Received: 2 July 2025 | Revised: 22 October 2025 | Accepted: 29 October 2025

DOI: 10.1002/pro.70379

TOOLS FOR PROTEIN SCIENCE

THE PROTEIN SOCIETY WILEY

PDB-CAT: A user-friendly tool to classify and analyze PDB protein–ligand complexes

Ariadna Llop-Peiró | **Said Trujillo-De León** | **Gerard Pujadas** | **Santiago Garcia-Vallvé** | **Aleix Gimeno**

Departament de Bioquímica i Biotecnologia, Research Group in Cheminformatics & Nutrition, Universitat Rovira i Virgili, Tarragona, Spain

Correspondence
Santiago Garcia-Vallvé and Aleix Gimeno, Universitat Rovira i Virgili. Departament de Bioquímica i Biotecnologia, Research Group in Cheminformatics & Nutrition, c/marcel.li domingo 1, 43007 Tarragona, Spain.
Email: santi.garcia-vallve@urv.cat; aleix.gimeno@urv.cat

Funding information
MCIN/AEI/10.13039/501100011033/FEDER, UE., Grant/Award Number: PID2022-138327OB-I00; Next Generation EU program, Grant/Award Number: 2022PMF-INV-14

Review Editor: Nir Ben-Tal

Abstract

The Protein Data Bank (PDB) contains more than 235,000 three-dimensional biostructures and is growing at a rate of nearly 10% per year. The PDB is essential to gain knowledge on how proteins and ligands interact and how these interactions are correlated with the quantitative activity of each ligand/target pair. Unfortunately, the lack of a tool that can classify structures as apo or holo, that is by no means straightforward, and differentiate between covalent and non-covalent ligand–protein complexes makes it difficult to obtain the structures that belong to each set. 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 through an efficient parallelized implementation. PDB-CAT uses a blacklist-based approach to automatically identify the ligand in a complex. It then classifies the PDB files based on ligand presence: structures without a ligand are classified as apo, whereas those with a ligand are classified as covalently or non-covalently bound, depending on the type of binding. As well as making this classification, the program can verify if there are any mutations in the protein sequence by comparing it to a reference sequence. An example is included to illustrate two different uses: the classification of SARS-CoV-2 Main Protease complexes depending on their variant, and the complete screening of the PDBbindv2020, achieved in <10 min. PDB-CAT is now available on GitHub (https://github.com/URV-cheminformatics/PDB-CAT) and the corresponding tutorial on GitBook (https://ariadnallopps-organization.gitbook.io/pdb-cat).

KEYWORDS
PDBx/mmCIF, protein data Bank, protein–ligand complexes, structural bioinformatics, structure-based drug discovery

1 | INTRODUCTION

The use of computational approaches, specifically virtual screening (VS), has emerged as an efficient strategy for drug discovery (Gimeno et al., 2019). VS includes approaches such as molecular docking and pharmacophore modeling, which have been successfully used to discover novel hits for various therapeutic targets (Kumalo et al., 2015). Computer-aided drug discovery approaches can be divided into two types: structure-based, which focus on the biological target, or ligand-based, which focus on the structural and physicochemical properties of ligands (Vázquez et al., 2020). One of the most widely used

tools in structure-based drug discovery is protein-ligand docking, which uses the individual structures of both the protein and ligand to predict the 3D coordinates of their resulting complex (Paggi et al., 2024). Docking programs typically generate multiple possible binding orientations for the ligand, known as docked poses, and then use a scoring function to rank these poses in terms of their predicted affinity for the target (Yang et al., 2022). Training scoring functions with experimental affinity data from 3D complexes in the PDB (Burley et al., 2023) has proven essential for improving the quantitative prediction of binding affinity (Wang et al., 2023).

According to the RCSB PDB data, the PDB contains more than 235,000 structures and it is expanding rapidly, with more than 6000 new structures released just in 2025 (Figure 1) (RCSB PDB, 2024a). In the design of a VS, multiple structures of the same protein target may exist, and not all will give equal results in a structure-based strategy. This is the case of the SARS-CoV-2 main protease (M-pro) (Macip et al., 2021). 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 1500 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 making a selection and beginning the VS process (Macip et al., 2022). Moreover, not all structures are appropriate for a particular purpose and some give better results than others for protein-ligand docking (Llop-Peiró et al., 2024). Additionally, drug-discovery protocols depend on whether the ligands are covalently or non-covalently bound to the protein (Macip et al., 2022).

