doi.org/10.26434/chemrxiv.12806819.v3

CAVIAR: a method for automatic cavity detection, description and decomposition into subcavities

Jean-Rémy Marchand, Bernard Pirard, Peter Ertl, Finton Sirockin

Submitted date: 07/05/2021 • Posted date: 10/05/2021

Citation information: Marchand, Jean-Rémy; Pirard, Bernard; Ertl, Peter; Sirockin, Finton (2020): CAVIAR: a method for automatic cavity detection, description and decomposition into subcavities. ChemRxiv. Preprint. https://doi.org/10.26434/chemrxiv.12806819.v3

The accurate description of protein binding sites is essential to the determination of similarity and the application of machine learning methods to relate the binding sites to observed functions. This work describes CAVIAR, a new open source tool for generating descriptors for binding sites, using protein structures in PDB and mmCIF format as well as trajectory frames from molecular dynamics simulations as input. The applicability of CAVIAR descriptors is showcased by computing machine learning predictions of binding site ligandability. The method can also automatically assign subcavities, even in the absence of a bound ligand. The defined subpockets mimic the empirical definitions used in medicinal chemistry projects. It is shown that the experimental binding affinity scales relatively well with the number of subcavities filled by the ligand, with compounds binding to more than three subcavities having nanomolar or better affinities to the target. The CAVIAR descriptors and methods can be used in any machine learning-based investigations of problems involving binding sites, from protein engineering to hit identification. The full software code is available on GitHub and a conda package is hosted on Anaconda cloud.

CAVIAR: a method for automatic cavity detection, description and decomposition into subcavities

Jean-Rémy Marchand, Bernard Pirard, Peter Ertl, Finton Sirockin**

Novartis Institutes for Biomedical Research, Fabrikstrasse 16, 4056 Basel, Switzerland

Contact: jean-remy.marchand@novartis.com; finton.sirockin@novartis.com

ORCID
Jean-Remy Marchand: 0000-0002-8002-9457
Bernard Pirard: 0000-0003-0702-0955
Peter Ertl: 0000-0001-6496-4448
Finton Sirockin: 0000-0003-2536-7485

ABSTRACT

The accurate description of protein binding sites is essential to the determination of similarity and the application of machine learning methods to relate the binding sites to observed functions. This work describes CAVIAR, a new open source tool for generating descriptors for binding sites, using protein structures in PDB and mmCIF format as well as trajectory frames from molecular dynamics simulations as input. The applicability of CAVIAR descriptors is showcased by computing machine learning predictions of binding site ligandability. The method can also automatically assign subcavities, even in the absence of a bound ligand. The defined subpockets mimic the empirical definitions used in medicinal chemistry projects. It is shown that the experimental binding affinity scales relatively well with the number of subcavities filled by the ligand, with compounds binding to more than three subcavities having nanomolar or better affinities to the target. The CAVIAR descriptors and methods can be used in any machine learning-based investigations of problems involving binding sites, from protein engineering to hit identification. The full software code is available on GitHub and a conda package is hosted on Anaconda cloud.

KEYWORDS

Binding pocket; cavity descriptors; subcavities; subpocket; ligandability; fragment-based drug design

DECLARATIONS

Funding. This work was supported by the postdoctoral office of the Novartis Institutes for Biomedical Research.

Conflicts of interest. We declare no conflict of interest.

Availability of data and material. All data are publicly available on GitHub.

Code availability. Installation notes, user manual and support for CAVIAR are available at https://jr-marchand.github.io/caviar/. The GitHub repository hosts the CAVIAR source code and validation sets at https://github.com/jr-marchand/caviar. A conda package is hosted on Anaconda cloud at https://anaconda.org/jr-marchand/caviar. Source code and data available under a MIT license.

Authors' contributions. The study was designed by all authors. J.R.M. wrote the software and performed the analysis. J.R.M. and F.S. analyzed the results. The manuscript was written by J.R.M. and F.S.. All authors have given approval to the final version of the manuscript.

INTRODUCTION

The PDB hosts more than 150,000 experimentally determined structures of macromolecules. Drug targets are particularly well represented in this dataset, with 88% of the targets of new molecular entities approved by the US food and drug administration in the period 2010-2016 being publicly and freely accessible in the PDB at date of approval [1]. This great wealth of data provides fantastic opportunities to extract meaningful information for drug design efforts. Protein cavities are at the basis of the functions of folded proteins, from enzymatic activity to binding of endogenous molecules and signal transduction. Small sets of binding pockets can be characterized manually by analyzing holo structures of protein-ligand complexes. However, the analysis of bigger datasets and the application of machine learning (ML) methods, requires automatic algorithms to describe binding sites. The cavity detection field has been prolific in the last three decades [2–4]. Successful applications include the prediction of target ligandability [5–9], identification of off-targets [10–14], functional annotation [15–18] and ligand design and drug repurposing [19–22].

Structure-based cavity detection methods can be grouped into two general families: energy-based algorithms and geometry-based [2, 23, 24]. Energy-based methods rely on the calculation of the interaction energy between chemical or pseudo-chemical probes and the surface of proteins. They can produce very valuable information about hot spots for intermolecular interactions for medicinal chemistry, but may require a careful preparation of the protein (*e.g.*, typing and protonation) and are inherently computationally intensive [25–30]. Geometry-based methods are less resource demanding and potentially more resilient to small changes in the pocket through side-chain positioning, which gives them a different scope as they can be applied on large scales and easily automated for the integration in workflows. Cavities are detected based on their shapes and are sometimes augmented with other properties, *e.g.*, buriedness, pharmacophores, or conservation of certain residues overrepresented in binding pockets [10, 31–34]. A variety of geometry-based methods for pocket detection have been developed, *i.e.*, algorithms relying on (1) enclosure of grid points around the protein, (2) space filling, (3) Voronoi diagram, and (4) imaging science (Table 1). Consensus methods combining results from more than one method have also been described [35, 36]. Recent versions of these software perform generally well on validation datasets, with a reported ability to detect the correct ligand binding pocket in their top three scoring cavities being around 80 to 90% [4].

Table 1. Common methods for geometry-based cavity detection.

Method	*Core principle*	*Representative examples*
Enclosure of grid points	The enclosure of grid points around the protein, *i.e.*, how many close contacts with protein atoms, defines potential cavities	POCKET [37], LIGSITE [38], PocketDepth[a, d] [39], PocketPicker [23], McVol[a] [40], VICE [41], VolSite [9], SiteMap[b, c] [7]
Space filling	Spheres are placed around the protein surface to detect empty spaces in the protein convex hull	SURFNET [42], PASS [43], PHECOM[a] [44], KVFinder[a] [45], GHECOM[a, d] [46], SCREEN [8], POCASA[d] [47]

Voronoi diagram	The Voronoi decomposition of the space of protein atoms serves as basis to identify clefts	FindSurf [48], CAST[c, d] [49], APROPOS [50], Fpocket[a, b, c, d] [6], SiteFinder[c] (MOE)
Imaging science	Gaussian surfaces approximate the protein shape	DoGSite[d] [24], CavVis[a] [51]

[a]Open-source software available at the time of the study.
[b]Software permitting the analysis of molecular dynamics trajectories.
[c]mmCIF format ready software.
[d]Webserver.

Cavity segmentation into subcavities is crucial in structure-based drug design, to help medicinal chemists to optimize properties like potency and selectivity [29, 52–55]. Similar proteins may have binding pockets with different subcavities and dissimilar proteins may have conserved subcavities. Many of the largest drug target classes exhibit geometrically well-defined subpockets, such as proteases, kinases and GPCRs, which are used extensively in order to develop selective compounds [53]. Two independent studies concluded that drug-like ligands typically occupy about a third of the volume of the whole binding pocket, filling only some of the subpockets [56, 57]. Efforts have been made to try to characterize the chemical fragment preference of certain residues [58, 59] and link the fragment chemical space to binding pocket microenvironments [60–63]. These methods extract and store information of fragmented ligands from the PDB and their interactions with surrounding amino acids. However, they lack a clear protein-centric definition of the subcavities and circumvent it by running queries on empirical ligand- or coordinate-based definition of subpockets. One potential exception is DoGSite, which was developed as a ligand-agnostic cavity identification tool, borrowing concepts from the computational image recognition field [24]. Briefly, DoGSite uses a difference of Gaussians algorithm to identify hotspots, inflates them before merging into larger cavities. Primitive hotspots are smaller components of very large pockets and may be eventually called subpockets. However, the subsites defined by DoGSite do not reproduce the concept of subpockets used in medicinal chemistry, *i.e.*, small and localized parts of a cavity encompassing defined functional groups of a ligand. DoGSite hotspots circumvent pocket overspanning, with one single subpocket encompassing the ligand in 87% of the cases [24, 64]. Therefore, the automatic decomposition of (apo) binding pockets into medicinal chemistry-compatible subcavities is an unresolved matter.

This study describes CAVIAR, an open-source software in python for the generation of cavity descriptors. CAVIAR is freely available under the MIT license and can be easily adapted for use by many other application. Descriptors are exported as python objects that are compatible with widely used libraries such as scikit-learn [65], TensorFlow [66] and PyTorch [67]. The manuscript is structured as follows. First, we perform a brief review of some of the main published methodologies for cavity identification and subcavity decomposition. This is the foundation upon which the descriptors are built. Then, we describe the details of the cavity detection and segmentation algorithms of CAVIAR. Our results section benchmark CAVIAR against other standard methods, showing that CAVIAR has best in class performance for cavity identification on standard datasets and can handle major cavity overspanning issues without losing performance. The quality of the subpocket segmentation algorithm is discussed based on representative examples. After that, we examine the performance of key global descriptors for liganded and apo

pockets at the PDB level; and demonstrate the relationship between the number of subcavities filled by ligands and the binding affinities to their target. We showcase an application of CAVIAR descriptors for generating ML models for ligandability prediction. Finally, limitations of the method are critically discussed.

MATERIAL AND METHODS

Cavity identification

CAVIAR supports input in PDB and mmCIF format, as well as DCD molecular dynamics trajectory files. The molecular dynamics frames analysis routines contain an orientation-invariant clustering of pockets to determine the occupancy of cavities within a trajectory and identify the representative structures (details in Supplementary Material 1). The cavity detection algorithm is inspired by well-established concepts of enclosure of grid points algorithms and augmented with novel ideas to refine the resulting cavities, *e.g.*, double pass to estimate buriedness, intense trimming of spurious points and exclusion of loosely connected nodes. Protein atoms are enclosed in a cubic grid, with a spacing of 1.0 Å and a margin of 2.0 Å around the minimum and maximum coordinates on each axis. Grid points further than 6.0 Å from protein atoms are filtered out for computational efficiency. Grid points within the protein surface, *i.e.*, within 1.0 Å of the van der Waals envelope of an atom, are assigned as a protein type. The remaining grid points are considered as solvent grid points and investigated further (Fig. 1a). For each solvent grid point, fourteen cubic directions, (three axes and the four cubic diagonals, all in both positive and negative directions), are investigated for contacts with protein grid points. Exploring each direction, a counter is incremented if a protein grid point is encountered within four grid spacings, (4 Å for the three axis and 6.9 Å for the cubic diagonals, assuming a grid spacing of 1 Å). The final counter value for a grid point has a value between 0 and 14, and represents the "buriedness" of a solvent grid point (Fig. 1b). Grid points with a buriedness of 8 or above are considered as putative cavity grid points, and grid points with a buriedness of 7 or less are investigated in a second pass. The second pass is similar to the first one, except that only solvent grid points that are in vicinity (three grid spacings) of the previously defined (putative) cavity grid points are investigated. Solvent grid points with at least 8 contacts with putative grid points are added to the set of putative cavity grid points. This second pass is necessary to include points that are in the middle of large cavities and may be missed by the first pass, which would otherwise create voids in large cavities (Fig. 1c). One risk associated with grid based algorithms is cavity overspanning, *i.e.*, very large cavities can be created by expanding the site over the surface of the protein: these cavities have cavity grid points connecting cavities that should not be joined (Fig. 1d). To address this, we developed a metric to estimate how a grid point is surrounded by its peers within its cavity ensemble. The number of surrounding cavity grid points within 2 grid spacings ($N_{neighbors}$, max. = 124) and their average buriedness (B_{avg}, max. = 14) is used to calculate a "trim score" ($score_{trim}$, equation 1). The trim score measures how mingled a cavity grid point is in a set of cavity grid points. Points with a trim score below 500 are removed(Fig. 1e-f).

$$score_{trim} = N_{neighbors} * 10^{B_{avg}/10} \quad \text{(Eq. 1)}$$

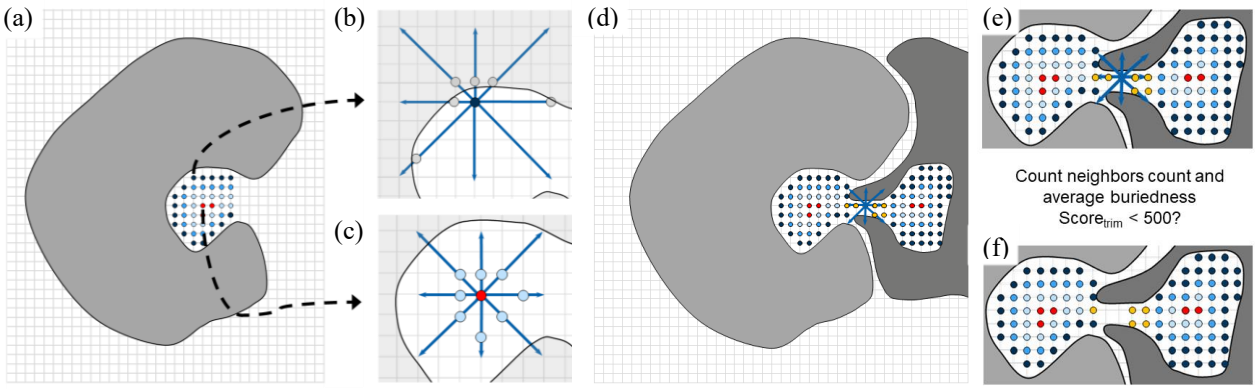


Fig. 1. Visual depiction of the grid-based cavity identification algorithm. (a) The protein, represented as a gray shape, is embedded in a regular 3D grid. (b) The number of contacts between grid points outside of the protein surface and grid points inside the protein surface is investigated; this defines putative cavity grid points. (c) A second pass detects grid points surrounded by putative grid points that would have been missed in B. (d) Cavity grid points connect a cavity in the light gray protein chain and another one in the dark gray chain. (e) Zoom into the cavity grid points of panel d. Grid points are colored according to a color gradient representing buriedness. Dark blue points are more buried, light blue less buried. Red points symbolize grid points added by the second pass and yellow points connect the two cavities and could be trimmed out. For each cavity point, the count of neighbors and the average buriedness are measured to calculate the trim score. (f) Grid points with a trim score below 500 are eliminated from the cavity grid points set. Two yellow grid points are discarded, and the cavity is split into two cavities.

