



HAL
open science

Cavity Characterization in Supramolecular Cages

João Guerra, Luiz Alves, Didier Bourissou, Paulo Lopes-De-Oliveira, György Szalóki

► **To cite this version:**

João Guerra, Luiz Alves, Didier Bourissou, Paulo Lopes-De-Oliveira, György Szalóki. Cavity Characterization in Supramolecular Cages. *Journal of Chemical Information and Modeling*, In press, 10.1021/acs.jcim.3c00328 . hal-04101561

HAL Id: hal-04101561

<https://hal.science/hal-04101561v1>

Submitted on 20 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Cavity Characterization in Supramolecular Cages

João V. S. Guerra,^a Luiz F. G. Alves,^a Didier Bourissou,^b Paulo S. Lopes-de-Oliveira,^{a} György Szalóki^{b*}*

AUTHOR ADDRESS (a) Brazilian Center for Research in Energy and Materials (CNPEM), Brazilian Biosciences National Laboratory (LNBio), Rua Giuseppe Máximo Scolfaro, 10000, Bosque das Palmeiras, Campinas, SP, 13083-100, Brazil. (b) Laboratoire Hétérochimie Fondamentale et Appliquée (LHFA, UMR 5069), CNRS, Université Toulouse III – Paul Sabatier, 118 Route de Narbonne, Toulouse 31062, Cedex 09, France.

KEYWORDS Cavity characterization, Supramolecular cages, MOCs

ABSTRACT

Confining molecular guests within artificial hosts has provided a major driving force in the rational design of supramolecular cages with tailored properties. Over the last 30 years, a set of design strategies have been developed that enabled the controlled synthesis of a myriad of cages. Recently, there has been a growing interest in involving *in silico* methods in this toolbox. Cavity shape and size are important parameters that can be easily accessed by inexpensive geometric algorithms. Although these algorithms are well developed for the detection of non-artificial cavities (e.g., enzymes), they are not routinely used for the rational design of supramolecular cages. In order to test the capabilities of this tool, we have evaluated the performance and characteristics of 7 different cavity characterization software in the context of 22 analogues of well-known supramolecular cages. Among the tested software, KVFinder project and Fpocket proved to be the most adapted to characterize supramolecular cavities. With the results of this work, we aim to popularize this underused technique within the supramolecular community.

INTRODUCTION

1. General Introduction

Enzymes are crucial to all living systems: Indeed, more than 99% of biologically relevant reactions are mediated by natural proteins. They are truly remarkable catalysts that can attain exceptional rate enhancement in many biological transformations.¹ Reactive intermediates and transition states are stabilized at the active site of enzymes, thus providing a lower energy pathway for reactions to take place. This active site is buried (confined) within a cavity, called an enzyme pocket.² This confinement offers different advantages: (1) rate acceleration is achieved through predisposition and increased local concentration of reactive centers; and (2) steric constraints alter the normal (solution phase) selectivity of certain reactions. Consequently, enzymes are capable of efficiently synthesizing complex biomolecules with unparalleled stereocontrol. Taking advantage of their efficiency, they have been implemented in synthetic chemistry on both laboratory and industrial scales. Unfortunately, their synthetic use is limited, as this efficiency is met with high substrate specificity or in other words: poor substrate scope. Fortunately, the principle of confinement is transposable³ from biological systems to any artificially designed host, with cavities of various shapes and sizes: zeolites,⁴ metal-organic frameworks (MOFs),⁵ covalent organic frameworks (COFs)^{5b} and supramolecular cages. More importantly, many supramolecular cages have been prepared over the past 25 years using different strategies: covalent, H-bonded and metal organic cages (MOCs).⁶ They have been shown to encapsulate (confine) reactive species within their cavities, just like enzymes, and there is now solid evidence that confinement within these artificial cavities influences the rate and selectivity of various reactions.⁷

In recent years, a large amount of experimental data has been accumulated, which has raised interest in gaining a more detailed picture of the encapsulation process *in silico*.⁸ Understanding the underlying factors that drive the encapsulation of reactive intermediates is primordial for the rational design of new supramolecular cages with improved catalytic properties. Indeed, recent efforts have been devoted to develop freely available tools (e.g., cgbind) for the computer-aided prediction of binding affinities.⁹ However, encapsulation is a complex process, both from thermodynamic and kinetic point of view. Classical simulations (e.g., molecular dynamics, Grand Canonical Monte Carlo) and DFT computations are useful tools to study these complex

molecular systems. However, such complex and large molecular systems at density functional theory (DFT) level are time-consuming. Recent work by Major *et al.* and Ujaque and Maréchal *et al.* have focused on this problem by integrating quantum mechanics concepts to molecular dynamics simulations.¹⁰ These calculations have provided useful insights about encapsulation in artificial hosts, but their use in a predictive manner is still premature today. On the other hand, cavity size and shape are invaluable parameters, which can serve as good estimates of the steric feasibility of encapsulation processes. Indeed, the binding energies showed a good correlation with the guest size (V_{guest}) and $V_{\text{guest}}/V_{\text{cavity}}$ ratio according to the work of Ujaque and Maréchal *et al.*^{10a} In addition, these parameters can be easily accessed by inexpensive *in silico* geometric algorithms, which have been originally developed to detect cavities in protein systems. Although the close analogy between enzyme pockets and artificial cavities is evident (*vide supra*), these methods have not been routinely used in the context of supramolecular cages.¹¹

Therefore, the aim of this work is to present a state of the art of different methods and software available to detect and describe cavities in supramolecular cages, in order to make these *in silico* approaches more accessible to the supramolecular community. The cavity volume is an important parameter for the rational design of new supramolecular cages with improved catalytic properties.

2. Cavity detection in computational chemistry

First, we are going to provide a brief introduction to the different strategies for detecting cavities in biomolecules (e.g., proteins, DNA, RNA, and lipid membranes). The discussion will only focus on the necessary background and terminology that will serve readers in the following parts of this section. Our goal is to identify robust software that can characterize cavities in supramolecular cages and provide a comprehensive analysis of them.¹² Selected software will be discussed below, keeping a simplified description along with visual aids to help novice readers.

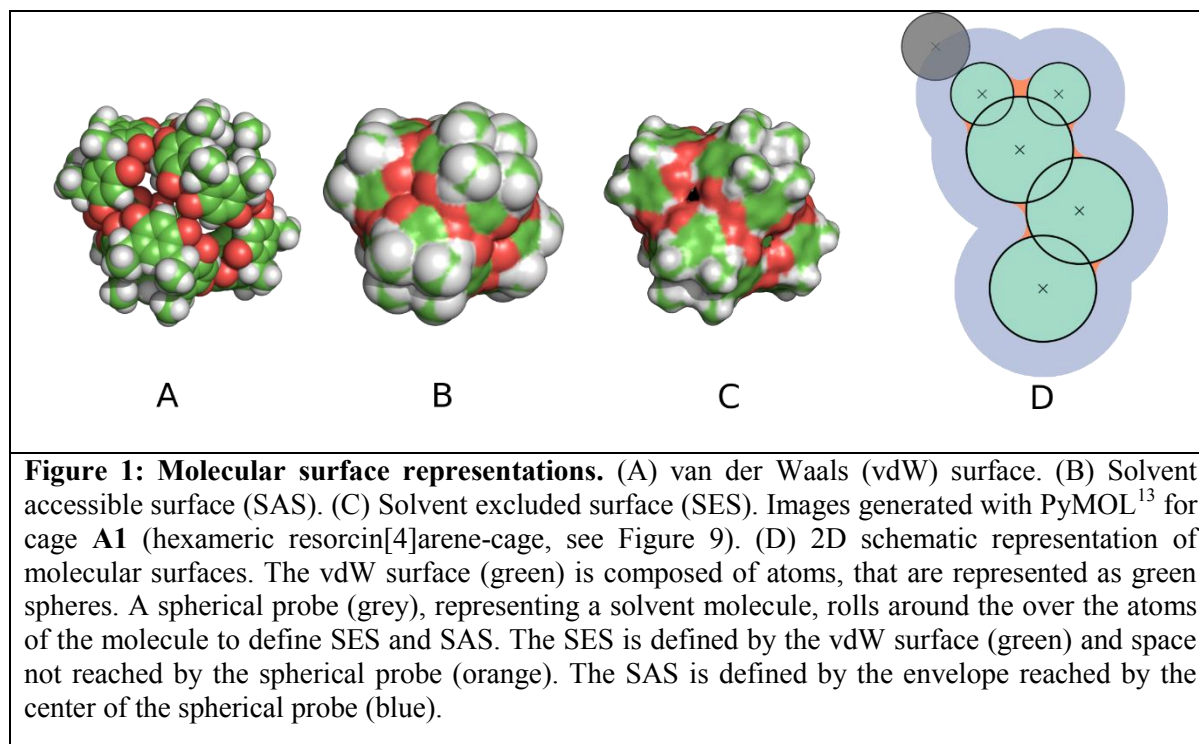
2.1 Surface representations

For modeling purposes, the host (biomolecule) and the guest (substrate) are usually described by a hard sphere model, that only considers atomic positions and radii to represent the molecular structure.^{9a,b} Based on this, the most common mathematical formulations of molecular surfaces are (Figure 1):

(A) *van der Waals (vdW) surface* represents each atom by a sphere whose radius is proportional to its vdW radius. The vdW surface is represented as the union of those spherical atoms;

(B) *solvent accessible surface (SAS)* represents the regions of a molecule that can be accessed by a solvent molecule (e.g., a water molecule), which is approximated by a spherical probe;

(C) *solvent excluded surface (SES)* is similar to SAS, but the outer shell of the probe is considered, instead of the probe center.



2.2 Cavity detection methods

Over the last 40 years, several computational methods have been developed for the characterization of protein surfaces. These methods can be classified into three main categories:

- ***Evolutionary-based methods*** are based on searching for conserved residues in multiple sequence alignments and information from known binding site profiles;
- ***Energy-based methods*** identify binding sites from energetic interaction between the target biomolecule and a probe molecule, usually a chemical group;
- ***Geometry-based methods*** identify cavities by analyzing the geometrical characteristics of the molecular surface.

Methods in each category have their own benefits and drawbacks.^{9b} Evolutionary algorithms strongly rely on sequence information or active binding site databases and the quality of the alignment procedure, while energetic methods depend on filtering procedures, force field parametrizations, and scoring functions used. On the other hand, geometrical detection methods are relatively simple, straightforward and do not require any non-geometric knowledge, only the protein structural data file, containing the Cartesian coordinates of the atoms, that can be easily accessed in the *Worldwide Protein Data Bank* (wwPDB).¹⁴ Once the atoms' coordinates are available, geometric methods should be able to represent any kind of biomolecule. Although purely geometric methods are efficient in identifying all types of cavities of a target molecule, identifying those that are functionally relevant poses a problem. However, cavity characterization in terms of well-chosen physicochemical properties can lead to the identification of functionally relevant cavities, i.e., binding sites for a specific set of ligands.

In general, geometry-based algorithms are the most frequently used to detect protein cavities. While evolutionary-based methods would be limited to proteins because they rely on principles of biological evolution, energy-based methods could be applicable but would require fine-tuning of force field parameters tailored to supramolecular cages. Since supramolecular cages may have distinct properties compared to proteins, methods that rely on geometrical prior knowledge (e.g., 3D Cartesian coordinates, atom size) are desirable. Therefore, the aim of our article is to evaluate and compare their efficiency in the context of supramolecular cages.

