JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

Application Note

Cite This: *J. Chem. Inf. Model.* 2018, 58, 2387−2391

pubs.acs.org/jcim

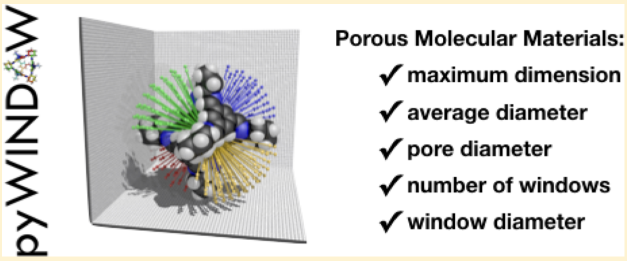
pywindow: Automated Structural Analysis of Molecular Pores

Marcin Miklitz and Kim E. Jelfs*

Department of Chemistry, Imperial College London, South Kensington, London SW7 2AZ, United Kingdom

Supporting Information

ABSTRACT: Structural analysis of molecular pores can yield important information on their behavior in solution and in the solid state. We developed pywindow, a python package that enables the automated analysis of structural features of porous molecular materials, such as molecular cages. Our analysis includes the cavity diameter, number of windows, window diameters, and average molecular diameter. Molecular dynamics trajectories of molecular pores can also be analyzed to explore the influence of flexibility. We present the methodology, validation, and application of pywindow for the analysis of molecular pores, metal−organic polyhedra, and some instances of framework materials. pywindow is freely available from github.com/JelfsMaterialsGroup/pywindow.



■ INTRODUCTION

The bulk properties of porous solids are greatly influenced by structural variations at the atomistic level.[1] The pore network, as defined by the size, shape, and interconnectivity of the pores, naturally impacts the material's properties. Thus, structural characterization is often a key part of computer-aided design of porous materials.[2,3] Common structural descriptors for porous networks are the largest cavity diameter (LCD), the largest sphere that can be inserted in the porous material without overlapping with any of the atoms, and the pore limiting diameter (PLD), the largest sphere that can freely diffuse through the porous network without overlapping with any of the atoms.[4] The LCD and PLD, in tandem with the computationally determined pore size distribution (PSD), surface area (SA), and pore volume, are often used to estimate the porous material's performance in silico.[5]

Various software exists for the structural characterization of porous materials. An early example, HOLE,[6] determines the ion channel dimensionality with a Monte Carlo (MC) simulated annealing procedure by probing the sphere sizes that can fit along the channel. MC sampling is commonly used to calculate the PLD, LCD, PSD, and SA in ZEOMICS,[7] MOFomics,[8] and Poreblazer.[9] The Zeo++ package[10] determines the pore network interconnectivity with a Voronoi tessellation.[11] The channel interconnectivity allows one to distinguish between accessible and nonaccessible SA. Lastly, the pore cavity shape can be analyzed in terms of its asphericity, acylindricity, and relative shape anisotropy with pyMolDyn,[12] which is also equipped with a graphical user interface.

The provided software examples are mostly intended for the analysis of porous materials in the solid state. However, if one wishes to characterize molecules of intrinsic porosity, also known as "molecular pores",[13] there are not many software options. Molecular pores are cage- or belt-like molecules that host an internal cavity that can be accessed by two or more windows. As these molecules are often soluble in common solvents, they can find in-solution applications such as sensing[14] or be incorporated into mixed-matrix membranes for molecular separations.[15,16] In the bulk, molecular pores can self-assemble and form a porous molecular material (PMM).[17] However, the structure of the resulting PMM is often not known until it is resolved experimentally, and it is dependent on the solvent and conditions used[18] and can be polymorphic[19] or amorphous.[20] Simultaneously, computational crystal structure predictions are still too expensive to be used routinely.[21] Thus, the structural descriptors of individual molecular pores, such as the void size and window diameter, and their relationship to properties might be more informative.[22] This is especially true if the isolated molecules are studied or the solid-state assembly and the void connectivity have a minimal effect on the properties of the resulting PMMs. Recently, Garcia et al.[23] reported the analysis of a single molecule's surroundings and the projection of void regions within the perimeter of a molecule. This method identifies windows but does not allow calculation of the window diameters.