Putative grid points are embedded in a graph data structure, where edges are built around adjacent grid points (nodes) in the cube. Bridges and self-loops are filtered out, as well as nodes with a degree of three or less. At this stage, clusters of more than 40 grid points are identified as cavities. Cavity grid points are assigned pseudotype descriptors according to the pharmacophore type of the closest protein atom: aliphatic, aromatic, hydrogen bond donor/acceptor, negative (charged group of Asp/Glu), positive (charged group of Lys/Arg), and other specific types (S atom of Cys, ring of His, metal ion). Global cavity descriptors are calculated and stored: hydrophobicity (proportion of grid points of type aliphatic and aromatic), cavity score (equation 2), median buriedness of cavity points, cavity size in grid points, presence of a ligand, list of cavity residues and if the cavity has missing atoms, alternate locations or is between different protein chains. By default, cavities with missing residues or belonging to the 8th quantile of buriedness of 10 or less are excluded. This additional filtering step is performed to avoid generating noise from spurious cavities based on missing atoms, or cavities unlikely to be binding pockets because of high solvent exposure. Finally, cavities are ranked according to the cavity score (score$_{cavity}$, equation 2) and exported as a PDB file.

$$score_{cavity} = \frac{size * median * q}{100} \qquad \text{(Eq. 2)}$$

where *size* is the size of the cavity in grid points, *median* is the median buriedness and q is the 8th quantile of buriedness.

Calculations are performed with NumPy (1.17.3) and SciPy (1.4.1). Graph operations use the NetworkX (2.4) library. A total of 190,080 combinations of parameters was scanned to optimize cavity detection and to avoid overspanning. Details can be found in the Supplementary Material items S1 and S2. Descriptors are accessible directly from the CAVIAR library or can be exported as Python pickled format and include the following global cavity descriptors: size, statistics of distribution of pharmacophores and buriedness (average, median, quantiles), $score_{cavity}$, list of residues, hydrophobicity and count of subcavities. Local descriptors that cover the information of all grid points either as a NetworkX graph or as 3D image are also available: buriedness, pharmacophore, local asphericity, $score_{trim}$ and subcavity affiliation.

Validation sets for cavity identification

We assembled the following datasets: from the literature, Kahraman *et al* [57], Huang and Schroeder [33], the 198 drug-target set of MetaPocket [36], the DUD-e 102 targets [68]; from databases, scPDB [69] and PDBbind [70]; from our own compiled datasets, GPCR set and drugs set. The GPCR set contains 174 GPCR structures with drug-like ligands, including orthosteric and allosteric binders. The drugs set contains 540 drugs in PDB structures curated from the RCSB PDB drug mapping tool. The complete set of PDB files used for validation is available in the GitHub repository of the CAVIAR package (link in the code availability paragraph), dataset size and count of unique ligands per dataset are indicated in Table 2.

These datasets vary by size, from under 100 in the literature sets to more than 11,000 in scPDB database, as well as in their scope and composition. The use of multiple datasets is aimed at avoiding any bias arising in one dataset. There is a discrepancy between some of the numbers in the published cavity identification validation sets and our data. For example, the original "MetaPocket" dataset contains 198 drug targets, while there are 196 PDB entries in our "MetaPocket" validation set. Two of the structures in the original dataset have been removed from the distribution of released RCSB PDB entries. The absence of the specified ligand identifier in the PDB file, as well as duplicated PDB entries are two other reasons for count inconsistencies. Success in cavity identification is defined by the overlap between cavity points and ligand atoms within 1 Å. The comparison with other algorithms is performed on Huang and Schroeder's dataset [33]: success is defined by the presence of a ligand atom within 4 Å of the geometric center of the cavity, for direct comparison with the literature.

Subcavity decomposition

Either all available cavities, liganded cavities, or user-specified cavities can be investigated for subcavities decomposition. We borrowed concepts from computer image recognition for cavity segmentation [71]. First, the cavity grid points ensemble is converted into a 3D image, which is then remodeled with an Euclidean distance transform. Grid points are assigned values corresponding to their distance to the cavity surface. The points with the highest values are used as seeds for a watershed algorithm [71], which segments cavities into subgroups. Seed points are separated by at least 3 Å, to prevent over-segmentation. The watershed algorithm uses the values from the Euclidean distance at each cavity grid point as markers of local topography to flood basins starting from each seed until the different basins meet (Fig. 2a-d).

The watershed algorithm tends to over-segment images [71]. A careful definition of the seed points and topological values is necessary in order to obtain a reasonable separation of objects. We tried to balance the Euclidean distance transform values with the local pharmacophore information around each grid point (Shannon entropy of pharmacophore values), but it did not significantly change the results. As a consequence, we implemented steps to merge small "spurious" subcavities with their largest neighbor (Fig. 2e-h). The first step involves the detection of small subcavities (less than 50 grid points). Then, the number of direct contacts, *i.e.*, at 1 Å, between these small subcavities and other subcavities is calculated. If more than two thirds of the small subcavity grid points are in contact with neighboring subcavities, we merge it with its neighbor involved in the most contacts. Subcavities filling these two criteria are usually either extended and lying on top of another subcavity (Fig. 2e), or interstitial and disk-shaped between several subcavities (Fig. 2f). Image segmentation routines are performed with the scikit-image (0.16.2) library.

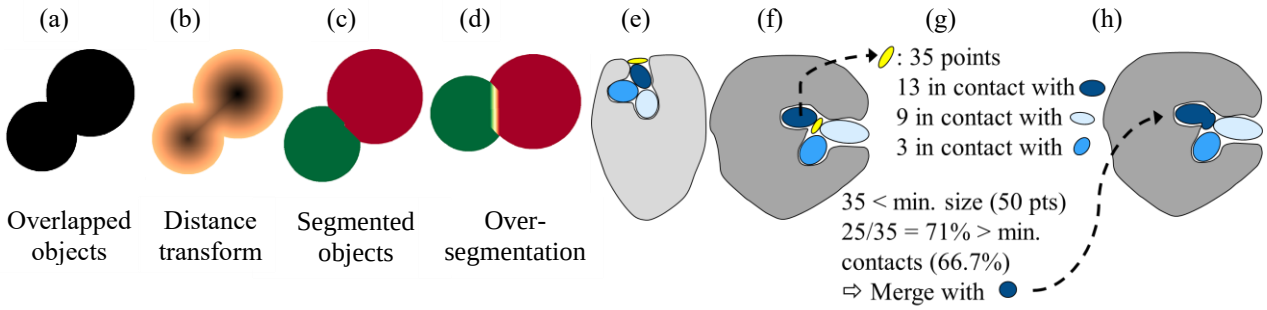


Fig. 2. 2D representation of the watershed algorithm. (a) Two overlapping circles, *e.g.*, a cavity that we could segment. (b) The local topography of the image is defined by an Euclidean distance transform of the original image. The darkest points are the most distant points to the image boundary. (c) Segmented image, with two objects, one in green and one in red, separated after applying the watershed algorithm. (d) An example obtained by moving the left object. In this case, an additional seed is defined in between the two object, and generates a spurious third segment in light yellow. (e) and (f) are two examples of cavity oversegmentation. In some cases, flat subcavities are created at the surface (e), and sometimes they are generated in between other main subcavities (f). (g) Summary of the criteria used to detect potentially spurious subcavities and identify the merging partner. (h) Result of (g) on (e). In both (d) and (e), the yellow subcavity is merged with the dark blue one.

To qualitatively assess the relevance of the subcavity decomposition tool, we assembled a carefully hand-picked dataset of 59 proteins (available in the GitHub repository of the CAVIAR package, link in the code availability paragraph) for which subcavities can be defined with a high level of confidence, based on experimental knowledge. This dataset contains 17 protease structures, which are a gold standard for proteins with binding pockets divided in precisely defined subpockets. In addition, we compiled 13 GPCR structures, 5 bromodomains, 5 kinases, 2 acetylcholine esterases, 3 ligases E3, and 14 other structures: FKBP, EGFR, Glucocorticoid receptor, TLR4, SMO, DOT1L, CYP51, SYI1, Acetylcholine receptor, HMGCoA reductase, tubulin alpha, NaK ATPase, Alpha amylase, and HSP90 alpha.

RESULTS AND DISCUSSION

Performance of CAVIAR for cavity identification

The definition of a cavity is a case-by-case subjective concept, which makes it difficult to extract meaningful statistics for the comparison of pocket identification algorithms. Success in cavity identification is defined as finding at least one ligand atom overlapping with cavity grid points. Table 2 shows a summary of results. CAVIAR successfully identifies almost all cavities in the large datasets, *e.g.*, reaching 99% of success on the 11,816 complexes of scPDB and 92% on the 4,227 cases of PDBBind. The performance is similarly high across all datasets except the MetaPocket set (81%). The MetaPocket dataset is enriched in very solvent-exposed ligand-protein complexes, with low curvature at the surface of the protein (*e.g.*, PDB codes 1pk2, 1gtb, 1lu1, 1q8m, 1sxk, 1tt6, 2c6g), which, by design, CAVIAR does not detect with default parameters (*cf.* limitations). Our validation datasets, especially the larger ones, contain a number of problematic PDB structures. More specifically, we noticed several cases of inappropriate ligand identifiers (*e.g.*, a cosolvent instead of the ligand-like compound) in the scPDB, PDBBind and MetaPocket datasets, which we corrected, but not exhaustively. The manual curation of all of these structures would be valuable, but is beyond the scope of this work. Interestingly, restricting the PDBBind dataset to high affinity complexes (micromolar affinity or better) results in a higher success rate for binding pocket identification (Table 2).

Table 2. General performance of CAVIAR on different datasets.

	n PDB	*n ligands*	*top 1*	*top 3*	*any*	*missed*
scPDB	**11,816**	**5,459**	**79%**	**94%**	**99%**	**1%**
PDBBind	**4,227**	**3,277**	**67%**	**84%**	**92%**	**8%**
PDBBind-HA*	**3,335**	**2,145**	**74%**	**90%**	**95%**	**5%**
Drugs	**554**	**257**	**67%**	**83%**	**96%**	**4%**
MetaPocket	**196**	**95**	**60%**	**76%**	**81%**	**19%**
GPCR	**174**	**123**	**89%**	**97%**	**99%**	**1%**
DUD-e	**102**	**102**	**83%**	**95%**	**96%**	**4%**
Kahraman	**98**	**12**	**77%**	**90%**	**95%**	**5%**

Percentages define success in finding the specified ligand in the top 1, top 3 of ranked cavities or at all (any). "n PDB" indicates the count of PDB structures in the dataset, and "n ligands" the count of unique ligands (the same ligand can be present in different PDB structures). *PDBBind-HA is the PDBBind dataset restricted to high affinity complexes, with an affinity of 1 μM or lower.

In addition, we used Huang and Schroeder's dataset [33], a dataset commonly used and for which we have performance data for various methods, to compare the results from CAVIAR to those of state of the art cavity identification software (Table 3). Overall, CAVIAR performs well both on the 48 unbound structures and the 48 bound structures, with success rates of 83% and 94% respectively in the top 3 ranked cavities. This is similar to the performances of VICE [41], DoGSite[24] and Fpocket [6]. CAVIAR fails on three cases of the bound dataset, all three are very exposed ligand on flat surfaces of the protein (Table S3).

Table 3. Comparison of CAVIAR against state of art methods for cavity identification on a dataset of 48 bound and 48 unbound diverse protein complexes.

	Top1		*Top3*	
method	**unbound**	**bound**	**unbound**	**bound**
VICE *[41]*	**83%**	**85%**	**90%**	**94%**
CAVIAR*	**77%**	**88%**	**85%**	**94%**
DoGSite *[24]*	**71%**	**83%**	**92%**	**92%**
Fpocket* *[6]*	**69%**	**83%**	**94%**	**92%**
LSite *[24]*	**75%**	**75%**	**85%**	**88%**
PocketPicker *[23]*	**69%**	**72%**	**85%**	**85%**
DSite *[24]*	**65%**	**69%**	**77%**	**79%**
LIGSITE *[38]*	**58%**	**69%**	**75%**	**87%**
CAST *[50]*	**58%**	**67%**	**75%**	**83%**
PASS *[43]*	**60%**	**63%**	**71%**	**81%**
SURFNET *[42]*	**52%**	**54%**	**75%**	**78%**

Percentages define success in finding the specified ligand in the top 1 or top 3 of ranked cavities. Values of all algorithms except CAVIAR were extracted from [24]. CAVIAR's success values were calculated with the definition of [24]. * indicates open-source software available at the time of the study.

Cavities are ranked according to their cavity descriptor scores, which are estimates of importance based on size and buriedness. This score was not developed with the intention to rank cavities with regards to their ligandability but rather to have a heuristic to limit the number of stored cavities in the case of a large scale analysis of the PDB. The software also provides a separate ligandability descriptor (Supplementary Material items S1 and S5). The cavity detection success values in the top 1 and top 3 in Tables 2 and 3 are underestimates of the actual performance of the tool in detecting ligand binding sites. First, interface cavities between protein chains are often larger in volume and will therefore have higher cavity scores (and ranks) than smaller enclosed binding sites. PDB files with repeats of a protein chain can contain repeats of the same cavity, with small variations of scores due to small rearrangements in the binding pocket or grid orientation dependency. These repeated cavities may not all contain ligands, which can place the actual liganded cavity at second or third instead of first rank. The (underestimated) statistics and the visual inspection of the results of the small datasets demonstrate CAVIAR's good performance in detecting liganded cavities with a high confidence, which is a good basis to proceed with detailed descriptor calculations.

Subcavity segmentation

We assembled a dataset of 59 diverse proteins to judge qualitatively the performance of CAVIAR for the decomposition of pockets into subpockets. These proteins are classified by the RCSB PDB as follows: 21 hydrolases, 14 membrane proteins, 7 transferases, 5 transcription regulators, 4 ligases, 2 oxidoreductases, 2 hormone receptors, 1 chaperone, 1 choline binding protein, 1 structural protein and 1 immune system protein. The subcavity segmentation algorithm fits qualitatively to the empirical description of binding subpockets in most cases, but depends on the quality of the detected cavity. In some cases, subpockets are missing because the cavity is not entirely detected, or there are spurious subcavities when the cavity has overspanned. Despite the

introduction of the merging step described above, the decomposition algorithm tends to oversegment cavities. We discuss here four cases of successful cases of cavity segmentation with CAVIAR (Fig. 3) as well as two cases of failures (Fig. 4). These results are also compared with DoGSite default output, to provide a state-of-the-art benchmark, although DoGSite's definition of subpockets is different to CAVIAR's medicinal chemistry focused approach (cf introduction). We selected these six cases with respect to CAVIAR, not a consensus of CAVIAR and DoGSite. The latter was run *a posteriori*, and may not represent an accurate depiction of DoGSite's performance. The whole set of results, including PDB files, cavities and subcavities, is available on the GitHub repository of the CAVIAR package (link in the Availability paragraph).

