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 and chemical biology. Many tools were developed in the last 40 years, but few of them are still available in 2020, open-source, and suitable for the analysis of large databases or for the integration in automatic workflows. 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, supporting PDB and mmCIF files as well as DCD trajectories from molecular dynamics simulations. The protein structure serves as input for automatic cavity detection and computation of properties, including ligandability. A subcavity segmentation algorithm 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 PDB shows 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 subcavities 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 a conda package is hosted on Anaconda cloud at https://anaconda.org/jr-marchand/caviar. The software suite is free and all of the source code is available under a permissive MIT license. The lists of PDB files used for validation, as well as the results of subpocket decomposition with CAVIAR and DoGSite are hosted on GitHub at https://github.com/jr-marchand/caviar/tree/master/validation_sets.

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

1. 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 (Westbrook and Burley, 2019). 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, such as the whole PDB, including structures without ligand, requires automatic algorithms to perform that task. The cavity detection field has been prolific in the last three decades (Simões *et al.*, 2017a; Volkamer *et al.*, 2018; Macari *et al.*, 2019). Successful applications include the prediction of target ligandability (Volkamer *et al.*, 2012; Le Guilloux *et al.*, 2009; Halgren, 2009; Nayal and Honig, 2006; Desaphy *et al.*, 2012), identification of off-targets (Ehrt *et al.*, 2016; Xie *et al.*, 2011; Möller-Acuña *et al.*, 2015; Schumann and Armen, 2013; Schirris *et al.*, 2015), functional annotation (Kuhn *et al.*, 2006; Kinoshita *et al.*, 2002; Konc *et al.*, 2013; Anand *et al.*, 2011) and ligand design and drug repurposing (Al-Gharabli *et al.*, 2006; Willmann *et al.*, 2012; Kooistra *et al.*, 2015; Weber *et al.*, 2004). Structure-based cavity detection methods can be grouped into two general families: energy-based algorithms and geometry-based (Simões *et al.*, 2017a; Weisel *et al.*, 2007; Volkamer, Griewel, *et al.*, 2010). 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 (Goodford, 1985; Bliznyuk and Gready, 1998; Ngan *et al.*, 2012; Laurie and Jackson, 2005; Marchand and Caflisch, 2018; Miranker and Karplus, 1991). 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 and easily automated for the integration in workflows. 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 (Simões *et al.*, 2017b; Ehrt *et al.*, 2016; Xie and Hwang, 2015; Huang and Schroeder, 2006; Capra *et al.*, 2009). 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 (Huang, 2009; Zhang *et al.*, 2011).

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 (Levitt and Banaszak, 1992), LIGSITE (Hendlich *et al.*, 1997), PocketDepth* (Kalidas and Chandra, 2008), PocketPicker (Weisel *et al.*, 2007), McVol* (Till and Ullmann, 2010), VICE (Tripathi and Kellogg, 2010), VolSite (Desaphy *et al.*, 2012), SiteMap (Halgren, 2009)
Space filling	Spheres are placed around the protein surface to detect empty spaces in the protein convex hull	SURFNET (Laskowski, 1995), PASS (Brady and Stouten, 2000), PHECOM* (Kawabata and Go, 2007), KVFinder* (Oliveira *et al.*, 2014), GHECOM* (Kawabata, 2010), SCREEN (Nayal and Honig, 2006), POCASA (Yu *et al.*, 2010)
Voronoi diagram	The Voronoi decomposition of the space of protein atoms serves as basis to identify clefts	FindSurf (Lewis, 1989), CAST (Peters *et al.*, 1996), APROPOS (Liang *et al.*, 1998), Fpocket* (Le Guilloux *et al.*, 2009), SiteFinder (MOE)
Imaging science	Gaussian surfaces approximate the protein shape	DoGSite (Volkamer, Griewel, *et al.*, 2010), CavVis*(Simões and Gomes, 2019)

* 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% (Macari *et al.*, 2019). However, many programs are either distributed as closed-source commercial packages/webservers or unavailable (Krivák and Hoksza, 2018). Cavity segmentation into subcavities is crucial in the era of structure-based drug discovery to help medicinal chemists optimizing properties like potency and selectivity (Marchand and Caflisch, 2018; Hajduk *et al.*, 1997; Bartolowits and Davisson, 2016; Erlanson *et al.*, 2016; Marchand *et al.*, 2017). 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 compouds (Bartolowits and Davisson, 2016). 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 (Wirth *et al.*, 2013; Kahraman *et al.*, 2007). Efforts have been made to try to characterize the chemical fragment preference of certain residues (Chan *et al.*, 2010; Wang *et al.*, 2011) and link the fragment chemical space to binding pocket microenvironments (Durrant *et al.*, 2011; Tang and Altman, 2014; Kalliokoski *et al.*, 2013; Wood *et al.*, 2012). 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, developed as a ligand-agnostic cavity identification tool, borrows concepts from the computational image recognition field (Volkamer, Griewel, *et al.*, 2010). Briefly,

