CageCavityCalc (C3): A computational tool for calculating and visualizing cavities in Molecular Cages

Vicente Martí-Centelles,[*,a,b,c] *Tomasz K. Piskorz,*[d] *Fernanda Duarte,*[*,d]

[a] Instituto Interuniversitario de Investigación de Reconocimiento Molecular y Desarrollo Tecnológico (IDM), Universitat Politècnica de València, Universitat de València, Valencia 46022, Spain

[b] CIBER de Bioingeniería Biomateriales y Nanomedicina, Instituto de Salud Carlos III, Spain

[c] Departamento de Química, Universitat Politècnica de València, Camí de Vera s/n, 46022, Valencia, Spain

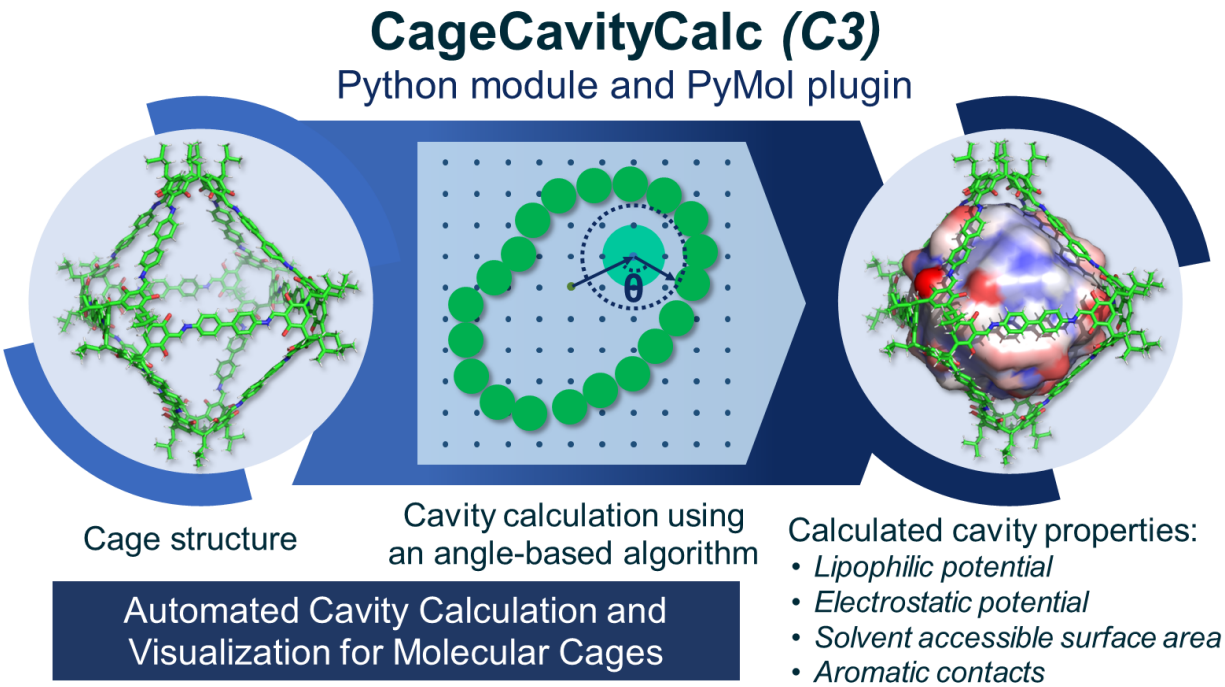
[d] Chemistry Research Laboratory, University of Oxford, Mansfield Road, Oxford OX1 3TA, United Kingdom

ABSTRACT

Organic(porous) and metal-organic cages are promising biomimetic platforms with diverse applications spanning recognition, sensing and catalysis. The key to the emergence of these functions is the presence of well-defined inner cavities capable of binding a wide range of guest molecules and modulating their properties. However, despite the myriad cage architectures currently available, the rational design of structurally diverse and functional cages with specific host–guest properties remain challenging. Efficiently predicting such properties is critical for accelerating the discovery of novel functional cages.

Herein, we introduce *CageCavityCalc (C3)*, a Python-based tool for calculating the cavity size of molecular cages. The code is available on GitHub at https://github.com/VicenteMartiCentelles/CageCavityCalc. *C3* utilizes a novel algorithm that enables the rapid calculation of cavity sizes for a wide range of molecular structures and porous systems. Moreover, *C3* facilitates easy visualization of the computed cavity size alongside hydrophobic and electrostatic potentials, providing insights into host-guest interactions within the cage. Furthermore, the calculated cavity can be visualized using widely available visualization software, such as PyMol, VMD, or Chimera. To enhance user accessibility, a PyMol plugin has been created, allowing non-specialists to use this tool without requiring computer programming expertise. We anticipate that the deployment of this computational tool will significantly streamline cage cavity calculations, thereby accelerating the discovery of functional cages.

GRAPHICAL ABSTRACT



1 Introduction

Discrete three-dimensional (3D) organic and metal-organic (porous) cages have emerged as promising biomimetic systems.[1,2,3] They offer synthetic modularity and tunability, enabling chemists to efficiently create structures with customized sizes and shapes from simple building blocks.[4,5,6,7] A critical factor contributing to their functional properties is the presence of a well-defined inner cavity capable of binding and even catalyzing chemical reactions, prompting chemists to draw parallels between these systems and enzymes.[8,9,10,11]

In molecular cages, the affinity of the host towards a particular guest depends on various structural and electronic parameters.[12] A key structural parameter used to assess the ability of a cage to act as a host is the relative cavity size of the cage and guest molecules. For instance, Rebek determined that 'closed' organic capsules exhibit binding when the guest occupies around 55% of the host cavity volume.[13] Since then, this rule has been extended to other supramolecular structures with varying success,[14,15,16] and also to enzymes.[17]

Several tools have been developed for the identification and characterization of protein binding pockets based on grid, rolling probe, or tessellation algorithms, including VOIDOO,[18] McVol,[19] Volarea,[20] GRASP,[21] CAVER,[22] CAST,[23] HOLLOW,[24] MDpocket,[25] ProteinVolume,[26] 3V,[27] Voronoia,[28] PoreBlazer,[29] Zeo++,[30] and CavVis.[31] However, most of these cannot be easily adapted for supramolecular cages, and software exclusively dedicated to cage architectures is scarce,[32] with PyWindow,[33] and MoloVol[34] being the only notable examples.

Most of these programs designed for proteins and cages utilize a *rolling probe* algorithm, typically using one or two probes.[35,36] The one-probe algorithm is best suited for cavities with small windows, which are the apertures within a cage structure connecting its enclosed cavity with the

external environment. This algorithm defines the cavity as the volume enclosed by a probe "rolling" around the entire cage structure without escaping (**Figure 1a**).[35,35] However, a limitation of the rolling one-probe algorithm is its failure when the probe's radius is smaller than the cage's windows, causing the probe to escape and leading to an overestimation of the volume. This problem can be solved by increasing the probe size to prevent it from escaping, but this underestimates the cavity volume as the gaps between the probe and the atoms of the cage are larger.

On the other hand, the rolling two-probe algorithm is suitable for cavities with larger windows, solving the probing escaping from the cavity issue encountered in the one-probe method. In this approach, the cavity is defined as the volume enclosed by a small probe, which cannot escape, when a second, larger probe blocks the cage's windows (**Figure 1b**). However, the two-probe rolling method can fail if the radius of the larger probe is too small, allowing it to move inside the cavity through the cage windows, and resulting in inaccurate volume calculations. In contrast, if the larger probe is too big, the calculated volume is overestimated by an "artificial volume" generated by the gap between the larger probe and the cage atoms. Indeed, this is a significant limitation of the rolling probe algorithm, as the success of the cavity volume calculation depends on the user-defined probe(s) radii, which must be carefully chosen and may not be universally applicable to all systems. Therefore, the results can be improved by a delicate fine-tuning of probe(s) radii in systems with large windows. This process is laborious and difficult to generalize across different systems or when analyzing molecular dynamics (MD) trajectories, where the window sizes dynamically change along the simulation time. Alternatively, one can calculate the largest sphere that fits into the cavity, as done in PyWindow, but this compromises the accuracy of the computed volume as cavities in cages often deviate from a perfect spherical shape.

To address the limitations of the *rolling one-* and *two-probe* methods for cavity calculation, we have developed an approach that uses an angle measurement technique (**Figure 1c**). This method relies on determining the angle θ formed by the center of mass (COM) of the cage, the probe, and the atom of the cage. For probes that do not overlap with cage atoms, one calculates the average of all angles θ for each cage atom within the threshold distance (automatically calculated as the diameter of the maximum escape sphere from the cavity). If the average angle is greater than 90º the probe is considered to be inside the cage; otherwise, it is considered to be outside of the cavity. The angle-based method offers a practical, easy-to-implement, and computationally efficient method that eliminates the need for manual parameter optimization, facilitating automated analyses across cages of different sizes, shapes, and window dimensions.