The first example is the binding pocket of the chaperone protein hsp90-α. It contains two subpockets, namely the adenine subpocket, where the natural ligand, ADP, binds, and a lipophilic subpocket, exploited by small molecule inhibitors to improve their selectivity profiles [72, 73]. CAVIAR correctly identifies the main binding pocket and splits it into two subcavities. One subpocket is occupied by the adenine head group of the ligand, and the other one by its iodo-benzodioxole group (Fig. 3a). DoGSite recognizes the two subpockets, but produces a much larger cavity and generates four subcavities in total (Fig. 3b). The second successful example is HIV-1 protease, which contains six well defined subsites, recognizing specifically amino acid side chains of the peptidic substrate [74, 75]. CAVIAR generates seven subcavities, six of which corresponds to the six standard subsites S1 to S3 and S1' to S3'. The S1 subsite is segmented into two subcavities, which correspond in the selected PDB entries to the piperazine and the benzofurane groups of the ligand (Fig. 3c, chemical groups in magenta and dark blue). In the literature, these two subcavities are referred to as the S1 and extended S1 pockets [74]. DoGSite, on the other hand, correctly predicts the binding pocket, but fails to decompose it into subpockets, *i.e.*, outputs only one single pocket (Fig. 3d). Our third example is the M1 muscarinic acetylcholine receptor (GPCR) complexed with an antagonist. Two subpockets overlap with the orthosteric pocket of the receptor, where the ligand is present, and three additional subpockets are detected at the level of the allosteric pocket (Fig. 3e). In both CAVIAR and DoGSite analyses, the orthosteric and allosteric pockets are connected. At the level of the orthosteric site, one of the two subcavities of CAVIAR overlaps with the amine binding subpocket and contains the quartenary amine of the ligand, while the other defines the more hydrophobic part of the binding pocket and hosts the two thiophen moieties of the inhibitor (Thal *et al.*, 2016). DoGSite results are similar to CAVIAR, except that the orthosteric pocketis not segmented into subsites (Fig. 3f). The last successful case is the EGFR kinase domain bound to lapatinib, for which CAVIAR detects six subpockets (Fig. 3g). The main binding region of ATP, *i.e.*, the adenine, ribose and phosphate regions, is described by one large subpocket, occupied by the main hinge binding motif of the ligand and its furan substituent [77]. More granularity appears at the front and back pockets. The front pocket is divided into two subpockets, not occupied by the ligand. The back pocket contains three subpockets, which correspond to three parts of the ligand: one contains the chloroaniline, one the flexible linker, and the last one the terminal fluorobenzene. The sulfonyl tail of the ligand is solvent exposed and not covered by any cavity grid point. The cavity from DoGSite overlaps similarly with the ligand, but does not decompose the pocket into subcomponents; it detects other connected subpockets far from the ligand binding groove, and produces a site pocket that significantly overspans (Fig. 3h).

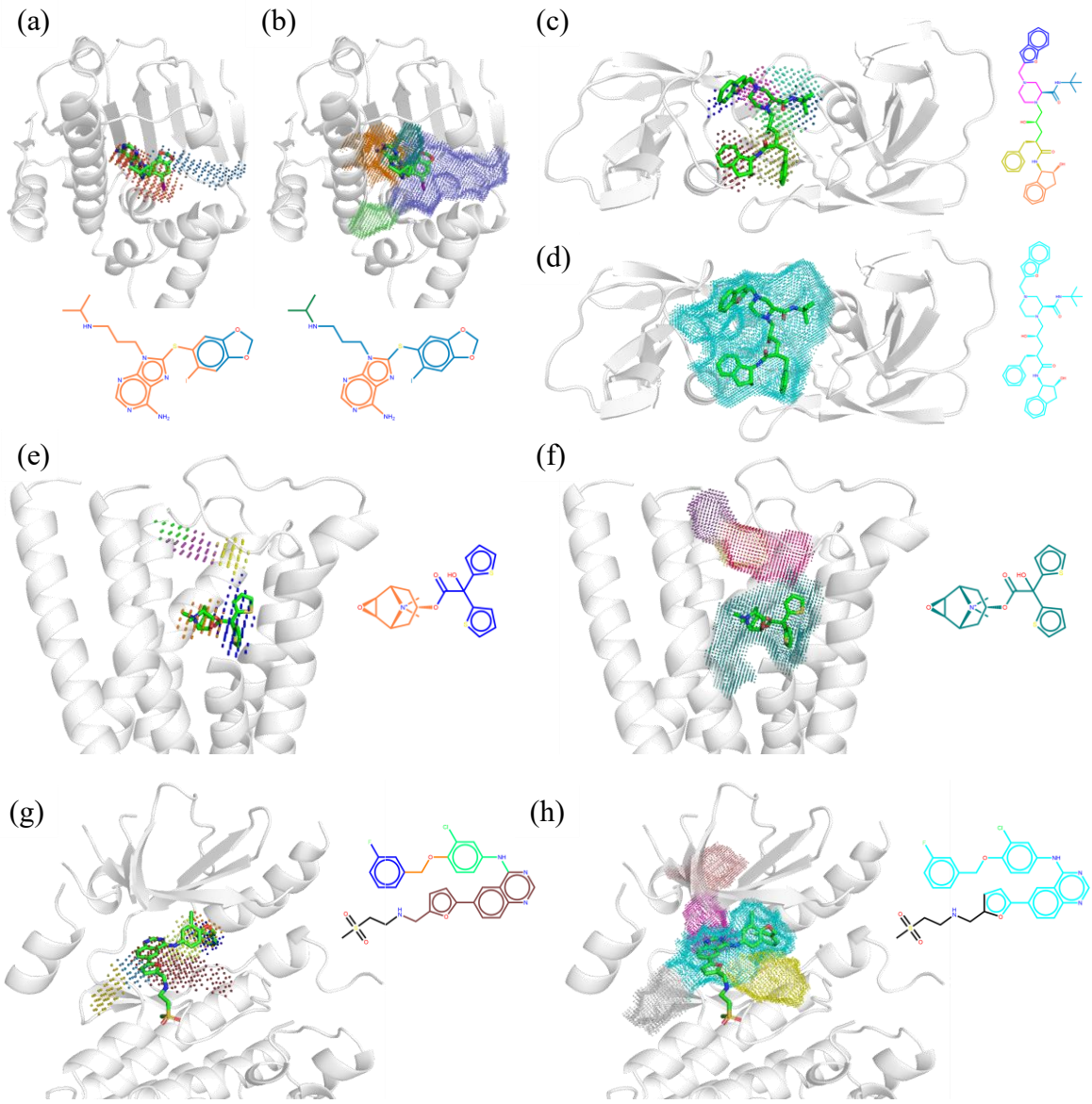


Fig. 3. Examples of successful decomposition by CAVIAR and comparison to DoGSite. In all panels, the atoms in the 2D structure of the ligand are depicted with a color code corresponding to the subcavity segmentation, or in black if not covered by subcavities. (a) and (b) Chaperone protein hsp90, PDB code 2fwz. (a) The CAVIAR subpocket algorithm correctly identifies the adenine pocket, in orange and the lipophilic pocket, in blue. (b) DoGSite identifies the two subpockets (same colors), but overspans. (c) and (d) HIV-1 protease, PDB code 1c70. (c) CAVIAR correctly identifies the six protease subsites (S3 in cyan, unoccupied by the ligand, S2 in light blue, S1 in pink and dark blue, S1' in green, S2' in yellow, and S3' in orange), as well as further decomposes the S1 site into its main site (pink) and an extended S1 pocket (dark blue). (d) DoGSite fails to segment the pocket into subsites. (e) and (f) M1 muscarinic acetylcholine receptor, GPCR, PDB code 5cxv. (e) CAVIAR detects two subpockets in the orthosteric site, which correspond to the amine site (orange spheres) and the lipophilic pocket (blue spheres). (f) DoGSite fails to segment the orthosteric pocket. (g) and (h) EGFR kinase, PDB code 1xkk. (g) CAVIAR pulls together one main subpocket for the adenine site, the sugar site and the phosphates region (red spheres). It

further splits the pocket into its front pocket region (two subpockets in light blue and yellow) and into its back pocket (three pockets in light green, orange and light blue). (h) DoGSite significantly overspans towards the back of the protein (salmon and pink dots).

In some cases, CAVIAR fails to produce any relevant deconstruction of cavities into subpockets. Examples of such include factor Xa (PDB code 2bqw) and HCV NS3 protease (PDB code 3kee). In both cases, parts of the ligands and the cavities are very solvent-exposed, which hinders the detection of the entirety of the cavities (Fig. 4). As the detected cavity is too small, it cannot be segmented effectively into subpockets. Both CAVIAR and DoGSite fail in these two cases, although DoGSite tends to detect larger portions of the binding pocket.

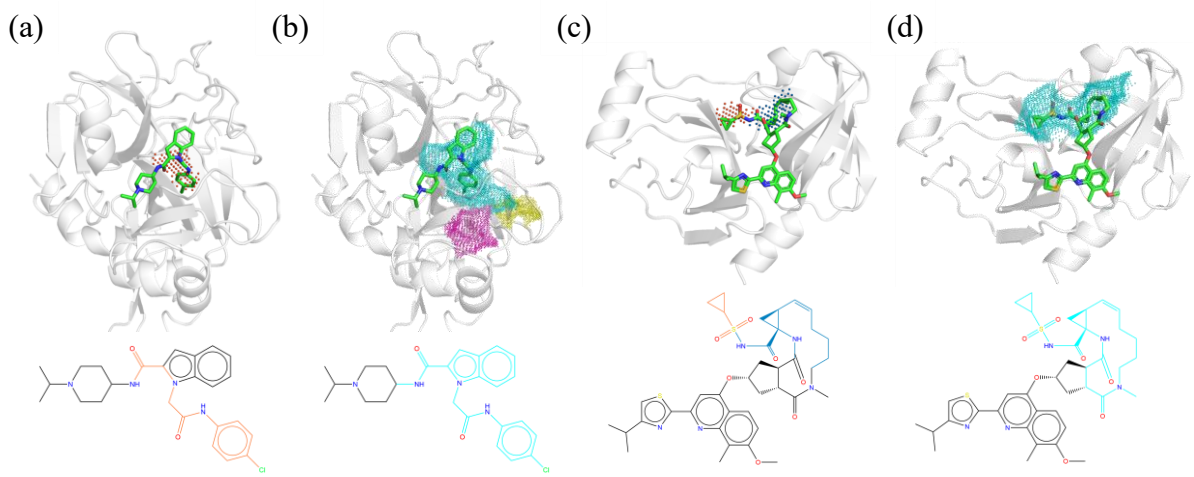


Fig. 4. Examples of unsuccessful decomposition by CAVIAR and comparison to DoGSite. In all panels, the 2D structure of the ligand is depicted with a color code corresponding to the subcavity segmentation, or in black if not covered by subcavities. (a) CAVIAR and (b) DoGSite cavity detection and segmentation of factor Xa protease, PDB code 2bqw. (c) CAVIAR and (d) DoGSite cavity detection and segmentation of HCV NS3 protease. In all cases, both approaches fail to describe correctly the entirety of the cavities and their complexity in terms of subpockets.

Liganded cavities have higher complexity than apo cavities

We analyzed 97,221 X-ray structures from the PDB that passed a filtering protocol: only X-ray structures with a resolution better than 2.5 Å; no flag as obsolete or other warnings in the PDB header. On average, each PDB structure has 8.3 ± 11.6 cavities and a median of 5, with the number of cavities per PDB file increasing with the number of residues in the PDB file and the number of protein chains. Cavities are segmented on average into 2.7 ± 2.9 subcavities, with a median of 2. About 140,000 of the 800,000 detected cavities are liganded, with an average ligand coverage of 79 ± 25% and a median of 88%. The analysis of holo cavities tend to show that cavities do no overspan significantly, as the average cavity coverage by ligand atom is 60 ± 31% and a median of 62%. This is a much higher cavity coverage compared to previous reports, which argued that ligands fill on average only a third of their binding pockets [56, 57]. If the analysis is focused on the drug-like ligands of the PDBBind dataset, the cavity coverage rises to 74 ± 26% with a median of 82%. Liganded cavities tend to be bigger, more hydrophobic, more ligandable and more

geometrically complex (segmented into more subcavities) compared to apo cavities (Table 4). Ligands occupy on average 2.5 ± 1.5 subcavities with a median of 2.

Table 4. Differences between liganded cavities and holo cavities in the PDB.

	Liganded cavities N=138,632	*Apo cavities N=668,621*
Size (Å^3)	353 ± 423 Median = 238	145 ± 208 Median = 83
Number of subcavities	4.4 ± 4.6 Median = 3.0	2.3 ± 2.3 Median = 2.0
Hydrophobicity	45 ± 17% Median = 43%	39 ± 17% Median = 38%
Ligandability	0.62 ± 0.27 Median = 0.60	0.51 ± 0.26 Median = 0.40

All comparisons are significant with Kolmogorov-Smirnov tests with a significance level of 0.01 (Table S8).

Binding affinity increases with the number of subcavities filled by the ligand

We compared the binding affinities of ligand to their targets and the number of subcavities they interact with using the PDBBind dataset and with a focus on two types of drug targets, proteases and kinases. The more subcavities a compound fills, the higher the affinity. This effect is more pronounced for compounds binding to more than three subcavities, most of them having a binding affinity in the nanomolar range or better (Fig. 5).

