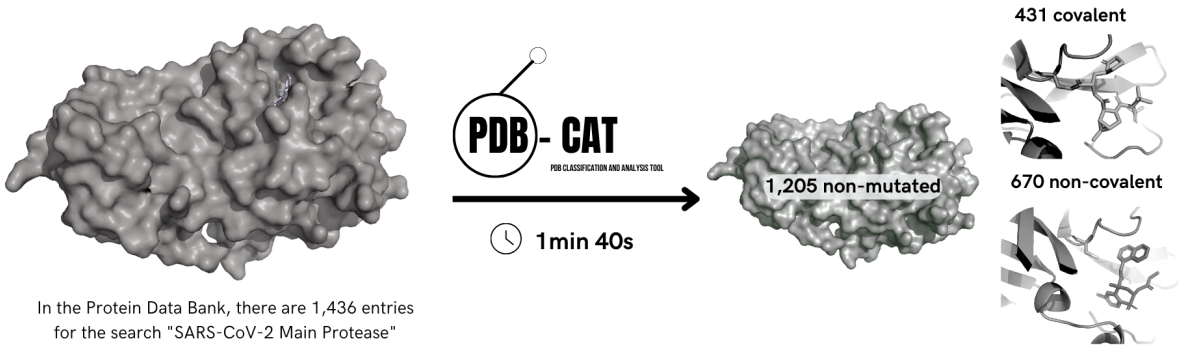
PDB-CAT: A User-Friendly Tool to Classify and Analyze PDB Protein-Ligand Complexes

Ariadna Llop-Peiró[1]*, Gerard Pujadas[1], Santiago Garcia-Vallvé[1] and Aleix Gimeno[1]

1. *Departament de Bioquímica i Biotecnologia, Universitat Rovira i Virgili, Research group in Cheminformatics & Nutrition, 43007 Tarragona, Catalonia, Spain*.

1. *Abstract*

The Protein Data Bank contains more than 223,000 three-dimensional biostructures and is growing at a rate of nearly 10% per year. The lack of a tool that facilitates the classification between apo and holo structures and differentiates between covalent and non-covalent ligand-protein complexes, makes it difficult to manage a large number of structures. To address this issue, we present PDB-CAT, a user-friendly tool that facilitates the categorization and extraction of key information from PDBx/mmCIF files. PDB-CAT is a program that classifies a group of protein structures based on their ligands into three categories: apo, covalently, and non-covalently bonded. Besides this classification, the program can verify if there are any mutations in the protein sequence by comparing it to a reference sequence. PDB-CAT is designed to be user-friendly, with its output clearly defining every entity present in each entry to facilitate decision-making. PDB-CAT is now available on GitHub (https://github.com/URV-cheminformatics/PDB-CAT).



Graphical Abstract

2. *Scientific Contribution*

Based on our understanding, there is currently no open-source automated tool developed for classifying Protein Data Bank structures into apo and holo forms, and further categorizing them based on the type of bond between the ligand and protein, which can be covalent or non-covalent. The straightforward and user-friendly PDB-CAT tool provides a quick and efficient way to address this issue.

3. *Keywords*

Protein Data Bank, Structure-based Drug Discovery, Protein-ligand complexes, Protein structure analysis, PDBx/mmCIF

4. *Introduction*

The use of computational tools, specifically high-throughput virtual screening (HTVS), has emerged as an efficient strategy for Drug Discovery (Gimeno et al. 2019). HTVS includes approaches such as molecular docking, and pharmacophore modelling, which have successfully been employed in the discovery of novel hits for various therapeutic targets (Kumalo et al. 2015). Computational-aided drug discovery approaches can be divided into two modalities: structure-based, centering on the biological target, or ligand-based, focusing on the structural and physicochemical ligand properties (Vázquez et al. 2020). To begin working with structure-based computational tools, the first step is to search for crystallized structures of the therapeutic target. One of the most popular databases is the Protein Data Bank (PDB) (Burley et al. 2023). According to the RCSB PDB data, the PDB contains more than 223,000 structures and it is expanding rapidly, with 14,472 new structures released just in 2023 (Fig. 1). In addition, the PDB also contains more than one million computed structure models from artificial intelligence, such as AlphaFold models (Jumper 2021).