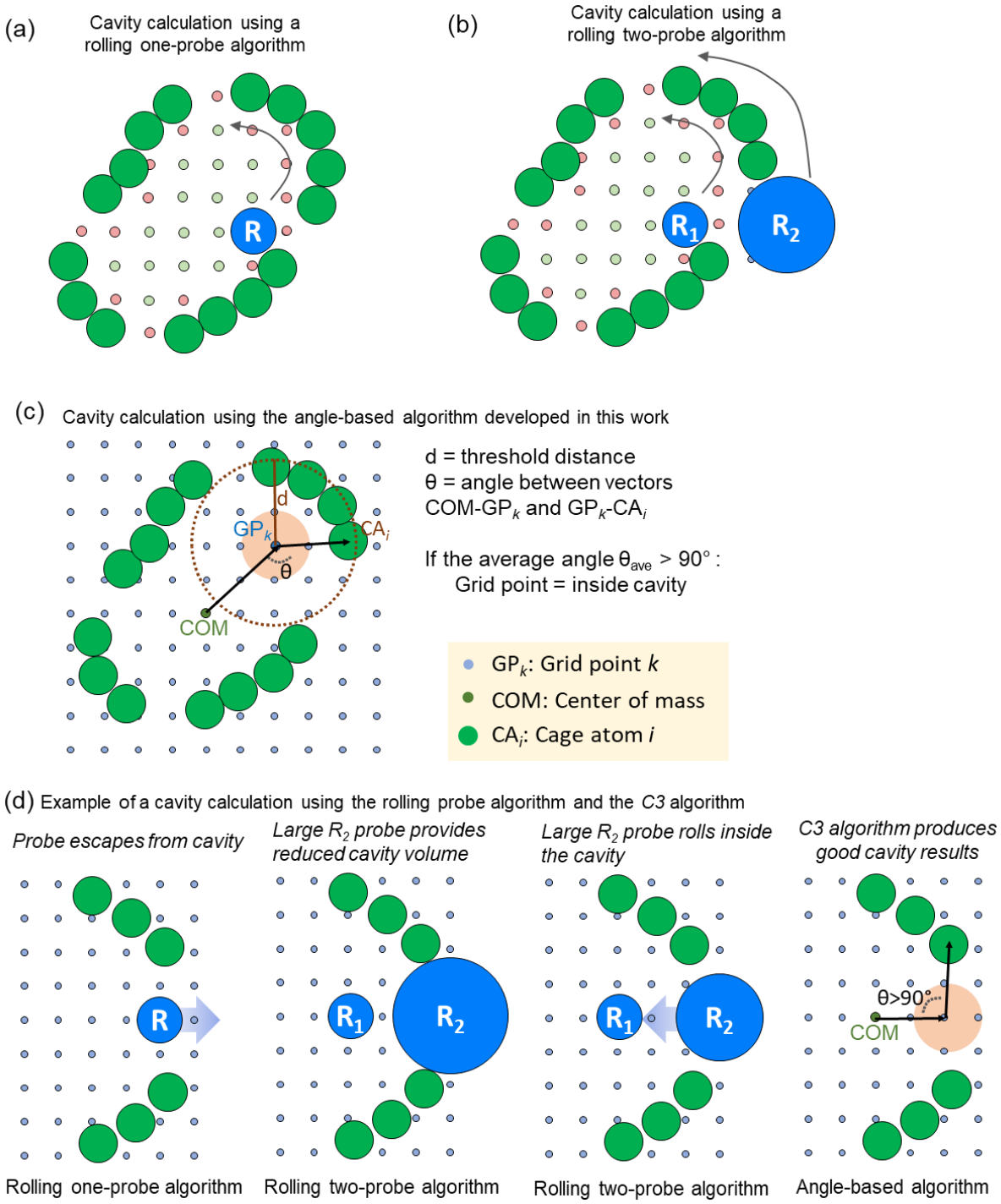


Figure 1. Schematic representation of cavity calculation algorithms using rolling probes (R, denoted by blue spheres: (a) *Rolling one-probe* algorithm: The probe moves within the cavity identifying accessible areas as cavity (green points) and, if the probe collides with cage atoms, these grid points are designated as non-cavity (blue). The calculation fails if the probe exits the cavity. (b) *Rolling two-probe* algorithm: it employs an inner small probe, R_1,

and a larger outer probe, R_2, which roll from the outside, blocking the cage windows to prevent the inner probe from exiting the cavity. However, if the radius of R_2 is small relative to the cage windows, it may enter the cavity. (c) *C3* angle-based algorithm developed herein: it uses the cavity's angle θ between the two vectors COM-GP_k and GP_k-CA_i. Grid point size is automatically determined by the grid resolution. (d) Illustrative examples that highlight the advantages of the angle-based algorithm over rolling probe methods.

In addition to host-guest size complementarity, guest binding is significantly influenced by electrostatic,[37,38] hydrogen bonding,[39] and Van der Waals interactions[37,40] with the host, which can be significantly affected by the solvent.[41] For example, in water, the hydrophobic effect plays a key role in driving guest binding.[42,43,44] Ward and coworkers demonstrated that hydrophobic guests exhibit 2−3 orders of magnitude higher association constants compared to polar guests,[45,46] Similarly, Ballester and coworkers established a linear relationship between binding free energies and the surface area of the non-polar guests' fragments, indicating a hydrophobic effect of 33–38 cal mol^{-1} $Å^{-2}$.[47]

Electrostatic complementarity is often visualized through electrostatic potential (ESP) surfaces.[48,49] Molecular ESP can be calculated using *ab initio* methods or from charge distributions obtained by numerically solving the Poisson–Boltzmann equation or its approximate form, the generalized Born model.[50,51] Additionally, hydrophobic interactions can be indirectly estimated using methods based on hydrophobic-lipophilic interactions (HLI), which quantify the tendency of nonpolar molecules to avoid contact with polar solvents, inducing aggregation and self-coiling.[52,53] Another parameter, molecular hydrophobicity potential (MHP), derived from 1-octanol/water partition coefficient (logP),[54,55] has proven useful in describing hydrophobicity in proteins,[56] analyzing protein pockets in docking engines, and predicting protein-ligand binding affinities.[57,58,59,60] However, despite its utility in ligand-protein binding analysis, the MHP

descriptor has not been commonly used to analyze cage cavities, likely due to the lack of user-friendly tools for their generation.

In this work, we introduce *C3*, a tool that calculates the cavity size electrostatic and hydrophobic potentials of molecular cages. The developed Python module can be used from the command line, python script, or as a plugin to the molecular visualization program PyMol.[33,34] Overall, the developed module enables the efficient characterization of cavity and host-guest properties.

Herein, we outline the methodology and demonstrate the ability of *C3* to determine the cavity volume of molecular cages with diverse sizes and shapes. We illustrate the utility of *C3* and highlight its advantages in terms of accuracy compared to rolling probe methods when assessing cavities of varying cages. Furthermore, we employ *C3* to compute the MHP and ESP surfaces in the calculated cavities aiding the determination of host-guest properties.

2 Results and Discussion

2.1 Software Outline

C3 is a stand-alone Python module designed for the characterization of organic and metallocage cavities. It employs scientific programming libraries, including NumPy,[61] SciPy,[62] and scikit-learn.[63] Optional functionalities can be enabled using the chemical programming libraries RDKit,[64] MDAnalysis,[65] OpenBabel,[66] and cgbind,[67] which facilitate the handling of chemical structures, including the identification of functional group and the calculation of molecular properties.

The core of the code is based on the *Cavity* class, featuring several functions: file reading, center of mass calculation, 3D grid setup, identification of grid points within the cage calculation. Of MHP and ESP, and the subsequent saving of results in PDB and/or PyMol formats. Additionally,

the *GridPoint* and *CageGrid* classes define attributes associated with each grid point in the 3D grid. The code also includes different functions for assigning hydrophobic values to the cage atoms, computing partial charges of the cage atoms, and calculating the maximum radius escape sphere, among other functions (See SI §S1 for a full description). Additionally, it offers a PyMol plugin with the corresponding *C3* graphical user interface to set up all calculation parameters of *C3* (see details in SI). **Figure 2** illustrates a simplified schematic diagram of *C3* functionality.

Upon loading the 3D cage structure into *C3*, a 3D grid filled with grid points is generated, with customizable size and grid spacing (default parameters are described in SI §S2). The cavity size is then calculated by iteratively examining all grid points to determine whether they form part of the cage cavity.

Once the cavity size is calculated, several properties can be computed using the molecular properties of each cage atom. These properties include aromatic contacts, solvent-accessible surface area (SASA), hydrophobicity (MHP), and/or electrostatic potential (ESP). The calculated values are stored as the B-factor in the generated cavity PDB file, facilitating their visualization in any standard molecular visualization software. Subsequent sections detail each of these steps.

