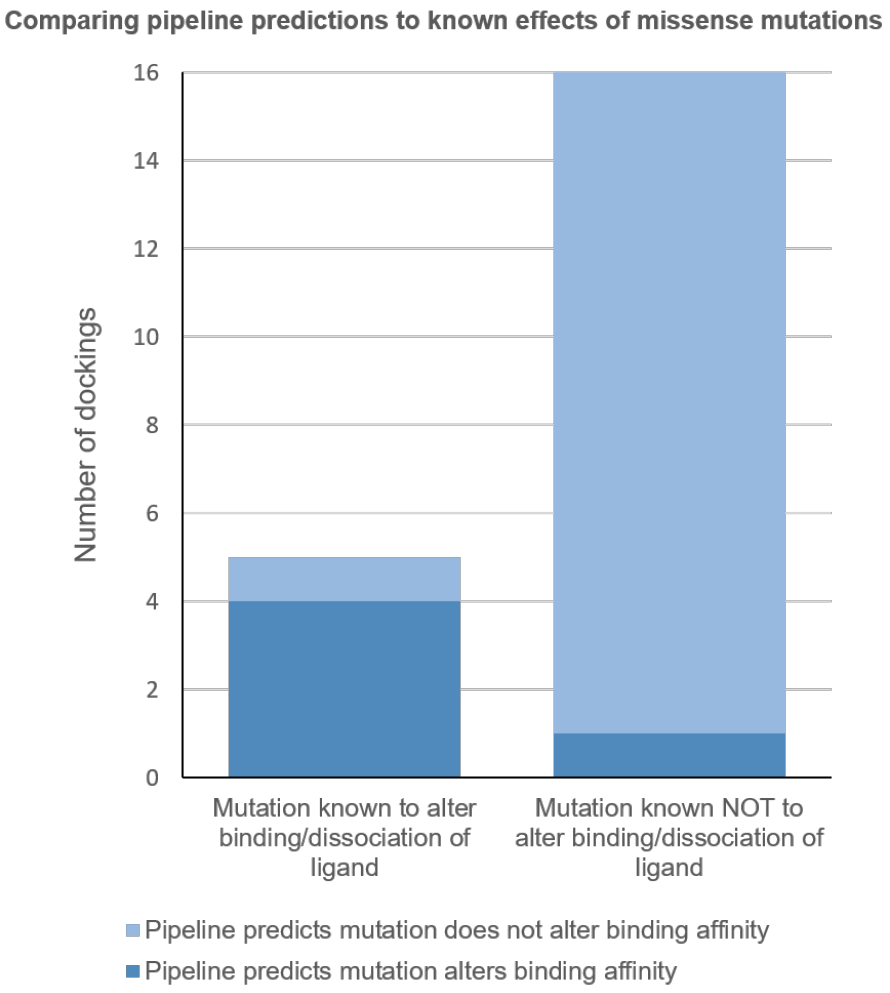


Figure 20: Analysis 3 results. Comparing the predicted effects of missense mutations to the known effects of missense mutations.

3.2.6 Case study: Testing pipeline on missense variants which are known to affect the ligand-binding affinity

Written by AS

Retinol is an alcoholic form of vitamin A which is stabilised by binding to intracellular retinol-binding proteins (CRBPs). According to the UniProt entry of retinol-binding protein 1 (P09455), there are three all-trans retinol-binding sites in this protein, at residue 41, 63 and 109. A group of researchers had study

the effects of LYS41LEU and GLN109LEU missense variants on the interaction between retinol and human CRBP types 1 (CRBP1) (Menozzi et al. 2017). X-ray crystal structure of wild type holo human CRBP1 (5LJB), and its LYS41LEU and GLN109LEU double mutant structure (5LJE) were deposited to the PDB database. Their research concluded that both LYS41LEU and GLN109LEU missense variants strongly decreased the binding affinity of retinol, particularly in double mutant structure (ref). Therefore our pipeline was used to predicted the impacts of both missense variants in wild type CRBP1 (5LJB) on retinol-binding and compared the mutated structure with double mutant structure (5LJE). The amino acid sequence of 5LJB was inputted into phyre2 to get the predicted structure which was the query protein structure of our pipeline. Top phyre2 template (1CRB), a crystal structure of CRBP1 in rat with bound retinol, with 96% identity was selected as the template for homology docking. The target missense mutation set for the pipeline was LYS41LEU and GLN109LEU in the same protein.

Retinol docked into a pocket inside the query protein structure and has a similar conformation to the actual ligand bound in the wild type structure (5LJB) with RMSD of 1.468 as shown in Figure 21 a, coloured as orange for the docked ligand conformation and red for the actual ligand conformation. Cartoon illustrations of the protein are coloured as yellow and red which represents the predicted structure of 5LJB and the original 5LJB, respectively. Protein-ligand interactions identified by the pipeline were 5 Van der Waals interacting residues and no hydrogen bonding residues. However, this conflict with the previous research results which stated that the hydroxyl end of retinol forms two hydrogen bonds with GLN109 and LYS40. These could be caused by two factors, first, different rotation of the hydroxyl group (labelled with a black circle in Figure 21 a) in two ligand conformations, and, second, slight alteration of protein side chain from protein modelling even though overall structure of the modelled structure is very similar to the original structure with RMSD of 1.768.

The top ligand conformation docked into the mutated structure generated with double mutations in the active site, LYS41LEU and GLN109LEU, is substantially different from the top ligand conformation in query structure (RMSD=9.603), the ligand is flipped horizontally as shown in Figure 21 c. This is a consequence of increasing protein internal pocket size (yellow surface for the query protein and blue surface for mutated protein) caused by the two mutations which offer more space for the ligand. In the query structure, the side chain of LYS-41 points toward the pocket which clash with the ligand aromatic ring, hence, limiting the ligand to place the aromatic ring in the opposite side of the pocket. However, the side chains of LEU-41 and LEU-109 in the mutated structure extend away from the pocket.