2.2.1 Geometry-based methods

A comprehensive taxonomy of geometry-based cavity detection approaches includes grid-, probe-, tessellation-, surface-based techniques, and their combination.^{9a,b,15} The technique employed to extract cavities from the molecular structure is mainly what differentiates them.

- **Grid-based algorithms** (e.g., POVME 3.0) represent a set of atoms as discrete points, usually using an axis-aligned 3D grid such as a scalar field, i.e., density map, where each discrete point is an integer or a boolean. These grid maps are used to group relevant empty (non-solute) grid points into cavities using voxel clustering algorithms. Typically, these methods employ simple data structures, which can represent a collection of data at a discrete point and identify cavities in an automated procedure. However, geometrical accuracy, computation time, and memory consumption are strongly dependent on grid resolution, i.e., grid-spacing sensitivity. Besides that, these methods are not rotation invariant, which means the orientation of a given molecule slightly impacts the accuracy, i.e., orientation sensitivity.

- **Probe-based algorithms** (e.g., pywindow, PHECOM¹⁶) use a set of atoms, considering their 3D coordinates and vdW radii, to represent the molecular surface, which is scanned by one or more probes, usually hard spheres, to investigate their accessibility levels. This technique can detect any type of cavity and is related to the spatial extent of putative ligands; however, it may struggle to unambiguously find and delineate the cavity-solvent boundaries, i.e., mouth opening ambiguity.

- **Tessellation-based algorithms** (e.g., Fpocket, CAVER 3.0) rely on computational geometry techniques, e.g., alpha-shapes, beta-shapes, Voronoi diagrams, and Apollonius graphs. Specifically, alpha-shapes and Voronoi tessellation use atomic centers, implicitly constant-radius spheres to model atoms, while beta-shapes and Apollonius-based methods depend on varying-radius spheres to model these atoms, which are explored to identify cavities. Usually, these methods do not depend on any molecular surface information to detect cavities but may struggle to identify the correct location of the binding site, identify and delineate the cavity-solvent boundaries and define the number of surface atoms.

- **Surface-based algorithms** (e.g., NSA,¹⁷ MSPocket¹⁸) do not employ a hard sphere model, but a molecular surface model, e.g., vdW, SES, SAS, and ligand excluded surface, which

defines the molecular interface and its environment. Molecular interface analysis identifies cavities based on the accessibility to a specific solvent or a ligand. In it, cavity detection operates in an automated manner, like grid-based methods, but does not suffer from mouth opening ambiguity. Also, they can sometimes suffer to detect all type of cavities and their full extent.

Furthermore, the above group of techniques can be combined (e.g., parKVFinder, pyKVFinder, ghecom, MoloVol, 3V,¹⁹ CAVER) to enhance their capabilities and reduce some of their deficiencies to achieve a more robust technique.

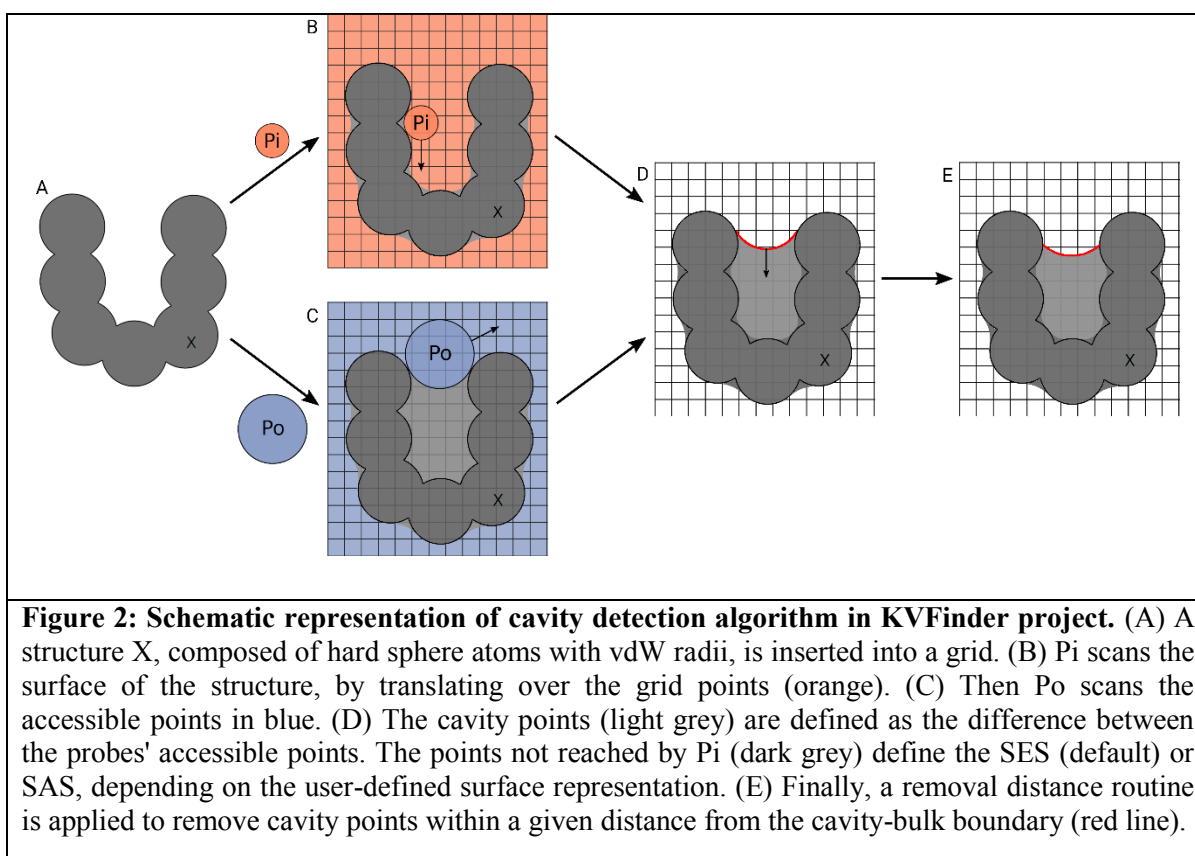
2.3 Cavity characterization in supramolecular cages

The vast majority of cavity detection software has been originally developed for proteins. While these algorithms are robust enough to describe and analyze any molecular system (e.g., DNA, RNA, inorganic materials, and supramolecular cages), to date, only a handful of software has been applied to supramolecular cages in order to evaluate structural characteristics such as shape/volume of their cavities and openings.¹⁰ We have carefully evaluated this collection of software and those that have not been previously used in the context of supramolecular cages. According to our evaluation criteria, the software should be able to detect cavities in supramolecular cages and calculate their volumes (see results in section 2.3), free of charge, well-documented, well-supported by the developers, and accessible from newcomers to experienced users (see SI, section 1). Based on these criteria, we have identified 7 cavity detection software (SI, Table S1). A brief description of their algorithms and functionalities can be found hereafter.

2.3.1 KVFinder project

The KVFinder project uses grid-and-sphere-based methods to detect, characterize and visualize biomolecular cavities. The cavity detection procedure applies a dual-probe algorithm based on the theory of mathematical morphology (Figure 2).²⁰ Basically, a small probe (Probe In - Pi) and a large probe (Probe Out - Po) are translated over the grid points, defining cavities as the non-overlapping regions scanned by these probes (Pi-Po). However, the KVFinder software,

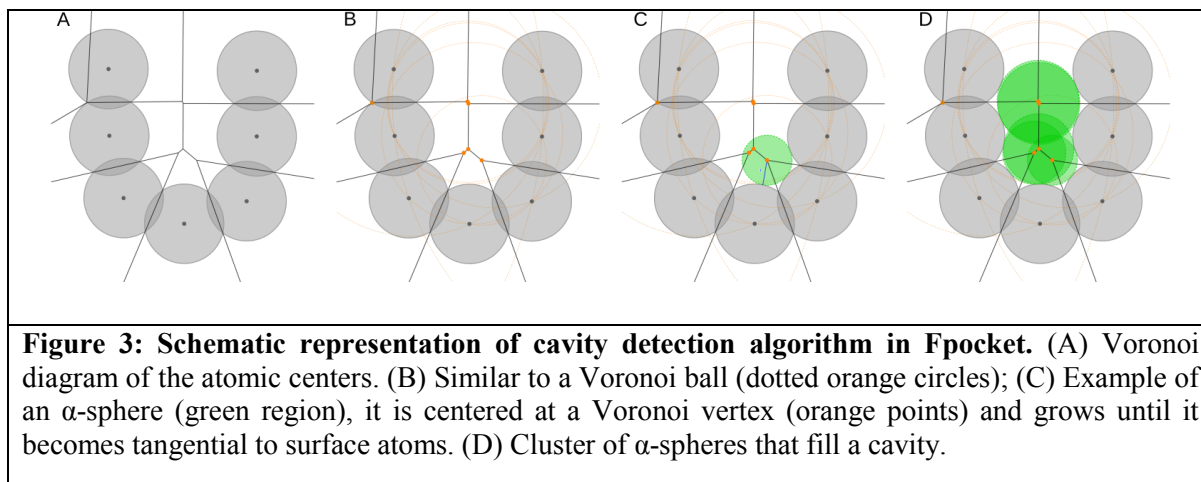
originally published in 2014, is deprecated, and the KVFinder project is currently composed of two independent software: Parallel KVFinder (parKVFinder)²¹ and Python-C Parallel KVFinder (pyKVFinder)¹¹ parKVFinder is available with an easy-to-use PyMOL plugin with an intuitive graphical user interface that allows users to explore customizable parameters for cavity detection and characterization. On the other hand, pyKVFinder is available as a Python package with efficient scripting routines built on easy-to-handle data structures, facilitating complex structural data analysis. Thus, pyKVFinder can be applied as a building block for data science, drug design, and drug discovery applications. In both, cavity characterization covers spatial (volume, area, and shape), constitutional (surrounding residues), hydrophathy, and depth characterizations.



2.3.2 Fpocket

Fpocket²² is a tessellation-based method that performs pocket detection based on the alpha

sphere concept (Figure 3).²³ The method determines the set of alpha spheres from the target structure, using qhull package²⁴, and eliminates spheres outside a minimum and maximum radius size. Cavities are clusters of alpha spheres, that are formed based on proximity and neighborhood relationships, with uninteresting cavities removed from further analysis. The remaining cavities are evaluated using a set of dpocket descriptors and ranked according to their putative capacity to bind a small molecule.



2.3.3 pywindow

pywindow²⁵ is a Python package that uses a sphere-based algorithm to analyze porous molecular materials (Figure 4). Cavity and window detection start by calculating the center of mass of the molecular structure. For characterizing cavity volume, a sphere is fitted into the pore without overlapping any neighboring atom. The sphere volume corresponds to the cavity volume. For window detection, the largest sphere is fitted into each cavity opening (i.e., window). The package uses uniformly distributed sampling points around the target pore and inspects vectors connecting these points to the center of mass to identify the largest sphere that does not overlap with any atom along each vector.