For all these reasons (i.e., to develop better scoring functions and select the best target structure for a drug-discovery protocol), it is important to distinguish between PDB structures with ligands that are covalently or non-covalently bound. The PDB lacks an option in its advanced filter to distinguish between free proteins (i.e., the apo-form) and protein–ligand complexes (i.e., holo-form), or between ligand–protein complexes that are non-covalently or covalently bound. However, identifying covalent complexes is not a trivial task. There are no tools that perform this identification automatically. Therefore, we have developed PDB-CAT, a tool that can automatically classify PDB structures in terms of whether they are in their apo-form or if their ligands are bound covalently or non-covalently.

The PDBx/mmCIF format is now the standard PDB archive distribution format and it does not have the limitations of the older PDB file format (RCSB PDB, 2024b). PDB-CAT exclusively works with this modern format and employs the PDBeCIF library (Van Ginkel et al., 2021), which allows users to navigate and extract data from CIF files efficiently. In addition to classifying entries based on the ligand, PDB-CAT also extracts information about all entities in a PDB entry and can verify if there are any mutations in the protein sequence by comparing it to a FASTA reference file. This option is particularly useful for SARS-CoV-2 M-pro, as there are multiple

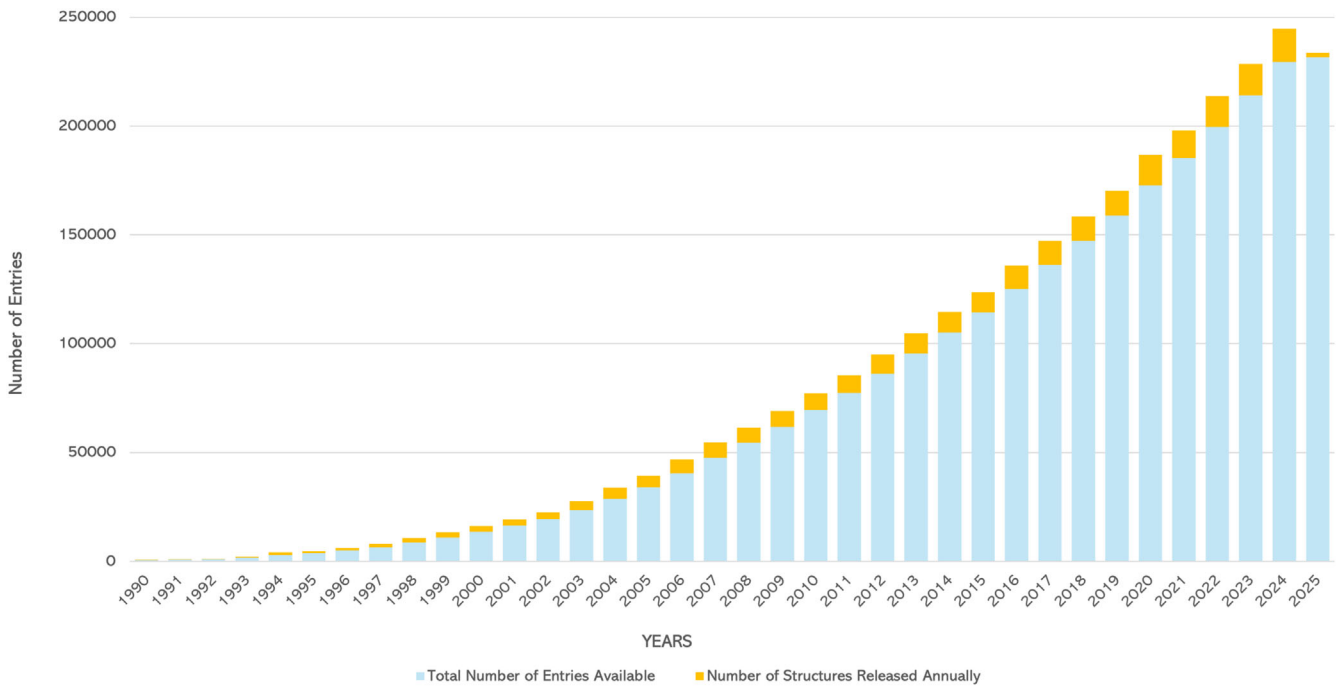

FIGURE 1 PDB statistics: Exponential growth of structures deposited in the PDB. In blue, the total number of entries available; in orange, the number of structures released each year (RCSB PDB, 2024a).

variants of the SARS-CoV-2 virus (Saldivar-Espinoza et al., 2023).

2 | 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 classifies each entity into one of three groups: polymer, non-polymer or branched.