Despite the massive difference between the top ligand conformations in query and mutated proteins, other models of ligand in the mutated protein have similar conformation to the ligand in query protein, especially the second (green) and third (cyan) ligand models with RMSD of 1.980 and 0.053, respectively (Figure 21 b). Given that there is one ligand model in the mutated structure that is almost identical to the ligand in query protein and the lack of evidence showing the missing of hydrogen bonds in mutated structure as there is already no hydrogen in the query structure, it is improper to conclude that the two mutations affects the ligand-binding of the protein as in the case of the previous research. This inconsistency might be owing to the side chains deviation of mutated structure (blue) from the actual mutant structure (magenta), as displayed in Figure 21 d, caused by missense variants coordinate file

generation step.

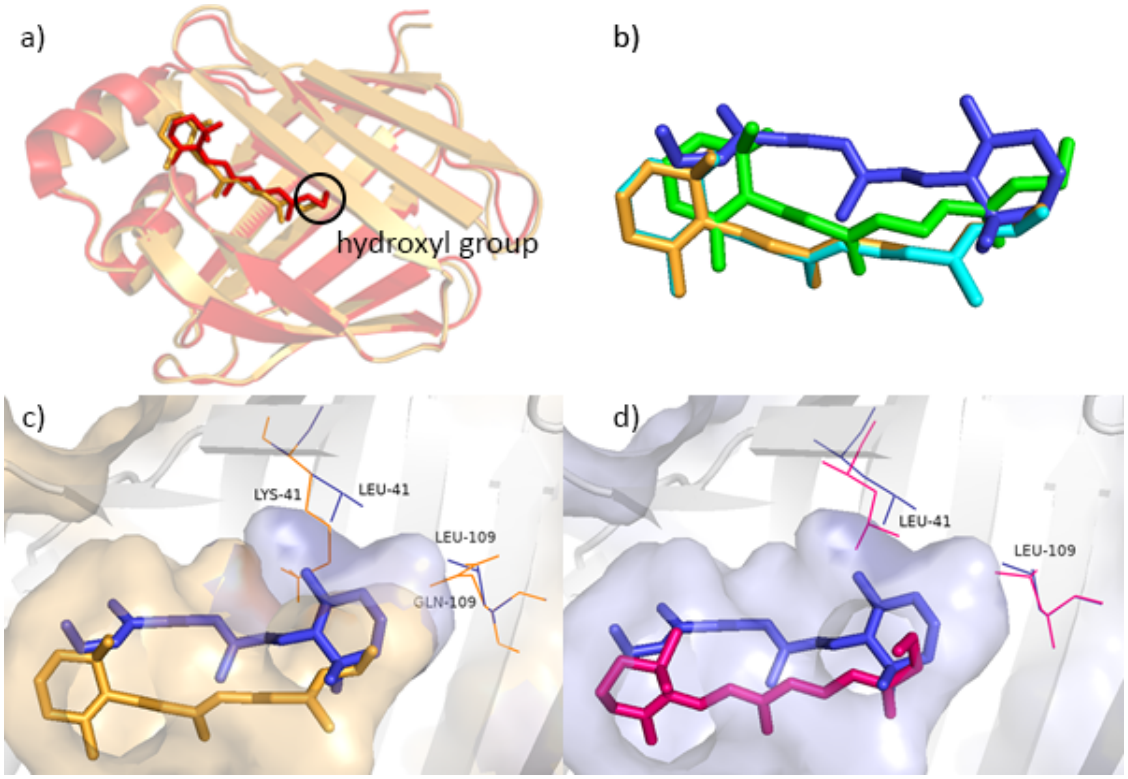


Figure 21: Comparison of pipeline results with original crystal structures of wild type and mutant CRBP1 known to affect retinol-binding affinity. a) Comparing positions and conformations of ligand docked into the query structure (orange) with ligand bound inside the original crystal structure 5LJB (red). Hydroxyl group of retinol is labelled with black circle. b) Top three ligand models in mutated structure (first, second, third coloured as blue, green and cyan, respectively) against the top ligand model in query structure (orange). c) Ligand docked into the pocket of query protein (orange) and mutated protein structure (blue). d) Ligand conformations in mutated structure (blue) and in mutant crystal structure 5LJE (magenta). Images were exported from Pymol showing ligand as sticks and protein pocket as a transparent surface of protein. The original and mutated residue are shown as lines coloured according to their protein colours and labelled with residue names.

4 Discussion

4.1 LigandTemplateFinder

Written by CH

We have demonstrated that our pipeline, LigandTemplateFinder, has broad coverage in its ability to find starting coordinates for ligands from a broad variety of targets with suitable templates identified for 71.1% of queries (Figure 7). This can be even further extended if the user knows the true ligands associated with the their query proteins from their own experimental analysis. Our method can return multiple suggested ligand/binding sites and is not limited to a single output like some methods can only return the largest cluster of continuous ligands (Roche et al. 2013).

Given our relatively high coverage, the use of a conservative binding site conservation threshold does not seem to have significantly impacted the performance like other residue conversation strategies have in other approaches (Wass et al. 2010). However, in time, we would like to perform a detailed analysis of how varying the conservation threshold impacts coverage and other measures of conservation, such as a fractional threshold for large binding sites or a "residue voting" procedure as used in Roche et al. (2013).

We propose that the two broad peaks in Tanimoto similarity seen in Figure 7b approximately correspond to templates that have the real ligand bound and those that have analogues bound. However, this would need to be verified with very careful analysis to distinguish between real analogues and native ligands in different protonation states. If the two distributions were found to be different, we could establish a similarity threshold to automatically distinguish true ligands and analogues bound in structures and use this as a ranking algorithm.