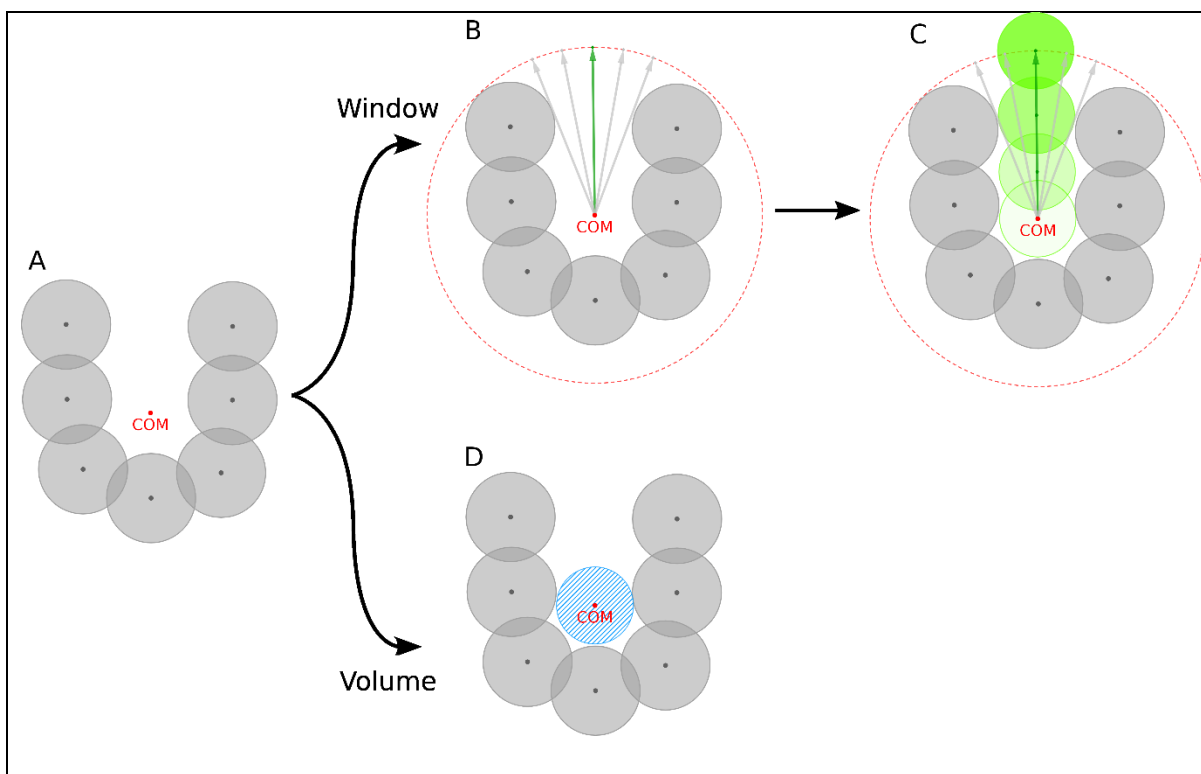
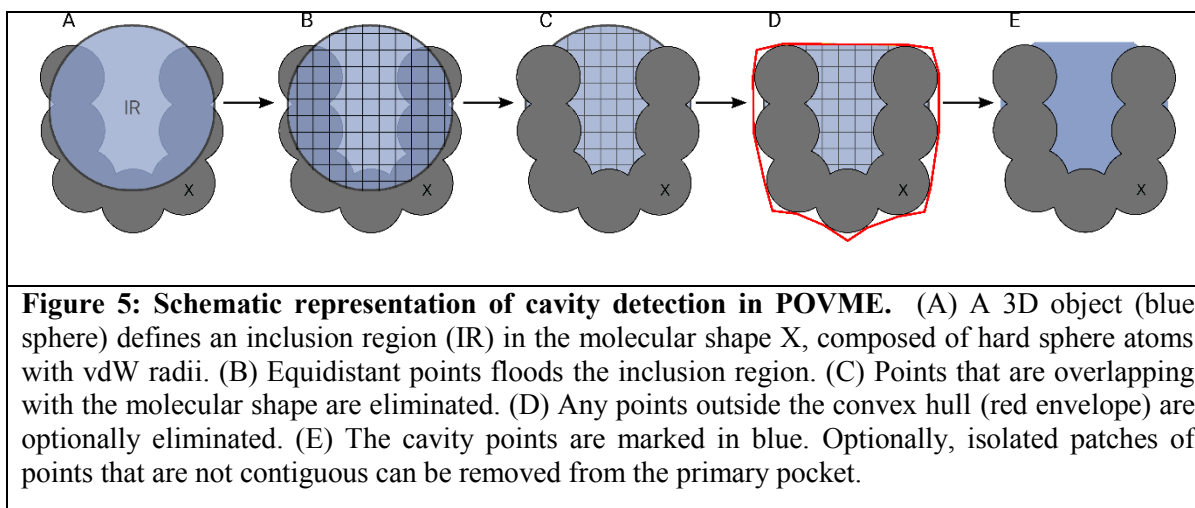


Figure 4: Schematic representation of cavity and window detection algorithms in pywindow. (A) A molecular shape is represented by hard sphere atoms, and the center of mass (COM) is calculated (red point). (B) Window detection starts by evenly distributing sampling points (red dots) in a sphere around the structure. Then, vectors (gray and green arrows) connecting the COM and the sampling points are inspected. Vectors that overlap with any of the structure atoms are removed. (C) For each window, the vector (green arrow) with the largest included sphere (green spheres) along its path is chosen, and the window's circular diameter is calculated. (D) From step A, the void volume of the structure is estimated by fitting the largest possible sphere (blue circle) with the COM as its center.

2.3.4 POVME 3.0

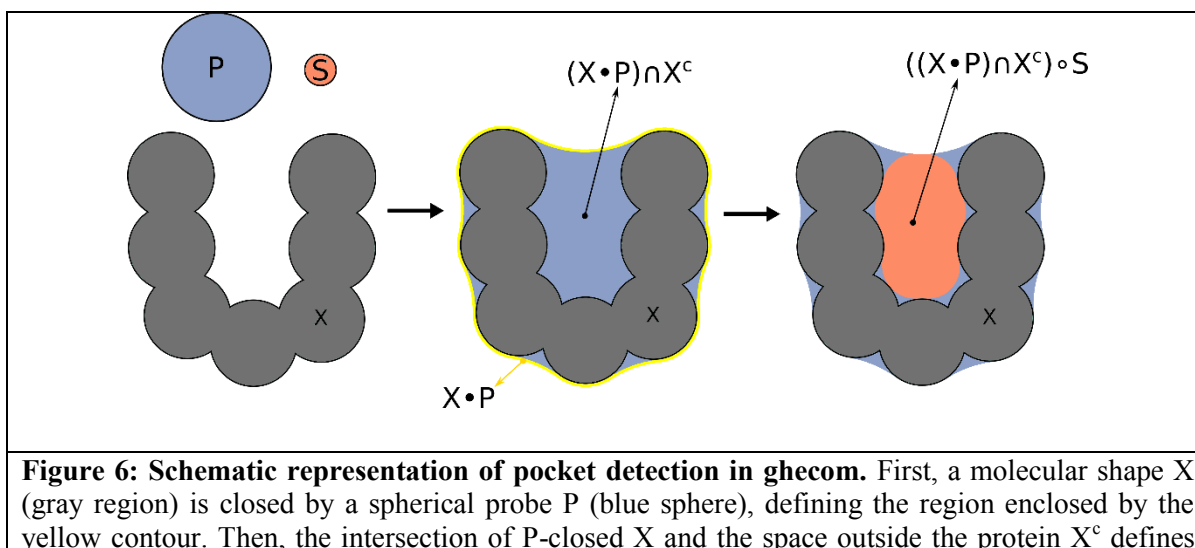
POcket Volume MEasurer 3.0 (POVME 3.0)²⁶ is a Python package that uses a grid-based method for determining pocket shape and volume (Figure 5). Cavity detection defines inclusion and exclusion regions using 3D objects (spheres, cylinders and cuboids). Briefly, POVME floods the inclusion regions with equidistant points, that do not overlap with any atom, and eliminates any points encompassed by the optional exclusion regions. Then, a convex hull defined around the target structure removes points that are outside its envelope. For each pocket, POVME performs a volume estimation and applies a BINANA coloring algorithm²⁷, that depicts hydrogen bonding donors, hydrogen bonding acceptors, aromatic stacking, hydrophobicity and

hydrophilicity.



2.3.5 ghecom

Grid-based HECOMi finder (ghecom)²⁸ is a grid-and-sphere-based method that detects deep and shallow pockets (Figure 6). ghecom combines mathematical morphology operations²⁹ with different spherical probes to report openness-closedness of a target molecular shape, which when compared reveal deep and shallow (multiscale) pockets. Thus, a single-linkage clustering method groups pockets and subsequently estimates their volumes. Additionally, ghecom calculates a metric called pocketness, which relates the volume and depth of points per residue or atom, which indicates how much it contributes to ligand binding.



the region not accessible to probe P (blue region), which is opened by a spherical probe S (orange sphere), where P is larger than S. Finally, the pocket (orange region) is defined by the space outside the molecular shape not accessible to P, but to S. For multiscale detection, different sizes of the spherical probe P are used.

2.3.6 CAVER

CAVER³⁰ is a method for computing tunnels and channels. The original version used a grid-and-surface-based approach, but in CAVER 3.0,³¹ a Voronoi diagram approach was adopted (Figure 7). The graphical user interface, CAVER Analyst 2.0,³² incorporates CAVER 3.0 and visually helps users in tunnel and cavity computations. CAVER 3.0 constructs a pseudo-Voronoi diagram of a target protein to identify pathways that resemble tunnels that connect cavities with the surrounding bulk solvent, which are characterized by length, average radius, and gorge radius. In addition to tunnel detection, CAVER Analyst 2.0 can also identify regions of empty space, where a small probe can enter from the outside, but a large probe cannot, like KVFinder project (Figure 2) and ghecom (Figure 6).

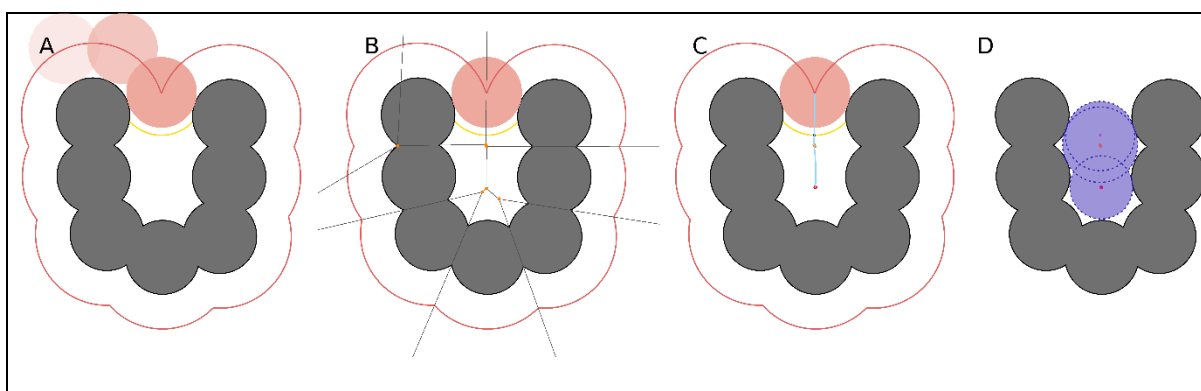


Figure 7: Schematic representation of tunnel and channel detection in CAVER 3.0 and CAVER Analyst 2.0. (A) A molecular shape is inspected by a spherical probe, called shell probe (in sphere), with a radius, specified by shell radius parameter, to define an outer SAS-surface (red line). From it, a distance, specified by shell depth parameter, is removed to define an inner surface (yellow line). (B) A pseudo-Voronoi diagram is constructed based on the molecular shape. Voronoi vertices (orange points) are used to create the tunnel/channel centerlines. (C) A start point (red point) is a user-defined parameter, defined as the molecular shape's center of mass, and an endpoint (blue point) is defined at the center of the inner surface. From the start point, the centerline passes through Voronoi edges and vertices to form a tunnel/channel until the outer surface and passing through the end point. (D) Spheres are fitted into all points of the centerline, from the start point to the endpoint, which defines the bottleneck radius (opening) along the tunnel/channel.