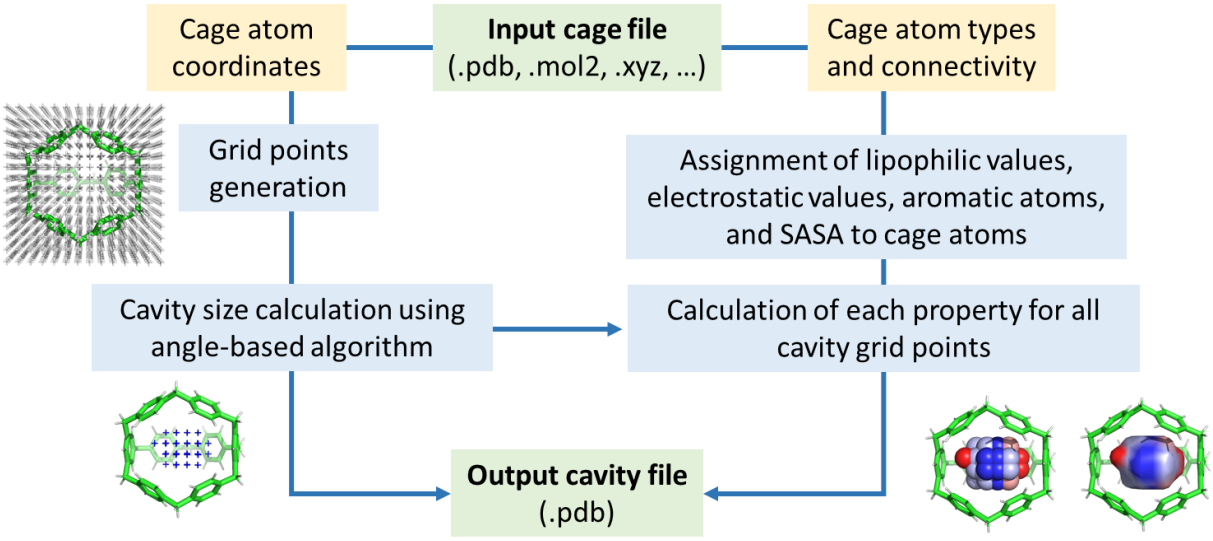


Figure 2. *C3* workflow to determine the cage cavity and the properties of each cavity grid point.

2.2 Overview of the software

2.2.1 Loading of the cage structure

The first step in *C3* involves loading the cartesian coordinates of the cage structure obtained by the user from either a crystal structure (CIF files) or via molecular modeling, employing software such as Stk[68] or cgbind.[67] *C3* supports various molecular file formats such as .xyz, .pdb, .mol, .mol2, and others handled by MDAnalysis.[65] The plugin stores the atom types and the cartesian coordinates in NumPy arrays.

2.2.2 Cavity volume calculation

In *C3*, the cavity volume calculation is performed without requiring user-defined parameters, as the default settings yield satisfactory results across a wide range of cages. The algorithm starts by generating a box around the cage structure with dimensions to fit all cage atoms followed by the generation of a 3D grid. By default, the grid spacing is set to 1 Å, but users can modify it. For

example, for cavity volumes exceeding 5000 $Å^3$ a grid size of 2 Å will be equally accurate and reduce computation time. The calculation of the cavity volume involves iterating over grid points and classifying them as either part of the cavity or outside of it. This selection involves two steps:

(a) Locate all cage atoms within a threshold distance from the selected grid point. By default, the threshold is set to the diameter of the maximum escape sphere from the cavity,[67] ensuring that cages with large windows are correctly processed without manual adjustments.

(b) For each grid point the cavity angle θ is calculated as the angle between two vectors: the vector defined by the center of mass of the cage (COM) and the selected grid point *k* (COM-GP_k), and the vector defined by the selected grid point *k* and the atom of the cage *i* (GP_k-CA_i)(see description of the angle in **Figure 1c**). The θ angle is calculated for each cage atom *i* in the threshold distance. Then all θ cavity angles are averaged to obtain $\theta_{average}$; if the average cavity angle $\theta_{average}$ is larger than 90º, the atom is considered inside the cage, otherwise, it is considered outside the cavity.

The process of iterating over all cage atoms within the threshold distance is sped up by using the SciPy Spatial KDTree algorithm, which partitions multidimensional data into a binary tree structure, enabling efficient nearest-neighbor searches and spatial queries.[62] To achieve this, the cartesian positions of all cage atoms are stored in a KDTree dictionary by atom types, significantly reducing the time required for obtaining atoms within a threshold distance compared to brute-force iteration. Once the iteration over all grid points in the grid is completed, and all grid points are assigned as within or outside the cavity, *C3* uses a DBSCAN clustering algorithm to identify the main cavity region, disregarding isolated grid points that do not correspond to the cavity.[69] In cases where more than one cavity is identified, the main cavity is selected based on the largest cluster, or alternatively, the cluster closest to the cage center of mass.

After extensive testing, we developed an algorithm that effectively addresses challenges encountered in dealing with large cavities, where the rolling probe method fails due to the inner probe escaping from the cavity. This is demonstrated when considering Cage **1**, an organic cage that features eight large windows, originally reported by Yuan and coworkers (**Figure 3a–c**).[70] In this system, the rolling two-probe method implemented in MoloVol struggles as the smaller probe escapes through the windows, thus requiring the use of a large radius for the outside probe (**Figure 3a**). This causes an overestimation of the volume due to the gap generated between the cage and the probe. For Cage **1**, the rolling two-probe method in MoloVol, using a probe radius of 11 Å and 1.4 Å for the large and small probe and a grid resolution of 0.5 Å, results in a volume of 8,934 Å^3 [34] (**Figure 3b**). However, this value represents an upper limit, as an "artificial volume" is generated by the gap between the cage and the probe. Attempts to minimize this "artificial volume" by reducing the large probe radius to 10 Å resulted in the large probe entering the cavity. To overcome this limitation, Yuan et al. manually blocked the cage windows using a "virtual plane", obtaining a volume of 7,026 Å^3 using the one-probe rolling probe method (probe radius of 1.4 Å) using VOIDOO[18] (**Figure 3c**).[70] Utilizing *C3*, with a 2.0 Å grid spacing, we obtained a volume of 6,800 Å^3, which is comparable to the volume obtained by Yuan and coworkers (**Figure 3b**).

We also tested *C3* on cages with small windows, where the rolling probe algorithm is expected to perform better. We analyzed Cage **2**, reported by Mastalerz and coworkers (**Figure 3d-f**).[71] In this case, the different tested algorithms gave a similar cavity volume, in the range 226–253 Å^3. The volume reported in reference 71 was 253 Å^3 using a probe radius of 1.4 Å in SwissPDBViewer (**Figure 3f**).[72] Using *C3* we obtained a volume of 237 Å^3 using a 0.5 Å grid spacing (**Figure 3d**). These results are comparable with those obtained with the rolling two-probe method in

MoloVol,[34] obtaining a volume of 226 Å^3 using a large probe radius of 3 Å, a small probe radius of 1.0 Å, and a grid resolution of 0.5 Å (**Figure 3e**). These results are therefore in line with the expected good performance of rolling probe methods when considering cages with small window sizes.

To demonstrate the utility of the *C3* algorithm, we tested it in 23 cage structures, successfully calculating cavity volumes ranging from 18 Å^3 to 66,022 Å^3 (see SI §S3 for details of cavity calculation of all tested examples). These examples illustrate how *C3* can accurately calculate cavity volumes for cages with varying cavity and window sizes.

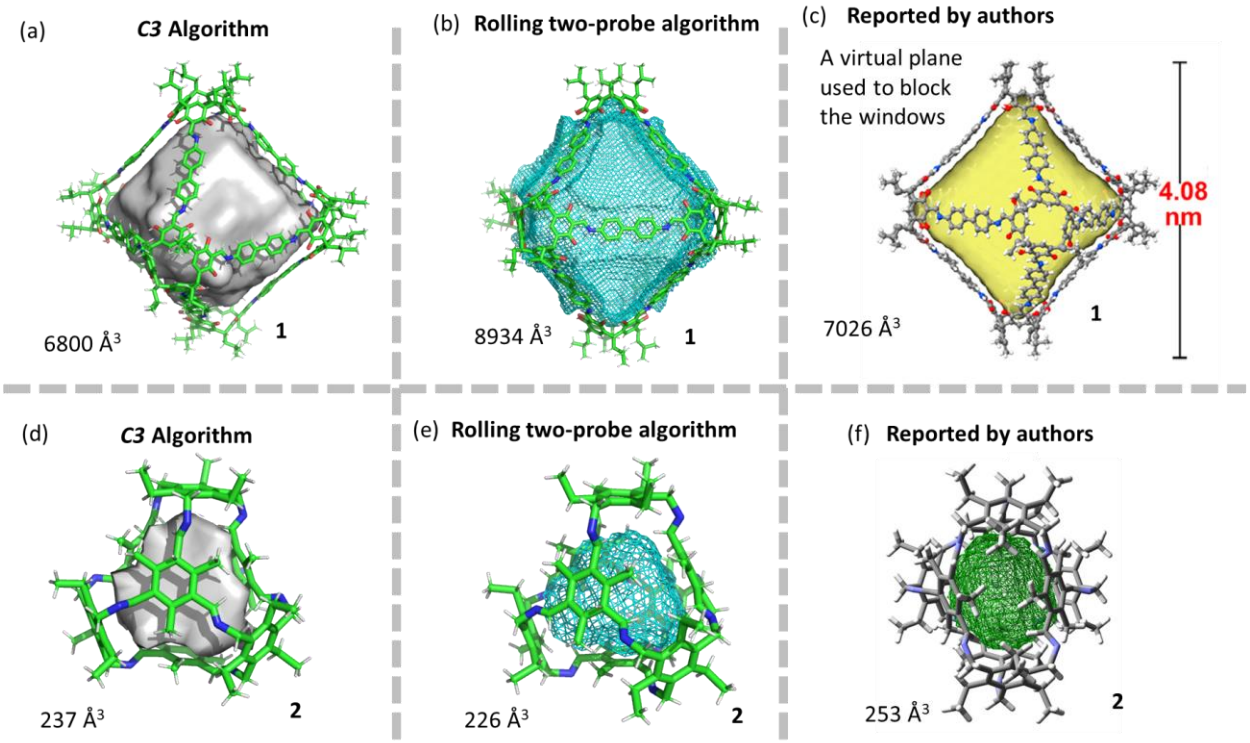


Figure 3. Comparison of the cavity calculated with different software. Cage **1** (CCDC 2014846)[70]: (a) *C3* using a resolution 2 Å, (b) Rolling two-probe algorithm using MoloVol with a large probe radius of 14 Å, a small probe radius of 3 Å, and grid resolution of 3 Å, (c) Calculation reported in Ref. 70 using VOIDOO and a virtual plane to block the cage's windows. Cage **2** (CCDC 1970309). (d) *C3* using a resolution of 0.5 Å, (e) Rolling two-probe