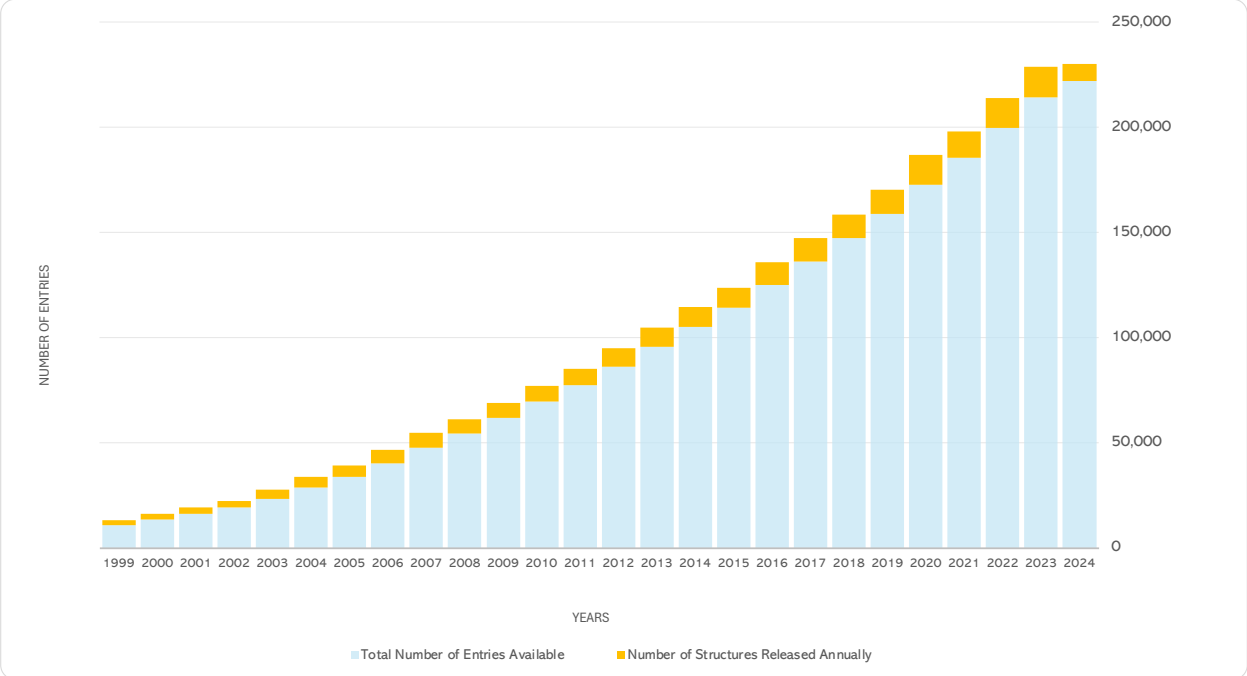


Figure 1. The PDB statistics: exponential growth of deposited structure in the PDB. In blue, the total number of entries available; in orange, the number of structures released each year (The RCSB PDB 2024).

In certain virtual screening (VS) studies, multiple structures of the same protein target may be available. This is the case of the SARS-CoV-2 Main Protease (M-pro). The global collaboration triggered by the SARS-CoV-2 pandemic has led to an unprecedented accumulation of data (Adamson et al. 2021). As a result, more than 1,400 crystal structures of SARS-CoV-2 M-pro have been deposited in the PDB. This abundance also underscores the importance of validating a specific set of structures before selecting one or several of them and beginning the VS process (Macip et al. 2022). Furthermore, not all structures may be appropriate for a certain purpose. Drug Discovery protocols change depending on whether the ligands are covalently or non-covalently bound with the protein. Therefore, it is important to distinguish between PDB structures with ligands that are covalently or non-covalently bound. In the case of SARS-CoV-2 M-pro, both types of ligands are present. In cases where a crystallized protein-ligand complex is available, it is recommended to avoid apo forms (Schaefer and Cheng 2023). In this context, we refer to the protein without ligands as the apo form, as this term is widely used in the drug discovery scientific community and enhances clarity (Khachatryan et al. 2024). Crystallized protein-ligand complexes provide detailed information on how the ligand binds to the protein and induces specific conformational changes in the active site. Such data are crucial for understanding binding affinity and designing effective inhibitors. Apo forms lack this context, as they do not reveal possible ligand-induced structural adjustments and interactions.