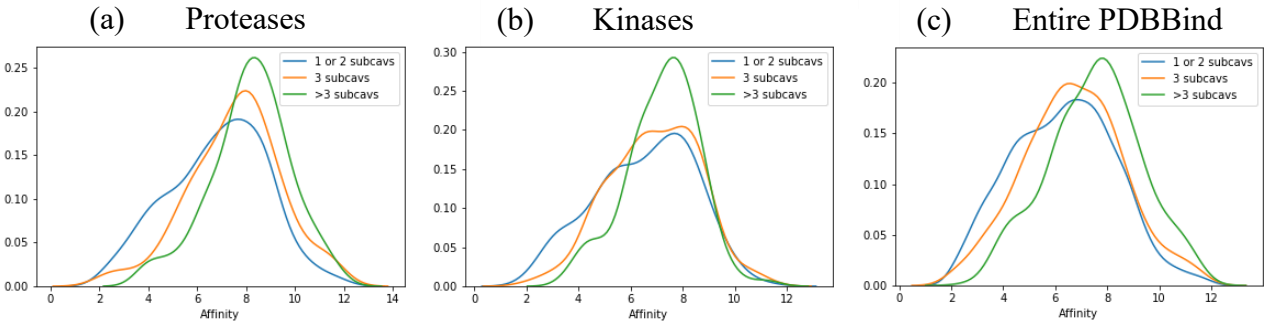


Fig. 5. Distribution of binding affinities expressed as –log(affinity) in function of the numbers of subcavities filled by the ligand. (a) Protease dataset. In blue, ligands filling one or two subcavities (n=453), orange three subcavities (n=154), and green four or more subcavities (n=194). The peak of activity is in all cases in the nanomolar range, however, the more subcavities are filled, the less there are micromolar or worse binders and the more low nanomolar or better binders are found. (b) Kinase dataset. Same colors as A, with 249 molecules binding to one or two subcavities, 122 to three and 103 to four or more. (c) Entire PDBBind dataset. Same colors as A, with 2,456 molecules binding to one or two subcavities, 800 to three and 579 to four or more.

Binding affinities increase linearly in the protease dataset as more subpockets are involved in ligand binding (Table 5). In detail, 801 unique proteases pockets contain ligands and the average –log(affinity) ranges from 6.7 for ligands filling only one subcavity to 7.0 for two, 7.5 for three

and 8.2 for four and more subcavities. Differences between subsets are significant according to Kolmogorov-Smirnov tests for all subsets, *i.e.*, one, two or three subcavities filled versus the four or more subcavities subset, but also joined subsets of two and less subcavities versus four and more, and three or less versus the four and more subcavities (detailed statistics in Table S8).

Table 5. Affinities according to number of subcavities bound by the ligand in the protease dataset.

	1 subcav	*2 subcavs*	*3 subcavs*	*>3 subcavs*
Proteases (n = 801)	6.7 +/- 2.0 (207)	7.0 +/- 2.0 (246)	7.5 +/- 1.9 (154)	8.2 +/- 1.6 (194)

Mean values and standard deviation of –log(affinity) are given, with the number of PDB entries for each category in parenthesis.

In general, if we extend the analysis to kinases and the rest of the PDBBind dataset, compounds filling four or more subpockets have a substantially more favorable binding affinity to their drug target. Only 9%, 16% and 19% of ligands binding to at least four subpockets have an affinity to their target in the micromolar range or lower in the proteases (17 out of 194), kinases (16 out of 103), and entire PDBBind datasets (111 out of 570), respectively. Compounds binding to a maximum of three subcavities are 29%, 36% and 42% in the micromolar or lower range, in the proteases, kinases and entire PDBBind datasets, respectively (Table 6).

Table 6. Comparison of binding affinities of ligands occupying up to three subcavities and ligands occupying more.

Micromolar or worse ligands occupying	*Proteases (801)*	*Kinases (474)*	*PDBBind (3,826)*
Up to 3 subcavities	29% (of 607)	36% (of 371)	42% (of 3,256)
> 3 subcavities	9% (of 194)	16% (of 103)	19% (of 570)

Numbers in parenthesis indicate the total count of unique PDB in each set. The proportion of weak binders binding to up to three subcavities is doubled to tripled in all datasets compared to ligands binding four or more subcavities.

Using CAVIAR binding site descriptors: ligandability

We tested CAVIAR descriptors in ML models of binding site ligandability predictions. Datasets from the non-redundant set of druggable and less druggable binding sites (NRDLD) [78], as described by others [9, 78, 79], were taken and fifteen ML models based on 27 global binding site descriptors generated with CAVIAR were built (further details in Supplementary Materials items S1 and S5). The best model, a k-nearest neighbors algorithm, performs similarly to state-of-the-art methods [9, 78, 79] on the minimalistic validation set of the NRDLD (23 ligandable and 14 unligandable binding sites), with a Matthews correlation coefficient of 0.73 (Table 7, S4 and S5). Interestingly, the five outliers in the model are non ligandable cavities being predicted as ligandable, but four of these five cavities are actually liganded. This model will benefit from a larger training set and a broader use of the different CAVIAR descriptors.

Table 7. Matthews correlation coefficient (MCC) and accuracy of five software for the prediction of ligandability of cavities.

	k-NN (CAVIAR)	***VolSite***	***DrugPred***	***PockDrug***	***Fpocket***	***SiteMap***
MCC	0.73	0.77	0.77	0.54	0.39	0.24
Accuracy	0.86	0.89	0.89	0.76	0.73	0.65

Values for VolSite [9], DrugPred [78], Fpocket [80] and SiteMap [7] are extracted from [9], values for PockDrug from [79].

Limitations of the method

The most obvious limitations of CAVIAR are inherent to the experimental data it relies on, primarily protein structures obtained with X-ray crystallography and cryogenic electron microscopy. This is common to all methods, and cannot be circumvented. If a flexible cryptic pocket of interest is not present in the structure given as input to CAVIAR, it will not detect it. While this limitation cannot be solved systematically, it can be mitigated by generating series of structures *in silico*, *e.g.*, by generating conformational ensembles from sampling methods [81–83]. Crystal contacts, artifacts and protein chain repeats can produce spurious non-productive interchain cavities (Fig. 6a). Significant work has been invested into detecting biologically relevant protein chains contacts [84, 85] and we plan to implement such an algorithm in later versions of our tool. The second intrinsic limitation of CAVIAR is that it is designed for discovering cavities that potentially bind small organic drug-like compounds, which tends to exclude surface patches such as protein-protein interfaces and very exposed ligand binding grooves (Fig. 6b). Different sets of parameters could be identified and optimized for detecting surface patches, or even protein-protein interaction interfaces; this is work in progress. Key parameter settings are stored in a configuration file:optimizing the software for the detection of exposed binding grooves mostly requires the assembly of carefully curated target optimization datasets.

Technically, CAVIAR suffers from other kind of limitations. As other cavity detection tools, it may overspan cavities because validation scores tend to reward larger pockets that are more likely to contain the ligand. We optimized the algorithm and the parameters to restrict to identifying cavities from the direct surroundings of known ligands, but edge cases still evade our best efforts and produce very large invaginations (Fig. 6c). The validation of protein cavity detection algorithms is not simple, as there is no ground truth definition of what is a “protein cavity” and so it is hard to design a meaningful validation dataset. This shortcoming is exacerbated for the segmentation of cavities into subcavities, as again a systematic definition simply does not exist, to our knowledge. Provided that the input cavity is correctly described, the subsite decomposition suffers from very few false negatives. In other words, it tends to produce more

subpockets rather than fewer, that is, the algorithm oversegments the pocket rather than fail to characterize a subcavity.

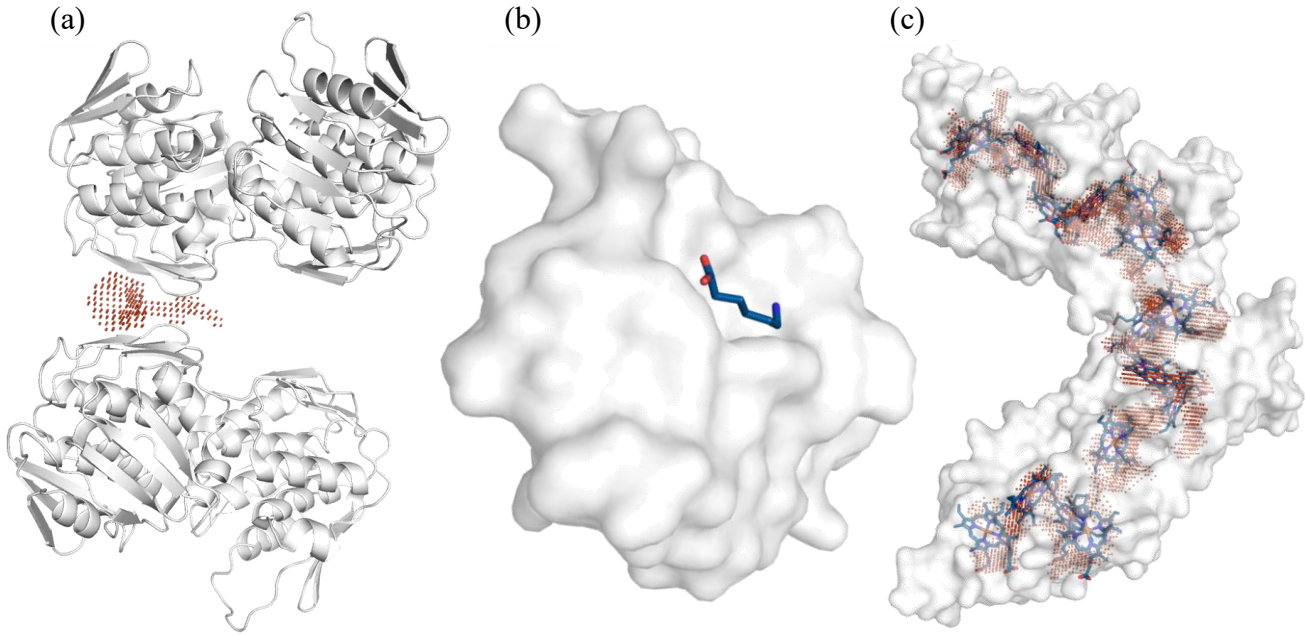


Fig. 6. Representative cases of failure with CAVIAR. (a) Spurious interchain cavity. A cavity, in orange spheres, is found at the interface between two protein chains, in white cartoons, which is a crystal contact and not biologically relevant (PDB code 1ejd). (b) Case of an exposed ligand, in blue sticks, on top of a flat surface of a protein, in white surface (PDB code 2pk4). The binding surface patch is too exposed to be detected with CAVIAR's default set of parameters. (c) A cavity, in orange spheres, overspans inside the entire protein chain, in white surface representation (PDB code 2cvc). However, in this case, numerous ligands, in blue sticks, are present everywhere inside the cavity.

CONCLUSIONS

The fruitful investigation of protein binding sites by ML requires robust and meaningful descriptors, which in turn rely upon a reliable cavity detection method. The open-source availability of CAVIAR on GitHub and Anaconda combined with its comprehensive Python interface defines it as a powerful toolkit for this purpose. The descriptors computed by the software are readily usable in standard ML packages, such as scikit-learn and TensorFlow. CAVIAR is mmCIF-ready, and incorporates a molecular dynamics trajectory parser; the subpocket characterization relies on on the protein, and does not require a bound ligand to work. A dedicated website is available with step-by-step usage notes and an extended manual to help tune CAVIAR to their datasets (see links in the code availability paragraph). The cavity detection, characterization and segmentation runs fast, ranging from a five seconds average on the DUD-e 102 targets, to a ten seconds average on the scPDB dataset on one core of a Xeon E5-4620 CPU of 2012 with a clock speed of 2.20 GHz. A qualitative comparison of CAVIAR, DoGSite,

Schrödinger's SiteMap and Fpocket on few test cases indicates that CAVIAR is much faster than DoGSite or SiteMap, but slower than Fpocket.

Some novel notions were introduced as an attempt to refine the cavity detection and address challenges that are not resolved in the literature: cavity overspanning of buriedness-based algorithms, and the analysis of protein subpockets. A novel approach to scoring and trimming of cavity points is presented; this has been shown to prevent major cavity overspanning. CAVIAR describes cavities that exhibits very high cavity coverage of ligand atoms, with a median of 62% in the entire PDB and 82% in the drug-like structural database PDBBind: other published method have ligand-cavity coverage of around 33% [56, 57]. CAVIAR is able to identify binding subsites in both apo and holo protein structures. CAVIAR aims at a systematic detection and classification of protein subcavities. The investigation of protein subcavities may help to understand selectivity issues or polypharmacological effects of certain drugs, also known as chemoisosterism of protein environments [86]. In other words, it is possible to define matched "subcavities" pairs of protein cavities in the same manner as matched molecular pairs of chemicals [87]. The notion of subcavity is not a well-defined concept and the robust partitioning of binding pockets into subpockets is an unmet need. The deconstruction of pockets into subcavities may help for partial cavity matching in the context of cavity comparison [88]. Our analysis of the PDB database led to the identification of significant differences between apo and holo cavities, in terms of size, ligandability, hydrophobicity and complexity. Finally, in line with the fragment-based drug design paradigm [52, 54], we found that the binding affinity of small molecule ligands scales with the number of subcavities they fill, with a propensity towards high affinities, in the nanomolar range or better, for ligands binding to more than three subcavities.

ACKNOWLEDGMENT

The authors thank Imtiaz Hossein, Michael Schaefer and Richard Lewis for insightful discussions. J.-R.M. thanks the ProDy development team and generally all contributors to open source codes for their crucial work.