algorithm using MoloVol with a large probe radius of 3 Å, a small probe radius of 1.0 Å, and grid resolution of 0.5 Å, (f) Calculation reported in Ref. 71 using SwissPDBViewer with a probe radius of 1.4 Å. [72] Figure 3c adapted with permission from Ref. 70. Copyright 2020 American Chemical Society. Figure 3f adapted with permission from Ref. 71. Copyright 2020 John Wiley and Sons.

An important feature of our algorithm is its compatibility with MD simulations, enabling the assessment of dynamic changes in volume over time. To illustrate this, we analyzed the MD trajectory of the $[Pd_4L_6]^{12+}$, Cage **3,**[73] using VOIDOO software that employs the rolling probe algorithm and our *C3* software. This comparison highlighted the advantages of the C3 angle-based algorithm over the rolling one-probe algorithm (**Figure 4**). The calculation of the cavity volume with VOIDOO with a standard probe size of radius 1.4 Å exhibited significant variations in the calculated volume over time of *ca.* 30% (V_{cavity} = 307 ± 97 Å^3), primarily due to over-/under-estimating the cavity volume in several frames, giving 0 values in some instances (see **Figure 4** blue line). While increasing the probe size (radius = 2.4 Å) mitigated this issue by preventing the probe from escaping from the cavity, it led to a much smaller cavity size with a *ca.* 60% variation over time (V_{cavity} = 57 ± 35 Å^3, see **Figure 4** green line). In contrast, our *C3* algorithm with default parameters provided a more robust and well-defined cavity with a volume variation over time of *ca.* 10% (V_{cavity} = 323 ± 31 Å^3, see **Figure 4** orange line). These results highlight the robustness of the angle-base algorithm of *C3* for analyzing MD trajectories.

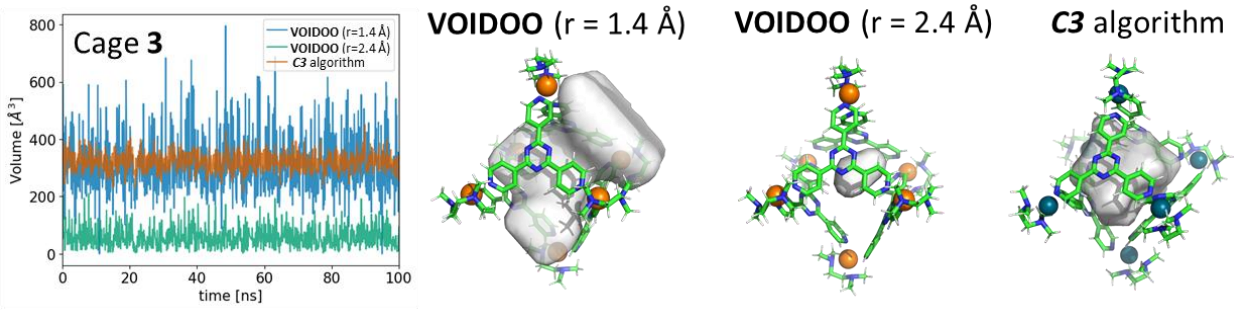


Figure 4. Analysis of the volume change during a 100 ns molecular dynamics trajectory of $[Pd_6L_4]^{12+}$ cage (CCDC 794242) using the rolling probe algorithm in VOIDOO with two different sizes of the probe (1.4 Å and 2.4 Å) and the angle-based method in *C3* (grid spacing = 1.0 Å).

We showed that the *C3* algorithm is efficient in calculating cavity volumes on cages with all types of shapes, cavity sizes, and window sizes. The angle-based algorithm of *C3* is especially efficient with cages with large windows, where the rolling probe algorithm fails due to the probe escaping from the cavity. Once the cavity calculation is completed, various properties of the cavity are computed for a more comprehensive understanding of the cavity's characteristics and the interaction with potential guest molecules, thereby aiding in enhancing and designing novel host-guest complexes. These properties include aromatic contacts, solvent accessible surface area (SASA), hydrophobicity (MHP) and/or electrostatic potential (ESP) described below.

2.2.3 Hydrophobic potential of the cavity

The hydrophobicity of the cage cavity serves to identify favorable interactions with nonpolar guests in aqueous media. This is achieved by assigning hydrophobic contributions (F_i) to each atom in the cage, which was tabulated by Ghose and coworkers for various atom types based on their contributions to the 1-octanol/water partition coefficient.[74]

Subsequently, the molecular hydrophobicity potential (MHP) is computed for each grid point within the cavity. MHP represents the cumulative hydrophobic contributions (F_i) from all neighboring N atoms, weighted by a distance function $f(d_{ik})$

$$\mathrm{MHP}_k = \sum_{i=1}^{N} F_i \cdot f(d_{ik}) \qquad \text{Eqn 1}$$

The commonly used distance functions are:

$$f(d_{ik}) = \frac{1}{1+d} \qquad \text{Audry's distance function} \qquad \text{Eqn 2}$$

$$f(d_{ik}) = e^{-d_{ik}} \qquad \text{Fauchère's distance function} \qquad \text{Eqn 3}$$

$$f(d_{ik}) = e^{\frac{-d_{ik}}{2}} \qquad \text{Modified Fauchère's distance function} \qquad \text{Eqn 4}$$

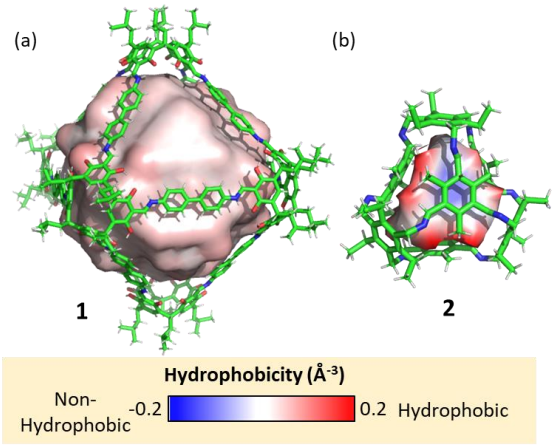
These functions exhibit maxima at zero followed by decay as distance increases, reflecting the diminishing hydrophobic influence from distant atoms. Audrey's function attempts to simulate the decay mimicking Coulomb interaction (Eqn 2). Despite lacking a physical foundation, it has proven useful for generating informative visualizations. Fauchère proposed an alternative exponential decay function based on observed proximity effects during octanol/water partition calculations (Eqn 3). Both Audry's and Fauchère's distance functions do not appear to be adequate beyond the Solvent-Accessible Surface (SAS) and the slightly modified Fauchère's distance function (Eqn 4) overcomes this limitation.[75,76,77]

The Hydrophobic Index (HI) is defined based on the distinction between negative MHP values (MHP^-) associated with polar regions and positive MHP values (MHP^+) corresponding to hydrophobic regions, as shown in Eqn 5.[57,58]

$$HI = \frac{MHP^+}{MHP^+ + |MHP^-|} \qquad \text{Eqn 5}$$

The calculated hydrophobic potential information is then stored as B-factor data for each cavity grid point alongside the original cage coordinates in a PDB format. This format enables users to visualize these values easily using standard visualization software such as Chimera, VMD, or PyMol (**Figure 5**).

To illustrate the use of *C3* for calculating cavity properties, we computed the MHP for Cages **1** and **2**. The average cavity hydrophobicity, 0.01 for Cage **1** and 0.09, for Cage **2**, indicates that the cavity of Cage **2** is 9 times more hydrophobic than Cage **1** (**Figure 5**). This illustrates that small cavities efficiently isolated by the cage walls are much more hydrophobic than large cages containing large windows. However, the algorithm is unsuitable for calculating the hydrophobicity of metallocages due to the lack of tabulated hydrophobic contributions for metals.