2.3.7 MoloVol

MoloVol³³ is a grid-and-sphere-based method that analyzes geometrical features in chemical structures. MoloVol detects cavities using one or two spherical probes that roll around a target molecular surface, which can be represented as vdW, SES, and SAS surfaces.. For a single probe, cavities are defined by isolated regions inside the target structure, while for two probes, a large probe defines the outer space, and a small probe defines the space within cavities, like KVFinder project (Figure 2) and ghecom (Figure 6). For each cavity, MoloVol estimates the volume and surface area, and classifies them based on the number of openings. The method is available with an easy-to-use graphical user interface and a command-line interface for high-throughput calculations.

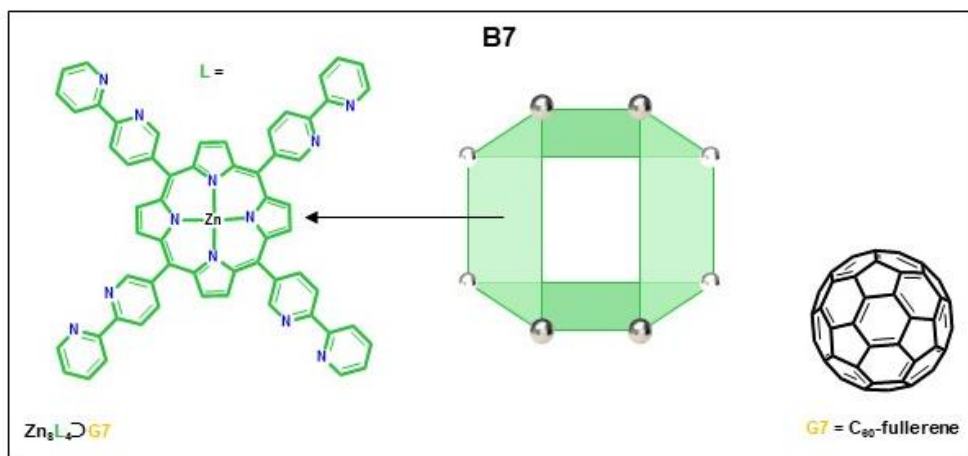
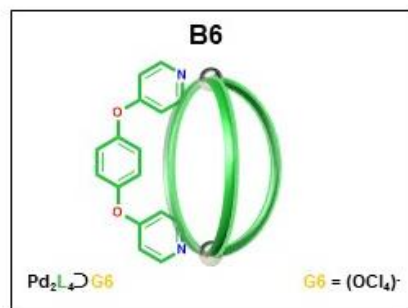
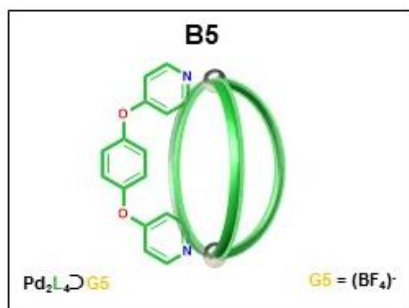
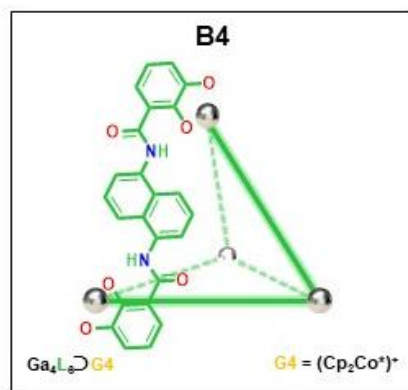
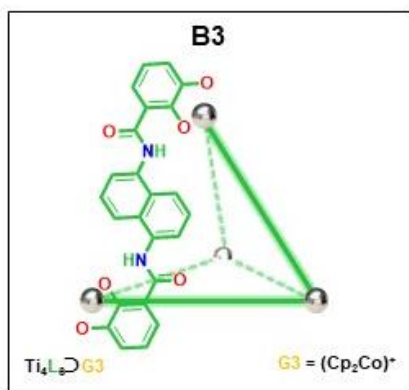
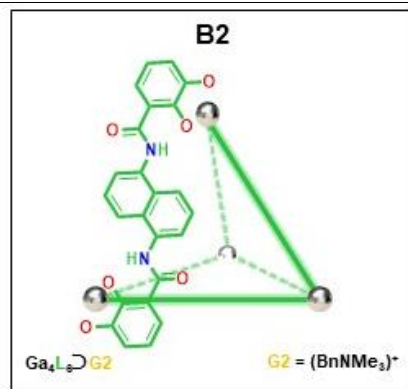
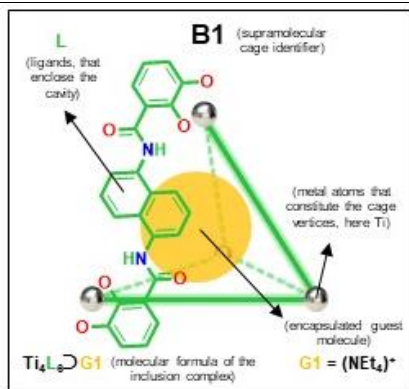
RESULTS AND DISCUSSION

First, a quantitative comparison between software is provided, that is based on two benchmark datasets of well-known supramolecular cages. Afterward, a qualitative comparison of the available software is presented, highlighting their strengths and weaknesses, with special attention to usability and applicability. With this quantitative/qualitative comparison in hand, we intend to guide users toward the software best suited to their needs.

3.1 Benchmark datasets

Two benchmark datasets, comprising 22 well-known supramolecular cages (Figure 8 and 9), have been selected from the supramolecular chemistry literature to evaluate the previously described software. The X-ray diffraction (XRD) structure of these cages are readily accessible from the Cambridge Structural Database (CSD),³⁴ which makes them an ideal starting point for our work. However, it should be noted that the prediction of guest binding based solely on cavity detection in XRD structures has its limitations. Firstly, XRD structures might not be available for the cage that needs to be studied. Secondly, volumetric data might be less meaningful for flexible cages. In such cases, cavity detection could be combined with other *in silico* methods (e.g., Grand Canonical Monte Carlo or molecular dynamics simulations) to

generate these input structures (trade-off between accuracy and simplicity). For more information on the selected structures, refer to the Supporting Information (SI, section 2, Table S2 and S3).



