Automatic cavity identification and decomposition into subpockets with CAVIAR

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

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

ABSTRACT

Motivation: The detection of small molecules binding sites in proteins is central to structure based drug design. Many tools were developed in the last 40 years, but only few of them are available today, open-source, and suitable for the analysis of large databases or for the integration in automatic workflows. In addition, no software can characterize subpockets solely with the information of the protein structure, a pivotal concept in fragment-based drug design.
Results: CAVIAR is a new open source tool for protein cavity identification and rationalization. Protein pockets are automatically detected based on the protein structure. It comprises a subcavity segmentation algorithm that decomposes binding sites into subpockets without requiring the presence of a ligand. The defined subpockets mimick the empirical definitions of subpockets in medicinal chemistry projects. A tool like CAVIAR may be valuable to support chemical biology, medicinal chemistry and ligand identification efforts. Our analysis of the entire PDB and the PDBBind confirms that liganded cavities tend to be bigger, more hydrophobic and more complex than apo cavities. Moreover, in line with the paradigm of fragment-based drug design, the binding affinity scales relatively well with the number of subcavities filled by the ligand. Compounds binding to more than three of the subcavities identified by CAVIAR are mostly in the nanomolar or better range of affinities to their target.
Availability and implementation: Installation notes, user manual and support for CAVIAR are available at https://jr-marchand.github.io/caviar/. The CAVIAR GUI and CAVIAR command line tool are available on GitHub at https://github.com/jr-marchand/caviar and the package is hosted on Anaconda cloud at https://anaconda.org/jr-marchand/caviar under a MIT license. The GitHub repository also hosts the validation datasets.
Contact: jean-remy.marchand@novartis.com; finton.sirockin@novartis.com

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 presents 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. Binding pockets can be characterized empirically by analyzing holo structures of the target in complex with a ligand, but the analysis of the entire PDB, including structures without ligand, requires automatic algorithms to perform that task. The cavity detection field has been prolific in the last three decades,[2–4] with some successful applications for the prediction of target ligandability,[5–9] identification of off-targets,[10–14] functional annotation,[15–18] 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. As such, they can result in very valuable information for medicinal chemistry, but may require a careful preparation of the protein and are inherently computationally intensive.[25–30] Geometry-based methods are less resource demanding and potentially more resilient to small changes in the pocket, which gives them a different scope as they can be applied on large scales. They detect cavities 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] Cavities are generally defined as clefts on the surface of the protein. A variety of geometry-based methods for pocket detection has 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]

Table 1. Main software 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*,[39] PocketPicker,[23] McVol*,[40] VICE,[41] VolSite,[9] SiteMap[7]
Space filling	Spheres are placed around the protein surface to detect empty spaces in the protein convex hull	SURFNET,[42] PASS,[43] PHECOM*,[44] KVFinder*,[45] GHECOM*,[46] SCREEN,[8] POCASA[47]
Voronoi diagram	The Voronoi decomposition of the space of protein atoms serves as basis to identify clefts	FindSurf,[48] CAST,[49] APROPOS;[50] Fpocket*,[6] SiteFinder (MOE)
Imaging science	Gaussian surfaces approximate the protein shape	DoGSite,[24] CavVis*[51]

* indicates open-source software available at the time of the study.

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 around 80 to 90%.[4] However, many programs are either available as closed-source commercial packages/webservers or unavailable. In order to enable the development of novel cavity comparison tools, we developed a comprehensive ligand-agnostic Python-based open-source platform for cavity detection and characterization, usable with a graphical user interface (GUI) and/or advanced command line tool. We started from a concept of enclosure of grid points algorithm and augmented them with novel ideas to refine the resulting cavities, *e.g.*, double pass to estimate buriedness, trimming of spurious points and exclusion of loosely connected nodes, size and hydrophobicity filters. In addition, we addressed blind spots of existing algorithms, such as

treatment of metals and structural waters in binding pockets, detection of cavities at the interface of multi-chain proteins or multi-domain complexes, and decomposition of binding cavities into meaningful subpockets. Cavity segmentation into subcavities is crucial in the era of structure-based drug discovery, with medicinal chemists attempting to improve potency and selectivity of their hits by filling protein subpockets.[29,52–55] Similar proteins may have binding pockets with different subcavities and dissimilar proteins may have conserved subcavities. Many drug targets exhibit well-defined subpockets geometrically, such as proteases, kinases and GPCRs, which are used extensively in order to develop selective compouds.[53] In addition, two independent studies concluded that drug-like ligands typically occupy about a third of their binding pockets, 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. DoGSite was developed as a ligand-agnostic cavity identification tool, borrowing concepts from the computational image recognition field.[24] In short, small hotspots are identified with a difference of Gaussians algorithm and then inflated to merge them into cavities. Original hotspots can be treated as subpockets without further processing. However, DoGSite is not open source and the validation of the pre-merging hotspots discovered by DoGSite for the characterization of subpockets have been discontinued, with further work defining selectivity subpockets as grid differences between pockets of distinct kinases.[64] The pertinence of the subsite decomposition produced by CAVIAR was assessed qualitatively and put in parallel with DoGSite's results. Finally, we performed an analysis of the PDB to characterize

differences between liganded and apo binding pockets, as well as an analysis of the binding affinities of ligands with regards to the number of subpockets they fill.

MATERIAL AND 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.[65] 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.[66] 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.

Cavity identification. The selected 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 around 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 a protein type. Remaining grid points are considered as solvent grid points and

investigated further (Figure 1, panel A). For each solvent grid point, the fourteen cubic directions, *i.e.*, the three axis and the four cubic diagonals in both positive and negative directions, are investigated for contacts with protein grid points. For each direction, if a protein grid point is encountered within four grid spacings, *viz.,* 4 Å for the three axis and 6.9 Å for the cubic diagonals (grid spacing of 1 Å), a counter is incremented. The final number for a grid point is comprised between 0 and 14, and represents the "buriedness" of a solvent grid point (Figure 1, panel B). 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 a second time. The second pass is similar to the first one, except that solvent grid points are investigated to be in vicinity (three grid spacings) of the previously defined cavity grid points. 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 (Figure 1, panel C).

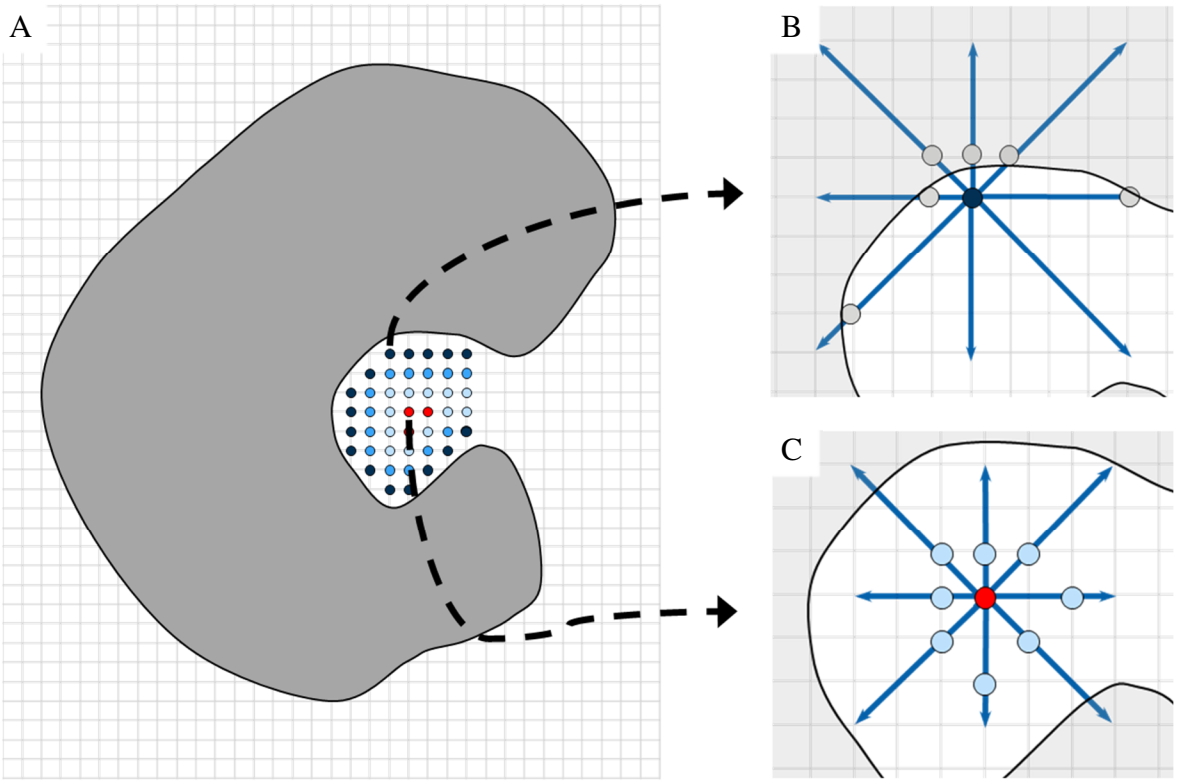


Figure 1. Grid-based cavity identification. (A) The protein, represented as a gray shape, is embedded in a grid. (B) The number of contacts between grid points outside of the protein surface and grid points inside the protein surface is investigated and defines putative cavity grid points. (C) A second pass detects grid points enclosed in putative grid points that would have been missed in B.