Cage	Cavity Volume (Å^3)	Average Cavity Hydrophobicity (Å^{-3})	Total cavity Hydrophobicity	Hydrophobic index (HI)
1	6800	0.01	97.9	1.00
2	225 Å^3	0.09	19.4	1.00

Figure 5. Calculation of the hydrophobicity of cages with the hydrophobic potential mapped on the cavity: (a) Cage **1** (CCDC 2014846 from Ref. 70) using 2.0 Å grid spacing; and (b) Cage **2** (CCDC 1970309 from Ref. 71) using 0.5 Å grid spacing.

2.2.4 Electrostatic potential of the cavity

The electrostatic potential (ESP) is a key parameter to understand the nature of the interactions within a molecular cage and predict the interaction with potential guests, particularly polar guests. We have implemented the calculation of the ESP in *C3* by mapping the partial charge contributions from the cage atoms onto a grid representing the cavity:

$$V_{grid} = k \sum_{i=0}^{N_{cage}} \frac{q_i}{d_{(grid,i)}}$$

Where k is the Coulomb constant, q_i is a partial charge of a cage atom, and $d_{(grid,i)}$ is the distance between the cage atom and the cavity grid point. The partial charges are determined using the electronegativity equalization method (EEM) implemented in Open Babel,[66] a widely accepted and efficient procedure for deriving charge-based descriptors in QSAR studies.[78,79,80]

The cavity ESP is a valuable tool for identifying favorable interactions between hosts and polar guests. For example, by computing the ESP for a $[Pd_2L_4]^{4+}$ cage and 1,4-dicyanobenzene as a potential guest, one can easily visualize the complementarity of their ESPs, suggesting the formation of a strongly bound host-guest complex (**Figure 6a-c**).[81] Negatively charged cages are relatively less common than positively charged ones. Nevertheless, we present an example of a purely organic cage, containing carboxylic groups, which results in a cavity with a negative ESP. This feature suggests that the cage can potentially encapsulate and stabilize positively charged guest molecules (**Figure 6d**).[82]

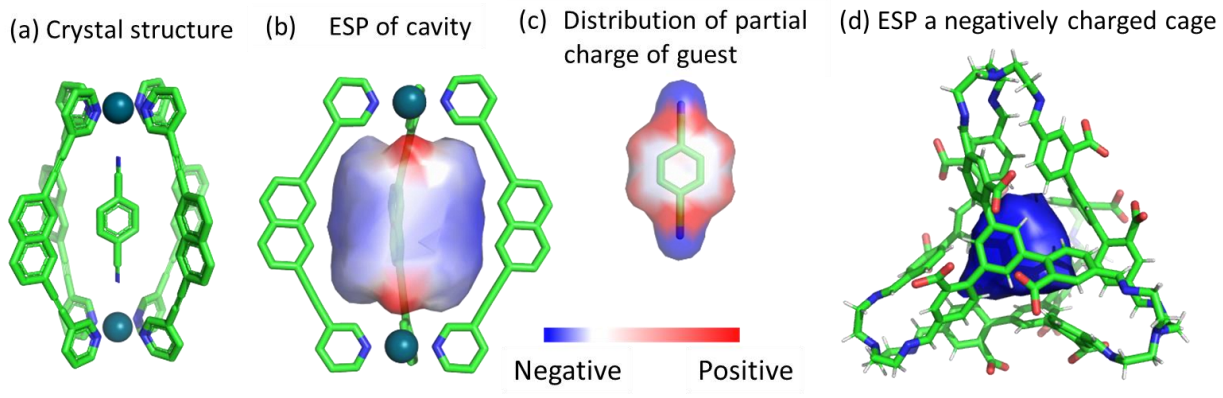


Figure 6. (a) Crystal structure of the host-guest complex $[Pd_2L_4]^{4+}$ (CCDC 2157968), (b) electrostatic potential mapped on a cavity of the empty cage $[Pd_2L_4]^{4+}$ (CCDC 2157967), (c) partial charge mapped on van der Waals sphere of 1,4-dicyanobenzene calculated using EEM implemented in OpenBabel (d) Example of the ESP of the negatively charged cage (CCDC 2053043). The ester groups of the original crystal structure have been replaced by carboxylic groups to show the negative charge effect on the cavity.

2.3 PyMol Plugin

In addition to the Python module and the command line interface, we have developed a graphical user interface (GUI) as a PyMol plugin that requires no programming skills. This GUI enables users to select a molecule and perform calculations with just a few clicks. To use it, the user simply needs to load the molecule in PyMol, open the *C3* plugin, select the molecule from the drop-down list, and then press the "Calculate volume" button. Optionally, the user can adjust the grid size, select the properties to calculate (and adjust their parameters, such as the hydrophobicity method, the distance function, etc.), or choose a clustering algorithm to remove noisy cavity points, *i.e.* to remove isolated cavity points that do not belong to the main cavity. As an example, we provide the output obtained for the calculation of a $[Pd_2L_4]^{4+}$ cage providing in the same PyMol session the output obtained for the cavity, with an individual output for each selected property (**Figure 7**).

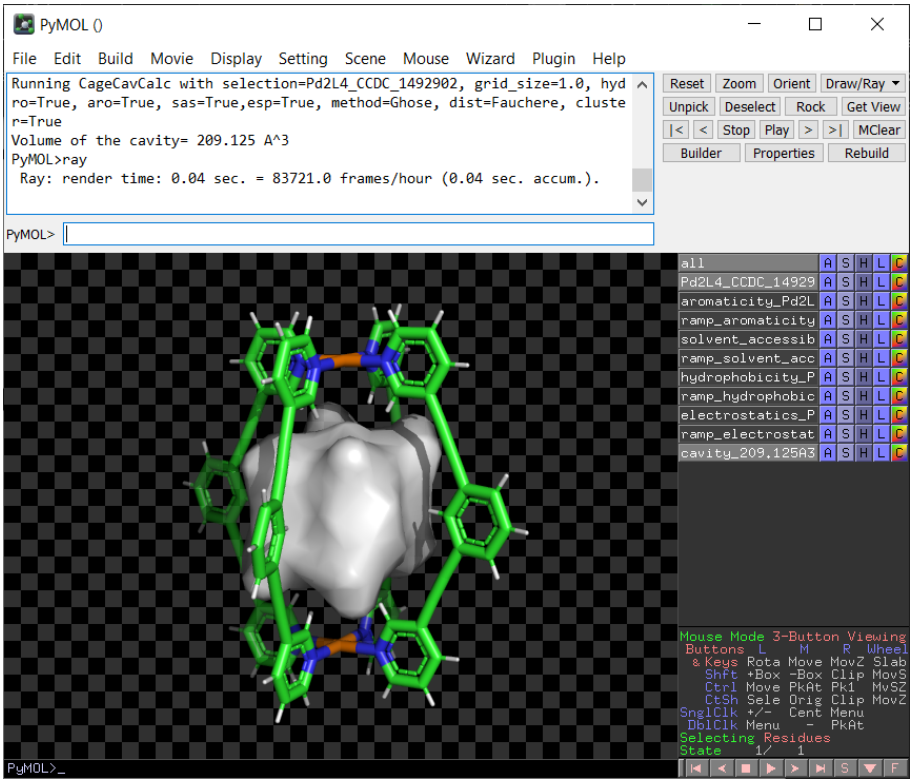


Figure 7. Screenshot of the results after running the *C3* PyMol plugin.

3 Conclusions

We have reported an automated cavity calculation software, *C3*, deployed as a Python module, for the calculation of the cavity of cages, as well as aromatic contacts, solvent accessible surface area (SASA), hydrophobicity (MHP), and/or electrostatic potential (ESP) contributions. The main advantage of our method is its easy use and general applicability to a wide range of porous structures without the need for parameter adjustment. Users can select the grid spacing to achieve the desired cavity resolution and reduce calculation time by using larger grid spacing. The method was benchmarked on a wide range of cage structures.

The cavity can be visualized with any chemical visualization software as the cavity output is stored in a PDB file containing the cavity grid points. To facilitate the use of the algorithm for non-specialized users, a plugin for molecular viewer PyMol was developed, enabling its use without requiring computer programming knowledge. We anticipate that the developed software will streamline the characterization of molecular cages and speed up the development of novel functional designs.

ASSOCIATED CONTENT

Data and code availability

Source code and associated Python files are freely available at https://github.com/VicenteMartiCentelles/CageCavityCalc

Supporting Information.

Description of the arguments that can be used in the *C3* Python module through the command line, description of the *C3* GUI of the PyMol plugin, and examples of cavities calculated for different cage structures are provided as supporting information (PDF).

AUTHOR INFORMATION

Corresponding Author

Vicente Martí-Centelles − Instituto Interuniversitario de Investigación de Reconocimiento Molecular y Desarrollo Tecnológico (IDM), Universitat Politècnica de València, Universitat de

València, 46022 Valencia, Spain; ORCID: https://orcid.org/0000-0002-9142-9392; Email: vimarce1@upv.es

Fernanda Duarte – Chemistry Research Laboratory, University of Oxford, Mansfield Road, Oxford OX1 3TA, United Kingdom; ORCID: https://orcid.org/0000-0002-6062-8209; Email: fernanda.duartegonzalez@chem.ox.ac.uk

Author Contributions

V.M.-C and T.K.P. wrote the code with support from all authors. All authors contributed to data analysis and writing of the manuscript. All authors have given approval to the final version of the manuscript.