We have shown that the use of our UniProt to ChEBI mappings can be extended to cover a significant portion of all protein sequences by using ligand transfer by homology. The presence of the same ligands being associated with two sequences of very low homology is consistent with similar analysis done on 4 and 3 digit Enzyme Commission (EC) numbers (Devos & Valencia 2000). It is worth noting that while this approach may return ligands as false positives, the ligand will only be suggested if there is a valid structural template with real/similar ligands bound into conserved binding sites. However, we would recommend experiments to determine whether this approach is more reliable than taking multiple ligands from structurally homologous structures to form large clusters as a way of determining which ligands should be transferred to the query (Wass et al. 2010, Roche et al. 2013).

4.1.1 Enhancing ligand modelling

Written by CH and SK

Written by CH

In it current implementation, our approach does not account for ligands that bind to residues across

multiple domains/subunits, as long as 3 residues somewhere in the binding site the ligands are returned. Comprehensively modelling ligands using a template approach across multiple domains/chains is a considerably harder problem (analogous to how most current protein structure prediction programs only model single domains (Kelley et al. 2015, Biasini et al. 2014)).

An example of this is seen in Figure 13a, where the inherited coordinates for the 3-phosphoadenosine group in the CAA ligand appear to not be making contacts with the protein surface. This is due to there being an extra motif in the template that was not present in our query. Whilst the pose in this example probably could be refined to fit the ground truth with docking (not performed in this instance to demonstrate ability excluding docking) it highlights a limitation in our approach. we would suggest adding two extra features to our program, each increasingly conservative; (i) ignore any ligands which coordinate with residues from multiple chains and (ii) do not transfer ligands if the coordinating residues corresponding form the template cannot be modelled.

Written by SK

Future work would involve establishing a ranking system of the PDB templates returned from the BLASTP search with the PDBe database. PDB templates would be ranked initially by their E-value and then structures with ligand information would be given priority for one-to-one threading by the Phyre2 modelling resource. Additional factors to consider are (i) similarity of PDB ligand to UniProt chemical compound, (ii) whether similarity is with a product or reactant of UniProt's catalytic reaction, and (iii) cofactor information in templates.

Moreover, certain enzymes require the presence of monovalent and/or divalent metal ions for their enzymatic activity. For instance, in the case of tartrate dehydrogenase (please refer to Modelling template with analogue in the results section), the construction of the enzyme's active site requires the presence of a divalent ion and cofactor. Future work could explore extracting cofactor and metal ion coordinates from the RCSB PDB database and how this would impact ligand binding.

In the case where an analogue, rather than the true biological ligand, is bound to the template an automated procedure would need to be established for extracting the analogue coordinates and replacing it with the biological ligand. A possible approach could include superimposing the chemical structure of the biological ligand to the analogue complex in the protein, followed by refinement using AutoDock.

4.1.2 Problems with binding site identification

Written by CH

A major limitation of our approach is the reliance on the BLAST alignment as a means to identify the conservation of the binding sites. Due to their highly divergent nature, it is very difficult to construct an accurate alignment from two weakly homologous sequences. Therefore, our cross-referencing approach means that if two sequences are misaligned by even one residue (as is common in regions of low conservation), we will potentially falsely label a binding site as not conserved. An example of this can be

seen in the yellow residues in 8 which where incorrectly predicted as being unconserved due to an error in the alignment (typical considering the query and template only share 26.7% identify. This is why we have left the binding site conservation threshold as an optional parameter for the user (setting this to 0 will bypass the feature entirely). Additionally, in its current implementation, our method is unable to handle structures that were solved with mutated or non-standard amino acids inserted in the sequence to aid crystallisation (Hassell et al. 2007). This could be rectified by taking the information about the mutated residues in the structure (provided by the PDBe REST API (Varadi et al. 2020)) and restoring the WT residues in the template sequence before we do our analysis. However, given the large coverage in our testing, we doubt this has greatly affected coverage.

More broadly, while the identification of homologous templates was not the main aim of this research, the use of a simple BLASTP search is a crude approach. We propose an alternative strategy similar to that used in Phyre2 to construct the alignment (Kelley et al. 2015). Phyre2 uses Hidden Markvol Model (HMM) to HMM matching (Söding 2005) and takes into account both the know secondary structure of the template and the predicted secondary structure of the query (as predicted by PSIPRED (Jones 1999)). The use of this approach would allow much greater sensitivity (ability to detect true homologs) when compared to BLAST or PSI-BLAST (Kelley et al. 2015).

However, the speed of BLAST, as well as our caching procedure, means we can explore and suggest templates with ligands for a large portion of the UniProt database (22,798,807 unique mappings) relatively quickly. For future work, we propose the creation of a publicly available resource where we have precomputed the suggested template for all the UniProt sequences for which we have mappings. This would allow users trying to increase the structural coverage of proteomes using homology modelling to use templates that enable better modelling of ligands into pockets.

Another limitation of our approach is the reliance on a single template to provide all the ligand pose information. This means that if the ligand we are inheriting is docked into the binding site in an unusual way this will heavily bias our results. This is why other methods, such as 3DLigandSite and FunFold2, use clustering approaches to derived a consensus for the ligand pose/binding residues (Wass et al. 2010, Roche et al. 2013). In the future, we would recommend having the starting coordinates for the ligand pose before docking being derived from largest cluster of ligand from homologous structures.

The use of refinement once of the ligand coordinates have been superimposed is not always necessary. For computational reasons, we would suggest using a program and can detect a Van der Waals clash between the inherited ligands and the protein (implemented using Biopython (Cock et al. 2009) what would then suggest/automatically run refinement.

4.1.3 Improvements in PDB ligand mapping

Written by CH

Our method struggles with a lack of mapping between the ligands in the PDBe and the ChEBI database. More specifically we rely on the chemical component identifier from the PDB file which sometimes is not associated with a ChEBI identifier (often the case when an obscure analogue is used).

