JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim

Article

CageCavityCalc (C3): A Computational Tool for Calculating and Visualizing Cavities in Molecular Cages

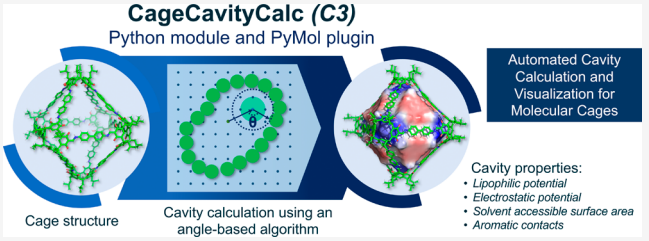
Vicente Martí-Centelles,* Tomasz K. Piskorz, and Fernanda Duarte*

Cite This: J. Chem. Inf. Model. 2024, 64, 5604−5616

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

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 remains 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 ChimeraX. To enhance user accessibility, a PyMol plugin has been created, allowing nonspecialists 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.



1. INTRODUCTION

Discrete three-dimensional (3D) organic and metal−organic (porous) cages have emerged as promising biomimetic systems.[1−4] They offer synthetic modularity and tunability, enabling chemists to efficiently create structures with customized sizes and shapes from simple building blocks.[5−8] 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.[9−12]

In molecular cages, the affinity of the host toward a particular guest depends on various structural and electronic parameters.[13] 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.[14] Since then, this rule has been extended to other supramolecular structures with varying success[15−17] and also to enzymes.[18]

The most commonly used algorithms to calculate cavities are geometric (see description of geometric algorithms below), as they only require the Cartesian coordinates of the molecule and are therefore easy to implement.[19−21] Based on geometric methods, several software tools have been developed for detecting, analyzing, and visualizing cavities, as analyzed in comprehensive reviews.[19−21] These tools are primarily used to identify protein-binding pockets but have also been used to calculate the cavities of supramolecular cages.[21] The following selected list shows examples of cavity calculation software for different types of hollow structures, such as proteins, cages, representative fragments of metal−organic frameworks, etc.: VOIDOO,[22] McVol,[23] Volarea,[24] CAVER,[25] KVFinder (including pyKVFinder, parKVFinder, and KVFinder-web),[26−29] Fpocket (including its molecular dynamics (MDs) implementation MDpocket and FpocketWeb),[30−32] PoreBlazer,[33] Zeo++,[34] CavVis,[35] POVME,[36] ghecom,[37] PyWindow,[38] MoloVol,[39] CAST,[40] ProteinVolume,[41] 3 V,[42] and Voronoia.[43] Additionally, other software that is no longer available includes GRASP[44] and HOLLOW.[45] It is worth noting that VOIDOO, which is probably the most popular tool used by the supramolecular community,[46−48] has not been updated since 1999, making it challenging to use on modern computers. For a comprehensive list of available tools, we refer the reader to comprehensive reviews that analyze cavity calculation software.[19−21]

The geometric algorithms for detecting cavities are classified into four categories based on the geometrical techniques: grid,

Received: March 1, 2024
Revised: June 5, 2024
Accepted: June 26, 2024
Published: July 9, 2024

https://doi.org/10.1021/acs.jcim.4c00355
J. Chem. Inf. Model. 2024, 64, 5604−5616

sphere, surface, and tessellation.[19,20] Each approach has its own strengths and weaknesses, as detailed by Simões et al.[20] and Krone et al.[19] Grid-based algorithms (e.g., POVME) map the atomic 3D coordinates of a molecule onto a grid and use clustering methods to identify the empty grid points that define the cavity. Sphere-based algorithms (e.g., PyWindow) use the atomic 3D coordinates and van der Waals radii of the molecule to define the molecular surface, then, the cavity is detected by scanning the surface with a probe, usually a hard sphere. Tessellation-based algorithms (*e.g.*, Fpocket and CAVER) employ surface representations, such as Voronoi diagrams, to divide the molecular space into regions, each with specific cavity detection methods. Surface-based algorithms define cavities based on the solvent-excluded surface, solvent-accessible surface (SAS), or ligand-accessible surface. For detailed information on these methods, we refer interested readers to comprehensive literature reviews.[19−21]

Major challenges in detecting cavities using geometric algorithms include defining the cavity boundary, known as mouth opening ambiguity (MOA);[19−21] grid-spacing sensitivity (GSS), which evaluates the impact of the grid voxel size; and protein-orientation sensitivity (POS), which assesses the effect of structure's orientation within the grid.[19−21] Simões et al.[20] describe that grid-based methods can present GSS and POS but have no difficulties in finding cavity mouth openings. Conversely, sphere-based methods can suffer from user-assisted cavity localization (UACL), meaning the method is not fully automated and input from the user is needed, MOA, and POS. These individual techniques can be combined to mitigate or even eliminate these challenges (e.g., the KVFinder project and ghecom). For instance, GSS is solved by setting the voxel size at most half of the radius of the water probe sphere, while MOA and POS are avoided by using large probe spheres to block cavity openings. Additionally, these combined methods do not suffer from UACL as cavity detection is automated.

In this work, we focus on the geometric rolling probe method that relies on a combination of grid- and sphere-based algorithms. This algorithm uses one or two probes and achieves varying degrees of success in detecting cavity mouth openings (i.e., cavity windows). 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).[19−21] 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. Therefore, the user needs to carefully evaluate, and often fine-tune, the parameters. This process is laborious and difficult to generalize across different systems or when analyzing 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. Therefore, efficient detection of cavity mouth openings remains a challenge for cages with large windows.

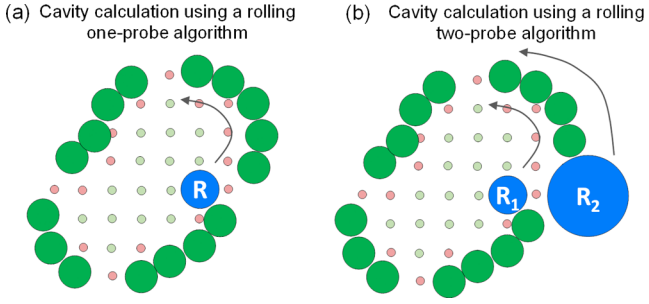


Figure 1. Schematic representation of geometric sphere- and grid-based methods of cavity calculation algorithms using rolling probes (R, denoted by blue spheres): (a) Rolling one-probe algorithm: The probe moves within the cavity, identifying accessible areas as cavity (green points), and if the probe collides with cage atoms, these grid points are designated as noncavity (red). 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.