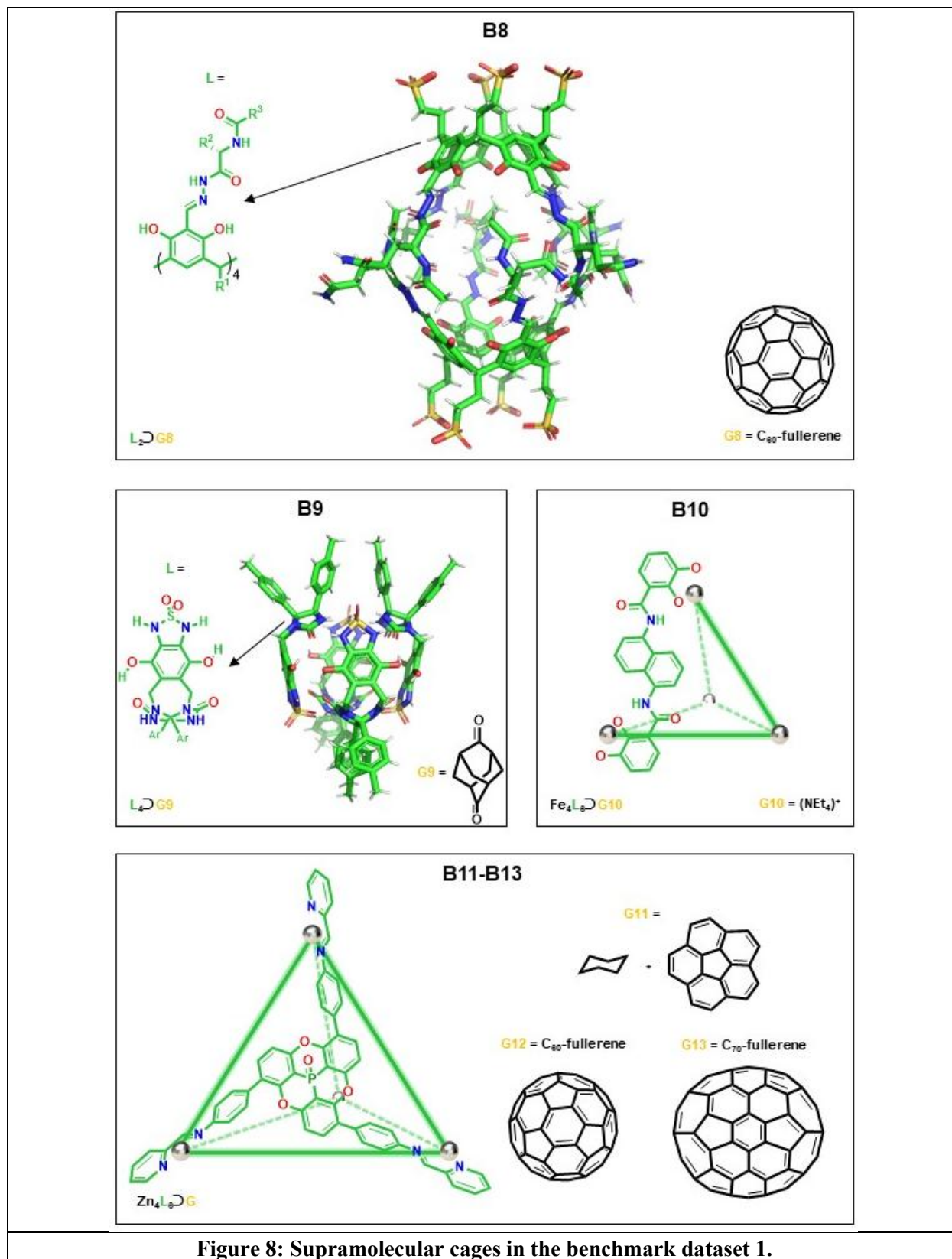
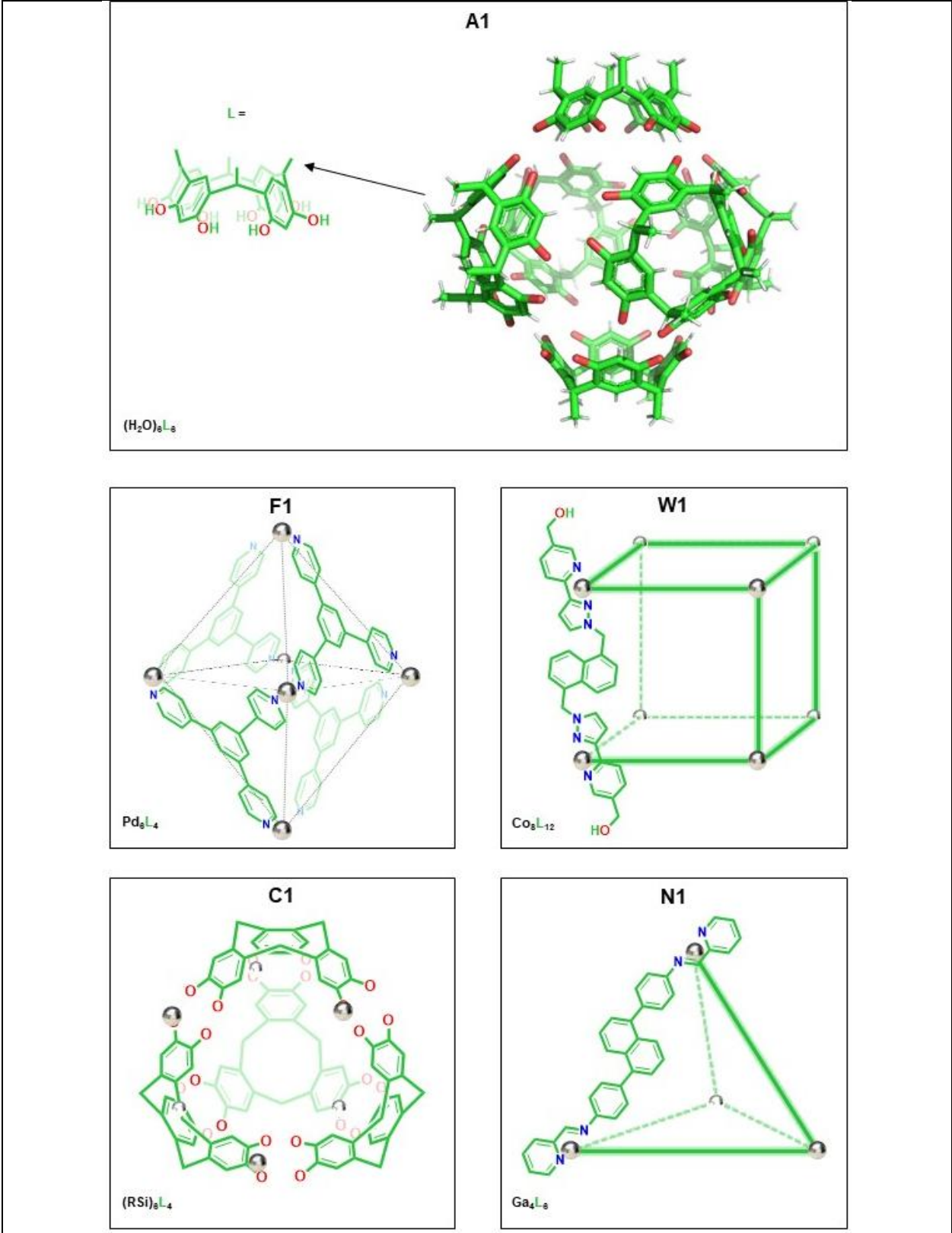
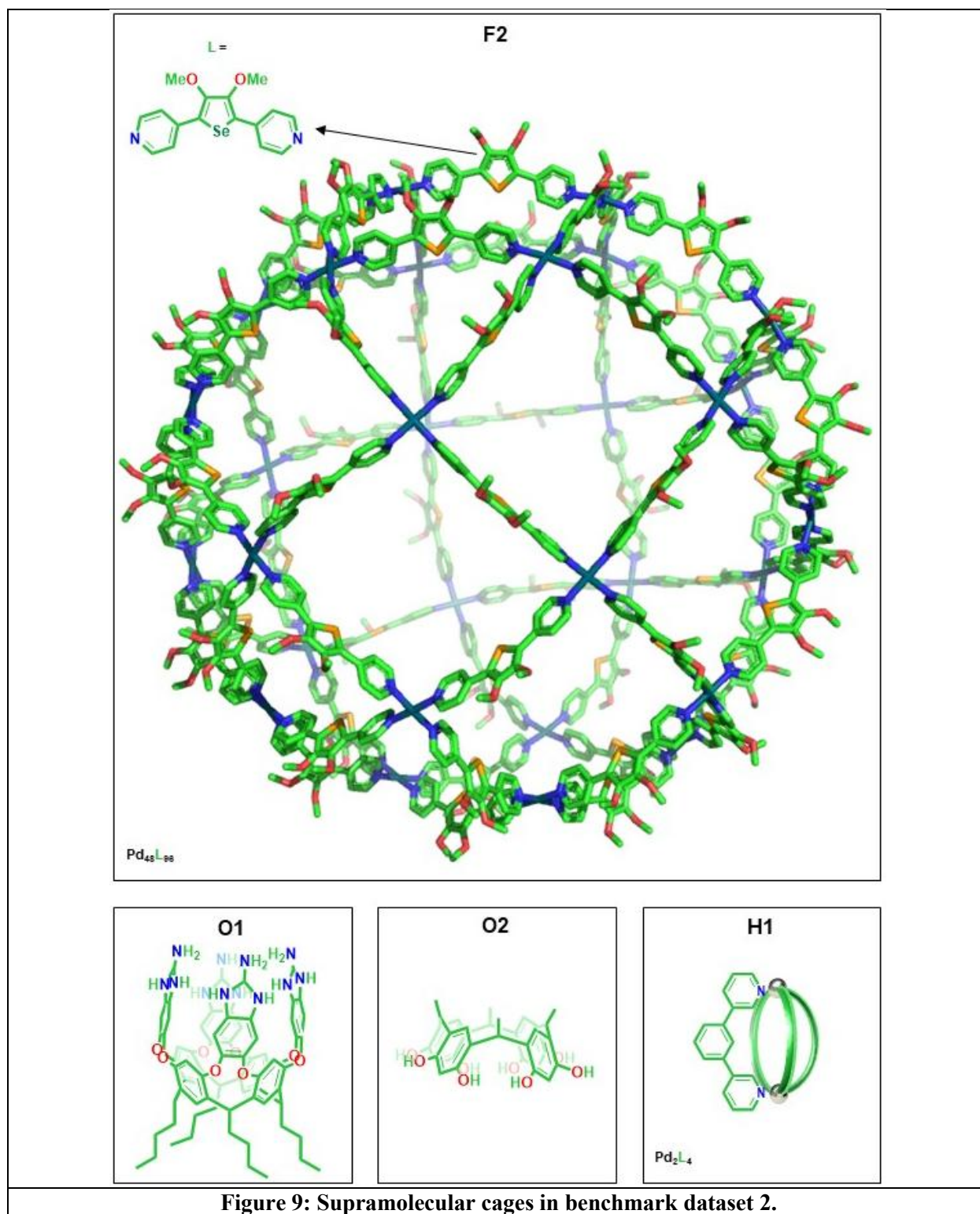


Figure 8: Supramolecular cages in the benchmark dataset 1.





The selection of supramolecular cages has been guided by two main criteria. First, to **test the capabilities and limitations of the previously discussed software**. Thus, data selection covers

a wide variety of supramolecular cages with different topological and morphological features (i.e., cavity/opening sizes and shapes). Cage diversity followed the categorization used to classify protein cavities (Figure 10).^{11b} It should be noted that even though covalent cages have not been included in the benchmarking datasets, their topological and morphological features are similar to the classes of supramolecular cages included in the datasets. Secondly, with this selection we aim to **provide reliable and comparable cavity volume data for supramolecular cages with well-documented and rich host-guest chemistry**. Indeed, benchmark dataset 2 features some of the most relevant examples of supramolecular cages from the literature. The majority of these cages lack data about their cavities and when available, cavity volumes were calculated with different software using different parameters. As a consequence, it is not possible to compare the cavity volumes of these cages and draw conclusions for a rational design. In this work, 7 different software have been used for each supramolecular cage (see next section), therefore the cavity volumes of these cages, calculated with the same software, are now comparable.


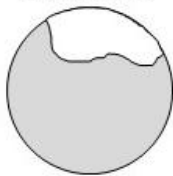
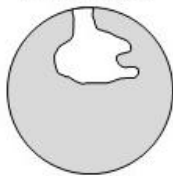

Type of Cavity	Void	Cleft/Groove	Invagination	Tunnel
				
Supramolecular cage	B1-B13, A1, R1, N1	O2	C1, F1, F2, H1, W1	O1

Figure 10: Categorization of supramolecular cage cavities based on the classification used for protein cavities.^{11b}

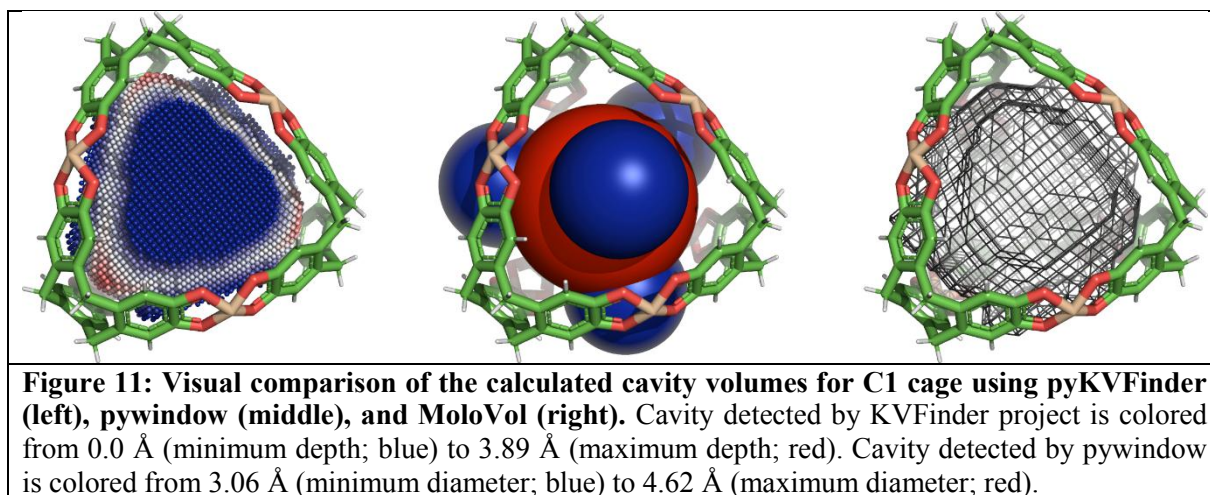
3.2 Parameter optimization

Cavity detection parameters of every software (except for pywindow) have been optimized for biomolecules. As the topology and morphology of supramolecular cages differ considerably from biomolecules, we have adjusted these parameters whenever necessary, in order to get the best performance and the most accurate visual representation of the cavity for each member of the benchmark datasets. We propose a new set of starting detection parameters for KVFinder project (Table S10) and MoloVol (Table S11), which users are advised to use on early runs to

have reasonable results. For a detailed description of the parameter optimization, refer to the SI (SI, section 3).

3 Software performance

The software evaluation comprises three distinct benchmarking analyses: visual modeling assessment, quantitative analysis, and qualitative analysis. First, we relied on visual modeling analysis of the identified cavities. In Figure 11, the cavity size of the supramolecular cage C1 was estimated by KVFinder, pywindow, and MoloVol as 558 \AA^3 , 412 \AA^3 , and 648 \AA^3 , respectively.