DoGSite uses a difference of Gaussians algorithm to identify hotspots and then inflates them before merging them into larger cavities. Original hotspots can be treated as subpockets without further processing. However, the subpockets as defined by DoGSite do not originally aim at reproducing the concept of subpockets from medicinal chemistry. They rather circumvent pocket overspanning, with one single subpocket entirely binding the ligand in 87% of the cases (Volkamer, Griewel, *et al.*, 2010; Volkamer, Grombacher, *et al.*, 2010). Moreover, the software 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 (Volkamer *et al.*, 2016). In order to enable the development of novel cavity comparison workflows, we present a comprehensive ligand-agnostic Python-based open-source platform for cavity detection and characterization, usable with a graphical user interface (GUI) or a command line tool. The usage of CAVIAR can be as simple as running the command 'caviar –code PDB-code', with automatic downloading from the RCSB PDB webservers, or as complex as using a parameter file with dozens of options, most CAVIAR settings being tunable. CAVIAR supports PDB and mmCIF files, as well as DCD molecular dynamics trajectory files. The later functionality contains an orientation invariant clustering of pockets in order to determine the occupancy of cavities within a trajectory and identify the representative structures. In addition, we addressed a blind spot of existing algorithms, *i.e.*, the decomposition of binding cavities into medicinal chemistry compatible subpockets. We assessed qualitatively the pertinence of the subsite decomposition produced by CAVIAR and put it in parallel with DoGSite's results. Our work also includes a comparative analysis of liganded and apo pockets at the PDB level. Finally, we investigated the relationship between the number of subcavities filled by ligands and the binding affinities to their target.

2. MATERIAL AND METHODS

2.1 Cavity identification

We started from well-established concepts of enclosure of grid points algorithms and augmented them 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, size and hydrophobicity filters. The 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 a protein type. Remaining grid points are considered as solvent grid points and investigated further (Fig. 1A). 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 (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 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 (Fig. 1C). 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 (Fig. 1D). 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 (Fig. 1E-F).

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

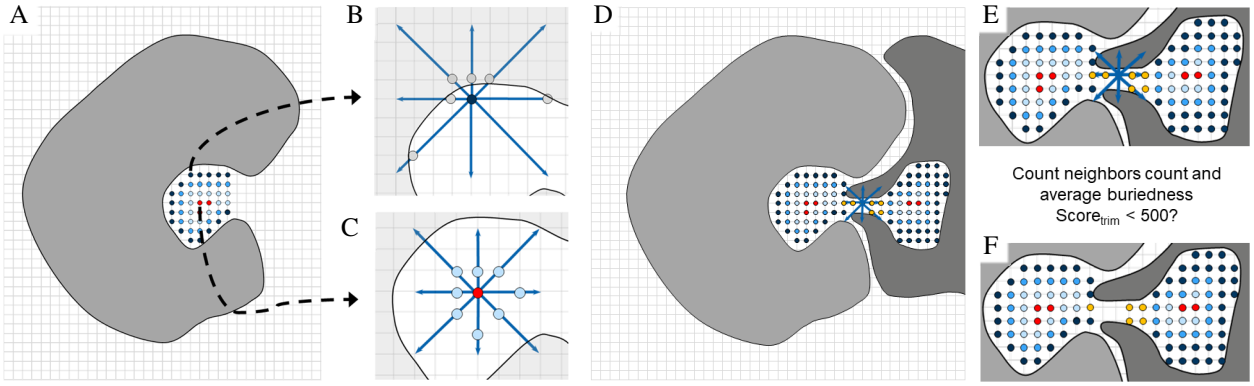


Fig. 1. Visual depiction of the grid-based cavity identification algorithm. (A) The protein, represented as a gray shape, is embedded in a regular 3D grid. (B) The number of contacts between grid points outside of the protein surface and grid points inside the protein surface is investigated and defines putative cavity grid points. (C) A second pass detects grid points surrounded by putative grid points that would have been missed in B. (D) Yellow points connect a cavity in the light gray protein chain and another one in the dark gray chain. (E) For each cavity point, the count of neighbors and the average buriedness are measured in order to calculate the trim score. (F) 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 protein 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 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.

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. A total of 190,080 combinations of parameters was optimized for cavity detection and to avoid overspanning. Details can be found in the Supplementary Material items S1 and S2.

2.2 Validation sets for cavity identification

We assembled different datasets extracted from literature sources, *i.e.*, Kahraman *et al* (Kahraman *et al.*, 2007), Huang and Schroeder (Huang and Schroeder, 2006), the 198 drug-target set of MetaPocket (Zhang *et al.*, 2011), the DUD-e 102 targets (Mysinger *et al.*, 2012); databases, *i.e.*, scPDB (Desaphy *et al.*, 2015) and PDBbind (Liu *et al.*, 2015); 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 Availability paragraph).

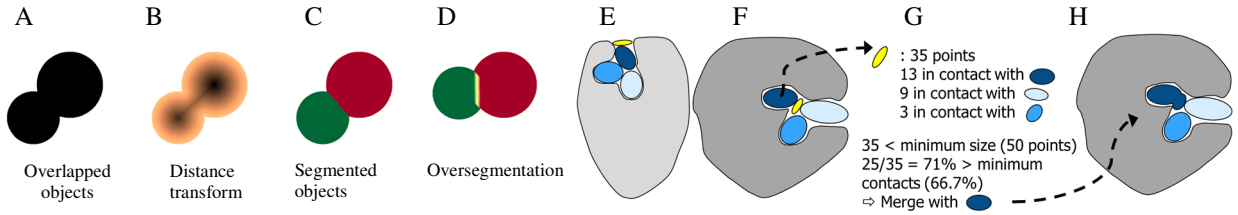
These datasets vary by size, *viz.*, from few dozens in the literature sets to more than 11,000 in scPDB database, and in their scope and composition. The use of multiple datasets is aimed at detecting any particularity arising in one dataset. There is a discrepancy between some of the numbers in the published cavity identification validation sets and our data. For example, the 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 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 count inconsistencies. 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(Huang and Schroeder, 2006) 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.

2.3 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 (Beucher, 1994), which segments cavities into subgroups. Seed points are separated by at least 3 Å, 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 (Fig. 2A-D).