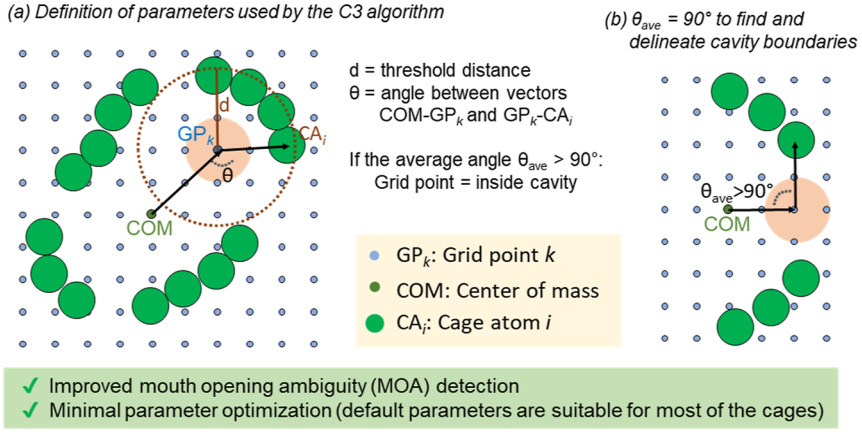


Figure 2. *C*3 angle-based algorithm developed herein: it uses the cavity's angle θ between the two vectors COM-GP_k and GP_k-CA_i. The grid point size is automatically determined by the grid resolution.

https://doi.org/10.1021/acs.jcim.4c00355
J. Chem. Inf. Model. 2024, 64, 5604−5616

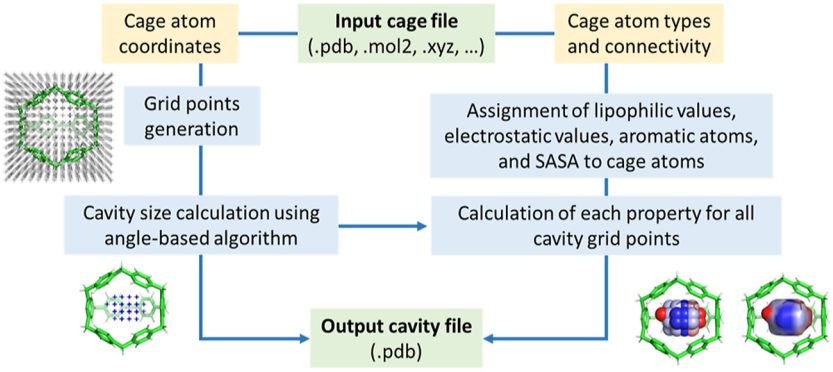


Figure 3. *C3* workflow to determine the cage cavity and the properties of each cavity grid point.

To address the challenge of detecting cavity mouth openings (i.e., cavity windows), taking the concept of the rolling probe algorithm, we developed a sphere- and grid-based method that uses a geometric algorithm based on a novel angle measurement technique to determine the cavity boundaries (Figure 2). 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, the algorithm calculates the average of all angles θ for each cage atom within a certain threshold distance, which is automatically calculated as the diameter of the maximum escape sphere from the cavity. If the average angle is greater than 90°, the probe is considered to be inside the cage; otherwise, it is considered to be outside of the cavity. This algorithm is specifically designed to calculate the internal cavity volume of molecular cages. The cavity is calculated in an automated manner and therefore not affected by UACL. In this work, we test the efficiency of the developed *C3* angle-based algorithm and its performance on MOA, GSS, and POS.

In addition to host−guest size complementarity, guest binding is significantly influenced by electrostatic,[49,50] hydrogen bonding,[51] and van der Waals interactions[49,52] with the host, which can be significantly affected by the solvent.[53] For example, in water, the hydrophobic effect plays a key role in driving guest binding.[54−56] Ward and co-workers demonstrated that hydrophobic guests exhibit 2−3 orders of magnitude higher association constants compared to polar guests.[57,58] Similarly, Ballester and co-workers established a linear relationship between binding free energies and the surface area of the nonpolar guests' fragments, indicating a hydrophobic effect of 33−38 cal mol^{-1} Å^{-2}.[59]

Electrostatic complementarity is often visualized through electrostatic potential (ESP) surfaces.[60,61] 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.[62,63] Additionally, hydrophobic interactions can be indirectly estimated using methods based on hydrophobic−lipophilic interactions, which quantify the tendency of nonpolar molecules to avoid contact with polar solvents, inducing aggregation and self-coiling.[64,65] Another parameter, molecular hydrophobicity potential (MHP), derived from the 1-octanol/water partition coefficient (logP),[66,67] has proven useful in describing hydrophobicity in proteins,[68] analyzing protein pockets in docking engines, and predicting protein−ligand binding affinities.[69−72] 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, in a python script, or as a plugin to the molecular visualization program PyMol.[73] 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 state-of-the-art methods when assessing 16 cavities with different topological and morphological features. Furthermore, we employ *C3* to compute the MHP and ESP surfaces in the calculated cavities, aiding in 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. *C3* can be installed using the pip package manager by running the command "pip install CageCavityCalc". The calculations in this article have been performed using *C3* version 1.0.5. Detailed installation instructions are provided in Supporting Information §3. *C3* is compatible with Windows, macOS, and Linux, requiring Python 3.7 or later. It employs scientific programming libraries, including NumPy (v1.26.4),[74] SciPy (v1.13.0),[75] and scikit-learn (v1.4.2).[76] Optional functionalities can be enabled using the chemical programming libraries RDKit (2023.09.5),[77] MDAnalysis (v2.7.0),[78] OpenBabel (v3.1.0),[79] and Cgbind (v1.0.3),[80] which facilitate the handling of chemical structures, including the identification of functional groups and the calculation of molecular properties, or the generation of cage structures from ligands and metals (i.e., using *Cgbind*, as shown in Supporting Information section S1).

The core of the code is based on the *Cavity* class, which features several functions: file reading, COM 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 Supporting Information §S1 for a full

https://doi.org/10.1021/acs.jcim.4c00355
J. Chem. Inf. Model. 2024, 64, 5604−5616