PDB-CAT is designed to work with PDB files that contain a single model. Since NMR structures consist of multiple models, they are discarded, as the calculation of interatomic distances is not straightforward when several conformations are present. These distances are essential for classifying ligands as covalently or non-covalently bound. If an NMR structure is provided, the program will skip it and print a message "Warning. NMR structure {name} is not analyzed". The same applies to occupancy: occupancy levels are checked, and only atoms with 1.00 occupancy are considered for covalent bond detection. In the CSV output, covalent bonds are reported together with the occupancy values of the atoms involved, for example, SG (CYS29) (Occup.: 1.00) – C23 (H9B.) (Occup.: 1.00) – (1.75 Å).

2.1 | Protein and ligand identification

To identify the main protein and its subunits from an mmCIF file, they are always defined as polypeptide polymers, and it has been set that the polypeptide should consist of more than 20 residues (Figure 2). Polymer information is located in the entity and entity_poly categories of the PDBx/mmCIF format. Figure 2 summarizes the steps PDB-CAT takes to identify and classify PDB entries based on their bounded ligands. Alternatively, structures without a ligand are classified as apo.

The PDB categorizes small molecules such as ions, cofactors, inhibitors, and drugs as non-polymer entities. However, identifying polymeric entities, such as peptides or saccharides, as ligands is by no means straightforward. After the protein has been identified, PDB-CAT classifies any other polypeptide polymer entities in the structure as peptide ligands, using a length threshold that is set to 20 residues by default but can be modified by the user (Figure 2). If the entity is longer than the threshold, then it will be classified as another chain or subunit. The Biologically Interesting Molecule Reference Dictionary (BIRD) dictionary (https://www.wwpdb.org/data/bird), contains information on peptide-like inhibitors and common oligosaccharides. Some of the mmCIF files contain these BIRD

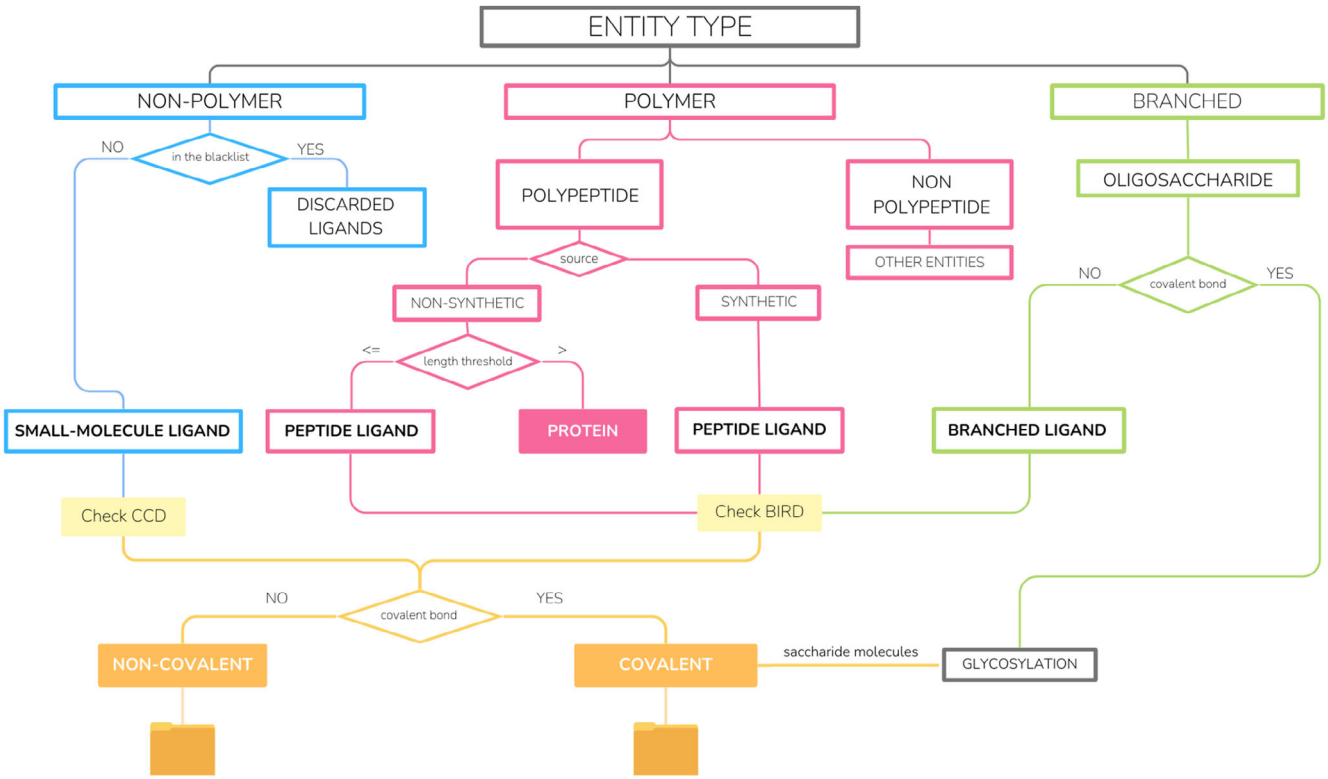

FIGURE 2 Flowchart illustrating the steps taken by PDB-CAT to identify and classify ligands. The process includes ligand detection, and further categorization based on the presence or absence of a covalent bond.