The watershed algorithm tends to over-segment images (Beucher, 1994). 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 are 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 laying on top of another subcavity (Fig. 2E), or interstitial and disk-shaped between several subcavities (Fig. 2F). Image segmentation routines are performed with scikit-image (0.16.2).

Fig. 2. 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. (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 rationale to detect potentially spurious subcavities and identify its merging partner. (H) Result of (G) on (E). In both (D) and (E), the yellow subcavity is merged with the dark blue one.

In order to qualitatively assess the relevance of the subcavity decomposition tool, we assembled a carefully hand-picked dataset of 59 proteins 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 of 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.

3. RESULTS AND DISCUSSION

3.1 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 (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 problematic PDB structures. More specifically, 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 manual curation of all of these structures is beyond the scope of this work. Interestingly, restricting the PDBBind dataset to high affinity complexes with a 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%**

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(Huang and Schroeder, 2006) to compare the performance of CAVIAR to 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 (Tripathi and Kellogg, 2010), DoGSite(Volkamer, Griewel, *et al.*, 2010)

and Fpocket (Le Guilloux *et al.*, 2009). CAVIAR fails on three cases of the bound dataset, all three are very exposed ligand on flat surfaces of the protein (Supplementary 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 *(Tripathi and Kellogg, 2010)*	**83%**	**85%**	**90%**	**94%**
CAVIAR*	**77%**	**88%**	**85%**	**94%**
DoGSite *(Volkamer, Griewel,* et al.*, 2010)*	**71%**	**83%**	**92%**	**92%**
Fpocket* *(Le Guilloux* et al.*, 2009)*	**69%**	**83%**	**94%**	**92%**
LSite *(Volkamer, Griewel,* et al.*, 2010)*	**75%**	**75%**	**85%**	**88%**
PocketPicker *(Weisel* et al.*, 2007)*	**69%**	**72%**	**85%**	**85%**
DSite *(Volkamer, Griewel,* et al.*, 2010)*	**65%**	**69%**	**77%**	**79%**
LIGSITE *(Hendlich* et al.*, 1997)*	**58%**	**69%**	**75%**	**87%**
CAST *(Liang* et al.*, 1998)*	**58%**	**67%**	**75%**	**83%**
PASS *(Brady and Stouten, 2000)*	**60%**	**63%**	**71%**	**81%**
SURFNET *(Laskowski, 1995)*	**52%**	**54%**	**75%**	**78%**

Values of all algorithms except CAVIAR were extracted from (Volkamer, Griewel, *et al.*, 2010). CAVIAR's success values were calculated with the definition of (Volkamer, Griewel, *et al.*, 2010). * indicates open-source software available at the time of the study.

Cavities are ranked according to their cavity scores, which is an estimate of a cavity's 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. In addition, it may be unsettling for the user to get an unsorted list of results. The software also provides a separate ligandability score (Supplementary Material items S1 and S4). 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 of all, interface cavities between protein chains are often big and will therefore have higher cavity scores (and ranks) than potentially 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 the ligand, which can place the liganded cavity 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.

3.2 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 cavity is not entirely detected, or on the contrary, spurious subcavities are present when the cavity overspans. 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. 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 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 (Pirard and Ertl, 2015; Huth *et al.*, 2007). 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 (Fig. 3A). DoGSite recognizes the two subpockets, but produces a very large cavity and generates four subcavities in total (Fig. 3B). The second successful example is the HIV-1 protease, which contains six well defined subsites, recognizing specifically aminoacid side chains of the peptidic substrate (Ghosh *et al.*, 2016; Munshi *et al.*, 2000). 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 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 S1 and extended S1 pockets (Ghosh *et al.*, 2016). 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) 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 (Fig. 3E). 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 thiophen moieties of the inhibitor (Thal *et al.*, 2016). DoGSite results are similar to CAVIAR, except that it does not segment the orthosteric pocket into subsites (Fig. 3F). The last successful case discussed here 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 phosphates regions, is described by one large subpocket, occupied by the main hinge binding motif of the ligand and its furan substituent (Wood *et al.*, 2004). 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. On the contrary, it detects other connected subpockets far from the ligand binding groove, and significantly overspans (Fig. 3H).