One of the risks associated with a grid based algorithm is cavity overspanning, *i.e.*, to favor very large cavities overflowing at the surface of the protein and have cavity grid points connecting cavities that should not be connected (Figure 2). To circumvent this, we developed a metrics 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) corresponding to how mingled a cavity grid point is, in a set of cavity grid points. Points with a trim score below 500 are trimmed out.

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

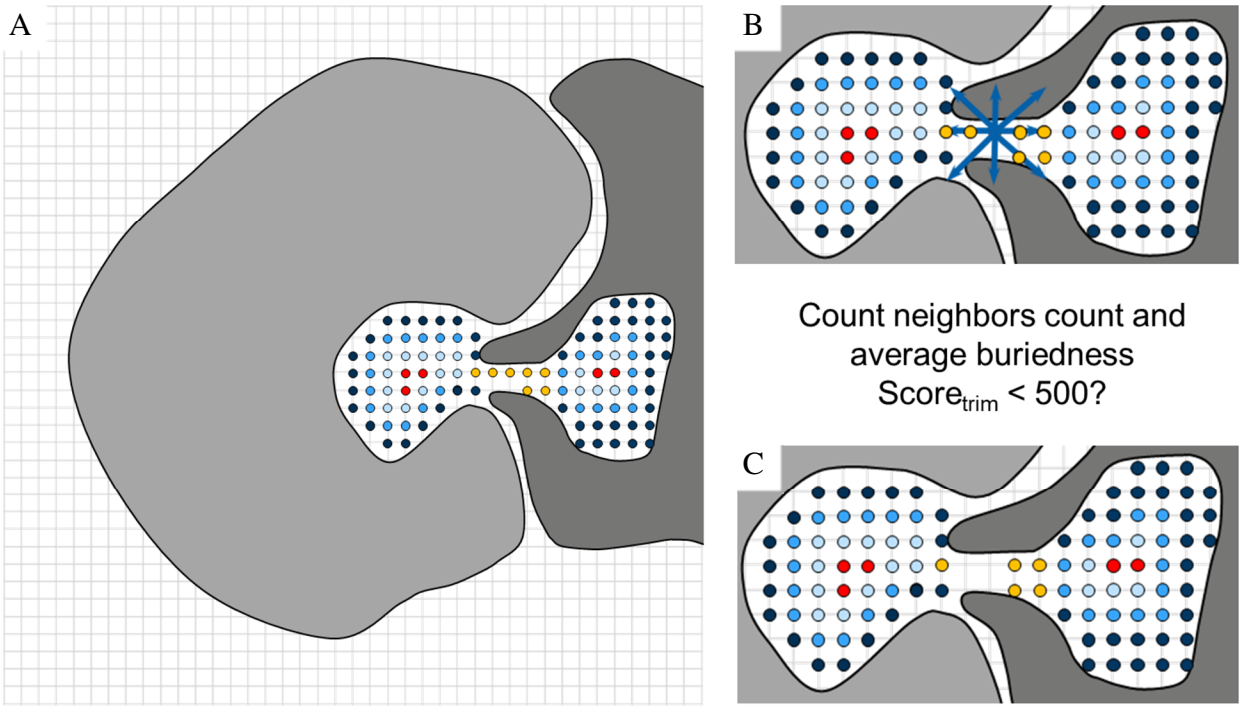


Figure 2. Cavity overspanning and trimming of spurious cavity points. (A) Yellow points are connecting a cavity in the light gray protein chain and another one in the dark gray chain. (B) For each cavity point, the count of neighbors and the average buriedness are measured in order to calculate the trim score. (C) Grid points with a trim score below 500 are eliminated from the cavity grid points set.

Putative grid points are embedded in a graph, where edges are built around adjacent grid points 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 pseudotypes according to the pharmacophore type of the closest atom: hydrophobic (aliphatic and aromatic), polar non charged (hydrogen bond donor/acceptor), negative (charged group of Asp/Glu), positive (charged group of Lys/Arg), other (S atom of Cys, ring of His, metal ion). Some properties are calculated and stored, *e.g.,* hydrophobicity, 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 a 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 they are overly exposed. 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.

Computationally intensive calculations are performed with NumPy (1.17.3) and SciPy (1.4.1), and therefore benefit from the performance and optimization of these packages. Graph methods rely on the NetworkX (2.4) library.

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 Supporting Information item 1. 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 3).

$$score_{optimization} = coverage_{ligand} + coverage_{cavity} \quad \text{(Eq. 3)}$$

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

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

Validation sets for cavity identification. We assembled different datasets extracted from literature sources, *i.e.*, Kahraman *et al*,[57] Huang and Schroeder,[33] the 198 drug-target set of MetaPocket,[36] the DUD-e 102 targets;[67] databases, *i.e.*, scPDB[68] and PDBbind;[69] as well as our own compiled datasets, *i.e.*, 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 Notes section).

These datasets vary by size, *viz.*, from few dozens in the literature sets to more than 11,000 in scPDB database, and in their scope and composition. The use of multiple datasets aimed at detecting any particularity 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 "MetaPocket" dataset is supposed to contain 198 drug targets, but we have 196 PDB in our "MetaPocket" validation set. Two of the structures in the original dataset were 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 discrepancies in numbers. Success in cavity identification is defined by the overlap between cavity points and ligand atoms within 1 Å. The direct comparison with other algorithms is performed on Huang and Schroeder's dataset[33] and defines success by the presence of a ligand atom within 4 Å of the geometric center of the cavity, in order to allow for direct comparison with the literature.

Ligandability assessments. The ligandability module contains a machine learning algorithm trained on the non-redundant set of druggable and less druggable binding sites (NRDLD),[70] with the same split between training set and test set as previous studies.[9,70,71] 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 (Supporting Information item S4). 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 between the classifier and the ground truth (Supporting Information table S4 and item S5). Machine learning was performed with scikit-learn (0.22.1).

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. 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,[72] which segments cavities into subgroups. Seed points are separated by at least 3 Å to each other, in order 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 (Figure 3).

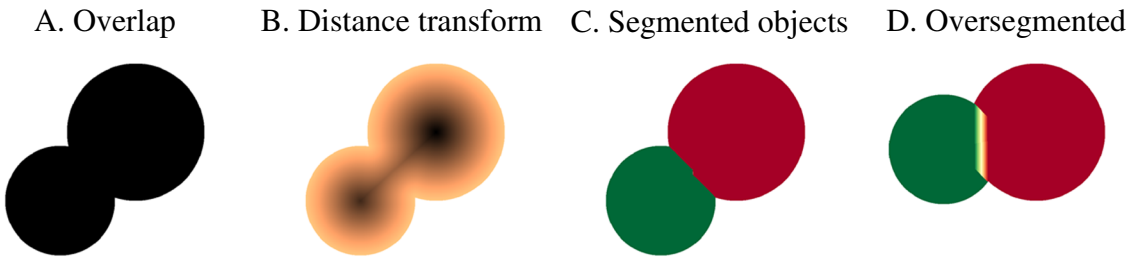


Figure 3. 2D representation of the watershed algorithm. (A) Two overlapping circles, *e.g.*, a cavity that we seek to segment. (B) The local topography of the image is defined by an Euclidean distance transform of the original image. The darkest points are the most distant points to the image boundary. (C) Segmented image, with two objects, one in green and one in red, being separated after applying the watershed algorithm. (D) Different 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.

The watershed algorithm tends to over-segment images.[72] 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 change significantly the results. We therefore developed a rationale to merge small "spurious" subcavities with their largest neighbor. First, we detect subcavities of size smaller than 50 grid points. For each small subcavity, direct contacts, *i.e.*, at 1 Å, with other subcavities are calculated. If more than two thirds of the small subcavity grid points are in contact with neighboring subcavities, we consider that this subcavity is most likely noise and we add it to the subcavity it has the most contacts with. In most cases, subcavities filling these two criteria are interstitial and disk-shaped between several subcavities, or extended and laying on top of another subcavity (Figure 4). Image segmentation routines are performed with scikit-image (0.16.2).