IDs, so PDB-CAT checks for BIRD IDs to retrieve more information about the ligands.

The next entity type is non-polymer and typically refers to small molecules. PDB-CAT considers a non-polymer entity as a ligand if it is not on a blacklist (Figure 2). The blacklist is a text list that contains the most common solvents, ions, and co-factors and can be modified by the user based on the target being analyzed. To remove an element from the blacklist just comment the line by writing the "#" symbol at the beginning. For example, a co-factor bound to a viral protease might be discarded by some computational chemists, while in other cases it might be important to consider. If there is a match with any element on the blacklist, the small molecule is then added to the list of discarded ligands (Figure 2). PDB-CAT also verifies the Chemical Component Dictionary (CCD) ID (https://www.wwpdb.org/data/ccd), which labels small molecule components, to gather additional information about the ligands, in the same way that it uses the BIRD ID.

The last entity type in the PDBx/mmCIF format is branched, which is what oligosaccharides are usually classified as. For these entities, PDB-CAT determines if they are bonded covalently to the protein. When an oligosaccharide forms a covalent bond with the protein, it is classified as glycosylation. If it does not, it is classified as a saccharide ligand (Figure 2).

2.2 | Covalently or non-covalently bonded ligands

Ligands are classified as covalently or non-covalently bonded depending on whether there is a covalent bond between the ligand and the protein.

To retrieve ligand information the program screens the CIF file annotations, specifically the _struct_conn field, to identify entries where the conn_type_id variable contains "covale". It then verifies whether the atoms of the covalent bond belong to the protein and a ligand entity. However, the _struct_conn field is not required for depositing a CIF file in the PDB. For that reason, the program identifies potential covalent ligands by checking heavy atom distances between the protein and the ligand (hydrogen atoms are ignored). By default, a single bond is considered covalent if a protein atom is within 1.95 Å of a ligand atom. This threshold was established by analyzing 1035 covalent complexes from PDBbindv2020 (Figure 3). The covalent_distance parameter is an argument of the write_output() function, set by default to 1.95 Å. Users can customize this threshold by providing a different value when calling the function in the notebook, for example, covalent_distance = 2.00, allowing flexibility in defining covalent bond criteria based on specific analysis requirements. A ligand is classified as covalent if at