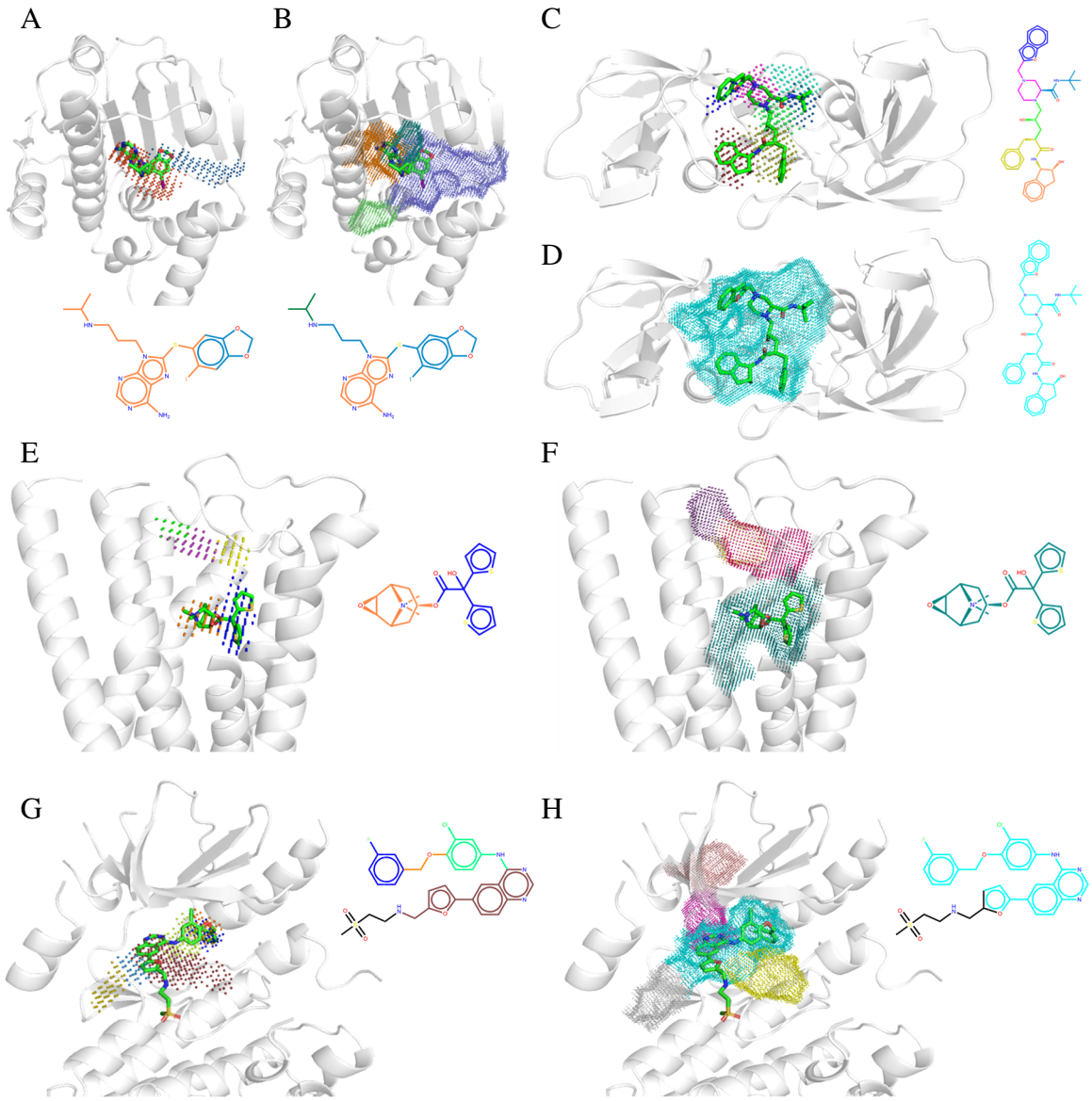


Fig. 3. Examples of successful decomposition by CAVIAR and comparison to DoGSite. In all panels, the 2D structure of the ligand is depicted with a color code corresponding to the subcavity segmentation, or in black if not covered by subcavities. (A) and (B) Chaperone protein hsp90, PDB code 2fwz. (A) The CAVIAR subpocket algorithm correctly identifies the adenine pocket, in orange and the lipophilic pocket, in blue. (B) DoGSite 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 of the cavities are very solvent-exposed, which hinders the detection of the entirety of the cavities (Fig. 4). 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.

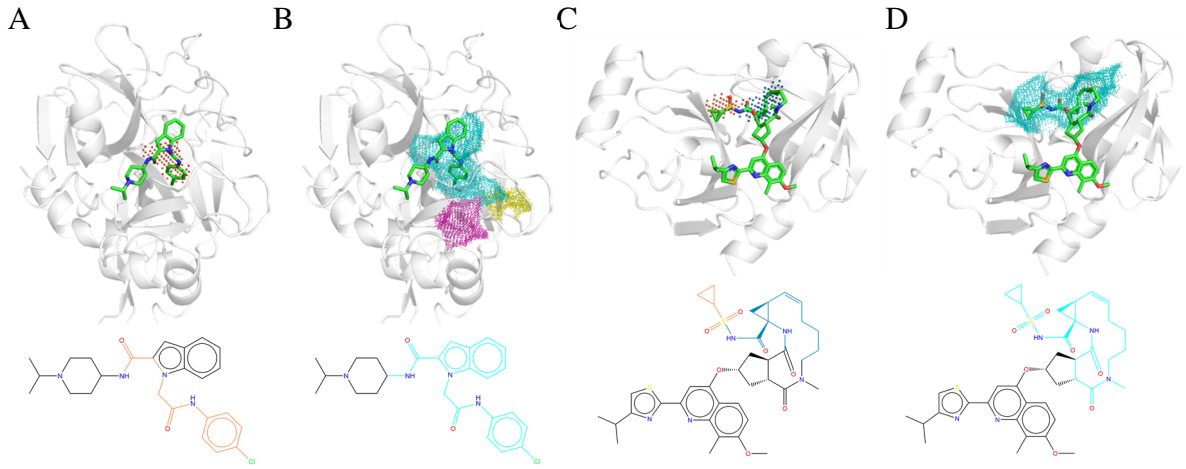


Fig. 4. Examples of unsuccessful decomposition by CAVIAR and comparison to DoGSite. In all panels, the 2D structure of the ligand is depicted with a color code corresponding to the subcavity segmentation, or in black if not covered by subcavities. (A) CAVIAR and (B) DoGSite cavity detection and segmentation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

3.3 Visual interface and command line tool

CAVIAR is available both as a GUI and as a command line tool. The command line tool comes with many options to provide experienced 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; the presence of a ligand can be investigated. There are a lot of user tunable parameters, therefore we developed a website to guide the user with extended information (link in the Availability paragraph). The GUI restricts the options to defaults 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. 5A). The second window relates to the subcavity decomposition and has similar options (Fig. 5B).