The void size and window diameters yield a general description of the individual molecular pores in the same fashion as the LCD and PLD describe a porous network, and in some particular molecular assemblies these can be the same. However, if the software described in the previous section is used to characterize PMMs, one does not definitively obtain the underlying features of the individual molecular pores but rather obtains the features of the pore network as a function of

Special Issue: Materials Informatics

Received: July 23, 2018
Published: September 10, 2018

DOI: 10.1021/acs.jcim.8b00490

the molecular pores' assembly. Simultaneously, the manual measurement of window diameters using visualization software is laborious and prone to human error. This approach is also unfeasible for the analysis of thousands of frames of molecular dynamics (MD) trajectories. Here we report the python package pywindow, released under an MIT license and available at github.com/JelfsMaterialsGroup/pywindow. The software enables rapid analysis of the static and dynamic features of molecular pores, such as pore breathing from MD trajectories.

SOFTWARE OVERVIEW

The pywindow package is intended for structural analysis of molecular pores, whether in isolation or as PMMs.[17] The software was validated against the well-established Zeo++ package[10] where the functionalities of the two overlap and a literature circumcircle method to determine a circular window diameter in porous organic cages.[24] pywindow has no underlying units and inherits those of the input file, but all molecular coordinates are assumed to be in angstroms. The structural analysis does not depend on the given topology and chemistry of the molecule. Thus, other materials such as coordination polyhedra or representative fragments of metal—organic frameworks can also be analyzed.

A simplified schematic of the structure of pywindow is presented in Figure 1. The Cartesian coordinates and atom types can be extracted from a commonly used file extension (e.g., XYZ, PDB, MOL), ported directly from the RDKit[25] molecule object, or extracted from the trajectory file loaded with one of the trajectory module classes (XYZ, PDB, DLPOLY). The last of these requires preprocessing steps to extract each individual frame from the trajectory and to decipher force field atom keys, features that are implemented in pywindow. The input source can contain one or more molecular pores that can be individually extracted from the system for the analysis. Molecules split over cell boundaries due to periodic boundary conditions are reconstructed (see the Supporting Information). Lastly, the calculated values can be stored in JSON format.

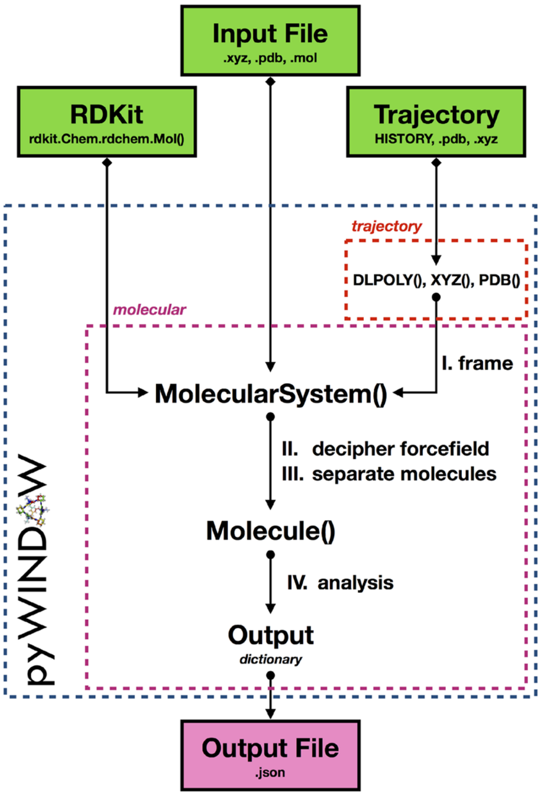


Figure 1. Schematic of pywindow and some possible workflows. The molecular module contains the MolecularSystem class, which is used to load in the input. In the case of trajectories, input is loaded through one of the available classes (DL_POLY, XYZ, or PDB) depending on the format. Each frame is then returned as a MolecularSystem object. If the MolecularSystem object contains multiple molecules, it is then separated into individual Molecule objects.