REFERENCES

1. Westbrook JD, Burley SK (2019) How Structural Biologists and the Protein Data Bank Contributed to Recent FDA New Drug Approvals. Structure 27:211–217. https://doi.org/10.1016/j.str.2018.11.007

2. Simões T, Lopes D, Dias S, et al (2017) Geometric Detection Algorithms for Cavities on Protein Surfaces in Molecular Graphics: A Survey. Comput Graph Forum 36:643–683. https://doi.org/10.1111/cgf.13158

3. Volkamer A, Behren MM von, Bietz S, Rarey M (2018) Prediction, Analysis, and Comparison of Active Sites. In: Applied Chemoinformatics. John Wiley & Sons, Ltd, pp 283–311

4. Macari G, Toti D, Polticelli F (2019) Computational methods and tools for binding site recognition between proteins and small molecules: from classical geometrical approaches to modern machine learning strategies. J Comput Aided Mol Des 33:887–903. https://doi.org/10.1007/s10822-019-00235-7

5. Volkamer A, Kuhn D, Rippmann F, Rarey M (2012) DoGSiteScorer: a web server for automatic binding site prediction, analysis and druggability assessment. Bioinformatics 28:2074–2075. https://doi.org/10.1093/bioinformatics/bts310

6. Le Guilloux V, Schmidtke P, Tuffery P (2009) Fpocket: An open source platform for ligand pocket detection. BMC Bioinformatics 10:168. https://doi.org/10.1186/1471-2105-10-168

7. Halgren TA (2009) Identifying and Characterizing Binding Sites and Assessing Druggability. J Chem Inf Model 49:377–389. https://doi.org/10.1021/ci800324m

8. Nayal M, Honig B (2006) On the nature of cavities on protein surfaces: Application to the identification of drug-binding sites. Proteins Struct Funct Bioinforma 63:892–906. https://doi.org/10.1002/prot.20897

9. Desaphy J, Azdimousa K, Kellenberger E, Rognan D (2012) Comparison and Druggability Prediction of Protein–Ligand Binding Sites from Pharmacophore-Annotated Cavity Shapes. J Chem Inf Model 52:2287–2299. https://doi.org/10.1021/ci300184x

10. Ehrt C, Brinkjost T, Koch O (2016) Impact of Binding Site Comparisons on Medicinal Chemistry and Rational Molecular Design. J Med Chem 59:4121–4151. https://doi.org/10.1021/acs.jmedchem.6b00078

11. Xie L, Evangelidis T, Xie L, Bourne PE (2011) Drug Discovery Using Chemical Systems Biology: Weak Inhibition of Multiple Kinases May Contribute to the Anti-Cancer Effect of Nelfinavir. PLOS Comput Biol 7:e1002037. https://doi.org/10.1371/journal.pcbi.1002037

12. Möller-Acuña P, Contreras-Riquelme JS, Rojas-Fuentes C, et al (2015) Similarities between the Binding Sites of SB-206553 at Serotonin Type 2 and Alpha7 Acetylcholine Nicotinic Receptors: Rationale for Its Polypharmacological Profile. PLOS ONE 10:e0134444. https://doi.org/10.1371/journal.pone.0134444

13. Schumann M, Armen RS (2013) Identification of Distant Drug Off-Targets by Direct Superposition of Binding Pocket Surfaces. PLOS ONE 8:e83533. https://doi.org/10.1371/journal.pone.0083533

14. Schirris TJJ, Ritschel T, Herma Renkema G, et al (2015) Mitochondrial ADP/ATP exchange inhibition: a novel off-target mechanism underlying ibipinabant-induced myotoxicity. Sci Rep 5:1–12. https://doi.org/10.1038/srep14533

15. Kuhn D, Weskamp N, Schmitt S, et al (2006) From the Similarity Analysis of Protein Cavities to the Functional Classification of Protein Families Using Cavbase. J Mol Biol 359:1023–1044. https://doi.org/10.1016/j.jmb.2006.04.024

16. Kinoshita K, Furui J, Nakamura H (2002) Identification of protein functions from a molecular surface database, eF-site. J Struct Funct Genomics 2:9–22. https://doi.org/10.1023/A:1011318527094

17. Konc J, Hodošček M, Ogrizek M, et al (2013) Structure-Based Function Prediction of Uncharacterized Protein Using Binding Sites Comparison. PLOS Comput Biol 9:e1003341. https://doi.org/10.1371/journal.pcbi.1003341

18. Anand P, Sankaran S, Mukherjee S, et al (2011) Structural Annotation of Mycobacterium tuberculosis Proteome. PLOS ONE 6:e27044. https://doi.org/10.1371/journal.pone.0027044

19. Al-Gharabli SI, Shah STA, Weik S, et al (2006) An Efficient Method for the Synthesis of Peptide Aldehyde Libraries Employed in the Discovery of Reversible SARS Coronavirus Main Protease (SARS-CoV Mpro) Inhibitors. ChemBioChem 7:1048–1055. https://doi.org/10.1002/cbic.200500533

20. Willmann D, Lim S, Wetzel S, et al (2012) Impairment of prostate cancer cell growth by a selective and reversible lysine-specific demethylase 1 inhibitor. Int J Cancer 131:2704–2709. https://doi.org/10.1002/ijc.27555

21. Kooistra AJ, Leurs R, de Esch IJP, de Graaf C (2015) Structure-Based Prediction of G-Protein-Coupled Receptor Ligand Function: A β-Adrenoceptor Case Study. J Chem Inf Model 55:1045–1061. https://doi.org/10.1021/acs.jcim.5b00066

22. Weber A, Casini A, Heine A, et al (2004) Unexpected Nanomolar Inhibition of Carbonic Anhydrase by COX-2-Selective Celecoxib: New Pharmacological Opportunities Due to Related Binding Site Recognition. J Med Chem 47:550–557. https://doi.org/10.1021/jm030912m

23. Weisel M, Proschak E, Schneider G (2007) PocketPicker: analysis of ligand binding-sites with shape descriptors. Chem Cent J 1:7. https://doi.org/10.1186/1752-153X-1-7

24. Volkamer A, Griewel A, Grombacher T, Rarey M (2010) Analyzing the Topology of Active Sites: On the Prediction of Pockets and Subpockets. J Chem Inf Model 50:2041–2052. https://doi.org/10.1021/ci100241y

25. Goodford PJ (1985) A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. J Med Chem 28:849–857. https://doi.org/10.1021/jm00145a002

26. Bliznyuk AA, Gready JE (1998) Identification and energetic ranking of possible docking sites for pterin on dihydrofolate reductase. J Comput Aided Mol Des 12:325–333. https://doi.org/10.1023/A:1008039000355

27. Ngan CH, Bohnuud T, Mottarella SE, et al (2012) FTMAP: extended protein mapping with user-selected probe molecules. Nucleic Acids Res 40:W271–W275. https://doi.org/10.1093/nar/gks441

28. Laurie ATR, Jackson RM (2005) Q-SiteFinder: an energy-based method for the prediction of protein–ligand binding sites. Bioinformatics 21:1908–1916. https://doi.org/10.1093/bioinformatics/bti315

29. Marchand J-R, Caflisch A (2018) In silico fragment-based drug design with SEED. Eur J Med Chem 156:907–917. https://doi.org/10.1016/j.ejmech.2018.07.042

30. Miranker A, Karplus M (1991) Functionality maps of binding sites: A multiple copy simultaneous search method. Proteins Struct Funct Bioinforma 11:29–34. https://doi.org/10.1002/prot.340110104

31. Simões T, Lopes D, Dias S, et al (2017) Geometric Detection Algorithms for Cavities on Protein Surfaces in Molecular Graphics: A Survey. Comput Graph Forum J Eur Assoc Comput Graph 36:643–683. https://doi.org/10.1111/cgf.13158

32. Xie Z-R, Hwang M-J (2015) Methods for Predicting Protein–Ligand Binding Sites. In: Kukol A (ed) Molecular Modeling of Proteins. Springer, New York, NY, pp 383–398

33. Huang B, Schroeder M (2006) LIGSITEcsc: predicting ligand binding sites using the Connolly surface and degree of conservation. BMC Struct Biol 6:19. https://doi.org/10.1186/1472-6807-6-19

34. Capra JA, Laskowski RA, Thornton JM, et al (2009) Predicting Protein Ligand Binding Sites by Combining Evolutionary Sequence Conservation and 3D Structure. PLOS Comput Biol 5:e1000585. https://doi.org/10.1371/journal.pcbi.1000585

35. Huang B (2009) MetaPocket: A Meta Approach to Improve Protein Ligand Binding Site Prediction. OMICS J Integr Biol 13:325–330. https://doi.org/10.1089/omi.2009.0045

36. Zhang Z, Li Y, Lin B, et al (2011) Identification of cavities on protein surface using multiple computational approaches for drug binding site prediction. Bioinformatics 27:2083–2088. https://doi.org/10.1093/bioinformatics/btr331

37. Levitt DG, Banaszak LJ (1992) POCKET: A computer graphies method for identifying and displaying protein cavities and their surrounding amino acids. J Mol Graph 10:229–234. https://doi.org/10.1016/0263-7855(92)80074-N

38. Hendlich M, Rippmann F, Barnickel G (1997) LIGSITE: automatic and efficient detection of potential small molecule-binding sites in proteins. J Mol Graph Model 15:359–363. https://doi.org/10.1016/S1093-3263(98)00002-3

39. Kalidas Y, Chandra N (2008) PocketDepth: a new depth based algorithm for identification of ligand binding sites in proteins. J Struct Biol 161:31–42. https://doi.org/10.1016/j.jsb.2007.09.005

40. Till MS, Ullmann GM (2010) McVol - A program for calculating protein volumes and identifying cavities by a Monte Carlo algorithm. J Mol Model 16:419–429. https://doi.org/10.1007/s00894-009-0541-y

41. Tripathi A, Kellogg GE (2010) A novel and efficient tool for locating and characterizing protein cavities and binding sites. Proteins Struct Funct Bioinforma 78:825–842. https://doi.org/10.1002/prot.22608

42. Laskowski RA (1995) SURFNET: A program for visualizing molecular surfaces, cavities, and intermolecular interactions. J Mol Graph 13:323–330. https://doi.org/10.1016/0263-7855(95)00073-9

43. Brady GP, Stouten PFW (2000) Fast prediction and visualization of protein binding pockets with PASS. J Comput Aided Mol Des 14:383–401. https://doi.org/10.1023/A:1008124202956

44. Kawabata T, Go N (2007) Detection of pockets on protein surfaces using small and large probe spheres to find putative ligand binding sites. Proteins 68:516–529. https://doi.org/10.1002/prot.21283

45. Oliveira SH, Ferraz FA, Honorato RV, et al (2014) KVFinder: steered identification of protein cavities as a PyMOL plugin. BMC Bioinformatics 15:197. https://doi.org/10.1186/1471-2105-15-197

46. Kawabata T (2010) Detection of multiscale pockets on protein surfaces using mathematical morphology. Proteins Struct Funct Bioinforma 78:1195–1211. https://doi.org/10.1002/prot.22639

47. Yu J, Zhou Y, Tanaka I, Yao M (2010) Roll: a new algorithm for the detection of protein pockets and cavities with a rolling probe sphere. Bioinformatics 26:46–52. https://doi.org/10.1093/bioinformatics/btp599

48. Lewis RA (1989) Determination of clefts in receptor structures. J Comput Aided Mol Des 3:133–147. https://doi.org/10.1007/BF01557724

49. Peters KP, Fauck J, Frömmel C (1996) The Automatic Search for Ligand Binding Sites in Proteins of Known Three-dimensional Structure Using only Geometric Criteria. J Mol Biol 256:201–213. https://doi.org/10.1006/jmbi.1996.0077

50. Liang J, Edelsbrunner H, Woodward C (1998) Anatomy of protein pockets and cavities: measurement of binding site geometry and implications for ligand design. Protein Sci Publ Protein Soc 7:1884–1897

51. Simões TMC, Gomes AJP (2019) CavVis—A Field-of-View Geometric Algorithm for Protein Cavity Detection. J Chem Inf Model 59:786–796. https://doi.org/10.1021/acs.jcim.8b00572

52. Hajduk PJ, Meadows RP, Fesik SW (1997) Discovering High-Affinity Ligands for Proteins. Science 278:497–499. https://doi.org/10.1126/science.278.5337.497

53. Bartolowits M, Davisson VJ (2016) Considerations of Protein Subpockets in Fragment-Based Drug Design. Chem Biol Drug Des 87:5–20. https://doi.org/10.1111/cbdd.12631

54. Erlanson DA, Fesik SW, Hubbard RE, et al (2016) Twenty years on: the impact of fragments on drug discovery. Nat Rev Drug Discov 15:605–619. https://doi.org/10.1038/nrd.2016.109

55. Marchand J-R, Dalle Vedove A, Lolli G, Caflisch A (2017) Discovery of Inhibitors of Four Bromodomains by Fragment-Anchored Ligand Docking. J Chem Inf Model 57:2584–2597. https://doi.org/10.1021/acs.jcim.7b00336

56. Wirth M, Volkamer A, Zoete V, et al (2013) Protein pocket and ligand shape comparison and its application in virtual screening. J Comput Aided Mol Des 27:511–524. https://doi.org/10.1007/s10822-013-9659-1

57. Kahraman A, Morris RJ, Laskowski RA, Thornton JM (2007) Shape Variation in Protein Binding Pockets and their Ligands. J Mol Biol 368:283–301. https://doi.org/10.1016/j.jmb.2007.01.086

58. Chan AWE, Laskowski RA, Selwood DL (2010) Chemical Fragments that Hydrogen Bond to Asp, Glu, Arg, and His Side Chains in Protein Binding Sites. J Med Chem 53:3086–3094. https://doi.org/10.1021/jm901696w

59. Wang L, Xie Z, Wipf P, Xie X-Q (2011) Residue Preference Mapping of Ligand Fragments in the Protein Data Bank. J Chem Inf Model 51:807–815. https://doi.org/10.1021/ci100386y

60. Durrant JD, Friedman AJ, McCammon JA (2011) CrystalDock: A Novel Approach to Fragment-Based Drug Design. J Chem Inf Model 51:2573–2580. https://doi.org/10.1021/ci200357y

61. Tang GW, Altman RB (2014) Knowledge-based Fragment Binding Prediction. PLOS Comput Biol 10:e1003589. https://doi.org/10.1371/journal.pcbi.1003589

62. Kalliokoski T, Olsson TSG, Vulpetti A (2013) Subpocket Analysis Method for Fragment-Based Drug Discovery. J Chem Inf Model 53:131–141. https://doi.org/10.1021/ci300523r

63. Wood DJ, Vlieg J de, Wagener M, Ritschel T (2012) Pharmacophore Fingerprint-Based Approach to Binding Site Subpocket Similarity and Its Application to Bioisostere Replacement. J Chem Inf Model 52:2031–2043. https://doi.org/10.1021/ci3000776

64. Volkamer A, Grombacher T, Rarey M (2010) Where are the boundaries? Automated pocket detection for druggability studies. J Cheminformatics 2:P11. https://doi.org/10.1186/1758-2946-2-S1-P11

65. Pedregosa F, Varoquaux G, Gramfort A, et al (2011) Scikit-learn: Machine Learning in Python. J Mach Learn Res 12:2825–2830

66. Martín Abadi, Ashish Agarwal, Paul Barham, et al (2015) TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems

67. Paszke A, Gross S, Massa F, et al (2019) PyTorch: An Imperative Style, High-Performance Deep Learning Library. Adv Neural Inf Process Syst 32:8026–8037

68. Mysinger MM, Carchia M, Irwin JohnJ, Shoichet BK (2012) Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. J Med Chem 55:6582–6594. https://doi.org/10.1021/jm300687e

69. Desaphy J, Bret G, Rognan D, Kellenberger E (2015) sc-PDB: a 3D-database of ligandable binding sites—10 years on. Nucleic Acids Res 43:D399–D404. https://doi.org/10.1093/nar/gku928

70. Liu Z, Li Y, Han L, et al (2015) PDB-wide collection of binding data: current status of the PDBbind database. Bioinformatics 31:405–412. https://doi.org/10.1093/bioinformatics/btu626

71. Beucher S (1994) Watershed, Hierarchical Segmentation and Waterfall Algorithm. In: Serra J, Soille P (eds) Mathematical Morphology and Its Applications to Image Processing. Springer Netherlands, Dordrecht, pp 69–76

72. Pirard B, Ertl P (2015) Evaluation of a Semi-Automated Workflow for Fragment Growing. J Chem Inf Model 55:180–193. https://doi.org/10.1021/ci5006355