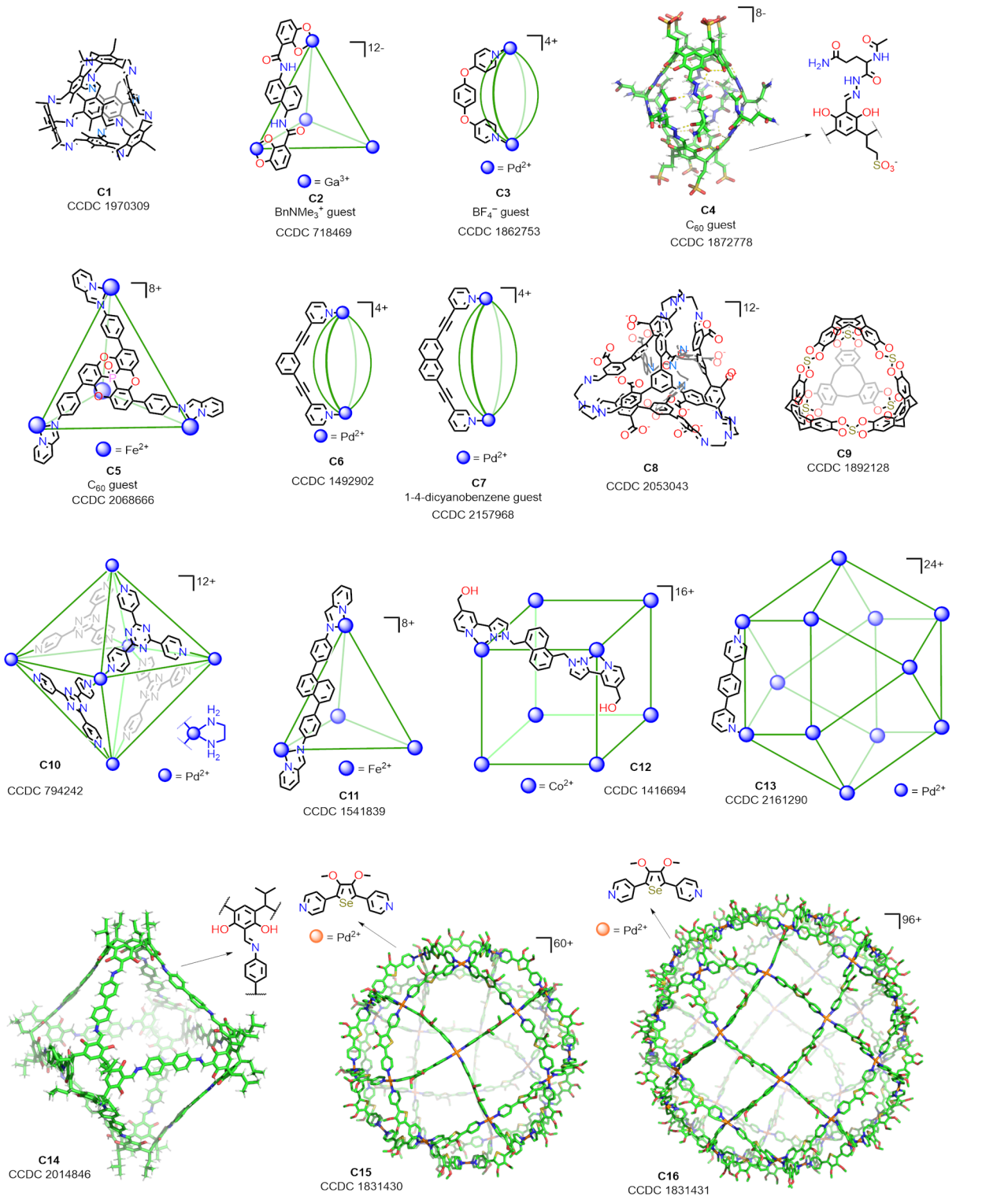


Figure 4. Data set used to test the efficiency of *C*3. The structures for **C1**−**C16** cages were extracted from the corresponding CIF files obtained from the CCDC.

description). Additionally, it offers a PyMol plugin (PyMol 3.0.0)[73] with the corresponding *C*3 graphical user interface (GUI) to set up all calculation parameters of *C*3 (see details in Supporting Information §S2). Figure 3 illustrates a simplified schematic diagram of *C*3 functionality.

Upon loading the 3D cage structure into *C*3, a 3D grid filled with grid points is generated, with customizable size and grid spacing (default parameters are described in Supporting Information §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 are computed using the molecular properties of each cage atom. These properties include aromatic contacts, SAS area (SASA), hydrophobicity (MHP), and/or ESP. The calculated values can be saved as B-factor in the generated cavity PDB file, facilitating

https://doi.org/10.1021/acs.jcim.4c00355
J. Chem. Inf. Model. 2024, 64, 5604−5616

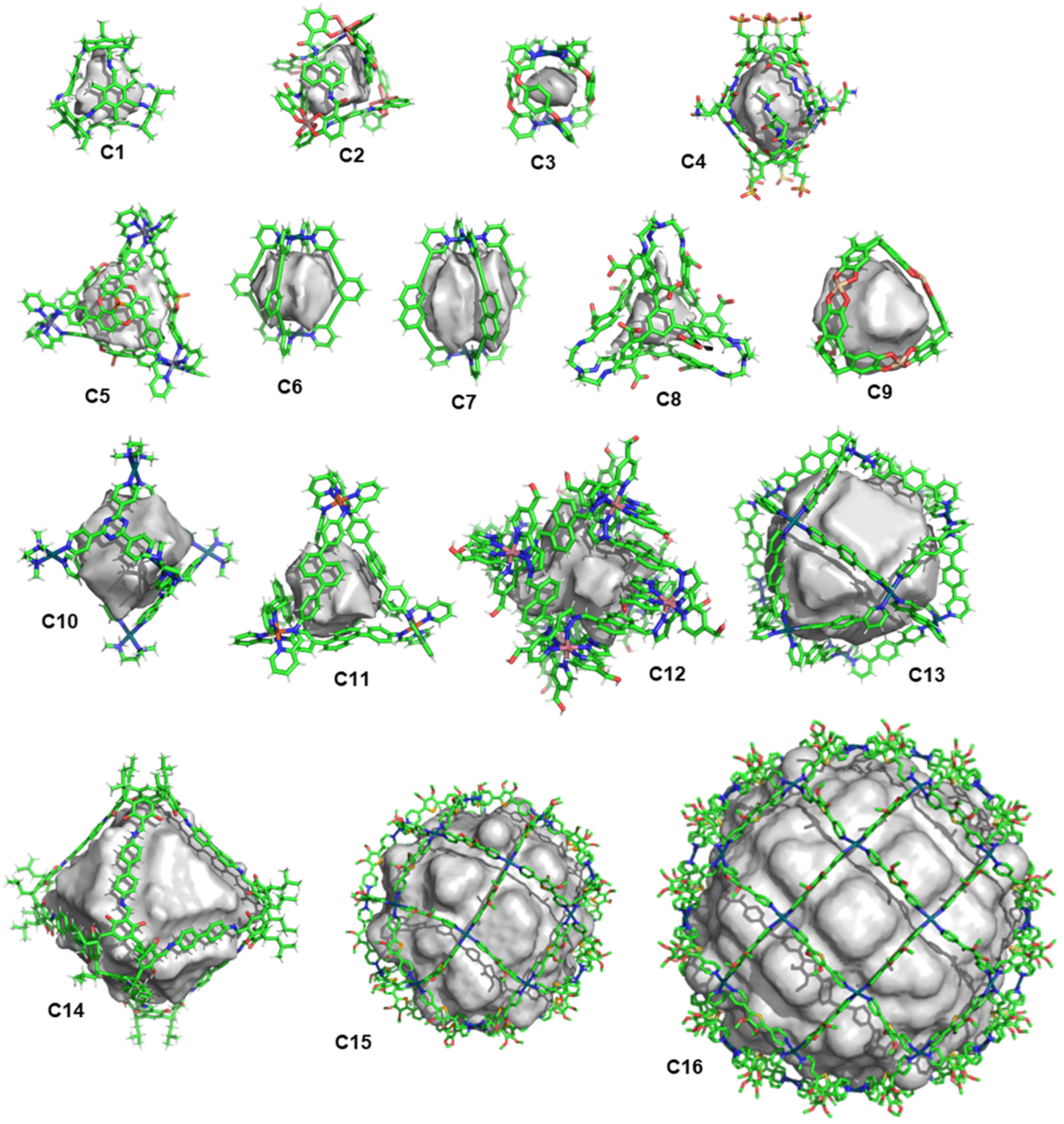


Figure 5. Computed cavities with *C*3 for molecular cages **C1**−**C16**.

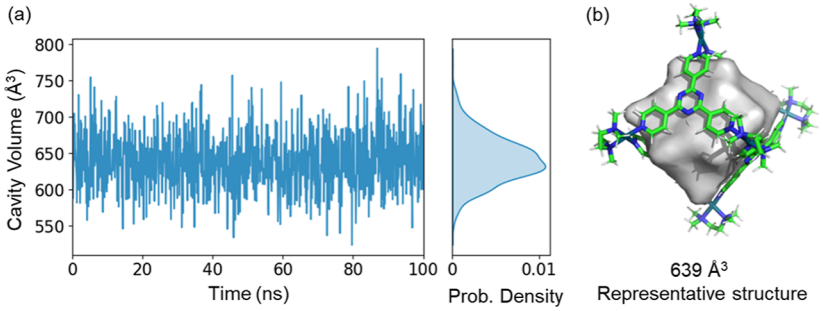
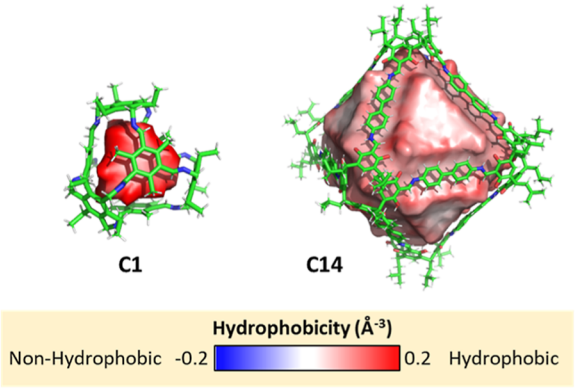