The PDB lacks an option in its advanced filter to distinguish between apo-form and holo-form, as well as to differentiate between ligand-protein complexes that are non-covalently or covalently bound. Moreover, no other tool has been found that performs this classification automatically without the need for manual searching. For that reason, we have developed PDB-CAT, a tool to automate the classification of PDBx/mmCIF structures depending on whether they are in their apo-form or if their ligands are bound covalently or non-covalently. The PDBx/mmCIF format is the standard PDB archive distribution format and it overcomes the limitations of the older PDB file format. As the PDBx/mmCIF format continues to evolve, PDB format files will become outdated (The RCSB PDB 2024b). Besides the classification based on the ligand, PDB-CAT extracts information about all entities presents in a PDB entry and can verify if there are any mutations in the protein

sequence by comparing it to a reference sequence. In the case of SARS-CoV-2 M-pro, this option is very useful, as there are several mutated sequences derived from different variants of the SARS-CoV-2 virus (Saldivar-Espinoza et al. 2023) .

5. *Design*

To parse PDBx/mmCIF files, PDB-CAT follows the entity hierarchy, which is central to the mmCIF format. This format defines a molecular entity as a chemically distinct component within an entry. PDB-CAT categorizes each entity into three classifications: *polymer*, *non-polymer*, and *branched*.

Protein and ligand identification

Figure 2 summarizes the different steps PDB-CAT follows to identify and classify the ligands. Alternatively, if an mmCIF file contains no identified ligands, it is classified as an apo form and labelled as *APO*. To identify the main protein or proteins from a PDB file, they are always defined as a polypeptide polymer, either isolated from a natural source or isolated from a genetically manipulated source and conformed by several residues (Fig. 2). All this information is located in the *entity* and *entity_poly* categories of the PDBx/mmCIF format.

The PDB categorizes small molecules such as ions, cofactors, inhibitors, and drugs as ligands. However, it is not straightforward to identify polymeric entities like peptide or saccharide ligands, as the PDB typically classifies them as separate entities rather than ligands. PDB-CAT addresses this issue by facilitating the identification of ligands and solvents, thus helping drug discovery scientists. After identifying the protein, any other polypeptide polymer entities present in the structure complex are classified as peptide ligands, using a threshold length variable, which is set to 15 residues by default but can be modified by the user (Fig. 2). If the entity's length is higher than the threshold, then it will be classified as another chain or subunit. Unlike the protein, a peptide ligand can also be a synthetic polymer. The Biologically Interesting Molecule Reference Dictionary (BIRD) dictionary (https://www.wwpdb.org/data/bird), contains information of peptide-like inhibitors and common oligosaccharides. Some of the mmCIF files contain these BIRD IDs, hence PDB-CAT checks for BIRD IDs to retrieve more information about the ligands.

The next entity type is *non-polymer*, typically referring to small molecules. The initial step to consider a non-polymer entity as a ligand involves checking whether the small molecule is listed in a blacklist

(Fig. 2). The blacklist consists of solvents, ions, and co-factors and can be modified by the user, depending on the target analyzed. For example, a co-factor bound to a viral protease might be discarded by some computational chemists, while in other enzymes, it could be important to consider. If a match is found with any element in the blacklist, the small molecule is then added to the list of discarded ligands (Fig. 2). PDB-CAT also verifies the Chemical Component Dictionary (CCD) ID (https://www.wwpdb.org/data/ccd), which details small molecule components, to gather additional information about the ligands, similarly to how it uses the BIRD ID.

The last entity type in the PDBx/mmCIF format is the *branched* type, where oligosaccharides are commonly categorized. In this case, the presence of a covalent bond with the protein is straightly considered. If an oligosaccharide forms a covalent bond with the protein, it is classified as a glycosylation. Otherwise, it is classified as a saccharide ligand, and look for BIRD IDs to retrieve more information about the saccharide (Fig. 2).