73. Huth JR, Park C, Petros AM, et al (2007) Discovery and Design of Novel HSP90 Inhibitors Using Multiple Fragment-based Design Strategies. Chem Biol Drug Des 70:1–12. https://doi.org/10.1111/j.1747-0285.2007.00535.x

74. Ghosh AK, Osswald HL, Prato G (2016) Recent Progress in the Development of HIV-1 Protease Inhibitors for the Treatment of HIV/AIDS. J Med Chem 59:5172–5208. https://doi.org/10.1021/acs.jmedchem.5b01697

75. Munshi S, Chen Z, Yan Y, et al (2000) An alternate binding site for the P1–P3 group of a class of potent HIV-1 protease inhibitors as a result of concerted structural change in the 80s loop of the protease. Acta Crystallogr D Biol Crystallogr 56:381–388. https://doi.org/10.1107/S0907444900000469

76. Thal DM, Sun B, Feng D, et al (2016) Crystal structures of the M1 and M4 muscarinic acetylcholine receptors. Nature 531:335–340. https://doi.org/10.1038/nature17188

77. Wood ER, Truesdale AT, McDonald OB, et al (2004) A Unique Structure for Epidermal Growth Factor Receptor Bound to GW572016 (Lapatinib): Relationships among Protein Conformation, Inhibitor Off-Rate, and Receptor Activity in Tumor Cells. Cancer Res 64:6652–6659. https://doi.org/10.1158/0008-5472.CAN-04-1168

78. Krasowski A, Muthas D, Sarkar A, et al (2011) DrugPred: A Structure-Based Approach To Predict Protein Druggability Developed Using an Extensive Nonredundant Data Set. J Chem Inf Model 51:2829–2842. https://doi.org/10.1021/ci200266d

79. Borrel A, Regad L, Xhaard H, et al (2015) PockDrug: A Model for Predicting Pocket Druggability That Overcomes Pocket Estimation Uncertainties. J Chem Inf Model 55:882–895. https://doi.org/10.1021/ci5006004

80. Schmidtke P, Barril X (2010) Understanding and Predicting Druggability. A High-Throughput Method for Detection of Drug Binding Sites. J Med Chem 53:5858–5867. https://doi.org/10.1021/jm100574m

81. Bacci M, Langini C, Vymětal J, et al (2017) Focused conformational sampling in proteins. J Chem Phys 147:195102. https://doi.org/10.1063/1.4996879

82. Laio A, Gervasio FL (2008) Metadynamics: a method to simulate rare events and reconstruct the free energy in biophysics, chemistry and material science. Rep Prog Phys 71:126601. https://doi.org/10.1088/0034-4885/71/12/126601

83. Kuzmanic A, Bowman GR, Juarez-Jimenez J, et al (2020) Investigating Cryptic Binding Sites by Molecular Dynamics Simulations. Acc Chem Res. https://doi.org/10.1021/acs.accounts.9b00613

84. Duarte JM, Srebniak A, Schärer MA, Capitani G (2012) Protein interface classification by evolutionary analysis. BMC Bioinformatics 13:334. https://doi.org/10.1186/1471-2105-13-334

85. Capitani G, Duarte JM, Baskaran K, et al (2016) Understanding the fabric of protein crystals: computational classification of biological interfaces and crystal contacts. Bioinformatics 32:481–489. https://doi.org/10.1093/bioinformatics/btv622

86. Jalencas X, Mestres J (2013) Chemoisosterism in the Proteome. J Chem Inf Model 53:279–292. https://doi.org/10.1021/ci3002974

87. Keefer CE, Chang G (2017) The use of matched molecular series networks for cross target structure activity relationship translation and potency prediction. MedChemComm 8:2067–2078. https://doi.org/10.1039/C7MD00465F

88. Krotzky T, Rickmeyer T, Fober T, Klebe G (2014) Extraction of Protein Binding Pockets in Close Neighborhood of Bound Ligands Makes Comparisons Simple Due to Inherent Shape Similarity. J Chem Inf Model 54:3229–3237. https://doi.org/10.1021/ci500553a

Supplementary Material

CAVIAR: a method for automatic cavity detection, description and decomposition into subcavities

Jean-Rémy Marchand, Bernard Pirard, Peter Ertl, Finton Sirockin**

Novartis Institutes for Biomedical Research, Fabrikstrasse 16, 4056 Basel, Switzerland

E-mail: jean-remy.marchand@novartis.com, finton.sirockin@novartis.com

Supplementary Material 1 – Extended methods

PDB parsing and object selection. PDB files are parsed and information from the header are retained for exclusion criteria and further analysis. The PDB parser is adapted from the ProDy source code.[1] Protein chains with fewer than thirty residues are excluded, hydrogen atoms are ignored. Metal ions and well-coordinated water molecules are retained. Well-coordinated water molecules are defined by contacts within 3.1 Å to at least three hydrogen bond donor/acceptor heavy atoms from the protein. The analysis of cavities in multi-chains PDB files produces noise from clefts formed at contact interfaces from different protein chains.[2] Frequently, the presence of more than one chain in PDB files comes from the packing of more than one protein chains in the crystal unit and does not account for productive interchain contacts responsible for the protein's activity. Thus, by default, the longest protein chain and the ones in contact with it, *viz.,* chains with at least 75 atomic interchain distances below 5.0 Å with the longest chain are selected for further analysis. Other options include selecting only the longest protein chain, one or more explicit user-specified chain, and all protein chains. All aforementioned parameters are accessible and modifiable via options (https://jr-marchand.github.io/caviar/advanced-use/configuration).

Parameters optimization for cavity identification. Many parameters are defined in our cavity detection algorithm and are accessible via the command line tool. We optimized the default settings to give the best performance on a hand curated data set of high quality protein cavities (Supporting Information item S1 and table S1). In short, this dataset contains very well defined cavities as well as challenging cases with cavities potentially overspanning or hard to detect. The list of parameter values tested can be found in the Supplementary Material item 2. In total, we tested 190,080 combinations of parameters on a dataset of 106 PDB structures. The score used for optimization consisted of a mixed step function using ligand coverage, *i.e.*, percentage of ligand atoms covered by cavity points, and cavity coverage*, i.e.*, percentage of cavity points covered by ligand atoms. For the ligand coverage, a threshold of 0.66 was used and any number below that threshold returned a value of 0. The same was done for cavity coverage with a threshold of 0.5. The step values of ligand and cavity coverage were then summed up to give the optimization score ($score_{optimization}$, equation S1).

$$score_{optimization} = coverage_{ligand} + coverage_{cavity} \quad \text{(Eq. S1)}$$

with $coverage_{ligand} = 0$ if $coverage_{ligand} < 0.66$,

$coverage_{cavity} = 0$ if $coverage_{cavity} < 0.5$

Pockets in molecular dynamics trajectories. Cavities are identified in each frame of the trajectory and cavity-lining residues are stored in an object. The distance matrix between all cavities is calculated based on the Sørensen–Dice coefficient. The distance is calculated with unique identifiers for residues, *e.g.*, "arginine residue number 459 of chain B", and does not allow for partial matches, *i.e.*, "arginine residue number 459 of chain B" and "arginine residue number 152 of chain A" are dissimilar. Pockets of the same frame are assigned arbitrarily the maximum distance of 1 in order to exclude them from appearing in the same cluster. The matrix of distances undergoes a hierarchical clustering with the average/UPGMA linkage strategy, with a maximum cophenetic distance threshold of 0.4. Clusters with a minimal occupancy within the trajectory of

5% are printed out alongside the name of cluster representative. The cluster representative is determined as the cavity with the smallest average distance to other cavities within its cluster. Clustering relies on SciPy 1.4.1, the linkage strategy, the distance threshold and the minimal occupancy of a cluster can be tuned via the CAVIAR configuration file setup (https://jr-marchand.github.io/caviar/advanced-use/configuration).

Ligandability assessments. The ligandability module contains a machine learning algorithm trained on the non-redundant set of druggable and less druggable binding sites (NRDLD),[3] with the same split between training set and test set as previous studies.[3–5] Among the 113 complexes, 71 proteins binding sites are considered "druggable" and 42 "undruggable". The training set contains 76 entries (48 "druggable", 28 "undruggable") and the test set 37 entries (23 "druggable", 14 "undruggable"). A list of 27 descriptors characterizing the chemical environment, the buriedness, and the size of the cavities was extracted to train the models: cavity size in number of grid points, cavity score, number of residues lining the cavity, median buriedness of the cavity, average buriedness of the cavity, 8^{th} quantile of buriedness of the cavity, percentage of hydrophobic grid points, percentage of polar non charged grid points, percentage of charged grid points, percentage of "other" grid points (*e.g.*, metal), percentage of points having a buriedness of 8, percentage of points having a buriedness of 9, percentage of points having a buriedness of 10, percentage of points having a buriedness of 11, percentage of points having a buriedness of 12, percentage of points having a buriedness of 13, percentage of points having a buriedness of 14, percentage of points having a pharmacophore "aliphatic", percentage of points having a pharmacophore "aromatic", percentage of points having a pharmacophore "hydrogen bond donor", percentage of points having a pharmacophore "hydrogen bond acceptor", percentage of points having a pharmacophore "hydrogen bond doneptor", percentage of points having a pharmacophore "positively charged", percentage of points having a pharmacophore "negatively charged", percentage of points having a pharmacophore "metal", percentage of points having a pharmacophore "S of cysteine", percentage of points having a pharmacophore "N of an histidine ring". The chemical environment is defined by the projection of the pharmacophore type of the protein atoms to their closest cavity grid points.

Fifteen machine learning algorithms were tested and ranked by Matthews correlation coefficient (MCC) between the classifier and the ground truth. Machine learning was performed with scikit-learn (0.22.1). The classifiers are the following ones, with default parameters of scikit-learn 0.22.1 unless specified otherwise: logistic regression, decision tree, random forest, k-nearest neighbors, linear discriminant analysis, the five bayesian methods in the naive Bayes module, the C-support vector class of the support vector machine module (with all the different kernels, and varied gamma), the multi-layer perceptron classifier of the neural network module (max_iter = 1000), Gaussian process (RBF kernel), AdaBoost, and quadratic discriminant analysis.

Visual interface and command line tool

CAVIAR is available both as a GUI, a command line tool and a Python library. The command line tool comes with many options to provide experienced users with batch use and the ability to tune the parameters for cavity searches and descriptors, to tune for particular protein families or types of cavities. Filters can be activated to include/exclude PDB files based on experimental method, resolution, deposition date, PDB version; metal atoms and well-coordinated water molecules can be incorporated or not in the search; the presence of a ligand can be investigated. The parameters are extensively documented on a website (link in the Availability paragraph). The GUI is for initial

investigations, and restricts the options to default values and consists of two windows. The first window relates to cavity identification, in which the user can specify a PDB code to download or a local PDB file, select a protein chain, exclude or not cavities with missing atoms and interchain cavities, whether to open PyMOL (The PyMOL Molecular Graphics System, Version 2.0 Schrödinger, LLC) to visualize the results and choose the automatic coloring scheme according to buriedness, cavity number or pharmacophore type (Fig. S1a). The second window relates to the subcavity decomposition and has similar options (Fig. S1b).

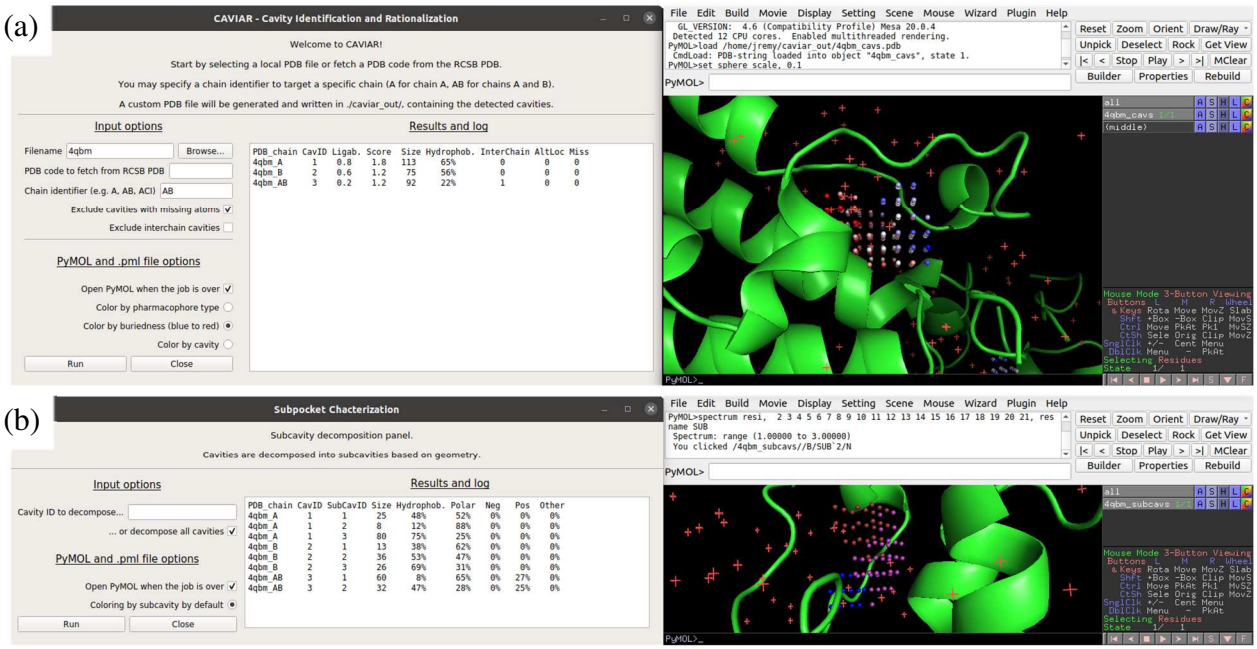


Fig. S1. Visual interface for the CAVIAR cavity detection (a) and subcavity decomposition (b).

Supplementary Material 2 – Parameters optimized for cavity detection.

The following listed parameters relate directly to the cavity identification algorithm and can be specified with the command line tool. Square brackets contain the values that were tested for the optimization; some parameters were not varied because of the exponential computational cost of adding a variable parameter to the list.

- boxmargin: [2.0],
- gridspace: [1.0],
- sizeprobe: [1.0],
- maxdistance: [6.0, 8.0],
- radius_cube: [4, 5],
- min_burial: [8, 9, 10],
- radius_cube_enc: [3, 4],
- min_burial_enc: [6, 7, 8, 9],
- min_degree: [3, 4],
- trim_degree: [2],
- trim_score: [500, 520, 540, 560, 580, 600, 620, 640, 660, 680, 700],
- min_points: [50, 55, 60, 65, 70],
- min_burial_q: [10, 11, 12],
- quantile: [0.6, 0.7, 0.8],
- max_hydrophobicity: [0.7, 0.8]

Table S1. Hand curated dataset for cavity detection parameters optimization.