Structural Properties of Molecular Pores. The maximum diameter of a molecule (D_{max}) is defined as the distance between the edges of the van der Waals spheres of the two atoms at the greatest distance from each other in the molecule. The intrinsic void diameter (D_{void}) is calculated as the distance from the center of mass (COM) of a molecule to the van der Waals edge of the closest atom. This value is then used to calculate the spherical pore volume (V_{void}). However, in some unsymmetric systems, the COM might not coincide with the actual pore center. Thus, we implemented the optimized intrinsic void diameter (D_{void_opt}) and volume (V_{void_opt}), which includes COM coordinate minimization to find a better estimate of the pore center. The average diameter of a molecule (D_{avg}) is determined as the mean distance from the COM of a molecule to its van der Waals surface. Values of D_{avg} can match experimentally determined solvodynamic diameters, as shown in our recent work.[26] The methods of calculating these structural properties are detailed in the Supporting Information.

Finally, the most important feature of pywindow is the window diameter calculations. A schematic of the method is presented in Figure 2. The process is fully automated and does not require prior visualization of the molecule or assumptions about the number of windows. From a sphere of sampling points distributed evenly around a molecular pore, vectors connecting these points and the molecule's COM are analyzed for the largest included sphere that can be placed at a given vector point without overlapping with any van der Waals spheres of the host. This allows calculation of the largest sphere diameter that can fit into that window and thus the window diameter (see the Supporting Information for further details).

APPLICATIONS OF PYWINDOW

Computational Details. The crystallographic information files (CIFs) were obtained from the Cambridge Structural Database (CSD).[27] The molecular pores are denoted by the CSD references under which the CIFs were found in the CSD. The analyses were then performed using pywindow. For comparison and validation, the window diameters with the circumcircle method[24] and the LCD and PLD using the Zeo++0.2.2 package[10] were calculated.

MD simulations were performed for the previously obtained **CC3** organic cage (code PUDXES) using DL_POLY2.20.[28] The OPLS_2005 all-atom force field[29] was used. DL_POLY2.20 software input files were prepared using DL_FIELD3.3.[30] A 0.2 ns equilibration followed by a 1 ns production run was performed with a time step of 0.5 fs at 300 K. The Coulomb summation for the electrostatic interactions

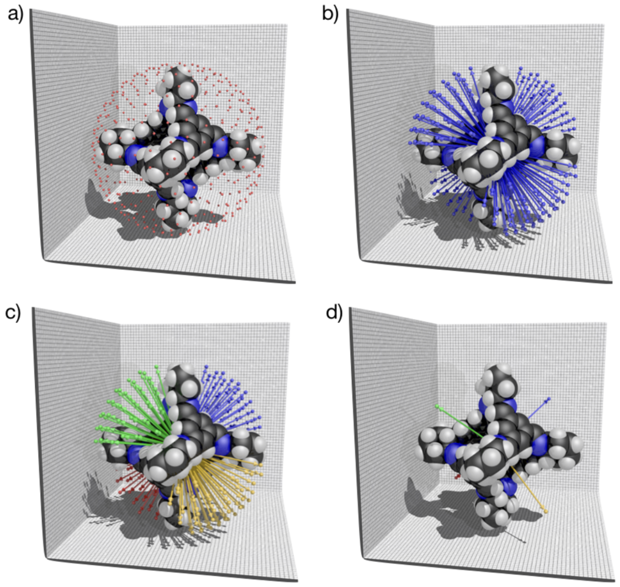


Figure 2. Steps to determine the window diameter. (a) A sphere of evenly distributed sampling points is generated around the host molecule. (b) Vectors connecting the center of mass of a molecule and the sampling points are analyzed for overlap with the host's atoms. (c) The vectors that do not overlap with any of the host's atoms (i.e., vectors passing through windows) are clustered into distinct windows. (d) For each window, the vector with the largest included sphere along its path is chosen, and the window's circular diameter is calculated.

and the leapfrog Verlet algorithm[31] were used. The image conversion was set to 0 in the CONFIG file.

Computational Expense. The time required to calculate a single window diameter of PUDXES is less than 1 s on a single CPU processor (2.4 GHz Quad Core Intel Xeon). However, this time will scale with number of windows, their size, and the maximum dimension of a molecule. The process of trajectory analysis is parallelized so that multiple frames can be analyzed simultaneously. Thus, the analysis is still cheap and easily achievable over a matter of hours or days at most with the benefit of an automated process for the extraction of the coordinates, reconstruction of the unit cell, and determination of structural properties of individual molecular pores.