Covalently or non-covalently bonded ligands

The categorization between covalently and non-covalently bonded ligands is determined by the presence of a covalent bond between the ligand and the protein. If a covalent bond between a ligand and any of the protein subunits is found, the ligand is classified as a covalently bonded ligand, and information on the specific amino acid to which it is attached is provided. If no covalent bond is found, the ligand is classified as non-covalently bonded ligand (Fig. 2).

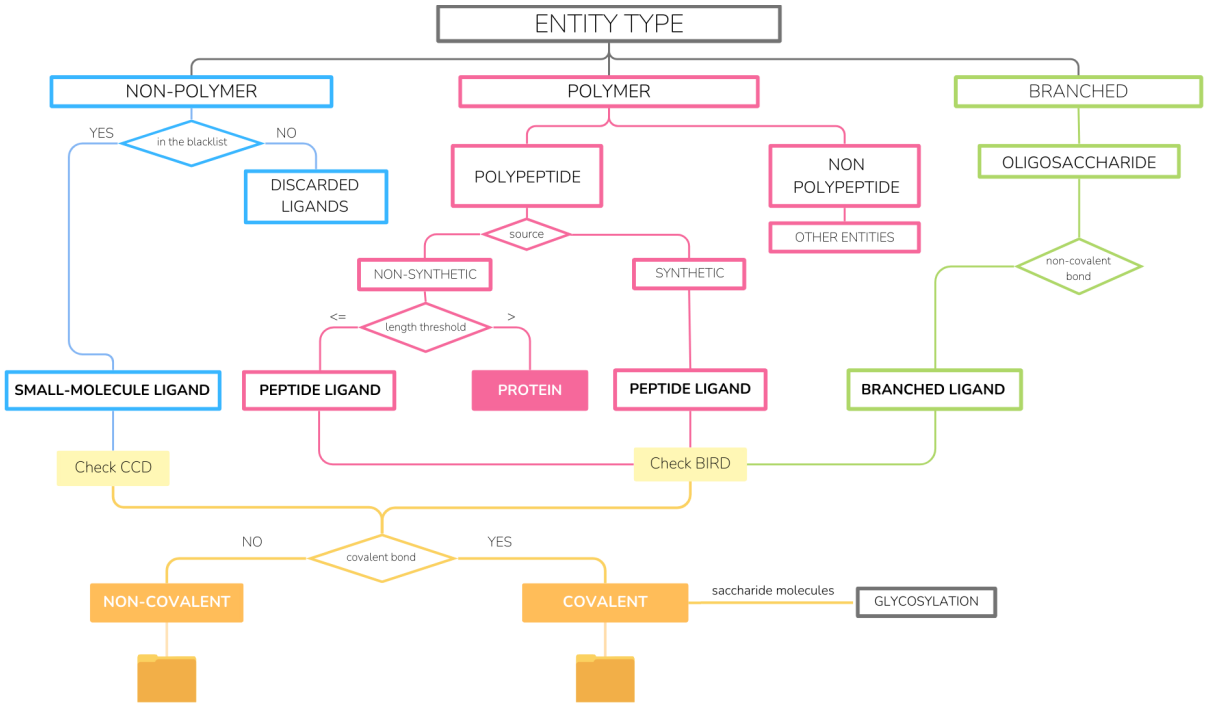


Figure 2. Flowchart illustrating the steps taken by PDB-CAT to identify and classify ligands. The process includes ligand detection, and further categorization based on the presence or absence of a covalent bond.

Mutation Analysis

The PDB-CAT algorithm can be performed in two ways: by first identifying mutations and then classifying the dataset, or by classifying the complete dataset directly. This can be defined in the Boolean variable: *mutation*. The mutation analysis compares the sequence of the protein entities to a reference sequence. This reference sequence should be defined before running the program and it is supposed to be a PDB structure known by the users, ensuring it is free of mutations and containing all the residues. Only one reference sequence can be defined, hence this option is only useful when analyzing structures of the same protein.

The program utilizes the Pairwise Alignment module of the biopython library (PMID-19304878) to extract information about mutations, residue locations, percentage of identity, and gaps between the sequences mentioned earlier. This information is available in the CSV output.