This visual assessment provides a good starting point for identifying outlying results; however, a finer quantitative comparison would be more appropriate for assessing software performance. Unfortunately, moving from qualitative to quantitative descriptors remains a challenge in the current field of cavity detection methods.^{11b} On the one hand, verification of calculated volumes is not possible, as the “real” volume of cavities in biological systems is unknown. It has been suggested that performing a quantitative assessment on a dataset of well-defined artificial cavities might offer a solution to this problem. Therefore, in this work, we address this problem by using a dataset of artificial cavities (supramolecular cages). In addition, we take this problem one step further, in order to have an even more realistic evaluation of software performance.

To assess the performance of any cavity detection software, the calculated volumes must be compared to the “real” cavity volume, a value that cannot be determined by experimental methods. To tackle this problem, we are taking advantage of Rebek’s rule of thumb: in any (biological and artificial) host-guest system, the guest occupies 55% of the available space within the cavity, when guest encapsulation is only driven by weak interactions (e.g., dipole-dipole, dipole-induced dipole, London dispersion forces).³⁵ It is important to note that the packing coefficient (PC) is defined as the $V_{\text{guest}}/V_{\text{cavity}}$ ratio (e.g., for 55%, PC = 0.55). Therefore, we have selected 13 XRD structures (benchmark dataset 1) from the CSD,³⁰ each of these inclusion complexes features a supramolecular cage with guest molecule(s) within its cavity (Figure 8, and Table S2). Guest vdW volumes were calculated (SI section 4, Figure S1), and the cavity sizes were estimated using Rebek’s rule. Then, the estimated volumes were compared to the volumes calculated with different software. Since the relative error (RE) and mean relative absolute error (MRAE) are good performance measures for prediction, we applied them to assess performance of the different software (Table 1).

Table 1: Software performance for benchmark dataset 1. Relative error (RE) with the Rebek's cavity volume as the reference can be found in parentheses. Estimated cavity volume = Guest vdW volume ÷ 0.55. Whenever available, the estimated cavity volumes from the original publications are depicted in the 'Vref' column. The detailed description of benchmark dataset 1 can be found in the SI (Section 2).

Inclusion complex	Guest vdW volume (Å ³)	Estimated cavity volume (Å ³)	Calculated Cavity Volumes							Vref
			(Å ³ (RE %))							
			KVFinder project	Fpocket	MoloVol	CAVER	ghecom	pywindow	POVME	
B1	150	273	283 (3.7)	247 (-9.5)	261 (-4.5)	396 (44.8)	175 (-35.9)	133 (-51.4)	109 (-60.1)	293
B2	155	281	269 (0.8)	279 (-0.6)	289 (2.7)	339 (20.8)	192 (-31.7)	90 (-68.1)	108 (-61.6)	285
B3	137	248	269 (8.1)	277 (11.6)	253 (1.6)	335 (34.9)	191 (-23.3)	113 (-54.6)	99 (-60.1)	270

B4	309	562	438 (-22.1)	434 (-22.7)	410 (-27.0)	474 (-15.7)	343 (-39.0)	251 (-55.4)	222 (-60.5)	434
B5	50	90	78 (-13.6)	84 (-6.5)	44 (-51.4)	65 (-27.5)	29 (-67.6)	37 (-59.2)	11 (-88.4)	52
B6	53	96	81 (-15.6)	82 (-14.3)	55 (-42.5)	79 (-17.4)	64 (-33.7)	40 (-58.8)	11 (-88.8)	55
B7	519	944	757 (-19.8)	771 (-18.3)	708 (-25.0)	778 (-17.6)	734 (-22.3)	492 (-47.9)	385 (-59.2)	810
B8	512	930	731 (-21.4)	805 (-13.5)	747 (-19.7)	682 (-26.7)	704 (-24.3)	517 (-44.4)	409 (-56.0)	-
B9	141	257	155 (-39.5)	181 (-29.5)	177 (-31.1)	173 (-32.5)	159 (-38.0)	122 (-52.4)	62 (-75.8)	184
B10	151	274	251 (-8.4)	225 (-17.7)	225 (-17.7)	351 (28.1)	161 (-41.1)	125 (-54.5)	94 (-65.6)	261
B11	307	558	496 (-11.0)	482 (-13.6)	435 (-22.0)	457 (-18.1)	401 (-28.1)	265 (-52.4)	208 (-62.7)	-
B12	524	954	737 (-22.7)	627 (-34.2)	650 (-31.8)	742 (-22.2)	606 (-36.5)	532 (-44.2)	319 (-66.5)	718
B13	618	1 123	872 (-22.3)	811 (-27.7)	821 (-26.9)	1031 (-8.1)	752 (-33.0)	555 (-50.5)	450 (-59.9)	925
MRAE (%)			16.1	16.9	23.4	24.2	35.0	53.4	66.6	-

First, to assess this quantitative analysis of the benchmark dataset 1, we must analyze the RE values, which consist of: (1) error related to the definitions of the cavity boundaries; and (2) error related to deviations in the PC from Rebek's rule. To minimize the first error, benchmark dataset 1 only contains supramolecular cages with well-defined, closed cavities (void, Figure 10), where the characterization of cavity boundaries is simple and straightforward. On the other hand,

deviations from Rebek's rule are more elusive. It has been shown that higher packing coefficients can be reached in inclusion complexes, where in addition to weak dipole-dipole interactions, other forces such as H-bonding, π - π and CH- π are at work.³⁶ This results in tighter packing (i.e., $PC > 0.55$), where the extra stabilization enthalpy counterbalances the loss of entropy (restricted movement of the guest within the cage). Therefore, negative RE is expected in these cases, due to the overestimation of cavity volume. For B4-B9, B12, and B13 cages, this trend can be clearly observed in the results of KVFinder project, Fpocket, MoloVol, and CAVER. For B5, B6, and B9, there is experimental evidence for H-bonding interactions (SI, Figure S2, and S3).³⁷ Furthermore, when considering the examples with fullerenes (B7, B8, B12 and B13), and $(Cp^*)_2Co^+$ (B4) as guests, in addition to weak dipole-dipole interactions, additional π - π and CH- π interactions favor strong host-guest association. For ghecom, pywindow, and POVME, a large negative RE is observed for each supramolecular cage, which suggests that the calculation error of these software becomes more significant. On the other hand, MRAE values clearly show that **KVFinder project and Fpocket, provide the most reliable cavity volumes**. A visual representation of the software performance (Figure 12A) shows that volumes calculated with KVFinder project and Fpocket are close to the estimated volume (i.e., zero RE) with a small spread compared to other software. Despite a few outlying results, the overall trend (*vide supra*) supports our approach of using Rebek's rule as the best estimate for the “real” cavity volume within these supramolecular cages.

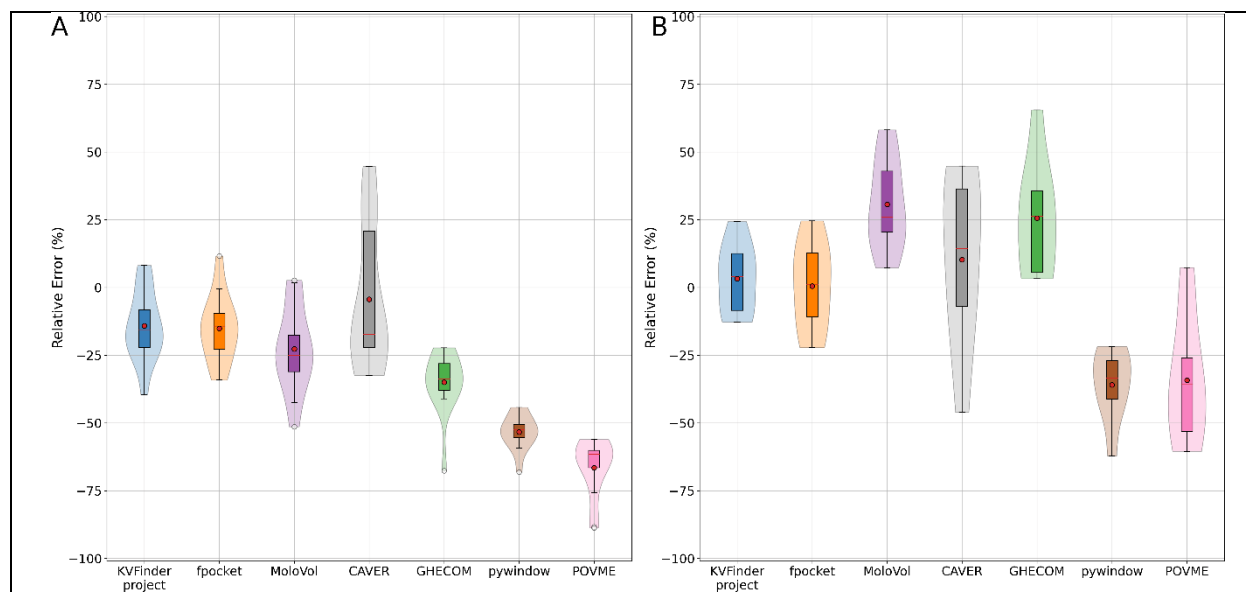


Figure 12: Boxplot of the relative error of the cavity volume in the benchmark datasets. (A) Benchmark dataset 1 with the Rebek's cavity volume as reference. (B) Benchmark dataset 2 with the

average volume of detected cavities as reference. In benchmark dataset 2, cavity volumes of O2 are excluded due to the high uncertainty related to some methods being unable to detect its cavity. The probability density function of the relative error for each software is estimated using the Kernel Density Estimate with Gaussian kernels and displayed as violin plots. The red line and point indicate the median and mean relative errors, respectively.

We then moved on to benchmark dataset 2 (Figure 9, SI section 2, Table S3) to test the capabilities and limitations of the software collection. This selection presents more challenging examples with different topological and morphological features (tunnels, clefts/grooves, large voids, large openings). However, these examples do not contain guests, so cavity volume estimation based on Rebek's rule is not possible. Consequently, the average volume of detected cavities was chosen as the point of comparison to calculated volumes (Table 2), which establishes an agreement between methods by mitigating deviations in volume data.

Table 2: Software performance for benchmark dataset 2. Average volumes have been calculated by averaging calculated volumes of pyKVFinder, parKVFinder, Fpocket, MoloVol and CAVER. Relative error (RE) with the average volumes can be found in parentheses. Whenever available, the estimated cavity volumes from the original publications are depicted in the 'Vref' column. The detailed description benchmark dataset 2 can be found in the SI (Section 2).