Figure 6. Analysis of the volume change during a 100 ns MDs trajectory of cage **C10** using *C*3 (cavity calculation parameters can be found in Supporting Information §S4). (a) Evolution of the volume over time and the corresponding probability distribution of cavity volumes. (b) Representative structure having a volume close to the mean cavity volume.

their visualization in any standard molecular visualization software; one property is saved per PDB file. Alternatively, users can calculate multiple properties using the PyMol plugin, which enables storing all computed properties in the same

https://doi.org/10.1021/acs.jcim.4c00355
J. Chem. Inf. Model. 2024, 64, 5604−5616



Cage	Cavity Volume (Å^3)	Average Cavity Hydrophobicity (Å^{-3})	Total cavity Hydrophobicity	Hydrophobic index (HI)
C1	209	0.123	26	1.00
C14	7784	0.025	195	1.00

Figure 7. Calculation of the hydrophobicity of cages **C1** and **C14** using the Ghose method and Fauchere distance function. Cavity calculation parameters are described in Table S2.

PyMol session file. This feature enables users to interactively select the properties they want to visualize and save a PDB file for each of them. The following sections detail each of these steps.

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[81] or Cgbind.[82] *C3* supports various molecular file formats such as .xyz, .pdb, .mol, .mol2, and others handled by MDAnalysis.[78] 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, allowing fast running times for small- and medium-sized cages (in this work, we categorize cages with approximately 300−400 non-H atoms, or 25−30 Å cage diameter as medium-sized). Users can adjust grid spacing based on computational resources. Typically, a finer grid (smaller spacing, typically 0.5 Å) is used for smaller cages, while a coarser grid (larger spacing, typically from 2 to 4 Å) is used for larger cages. The calculation of the cavity volume involves iterating over grid points and classifying them as either

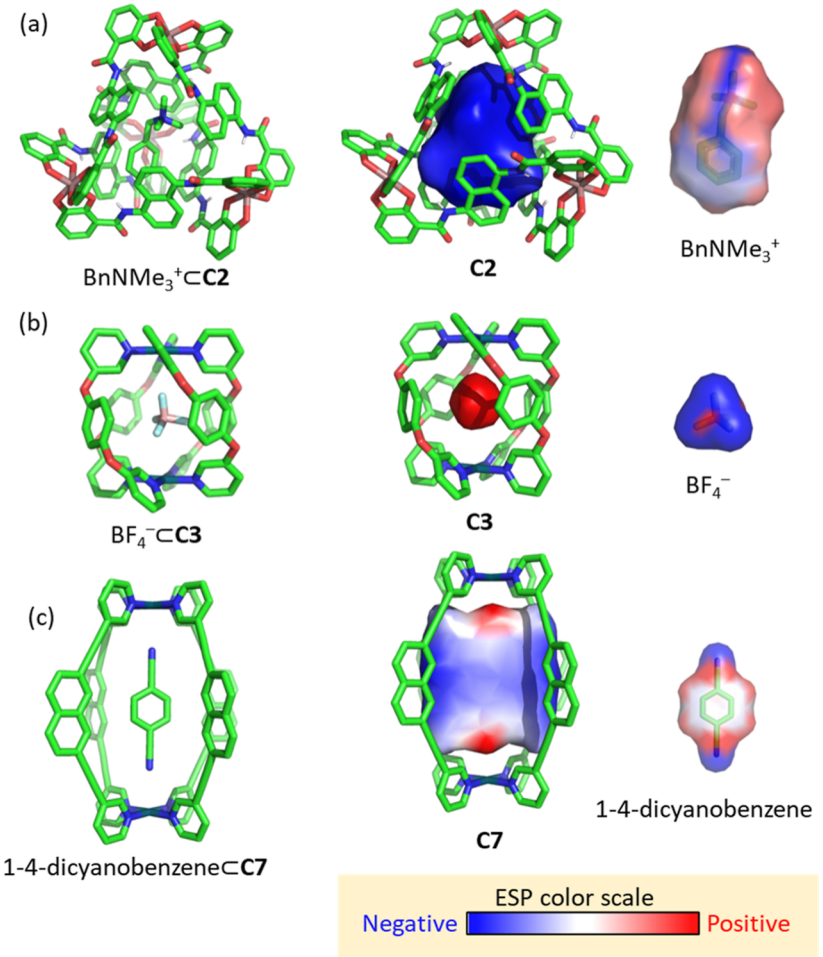


Figure 8. Crystal structure of host−guest complexes, ESP mapped on the cavity of empty cages, and partial charge mapped on the van der Waals sphere of guests were calculated using the EEM implemented in OpenBabel. (a) Cage **C8** with the BnNMe$_3^+$ guest. (b) Cage **C3** with the BF$_4^-$ guest. (c) Cage **C7** with the 1,4-dicyanobenzene guest.

https://doi.org/10.1021/acs.jcim.4c00355
J. Chem. Inf. Model. 2024, 64, 5604−5616

Table 1. Calculated Cavity Volumes with *C3* and State of the Art Cavity Calculation Software (Å^3)

cage	CCDC	median	C3	KVFinder project	Fpocket	MoloVol	CAVER	Ghecom	pywindow	POVME	VOIDOO	refs
C1	1970309	144	209	217	144	186	223	127	89	69	64	85
C2	718469	269	295	269[a]	279[a]	289[a]	339[a]	192[a]	90[a]	108[a]	101	86
C3	1862753	44	63	78[a]	84[a]	44[a]	65[a]	29[a]	37[a]	11[a]	14	87
C4	1872778	704	726	731[a]	805[a]	747[a]	682[a]	704[a]	517[a]	409[a]	396	88
C5	2068666	627	699	737[a]	627[a]	650[a]	742[a]	606[a]	532[a]	319[a]	447	89
C6	1492902	254	308	114	[b]	434	392	305	134	202	15	90
C7	2157968	403	496	384	[b]	619	448	422	174	355	26	91
C8	2053043	100	130	104	75	105	135	100	77	31	35	92
C9	1892128	617	672	558[a]	617[a]	648[a]	659[a]	669[a]	412[a]	373[a]	38	93
C10	794242	508	667	497	652	638	508	656	293	412	95	94
C11	1541839	461	458	488[a]	495[a]	526[a]	543[a]	461[a]	314[a]	211[a]	91	95
C12	1416694	613	613	866[a]	542[a]	984[a]	1008[a]	936[a]	263[a]	275[a]	117	96
C13	2161290	5240	5240	6849	4190	8641	8334	7933	4252	5021	867	97
C14	2014846	8073	7784	8336	8073	8997	8420	9051	5740	7973	1478	98
C15	1831430	37,728	31,327	37,728[a]	39,077[a]	62,843[a]	22,941[a]	59,147[a]	30,568[a]	45,704[a]	14,310	99
C16	1831431	62,568	65,722	52,474	47,480	121,832	10,687	49,342	62,568	86,464	92,290	99
MRAE (%)			16	20	27	32	33	21	33	38	70	

[a]Values from ref 21. [b]No internal cavity was obtained.