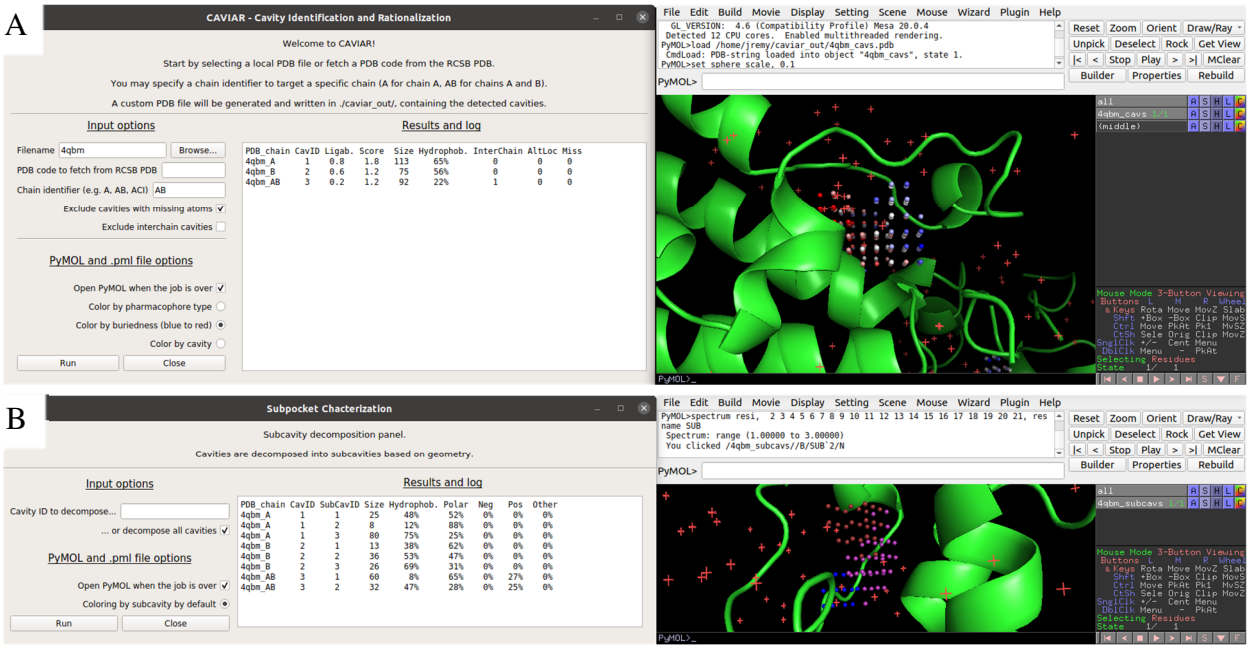


Fig. 5. Visual interface for the CAVIAR cavity detection (A) and subcavity decomposition (B).

3.4 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 better than 2.5 Å and no flag as obsolete or any other warning 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, arguing that ligands fill on average only a third of their binding pockets.(Wirth *et al.*, 2013; Kahraman *et al.*, 2007) 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 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 (Supplementary Table S8).

3.5 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 with a focus on 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 (Fig. 6).

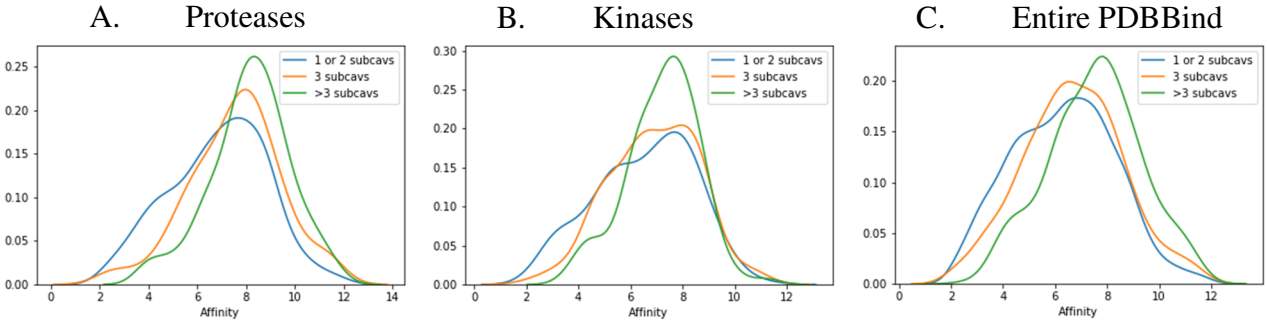


Fig. 6. Distribution of binding affinities expressed as –log(affinity) in function of the numbers of subcavities filled by the ligand. (A) Protease dataset. In blue, ligands filling one or two subcavities (n=453), orange three subcavities (n=154), and green four or more subcavities (n=194). The peak 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 when more subpockets are involved in ligand binding (Table 5). 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 Supplementary 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 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 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.

3.6 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 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 (Bacci *et al.*, 2017; Laio and Gervasio, 2008; Kuzmanic *et al.*, 2020). Crystal contacts, artifacts and protein chain repeats can produce spurious non-productive interchain cavities (Fig. 7A). Significant work has been invested into detecting biologically relevant protein chains contacts (Duarte *et al.*, 2012; Capitani *et al.*, 2016) 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 potentially binding small organic drug-like compounds, which, *de facto*, excludes surface patches such as protein-protein interfaces and very exposed ligand binding

grooves (Fig. 7B). We plan to address this issue in future release. Different sets of parameters could be identified and optimized for detecting surface patches, or even protein-protein interaction interfaces. Key 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 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 (Fig 7C). 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. Provided 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.