	PDB	*Lig*	*Name*	*EC number*
	2oyu	IBP	COX1	['1.14.99.1']
	1eqg	CEL	COX1	['1.14.99.1']
	3ln1	FLP	COX2	['1.14.99.1']
	3pgh	DF0	COX2	['1.14.99.1']
	4fm5	EP6	COX2	['1.14.99.1']
	4hra	D16	DOT1L	['2.1.1.43']
	1hvy	PSQ	TYSY	['2.1.1.45']
	1duv	IMG	OTC1	['2.1.3.3']
	1b8n	FMM	PNPH	['2.4.2.1']
	1xkk	MRK	EGFR	['2.7.1.112']
	3f9m	PP2	KINASE-HXK4	['2.7.1.2']
	1qpe	GD9	KINASE-LCK	['2.7.1.112']
	3dbs	NIL	KINASE-PI3K	['2.7.1.153']
	3cs9	AXI	KINASE-ABL1	['2.7.10.2']
	4twp	JIN	KINASE-ABL1	['2.7.10.2']
	2hzi	YAM	KINASE-ABL1	['2.7.10.2']
	3bz3	EUI	KINASE-FAK1	['2.7.10.2']
	4an2	4MK	KINASE-MEK1	['2.7.12.2']
	4mkc	PDS	KINASE-ALK	['2.7.10.1']
	2i0e	SM5	KINASE-PKC	['2.7.11.13']
	3d4q	DHG	KINASE-BRAF	['2.7.11.1']
	5p2p	OAP	PHOSPHOLIPASE-A2	['3.1.1.4']

1kvo	GC8	PHOSPHOLIPASE-A2	['3.1.1.4']
5foq	K27	AchE	['3.1.1.7']
2whr	E20	AchE	['3.1.1.7']
4ey7	F6P	AchE	['3.1.1.7']
1fbp	982	F16P1	['3.1.3.11']
2azr	THP	PTN1	['3.1.3.48']
1snc	CMP	NUC	['3.1.31.1']
2zmf	DAF	PDE10A	['3.1.4.17']
1ppi	GLC	ALPHA-AMYLASE	['3.2.1.1']
1byb	ST1	BETA-AMYLASE	['3.2.1.2']
1ivd	RA2	NEURAMINIDASE	['3.2.1.18']
1b9v	DAN	NEURAMINIDASE	['3.2.1.18']
2sim	VDM	NEURAMINIDASE	['3.2.1.18']
2jf4	MTA	TREA	['3.2.1.28']
1z5o	4BO	MTNN	['3.2.2.9']
3chp	11X	LEUKOTRIENE-A-4-HYDROLASE	['3.3.2.6']
3ftw	4BG	LEUKOTRIENE-A-4-HYDROLASE	['3.3.2.6']
3cho	52D	LEUKOTRIENE-A-4-HYDROLASE	['3.3.2.6']
3ful	80A	LEUKOTRIENE-A-4-HYDROLASE	['3.3.2.6']
3fum	798	LEUKOTRIENE-A-4-HYDROLASE	['3.3.2.6']
3fun	MIT	LEUKOTRIENE-A-4-HYDROLASE	['3.3.2.6']
1dwc	C60	THROMBIN	['3.4.21.5']
1rne	XV6	RENIN	['3.4.23.15']
1bv7	L75	HIV-1-PROTEASE	['3.4.23.16']
1c70	TPV	HIV-1-PROTEASE	['3.4.23.16']
1d4y	A85	HIV-1-PROTEASE	['3.4.23.16']
1dif	KNI	HIV-1-PROTEASE	['3.4.23.16']
1hpx	A77	HIV-1-PROTEASE	['3.4.23.16']
1hvi	017	HIV-1-PROTEASE	['3.4.23.16']
4hla	0GH	HIV-1-PROTEASE	['3.4.23.16']
4gid	ZEB	BETA-SECRETASE 1	['3.4.23.46']
1ctu	BMP	CYTIDINE-DEAMINASE	['3.5.4.5']
1x1z	I7A	PYRF	['4.1.1.23']
2pou	MQ0	CAH2	['4.2.1.1']
3std	CRP	SCYD	['4.2.1.94']
7std	PPG	SCYD	['4.2.1.94']
1m7y	EQU	1A1C	['4.4.1.14']
1ogx	A3S	SDIS	['5.3.3.1']
4rrh	MRC	THREONINE-TRNALIGASE2	['6.1.1.3']
1qu2	MRC	SYI1	['6.1.1.5']
1ffy	8NU	SYI1	['6.1.1.5']
6cm4	ETQ	D2-dopamine-receptor	
3pbl	0HK	D3-dopamine-receptor	
5cxv	0HK	M1-muscarinic-acetylcholine-receptor	
4u15	1KS	M3-mT4L	
4jkv	VDX	SMO	
1ie9	MOF	VitD3R	
1sr7	WOW	PROGESTERONE-RECEPTOR	
3kba	TES	PROGESTERONE-RECEPTOR	
2am9	198	ANDROGEN-RECEPTOR	
4ojb	HFT	ANDROGEN-RECEPTOR	

4oj9	EST	ANDROGEN RECEPTOR
1ere	MOF	ESTROGEN-RECEPTOR
4p6w	DAY	GLUCOCORTICOID-RECEPTOR
3bqd	486	GLUCOCORTICOID-RECEPTOR
5uc3	HCY	GLUCOCORTICOID-RECEPTOR
4p6x	PDN	GLUCOCORTICOID RECEPTOR
2q1v	AS4	ANCCR
2aa2	FLU	MCR
5dyo	COC	IGfab43.1HEAVY
1i7z	AN1	IGkappa
1lo3	KHA	IGkappa
1uwg	H71	IG?
2fwz	OBN	HSP90-ALPHA
3a3y	E55	NAKATPASE
2z65	RTL	TLR4
1rbp	MTB	RET4
1srf	BTN	STREPTAVIDIN
1stp	GAL	STREPTAVIDIN
1gca	HXA	DGAL
1mv9	G24	RXR
1q4x	T4B	THB
2nnq	K30	FABP4
3cjo	ANP	KIF11
2qwr	ATP	HSC70
5f1x	DHM	GRP78
1e00	FMN	OBP
1rcf	VIV	FLAVODOXIN
1oip	BTN	TTPA
2zsc	OHN	TAMAVIDIN2
3ix3	HTF	LASR
3qp2	ENO	CVIR
3ukj	CMP	RHOP2
4d7t	GLA	SPITD
9abp	GLA	ARAF

Supplementary Material 3 – Grid dependency of CAVIAR.

CAVIAR detects cavities with a grid-based algorithm. All grid-based algorithms depend on the orientation of the object in said grid. We investigated if a rotation of the protein influences the performance of CAVIAR on the large validation datasets (Table S1). Proteins were rotated of 30° around the Z axis. No significant difference between the original data and the rotated data emerges.

Table S2. Performance of CAVIAR for cavity identification after a 30° rotation around Z axis. First numbers are original statistics, second numbers are rotated statistics.

	Top 1	***Top 3***	***All***	***Missed***
scPDB	79% / 80%	94% / 94%	99% / 99%	1% / 1%
PDBBind	67% / 66%	84% / 85%	92% / 92%	8% / 8 %
PDBBind high affinity	74% / 76%	90% / 90%	95% / 96%	5% / 4%
Drugs	67% / 68%	83% / 83%	96% / 96%	4% / 4%

Supplementary Material 4 – Details of the 48 bound/unbound dataset

Table S3. Complete description of the performance of CAVIAR cavity identification on the validation set of 48 bound and 48 unbound PDB files.

Bound	***Unbound***	***Observations***	***RMSD****	***Name***
1bid cav 1	3tms cav 1	Very similar cavities	0.24	Thymidylate synthase
1cdo cav 1	8adh cav 1	Very similar cavities	1.48	Alcohol dehydrogenase
1dwd cav 1	1hxf cav 1	Very similar cavities	1.18	Alpha thrombin
1fbp cav 1	2fbp cav 1	Very similar cavities	1.57	Phosphohydrolase
1gca cav 1	1gcg cav 1	Very similar cavities	0.32	Galactose-binding protein
1hew cav 1	1hel cav 1	Very similar cavities	0.21	Acetylchitotriose
1hyt cav 1	1npc cav 1	Smaller cavity in unbound	0.88	Thermolysin
1inc cav 1	1esa fails	Sidechain rearrangement in unbound obstructs cavity	0.23	Elastase
1rbp cav 1	1brq cav 1	Cavities of similar sizes, but shape changes due to sidechain flip	0.62	Retinol binding protein

1rob cav 1	8rat cav 1	Very similar cavities	0.25	Ribonuclease A
1stp cav 1	1swb cav 1	Very similar cavities	0.32	Streptavidin
1ulb cav 1	1ula cav 1	Smaller cavity in unbound	1.20	Purine nucleoside phosphorylase
2ifb cav 1	1ifb cav 1	Very similar cavities	0.35	Fatty acid binding protein
3ptb cav 1	3ptn cav 2	Very similar cavities	0.25	Beta trypsin
2ypi cav 2	1ypi cav 1	Shapes significantly different due to loop rearrangement	1.15	Triose phosphate isomerase
4dfr cav 1	5dfr cav 1	Cavity partially detected in unbound due to loop not resolved	1.30	Dihydrofolate reductase
4phv cav 1	3phv cav 2	Vastly different cavities. Interchain cavities but only one chain in unbound	1.19	HIV 1 protease
5cna fails	2ctv fails	Fails in both, very solvent exposed ligand (sugar)	0.40	Concanavalin A
7cpa cav 1	5cpa cav 1	Very similar cavities	0.43	Carboxypeptidase
1a6w cav 1	1a6u fails	Cavity is already very small in bound, the contraction in unbound makes it go below the size threshold	0.27	FV fragment
1acj cav 1	1qif cav 1	Very similar cavities	0.65	Penicillopepsin
1apu cav 1	3app cav 1	Very similar cavities	0.40	Acetylcholinesterase
1blh cav 1	1djb cav 1	Very similar cavities	0.23	Methylphosphatase
1byb cav 1	1bya cav 1	Cavity larger in unbound due to large loop flip	0.82	Beta amylase
1hfc cav 1	1cge cav 1	Very similar cavities	0.36	Fibroblast collagenase
1ida cav 1	1hsi cav 1	Vastly different cavities due to opening in unbound	1.86	HIV 2 protease
1igj cav 2	1a4j cav 2	Loops movement change the shape of the cavity in unbound	1.08	IGG2A-KAPPA FAB
1imb cav 1	1ime cav 1	Very similar cavities	0.22	Inositol monophosphatase
1ivd cav 1	1nna cav 1	Very similar cavities	1.26	Sialidase
1mrg cav 1	1ahc cav 1	Very similar cavities	0.44	Alpha momorcharin
1mtw cav 1	2tga fails	Very small cavity in bound, below size threshold in unbound	0.42	Trypsin
1okm cav 1	4ca2 cav 1	Very similar cavities	0.22	Carbonic anydrase II
1pdz cav 1	1pdy cav 1	Very similar cavities	0.66	Enolase
1phd cav 1	1phc cav 1	Very similar cavities	0.17	Camphor 5 monooxygenase

1pso cav 1	1psn cav 1	Very similar cavities	0.33	Pepsin 3a
1qpe cav 1	3lck cav 1	Very similar cavities	0.24	Lck kinase
1rne cav 1	1bbs cav 1	Very similar cavities	0.77	Renin
1snc cav 1	1stn cav 1	Smaller cavity in unbound due to loop rearrangement	0.71	Staphylococcal nuclease
1srf cav 1	1pts cav 1	Very similar cavities	0.30	Streptavidin
2ctc cav 1	2ctb cav 1	Very similar cavities	0.15	Carboxypeptidase
2h4n cav 1	2cba cav 1	Very similar cavities	0.20	Carbonic anhydrase II
2pk4 fails	1krn fails	Fails in both cases due to very exposed and flat surface of interaction	0.31	Plasminogen kringle
2sim cav 2	2sil cav 2	Very similar cavities	0.14	Sialidase
2tmn cav 1	1l3f cav 1	Smaller in unbound, side chain rearrangement cuts the cavity	0.62	Thermolysin
3gch cav 1	1chg fails	Fails in unbound, loop occupies the cavity	1.79	Gamma chymotrypsin
3mth fails	6ins fails	Fails in both cases due to very exposed and flat surface of interaction		Methylparaben insulin
5p2p cav 1	3p2p cav 1	Smaller in unbound due to side chain rearrangement	0.41	Phospholipase
6rsa cav 1	7rat cav 1	Very similar cavities	0.18	Ribonuclease

* RMSD values were calculated in pymol 2.3.2 with the alignto command, matching pairs of Cα for automatic alignment.

Supplementary Material 5 – Ligandability module performances

We implemented a k-nearest neighbors (k-NN) algorithm in order to provide the user with a quick estimation of the ligandability of the detected cavities. This method was trained on the same dataset, with the same train/test sets as previously published methods,[3–7] with 27 descriptors extracted from our cavity detection algorithm. The k-NN method came out as the best classifier among the fifteen supervised learning algorithms we evaluated, and performs similarly to existing methods (Tables S4-5). In details, the k-NN algorithm had the following confusion-matrix based scores on the test set: MCC = 0.73, accuracy = 0.86, recall = 0.86, precision = 0.89, f1-score = 0.86. Of note, a dummy classifier of scikit-learn 0.22.1 gives the following results: MCC = -0.01, accuracy = 0.49, recall = 0.51, precision = 0.53, f1-score = 0.52, highlighting the interest of the Mathews correlation coefficient with such a dataset.

Table S4. Matthews correlation coefficient (MCC) and accuracy of five software for the prediction of ligandability of cavities.

	k-NN (CAVIAR)	***VolSite***	***DrugPred***	***PockDrug***	***Fpocket***	***SiteMap***
MCC	0.73	0.77	0.77	0.54	0.39	0.24
Accuracy	0.86	0.89	0.89	0.76	0.73	0.65

Values for VolSite [4], DrugPred [3], Fpocket [8] and SiteMap [7] are extracted from [4], values for PockDrug from [5].

In details, the ligandability module of CAVIAR correctly predicts all of the 23 “druggable” structures as ligandable, with four of them bearing a value of 0.6, which indicates low confidence in the prediction (Table S5). Interestingly, among these four cases, three are incorrectly predicted as non ligandable by at least one of the other four ligandability assessment software, including one wrongly assigned by all of the other tools as non ligandable. The prediction of poorly ligandable targets turns out to be a trickier exercise. CAVIAR mispredicts five of the fourteen “unligandable” targets as ligandable. Four of these five cases are liganded cavities, only one being apo. Our goal with this module is to give the user a quick idea of a cavity’s ligandability with a simple method. Values are discrete between 0 and 1 with a step of 0.2. We recommend to consider ligandability values for a given cavity of 0.8 and 1.0 as probably ligandable, 0.4 and 0.6 as inconclusive, and values of 0.2 and 0 as possibly very difficult to design a ligand for.