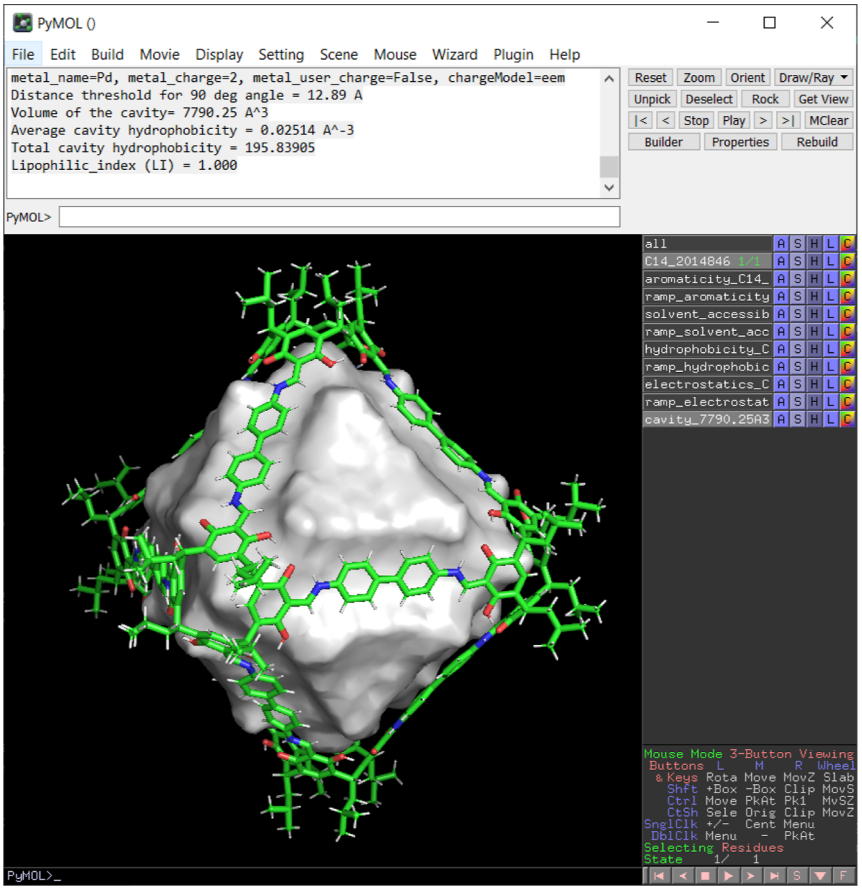


Figure 9. Screenshot of the results after running the *C3* PyMol plugin.

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 two times the window size, which is obtained from the diameter of the maximum escape sphere from the cavity,[82] 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 COM of the cage 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 2a). 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

https://doi.org/10.1021/acs.jcim.4c00355
J. Chem. Inf. Model. 2024, 64, 5604−5616

angle $\theta_{average}$ is larger than 90°, 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.[75] 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.[83] 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 COM.

To demonstrate the utility of the *C3* algorithm, we created a data set of 16 cages (**C1**−**C16**, Figure 4) extracted from the Cambridge Structural Database.[84] The data set includes cages with different topological and morphological features, including cages with well-defined closed cavities or medium-sized windows (**C1**−**C12**), cages with inclusion complexes (**C2**−**C5**), and cages with large windows (**C13**−**C16**).

When computing cavity volumes, the grid spacing and distance threshold for the 90-degree calculation must be carefully chosen to balance computational cost and accuracy. The grid spacing was set to 0.5 Å for cages **C1**−**C12** and 1.0−3.5 Å for cages **C13**−**C16**. As expected for a geometric algorithm, *C3* is sensitive to grid spacing. For example, increasing grid spacing from 0.5 to 1 Å in cage **C1** led to smaller computed volumes because the coarser grid (i.e., larger grid spacing) cannot capture the finer cavity details. To evaluate the grid-spacing effect on the computed volume, we computed the cavity volume of cages **C1**−**C16** using different grid spacings (see Figure S2 in Supporting Information §S6), observing that cavity volume is more sensitive to grid spacing for smaller cavities (e.g., **C1** and **C5**). We observed that reducing the grid spacing results in a convergence of the computed cavity volume that depends on the cavity size, with larger cavities achieving convergence more easily (see Figure S2 in Supporting Information §S6).

We tested MOA using cages with small, medium, and large windows. We observed that *C3* performs well in automatically identifying and delineating cavity boundaries (see Supporting Information §S7). The default distance threshold for the 90-degree calculation, set at 2.0 times the window size, works well for cages with completely closed cavities or medium-sized windows (**C1**−**5**, **C8**, and **C11**) and cages with significant openings (**C6**, **C7**, **C9**, **C10**, and **C12**). However, for cages with larger openings (**C13**−**16**), it is necessary to increase the threshold to 3.0−4.5 to obtain satisfactory results and prevent the probe from "escaping" the cavity. In such cases, users need to visually inspect the computed cavity and adjust the distance threshold for the 90-degree calculation as needed. If the probe is escaping the cavity through the window, increasing the distance threshold is recommended. Conversely, if the calculated cavity looks spherical with an underestimated volume, the distance threshold needs to be reduced.

Finally, we tested the POS of *C3* by randomly rotating the XYZ coordinates of the cages. For this, we generated five structures with random rotations for each cage **C1**−**C16**. We calculated the volume of each rotated cage structure using the same parameters and determined the mean volume and standard error (Table S3 in Supporting Information §S5). The computed mean volumes have a relative error of 0.9% relative to the mean volume, ranging from 0.2% for **C4** and **C11** to 2.8% for **C10**. Overall, comparing *C3*'s performance on POS (Table S3), GSS (Figure S2), and MOA (Figure S3), we observed that cavity volume is more affected by grid spacing (GSS) than MOA and POS. This highlights the importance of carefully setting up grid spacing by the user.

The cavity volume results obtained using *C3* for cages **C1**−**16** are depicted in Figure 5 and Table 1, with the parameters described in Table S2 in Supporting Information §S4. Initially, we computed cavity volumes of cages **C1**−**C16** using the default parameters of grid spacing 1 Å (except for cages **C15** and **C16**, which, due to hardware limitations, required larger grid spacing of 3 and 3.5 Å, respectively) and a distance threshold for the 90-degree angle of twice the window size. We then refined the computed volumes by optimizing the parameters. This included setting grid spacing to 0.5 Å for cages **C1**−**C12** and adjusting the distance threshold for the 90-degree calculation (in window size units) to 3 for cages **C13** and **C14** and 4.5 for cages **C15** and **C16**. Comparing the cavity volumes computed with the optimized and default parameters shows a mean relative absolute error (MRAE) of 26.7%, indicating that the default parameters provide reliable cavity volumes.

We also compared these results to those obtained using eight different cavity characterization software (KVFinder, Fpocket, MoloVol, CAVER, ghecom, pywindow, and POVME) recently studied by Lopes-de-Oliveira, György Szalóki, and co-workers,[21] and VOIDOO, widely used by the supramolecular community. The results are summarized in Table 1, and further details regarding cavity calculation parameters can be found in Supporting Information §S7. As this data set contains cages without guests and open-cavity cages with guests, for these cages, it is not possible to obtain the cavity volume estimates using Rebek's 55% rule. Therefore, to assess the relative performance between software, we obtained the cavity median volume for each **C1**−**C16** cage from the computed volumes with all tested software. The median was chosen instead of the mean as a better representation of the central tendency due to the non-normal distribution and the presence of outliers in the data. Subsequently, we calculated the MRAE relative to the median for each software and cage, i.e., $MRAE_{Cage}$ = the absolute value of 100 × (volume − median volume)/median volume. MRAE essentially measures the discrepancy between a software's calculated cavity volume and the median volume, expressed as a percentage. The obtained MRAE values were in accordance with literature data,[21] showing large outliers ($MRAE_{Cage}$ > 80%) for most of the cases for the computed volumes with the VOIDOO software, while *C3* yielded the lowest MRAE (16%), with a maximum $MRAE_{Cage}$ of 45% obtained for cage **C1**. Remarkably, it highlights the the accuracy of its cavity calculation algorithm.