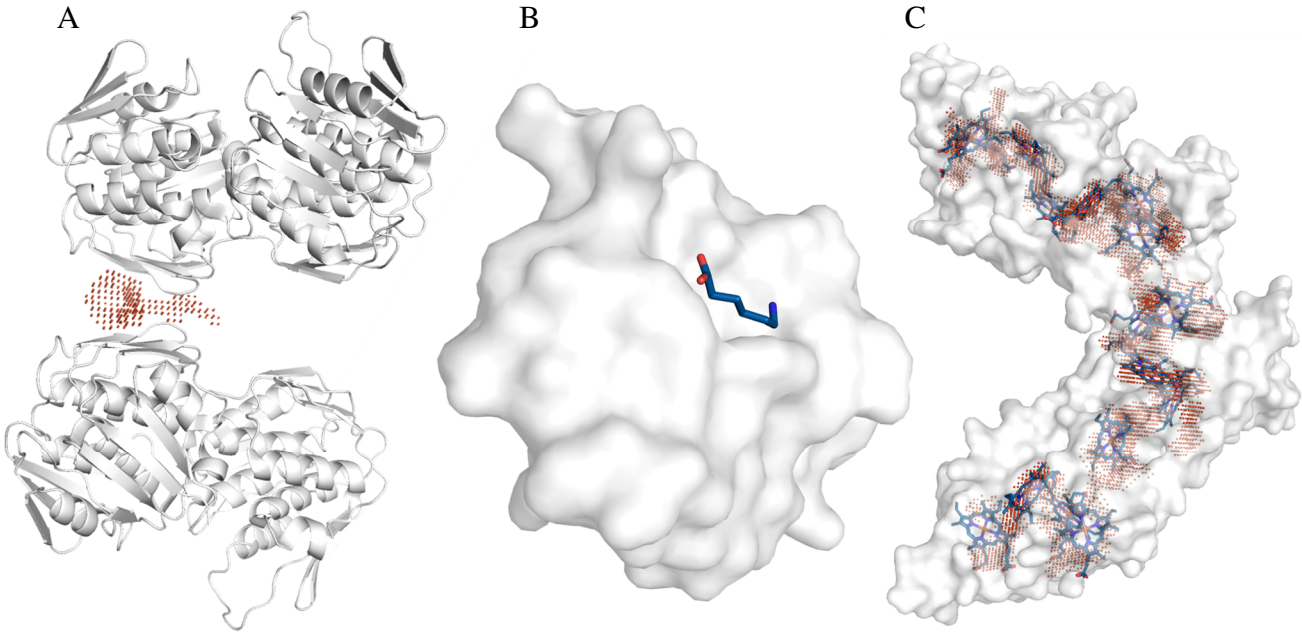


Fig. 7. Representative cases of failure with CAVIAR. (A) Spurious interchain cavity. A cavity, in orange spheres, is found at the interface between two protein chains, in white cartoons, which is a crystal contact and not biologically relevant (PDB code 1ejd). (B) Case of an exposed ligand, in blue sticks, 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.

4. 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, pluggable in automatic workflows 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, requiring pre-processing of the data, 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. CAVIAR is mmCIF-ready, which is important as the PDB format may be retired around 2021 (https://www.ebi.ac.uk/pdbe/about/news/mandatory-mmcif-format-crystallographic-depositions-pdb-0) as well as one of the few (Yuan *et al.*, 2020) pocket detection tools to incorporate a molecular dynamics trajectory parser, and the only open-source tool for subpocket characterization solely based on the protein. 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 Availability paragraph 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, file parsing, cavity and subcavity detection routines 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 (Jalencas and Mestres, 2013). 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 (Keefer and Chang, 2017). 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. The deconstruction of pockets into subcavities may help for partial cavity matching in the context of cavity comparison (Krotzky *et al.*, 2014). Our analysis of the PDB 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 (Hajduk *et al.*, 1997; Erlanson *et al.*, 2016), we found that the binding affinity of small molecule ligands scales 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.

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 the ProDy development team and generally all contributors to open source codes for their crucial work.

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.

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

REFERENCES

Al-Gharabli,S.I. *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.

Anand,P. *et al.* (2011) Structural Annotation of Mycobacterium tuberculosis Proteome. *PLOS ONE*, **6**, e27044.

Bacci,M. *et al.* (2017) Focused conformational sampling in proteins. *J. Chem. Phys.*, **147**, 195102.

Bartolowits,M. and Davisson,V.J. (2016) Considerations of Protein Subpockets in Fragment-Based Drug Design. *Chem. Biol. Drug Des.*, **87**, 5–20.

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

Bliznyuk,A.A. and Gready,J.E. (1998) Identification and energetic ranking of possible docking sites for pterin on dihydrofolate reductase. *J. Comput. Aided Mol. Des.*, **12**, 325–333.

Brady,G.P. and Stouten,P.F.W. (2000) Fast prediction and visualization of protein binding pockets with PASS. *J. Comput. Aided Mol. Des.*, **14**, 383–401.

Capitani,G. *et al.* (2016) Understanding the fabric of protein crystals: computational classification of biological interfaces and crystal contacts. *Bioinformatics*, **32**, 481–489.

Capra,J.A. *et al.* (2009) Predicting Protein Ligand Binding Sites by Combining Evolutionary Sequence Conservation and 3D Structure. *PLOS Comput. Biol.*, **5**, e1000585.

Chan,A.W.E. *et al.* (2010) Chemical Fragments that Hydrogen Bond to Asp, Glu, Arg, and His Side Chains in Protein Binding Sites. *J. Med. Chem.*, **53**, 3086–3094.

Desaphy,J. *et al.* (2012) Comparison and Druggability Prediction of Protein–Ligand Binding Sites from Pharmacophore-Annotated Cavity Shapes. *J. Chem. Inf. Model.*, **52**, 2287–2299.