The lack of coverage here means that our method is loosing potentially useful structural information for modeling ligands. In future, we would recommend generating the molecular fingerprints straight from the structure file (MOL2) of the ligands found in the templates (we have already automated the retrieval of the MOL2 files from the PDB website).

Another limitation with our approach of extracting the chemical component identifier from the PDB file is that some authors will give the same identifier to different derivatives/analogous. A good example of this is the CAA ligands found in the structures of 2IX5 and 1JQI seen in Figure 13. Even though these two ligands are CAA derivatives with different length fatty acid chains, both authors have named the ligands as "CAA" when they deposited the structure (Battaile et al. 2002,?). The extent to which this is a problem could be quantified by performing a similarity calculation between the fingerprints of ligands as obtained via our method (via PDBe and ChEBI) versus our proposed method of the MOL2 files. Any results returning less than 100% similarity would indicate

4.1.4 Improvements in UniProt ligand mapping

Written by SK

Our method relies on the molecules listed under the "catalytic activity" section of the UniProt database to identify whether the PDB template has the biological ligand bound. As this section provides all the participating molecules in the biochemical reaction catalysed by the enzyme, it will also include molecules such as ATP, NAD, FAD that are often cofactors rather than the biological ligand. Currently, such cofactor-like molecules will be returned by our pipeline as "catalytic activity/cofactors" and the user will have to manually explore whether the molecule is a cofactor or the true biological ligand. For a given structure, the PDBe database will indicate whether the chemical component is a cofactor or ligand. Future work could involve cross-referencing between UniProt and PDBe to confirm whether a molecule under the "catalytic activity/cofactors" section has been matched with a PDB ligand or cofactor.

Additionally, the "catalytic activity" section provides the substrates of the enzyme-catalysed reactions, without indicating these are reactants and products. In the case where catalysis is not bidirectional, this could impact whether the true biological ligand has been identified. For a limited number of entries, UniProt does provide the direction of the reaction, however, this information is not currently employed by our algorithm. Therefore, the user has to manually explore whether the similarity captured between a chemical component of the PDB template is with a reactant or product of the reaction. Future work could involve investigating whether the direction of the chemical reaction can be extracted either from UniProt (or Rhea) database, using the SIFTS mapping between the PDB structures and EC numbers (Velankar et al. 2012, Dana et al. 2019).

It could also be worth exploring the more advanced similarity scoring system of the PARITY algorithm for identifying PDB templates with biological relevant ligands (Tyzack et al. 2018). PARITY compares the chemical components within the PDB templates to both on the left-hand side and right-hand side of the catalytic reaction separately. The best match is returned and whether the match was with the reactants or products. On the downside, PARITY is more computationally expensive than our current

approach as it employs a maximum common substructure method for measuring similarity. As our program is often required to compute similarity between hundreds of compounds for a single run, we would need to evaluate the trade-off between computational cost and an increase in similarity performance for implementing such an approach.

4.1.5 Limitations in chemical similarity searches

Written by SK

The comparison of our similarity algorithm to ChEBI's highlighted several limitations of our current implementation. In particular, it elucidated that the Tanimoto coefficient does consider the size of the molecules. This issue arises from the fact that smaller molecules will naturally fewer have "1" (positive) bits. As the Tanimoto coefficient accounts for the number of shared positive bits, smaller molecules tend to have lower similarity scores (Leach & Gillet 2007). Additionally, in contrast to the ChEBI similarity search, our algorithm will distinguish between a molecular and ionic compound (eg ATP and ATP^{4-}). On the one hand, this is a positive feature as the oxidation state of a compound may impact its interactions with the binding site residues. On the other hand, this increases the complexity of distinguishing identifying the true biological ligand and an analogue. Additionally, this could increase the number of false negatives, especially with regards to smaller molecules. For instance, the Tanimoto score between *D*-alanine and *D*-alanine zwitterion is just 69% and under our default similarity threshold (70%) this match would have been missed. Further work would include a large-scale analysis of the effect of the molecular weight on the Tanimoto coefficient, to identify whether a lower similarity threshold is needed for smaller molecules.

The MACCS keys substructural-based fingerprints, currently employed by our algorithm, have a relatively short length of 166-bits, enabling efficient computational calculations. As MACCS keys are calculated based on the presence or absence of pre-defined functional groups, they cover information mostly related to atom bonds and types and include limited connectivity information (Xie et al. 2020). Hashed fingerprints such as RDKit fingerprint and ECFPs, include connectivity information and can also capture rarer molecular fragments. In future, it would be worth exploring whether the combination of MACCS keys with hashed fingerprints would improve the similarity search performance as well as the effect of this on the computational efficiency. Future work would also involve pre-computing molecular fingerprints representations for all compounds in the in-house mapping databases to further increase the computational efficiency of our pipeline.

A limitation of calculating chemical similarity with molecular fingerprints is that there is no established threshold equivalent to that BLAST searched where an E-value of below 0.001 is considered significant. Tanimoto similarity depends on fingerprint type and therefore the is no universal threshold above which match is significant (Cereto-Massagué et al. 2015). Future work would include obtaining a database with known or predicted analogues (such as the ZINC database) and biological ligands to attempt providing the user with a threshold above which a hit is most likely an analogue or the true biological ligand.

Overall, in our current approach chemical similarity is used as a filter and docking is essential for ruling out false positives.

4.2 Docking and missense variant prediction

4.2.1 Limitations and future work

Written by OW

Some of the predictions made from our pipeline on the benchmarking dataset are incorrect. A number of these inaccuracies are due to limitations of Vina that make it only suitable for docking under certain conditions.