To validate the accuracy of *C3*'s cavity detection algorithm, we compared the performance of *C3* against cavity volumes obtained from Rebek's 55% rule (i.e., cavity volume = guest volume/0.55) for the inclusion complexes of Benchmark Data set 1 reported by Guerra et al.[21] This data set was specifically designed with cages with well-defined and closed cavities to evaluate the performance of cavity calculation software. The cavity size values obtained from Rebek's rule show a good correlation with the *C3* computed cavity volume, with an MRAE

https://doi.org/10.1021/acs.jcim.4c00355
J. Chem. Inf. Model. 2024, 64, 5604−5616

of 22.6% (see Table S11 in Supporting Information §S9). This shows that *C*3 can properly assess volume estimation with performance close to that obtained with the KVFinder project (MRAE = 16.1%) and Fpocket (MRAE = 16.9%).[21] It is important to note that this rule is valid for cages with a rigid and closed cavity, and therefore deviations are expected with flexible open cages that may still allow a guest to bind even if cavity occupancy by the guest deviates significantly form the 55% rule.[3,13]

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 cage **C10**, which provides a robust and well-defined cavity with a volume variation over time of *ca.* 10% (V_{cavity} = 638 ± 38 Å^3, see Figure 6). These results highlight the robustness of the angle-base algorithm of *C*3 for analyzing MD trajectories.

We showed that the *C*3 algorithm is efficient in calculating cavity volumes on cages with all types of shapes, cavity sizes, and window sizes. The angle-based algorithm of *C*3 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, SASA, hydrophobicity (MHP), and/or ESP, as 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 were tabulated by Ghose and co-workers for various atom types based on their contributions to the 1-octanol/water partition coefficient.[100]

Subsequently, the 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 $f(d_{ik})$

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

The commonly used distance functions are:

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

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

$$f(d_{ik}) = e^{-d_{ik}/2} \text{ modified Fauchère's distance function} \tag{4}$$

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 (eq 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 (eq 3). Both Audry's and Fauchère's distance functions do not appear to be adequate beyond the SAS, and the slightly modified Fauchère's distance function (eq 4) overcomes this limitation.[101−103]

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 eq 5.[69,70]

$$\mathrm{HI} = \frac{\mathrm{MHP}^+}{\mathrm{MHP}^+ + |\mathrm{MHP}^-|} \tag{5}$$

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 7).

To illustrate the use of *C*3 for calculating cavity properties, we computed the MHP for cages **C1** and **C14**. The average cavity hydrophobicity, 0.025 for **C14** and 0.123, for **C1**, indicates that the cavity of **C1** is *ca.* 4.9 times more hydrophobic than **C14** (Figure 7). 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.

2.2.4. ESP of the Cavity. The 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 *C*3 by mapping the partial charge contributions from the cage atoms onto a grid representing the cavity (eq 6)

$$V_{\mathrm{grid}} = k \sum_{i=0}^{N_{\mathrm{cage}}} \frac{q_i}{d_{(\mathrm{grid},i)}} \tag{6}$$

where *k* is the Coulomb constant, q_i is a partial charge of a cage atom, and $d_{(\mathrm{grid},i)}$ 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,[79] a widely accepted and efficient procedure for deriving charge-based descriptors in QSAR studies.[104−106]

The cavity ESP is a valuable tool for identifying favorable interactions between hosts and polar guests. Cationic and anionic cages often encapsulate their respective counterions, as illustrated by complementary electric potentials of cation and ions for the **C2** cage and $BnNMe_3^+$ complex (Figure 8a) and the **C3** cage and BF_4^- complex (Figure 8b). Such analysis is also useful for neutral guests, as exemplified by the computed ESP for cage **C7** and 1,4-dicyanobenzene as potential guests. The complementarity of their ESPs suggests the formation of a strongly bound host−guest complex (Figure 8c).[91]

2.3. PyMol Plugin. In addition to the Python module and the command-line interface, we have developed a 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 *C*3 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 cage **C14**, providing in the PyMol session the

https://doi.org/10.1021/acs.jcim.4c00355
J. Chem. Inf. Model. 2024, 64, 5604−5616

output obtained for the cavity, with an individual output for each selected property (Figure 9).

3. CONCLUSIONS

We have introduced an automated cavity calculation software, *C*3, deployed as a Python module, for calculating cage cavities, as well as aromatic contacts, SASA, hydrophobicity (MHP), and/or ESP. The angle-based method that we developed offers a practical, easy-to-implement, and computationally efficient method, eliminating the need for parameter adjustments in most cases. However, results can be improved by fine-tuning the distance threshold for the 90-degree grid spacing. This approach facilitates automated analyses across cages of different sizes, shapes, and window dimensions. Users can vary the grid spacing to achieve the desired cavity resolution and optimize calculation time. Typically, smaller grid spacing is suitable for small cages, while larger spacing is preferable for large cages.

An important feature of our method is the improved performance in cages with large windows, providing improved MOA detection. The method was benchmarked on a wide range of cage structures with different topological and morphological features, including cages with well-defined closed cavities, inclusion complexes, and cages with large windows. The main advantages of our method are its easy use and general applicability to a wide range of porous structures.

The cavity can be visualized using 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 nonspecialized users, a plugin for the 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 Availability Statement

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

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.4c00355.

Description of the arguments that can be used in the *C*3 Python module through the command line and in Python scripts, description of the *C*3 PyMol plugin GUI use, installation instructions, parameters, and details of the calculated cavities (PDF)

*C*3 source code v1.0.5 (ZIP)

Tutorial of the PyMol plugin (MP4)

*C*3 computed cavity volume for the MD trajectory of cage **C10** (MP4)

■ AUTHOR INFORMATION

Corresponding Authors

Vicente Martí-Centelles − *Instituto Interuniversitario de Investigación de Reconocimiento Molecular y Desarrollo Tecnológico (IDM), Universitat Politècnica de València, Universitat de València, Camino de Vera s/n, Valencia 46022, Spain; CIBER de Bioingeniería Biomateriales y Nanomedicina, Instituto de Salud Carlos III, Madrid 28029, Spain; Departamento de Química, Universitat Politècnica de València, Camino de Vera s/n, Valencia 46022, Spain;* orcid.org/0000-0002-9142-9392; Email: vimarce1@upv.es

Fernanda Duarte − *Chemistry Research Laboratory, University of Oxford, Oxford OX1 3TA, U.K.;* orcid.org/0000-0002-6062-8209; Email: fernanda.duartegonzalez@chem.ox.ac.uk

Author

Tomasz K. Piskorz − *Chemistry Research Laboratory, University of Oxford, Oxford OX1 3TA, U.K.;* orcid.org/0000-0003-0716-6874

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.4c00355

Author Contributions

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

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

V.M.-C acknowledges financial support from Generalitat Valenciana (project CIDEGENT/2020/031), MICIU/AEI/10.13039/501100011033 (project PID2020-113256RA-I00), and MICIU/AEI/10.13039/501100011033 and European Union NextGenerationEU/PRTR (project CNS2023-144879). T.K.P. and F.D. acknowledge the financial support from EPSRC (EP/W010666/1 and EP/W009803/1). This research was supported by CIBER (CB06/01/2012), Instituto de Salud Carlos III, Ministerio de Ciencia e Innovación. Funding for open access charge: CRUE-Universitat Politècnica de València.