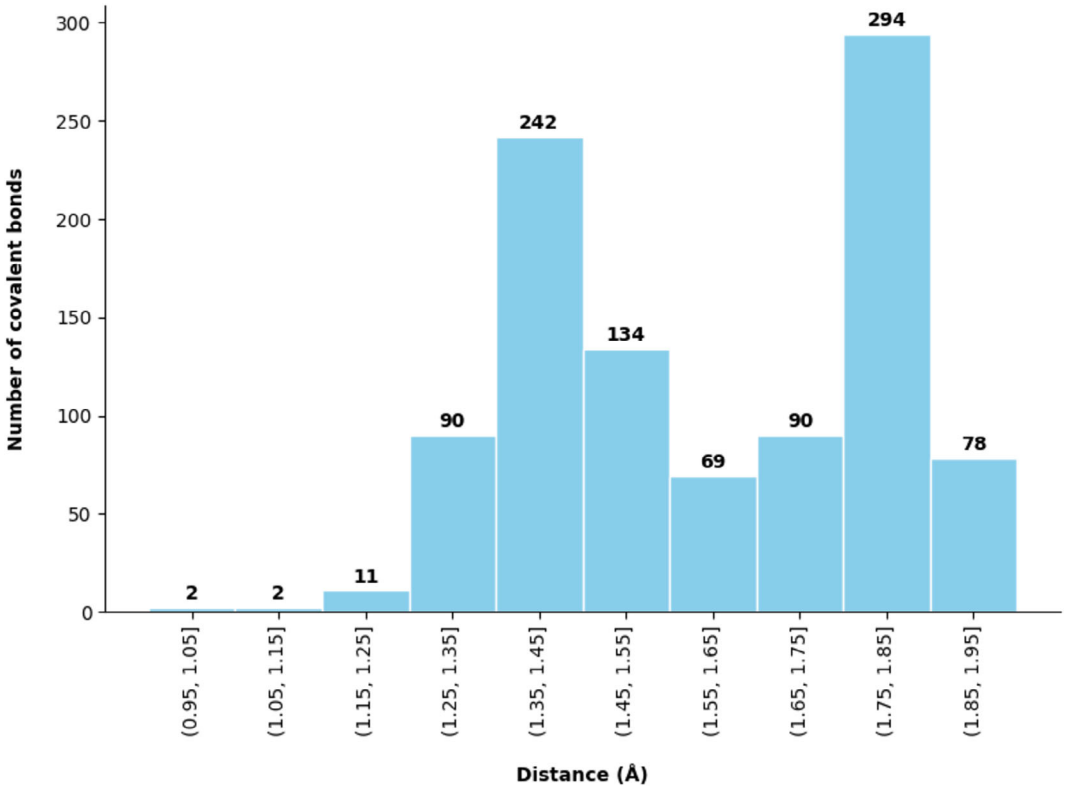

FIGURE 3 Covalent distance distribution in PDBbindv2020 for implicit covalent complexes. 95% of the covalent bonds are detected within 1.95 Å distance.

least one covalent bond is present; otherwise, it is categorized as non-covalent (Figure 2). The program can detect single, double, and triple bonds because the distances of the latter two are shorter and fall within the threshold.

This implementation improves classification accuracy, as CIF file annotations alone do not fully determine covalent bonding. For example, in PDBbindv2020, 20% of covalent complexes are not annotated as covalent but can be identified using coordinate-based calculations.

2.3 | Mutation analysis

In addition to classifying a set of PDB complexes, the PDB-CAT algorithm can also identify mutations in the residues of the protein. This can be defined in the Boolean variable: mutation. The mutation analysis compares the sequence of the protein entities to a FASTA file that contains one or more reference sequences. The algorithm selects the most similar sequence for each entry to perform the alignment. The CSV output includes the FASTA sequence ID selected for alignment for each entry and information about mutations, residue locations, percentage of identity, and gaps obtained with the Pairwise Alignment module of the Biopython library (Cock et al., 2009). This analysis of mutations enables the rapid identification of different versions of the same protein and makes it possible to distinguish between sequences that are identical and those that contain either natural or artificial mutations (e.g., those introduced to analyze the role of a specific residue or for other experimental purposes).

2.4 | 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 one PDB ID and provides a comprehensive set of information about the entry. This information includes details about the protein, such as the title of the PDB file, a description, the number of subunits, the subunit IDs (referred to as chains), and the number of residues for each subunit. It also indicates whether the protein is part of a complex. The CSV presents information about ligands, including their name, type, function, and whether they form a covalent bond with the protein. In this section, glycosylation sites are distinguished and differentiated from other covalent ligand bonds. The CSV also gives information about discarded ligands (any element from the blacklist that is bonded to the protein). The final columns cover mutation information: that is to say, the ID of the FASTA sequence, the number of mutations, their locations, identity percentages, and any gaps in the sequence.

In the second CSV file, each line corresponds to an entity bonded to a protein. The format is straightforward, with information about 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, when the bonded molecule corresponds to a covalently attached glycan or sugar moiety, PDB-CAT labels it as "glycosylation" to prevent misclassification as a covalent ligand.

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 (Figure 4).

3 | IMPLEMENTATION

3.1 | 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.

3.2 | Parallelization

PDB-CAT has been designed to be run in parallel. Our implementation of parallelization optimizes computational resource utilization through a structured approach. A real-time monitoring function, leveraging psutil, tracks RAM and CPU usage to prevent resource saturation. The system dynamically detects available cores using multiprocessing, enabling users to allocate a fraction of them (cpu_use) for parallel execution. For instance, on an 8-core system, setting cpu_use = 0.5 results in the creation of four parallel processes, balancing workload distribution while avoiding hardware overload. The cpu_use parameter is an argument of the write_output() function, which is set by default to cpu_use = 1. This value can be customized by providing a different input when calling the function in the notebook, allowing users to adjust resource allocation based on system constraints and analysis requirements.

To ensure efficient execution, files are evenly distributed among processes, which run independently before consolidating results. The threading library allows monitoring to operate separately from the main execution, preventing performance interference. Additionally, concurrent.futures (ProcessPoolExecutor) manages process distribution dynamically, optimizing parallel execution without requiring manual core