Output

The PDB-CAT program generates two CSV files: one protein-centered and the other ligand-centered. In the first CSV file, each line corresponds to a PDB ID and provides a comprehensive set of information about the entry. This section includes details related to the protein, such as the title of the PDB file, protein description, number of subunits, and subunit IDs (referred to as chains), along with the number of residues for each subunit. It also indicates whether the protein is part of a complex. Following this, the CSV includes information about discarded ligands, elements from a blacklist that are bonded to the protein, and branched molecules. For each branched molecule, details such as name, type, function, and the presence of a covalent bond are provided. Next, the CSV presents information about ligands, including their name, type, function, and whether they form a covalent bond with the protein. The final columns cover mutation information, specifying the number of mutations, their locations, identity percentages, and any gaps present in the sequence.

In the second CSV file, each line corresponds to an entity bonded to a protein. The format is straightforward, detailing the ID of the protein and the bonded molecule including its name, type, function, and whether it forms a covalent bond. If a covalent bond is present, the specific residue with which it binds is specified. Additionally, if the bonded molecule is a glycosylation, this information is also provided.

Finally, the program creates separate folders to categorize apo structures, covalent complexes, and non-covalent complexes. When the *mutation* filter is applied, this classification occurs within the non-mutated folder. Additionally, a mutated folder is created alongside the non-mutated one (Fig. 3).

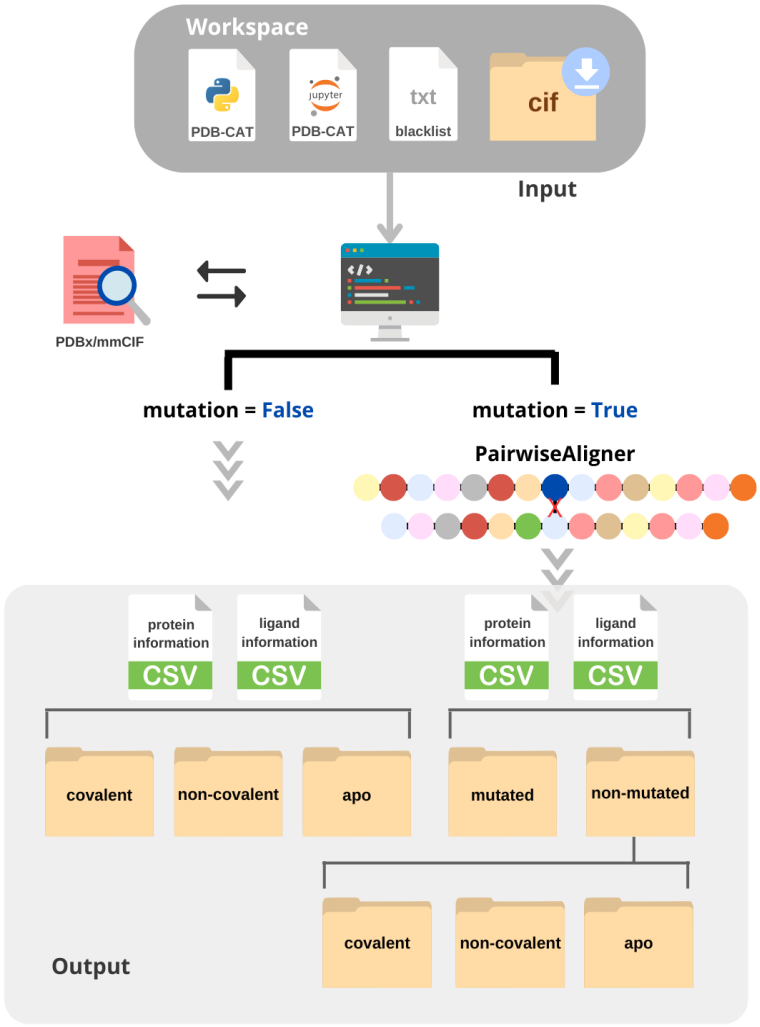


Figure 3. Workflow Diagram of the PDB-CAT: Within the workspace, you will find a Jupyter Notebook file, a Python module containing some functions, the blacklist file, and the directory containing the input files in mmCIF format. To execute the *mutation* mode, the reference file should be part of the dataset. The program generates two CSV files containing all relevant information, as well as several folders.