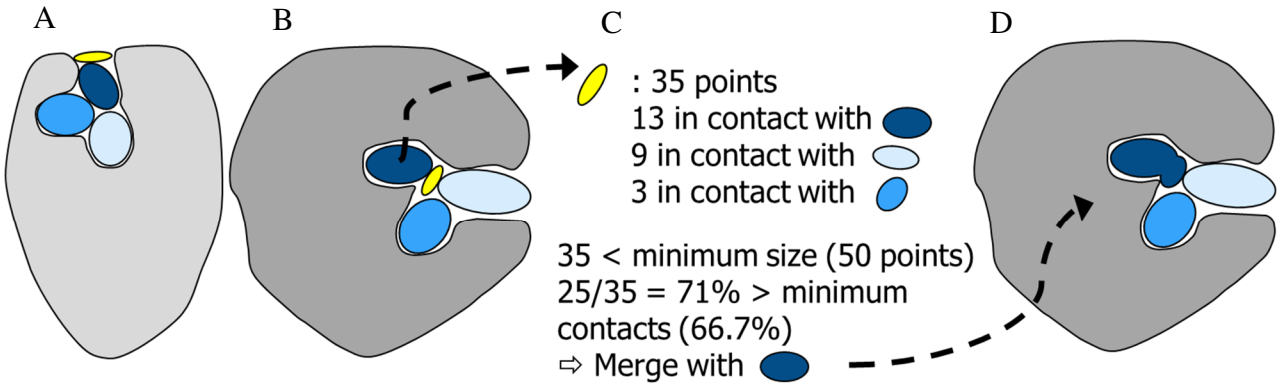


Figure 4. Merging spurious subcavities. (A) and (B) are two examples of cavity oversegmentation. In some cases, flat subcavities are created at the surface (A), and sometimes they are generated in between other main subcavities (B). (C) Summary of the rationale to detect potentially spurious

subcavities and identify its merging partner. (D) Result of (C) on (B). In both (A) and (B), the yellow subcavity is merged with the dark blue one.

Validation sets for subcavity analysis. A carefully hand-picked dataset of 59 proteins for which subcavities can be defined with a high level of confidence, based on experimental knowledge, was assembled. This dataset contains 17 protease structures, which are a gold standard of proteins with binding pockets divided in precisely defined subpockets. In addition, we compiled 13 structures of GPCRs, 5 of bromodomains, 5 of kinases, 2 of acetylcholine esterases, 3 of 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, and results can be found in Table 2. 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 accross all datasets except the MetaPocket dataset that plateaus at 81%. The MetaPocket dataset is enriched in very solvent-exposed ligand-protein complexes, with a flat 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 certain number of noisy PDB structures. For example, we noticed several cases of wrong ligand identifier (*e.g.*, a cosolvent instead of the ligand-like compound) in the scPDB, PDBBind and MetaPocket datasets, which we corrected, but non-exhaustively. The hand validation of all of these structures is beyond the scope of this work. It is interesting to note that if we restrict the PDBBind dataset to high affinity complexes with a micromolar affinity or better, the success in identifying binding pockets raises in comparison to the whole PDBBind dataset (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%**

Success percentages are defined as 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 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] to compare the performance of CAVIAR to state of art cavity identification software (Table 3). Overall, CAVIAR performs well both on the 48 unbound structures and the 48 bound structures, with a success 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 occurrences in the bound dataset, all three are very exposed ligand on flat surfaces of the protein (Supporting Information 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%**

Values of all algorithms except CAVIAR were extracted from [24]. CAVIAR's success values were calculated with the definition of [24].

Cavities are ranked according to their cavity scores, which is a raw estimate of a cavity's interest and relies only on size and buriedness. This score was not developed with the intention to order cavities with regards to their ligandability but rather to have an heuristic to limit the number of stored cavities in the case of a large scale analysis of the PDB. In addition, it may be unsettling for the user to get an unordered list of results. Interface cavities between protein chains are often big and will therefore have higher scores than small voids inside a protein chain. 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 the ligand, which can place the liganded cavity second or third instead of first rank. For these reasons, the top 1 and top 3 values in tables 1 and 2 are underestimates of the quality of the cavity detection algorithm. 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.

Performance of the k-NN classifier for ligandability predictions. 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,[6,7,9,70,71] 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 (Table 4 and Supporting Information table S4 and item S5).

Table 4. 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,[70] Fpocket,[73] and SiteMap[7] are extracted from [9], values for PockDrug from [71].

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 (Supporting Information Table S4). 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.

Subcavity segmentation. We assembled a dataset of 59 diverse proteins to judge qualitatively the performance of 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 entirety of the cavity is not detected, or on the contrary, spurious subcavities are present when the cavity overspans. In spite of our merging rationale, the decomposition algorithm tends to oversegment cavities. We discuss here four cases of what we think are successful cases of cavity segmentation with CAVIAR (Figure 5), two cases of failures (Figure 6) and compare them to DoGSite default output. These six examples were selected with respect to CAVIAR, not a consensus of CAVIAR and DoGSite, which was run separately, and may not represent an accurate depiction of DoGSite's performance. The rest of the results is available on the GitHub repository of the CAVIAR package (link in the Notes section).

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, mostly utilized by small molecule inhibitors to improve their selectivity profiles.[74,75] CAVIAR identifies correctly 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 (Figure 5A). DoGSite recognizes the two subpockets, but produces a larger result and generates four subcavities in total (Figure 5B). The second example is the HIV-1 protease, which contains six well defined subsites, recognizing specifically aminoacid side chains of the target peptide to cleave.[76,77] CAVIAR generates seven subcavities, six of which corresponds to the six landmark subsites S1 to S3 and S1' to S3'. The S1 subsite is segmented into two subcavities, which correspond in the PDB of our experiment to the piperazine and the benzofurane

groups of the ligand (Figure 5C, chemical groups in magenta and dark blue), and in the literature to the S1 and extended S1 pockets.[76] DoGSite, on the other hand, correctly predicts the binding pocket, but fails to decompose it into subpockets, *i.e.*, outputs only one single pocket (Figure 5D). Our third example is the GPCR of the M1 muscarinic acetylcholine receptor bound to 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 (Figure 5E). Both in CAVIAR and DoGSite, 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 thiophene moieties of the inhibitor.[78] DoGSite results are similar to CAVIAR, except that it does not segment the orthosteric pocket into subsites (Figure 5F). The last successful case discussed here is the EGFR kinase domain bound to lapatinib, for which CAVIAR detects six subpockets (Figure 5G). The main binding region of ATP, *i.e.*, the adenine, sugar and phosphates regions, is described by one large subpocket, occupied by the main hinge binding motif of the ligand and its furan attachment.[79] 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 overlapped by any cavity grid point. The cavity from DoGSite overlaps similarly with the ligand, but does not decompose the pocket into subcomponents. On the contrary, it detects other connected subpockets far from the ligand binding groove, and significantly overspans (Figure 5H).