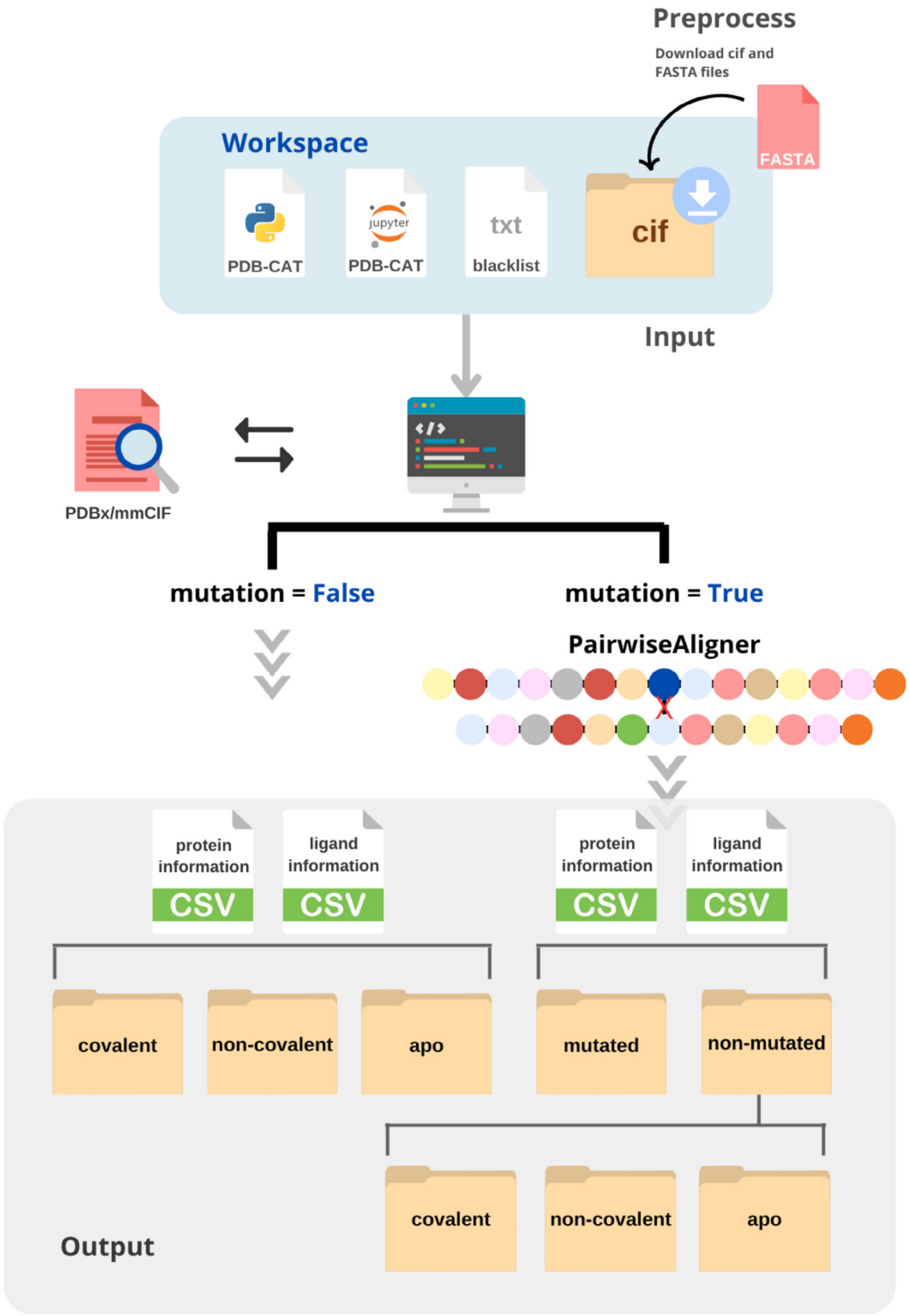

FIGURE 4 Workflow diagram of the PDB-CAT: In the workspace, you will find a Jupyter Notebook file, a Python module containing some functions, the blacklist file, and the directory containing the input files in mmCIF format. To execute the mutation mode, the reference file should be part of the dataset. The program generates two CSV files containing all the relevant information, as well as several folders.

assignment. This approach minimizes overhead and enhances computational efficiency.

3.3 | How to use

A complete tutorial is available in Gitbook (https://ariadnallopps-organization.gitbook.io/pdb-cat). The first step before running the code is to establish the dataset of structures. This involves downloading the structure files locally from the PDB. Given the ongoing transition of PDB to PDBx/mmCIF format, it is essential that the input files are in this CIF format.

The PDB-CAT program can be executed by a Jupyter Notebook. This notebook contains a cell for code customization so that interaction with the code is clear and concise. Four of the variables in the main code can be modified; explanations are provided on how to do so in the Gitbook tutorial. Note that the code can be run in Google Colab.

3.4 | Requirements

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

4 | RESULTS

The following two examples illustrate different applications of PDB-CAT: (i) a full screening of PDBbindv2020, comprising over 19,000 structures, where the program's parallelization capabilities help reduce execution time; and (ii) a classification of SARS-CoV-2 Main Protease complexes by variant, using a reference sequence to identify mutations in the protein sequences.