ACKNOWLEDGMENTS

V.M.-C acknowledges financial support from Generalitat Valenciana (project CIDEGENT/2020/031) and MCIN/ AEI /10.13039/501100011033 (project PID2020-113256RA-I00). T.K.P and F.D. acknowledge the financial support from EPSRC (EP/W010666/1 and EP/W009803/1).

REFERENCES

[1] Montà-González, G.; Sancenón, F.; Martínez-Máñez, R.; Martí-Centelles, V. Purely Covalent Molecular Cages and Containers for Guest Encapsulation. *Chem. Rev.* **2022**, *122*, 13636-13708. https://doi.org/10.1021/acs.chemrev.2c00198

[2] Raee, E.; Yang ,Y.; Liu, T. Supramolecular structures based on metal-organic cages. *Giant*, **2021**, *5*, 100050. https://doi.org/10.1016/j.giant.2021.100050

[3] Piskorz, T. K.; Martí-Centelles, V.; Spicer, R. L.; Duarte, F.; Lusby, P. J. Picking the Lock of Coordination Cage Catalysis. *Chem. Sci.* **2023**, *14* (41), 11300–11331. https://doi.org/10.1039/d3sc02586a

[4] Martí-Centelles, V. Kinetic and thermodynamic concepts as synthetic tools in supramolecular chemistry for preparing macrocycles and molecular cages. *Tetrahedron Let.* **2022**, *93*, 153676. https://doi.org/10.1016/j.tetlet.2022.153676

[5] M. Lewis, J. E. Molecular engineering of confined space in metal–organic cages. *Chem. Commun.* **2022**, *58*, 13873-13886. https://doi.org/10.1039/D2CC05560K

[6] Mcconnell, A. J. Metallosupramolecular cages: from design principles and characterisation techniques to applications. *Chem. Soc. Rev.* **2022**, *51*, 2957–2971. https://doi.org/10.1039/d1cs01143j

[7] Benchimol, E.; Nguyen, B.-N. T.; Ronson, T. K.; Nitschke, J. R. Transformation Networks of Metal–organic Cages Controlled by Chemical Stimuli. *Chem. Soc. Rev.* **2022**, *51*, 5101–5135. https://doi.org/10.1039/d0cs00801j

[8] Kim, D.; Lee, H.-S.; Kim, Y.-M.; Kim, J.; Lee, C.-H. Metal-Organic Cages for Molecular Separations. *Nat. Rev. Chem.* **2020**, *4*, 40-53. https://doi.org/10.1038/s41570-020-00246-1

[9] Ham, R.; Nielsen, C. J.; Pullen, S.; Reek, J. N. H. Supramolecular Coordination Cages for Artificial Photosynthesis and Synthetic Photocatalysis. *Chem. Rev.* **2023**, *123* (9), 5225–5261. https://doi.org/10.1021/acs.chemrev.2c00759

[10] Martí-Centelles, V.; Lawrence, A. L.; Lusby, P. J. High Activity and Efficient Turnover by a Simple, Self-Assembled Artificial "Diels-Alderase". *J. Am. Chem. Soc.* **2018**, *140*, 2862-2868. https://doi.org/10.1021/jacs.7b12146

[11] Young, T. A.; Martí-Centelles, V.; Wang, J.; Lusby, P. J.; Duarte, F. Rationalizing the Activity of an "Artificial Diels-Alderase": Establishing Efficient and Accurate Protocols for Calculating Supramolecular Catalysis. *J. Am. Chem. Soc.* **2020**, *142*, 1300-1310. https://doi.org/10.1021/jacs.9b10302

[12] 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. https://doi.org/10.1021/acscatal.2c00837

[13] Mecozzi, S.; Rebek, Jr., J. The 55 % Solution: A Formula for Molecular Recognition in the Liquid State. *Chem. Eur. J.* **1998**, *4* (6), 1016–1022. https://doi.org/10.1002/(SICI)1521-3765(19980615)4:6<1016::AID-CHEM1016>3.0.CO;2-B

[14] Turega, S.; Cullen, W.; Whitehead, M.; Hunter, C. A.; Ward, M. D. Mapping the Internal Recognition Surface of an Octanuclear Coordination Cage Using Guest Libraries. *J. Am. Chem. Soc.* **2014**, *136* (23), 8475–8483. https://doi.org/10.1021/ja504269m

[15] Hristova, Y. R.; Smulders, M. M. J.; Clegg, J. K.; Breiner, B.; Nitschke, J. R. Selective Anion Binding by a "chameleon" Capsule with a Dynamically Reconfigurable Exterior. *Chem. Sci.* **2011**, *2* (4), 638–641. https://doi.org/10.1039/c0sc00495b

[16] Symmers, P. R.; Burke, M. J.; August, D. P.; Thomson, P. I. T.; Nichol, G. S.; Warren, M. R.; Campbell, C. J.; Lusby, P. J. Non-equilibrium Cobalt(iii) "click" Capsules. *Chem. Sci.* **2015**, *6* (1), 756–760. https://doi.org/10.1039/c4sc03036b

[17] Zürcher, M.; Gottschalk, T.; Meyer, S.; Bur, D.; Diederich, F. Exploring the Flap Pocket of the Antimalarial Target Plasmepsin II: The "55 % Rule" Applied to Enzymes. *ChemMedChem* **2008**, *3* (2), 237–240. https://doi.org/10.1002/cmdc.200700236

[18] Kleywegt, G. J.; Alwyn Jones, T. Detection, Delineation, Measurement and Display of Cavities in Macromolecular Structures. *Acta Crystallogr. Sect. D Biol. Crystallogr.* **1994**, *50* (2), 178–185. https://doi.org/10.1107/S0907444993011333

[19] Till, M. S.; Ullmann, G. M. McVol - A Program for Calculating Protein Volumes and Identifying Cavities by a Monte Carlo Algorithm. *J. Mol. Model.* **2010**, *16* (3), 419–429. https://doi.org/10.1007/s00894-009-0541-y

[20] Ribeiro, J. V.; Tamames, J. A. C.; Cerqueira, N. M. F. S. A.; Fernandes, P. A.; Ramos, M. J. Volarea - A Bioinformatics Tool to Calculate the Surface Area and the Volume of Molecular Systems. *Chem. Biol. Drug Des.* **2013**, *82* (6), 743–755. https://doi.org/10.1111/CBDD.12197

[21] Nicholls, A.; Sharp, K. A.; Honig, B. Protein Folding and Association: Insights From the Interfacial and Thermodynamic Properties of Hydrocarbons. *Proteins Struct. Funct. Genet.* **1991**, *11*, 281–296. https://doi.org/10.1002/prot.340110407

[22] 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* (1), 1–9. https://doi.org/10.1186/1471-2105-7-316

[23] Liang, J.; Edelsbrunner, H.; Woodward, C. Anatomy of Protein Pockets and Cavities: Measurement of Binding Site Geometry and Implications for Ligand Design. *Protein Sci.* **1998**, *7* (9), 1884–1897. https://doi.org/10.1002/PRO.5560070905

[24] Ho, B. K.; Gruswitz, F. HOLLOW: Generating Accurate Representations of Channel and Interior Surfaces in Molecular Structures. *BMC Struct. Biol.* **2008**, *8* (1), 49. https://doi.org/10.1186/1472-6807-8-49

[25] Schmidtke, P.; Bidon-chanal, A.; Luque, F. J.; Barril, X. MDpocket: Open-Source Cavity Detection and Characterization on Molecular Dynamics Trajectories. *Bioinformatics* **2011**, *27* (23), 3276–3285. https://doi.org/10.1093/BIOINFORMATICS/BTR550

[26] Chen, C. R.; Makhatadze, G. I. ProteinVolume: Calculating Molecular van Der Waals and Void Volumes in Proteins. *BMC Bioinformatics* **2015**, *16* (1). https://doi.org/10.1186/S12859-015-0531-2

[27] Voss, N. R.; Gerstein, M. 3V: Cavity, Channel and Cleft Volume Calculator and Extractor. *Nucleic Acids Res.* **2010**, *38* (SUPPL. 2), W555–W562. https://doi.org/10.1093/nar/gkq395

[28] Rother, K.; Hildebrand, P. W.; Goede, A.; Gruening, B.; Preissner, R. Voronoia: Analyzing Packing in Protein Structures. *Nucleic Acids Res.* **2009**, *37*, D393–D395. https://doi.org/10.1093/nar/gkn769

[29] Sarkisov, L.; Bueno-Perez, R.; Sutharson, M.; Fairen-Jimenez, D. Materials Informatics with PoreBlazer v4.0 and the CSD MOF Database. *Chem. Mater.* **2020**, *32* (23), 9849–9867. https://doi.org/10.1021/acs.chemmater.0c03575

[30] Willems, T. F.; Rycroft, C. H.; Kazi, M.; Meza, J. C.; Haranczyk, M. Algorithms and Tools for High-Throughput Geometry-Based Analysis of Crystalline Porous Materials. *Microporous Mesoporous Mater.* **2012**, *149* (1), 134–141. https://doi.org/10.1016/J.MICROMESO.2011.08.020

[31] Simões, T. M. C.; Gomes, A. J. P. CavVis—A Field-of-View Geometric Algorithm for Protein Cavity Detection. *J. Chem. Inf. Model.* **2019**, *59*(2), 786–796. https://doi.org/10.1021/acs.jcim.8b00572

[32] Guerra, J. V. S.; Alves, L. F. G.; Bourissou, D.; Lopes-De-Oliveira, P. S.; Szalóki, G. Cavity Characterization in Supramolecular Cages. *J. Chem. Inf. Model.* **2023**, *63*, 3772–3785, https://doi.org/10.1021/acs.jcim.3c00328.