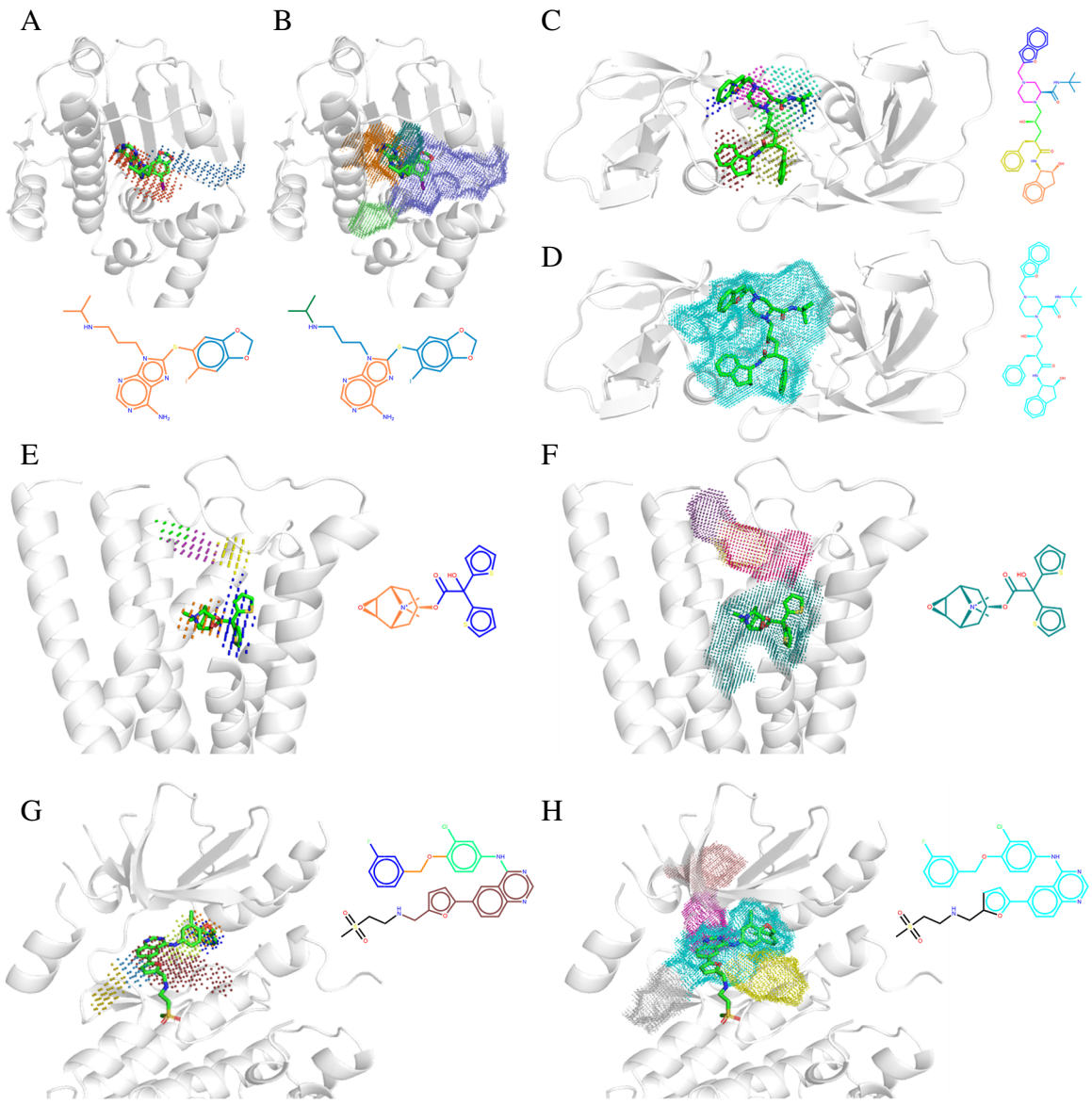


Figure 5. Examples of successful decomposition by CAVIAR and comparison to DoGSite. In all panels, the 2D structure of the ligand is depicted with a color code corresponding to the subcavity segmentation, or in black if not covered by subcavities. (A) and (B) Chaperone protein hsp90, PDB code 2fwz. (A) The CAVIAR subpocket algorithm correctly identifies the adenine pocket, in orange and the lipophilic pocket, in blue. (B) DoGSite also 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 successfully identifies the pocket, but fails to segment it 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). The allosteric cavity is connected to the orthosteric one, and is decomposed into three subpockets. (F) DoGSite similarly detects and connects both the orthosteric and allosteric sites, but 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 does not segment the ligand binding pocket into further elements and 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 of the cavities are very solvent-exposed, which hinders the detection of the entirety of the cavities (Figure 6). Since 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.

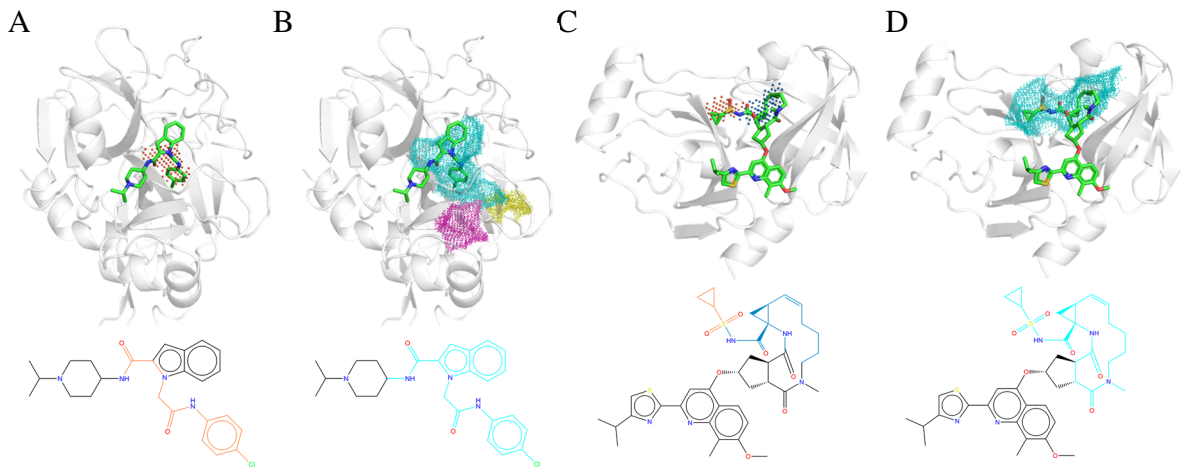


Figure 6. Examples of unsuccessful decomposition by CAVIAR and comparison to DoGSite. In all panels, the 2D structure of the ligand is depicted with a color code corresponding to the subcavity segmentation, or in black if not covered by subcavities. (A) CAVIAR and (B) DoGSite cavity detection and segmentation of factor Xa protease, PDB code 2bqw. (C) CAVIAR and (D) DoGSite cavity detection and segmentation of HCV NS3 protease. In all cases, both software fail at describing correctly the entirety of the cavities and their complexity in terms of subpockets.

Visual interface. CAVIAR is available both as a GUI and as a command line tool. The GUI is developed to be as friendly and transparent as possible for the user, on the contrary to the command line tool. The command line tool comes with many options to provide power users with batch use and the ability to tune their cavity searches. For instance, most parameters of the grid search can be adjusted and tuned 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; presence of a ligand and how much of its atoms are covered by cavity grid points can be investigated. The profusion of parameters can be unsettling, thefore a website was developed to guide the user with extended information (link in the Notes section). The GUI restricts the options

to default 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[80] to visualize the results and choose the automatic coloring scheme according to buriedness, cavity number or pharmacophore type (Figure 7A). The second window relates to the subcavity decomposition and has similar options (Figure 7B).

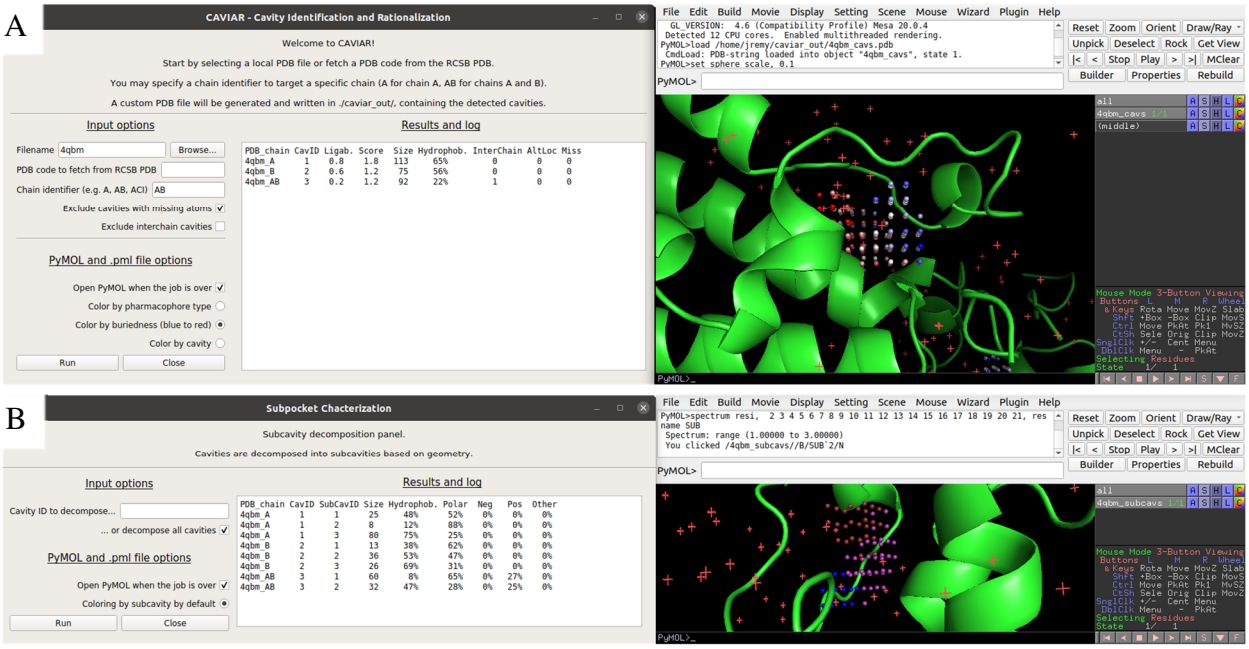


Figure 7. Visual interface for the CAVIAR cavity detection (A) and subcavity decomposition (B) algorithms.

Liganded cavities are more complex than apo cavities. We analyzed 97,221 X-ray structures from the PDB that passed a brief filtering protocol, *viz.*, only X-ray crystallography structures with a resolution below 2.5 Å and no classification as obsolete or having caveats 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 cavities we detected 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, arguing that ligands fill on average only a third of their binding pockets.[56,57] If we focus our analysis 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 complex geometrically (segmented into more subcavities) compared to apo cavities (Table 5). Ligands occupy on average 2.5 ± 1.5 subcavities with a median of 2.

Table 5. 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

Liganded cavities are bigger, more hydrophobic, more ligandable and more complex geometrically (more subcavities) than apo cavities. All comparisons are significant with Kolmogorov-Smirnov tests with a significance level of 0.01 (Supporting Information Table S6).

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 on the PDBBind dataset and on more particularly two types of drug targets in the PDBBind, proteases

and kinases. The more subcavities a compound fills, the higher the affinity. This effect is particularly striking for compounds binding to more than three subcavities, most of them bearing a binding affinity in the nanomolar range or better (Figure 8).