4.1 | Full screening of PDBbindv2020

A dataset of protein–ligand complexes was extracted from the refined set of PDBbindv2020 (Wang et al., 2004). The PDBbindv2020 database (https://www.pdbbind-plus.org.cn/) is the largest open-source collection of protein–ligand complexes, providing information on both binding affinities and known 3D crystal structures. Updated annually, the 2020 version comprises 19,443 protein–ligand complexes and experimentally measured binding affinity data. PDB-CAT efficiently analyzed this dataset in under 10 min. The program was run locally with Jupyter Notebook and used 10 GB of RAM and 15% of the 32 CPU nodes. A total of 303 NMR structures were discarded. Of the remaining 19,139 structures, 97% were classified as protein–ligand complexes, while the remaining 3% were identified as apo forms. This is because these entries contained a blacklist component, which had been labeled as a ligand in the PDBbindv2020 dataset. Among the 19,139 structures, 1035 were identified as covalent complexes. Notably, 265 of these complexes (20%) were detected through the distance-based analysis, despite not being annotated as covalent complexes in the original PDB entries. Note that the mutation filter was not used in this validation because of the wide range of proteins found in the dataset. The CSV of the PDBbindv2020 is included in the example folder of the repository.

4.2 | SARS-CoV-2 Main protease

In this example, 1550 PDB structures available in February 2025 and containing the SARS-CoV-2 M-pro were analyzed. The corresponding PDBx/mmCIF files were subject to a thorough analysis, including mutation categorization. A FASTA file containing the sequence of the first crystallized M-pro structure (PDB ID: 6LU7) and that of the M-pro Omicron variant (B.1.1.529) (Sacco et al., 2022) (PDB ID: 7TOB) was used as reference.

Out of the 1550 M-pro structures, 1268 had sequences identical to 6LU7 and 20 matched 7TOB (Omicron variant), so PDB-CAT classified these 1288 structures as non-mutated. The remaining 262 structures contained either substitutions at catalytic residues (e.g., C145A) or mutations such as T21I, L50F, E166V, and A173V, which have been observed experimentally in viral passages and may contribute to resistance against M-pro inhibitors. The non-mutated set was further classified into 112 apo structures, 322 covalent complexes, and 854 non-covalent complexes. For each structure, information about the exact mutated residues (if any), sequence identity percentage, and the number of gaps compared to the reference sequences was extracted. The CSV file also contains all the crucial information, and it is available in the example folder.

5 | CONCLUSIONS

PDB-CAT is a unique tool for classifying PDB structures into apo forms, covalent complexes, and non-covalent complexes as well as for detecting covalent bonds when they are not explicitly described in the PDB file. PDB-CAT also allows the comparison of multiple variants or mutations of a protein. It is also a valuable resource for researchers managing the vast

amount of data from the Protein Data Bank, especially for computational chemists who deal with multiple structures of the same protein or who want to correlate binding affinity data with the 3D structure of the corresponding complex.

AUTHOR CONTRIBUTIONS

Ariadna Llop-Peiró: Investigation; data curation; software; methodology; writing – original draft; validation. **Said Trujillo-De León:** Software; methodology. **Gerard Pujadas:** Supervision; conceptualization; writing – review and editing. **Santiago Garcia-Vallvé:** Conceptualization; supervision; writing – review and editing. **Aleix Gimeno:** Project administration; supervision; conceptualization; writing – review and editing.

FUNDING INFORMATION

This study was supported by the project PID2022-138327OB-I00 financed by MCIN/AEI/10.13039/501100011033/FEDER, UE. Ariadna Llop-Peiró is the 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.

DATA AVAILABILITY STATEMENT

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/. A tutorial on how to use the software can be found at: 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 online in the source repository: https://github.com/URV-cheminformatics/PDB-CAT/example.

ORCID

Ariadna Llop-Peiró https://orcid.org/0009-0006-9785-3852
Gerard Pujadas https://orcid.org/0000-0003-2598-8089
Santiago Garcia-Vallvé https://orcid.org/0000-0002-0348-7497
Aleix Gimeno https://orcid.org/0000-0002-7654-6689

REFERENCES

Adamson CS, Chibale K, Goss RJM, Jaspars M, Newman DJ, Dorrington RA. Antiviral drug discovery: preparing for the next pandemic. Chemical Society Reviews. 2021;50(6):3647–55. https://doi.org/10.1039/d0cs01118e

Burley SK, Bhikadiya C, Bi C, Bittrich S, Chao H, Chen L, et al. Delivery of experimentally-determined PDB structures alongside one million computed structure models of proteins from artificial intelligence/machine learning. Nucleic Acids Research. 2023;51(D1):D488–508. https://doi.org/10.1093/nar/gkac1077