Desaphy,J. *et al.* (2015) sc-PDB: a 3D-database of ligandable binding sites—10 years on. *Nucleic Acids Res.*, **43**, D399–D404.

Duarte,J.M. *et al.* (2012) Protein interface classification by evolutionary analysis. *BMC Bioinformatics*, **13**, 334.
Durrant,J.D. *et al.* (2011) CrystalDock: A Novel Approach to Fragment-Based Drug Design. *J. Chem. Inf. Model.*, **51**, 2573–2580.
Ehrt,C. *et al.* (2016) Impact of Binding Site Comparisons on Medicinal Chemistry and Rational Molecular Design. *J. Med. Chem.*, **59**, 4121–4151.
Erlanson,D.A. *et al.* (2016) Twenty years on: the impact of fragments on drug discovery. *Nat. Rev. Drug Discov.*, **15**, 605–619.
Ghosh,A.K. *et al.* (2016) Recent Progress in the Development of HIV-1 Protease Inhibitors for the Treatment of HIV/AIDS. *J. Med. Chem.*, **59**, 5172–5208.
Goodford,P.J. (1985) A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. *J. Med. Chem.*, **28**, 849–857.
Hajduk,P.J. *et al.* (1997) Discovering High-Affinity Ligands for Proteins. *Science*, **278**, 497–499.
Halgren,T.A. (2009) Identifying and Characterizing Binding Sites and Assessing Druggability. *J. Chem. Inf. Model.*, **49**, 377–389.
Hendlich,M. *et al.* (1997) LIGSITE: automatic and efficient detection of potential small molecule-binding sites in proteins. *J. Mol. Graph. Model.*, **15**, 359–363.
Huang,B. (2009) MetaPocket: A Meta Approach to Improve Protein Ligand Binding Site Prediction. *OMICS J. Integr. Biol.*, **13**, 325–330.
Huang,B. and Schroeder,M. (2006) LIGSITEcsc: predicting ligand binding sites using the Connolly surface and degree of conservation. *BMC Struct. Biol.*, **6**, 19.
Huth,J.R. *et al.* (2007) Discovery and Design of Novel HSP90 Inhibitors Using Multiple Fragment-based Design Strategies. *Chem. Biol. Drug Des.*, **70**, 1–12.
Jalencas,X. and Mestres,J. (2013) Chemoisosterism in the Proteome. *J. Chem. Inf. Model.*, **53**, 279–292.
Kahraman,A. *et al.* (2007) Shape Variation in Protein Binding Pockets and their Ligands. *J. Mol. Biol.*, **368**, 283–301.
Kalidas,Y. and Chandra,N. (2008) PocketDepth: a new depth based algorithm for identification of ligand binding sites in proteins. *J. Struct. Biol.*, **161**, 31–42.
Kalliokoski,T. *et al.* (2013) Subpocket Analysis Method for Fragment-Based Drug Discovery. *J. Chem. Inf. Model.*, **53**, 131–141.
Kawabata,T. (2010) Detection of multiscale pockets on protein surfaces using mathematical morphology. *Proteins Struct. Funct. Bioinforma.*, **78**, 1195–1211.
Kawabata,T. and 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.
Keefer,C.E. and Chang,G. (2017) The use of matched molecular series networks for cross target structure activity relationship translation and potency prediction. *MedChemComm*, **8**, 2067–2078.
Kinoshita,K. *et al.* (2002) Identification of protein functions from a molecular surface database, eF-site. *J. Struct. Funct. Genomics*, **2**, 9–22.
Konc,J. *et al.* (2013) Structure-Based Function Prediction of Uncharacterized Protein Using Binding Sites Comparison. *PLOS Comput. Biol.*, **9**, e1003341.
Kooistra,A.J. *et al.* (2015) Structure-Based Prediction of G-Protein-Coupled Receptor Ligand Function: A β-Adrenoceptor Case Study. *J. Chem. Inf. Model.*, **55**, 1045–1061.
Krivák,R. and Hoksza,D. (2018) P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *J. Cheminformatics*, **10**, 39.

Krotzky,T. *et al.* (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.

Kuhn,D. *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.

Kuzmanic,A. *et al.* (2020) Investigating Cryptic Binding Sites by Molecular Dynamics Simulations. *Acc. Chem. Res.*

Laio,A. and Gervasio,F.L. (2008) Metadynamics: a method to simulate rare events and reconstruct the free energy in biophysics, chemistry and material science. *Rep. Prog. Phys.*, **71**, 126601.

Laskowski,R.A. (1995) SURFNET: A program for visualizing molecular surfaces, cavities, and intermolecular interactions. *J. Mol. Graph.*, **13**, 323–330.

Laurie,A.T.R. and Jackson,R.M. (2005) Q-SiteFinder: an energy-based method for the prediction of protein–ligand binding sites. *Bioinformatics*, **21**, 1908–1916.

Le Guilloux,V. *et al.* (2009) Fpocket: An open source platform for ligand pocket detection. *BMC Bioinformatics*, **10**, 168.

Levitt,D.G. and Banaszak,L.J. (1992) POCKET: A computer graphies method for identifying and displaying protein cavities and their surrounding amino acids. *J. Mol. Graph.*, **10**, 229–234.

Lewis,R.A. (1989) Determination of clefts in receptor structures. *J. Comput. Aided Mol. Des.*, **3**, 133–147.

Liang,J. *et al.* (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.

Liu,Z. *et al.* (2015) PDB-wide collection of binding data: current status of the PDBbind database. *Bioinformatics*, **31**, 405–412.

Macari,G. *et al.* (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.