Table S5. Detailed scores of CAVIAR ligandability module on the test set.

PDB	*Name*	*k-NN (CAVIAR)*	*DrugPred [3]*	*SiteMap [7]*	*Fpocket [6]*	*VolSite [4]*
druggable						
1e66	acetylcholinesterase	1.0	0.81	1.14	0.75	0.8
1fk9	HIV reverse transcriptase	0.8	0.79	1.27	0.84	1.07
1kzn	DNA gyrase	0.8	0.81	1.04	0.75	-0.48
1lox	15-lipoxygenase	0.8	1.15	1.13	0.76	1.01
1oq5	carbonic anhydrase II	0.6	0.77	1	0.1	0.48
1owe	urokinase plasminogen activator	0.8	0.4	0.93	0.25	0.28
1pmn	C-Jun kinase	0.8	0.93	1.09	0.88	1.25
1pwm	aldose reductase	1.0	0.94	0.97	0.86	0.67
1q41	glycogen synthase kinase 3	0.8	0.55	1.09	0.46	1.15
1r55	ADAM33	0.6	0.69	0.89	0.08	0.07
1sqn	progesterone receptor	0.8	1.11	1.28	0.95	1.99
1t46	c-Kit kinase	1.0	1.17	1.12	0.84	1.37
1unl	cyclin-dependent kinase 5	1.0	0.56	1.06	0.12	0.47
1uou	thymidine phosphorylase	0.6	0.28	ND	0.4	-0.65
1xoz	phosphodiesterase 5A	1.0	1.14	1.1	0.81	1.06
2aa2	mineralocorticoid receptor	0.8	1.02	1.24	0.92	1.16
2cl5	catechol-O-methyltransferase	0.8	0.82	1.19	0.7	1.48
2i1m	FMS kinase	1.0	0.74	1.1	0.82	0.7
3b68	androgen receptor	0.8	1.13	1.29	0.95	2.09
3etr	xanthine oxidase	0.6	0.85	1.13	0.67	1.01
3f0r	histone deacetylase 8	0.8	0.89	1.12	0.59	0.96
3f1q	dihydroorotate dihydrogenase	0.8	1.15	1.2	0.9	1.73
3ia4	dihydrofolate reductase	0.8	0.79	1.07	0.65	0.63
undruggable						
1ajs	aspartate aminotransferase	0.0	0.49	1.14	0.6	-0.6
1b74	glutamate racemase	0.8	0.41	1.05	0.56	0.6
1bls	Beta-lactamase	1.0	0.34	1.04	0.26	-0.01
1bmq	interleukin-1beta-converting enzyme	0.0*	0.38	0.79	0.01	-0.05
1ec9	D-glucarate dehydratase	0.0*	-0.31	1.03	0.15	-1.28
1g98	phosphoglucose isomerase	0.0	0.09	1.14	0.03	-0.57
1kc7	pyruvate phosphate dikinase	0.8	0.01	0.86	0.01	-1.61
1m0n	dialkylgycine decarboxylase	0.4	0.5	0.98	0.76	0.46
1mai	phospholipase C	0.0	0.09	0.9	0.03	-1.93
1od8	xylanase	0.2	0.06	0.79	0.05	-1.01
1px4	beta-galactosidase	0.6	0.55	1.06	0.13	-0.03
1v16	α-keto acid dehydrogenase	0.4	0.41	1.08	0.02	-0.57
1wvc	CDP-D-glucose synthase	0.8	0.65	1.03	0.67	-0.52
3jdw	L-arginine:glycine amidinotransferase	0.2	0.17	1.06	0.09	-0.18

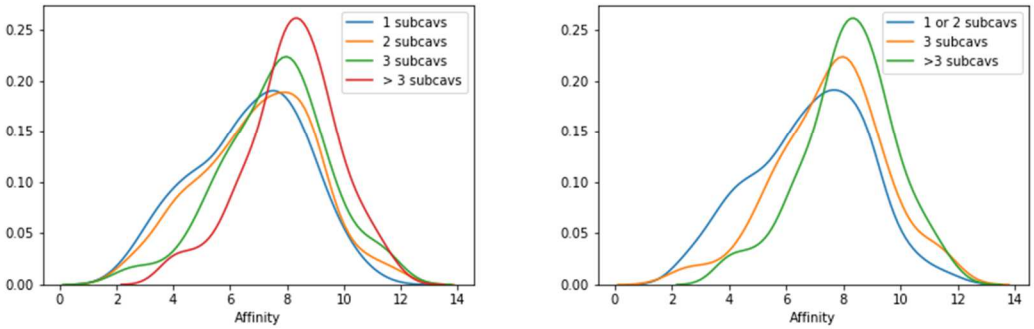
Red values indicate misclassification, yellow values are inconclusive. Values and ligandability thresholds of other software are extracted from [[4]]. *No cavity detected, value assigned to 0.0.

Supplementary Material 6 – Additional statistics on the PDB, scPDB and PDBBind; Kolmogorov-Smirnov significance values

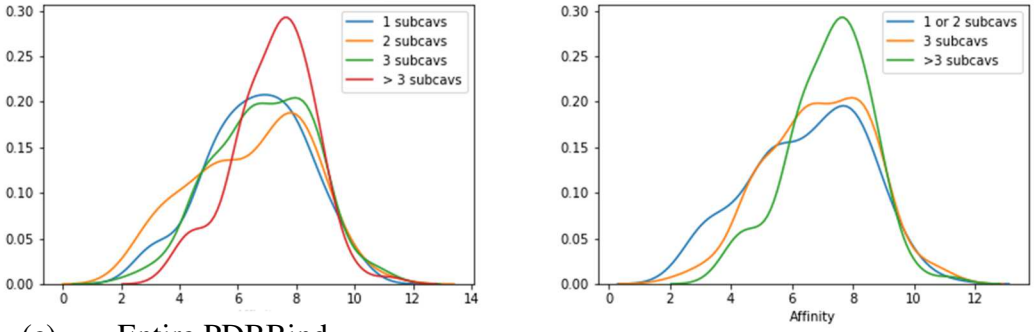
Table S6. Additional statistics on the PDB, scPDB and PDBBind.

	Entire PDB		***scPDB***		***PDBBind***	
	avg ± std	median	avg ± std	median	avg ± std	median
N cavities	8.3 ± 11.6	5	8.0 ± 11.6	5	7.0 ± 10.7	4
Cavity size (Å^3)	180 ± 270	94	201 ± 282	99	172 ± 258	95
Size holo cavities (Å^3)	353 ± 423	238	473 ± 416	385	309 ± 372	219
Size apo cavities (Å^3)	145 ± 208	83	140 ± 195	82	134 ± 199	79
Hydrophobicity	40 ± 17%	39%	42 ± 17%	41%	40 ± 17%	39%
Hydrophobicity holo cavities	45 ± 17%	43%	49 ± 14%	48%	46 ± 16%	43%
Hydrophobicity apo cavities	39 ± 17%	38%	40 ± 18%	39%	39 ± 17%	37%
Ligandability holo cavities	0.62 ± 0.27	0.6	0.73 ± 0.23	0.8	0.65 ± 0.26	0.6
Ligandability apo cavities	0.51 ± 0.26	0.4	0.52 ± 0.27	0.6	0.5 ± 0.26	0.4
N subcavities holo cavities	4.4 ± 4.6	3	5.3 ± 4.5	4	3.7 ± 3.8	3
N subcavities apo cavities						
Ligand coverage	79 ± 25%	88%	81 ± 23%	91%	78 ± 23%	87%
Cavity coverage	60 ± 31%	62%	65 ± 26%	67%	74 ± 26%	82%
N subcavs covered by ligand	2.5 ± 1.5	2	2.8 ± 1.4	3	2.3 ± 1.2	2

(a) Proteases



(b) Kinases



(c) Entire PDBBind

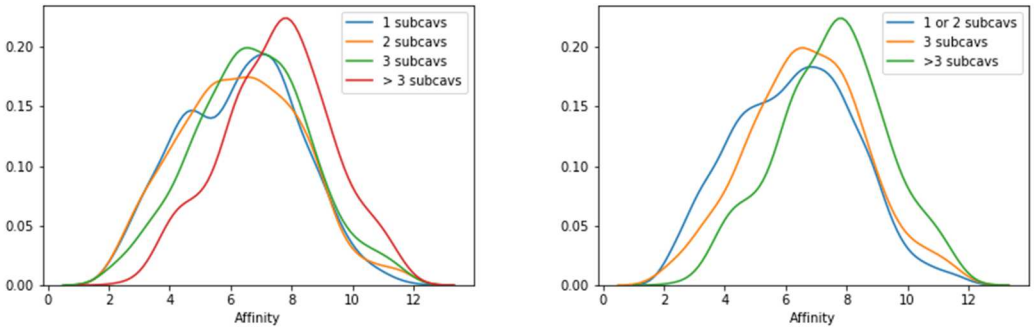


Fig. S2. Additional information on the occurrence of binding affinities, colored by the number of subcavities filled by the ligand. (a) Proteases, (b) Kinases and (c) entire PDBBind.

Table S7. Kolmogorov-Smirnov significance values.

Compared datasets	*D statistics*	*P value*	*Samples sizes*
All PDB			
Size holo vs apo cavities	0.41	$< 10^{-308}$	10^5 vs 10^5
Hydrophobicity holo vs apo cavities	0.15	$< 10^{-308}$	10^5 vs 10^5
Ligandability holo vs apo cavities	0.17	$< 10^{-308}$	10^5 vs 10^5
Count of subcavities holo vs apo cavities	0.33	$< 10^{-308}$	10^5 vs 10^5
scPDB			
Size holo vs apo cavities	0.67	$< 10^{-308}$	10^4 vs 10^4
Hydrophobicity holo vs apo cavities	0.26	$< 10^{-308}$	10^4 vs 10^4
Ligandability holo vs apo cavities	0.34	$< 10^{-308}$	10^4 vs 10^4
Count of subcavities holo vs apo cavities	0.51	$< 10^{-308}$	10^4 vs 10^4
PDBBind			
Size holo vs apo cavities	0.46	$< 10^{-308}$	10^3 vs 10^4
Hydrophobicity holo vs apo cavities	0.20	10^{-178}	10^3 vs 10^4
Ligandability holo vs apo cavities	0.25	10^{-260}	10^3 vs 10^4
Count of subcavities holo vs apo cavities	0.31	$< 10^{-308}$	10^3 vs 10^4
Proteases			
Ligand in 1 subcav vs ligand in >3 subcavs	0.34	10^{-11}	207 vs 194
Ligand in 2 subcav vs ligand in >3 subcavs	0.28	10^{-8}	246 vs 194
Ligand in 3 subcav vs ligand in >3 subcavs	0.18	10^{-3}	154 vs 194
Ligand in 1 or 2 subcav vs ligand in >3 subcavs	0.31	10^{-12}	453 vs 194
Ligand in 1, 2 or 3 subcav vs ligand in >3 subcavs	0.27	10^{-10}	607 vs 194
Kinases			
Ligand in 1 subcav vs ligand in >3 subcavs	0.21	0.03	88 vs 103
Ligand in 2 subcav vs ligand in >3 subcavs	0.27	10^{-4}	161 vs 103
Ligand in 3 subcav vs ligand in >3 subcavs	0.14	0.17	122 vs 103
Ligand in 1 or 2 subcav vs ligand in >3 subcavs	0.25	10^{-4}	249 vs 103
Ligand in 1, 2 or 3 subcav vs ligand in >3 subcavs	0.21	10^{-3}	371 vs 103
Entire PDBBind			
Ligand in 1 subcav vs ligand in >3 subcavs	0.25	10^{-22}	1177 vs 579
Ligand in 2 subcav vs ligand in >3 subcavs	0.26	10^{-24}	1279 vs 579
Ligand in 3 subcav vs ligand in >3 subcavs	0.20	10^{-12}	800 vs 579
Ligand in 1 or 2 subcav vs ligand in >3 subcavs	0.25	10^{-25}	2456 vs 579
Ligand in 1, 2 or 3 subcav vs ligand in >3 subcavs	0.24	10^{-24}	3256 vs 579

In black, datasets for which we can conclude from the KS test that the two distributions compared are not drawn from the same sample with an α value of 1%. In orange, the case for which the certainty is between 1 and 5%. In red, the case in which we cannot exclude that the samples are similar. Cf Massey, F. J. The Kolmogorov-Smirnov Test for Goodness of Fit, *Journal of the American Statistical Association*, **1951**, 46 (253), 68—78.

Supplementary references

1. Bakan A, Meireles LM, Bahar I (2011) ProDy: Protein Dynamics Inferred from Theory and Experiments. Bioinformatics 27:1575–1577. https://doi.org/10.1093/bioinformatics/btr168

2. Weisel M, Proschak E, Kriegl JM, Schneider G (2009) Form follows function: Shape analysis of protein cavities for receptor-based drug design. PROTEOMICS 9:451–459. https://doi.org/10.1002/pmic.200800092

3. Krasowski A, Muthas D, Sarkar A, et al (2011) DrugPred: A Structure-Based Approach To Predict Protein Druggability Developed Using an Extensive Nonredundant Data Set. J Chem Inf Model 51:2829–2842. https://doi.org/10.1021/ci200266d

4. Desaphy J, Azdimousa K, Kellenberger E, Rognan D (2012) Comparison and Druggability Prediction of Protein–Ligand Binding Sites from Pharmacophore-Annotated Cavity Shapes. J Chem Inf Model 52:2287–2299. https://doi.org/10.1021/ci300184x

5. Borrel A, Regad L, Xhaard H, et al (2015) PockDrug: A Model for Predicting Pocket Druggability That Overcomes Pocket Estimation Uncertainties. J Chem Inf Model 55:882–895. https://doi.org/10.1021/ci5006004

6. Le Guilloux V, Schmidtke P, Tuffery P (2009) Fpocket: An open source platform for ligand pocket detection. BMC Bioinformatics 10:168. https://doi.org/10.1186/1471-2105-10-168

7. Halgren TA (2009) Identifying and Characterizing Binding Sites and Assessing Druggability. J Chem Inf Model 49:377–389. https://doi.org/10.1021/ci800324m

8. Schmidtke P, Barril X (2010) Understanding and Predicting Druggability. A High-Throughput Method for Detection of Drug Binding Sites. J Med Chem 53:5858–5867. https://doi.org/10.1021/jm100574m