Cock PJ, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. Bioinformatics (Oxford, England). 2009;25(11):1422–3. https://doi.org/10.1093/bioinformatics/btp163

Gimeno A, Ojeda-Montes MJ, Tomás-Hernández S, Cereto-Massagué A, Beltrán-Debón R, Mulero M, et al. The light and dark sides of virtual screening: what is there to know? International Journal of Molecular Sciences. 2019;20(6):1375. https://doi.org/10.3390/ijms20061375

Kumalo HM, Bhakat S, Soliman ME. Theory and applications of covalent docking in drug discovery: merits and pitfalls. Molecules (Basel, Switzerland). 2015;20(2):1984–2000. https://doi.org/10.3390/molecules20021984

Llop-Peiró A, Macip G, Garcia-Vallvé S, Pujadas G. Are protein-ligand docking programs good enough to predict experimental poses of noncovalent ligands bound to the SARS-CoV-2 main protease? Drug Discovery Today. 2024;29(10):104137. https://doi.org/10.1016/j.drudis.2024.104137

Macip G, Garcia-Segura P, Mestres-Truyol J, Saldivar-Espinoza B, Ojeda-Montes MJ, Gimeno A, et al. Haste makes waste: a critical review of docking-based virtual screening in drug repurposing for SARS-CoV-2 main protease (M-pro) inhibition. Medicinal Research Reviews. 2022;42(2):744–69. https://doi.org/10.1002/med.21862

Macip G, Garcia-Segura P, Mestres-Truyol J, Saldivar-Espinoza B, Pujadas G, Garcia-Vallvé S. A review of the current landscape of SARS-CoV-2 Main protease inhibitors: have we hit the Bullseye yet? International Journal of Molecular Sciences. 2021;23(1):259. https://doi.org/10.3390/ijms23010259

Paggi JM, Pandit A, Dror RO. The art and science of molecular docking. Annual Review of Biochemistry. 2024;93(1):389–410. https://doi.org/10.1146/annurev-biochem-030222-120000

RCSB PDB. PDB statistics: Overall growth of released structures per year. The RCSB Protein Data Bank. 2024a https://www.rcsb.org/stats/growth/growth-released-structures

RCSB PDB. Beginner's guide to PDB structures and the PDBx/mmCIF format. 2024b https://pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/beginner%E2%80%99s-guide-to-pdbx-mmcif

Sacco MD, Hu Y, Gongora MV, Meilleur F, Kemp MT, Zhang X, et al. The P132H mutation in the main protease of omicron SARS-CoV-2 decreases thermal stability without compromising catalysis or small-molecule drug inhibition. Cell Research. 2022;32(5):498–500. https://doi.org/10.1038/s41422-022-00640-y

Saldivar-Espinoza B, Garcia-Segura P, Novau-Ferré N, Macip G, Martínez R, Puigbò P, et al. The mutational landscape of SARS-CoV-2. International Journal of Molecular Sciences. 2023;24(10):9072. https://doi.org/10.3390/ijms24109072

van Ginkel G, Pravda L, Dana JM, Varadi M, Keller P, Anyango S, et al. PDBeCIF: an open-source mmCIF/CIF parsing and processing package. BMC Bioinformatics. 2021;22:383. https://doi.org/10.1186/s12859-021-04271-9

Vázquez J, López M, Gibert E, Herrero E, Luque FJ. Merging ligand-based and structure-based methods in drug discovery: an overview of combined virtual screening approaches. Molecules. 2020;25(20):4723. https://doi.org/10.3390/molecules25204723

Wang R, Fang X, Lu Y, Wang S. The PDBbind database: collection of binding affinities for protein-ligand complexes with known three-dimensional structures. Journal of Medicinal Chemistry. 2004;47(12):2977–80. https://doi.org/10.1021/jm030580l

Wang Y, Jiao Q, Wang J, Cai X, Zhao W, Cui X. Prediction of protein-ligand binding affinity with deep learning. Computational and Structural Biotechnology Journal. 2023;21:5796–806. https://doi.org/10.1016/j.csbj.2023.11.009

Yang C, Chen EA, Zhang Y. Protein-ligand docking in the machine-learning era. Molecules (Basel, Switzerland). 2022;27(14):4568. https://doi.org/10.3390/molecules27144568

How to cite this article: Llop-Peiró A, Trujillo-De León S, Pujadas G, Garcia-Vallvé S, Gimeno A. PDB-CAT: A user-friendly tool to classify and analyze PDB protein–ligand complexes. Protein Science. 2025;34(12):e70379. https://doi.org/10.1002/pro.70379