■ 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.

(2) Raee, E.; Yang, Y.; Liu, T. Supramolecular structures based on metal-organic cages. *Giant* **2021**, *5*, 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*, 11300−11331.

(4) Montà-González, G.; Ortiz-Gómez, E.; López-Lima, R.; Fiorini, G.; Martínez-Máñez, R.; Martí-Centelles, V. Water-Soluble Molecular Cages for Biological Applications. *Molecules* **2024**, *29*, 1621.

(5) Martí-Centelles, V. Kinetic and thermodynamic concepts as synthetic tools in supramolecular chemistry for preparing macrocycles and molecular cages. *Tetrahedron Lett.* **2022**, *93*, 153676.

(6) Lewis, J. E. Molecular engineering of confined space in metal-organic cages. *Chem. Commun.* **2022**, *58*, 13873−13886.

(7) Mcconnell, A. J. Metallosupramolecular cages: from design principles and characterisation techniques to applications. *Chem. Soc. Rev.* **2022**, *51*, 2957−2971.

(8) 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.

(9) Zhang, D.; Lee, H.-S.; Kim, Y.-M.; Kim, J.; Lee, C.-H. Metal-Organic Cages for Molecular Separations. *Nat. Rev. Chem* **2021**, *5*, 168−182.

(10) Ham, R.; Nielsen, C. J.; Pullen, S.; Reek, J. N. H. Supramolecular Coordination Cages for Artificial Photosynthesis and Synthetic Photocatalysis. *Chem. Rev.* **2023**, *123*, 5225−5261.

(11) 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.
(12) 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.
(13) 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.
(14) Mecozzi, S.; Rebek, J., Jr The 55% Solution: A Formula for Molecular Recognition in the Liquid State. *Chem.—Eur. J.* **1998**, *4*, 1016−1022.
(15) 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*, 8475−8483.
(16) 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*, 638−641.
(17) 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*, 756−760.
(18) 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*, 237−240.
(19) 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.
(20) Simões, T.; Lopes, D.; Dias, S.; Fernandes, F.; Pereira, J.; Jorge, J.; Bajaj, C.; Gomes, A. Geometric Detection Algorithms for Cavities on Protein Surfaces in Molecular Graphics: A Survey. *Comput. Graph. Forum* **2017**, *36*, 643−683.
(21) 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.
(22) Kleywegt, G. J.; Jones, T. A. Detection, Delineation, Measurement and Display of Cavities in Macromolecular Structures. *Acta Crystallogr. Sect. D Biol. Crystallogr.* **1994**, *50*, 178−185.
(23) Till, M. S.; Ullmann, G. M. McVol - A Program for Calculating Protein Volumes and Identifying Cavities by a Monte Carlo Algorithm. *J. Mol. Model.* **2010**, *16*, 419−429.
(24) 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*, 743−755.
(25) 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 Bioinf.* **2006**, *7*, 1−9.
(26) 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 Bioinf.* **2021**, *22*, 607−620.
(27) 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 Bioinf.* **2014**, *15*, 197−205.
(28) da Silva Guerra, J. V.; Ribeiro Filho, H. V.; Bortot, L. O.; Honorato, R. V.; Pereira, J. G. de C.; de Oliveira, P. S. L. ParKVFinder: A thread-level parallel approach in biomolecular cavity detection. *Softw. X* **2020**, *12*, 100606.
(29) Guerra, J. V. S.; Ribeiro-Filho, H. V.; Pereira, J. G. C.; Lopes-de-Oliveira, P. S. KVFinder-web: a web-based application for detecting and characterizing biomolecular cavities. *Nucleic Acids Res.* **2023**, *51*, W289−W297.
(30) Le Guilloux, V.; Schmidtke, P.; Tuffery, P. Fpocket: An Open Source Platform for Ligand Pocket Detection. *BMC Bioinf.* **2009**, *10*, 168.
(31) Schmidtke, P.; Bidon-chanal, A.; Luque, F. J.; Barril, X. MDpocket: Open-Source Cavity Detection and Characterization on Molecular Dynamics Trajectories. *Bioinformatics* **2011**, *27*, 3276−3285.
(32) Kochnev, Y.; Durrant, J. D. FPocketWeb: Protein Pocket Hunting in a Web Browser. *J. Cheminf.* **2022**, *14*, 58.
(33) 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*, 9849−9867.
(34) 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*, 134−141.
(35) Simões, T. M. C.; Gomes, A. J. P. CavVis—A Field-of-View Geometric Algorithm for Protein Cavity Detection. *J. Chem. Inf. Model.* **2019**, *59*, 786−796.
(36) 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.
(37) Kawabata, T. Detection of multiscale pockets on protein surfaces using mathematical morphology. *Proteins: Struct., Funct., Bioinf.* **2010**, *78*, 1195−1211.
(38) Miklitz, M.; Jelfs, K. E. Pywindow: Automated Structural Analysis of Molecular Pores. *J. Chem. Inf. Model.* **2018**, *58*, 2387−2391.
(39) 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*, 1033−1044.
(40) 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*, 1884−1897.
(41) Chen, C. R.; Makhatadze, G. I. ProteinVolume: Calculating Molecular van Der Waals and Void Volumes in Proteins. *BMC Bioinf.* **2015**, *16*, 101.
(42) Voss, N. R.; Gerstein, M. 3V: Cavity, Channel and Cleft Volume Calculator and Extractor. *Nucleic Acids Res.* **2010**, *38*, W555−W562.
(43) Rother, K.; Hildebrand, P. W.; Goede, A.; Gruening, B.; Preissner, R. Voronoia: Analyzing Packing in Protein Structures. *Nucleic Acids Res.* **2009**, *37*, D393−D395.
(44) 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.
(45) Ho, B. K.; Gruswitz, F. HOLLOW: Generating Accurate Representations of Channel and Interior Surfaces in Molecular Structures. *BMC Struct. Biol.* **2008**, *8* (1), 49.
(46) Plajer, A. J.; Percástegui, E. G.; Santella, M.; Rizzuto, F. J.; Gan, Q.; Laursen, B. W.; Nitschke, J. R. Fluorometric Recognition of Nucleotides within a Water-Soluble Tetrahedral Capsule. *Angew. Chem., Int. Ed.* **2019**, *58*, 1087−1091.
(47) Davies, J. A.; Ronson, T. K.; Nitschke, J. R. Twisted rectangular subunits self-assemble into a ferritin-like capsule. *Chem.* **2022**, *8*, 1099−1106.
(48) Bloch, W. M.; Horiuchi, S.; Holstein, J. J.; Drechsler, C.; Wuttke, A.; Hiller, W.; Mata, R. A.; Clever, G. H. Maximized axial helicity in a Pd_2L_4 cage: inverse guest size-dependent compression and mesocate isomerism. *Chem. Sci.* **2023**, *14*, 1524−1531.
(49) Steed, J. W.; Atwood, J. L. *Supramolecular Chemistry*, 2nd ed.; John Wiley & Sons, Ltd: Chippenham, UK, 2009.
(50) 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*, 4012−4021.
(51) Cooke, G.; Rotello, V. M. Methods of Modulating Hydrogen Bonded Interactions in Synthetic Host-guest Systems. *Chem. Soc. Rev.* **2002**, *31*, 275−286.
(52) Gómez-González, B.; García-Río, L.; Basílio, N.; Mejuto, J. C.; Simal-Gandara, J. Molecular Recognition by Pillar[5]arenes: Evidence