6. *Implementation*

Availability

The source code is readily available as a Jupyter Notebook on GitHub (https://github.com/URV-cheminformatics/PDB-CAT). It can be cloned following the instructions written in the readme file, or it can be opened directly in Google Colab for those who are less familiar with coding.

How to use

Before running this Notebook, the initial step is to establish the dataset of structures. This involves downloading the structure files locally from the PDB. Given the ongoing transition of PDB to the PDBx/mmCIF format, it is essential that the input files are in this format. The mmCIF files should be save in the *cif* folder.

The blacklist can be found in the GitHub repository and as mentioned, it is an editable list of solvent, co-factor, and ion IDs — a collection of small molecules that are not considered ligands. This blacklist should be customized to reflect the unique properties of each protein of interest. To remove an element in the blacklist just comment the line by writing the "#" symbol at the beginning of the line.

The PDB-CAT notebook includes a dedicated cell for code customization, ensuring clear and concise interaction with the code. There are eight variables in the main code that can be modified; each is detailed to help the user understand how to customize them. Note that the notebook can also be run by default to classify the structures available in the cif folder, without using the *mutation* filter.

Requeriments

This program uses Python 3 and requires the following packages: *biopython*, *pdbecif*, *pandas*, *re*, *os*, and *shutil*. Additionally, the pdbcat module, which is in the repository, should be imported. The GitHub repository includes a *requirements.txt* file to simplify the installation process, which is automatically handled in Google Colab environments.

7. *Results*

Validation

To validate PDB-CAT, a dataset of protein–ligand complexes was extracted from the refined set of PDBbind (Wang et al. 2004). The PDBbind database (http://www.pdbbind.org.cn/) is the largest collection of protein–ligand complexes, providing information on both binding affinities and known 3D crystal structures (Wang et al. 2015). Updated annually, the 2020 version comprises 19,443 protein–ligand complexes featuring experimentally measured binding affinity data.

PDB-CAT efficiently analyzed this dataset of 19,443 protein-ligand complexes in under 20 minutes. Of the 19,443 entries, 2.83% (550 entries) were apo forms. Specifically, 470 cases were linked to a blacklist component, identified as a ligand in the PDBBind dataset. Additionally, 70 entries had a peptide ligand with 15 or more residues. The remaining 10 cases were complexes with nucleic acids, categorized here as apo forms, but the PDB-CAT still provides information about the bonding with nucleic acids in an *Other Entities* column. Note that the *mutation* filter was not used in this validation because of the diversity of proteins found in the dataset.

We also used the PDBBind dataset to validate the classification of covalent and non-covalent ligands. From the 19,443 protein-ligand complexes, PDBBind identifies 315 as covalent complexes. PDB-CAT classified 40 of these 315 complexes as non-covalent. This discrepancy arises because, while the proximity of atoms suggests the presence of a covalent bond, the PDB files do not explicitly specify its presence. This is a limitation of the PDB-CAT program, as it exclusively relies on the information available in the PDB files. Additionally, PDB-CAT identified 1,285 covalent complexes within the PDBBind 2020 dataset. We reviewed a portion of these complexes to verify the presence of a covalent bond between the ligand and the protein, showing that the PDBBind dataset does not provide a complete classification of covalently bound ligands, highlighting the utility of our program.

SARS-CoV-2 Main Protease (M-pro)

As an example of the use of the PDB-CAT program, 1,436 PDB structures containing the SARS-CoV-2 M-pro were analyzed. The PDBx/mmCIF files underwent a thorough analysis and mutation categorization. Out of the 1,436 M-pro structures downloaded from the PDB, 1,205 were identified as non-mutated. These structures were further classified into 104 apo structures, 431 covalent complexes, and 670 non-covalent complexes. Additionally, among the covalent complexes, 27 were specifically identified as peptide ligands. A CSV file was also created to compile all the crucial information. For each ligand code, information related to the specific chain letter identifier and residue number was collected. As mentioned previously, details about the type of bond and the peptide nature were described for each case.

Furthermore, as the *mutation* option was executed, 231 mutations were analyzed. Information about the exact mutated residue, identity percentage, and gaps in the sequence compared to the reference sequence was extracted.