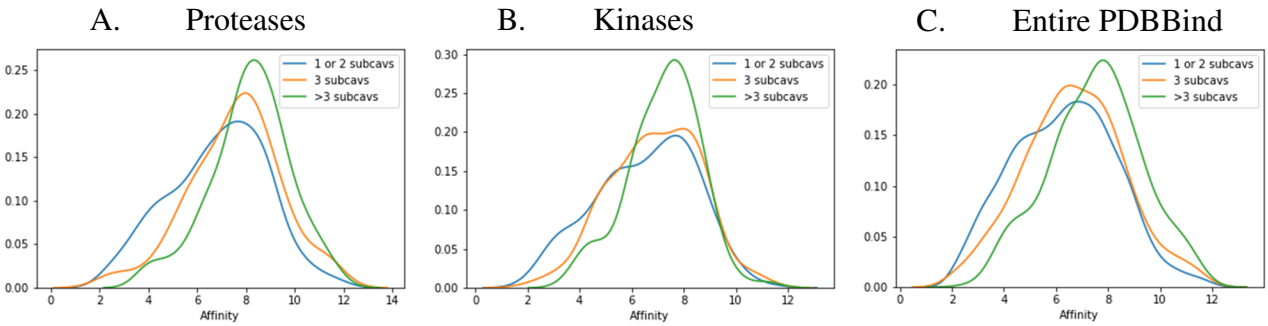


Figure 8. Distribution of binding affinities expressed as –log(affinity) in function of the numbers of subcavities filled by the ligand. (A) Protease dataset. In blue, ligands filling one or two subcavities (n=453), orange three subcavities (n=154), and green four or more subcavities (n=194). The peak of activity is in all cases in the nanomolar range, however, the more subcavities are filled, the less there is 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 when more subpockets are involved in ligand binding (Table 6). In details, 801 unique proteases pockets are liganded and the –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 Supporting Information Table S6).

Table 6. 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 bear 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 worse in the proteases (17 out of 194), kinases (16 out of 103), and entire PDBBind datasets (111 out of 570), respectively. On the contrary, compounds binding to a maximum of three subcavities are 29%, 36% and 42% in the micromolar or worse range, in the proteases, kinases and entire PDBBind datasets, respectively (Table 7).

Table 7. 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.

Limitations of the method. The main limitations of CAVIAR are inherent to the experimental data it relies on, primarily protein structure obtained with X-ray crystallography. This induces a series of caveats that cannot be circumvented. Only proteins with a resolved static structure can be investigated. 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 producing homology structures and generating conformational ensembles from sampling methods.[81–83] Crystal contacts, artifacts and protein chain repeats can produce spurious non-productive interchain cavities (Figure 9A). Significant work has been invested into detecting biologically relevant protein chains contacts,[84,85] and we may implement such an algorithm in later versions of our tool. The second intrinsic limitation of CAVIAR is that it is designed for discovering cavities potentially binding small organic drug-like compounds, which, *de facto*, excludes surface patches such as protein-protein interfaces and very exposed ligand binding grooves (Figure 9B). This may change in the future: different sets of parameters can be optimized for detecting surface patches, or even protein-protein interaction interfaces. Critical settings are accessible via a configuration file and 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 most cavity detection tools, it may overspan cavities, in particular because validation routines tend to reward larger pockets. We optimized the default set of parameters to restrict cavities to direct protein surroundings of

known ligands, but some cases still evade our optimization and produce very large invaginations (Figure 9C). Finally, the validation of a protein cavity detection algorithm is arduous, due to the inherent fuzziness of the definition of what is a "protein cavity" and the long-standing difficulty to design a meaningful validation dataset. This shortcoming is exacerbated for the segmentation of cavities into subcavities, for which a systematic definition simply does not exist to our knowledge. Providing that the input cavity is correct, the subsite decomposition suffers from very few false negatives. In other words, it tends to produce more subpockets than less, *e.g.*, oversegment the pocket rather than fail to characterize a subcavity.

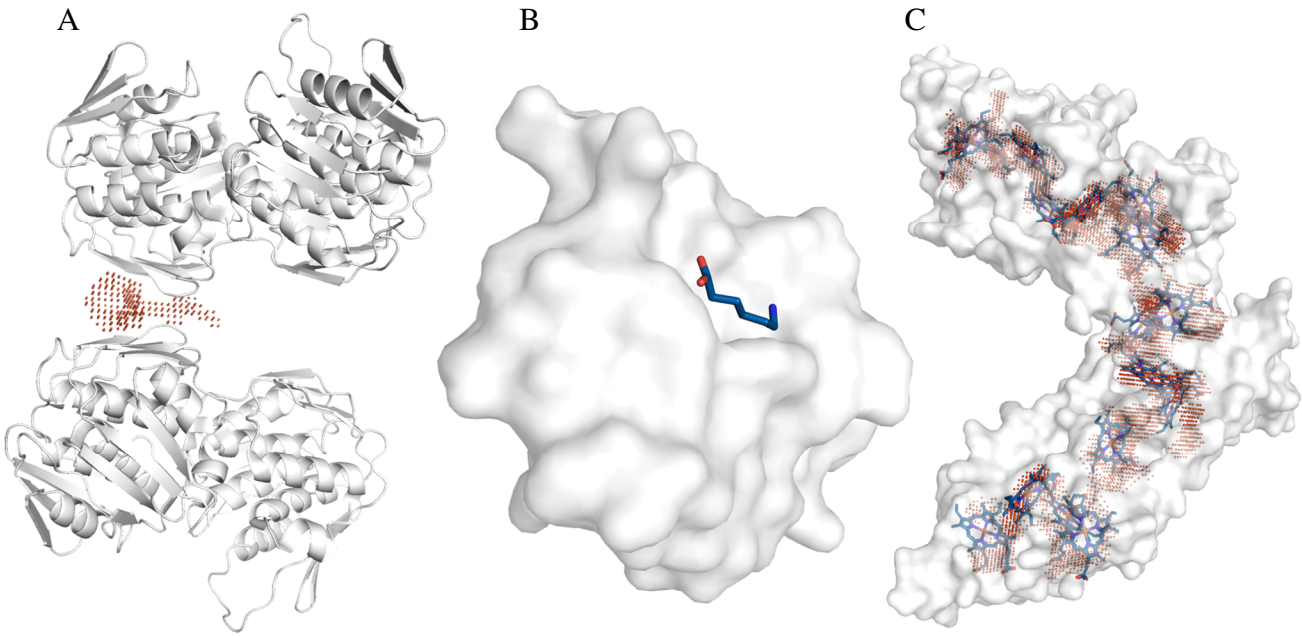


Figure 9. Representative cases of failure with CAVIAR. (A) Spurious interchain cavity. A cavity, in orange spheres, is found at the interface between two protein chains, in white cartoons, which is a crystal contact and not biologically relevant (PDB code 1ejd). (B) Case of an exposed ligand, in blue 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 protein.

CONCLUSIONS

The most fruitful applications of cavity detection tools depend on what questions the generated data is used to answer to: can we use the generated cavities and cavity descriptors to evaluate ligandability or to compare cavities? Both these applications require to have a cavity detection software with high performance, accessible, and tunable for one's needs. Many of the published software are closed-source, incorporated in commercial packages, accessible only in the form of webservers, or, more often than not, simply no longer accessible. The open-source availability of CAVIAR on GitHub and Anaconda combined with its comprehensive Python language defines it as a powerful toolkit to build upon with. A dedicated website is available with step-by-step usage notes and an extended manual to help the community adjust CAVIAR to their needs (See the Notes section for the website, GitHub and Anaconda links). The cavity detection, characterization and segmentation runs fast, ranging from a five seconds average on the DUD-e 102 targets (including tool initialization, reading and writing the files) 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. We did not run a systematic benchmark of the computational efficiency of CAVIAR against other similar software. The 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.

Moreover, some novel notions were introduced as an attempt to refine the cavity detection and address challenges that are not resolved in the literature, such as cavity overspanning of buriedness-based algorithms and the analysis of protein subpockets. The comparative investigation of protein subcavities may help to understand selectivity issues or polypharmacological effect 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 comparably to what is done with matched molecular pairs of chemicals.[87] The notion of subcavity is an ill-defined concept and the robust partitioning of binding pockets into subpockets is an unmet need in medicinal chemistry and chemical biology. CAVIAR aims at a systematic detection and classification of protein subcavities. Moreover, the deconstruction of pockets into subcavities may help for partial cavity matching in the context of cavity comparison.[88] Our analysis of the PDB resulted in 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 reasonably with the number of subcavities they fill, with a propensity to high affinities, in the nanomolar range or better, for ligands binding to more than three subcavities.

ASSOCIATED MATERIAL

Supporting Information.

List of parameters optimized for cavity detection, analysis of the potential grid dependency of CAVIAR's algorithm, details of results on the 48 bound/unbound validation dataset, presentation of the ligandability descriptors and ligandability module, additional statistics on the PDB, scPDB, PDBBind and significance tests (PDF)

AUTHOR INFORMATION

Corresponding Author

*J.-R.M., e-mail: jean-remy.marchand@novartis.com

*F.S., e-mail: 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

Author 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.

Notes

Installation notes, user manual and help for CAVIAR are available at https://jr-marchand.github.io/caviar/. The CAVIAR GUI and CAVIAR command line tool are available on GitHub at https://github.com/jr-marchand/caviar and on Anaconda cloud at https://anaconda.org/jr-marchand/caviar under a MIT license. The GitHub repository also hosts the validation datasets.