https://doi.org/10.1021/acs.jcim.4c00355
J. Chem. Inf. Model. 2024, 64, 5604−5616

for Simultaneous Electrostatic and Hydrophobic Interactions. *Pharmaceutics* **2022**, *14*, 60.
(53) 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*, 15022−15026.
(54) 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*, 15204−15207.
(55) 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*, 8362−8367.
(56) 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*, 4552−4559.
(57) 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*, 8475−8483.
(58) 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*, 2744−2751.
(59) Escobar, L.; Ballester, P. Quantification of the Hydrophobic Effect Using Water-Soluble Super Aryl-Extended Calix[4]Pyrroles. *Org. Chem. Front.* **2019**, *6*, 1738−1748.
(60) 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.
(61) Zhang, J.; Lu, T. Efficient evaluation of electrostatic potential with computerized optimized code. *Phys. Chem. Chem. Phys.* **2021**, *23*, 20323−20328.
(62) Baker, N. Biomolecular Applications of Poisson-Boltzmann Methods. *Rev. Comput. Chem.* **2005**, *21*, 349−379.
(63) Still, W.; Tempczyk, A.; Hawley, R.; Hendrickson, T. Semianalytical treatment of solvation for molecular mechanics and dynamics. *J. Am. Chem. Soc.* **1990**, *112*, 6127−6129.
(64) Ginex, T.; Vazquez, J.; Gilbert, E.; Herrero, E.; Luque, F. J. Lipophilicity in Drug Design: An Overview of Lipophilicity Descriptors in 3D-QSAR Studies. *Future Med. Chem.* **2019**, *11*, 1177−1193.
(65) Jiang, X. Hydrophobic-lipophilic Interactions. Aggregation and Self-coiling of Organic Molecules. *Acc. Chem. Res.* **1988**, *21*, 362−367.
(66) 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.
(67) Furet, P.; Sele, A.; Cohen, N. C. 3D Molecular Lipophilicity Potential Profiles: A New Tool in Molecular Modeling. *J. Mol. Graph.* **1988**, *6*, 182−189.
(68) 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.
(69) 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*, 1319−1327.
(70) 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*, 587−596.
(71) 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*, 1201−1202.
(72) 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*, 388−398.
(73) Delano, W. L. PyMOL: An Open-Source Molecular Graphics Tool, 2002. https://sourceforge.net/projects/pymol/(accessed April 8, 2024).
(74) 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*, 357−362.
(75) 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. *Nat. Methods* **2020**, *17*, 261−272.
(76) 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.
(77) RDKit: Open-Source Cheminformatics Software. http://www.rdkit.org (accessed Dec 17, 2021).
(78) 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; pp 98−105.
(79) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An Open Chemical Toolbox. *J. Cheminf.* **2011**, *3*, 33.
(80) 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*, 3546−3557.
(81) Turcani, L.; Berardo, E.; Jelfs, K. E. Stk: A Python Toolkit for Supramolecular Assembly. *J. Comput. Chem.* **2018**, *39*, 1931−1942.
(82) 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*, 3546−3557.
(83) Hahsler, M.; Piekenbrock, M.; Doran, D. Dbscan: Fast Density-Based Clustering with R. *J. Stat. Soft.* **2019**, *91*, 1−30.
(84) Groom, C. R.; Bruno, I. J.; Lightfoot, M. P.; Ward, S. C. The Cambridge Structural Database. *Acta Crystallogr. Sect. B Struct. Sci.* **2016**, *72*, 171−179.
(85) 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*, 183−190.
(86) 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.
(87) Steel, P. J.; McMorran, D. A. Selective Anion Recognition by a Dynamic Quadruple Helicate. *Chem.—Asian J.* **2019**, *14*, 1098−1101.
(88) Eichstaedt, K.; Szpotkowski, K.; Grajda, M.; Gilski, M.; Wosicki, S.; Jaskólski, M.; Szumna, A. Self-Assembly and Ordering of Peptide-Based Cavitands in Water and DMSO: The Power of Hydrophobic Effects Combined with Neutral Hydrogen Bonds. *Chem.—Eur. J.* **2019**, *25*, 3091−3097.
(89) Yang, Y.; Ronson, T. K.; Lu, Z.; Zheng, J.; Vanthuyne, N.; Martinez, A.; Nitschke, J. R. A curved host and second guest cooperatively inhibit the dynamic motion of corannulene. *Nat. Commun.* **2021**, *12*, 4079.
(90) 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*, 15022−15026.
(91) 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*, 1827−1832.
(92) 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*, 18815−18820.
(93) Kawakami, Y.; Ogishima, T.; Kawara, T.; Yamauchi, S.; Okamoto, K.; Nikaido, S.; Souma, D.; Jin, R.-H.; Kabe, Y. Silane catecholates: versatile tools for self-assembled dynamic covalent bond chemistry. *Chem. Commun.* **2019**, *55*, 6066−6069.
(94) Horiuchi, S.; Murase, T.; Fujita, M. Diels-Alder Reactions of Inert Aromatic Compounds within a Self-Assembled Coordination Cage. *Chem.—Asian J.* **2011**, *6*, 1839−1847.
(95) Ronson, T. K.; Meng, W.; Nitschke, J. R. Design Principles for the Optimization of Guest Binding in Aromatic-Paneled $Fe^{II}_4L_6$ Cages. *J. Am. Chem. Soc.* **2017**, *139*, 9698−9707.
(96) Cullen, W.; Misuraca, M. C.; Hunter, C. A.; Williams, N. H.; Ward, M. D. Highly efficient catalysis of the Kemp elimination in the cavity of a cubic coordination cage. *Nat. Chem.* **2016**, *8*, 231−236.
(97) Li, R.-J.; Tarzia, A.; Posligua, V.; Jelfs, K. E.; Sanchez, N.; Marcus, A.; Baksi, A.; Clever, G. H.; Fadaei-Tirani, F.; Severin, K. Orientational self-sorting in cuboctahedral Pd cages. *Chem. Sci.* **2022**, *13*, 11912−11917.
(98) 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*, 18060−18072.
(99) Fujita, D.; Ueda, Y.; Sato, S.; Mizuno, N.; Kumasaka, T.; Fujita, M. Self-assembly of Tetravalent Goldberg Polyhedra from 144 Small Components. *Nature* **2016**, *540*, 563−566.
(100) 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*, 3762−3772.
(101) 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.
(102) 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.
(103) Fauchére, J.-L.; Quarendon, P.; Kaetterer, L. Estimating and representing hydrophobicity potential. *J. Mol. Graph.* **1988**, *6*, 203−206.
(104) 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*, 4315−4320.
(105) Bultinck, P.; Langenaeker, W.; Carbó-Dorca, R.; Tollenaere, J. P. Fast Calculation of Quantum Chemical Molecular Descriptors from the Electronegativity Equalization Method. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 422−428.
(106) Geidl, S.; Bouchal, T.; Raček, T.; SvobodováVařeková, R.; Hejret, V.; Křenek, A.; Abagyan, R.; Koča, J. High-quality and universal empirical atomic charges for chemoinformatics applications. *J. Cheminf.* **2015**, *7*, 59.

https://doi.org/10.1021/acs.jcim.4c00355
J. Chem. Inf. Model. 2024, 64, 5604−5616