8. *Conclusions*

PDB-CAT is a unique tool for classifying PDB structures into ligand-free forms, covalent complexes, and non-covalent complexes and for detecting mutations and gaps between structures of the same protein. Additionally, it serves as a valuable resource for researchers managing the vast amount of data from the Protein Data Bank, especially for computational chemists that have to deal with multiple structures of the same protein. This program also contributes to the format transition from PDB to PDBx/mmCIF.

9. *Availability of data and materials*

The software and dataset are open-source and available for public use under the GNU Affero General Public License v3.0. Project name: PDB-CAT; Project Homepage: https://github.com/URV-cheminformatics/PDB-CAT; Installation Instructions: can be found at: https://github.com/URV-cheminformatics/PDB-CAT/README.md or https://ariadnallopps-organization.gitbook.io/pdb-cat/ Operating Systems: Platform independent; Programming Language: Python; Other Requirements: dependencies are listed with installation instructions; License: GNU Affero General Public License v 3.0; Data: Included with package on download or can be found online in the source repository: https://github.com/URV-cheminformatics/PDB-CAT /example

10. *Acknowledgements*

This work was supported by the project PID2022-138327OB-I00 financed by financed by the Ministerio de Ciencia e Innovación (MCIN)/Agencia Estatal de Investigación (AEI)/10.13039/501100011033/FEDER, UE. AL-P. is recipient of the pre-doctoral grant 2022PMF-INV-14 from the INVESTIGO call that is financed by the Next Generation EU program (through the Recovery and Resilience Facility initiative), the Public Service of State Employment (SEPE) from the Spanish Government and Universitat Rovira i Virgili

References

1. Adamson CS, Chibale K, Goss RJM, et al (2021) Antiviral drug discovery: Preparing for the next pandemic. Chem Soc Rev 50:3647–3655

2. Burley SK, Bhikadiya C, Bi C, et al (2023) RCSB Protein Data Bank (RCSB.org): delivery of experimentally-determined PDB structures alongside one million computed structure models of proteins from artificial intelligence/machine learning. Nucleic Acids Res 51:D488–D508. https://doi.org/10.1093/nar/gkac1077
3. Gimeno A, Ojeda-Montes MJ, Tomás-Hernández S, et al (2019) The light and dark sides of virtual screening: What is there to know? Int J Mol Sci 20
4. Jumper J et al. (2021) Highly accurate protein structure prediction with AlphaFold. Nature 596:583–589. https://doi.org/10.1038/s41586-021-03819-2
5. Khachatryan H, Matevosyan M, Harutyunyan V, et al (2024) Computational evaluation and benchmark study of 342 crystallographic holo-structures of SARS-CoV-2 Mpro enzyme. Sci Rep 14:. https://doi.org/10.1038/s41598-024-65228-5
6. Kumalo HM, Bhakat S, Soliman MES (2015) Theory and applications of covalent docking in drug discovery: Merits and pitfalls. Molecules 20:1984–2000
7. Macip G, Garcia-Segura P, Mestres-Truyol J, et al (2022) Haste makes waste: A critical review of docking-based virtual screening in drug repurposing for SARS-CoV-2 main protease (M-pro) inhibition. Med Res Rev 42:744–769
8. Saldivar-Espinoza B, Garcia-Segura P, Novau-Ferré N, et al (2023) The Mutational Landscape of SARS-CoV-2. Int J Mol Sci 24:. https://doi.org/10.3390/ijms24109072
9. Schaefer D, Cheng X (2023) Recent Advances in Covalent Drug Discovery. Pharmaceuticals 16
10. The RCSB PDB (2024) PDB Statistics: Overall Growth of Released Structures Per Year. In: The RCSB PDB
11. Vázquez J, López M, Gibert E, et al (2020) Merging ligand-based and structure-based methods in drug discovery: an overview of combined virtual screening approaches. Molecules 25
12. Wang R, Fang X, Lu Y, Wang S (2004) The PDBbind database: Collection of binding affinities for protein-ligand complexes with known three-dimensional structures. J Med Chem 47:2977–2980. https://doi.org/10.1021/jm030580l
13. Wang Y, Guo Y, Kuang Q, et al (2015) A comparative study of family-specific protein-ligand complex affinity prediction based on random forest approach. J Comput Aided Mol Des 29:349–360. https://doi.org/10.1007/s10822-014-9827-y