ACKNOWLEDGMENT

The authors thank Imtiaz Hossein and Michael Schaefer for insightful discussions. This work was supported by the postdoctoral office of Novartis. J.-R.M. thanks all contributors to open source codes for their crucial work.

ABBREVIATIONS

CAVIAR, cavity identification and rationalization; PDB, protein data bank; GUI, graphical user interface; GPCR, G-protein coupled receptor

REFERENCES

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

(2) Simões, T.; Lopes, D.; Dias, S.; Fernandes, F.; Pereira, J.; Jorge, J.; Bajaj, C.; Gomes, A. Geometric Detection Algorithms for Cavities on Protein Surfaces in Molecular Graphics: A Survey. *Comput. Graph. Forum* **2017**, *36*, 643–683. https://doi.org/10.1111/cgf.13158.

(3) Volkamer, A.; Behren, M. M. von; Bietz, S.; Rarey, M. Prediction, Analysis, and Comparison of Active Sites. In *Applied Chemoinformatics*; John Wiley & Sons, Ltd, 2018; pp 283–311. https://doi.org/10.1002/9783527806539.ch6g.

(4) Macari, G.; Toti, D.; Polticelli, F. 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.* **2019**, *33*, 887–903. https://doi.org/10.1007/s10822-019-00235-7.

(5) Volkamer, A.; Kuhn, D.; Rippmann, F.; Rarey, M. DoGSiteScorer: A Web Server for Automatic Binding Site Prediction, Analysis and Druggability Assessment. *Bioinformatics* **2012**, *28*, 2074–2075. https://doi.org/10.1093/bioinformatics/bts310.

(6) Le Guilloux, V.; Schmidtke, P.; Tuffery, P. Fpocket: An Open Source Platform for Ligand Pocket Detection. *BMC Bioinformatics* **2009**, *10*, 168. https://doi.org/10.1186/1471-2105-10-168.
(7) Halgren, T. A. Identifying and Characterizing Binding Sites and Assessing Druggability. *J. Chem. Inf. Model.* **2009**, *49*, 377–389. https://doi.org/10.1021/ci800324m.
(8) Nayal, M.; Honig, B. On the Nature of Cavities on Protein Surfaces: Application to the Identification of Drug-Binding Sites. *Proteins Struct. Funct. Bioinforma.* **2006**, *63*, 892–906. https://doi.org/10.1002/prot.20897.
(9) Desaphy, J.; Azdimousa, K.; Kellenberger, E.; Rognan, D. Comparison and Druggability Prediction of Protein–Ligand Binding Sites from Pharmacophore-Annotated Cavity Shapes. *J. Chem. Inf. Model.* **2012**, *52*, 2287–2299. https://doi.org/10.1021/ci300184x.
(10) Ehrt, C.; Brinkjost, T.; Koch, O. Impact of Binding Site Comparisons on Medicinal Chemistry and Rational Molecular Design. *J. Med. Chem.* **2016**, *59*, 4121–4151. https://doi.org/10.1021/acs.jmedchem.6b00078.
(11) Xie, L.; Evangelidis, T.; Xie, L.; Bourne, P. E. Drug Discovery Using Chemical Systems Biology: Weak Inhibition of Multiple Kinases May Contribute to the Anti-Cancer Effect of Nelfinavir. *PLOS Comput. Biol.* **2011**, *7*, e1002037. https://doi.org/10.1371/journal.pcbi.1002037.
(12) Möller-Acuña, P.; Contreras-Riquelme, J. S.; Rojas-Fuentes, C.; Nuñez-Vivanco, G.; Alzate-Morales, J.; Iturriaga-Vásquez, P.; Arias, H. R.; Reyes-Parada, M. Similarities between the Binding Sites of SB-206553 at Serotonin Type 2 and Alpha7 Acetylcholine Nicotinic Receptors: Rationale for Its Polypharmacological Profile. *PLOS ONE* **2015**, *10*, e0134444. https://doi.org/10.1371/journal.pone.0134444.
(13) Schumann, M.; Armen, R. S. Identification of Distant Drug Off-Targets by Direct Superposition of Binding Pocket Surfaces. *PLOS ONE* **2013**, *8*, e83533. https://doi.org/10.1371/journal.pone.0083533.
(14) Schirris, T. J. J.; Ritschel, T.; Herma Renkema, G.; Willems, P. H. G. M.; Smeitink, J. A. M.; Russel, F. G. M. Mitochondrial ADP/ATP Exchange Inhibition: A Novel off-Target Mechanism Underlying Ibipinabant-Induced Myotoxicity. *Sci. Rep.* **2015**, *5*, 1–12. https://doi.org/10.1038/srep14533.
(15) Kuhn, D.; Weskamp, N.; Schmitt, S.; Hüllermeier, E.; Klebe, G. From the Similarity Analysis of Protein Cavities to the Functional Classification of Protein Families Using Cavbase. *J. Mol. Biol.* **2006**, *359*, 1023–1044. https://doi.org/10.1016/j.jmb.2006.04.024.
(16) Kinoshita, K.; Furui, J.; Nakamura, H. Identification of Protein Functions from a Molecular Surface Database, EF-Site. *J. Struct. Funct. Genomics* **2002**, *2*, 9–22. https://doi.org/10.1023/A:1011318527094.
(17) Konc, J.; Hodošček, M.; Ogrizek, M.; Konc, J. T.; Janežič, D. Structure-Based Function Prediction of Uncharacterized Protein Using Binding Sites Comparison. *PLOS Comput. Biol.* **2013**, *9*, e1003341. https://doi.org/10.1371/journal.pcbi.1003341.
(18) Anand, P.; Sankaran, S.; Mukherjee, S.; Yeturu, K.; Laskowski, R.; Bhardwaj, A.; Bhagavat, R.; Consortium, O.; Brahmachari, S. K.; Chandra, N. Structural Annotation of Mycobacterium Tuberculosis Proteome. *PLOS ONE* **2011**, *6*, e27044. https://doi.org/10.1371/journal.pone.0027044.
(19) Al-Gharabli, S. I.; Shah, S. T. A.; Weik, S.; Schmidt, M. F.; Mesters, J. R.; Kuhn, D.; Klebe, G.; Hilgenfeld, R.; Rademann, J. 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* **2006**, *7*, 1048–1055. https://doi.org/10.1002/cbic.200500533.

(20) Willmann, D.; Lim, S.; Wetzel, S.; Metzger, E.; Jandausch, A.; Wilk, W.; Jung, M.; Forne, I.; Imhof, A.; Janzer, A.; Kirfel, J.; Waldmann, H.; Schüle, R.; Buettner, R. Impairment of Prostate Cancer Cell Growth by a Selective and Reversible Lysine-Specific Demethylase 1 Inhibitor. *Int. J. Cancer* **2012**, *131*, 2704–2709. https://doi.org/10.1002/ijc.27555.

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

(22) Weber, A.; Casini, A.; Heine, A.; Kuhn, D.; Supuran, C. T.; Scozzafava, A.; Klebe, G. Unexpected Nanomolar Inhibition of Carbonic Anhydrase by COX-2-Selective Celecoxib: New Pharmacological Opportunities Due to Related Binding Site Recognition. *J. Med. Chem.* **2004**, *47*, 550–557. https://doi.org/10.1021/jm030912m.

(23) Weisel, M.; Proschak, E.; Schneider, G. PocketPicker: Analysis of Ligand Binding-Sites with Shape Descriptors. *Chem. Cent. J.* **2007**, *1*, 7. https://doi.org/10.1186/1752-153X-1-7.

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

(25) Goodford, P. J. A Computational Procedure for Determining Energetically Favorable Binding Sites on Biologically Important Macromolecules. *J. Med. Chem.* **1985**, *28*, 849–857. https://doi.org/10.1021/jm00145a002.

(26) Bliznyuk, A. A.; Gready, J. E. Identification and Energetic Ranking of Possible Docking Sites for Pterin on Dihydrofolate Reductase. *J. Comput. Aided Mol. Des.* **1998**, *12*, 325–333. https://doi.org/10.1023/A:1008039000355.

(27) Ngan, C. H.; Bohnuud, T.; Mottarella, S. E.; Beglov, D.; Villar, E. A.; Hall, D. R.; Kozakov, D.; Vajda, S. FTMAP: Extended Protein Mapping with User-Selected Probe Molecules. *Nucleic Acids Res.* **2012**, *40*, W271–W275. https://doi.org/10.1093/nar/gks441.

(28) Laurie, A. T. R.; Jackson, R. M. Q-SiteFinder: An Energy-Based Method for the Prediction of Protein–Ligand Binding Sites. *Bioinformatics* **2005**, *21*, 1908–1916. https://doi.org/10.1093/bioinformatics/bti315.

(29) Marchand, J.-R.; Caflisch, A. In Silico Fragment-Based Drug Design with SEED. *Eur. J. Med. Chem.* **2018**, *156*, 907–917. https://doi.org/10.1016/j.ejmech.2018.07.042.

(30) Miranker, A.; Karplus, M. Functionality Maps of Binding Sites: A Multiple Copy Simultaneous Search Method. *Proteins Struct. Funct. Bioinforma.* **1991**, *11*, 29–34. https://doi.org/10.1002/prot.340110104.