[33] Miklitz, M.; Jelfs, K. E. Pywindow: Automated Structural Analysis of Molecular Pores. *J. Chem. Inf. Model.* **2018**, *58* (12), 2387–2391. https://doi.org/10.1021/acs.jcim.8b00490

[34] Maglic, J. B.; Lavendomme, R. MoloVol: An Easy-to-Use Program for Analyzing Cavities, Volumes and Surface Areas of Chemical Structures. *J. Appl. Crystallogr.* **2022**, *55* (4), 1033–1044. https://doi.org/10.1107/s1600576722004988

[35] Yu, J.; Zhou, Y.; Tanaka, I.; Yao, M. Roll: a new algorithm for the detection of protein pockets and cavities with a rolling probe sphere. *Bioinformatics* **2010**, *26*(1), 46–52. https://doi.org/10.1093/bioinformatics/btp599

[36] Simões, T.; Lopes, D.; Dias, S.; Fernandes, F.; Pereira, J.; Jorge, J.; Bajaj, C.; Gomes, A. Geometric Detection Algorithms for Cavities on Protein Surfaces in Molecular Graphics: A Survey. *Comput. Graph. Forum* **2017**, *36* (8), 643–683. https://doi.org/10.1111/cgf.13158

[37] Steed, J. W; Atwood, J. L. *Supramolecular Chemistry (2nd Edition)* John Wiley & Sons, Ltd, Chippenham, UK, 2009. https://doi.org/10.1002/9780470740880

[38] Moghaddam, S.; Inoue, Y.; Gilson, M. K. Host−Guest Complexes with Protein−Ligand-like Affinities: Computational Analysis and Design, *J. Am. Chem. Soc.* **2009**, *131* (11), 4012–4021. https://doi.org/10.1021/ja808175m

[39] Cooke, G.; Rotello, V. M. Methods of Modulating Hydrogen Bonded Interactions in Synthetic Host–guest Systems. *Chem. Soc. Rev.* **2002**, *31* (5), 275–286. https://doi.org/10.1039/b103906g

[40] Gómez-González, B.; García-Río, L.; Basílio, N.; Mejuto, J. C.; Simal-Gandara, J. Molecular Recognition by Pillar[5]arenes: Evidence for Simultaneous Electrostatic and Hydrophobic Interactions, *Pharmaceutics*, **2021**, *14* (1), 60. https://doi.org/10.3390/pharmaceutics14010060

[41] August, D. P.; Nichol, G. S.; Lusby, P. J. Maximizing Coordination Capsule-Guest Polar Interactions in Apolar Solvents Reveals Significant Binding. *Angew. Chem. Int. Ed.* **2016**, *55* (48), 15022–15026. https://doi.org/10.1002/anie.201608229

[42] Li, Y.-H.; Zhang, Y.; Legrand, Y.-M.; Van Der Lee, A.; Jiang, J.-J.; Chen, C.-X.; Su, C.-Y.; Barboiu, M. Hydrophobic Metallo-supramolecular Pd_2L_4 Cages for Zwitterionic Guest Encapsulation in Organic Solvents. *Dalton Trans.* **2017**, *46* (44), 15204–15207. https://doi.org/10.1039/c7dt03517a

[43] Hastings, C. J.; Pluth, M. D.; Biros, S. M.; Bergman, R. G.; Raymond, K. N. Simultaneously Bound Guests and Chiral Recognition: A Chiral Self-assembled Supramolecular Host Encapsulates Hydrophobic Guests. *Tetrahedron* **2008**, *64* (36), 8362–8367. https://doi.org/10.1016/j.tet.2008.05.131

[44] Brumaghim, J.; Michels, M.; Raymond, K. Hydrophobic Chemistry in Aqueous Solution: Stabilization and Stereoselective Encapsulation of Phosphonium Guests in a Supramolecular Host. *Eur. J. Org. Chem.* **2004**, *2004* (22), 4552–4559. https://doi.org/10.1002/ejoc.200400428

[45] Turega, S.; Cullen, W.; Whitehead, M.; Hunter, C. A.; Ward, M. D. Mapping the Internal Recognition Surface of an Octanuclear Coordination Cage Using Guest Libraries. *J. Am. Chem. Soc.* **2014**, *136* (23), 8475–8483. https://doi.org/10.1021/ja504269m

[46] Whitehead, M.; Turega, S.; Stephenson, A.; Hunter, C. A.; Ward, M. D. Quantification of Solvent Effects on Molecular Recognition in Polyhedral Coordination Cage Hosts. *Chem. Sci.* **2013**, *4* (7), 2744–2751. https://doi.org/10.1039/C3SC50546D

[47] Escobar, L.; Ballester, P. Quantification of the Hydrophobic Effect Using Water-Soluble Super Aryl-Extended Calix[4]Pyrroles. *Org. Chem. Front.* **2019**, *6* (11), 1738–1748. https://doi.org/10.1039/c9qo00171a

[48] Weiner, P. K.; Langridge, R.; Blaney, J. M.; Schaefer, R.; Kollman, P. A. Electrostatic Potential Molecular Surfaces. *Proc. Natl. Acad. Sci. U. S. A.* **1982**, *79*, 3754– 3758. https://doi.org/10.1073/pnas.79.12.3754

[49] Zhang, J.; Lu, T. Efficient evaluation of electrostatic potential with computerized optimized code. *Phys. Chem. Chem. Phys.* **2021**, *23*, 20323-20328. https://doi.org/10.1039/D1CP02805G

[50] A. Baker, N. Biomolecular Applications of Poisson–Boltzmann Methods, *Rev. Comput. Chem.* **2005**, *21*, 349-379. https://doi.org/10.1002/0471720895.ch5

[51] C. Still, W.; Tempczyk, A.; C. Hawley, R.; Hendrickson, T. Semianalytical treatment of solvation for molecular mechanics and dynamics. *J. Am. Chem. Soc.* **1990**, *112*, 6127-6129. https://doi.org/10.1021/ja00172a038

[52] Ginex, T.; Vazquez, J.; Gilbert, E.; Herrero, E.; Luque, F. J. Lipophilicity in Drug Design: An Overview of Lipophilicity Descriptors in 3D-QSAR Studies, **2019**, *11* (10), 1177–1193. https://doi.org/10.4155/FMC-2018-0435

[53] Jiang, X. Hydrophobic-lipophilic Interactions. Aggregation and Self-coiling of Organic Molecules. *Acc. Chem. Res.* **1988**, *21 (10)*, 362–367. https://doi.org/10.1021/ar00154a002

[54] Audry, E.; Dubost, J. P.; Colleter, J. C.; Dallet, P. A new approach to structure-activity relations: the molecular lipophilicity potential. *Eur. J. Med. Chem.* **1986**, *21*, 71−72.

[55] Furet, P.; Sele, A.; Cohen, N. C. 3D Molecular Lipophilicity Potential Profiles: A New Tool in Molecular Modeling. *J. Mol. Graph.* **1988**, *6* (4), 182–189. https://doi.org/10.1016/S0263-7855(98)80001-5

[56] Laguerre, M.; Saux, M.; Dubost, J. P.; Carpy, A. MLPP: A Program for the Calculation of Molecular Lipophilicity Potential in Proteins. *Pharm. Sci.* **1997**, *3*, 217–222. https://doi.org/10.1111/j.2042-7158.1997.tb00257.x

[57] Nurisso, A.; Bravo, J.; Carrupt, P. A.; Daina, A. Molecular Docking Using the Molecular Lipophilicity Potential as Hydrophobic Descriptor: Impact on GOLD Docking Performance. *J. Chem. Inf. Model.* **2012**, *52* (5), 1319–1327. https://doi.org/10.1021/ci200515g

[58] Oberhauser, N.; Nurisso, A.; Carrupt, P. A. MLP Tools: A PyMOL Plugin for Using the Molecular Lipophilicity Potential in Computer-Aided Drug Design. *J. Comput. Aided. Mol. Des.* **2014**, *28* (5), 587–596. https://doi.org/10.1007/s10822-014-9744-0

[59] Pyrkov, T. V.; Chugunov, A. O.; Krylov, N. A.; Nolde, D. E.; Efremov, R. G. PLATINUM: A Web Tool for Analysis of Hydrophobic/Hydrophilic Organization of Biomolecular Complexes. *Bioinformatics* **2009**, *25* (9), 1201–1202. https://doi.org/10.1093/bioinformatics/btp111