Firstly, Vina takes the protein as a rigid body. In nature proteins are able to move and it has been shown that conformational changes in proteins are often observed when a going from an unbound to a ligand-bound state (Bennett et al. 1984). Another shortcoming of Vina is that water molecules are removed for the docking process. In nature water molecules can sometimes mediate protein-ligand binding and stabilize the complex through hydrogen bonding (Ball 2008, Poornima & Dean 1995). Both of these factors may affect the accuracy of some ligand dockings. Through more advanced pre-processing steps we could add flexibility to the protein; this however adds computational cost as the possible docking conformations increases rapidly with the size of the protein. We can further develop this pipeline to allow for optional protein flexibility when rigid protein docking doesn't work. This can be done using the AutoDockTools utilities scripts. Methods of explicit hydration have been successfully incorporated into docking (Forli & Olson 2012), this could also be implemented into the pipeline.

Another limitation we face is that our pipeline fails to incorporate co-factors into the docking procedure. Currently only one protein and one ligand are docked together. This may result in inaccurate docking predictions for ligands that require cofactors.

When specifying the search space for the docking we draw the minimum sized box possible around the inherited coordinates and then increase the size of this box by 1 Å in each plane. This box area may be too small to dock a ligand into as ligands are not always fixed in the binding site and can move and rotate. We therefore need to account for this by increasing the size of the grid box. The developers of Autodock used an increase of 10 Å in each plane for their redocking studies to allow room for the ligand to rotate (Trott & Olson 2009). In the future we could set the increase to be 10 Å by default and we could also add a feature to our pipeline that allows users to specify the size of the search space manually.

Another cause of ligand docking inaccuracy is simply a lack of information on the template-ligand interaction meaning the coordinates inherited are insufficient to define the proper search space. One option to fix this is to perform a global search of the protein however this is computationally expensive and may not provide the correct docking. Instead a ligand binding site prediction software such a 3DLigandSite could be implemented into this pipeline (Wass et al. 2010). This would be useful for predicting binding sites when residue information is not available and also verifying the predicted bindings sites when residue information is available.

So far, our pipeline will not give a prediction on the effect of a missense mutation, it will simply output information on the dockings of the ligand with normal and mutant proteins and allow users to interpret the results. Further development of our pipeline to calculate the change in binding affinity and automate RMSD calculations between ligands would allow further streamlining of missense mutation analysis. We would then be able give a conclusive statement on the impact of a missense variant.

Another area of future development would be to create a user interface where users can run this pipeline by uploading files onto a GUI. This would allow for a more premium user experience as well as permitting the use of images and animations of the dockings to be displayed in real time for the user to visualise the effects of missense mutations.

Written by AS

When comparing a mutated protein structure generated by PyMOL in our pipeline with a real mutant structure in the PDB database, there are some inconsistencies in the side chain conformations of the mutated residues. As a result, the conformation of ligand docked into the mutated structure could be completely different from the ligand bound in the real mutant structure, see Figure 21 d for example. This could therefore lead to a false positive prediction of the impact of a missense variant. The problem could be addressed by using other programs to mutate the target residues, for instance, Missense3D which is a web-service used to predict the structural changes caused by missense variants (Ittisoponpisan et al. 2019).

However, the suggested solution is not guaranteed to resolve the issue as it is common for the computationally modelled structure and experimentally determined structure to have slightly different conformations especially in the side-chains. The reason for this is that the computational method would predict a protein conformation with the lowest energy while it is not necessary for the native state of the protein to have the lowest energy conformation (Deng et al. 2018). This principle also applies to the docked ligands as the conformation of the ligand bound inside the real structure might not correspond to the top ligand model with highest binding affinity predicted by Vina. In fact, it might be the lower rank models that have lower RMSD when compared to the real ligand conformation. This is due to the fact that other factors such as ions, co-factors, water molecules or even the flexibility of the protein itself can also play a critical roles in natural condition ligand binding.

The ultimate goal of the pipeline is to link with the LigandTemplateFinder pipeline and be integrated into Phyre2 as a new feature to predict the impact of missense variants on the interactions between query protein and target ligand. This would make it possible to automate the homology modelling of query structure and template selection process for homology-based docking as the LigandTemplateFinder provides a list of potential templates with target ligand bound. It would be much easier than manually selecting the template from Phyre2 since not all Phyre2 template have the target ligand bound. In addition, this increases the probability of choosing the best template leading to better docking quality because the LigandTemplateFinder searches for templates winthn the PDBe database while the library of Phyre2 templates is static and has a limited number of templates.

5 Conclusion

Written by CH

We have shown that existing template-based protein structure prediction servers, such as Phyre2 (Kelley et al. 2015), can be easily augmented to predict protein-ligand complexes. Using our conservative method, we are able to suggest templates for 71.1% of queries from a randomly selected dataset of 1,000 UniProt sequences. All of these templates contained the true biological ligand or a closely related analogue that can be used for starting coordinates when undertaking ligand modelling. Additionally, we have not only shown that the careful application of docking can be used to refine our predicted ligand poses, but also opens the possibility for accessing the impact that missense mutations have on ligand binding and therefore, disease.

Finally, we suggest that future efforts should be directed into integrating the two pieces of work to improve the Phyre2 (Kelley et al. 2015) and Missense3D (Ittisoponpisan et al. 2019) resources and creating a separate resource where suggested templates have been pre-computed for all UniProt sequences for the purposes of ligand modelling.

References

Awale, M. & Reymond, J.-L. (2014), 'A multi-fingerprint browser for the zinc database', *Nucleic acids research* **42**(W1), W234–W239.