(31) Simões, T.; Lopes, D.; Dias, S.; Fernandes, F.; Pereira, J.; Jorge, J.; Bajaj, C.; Gomes, A. Geometric Detection Algorithms for Cavities on Protein Surfaces in Molecular Graphics: A Survey. *Comput. Graph. Forum J. Eur. Assoc. Comput. Graph.* **2017**, *36*, 643–683. https://doi.org/10.1111/cgf.13158.

(32) Xie, Z.-R.; Hwang, M.-J. Methods for Predicting Protein–Ligand Binding Sites. In *Molecular Modeling of Proteins*; Kukol, A., Ed.; Methods in Molecular Biology; Springer: New York, NY, 2015; pp 383–398. https://doi.org/10.1007/978-1-4939-1465-4_17.

(33) Huang, B.; Schroeder, M. LIGSITEcsc: Predicting Ligand Binding Sites Using the Connolly Surface and Degree of Conservation. *BMC Struct. Biol.* **2006**, *6*, 19. https://doi.org/10.1186/1472-6807-6-19.

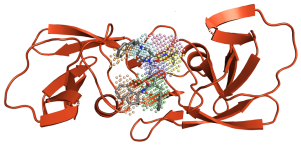
(34) Capra, J. A.; Laskowski, R. A.; Thornton, J. M.; Singh, M.; Funkhouser, T. A. Predicting Protein Ligand Binding Sites by Combining Evolutionary Sequence Conservation and 3D Structure. *PLOS Comput. Biol.* **2009**, *5*, e1000585. https://doi.org/10.1371/journal.pcbi.1000585.
(35) Huang, B. MetaPocket: A Meta Approach to Improve Protein Ligand Binding Site Prediction. *OMICS J. Integr. Biol.* **2009**, *13*, 325–330. https://doi.org/10.1089/omi.2009.0045.
(36) Zhang, Z.; Li, Y.; Lin, B.; Schroeder, M.; Huang, B. Identification of Cavities on Protein Surface Using Multiple Computational Approaches for Drug Binding Site Prediction. *Bioinformatics* **2011**, *27*, 2083–2088. https://doi.org/10.1093/bioinformatics/btr331.
(37) Levitt, D. G.; Banaszak, L. J. POCKET: A Computer Graphies Method for Identifying and Displaying Protein Cavities and Their Surrounding Amino Acids. *J. Mol. Graph.* **1992**, *10*, 229–234. https://doi.org/10.1016/0263-7855(92)80074-N.
(38) Hendlich, M.; Rippmann, F.; Barnickel, G. LIGSITE: Automatic and Efficient Detection of Potential Small Molecule-Binding Sites in Proteins. *J. Mol. Graph. Model.* **1997**, *15*, 359–363. https://doi.org/10.1016/S1093-3263(98)00002-3.
(39) Kalidas, Y.; Chandra, N. PocketDepth: A New Depth Based Algorithm for Identification of Ligand Binding Sites in Proteins. *J. Struct. Biol.* **2008**, *161*, 31–42. https://doi.org/10.1016/j.jsb.2007.09.005.
(40) Till, M. S.; Ullmann, G. M. McVol - A Program for Calculating Protein Volumes and Identifying Cavities by a Monte Carlo Algorithm. *J. Mol. Model.* **2010**, *16*, 419–429. https://doi.org/10.1007/s00894-009-0541-y.
(41) Tripathi, A.; Kellogg, G. E. A Novel and Efficient Tool for Locating and Characterizing Protein Cavities and Binding Sites. *Proteins Struct. Funct. Bioinforma.* **2010**, *78*, 825–842. https://doi.org/10.1002/prot.22608.
(42) Laskowski, R. A. SURFNET: A Program for Visualizing Molecular Surfaces, Cavities, and Intermolecular Interactions. *J. Mol. Graph.* **1995**, *13*, 323–330. https://doi.org/10.1016/0263-7855(95)00073-9.
(43) Brady, G. P.; Stouten, P. F. W. Fast Prediction and Visualization of Protein Binding Pockets with PASS. *J. Comput. Aided Mol. Des.* **2000**, *14*, 383–401. https://doi.org/10.1023/A:1008124202956.
(44) Kawabata, T.; Go, N. Detection of Pockets on Protein Surfaces Using Small and Large Probe Spheres to Find Putative Ligand Binding Sites. *Proteins* **2007**, *68*, 516–529. https://doi.org/10.1002/prot.21283.
(45) Oliveira, S. H.; Ferraz, F. A.; Honorato, R. V.; Xavier-Neto, J.; Sobreira, T. J.; de Oliveira, P. S. KVFinder: Steered Identification of Protein Cavities as a PyMOL Plugin. *BMC Bioinformatics* **2014**, *15*, 197. https://doi.org/10.1186/1471-2105-15-197.
(46) Kawabata, T. Detection of Multiscale Pockets on Protein Surfaces Using Mathematical Morphology. *Proteins Struct. Funct. Bioinforma.* **2010**, *78*, 1195–1211. https://doi.org/10.1002/prot.22639.
(47) Yu, J.; Zhou, Y.; Tanaka, I.; Yao, M. Roll: A New Algorithm for the Detection of Protein Pockets and Cavities with a Rolling Probe Sphere. *Bioinformatics* **2010**, *26*, 46–52. https://doi.org/10.1093/bioinformatics/btp599.
(48) Lewis, R. A. Determination of Clefts in Receptor Structures. *J. Comput. Aided Mol. Des.* **1989**, *3*, 133–147. https://doi.org/10.1007/BF01557724.

(49) Peters, K. P.; Fauck, J.; Frömmel, C. The Automatic Search for Ligand Binding Sites in Proteins of Known Three-Dimensional Structure Using Only Geometric Criteria. *J. Mol. Biol.* **1996**, *256*, 201–213. https://doi.org/10.1006/jmbi.1996.0077.
(50) Liang, J.; Edelsbrunner, H.; Woodward, C. Anatomy of Protein Pockets and Cavities: Measurement of Binding Site Geometry and Implications for Ligand Design. *Protein Sci. Publ. Protein Soc.* **1998**, *7*, 1884–1897.
(51) Simões, T. M. C.; Gomes, A. J. P. CavVis—A Field-of-View Geometric Algorithm for Protein Cavity Detection. *J. Chem. Inf. Model.* **2019**, *59*, 786–796. https://doi.org/10.1021/acs.jcim.8b00572.
(52) Hajduk, P. J.; Meadows, R. P.; Fesik, S. W. Discovering High-Affinity Ligands for Proteins. *Science* **1997**, *278*, 497–499. https://doi.org/10.1126/science.278.5337.497.
(53) Bartolowits, M.; Davisson, V. J. Considerations of Protein Subpockets in Fragment-Based Drug Design. *Chem. Biol. Drug Des.* **2016**, *87*, 5–20. https://doi.org/10.1111/cbdd.12631.
(54) Erlanson, D. A.; Fesik, S. W.; Hubbard, R. E.; Jahnke, W.; Jhoti, H. Twenty Years on: The Impact of Fragments on Drug Discovery. *Nat. Rev. Drug Discov.* **2016**, *15*, 605–619. https://doi.org/10.1038/nrd.2016.109.
(55) Marchand, J.-R.; Dalle Vedove, A.; Lolli, G.; Caflisch, A. Discovery of Inhibitors of Four Bromodomains by Fragment-Anchored Ligand Docking. *J. Chem. Inf. Model.* **2017**, *57*, 2584–2597. https://doi.org/10.1021/acs.jcim.7b00336.
(56) Wirth, M.; Volkamer, A.; Zoete, V.; Rippmann, F.; Michielin, O.; Rarey, M.; Sauer, W. H. B. Protein Pocket and Ligand Shape Comparison and Its Application in Virtual Screening. *J. Comput. Aided Mol. Des.* **2013**, *27*, 511–524. https://doi.org/10.1007/s10822-013-9659-1.
(57) Kahraman, A.; Morris, R. J.; Laskowski, R. A.; Thornton, J. M. Shape Variation in Protein Binding Pockets and Their Ligands. *J. Mol. Biol.* **2007**, *368*, 283–301. https://doi.org/10.1016/j.jmb.2007.01.086.
(58) Chan, A. W. E.; Laskowski, R. A.; Selwood, D. L. Chemical Fragments That Hydrogen Bond to Asp, Glu, Arg, and His Side Chains in Protein Binding Sites. *J. Med. Chem.* **2010**, *53*, 3086–3094. https://doi.org/10.1021/jm901696w.
(59) Wang, L.; Xie, Z.; Wipf, P.; Xie, X.-Q. Residue Preference Mapping of Ligand Fragments in the Protein Data Bank. *J. Chem. Inf. Model.* **2011**, *51*, 807–815. https://doi.org/10.1021/ci100386y.
(60) Durrant, J. D.; Friedman, A. J.; McCammon, J. A. CrystalDock: A Novel Approach to Fragment-Based Drug Design. *J. Chem. Inf. Model.* **2011**, *51*, 2573–2580. https://doi.org/10.1021/ci200357y.
(61) Tang, G. W.; Altman, R. B. Knowledge-Based Fragment Binding Prediction. *PLOS Comput. Biol.* **2014**, *10*, e1003589. https://doi.org/10.1371/journal.pcbi.1003589.
(62) Kalliokoski, T.; Olsson, T. S. G.; Vulpetti, A. Subpocket Analysis Method for Fragment-Based Drug Discovery. *J. Chem. Inf. Model.* **2013**, *53*, 131–141. https://doi.org/10.1021/ci300523r.
(63) Wood, D. J.; Vlieg, J. de; Wagener, M.; Ritschel, T. Pharmacophore Fingerprint-Based Approach to Binding Site Subpocket Similarity and Its Application to Bioisostere Replacement. *J. Chem. Inf. Model.* **2012**, *52*, 2031–2043. https://doi.org/10.1021/ci3000776.