[60] Pyrkov, T. V.; Kosinsky, Y. A.; Arseniev, A. S.; Priestle, J. P.; Jacoby, E.; Efremov, R. G. Complementarity of Hydrophobic Properties in ATP-Protein Binding: A New Criterion to Rank Docking Solutions. *Proteins Struct. Funct. Genet.* **2007**, *66* (2), 388–398. https://doi.org/10.1002/prot.21122

[61] Harris, C. R.; Millman, K. J.; Van Der Walt, S. J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; Kern, R.; Picus, M.; Hoyer, S.; Van Kerkwijk, M. H.; Brett, M.; Haldane, A.; Del Río, J. F.; Wiebe, M.; Peterson, P.; Gérard-Marchant, P.; Sheppard, K.; Reddy, T.; Weckesser, W.; Abbasi, H.; Gohlke, C.; Oliphant, T. E. Array Programming with NumPy. *Nature* **2020**, *585* (7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

[62] Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; Van Der Walt, S. J.; Brett, M.; Wilson, J.; Millman, K. J.; Mayorov, N.; Nelson, A. R. J.; Jones, E.; Kern, R.; Larson, E.; Carey, C. J.; Polat, İ.; Feng, Y.; Moore, E. W.; Vanderplas, J.; Laxalde, D.; Perktold, J.; Cimrman, R.; Henriksen, I.; Quintero, E. A.; Harris, C. R.; Archibald, A. M.; Ribeiro, A. H.; Pedregosa, F.; Van Mulbregt, P.; Vijaykumar, A.; Bardelli, A. P.; Rothberg, A.; Hilboll, A.; Kloeckner, A.; Scopatz, A.; Lee, A.; Rokem, A.; Woods, C. N.; Fulton, C.; Masson, C.; Häggström, C.; Fitzgerald, C.; Nicholson, D. A.; Hagen, D. R.; Pasechnik, D. V.; Olivetti, E.; Martin, E.; Wieser, E.; Silva, F.; Lenders, F.; Wilhelm, F.; Young, G.; Price, G. A.; Ingold, G.-L.; Allen, G. E.; Lee, G. R.; Audren, H.; Probst, I.; Dietrich, J. P.; Silterra, J.; Webber, J. T.; Slavič, J.; Nothman, J.; Buchner, J.; Kulick, J.; Schönberger, J. L.; De Miranda Cardoso, J. V.; Reimer, J.; Harrington, J.; Rodríguez, J. L. C.; Nunez-Iglesias, J.; Kuczynski, J.; Tritz, K.; Thoma, M.; Newville, M.; Kümmerer, M.; Bolingbroke, M.; Tartre, M.; Pak, M.; Smith, N. J.; Nowaczyk, N.; Shebanov, N.; Pavlyk, O.; Brodtkorb, P. A.; Lee, P.; Mcgibbon, R. T.; Feldbauer, R.; Lewis, S.; Tygier, S.; Sievert, S.; Vigna, S.; Peterson, S.; More, S.; Pudlik, T.; Oshima, T.; Pingel, T. J.; Robitaille, T. P.; Spura, T.; Jones, T. R.; Cera, T.; Leslie, T.; Zito, T.; Krauss, T.; Upadhyay, U.; Halchenko, Y. O.; Vázquez-Baeza, Y. Scipy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **2020**, *17* (3), 261–272. https://doi.org/10.1038/s41592-019-0686-2

[63] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, 12, 2825–2830. https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

[64] RDKit: Open-Source Cheminformatics Software. http://www.rdkit.org (accessed 17th December 2021)

[65] Gowers, R.; Linke, M.; Barnoud, J.; Reddy, T.; Melo, M.; Seyler, S.; Domański, J.; Dotson, D.; Buchoux, S.; Kenney, I.; Beckstein, O. Mdanalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations. In *Proceedings of the Python in Science Conference*; Proceedings of the Python in Science Conference, **2016**, 98–105. https://doi.org/10.25080/majora-629e541a-00e

[66] O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An Open Chemical Toolbox. *J. Cheminformatics* **2011**, *3 (1)*, 33. https://doi.org/10.1186/1758-2946-3-33

[67] Young, T. A.; Gheorghe, R.; Duarte, F. Cgbind: A Python Module and Web App for Automated Metallocage Construction and Host–guest Characterization. *J. Chem. Inf. Model.* **2020**, *60* (7), 3546–3557. https://doi.org/10.1021/acs.jcim.0c00519

[68] Turcani, L.; Berardo, E.; Jelfs, K. E. Stk: A Python Toolkit for Supramolecular Assembly. *J. Comput. Chem.* **2018**, *39* (23), 1931–1942. https://doi.org/10.1002/jcc.25377

[69] Hahsler M.; Piekenbrock M.; Doran D. Dbscan: Fast Density-Based Clustering with R *J. Stat. Soft.* **2019**, *91* (1), 1-30. https://doi.org/10.18637/jss.v091.i01

[70] Su, K.; Wang, W.; Du, S.; Ji, C.; Zhou, M.; Yuan, D. Reticular Chemistry in the Construction of Porous Organic Cages. *J. Am. Chem. Soc.* **2020**, *142* (42), 18060–18072. https://doi.org/10.1021/jacs.0c07367

[71] Lauer, J. C.; Pang, Z.; Janßen, P.; Rominger, F.; Kirschbaum, T.; Elstner, M.; Mastalerz, M. Host-Guest Chemistry of Truncated Tetrahedral Imine Cages with Ammonium Ions. *ChemistryOpen* **2020**, *9* (2), 183–190. https://doi.org/10.1002/open.201900357

[72] Guex, N.; Peitsch, M. C. SWISS-MODEL and the Swiss-Pdb Viewer: An Environment for Comparative Protein Modeling. *Electrophoresis* **1997**, *18* (15), 2714–2723. https://doi.org/10.1002/ELPS.1150181505

[73] Horiuchi, S.; Murase, T.; Fujita, M. Diels-alder Reactions of Inert Aromatic Compounds Within a Self-assembled Coordination Cage. *Chem. Asian J.* **2011**, *6* (7), 1839–1847. https://doi.org/10.1002/asia.201000842

[74] Ghose, A. K.; Viswanadhan, V. N.; Wendoloski, J. J. Prediction of Hydrophobic (Lipophilic) Properties of Small Organic Molecules Using Fragmental Methods: An Analysis of ALOGP and CLOGP Methods. *J. Phys. Chem. A* **1998**, *102* (21), 3762–3772. https://doi.org/10.1021/JP980230O

[75] Gaillard, P.; Carrupff, P.-A.; Testa, B.; Boudon, A. Molecular Lipophilicity Potential, a Tool in 3D QSAR: Method and Applications. *J. Comput. Aided. Mol. Des.* **1994**, *8*, 83–96. https://doi.org/10.1007/bf00119860

[76] Heiden, W.; Moeckel, G.; Brickmann, J. A New Approach to Analysis and Display of Local Lipophilicity/ Hydrophilicity Mapped on Molecular Surfaces. *J. Comput. Aided. Mol. Des.* **1993**, *7*, 503–514. https://doi.org/10.1007/bf00124359

[77] Fauchére, J.-L., Quarendon, P., Kaetterer, L. Estimating and representing hydrophobicity potential. *J. Molecular Graphics*, **1988**, *6*(4), 203–206. https://doi.org/10.1016/s0263-7855(98)80004-0

[78] Mortier, W. J.; Ghosh, S. K.; Shankar, S. Electronegativity-equalization Method for the Calculation of Atomic Charges in Molecules. *J. Am. Chem. Soc.* **1986**, *108* (15), 4315–4320. https://doi.org/10.1021/ja00275a013

[79] Bultinck, P.; Langenaeker, W.; Carbó-Dorca, R.; Tollenaere, J. P. Fast Calculation of Quantum Chemical Molecular Descriptors from the Electronegativity Equalization Method. *J. Chem. Infor. Comp. Sci.* **2003**, *43* (2), 422–428. https://doi.org/10.1021/ci0255883

[80] Geidl, S.; Bouchal, T.; Raček, T.; Svobodová Vařeková, R.; Hejret, V.; Křenek, A.; Abagyan, R.; Koča, J. High-quality and universal empirical atomic charges for chemoinformatics applications. *J. Cheminform.* **2015**, *7* (1), 59. https://doi.org/10.1186/s13321-015-0107-1

[81] O'Connor, H. M.; Tipping, W. J.; Vallejo, J.; Nichol, G. S.; Faulds, K.; Graham, D.; Brechin, E. K.; Lusby, P. J. Utilizing Raman Spectroscopy as a Tool for Solid- and Solution-phase Analysis of Metalloorganic Cage Host–guest Complexes. *Inorg. Chem.* **2023**, *62* (5), 1827–1832. https://doi.org/10.1021/acs.inorgchem.2c00873

[82] Chen, Y.; Wu, G.; Chen, B.; Qu, H.; Jiao, T.; Li, Y.; Ge, C.; Zhang, C.; Liang, L.; Zeng, X.; Cao, X.; Wang, Q.; Li, H. Self-assembly of a Purely Covalent Cage with Homochirality by Imine Formation in Water. *Angew. Chem. Int. Ed.* **2021**, *60* (34), 18815–18820. https://doi.org/10.1002/anie.202106428