			Calculated Cavity Volumes (\AA^3 (RE %))							
Type of cavity	Cage	Average Volume	KVFinder project	Fpocket	MoloVol	CAVER	ghecom	pywindow	POVME	V _{ref}
Void	A1	1315	1399 (6.4)	1387 (5.5)	1411 (7.3)	1778 (35.2)	1362 (3.6)	1029 (-21.8)	840 (-36.1)	1375
Invagination	C1	549	558 (1.6)	617 (12.3)	648 (17.9)	569 (3.6)	669 (21.7)	412 (-25.0)	373 (-32.1)	-
	F1	500	463 (-7.5)	483 (-3.5)	653 (30.6)	481 (-3.9)	655 (30.9)	306 (-38.8)	461 (-7.8)	-
	F2	42572	37728 (-11.3)	39077 (-8.2)	62843 (47.6)	22941 (-46.1)	59147 (38.9)	30568 (-28.2)	45704 (7.4)	-
	H1	259	226 (-12.8)	211 (-18.6)	409 (58.1)	215 (-16.8)	429 (65.5)	154 (-40.4)	168 (-35.1)	-
	N1	434	488 (12.4)	495 (14.0)	526 (21.2)	543 (25.1)	461 (6.2)	314 (-27.6)	211 (-51.4)	-
	O1	142	160 (12.9)	177 (24.5)	172 (21.2)	199 (40.0)	146 (3.2)	80 (-43.6)	59 (-58.4)	-
Tunnel	W1	696	866 (24.4)	542 (-22.2)	984 (41.3)	1008 (44.8)	936 (34.4)	263 (-62.2)	275 (-60.5)	400
Cleft/ Groove	O2	20	36 (78.8)	0* (-100.0)	37 (41.3)	0* (-100.0)	0* (-100.0)	6 (-62.2)	2 (-60.5)	-
MRAE (%)			18.7	23.2	36.4	35.1	33.8	39.9	42.0	-

(*) Cavity not detected.

In summary, each software has its own capabilities and shortcomings that derive from the type of cavity detection method applied, its implementation, and its software interfaces (Table 3). In general, grid-and-sphere-based and tessellation-based methods give the most precise cavity

volumes, while grid-based and sphere-based methods performed worse. As previously described, pywindow uses an algorithm that defines the cavity volume as the volume of the largest sphere that can be fitted within the cavity. Therefore, pywindow consistently underestimates the cavity size, especially clear for non-spherical cavities (F1, H1, N1, W1 and O1) and shallow grooves (O2). Grid-, sphere-, and grid-and-sphere-based methods can detect any type of cavities in supramolecular cages, except for ghecom and CAVER Analyst 2.0, which, together with tessellation-based methods, fail to detect clefts and grooves. Taking into account our assessment, KVFinder project and Fpocket overperformed the other software in benchmarking, as they provide volumes closest to the estimated and average volumes, with a small spread compared to other software (Figure 12B).

All these methods are accessible to users with a simple and well-documented installation, configuration, and execution. From this perspective, the software interfaces available to users, i.e., Graphical User Interface (GUI), Command Line Interface (CLI), Application Programming Interface (API), and Web Application, dictate the pool of viable applications for each software. Commonly, GUIs and Web Applications are simpler and easier for less experienced users to execute due to visual aids that intuitively guide the user through the analysis pipeline. However, these interfaces lack efficiency when performing analysis with large datasets, e.g., high-throughput analysis (HTA), molecular dynamics (MD) simulations, machine learning (ML), deep learning (DL), virtual screening (VS) applications, and automating pipelines. In this sense, CLIs and APIs are efficient and integrable software interfaces, which allow the development of applications with large datasets and pipeline automation; however, they are highly dependent on their documentation to the user interact with the software efficiently. Also, the main drawback of CLI is that they are black-box applications, which do not allow users to fully customize pipeline automation since some variables are not definable by the user. On the other hand, APIs (e.g., pyKVFinder, pywindow, and fpocket) are the most versatile interfaces, which users can use them as building blocks for more complex applications and/or integrate them with third-party scientific packages, e.g., numpy, scipy, scikit-learn and matplotlib. Additionally, pyKVFinder has its core data structures accessible and easy-to-handle, allowing the development of new characterizations and applications built around them.

Table 3: Assessment of different cavity detection methods.

Cavity detection software	Category	Interface				Cavity-type				Ref.
		GUI	CLI	API	Web	Void	Invagination	Tunnel	Cleft/Groove	
KVFinder project	grid-and-sphere-based	x	x	x	x	+++	+++	+++	+++	14,15
Fpocket	tessellation-based	x	x	x	x	+++	+++	+++	NA	17
MoloVol	grid-and-sphere-based	x	x		x	+++	++	+++	+++	28
CAVER 3.0	tessellation - based	x	x		x	++	++	++	NA	26
CAVER Analyst 2.0	grid-and-sphere-based	x	x							25
ghecom	grid-and-sphere-based	x	x		x	+++	++	+++	NA	23
pywindow	sphere-based			x		++	+	+	+	20
POVME	grid-based	x	x			+	+	+	+	21

CONCLUSIONS

In this work, we present the first evaluation of cavity characterization methods in the context of supramolecular cages. Our thorough literature review led us to identify 7 cavity detection software, which were evaluated on two benchmark datasets of well-known supramolecular cages with diverse cavity shapes and sizes. All of this software is available and accessible even for less experienced users, with user-friendly interfaces, simple installation, and well-documented configuration, and execution. Our results show that KVFinder project and Fpocket are the most accurate software for characterizing supramolecular cavities. In order to obtain the most detailed picture of the software performance, we took advantage of Rebek's rule. This new strategy (and the benchmark dataset 1) also offers a solution to the current problem of missing gold standard reference data in the field of cavity detection. In summary, this work aims to provide an entry

point in the field of cavity characterization for potential users in the supramolecular community. This invaluable technique provides an added value to the toolbox of rational design of supramolecular cages with improved physicochemical properties.

ASSOCIATED CONTENT

Supporting Information. Detailed description of software and benchmark dataset selections, guest volume calculations and optimized detection parameters for each software can be found in the SI (PDF).

Data and Software Availability

The benchmark datasets used in this study, which include the structural data files of each supramolecular cage and guest, are available at Zenodo: <https://doi.org/10.5281/zenodo.7702311> (Zenodo, 07-04-2023). Additionally, the source code (v1.0.0) for the benchmarking procedure can be found at <https://github.com/LBC-LNBio/SMC-Benchmarking> (GitHub, 07-04-2023). All software used in the benchmarking, including pyKVFinder (v0.5.1; <https://github.com/LBC-LNBio/pyKVFinder>, GitHub, 07-04-2023), parKVFinder (v1.2.0; <https://github.com/LBC-LNBio/parKVFinder>, GitHub, 07-04-2023), Fpocket (v3.1.4.2; <https://github.com/Discngine/fpocket>, GitHub, 07-04-2023), pywindow (v0.0.4; <https://github.com/marcinmiklitz/pywindow>, GitHub, 07-04-2023), POVME 3.0 (v3.0.35; <https://github.com/POVME/POVME3>, GitHub, 07-04-2023), ghecom (21/07/2020; <https://pdj.org/ghecom/>, GHECOM, 07-04-2023), CAVER 3.0 (v3.0.2; <https://caver.cz>, CAVER 07-04-2023), CAVER Analyst 2 (v2.0b2; <https://caver.cz>, CAVER 07-04-2023), and MoloVol (v1.0.0; <https://molovol.com/>, MoloVol, 07-04-2023), are freely available on the internet.

AUTHOR INFORMATION

Corresponding Authors

Paulo S. Lopes-de-Oliveira - Brazilian Center for Research in Energy and Materials (CNPEM), Brazilian Biosciences National Laboratory (LNBio), Rua Giuseppe Máximo Solfaro, 10000 Bosque das Palmeiras, Campinas, SP, 13083-100, Brazil. E-mail: paulo.oliveira@lnbio.cnpem.br, <https://orcid.org/0000-0002-1287-8019>

György Szalóki - Laboratoire Hétérochimie Fondamentale et Appliquée (LHFA, UMR 5069), CNRS, Université Toulouse III – Paul Sabatier, 118 Route de Narbonne, Toulouse 31062, Cedex 09, France. E-mail: gyorgy.szaloki@univ-tlse3.fr, <https://orcid.org/0000-0002-6950-9919>

Authors

Joao V. S. Guerra - Brazilian Center for Research in Energy and Materials (CNPEM), Brazilian Biosciences National Laboratory (LNBio), Rua Giuseppe Máximo Solfaro, 10000, Bosque das Palmeiras, Campinas, SP, 13083-100, Brazil. E-mail: joao.guerra@lnbio.cnpem.br, <https://orcid.org/0000-0002-6800-4425>

Luiz F. G. Alves - Brazilian Center for Research in Energy and Materials (CNPEM), Brazilian Biosciences National Laboratory (LNBio), Rua Giuseppe Máximo Solfaro, 10000, Bosque das Palmeiras, Campinas, SP, 13083-100, Brazil. E-mail: luiz.alves@lnbio.cnpem.br, <https://orcid.org/0000-0002-6596-405X>.

Didier Bourissou - Laboratoire Hétérochimie Fondamentale et Appliquée (LHFA, UMR 5069), CNRS, Université Toulouse III – Paul Sabatier, 118 Route de Narbonne, Toulouse 31062, Cedex 09, France. E-mail: didier.bourissou@univ-tlse3.fr, <https://orcid.org/0000-0002-0249-1769>

Author Contributions

J.V.S.G and L.F.G.A. performed the calculations. J.V.S.G, L.F.G.A. and G.S. contributed to the conceptualization, data interpretation and manuscript editing. G.S. designed and directed the project. The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

Funding Sources

The Centre National de la Recherche Scientifique (CNRS), Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) [grant number 2018/00629-0] and Brazilian Center for Research in Energy and Materials (CNPEM) are kindly acknowledged for the funding.

Notes

The authors declare no competing financial interest.

ABBREVIATIONS

MOC – metal organic cage, MOF – metal organic framework, COF – covalent organic framework, DFT – density functional theory, vdW – van der Waals, SAS – solvent accessible surface, SES – solvent excluded surface, RE – relative error, MRAE – mean relative absolute error, PC - packing coefficient, GUI – graphical user interface, CLI – command line interface, API – application programming interface, HTA – high throughput analysis, MD – molecular dynamics, ML – machine learning, DL – deep learning, VS – virtual screening, CSD – Cambridge Structural Database, XRD – X-ray diffraction

REFERENCES

¹ Silverman, R. B. *The Organic Chemistry of Enzyme-Catalyzed Reactions* **2002** (Elsevier Science). Punekar, N. S. *ENZYMES: Catalysis, Kinetics and Mechanisms* **2018** (Springer).

² Active sites in enzymes are usually located within the largest and deepest cavity in the structure. See: Laskowski, R. A.; Luscombe, N. M.; Swindells, M. B.; Thornton, J. M. Protein clefts in molecular recognition and function. *Protein Sci.* **1996**, *5*, 2438-2452.

³ Grommet, A. B.; Feller, M.; Klajn, R. Chemical reactivity under nanoconfinement. *Nat. Nanotechnol.* **2020**, *15*, 256-271.

⁴ (a) Xu, H.; Wu, P. New progress in zeolite synthesis and catalysis. *Natl. Sci. Rev.* **2022**, *9*. (b) Gong, X.; Çağlayan, M.; Ye, Y.; Liu, K.; Gascon, J.; Chowdhury, A. D. First-Generation Organic Reaction Intermediates in Zeolite Chemistry and Catalysis. *Chem. Rev.* **2022**, *122*, 14275-14345.