Marchand,J.-R. *et al.* (2017) Discovery of Inhibitors of Four Bromodomains by Fragment-Anchored Ligand Docking. *J. Chem. Inf. Model.*, **57**, 2584–2597.

Marchand,J.-R. and Caflisch,A. (2018) In silico fragment-based drug design with SEED. *Eur. J. Med. Chem.*, **156**, 907–917.

Miranker,A. and Karplus,M. (1991) Functionality maps of binding sites: A multiple copy simultaneous search method. *Proteins Struct. Funct. Bioinforma.*, **11**, 29–34.

Möller-Acuña,P. *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.

Munshi,S. *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.

Mysinger,M.M. *et al.* (2012) Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *J. Med. Chem.*, **55**, 6582–6594.

Nayal,M. and 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.

Ngan,C.H. *et al.* (2012) FTMAP: extended protein mapping with user-selected probe molecules. *Nucleic Acids Res.*, **40**, W271–W275.

Oliveira,S.H. *et al.* (2014) KVFinder: steered identification of protein cavities as a PyMOL plugin. *BMC Bioinformatics*, **15**, 197.
Peters,K.P. *et al.* (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.
Pirard,B. and Ertl,P. (2015) Evaluation of a Semi-Automated Workflow for Fragment Growing. *J. Chem. Inf. Model.*, **55**, 180–193.
Schirris,T.J.J. *et al.* (2015) Mitochondrial ADP/ATP exchange inhibition: a novel off-target mechanism underlying ibipinabant-induced myotoxicity. *Sci. Rep.*, **5**, 1–12.
Schumann,M. and Armen,R.S. (2013) Identification of Distant Drug Off-Targets by Direct Superposition of Binding Pocket Surfaces. *PLOS ONE*, **8**, e83533.
Simões,T. *et al.* (2017a) Geometric Detection Algorithms for Cavities on Protein Surfaces in Molecular Graphics: A Survey. *Comput. Graph. Forum*, **36**, 643–683.
Simões,T. *et al.* (2017b) Geometric Detection Algorithms for Cavities on Protein Surfaces in Molecular Graphics: A Survey. *Comput. Graph. Forum J. Eur. Assoc. Comput. Graph.*, **36**, 643–683.
Simões,T.M.C. and Gomes,A.J.P. (2019) CavVis—A Field-of-View Geometric Algorithm for Protein Cavity Detection. *J. Chem. Inf. Model.*, **59**, 786–796.
Tang,G.W. and Altman,R.B. (2014) Knowledge-based Fragment Binding Prediction. *PLOS Comput. Biol.*, **10**, e1003589.
Thal,D.M. *et al.* (2016) Crystal structures of the M1 and M4 muscarinic acetylcholine receptors. *Nature*, **531**, 335–340.
Till,M.S. and Ullmann,G.M. (2010) McVol - A program for calculating protein volumes and identifying cavities by a Monte Carlo algorithm. *J. Mol. Model.*, **16**, 419–429.
Tripathi,A. and Kellogg,G.E. (2010) A novel and efficient tool for locating and characterizing protein cavities and binding sites. *Proteins Struct. Funct. Bioinforma.*, **78**, 825–842.
Volkamer,A., Griewel,A., *et al.* (2010) Analyzing the Topology of Active Sites: On the Prediction of Pockets and Subpockets. *J. Chem. Inf. Model.*, **50**, 2041–2052.
Volkamer,A. *et al.* (2012) DoGSiteScorer: a web server for automatic binding site prediction, analysis and druggability assessment. *Bioinformatics*, **28**, 2074–2075.
Volkamer,A. *et al.* (2016) Identification and Visualization of Kinase-Specific Subpockets. *J. Chem. Inf. Model.*, **56**, 335–346.
Volkamer,A. *et al.* (2018) Prediction, Analysis, and Comparison of Active Sites. In, *Applied Chemoinformatics*. John Wiley & Sons, Ltd, pp. 283–311.
Volkamer,A., Grombacher,T., *et al.* (2010) Where are the boundaries? Automated pocket detection for druggability studies. *J. Cheminformatics*, **2**, P11.
Wang,L. *et al.* (2011) Residue Preference Mapping of Ligand Fragments in the Protein Data Bank. *J. Chem. Inf. Model.*, **51**, 807–815.
Weber,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.
Weisel,M. *et al.* (2007) PocketPicker: analysis of ligand binding-sites with shape descriptors. *Chem. Cent. J.*, **1**, 7.
Westbrook,J.D. and Burley,S.K. (2019) How Structural Biologists and the Protein Data Bank Contributed to Recent FDA New Drug Approvals. *Structure*, **27**, 211–217.
Willmann,D. *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.

Wirth,M. *et al.* (2013) Protein pocket and ligand shape comparison and its application in virtual screening. *J. Comput. Aided Mol. Des.*, **27**, 511–524.

Wood,D.J. *et al.* (2012) Pharmacophore Fingerprint-Based Approach to Binding Site Subpocket Similarity and Its Application to Bioisostere Replacement. *J. Chem. Inf. Model.*, **52**, 2031–2043.

Wood,E.R. *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.

Xie,Li *et al.* (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.

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

Yu,J. *et al.* (2010) Roll: a new algorithm for the detection of protein pockets and cavities with a rolling probe sphere. *Bioinformatics*, **26**, 46–52.

Yuan,J.-H. *et al.* (2020) Druggability Assessment in TRAPP Using Machine Learning Approaches. *J. Chem. Inf. Model.*, **60**, 1685–1699.

Zhang,Z. *et al.* (2011) Identification of cavities on protein surface using multiple computational approaches for drug binding site prediction. *Bioinformatics*, **27**, 2083–2088.