Ball, P. (2008), 'Water as an active constituent in cell biology', *Chemical Reviews* **108**(1), 74–108. PMID: 18095715.
URL: *https://doi.org/10.1021/cr068037a*

Battaile, K. P., Molin-Case, J., Paschke, R., Wang, M., Bennett, D., Vockley, J. & Kim, J.-J. P. (2002), 'Crystal structure of rat short chain acyl-coa dehydrogenase complexed with acetoacetyl-coa: comparison with other acyl-coa dehydrogenases', *Journal of Biological Chemistry* **277**(14), 12200–12207.

Bennett, W. S., Huber, R. & Engel, J. (1984), 'Structural and functional aspects of domain motions in proteins', *Critical Reviews in Biochemistry* **15**(4), 291–384. PMID: 6325088.
URL: *https://doi.org/10.3109/10409238409117796*

Berman, H., Henrick, K., Nakamura, H. & Markley, J. L. (2007), 'The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data', *Nucleic Acids Research* **35**(Database), D301–D303.
URL: *https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkl971*

Berman, H. M. (2000), 'The Protein Data Bank', *Nucleic Acids Research* **28**(1), 235–242.
URL: *https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/28.1.235*

Bhattacharya, R., Rose, P. W., Burley, S. K. & Prlić, A. (2017), 'Impact of genetic variation on three dimensional structure and function of proteins', *PLOS ONE* **12**(3), e0171355.

Biasini, M., Bienert, S., Waterhouse, A., Arnold, K., Studer, G., Schmidt, T., Kiefer, F., Cassarino, T. G., Bertoni, M., Bordoli, L. et al. (2014), 'Swiss-model: modelling protein tertiary and quaternary structure using evolutionary information', *Nucleic acids research* **42**(W1), W252–W258.

Boehr, D. D., Nussinov, R. & Wright, P. E. (2009), 'The role of dynamic conformational ensembles in biomolecular recognition', *Nature chemical biology* **5**(11), 789–796.

Bolton, E. E., Wang, Y., Thiessen, P. A. & Bryant, S. H. (2008), Pubchem: integrated platform of small molecules and biological activities, *in* 'Annual reports in computational chemistry', Vol. 4, Elsevier, pp. 217–241.

Brylinski, M. & Skolnick, J. (2009), 'Findsite lhm: a threading-based approach to ligand homology modeling', *PLoS Comput Biol* **5**(6), e1000405.

Cereto-Massagué, A., Ojeda, M. J., Valls, C., Mulero, M., Garcia-Vallvé, S. & Pujadas, G. (2015), 'Molecular fingerprint similarity search in virtual screening', *Methods* **71**, 58–63. Virtual Screening.
URL: *https://www.sciencedirect.com/science/article/pii/S1046202314002631*

Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B. & de Hoon, M. J. L. (2009), 'Biopython: freely available Python tools for computational molecular biology and bioinformatics', *Bioinformatics* **25**(11), 1422–1423.

Cokelaer, T., Pultz, D., Harder, L. M., Serra-Musach, J. & Saez-Rodriguez, J. (2013), 'Bioservices: a common python package to access biological web services programmatically', *Bioinformatics* **29**(24), 3241–3242.

Csermely, P., Palotai, R. & Nussinov, R. (2010), 'Induced fit, conformational selection and independent dynamic segments: an extended view of binding events', *Nature Precedings* pp. 1–1.

Dana, J. M., Gutmanas, A., Tyagi, N., Qi, G., O'Donovan, C., Martin, M. & Velankar, S. (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 research* **47**(D1), D482–D489.

Deng, H., Jia, Y. & Zhang, Y. (2018), 'Protein structure prediction', *International Journal of Modern Physics B* **32**(18), 1840009.
URL: *https://www.worldscientific.com/doi/abs/10.1142/S021797921840009X*

Devos, D. & Valencia, A. (2000), 'Practical limits of function prediction', *Proteins: Structure, Function, and Bioinformatics* **41**(1), 98–107.

Du, X., Li, Y., Xia, Y.-L., Ai, S.-M., Liang, J., Sang, P., Ji, X.-L. & Liu, S.-Q. (2016), 'Insights into Protein–Ligand Interactions: Mechanisms, Models, and Methods', *International Journal of Molecular Sciences* **17**(2), 144.
URL: *http://www.mdpi.com/1422-0067/17/2/144*

Durant, J. L., Leland, B. A., Henry, D. R. & Nourse, J. G. (2002), 'Reoptimization of mdl keys for use in drug discovery', *Journal of chemical information and computer sciences* **42**(6), 1273–1280.

Fiser, A. (2010), 'Template-based protein structure modeling', *Computational biology* pp. 73–94.

Forli, S. & Olson, A. J. (2012), 'A force field with discrete displaceable waters and desolvation entropy for hydrated ligand docking', *Journal of medicinal chemistry* **55**(2), 623–638.

Gaulton, A., Hersey, A., Nowotka, M., Bento, A. P., Chambers, J., Mendez, D., Mutowo, P., Atkinson, F., Bellis, L. J., Cibrián-Uhalte, E. et al. (2017), 'The chembl database in 2017', *Nucleic acids research* **45**(D1), D945–D954.

Govindaraj, R. G. & Brylinski, M. (2018), 'Comparative assessment of strategies to identify similar ligand-binding pockets in proteins', *BMC bioinformatics* **19**(1), 1–17.

Greener, J. G., Kandathil, S. M. & Jones, D. T. (2019), 'Deep learning extends de novo protein modelling coverage of genomes using iteratively predicted structural constraints', *Nature communications* **10**(1), 1–13.

Hamelryck, T. & Manderick, B. (2003), 'PDB file parser and structure class implemented in Python', *Bioinformatics* **19**(17), 2308–2310.

Hassell, A. M., An, G., Bledsoe, R. K., Bynum, J. M., Carter, H. L., Deng, S.-J., Gampe, R. T., Grisard, T. E., Madauss, K. P., Nolte, R. T. et al. (2007), 'Crystallization of protein–ligand complexes', *Acta Crystallographica Section D: Biological Crystallography* **63**(1), 72–79.