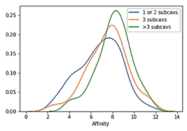
(64) Volkamer, A.; Eid, S.; Turk, S.; Rippmann, F.; Fulle, S. Identification and Visualization of Kinase-Specific Subpockets. *J. Chem. Inf. Model.* **2016**, *56*, 335–346. https://doi.org/10.1021/acs.jcim.5b00627.
(65) Bakan, A.; Meireles, L. M.; Bahar, I. ProDy: Protein Dynamics Inferred from Theory and Experiments. *Bioinformatics* **2011**, *27*, 1575–1577. https://doi.org/10.1093/bioinformatics/btr168.
(66) Weisel, M.; Proschak, E.; Kriegl, J. M.; Schneider, G. Form Follows Function: Shape Analysis of Protein Cavities for Receptor-Based Drug Design. *PROTEOMICS* **2009**, *9*, 451–459. https://doi.org/10.1002/pmic.200800092.
(67) Mysinger, M. M.; Carchia, M.; Irwin, J. J.; Shoichet, B. K. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *J. Med. Chem.* **2012**, *55*, 6582–6594. https://doi.org/10.1021/jm300687e.
(68) Desaphy, J.; Bret, G.; Rognan, D.; Kellenberger, E. Sc-PDB: A 3D-Database of Ligandable Binding Sites—10 Years On. *Nucleic Acids Res.* **2015**, *43*, D399–D404. https://doi.org/10.1093/nar/gku928.
(69) Liu, Z.; Li, Y.; Han, L.; Li, J.; Liu, J.; Zhao, Z.; Nie, W.; Liu, Y.; Wang, R. PDB-Wide Collection of Binding Data: Current Status of the PDBbind Database. *Bioinformatics* **2015**, *31*, 405–412. https://doi.org/10.1093/bioinformatics/btu626.
(70) Krasowski, A.; Muthas, D.; Sarkar, A.; Schmitt, S.; Brenk, R. DrugPred: A Structure-Based Approach To Predict Protein Druggability Developed Using an Extensive Nonredundant Data Set. *J. Chem. Inf. Model.* **2011**, *51*, 2829–2842. https://doi.org/10.1021/ci200266d.
(71) Borrel, A.; Regad, L.; Xhaard, H.; Petitjean, M.; Camproux, A.-C. PockDrug: A Model for Predicting Pocket Druggability That Overcomes Pocket Estimation Uncertainties. *J. Chem. Inf. Model.* **2015**, *55*, 882–895. https://doi.org/10.1021/ci5006004.
(72) Beucher, S. Watershed, Hierarchical Segmentation and Waterfall Algorithm. In *Mathematical Morphology and Its Applications to Image Processing*; Serra, J., Soille, P., Eds.; Computational Imaging and Vision; Springer Netherlands: Dordrecht, 1994; pp 69–76. https://doi.org/10.1007/978-94-011-1040-2_10.
(73) Schmidtke, P.; Barril, X. Understanding and Predicting Druggability. A High-Throughput Method for Detection of Drug Binding Sites. *J. Med. Chem.* **2010**, *53*, 5858–5867. https://doi.org/10.1021/jm100574m.
(74) Pirard, B.; Ertl, P. Evaluation of a Semi-Automated Workflow for Fragment Growing. *J. Chem. Inf. Model.* **2015**, *55*, 180–193. https://doi.org/10.1021/ci5006355.
(75) Huth, J. R.; Park, C.; Petros, A. M.; Kunzer, A. R.; Wendt, M. D.; Wang, X.; Lynch, C. L.; Mack, J. C.; Swift, K. M.; Judge, R. A.; Chen, J.; Richardson, P. L.; Jin, S.; Tahir, S. K.; Matayoshi, E. D.; Dorwin, S. A.; Ladror, U. S.; Severin, J. M.; Walter, K. A.; Bartley, D. M.; Fesik, S. W.; Elmore, S. W.; Hajduk, P. J. Discovery and Design of Novel HSP90 Inhibitors Using Multiple Fragment-Based Design Strategies. *Chem. Biol. Drug Des.* **2007**, *70*, 1–12. https://doi.org/10.1111/j.1747-0285.2007.00535.x.
(76) Ghosh, A. K.; Osswald, H. L.; Prato, G. Recent Progress in the Development of HIV-1 Protease Inhibitors for the Treatment of HIV/AIDS. *J. Med. Chem.* **2016**, *59*, 5172–5208. https://doi.org/10.1021/acs.jmedchem.5b01697.
(77) Munshi, S.; Chen, Z.; Yan, Y.; Li, Y.; Olsen, D. B.; Schock, H. B.; Galvin, B. B.; Dorsey, B.; Kuo, L. C. 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.* **2000**, *56*, 381–388. https://doi.org/10.1107/S0907444900000469.
(78) Thal, D. M.; Sun, B.; Feng, D.; Nawaratne, V.; Leach, K.; Felder, C. C.; Bures, M. G.; Evans, D. A.; Weis, W. I.; Bachhawat, P.; Kobilka, T. S.; Sexton, P. M.; Kobilka, B. K.; Christopoulos, A. Crystal Structures of the M1 and M4 Muscarinic Acetylcholine Receptors. *Nature* **2016**, *531*, 335–340. https://doi.org/10.1038/nature17188.
(79) Wood, E. R.; Truesdale, A. T.; McDonald, O. B.; Yuan, D.; Hassell, A.; Dickerson, S. H.; Ellis, B.; Pennisi, C.; Horne, E.; Lackey, K.; Alligood, K. J.; Rusnak, D. W.; Gilmer, T. M.; Shewchuk, L. 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.* **2004**, *64*, 6652–6659. https://doi.org/10.1158/0008-5472.CAN-04-1168.
(80) *The PyMOL Molecular Graphics System, Version 2.0 Schrödinger, LLC.*
(81) Bacci, M.; Langini, C.; Vymětal, J.; Caflisch, A.; Vitalis, A. Focused Conformational Sampling in Proteins. *J. Chem. Phys.* **2017**, *147*, 195102. https://doi.org/10.1063/1.4996879.
(82) Laio, A.; Gervasio, F. L. Metadynamics: A Method to Simulate Rare Events and Reconstruct the Free Energy in Biophysics, Chemistry and Material Science. *Rep. Prog. Phys.* **2008**, *71*, 126601. https://doi.org/10.1088/0034-4885/71/12/126601.
(83) Kuzmanic, A.; Bowman, G. R.; Juarez-Jimenez, J.; Michel, J.; Gervasio, F. L. Investigating Cryptic Binding Sites by Molecular Dynamics Simulations. *Acc. Chem. Res.* **2020**. https://doi.org/10.1021/acs.accounts.9b00613.
(84) Duarte, J. M.; Srebniak, A.; Schärer, M. A.; Capitani, G. Protein Interface Classification by Evolutionary Analysis. *BMC Bioinformatics* **2012**, *13*, 334. https://doi.org/10.1186/1471-2105-13-334.
(85) Capitani, G.; Duarte, J. M.; Baskaran, K.; Bliven, S.; Somody, J. C. Understanding the Fabric of Protein Crystals: Computational Classification of Biological Interfaces and Crystal Contacts. *Bioinformatics* **2016**, *32*, 481–489. https://doi.org/10.1093/bioinformatics/btv622.
(86) Jalencas, X.; Mestres, J. Chemoisosterism in the Proteome. *J. Chem. Inf. Model.* **2013**, *53*, 279–292. https://doi.org/10.1021/ci3002974.
(87) Keefer, C. E.; Chang, G. The Use of Matched Molecular Series Networks for Cross Target Structure Activity Relationship Translation and Potency Prediction. *MedChemComm* **2017**, *8*, 2067–2078. https://doi.org/10.1039/C7MD00465F.
(88) Krotzky, T.; Rickmeyer, T.; Fober, T.; Klebe, G. Extraction of Protein Binding Pockets in Close Neighborhood of Bound Ligands Makes Comparisons Simple Due to Inherent Shape Similarity. *J. Chem. Inf. Model.* **2014**, *54*, 3229–3237. https://doi.org/10.1021/ci500553a.

TABLE OF CONTENTS GRAPHIC



Protein-based definition of cavities and their subcavities



More subpockets filled ⇨ higher affinity