⁵ (a) Pascanu, V.; Miera, G. G.; Inge, A. K.; Martín-Matute, B. Metal–Organic Frameworks as Catalysts for Organic Synthesis: A Critical Perspective. *J. Am. Chem. Soc.* **2019**, *141*, 7223-7234. (b) Freund, R.; Zaremba, O.; Arnauts G.; Ameloot, R.; Skorupskii, G.; Dincă, M.; Bavykina, A.; Gascon, J.; Ejsmont, A.; Goscianska, J.; Kalmutzki, M.; Lächelt, U.; Ploetz, E.; Diercks, C. S.; Wuttke, S. The Current Status of MOF and COF Applications. *Angew. Chem. Int. Ed.* **2021**, *60*, 23975-24001.

⁶ For covalent and H-bonded cages see: (a) Montà-González, G.; Sancenón, F.; Martínez-Mañez, R.; Martí-Centelles, V. Purely Covalent Molecular Cages and Containers for Guest

Encapsulation. *Chem. Rev.* **2022**, *122*, 13636-13708. For metal-organic cages see: (b) Cook, T. R.; Stang, P. J. Recent Developments in the Preparation and Chemistry of Metallacycles and Metallacages *via* Coordination. *Chem. Rev.* **2015**, *115*, 7001-7045. (c) McConnell, A; J. Metallosupramolecular cages: from design principles and characterisation techniques to applications. *Chem. Soc. Rev.* **2022**, *51*, 2957-2971.

⁷ (a) Morimoto, M.; Bierschenk, S. M.; Xia, K. T.; Bergman, R. G.; Raymond, K. N.; Toste, F. D. Advances in supramolecular host-mediated reactivity. *Nat. Catal.* **2020**, *3*, 969-984. (b) Forgan, R. S.; Lloyd G. O. Monographs in Supramolecular Chemistry: 31. Reactivity in Confined Spaces. **2021** (The Royal Society of Chemistry).

⁸ Piskorz, T. K.; Martí-Centelles, V.; Young, T. A.; Lusby, P. J.; Duarte, F. Computational Modeling of Supramolecular Metallo-organic Cages—Challenges and Opportunities. *ACS Catal.* **2022**, *12*, 5806-5826.

⁹ Young, T. A.; Gheorghe, R.; Duarte, F. cgbind: A Python Module and Web App for Automated Metallogage Construction and Host–Guest Characterization. *J. Chem. Inf. Model.* **2020**, *60*, 3546-3557.

¹⁰ (a) Norjmaa, G.; Vidossich, P.; Maréchal, J.-D.; Ujaque, G. Modeling Kinetics and Thermodynamics of Guest Encapsulation into the $[M_4L_6]^{12-}$ Supramolecular Organometallic Cage. *J. Chem. Inf. Model.* **2021**, *61*, 4370-4381. (b) Pahima, E.; Zhang, Q.; Tiefenbacher, K.; Major, D. T. Discovering Monoterpene Catalysis Inside Nanocapsules with Multiscale Modeling and Experiments. *J. Am. Chem. Soc.* **2019**, *141*, 6234-6246.

¹¹ For an exhaustive list of publications that have used different software for cavity detection in supramolecular cages, see SI S2.

¹² For comprehensive reviews on cavity detection in enzymes please see: (a) Krone, M.; Kozlíková, B.; Lindow, N.; Baaden, M.; Baum, D.; Parulek, J.; Hege, H.-C.; Viola, I. Visual Analysis of Biomolecular Cavities: State of the Art. *Comput. Graph Forum* **2016**, *35*, 527-551. (b) Tiago, S.; Daniel, L.; Sérgio, D.; Francisco, F.; João, P.; Joaquim, J.; Chandrajit, B.; Abel, G. Geometric Detection Algorithms for Cavities on Protein Surfaces in Molecular Graphics: A Survey. *Comput. Graph Forum* **2017**, *36*, 643-683. (c) Brezovsky, J.; Chovancova, E.; Gora, A.; Pavelka, A.; Biedermannova, L.; Damborsky, J. Software tools for identification, visualization and analysis of protein tunnels and channels. *Biotechnol. Adv.* **2013**, *31*, 38-49. (d) Henrich, S.; Salo-Ahena, O. M. H.; Huang, B.; Rippmann, F.; Cruciani, G.; Wade, R. C. Computational approaches to identifying and characterizing protein binding sites for ligand design. *J. Mol. Recognit.* **2010**, *23*, 209-219.

¹³ Schrödinger, L. & DeLano, W., **2020**. PyMOL, Available at: <http://www.pymol.org/pymol> (PyMOL, 07-04-2023).

¹⁴ <https://www.wwpdb.org> (PDB Worldwide Protein Data Bank, 07-04-2023).

¹⁵ Guerra, J. V. d.; Ribeiro-Filho, H. V.; Jara, G. E.; Bortot, L. O.; Pereira, J. G. d.; Lopes-de-Oliveira, P. S. pyKVFinder: an efficient and integrable Python package for biomolecular cavity detection and characterization in data science. *BMC Bioinformatics* **2021**, *22*, 607-620.

¹⁶ Kawabata, T.; Go, N. Detection of pockets on protein surfaces using small and large probe spheres to find putative ligand binding sites. *Proteins: Structure, Function, and Bioinformatics* **2007**, *68*, 516-529.

¹⁷ Del Carpio, C. A.; Takahashi, Y.; Sasaki, S.-i. A new approach to the automatic identification of candidates for ligand receptor sites in proteins: (I) Search for pocket regions. *J. Mol. Graphics* **1993**, *11*, 23-29.

¹⁸ Zhu, H.; Pisabarro, M. T. MSPocket: an orientation-independent algorithm for the detection of ligand binding pockets. *Bioinformatics* **2010**, *27*, 351-358.

¹⁹ Voss, Neil R.; Gerstein, M. 3V: cavity, channel and cleft volume calculator and extractor. *Nucleic Acids Res.* **2010**, *38*, W555-W562.

²⁰ Oliveira, S. H. P.; Ferraz, F. A. N.; Honorato, R. V.; Xavier-Neto, J.; Sobreira, T. J. P.; Lopes-de-Oliveira, P. S. L. KVFinder: steered identification of protein cavities as a PyMOL plugin. *BMC Bioinformatics* **2014**, *15*, 197.

²¹ Guerra, J. V. d.; Ribeiro-Filho, H. V.; Bortot, L. O.; Honorato, R. V.; Pereira, J. G. d.; Lopes-de-Oliveira, P. S. ParKVFinder: A thread-level parallel approach in biomolecular cavity detection. *SoftwareX* **2020**, *12*, 100606-100614.

²² Le Guilloux, V.; Schmidtke, P.; Tuffery, P. Fpocket: An open source platform for ligand pocket detection. *BMC Bioinformatics* **2009**, *10*, 168.

-
- ²³ Liang, J.; Edelsbrunner, H.; Woodward, C. Anatomy of protein pockets and cavities: measurement of binding site geometry and implications for ligand design. *Protein science: a publication of the Protein Society* **1998**, *7*, 1884–1897.
- ²⁴ Barber, B. C.; Dobkin, D. P.; Huhdanpaa, H. 1996. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.* **1996**, *22*, 469–483.
- ²⁵ Miklitz, M.; Jelfs, K. E. pywindow: Automated Structural Analysis of Molecular Pores. *J. Chem. Inf. Model.* **2018**, *58*, 2387-2391.
- ²⁶ (a) Durrant, J. D.; de Oliveira, C. A. F.; McCammon, J. A. POVME: an algorithm for measuring binding-pocket volumes. *J. Mol. Graph. Model.* **2011**, *29*, 773–776. (b) Durrant, J. D.; Votapka, L.; Sørensen, J.; Amaro, R. E. POVME 2.0: An Enhanced Tool for Determining Pocket Shape and Volume Characteristics. *J. Chem. Theory Comput.* **2014**, *10*, 5047–5056. (c) Wagner, J. R.; Sørensen, J.; Hensley, N.; Wong, C.; Zhu, C.; Perison, T.; Amaro, R. E. *J. Chem. Theory Comput.* **2017** *13*, 4584-4592.
- ²⁷ Durrant, J. D.; McCammon, J. A. BINANA: a novel algorithm for ligand-binding characterization. *J. Mol. Graph. Model.* **2011**, *29*, 888–893.
- ²⁸ Kawabata T. Detection of multiscale pockets on protein surfaces using mathematical morphology. *Proteins: Structure, Function, and Bioinformatics* **2010**, *78*, 1195–1211.
- ²⁹ (a) Ripley, B. D. *Journal of the Royal Statistical Society. Series A (General)* **1976**, *139*, 277-278. (b) Diggle, P. J. *Biometrics* **1983**, *39*, 536-537.

³⁰ Petřek, M.; Otyepka, M.; Banáš, P.; Košinová, P.; Koča, J.; Damborský, J. CAVER: a new tool to explore routes from protein clefts, pockets and cavities *BMC Bioinformatics* **2006**, *7*, 316-325.

³¹ Chovancova, E.; Pavelka, A.; Benes, P.; Strnad, O.; Brezovsky, J.; Kozlikova, B.; Gora, A.; Sustr, V.; Klvana, M.; Medek, P.; Biedermannova, L.; Sochor, J.; Damborsky, J. CAVER 3.0: A Tool for the Analysis of Transport Pathways in Dynamic Protein Structures. *PLoS Comput. Biol.* **2012**, *8*, e1002708.

³² Jurcik, A.; Bednar, D.; Byska, J.; Marques, S. M.; Furmanova, K.; Daniel, L.; Kokkonen, P.; Brezovsky, J.; Strnad, O.; Stourac, J.; Pavelka, A.; Manak, M.; Damborsky, J.; Kozlikova, B. CAVER Analyst 2.0: analysis and visualization of channels and tunnels in protein structures and molecular dynamics trajectories. *BMC Bioinformatics* **2018**, *34*, 3586-3588.

³³ Maglic, J. B.; Lavendomme, R. MoloVol: an easy-to-use program for analyzing cavities, volumes and surface areas of chemical structures. *J. Appl. Cryst.* **2022**, *55*, 1033-1044.

³⁴ <https://www.ccdc.cam.ac.uk> (The Cambridge Crystallographic Data Centre, 07-04-2023).

³⁵ Mecozzi, S.; Rebek, J. The 55 % Solution: A Formula for Molecular Recognition in the Liquid State. *Chem. Eur. J.* **1998**, *4*, 1016-1022.

³⁶ Pluth, M. D.; Johnson, D. W.; Szigethy, G.; Davis, A. V.; Teat, S. J.; Oliver, A. G.; Bergman, R. G.; Raymond, K. N. Structural Consequences of Anionic Host–Cationic Guest Interactions in a Supramolecular Assembly. *Inorg. Chem.* **2009**, *48*, 111-120.

³⁷ (a) Steel, P. J.; McMorran, D. A. Selective Anion Recognition by a Dynamic Quadruple Helicate. *Chem. Asian J.* **2019**, *14*, 1098. (b) Johnson, D. W.; Hof, F.; Iovine, P. M.; Nuckolls, C.; Rebek, J. *Angew. Chem. Int. Ed.* **2002**, *41*, 3793-3796. (c) Lisboa, L. S.; Preston, D.; McAdam, C. J.; Wright, L. J.; Hartinger, C. G.; Crowley, J. D. Heterotrimetallic Double Cavity Cages: Syntheses and Selective Guest Binding. *Angew. Chem. Int. Ed.* **2022**, *61*, e202201700.

TOC

**Cavity Characterization in
Supramolecular Cages**

João V. S. Guerra, Luiz F. G.
Alves, Didier Bourissou,
Paulo S. Lopes-de-Oliveira,*
and György Szalóki*