Hastings, J., Owen, G., Dekker, A., Ennis, M., Kale, N., Muthukrishnan, V., Turner, S., Swainston, N., Mendes, P. & Steinbeck, C. (2016), 'Chebi in 2016: Improved services and an expanding collection of metabolites', *Nucleic acids research* **44**(D1), D1214–D1219.

Ittisoponpisan, S., Islam, S. A., Khanna, T., Alhuzimi, E., David, A. & Sternberg, M. J. (2019), 'Can predicted protein 3d structures provide reliable insights into whether missense variants are disease associated?', *Journal of Molecular Biology* **431**(11), 2197–2212.

Jelenković, P. R. & Radovanović, A. (2004), 'Least-recently-used caching with dependent requests', *Theoretical computer science* **326**(1-3), 293–327.

Jones, D. T. (1999), 'Protein secondary structure prediction based on position-specific scoring matrices', *Journal of molecular biology* **292**(2), 195–202.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Tunyasuvunakool, K., Ronneberger, O., Bates, R., Zidek, A., Bridgland, A. et al. (2020), 'High accuracy protein structure prediction using deep learning', *Fourteenth Critical Assessment of Techniques for Protein Structure Prediction (Abstract Book)* **22**, 24.

Kelley, L. A., Mezulis, S., Yates, C. M., Wass, M. N. & Sternberg, M. J. E. (2015), 'The Phyre2 web portal for protein modeling, prediction and analysis', *Nature Protocols* **10**(6), 845–858.

Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B. A., Thiessen, P. A., Yu, B. et al. (2021), 'Pubchem in 2021: new data content and improved web interfaces', *Nucleic Acids Research* **49**(D1), D1388–D1395.

Koshland Jr, D. (1958), 'Application of a theory of enzyme specificity to protein synthesis', *Proceedings of the National Academy of Sciences of the United States of America* **44**(2), 98.

Landrum, G. (2016), 'Rdkit: Open-source cheminformatics software'.
URL: *https://github.com/rdkit/rdkit/releases/tag/Release*$_2$016$_0$9$_4$

Leach, A. R. & Gillet, V. J. (2007), *An introduction to chemoinformatics*, Springer.

Liu, X., Jian, X. & Boerwinkle, E. (2011), 'dbNSFP: A lightweight database of human nonsynonymous SNPs and their functional predictions', *Human Mutation* **32**(8), 894–899.
URL: *https://onlinelibrary.wiley.com/doi/10.1002/humu.21517*

Mackenzie, J., Pedersen, L., Arent, S. & Henriksen, A. (2006), 'Controlling electron transfer in acyl-coa oxidases and dehydrogenases: A structural view*', *Journal of Biological Chemistry* **281**(41), 31012–31020.
URL: *https://www.sciencedirect.com/science/article/pii/S0021925819893158*

Malik, R. & Viola, R. E. (2010), 'Structural characterization of tartrate dehydrogenase: a versatile enzyme catalyzing multiple reactions', *Acta Crystallographica Section D: Biological Crystallography* **66**(6), 673–684.

Menozzi, I., Vallese, F., Polverini, E., Folli, C., Berni, R. & Zanotti, G. (2017), 'Structural and molecular determinants affecting the interaction of retinol with human CRBP1', *Journal of Structural Biology* **197**(3), 330–339.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S1047847716302672*

Michino, M., Abola, E., Brooks, C. L., Dixon, J. S., Moult, J. & Stevens, R. C. (2009), 'Community-wide assessment of gpcr structure modelling and ligand docking: Gpcr dock 2008', *Nature Reviews Drug Discovery* **8**(6), 455–463.

Morgat, A., Lombardot, T., Coudert, E., Axelsen, K., Neto, T. B., Gehant, S., Bansal, P., Bolleman, J., Gasteiger, E., De Castro, E. et al. (2020), 'Enzyme annotation in uniprotkb using rhea', *Bioinformatics* **36**(6), 1896–1901.

Morris, G. M., Huey, R., Lindstrom, W., Sanner, M. F., Belew, R. K., Goodsell, D. S. & Olson, A. J. (2009), 'AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility', *Journal of Computational Chemistry* **30**(16), 2785–2791.
URL: *http://doi.wiley.com/10.1002/jcc.21256*

Muegge, I. & Mukherjee, P. (2016), 'An overview of molecular fingerprint similarity search in virtual screening', *Expert Opinion on Drug Discovery* **11**(2), 137–148. PMID: 26558489.
URL: *https://doi.org/10.1517/17460441.2016.1117070*

Ortiz, A. R., Strauss, C. E. & Olmea, O. (2002), 'Mammoth (matching molecular models obtained from theory): an automated method for model comparison', *Protein Science* **11**(11), 2606–2621.

pandas development team, T. (2020), 'pandas-dev/pandas: Pandas 1.1.3'.
URL: *https://doi.org/10.5281/zenodo.4067057*

Poornima, C. & Dean, P. (1995), 'Hydration in drug design. 1. multiple hydrogen-bonding features of water molecules in mediating protein-ligand interactions', *Journal of computer-aided molecular design* **9**(6), 500–512.

Rácz, A., Bajusz, D. & Héberger, K. (2018), 'Life beyond the tanimoto coefficient: similarity measures for interaction fingerprints', *Journal of cheminformatics* **10**(1), 1–12.

Rijnbeek, M. & Steinbeck, C. (2009), 'Orchem-an open source chemistry search engine for oracle®', *Journal of cheminformatics* **1**(1), 1–11.

Roche, D. B., Buenavista, M. T. & McGuffin, L. J. (2013), 'The funfold2 server for the prediction of protein–ligand interactions', *Nucleic acids research* **41**(W1), W303–W307.

Schrödinger, LLC (2020), The PyMOL molecular graphics system, version 2.4.1.

Skinnider, M. A., Dejong, C. A., Franczak, B. C., McNicholas, P. D. & Magarvey, N. A. (2017), 'Comparative analysis of chemical similarity methods for modular natural products with a hypothetical structure enumeration algorithm', *Journal of cheminformatics* **9**(1), 1–15.

Söding, J. (2005), 'Protein homology detection by hmm–hmm comparison', *Bioinformatics* **21**(7), 951–960.

Sterling, T. & Irwin, J. J. (2015), 'Zinc 15–ligand discovery for everyone', *Journal of chemical information and modeling* **55**(11), 2324–2337.

Tipton, P. A. & Peisach, J. (1990), 'Characterization of the multiple catalytic activities of tartrate dehydrogenase', *Biochemistry* **29**(7), 1749–1756.

Trott, O. & Olson, A. J. (2009), 'AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading', *Journal of Computational Chemistry* .

Tyzack, J. D., Fernando, L., Ribeiro, A. J., Borkakoti, N. & Thornton, J. M. (2018), 'Ranking enzyme structures in the pdb by bound ligand similarity to biological substrates', *Structure* **26**(4), 565–571.

UniProt: The universal protein knowledgebase in 2021 (2021), *Nucleic Acids Research* **49**(D1), D480–D489.

Varadi, M., Berrisford, J., Deshpande, M., Nair, S. S., Gutmanas, A., Armstrong, D., Pravda, L., Al-Lazikani, B., Anyango, S., Barton, G. J. et al. (2020), 'Pdbe-kb: a community-driven resource for structural and functional annotations', *Nucleic Acids Research* **48**(D1), D344–D353.

Velankar, S., Dana, J. M., Jacobsen, J., Van Ginkel, G., Gane, P. J., Luo, J., Oldfield, T. J., O'Donovan, C., Martin, M.-J. & Kleywegt, G. J. (2012), 'Sifts: structure integration with function, taxonomy and sequences resource', *Nucleic acids research* **41**(D1), D483–D489.

Wass, M. N., Kelley, L. A. & Sternberg, M. J. (2010), '3dligandsite: predicting ligand-binding sites using similar structures', *Nucleic acids research* **38**(suppl_2), W469–W473.

Westbrook, J. D., Shao, C., Feng, Z., Zhuravleva, M., Velankar, S. & Young, J. (2015), 'The chemical component dictionary: complete descriptions of constituent molecules in experimentally determined 3d macromolecules in the protein data bank', *Bioinformatics* **31**(8), 1274–1278.

Wu, Q., Peng, Z., Zhang, Y. & Yang, J. (2018), 'Coach-d: improved protein–ligand binding sites prediction with refined ligand-binding poses through molecular docking', *Nucleic acids research* **46**(W1), W438–W442.

Xie, L., Xu, L., Kong, R., Chang, S. & Xu, X. (2020), 'Improvement of prediction performance with conjoint molecular fingerprint in deep learning', *Frontiers in pharmacology* **11**.

Xin, J., Mark, A., Afrasiabi, C., Tsueng, G., Juchler, M., Gopal, N., Stupp, G. S., Putman, T. E., Ainscough, B. J., Griffith, O. L., Torkamani, A., Whetzel, P. L., Mungall, C. J., Mooney, S. D., Su, A. I. & Wu, C. (2016), 'High-performance web services for querying gene and variant annotation', *Genome Biology* **17**(1), 91.
URL: *https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0953-9*

Yang, J., Anishchenko, I., Park, H., Peng, Z., Ovchinnikov, S. & Baker, D. (2020), 'Improved protein structure prediction using predicted interresidue orientations', *Proceedings of the National Academy of Sciences* **117**(3), 1496–1503.

Zhao, J., Cao, Y. & Zhang, L. (2020), 'Exploring the computational methods for protein-ligand binding site prediction', *Computational and structural biotechnology journal* **18**, 417–426.

6 Appendix

Table 3: Chemical compounds and associated ChEBI IDs returned as "catalytic activity/cofactors"

Chemical Name	**ChEBI IDs with 100% ChEBI similarity**
ATP	30616, 57299, 15422, 145541
ADP	16761, 143969, 87518, 456216
GTP	57600, 15996, 145560, 62908, 37565, 85961, 53011, 143964, 142410, 58215, 87133, 71477, 77828, 74655, 74429, 50226, 50210, 17633, 16690
GDP	58189, 17552, 143970, 65180, 58595, 63729, 63714, 28862, 63730
NADH	16908, 77311, 57945, 143948, 77176, 57783, 77177, 77312, 16474
NADP(3-)	18009, 44409, 77018, 58349, 77174, 57540, 77173, 143906, 77017, 44215, 15846
NADPH	16474, 77312, 77177, 57783, 57945, 143948, 77176, 16908, 77311
FAD	16238, 57692, 143270
FADH	17877, 58307, 30788
Water	15377, 149692, 25805, 29412, 30490, 29191, 29193, 29194, 33806, 33811, 33813, 33815, 33818, 33819, 29356, 41981, 29374, 29915, 36932, 36933, 29375, 16234
S-adenosyl-L-methionine	15414, 33440, 33442, 67040, 67009, 59789, 142093, 16680, 156255, 57856, 142094
adenosine 5'-monophosphate	16027, 28931, 28223, 40721, 60880, 456215, 95121, 77740, 143978, 53098
hydrogenphosphate	7794, 138518, 44976, 26078, 33462, 48108, 30931, 39745, 43474, 29925, 29931, 29932, 18367, 52915, 52641, 67140, 68546, 16838, 32958, 71285, 16215, 140358
diphosphate(3-)	18036, 48314, 18361, 29888, 39949, 32959, 33017, 45212, 33019, 48313, 48315, 48316, 16838, 68836, 68549, 68550