Window Diameters and Number of Windows. The first step in validation of the pywindow package was the comparison of the calculated window diameters to those obtained using other methods. The porous organic cage (**CC3**) PUDXES system was chosen as a test case for two reasons. First, PUDXES was previously studied using the circumcircle method to calculate window diameters, allowing for a direct comparison with that method. Second, the window-to-window assembly of molecules in the crystal structure results in the window diameter coinciding with the narrowest point in the porous network, and therefore, it is possible to compare the PLD output by Zeo++ to the window diameter. PUDXES is shown in Figure 3.

The result of a comparison of the pywindow window analysis (D_{window}) to the window diameters determined with the circumcircle method and the PLD output by Zeo++ software is given in Table 1. The circumcircle method is also equivalent to a manual measurement in any software with a graphical user interface. The results for the three methods are comparable. The Zeo++ results give the largest window diameter of 3.66 Å. The circumcircle method gives a value of 3.63 Å and pywindow a value of 3.64 Å.

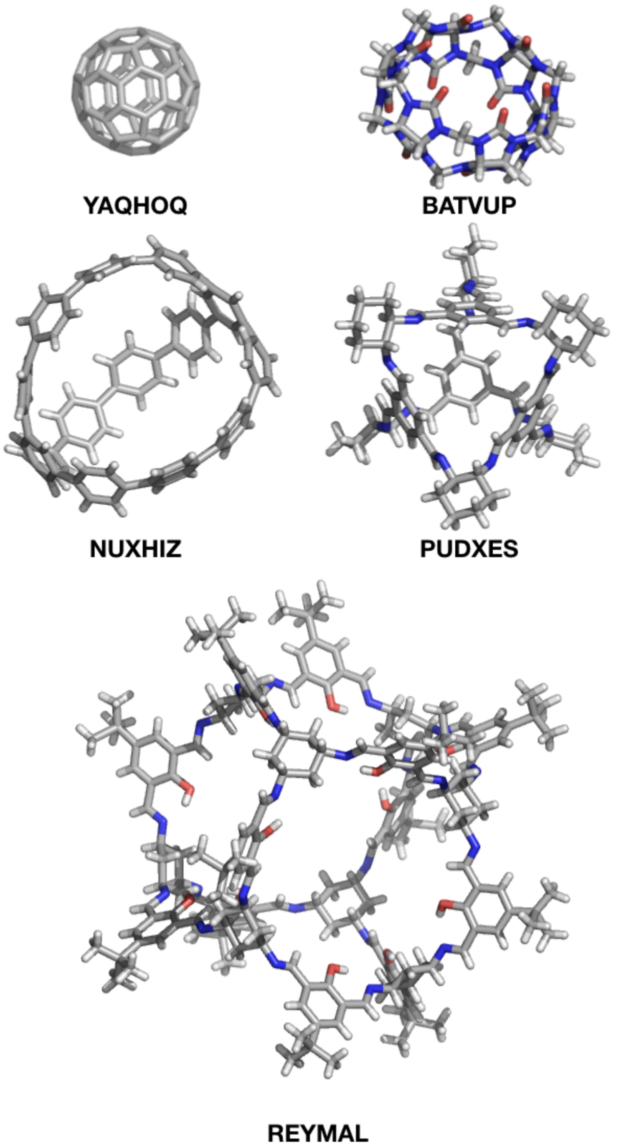


Figure 3. Five molecular pores chosen for pywindow validation. CSD reference codes are given below the molecules.

Table 1. Comparison of the Four Window Diameters Determined for PUDXES Using pywindow, the Circumcircle Method, and Zeo++[a]

method	$D_{win.1}$ [Å]	$D_{win.2}$ [Å]	$D_{win.3}$ [Å]	$D_{win.4}$ [Å]
pywindow	3.64	3.64	3.64	3.63
circumcircle	3.63	3.63	3.63	3.63
Zeo++0.2.2	3.66	3.66	3.66	3.66

[a] Only one PLD is generated by Zeo++. However, on the basis of symmetry considerations we report the value for each window separately.

Next, a set of molecular pores that are intrinsically porous and possess a variety of numbers of windows were analyzed with pywindow. These are shown in Figure 3. The BATVUP, NUXHIZ, PUDXES, and REYMAL molecules contain two, three, four, and six windows, respectively. YAQHOQ (C_{60}) has no windows and was used as a control. The results are shown in Table 2. The total number of windows identified for each molecule equals the number of windows identified from visual inspection.

Table 2. Structural Parameters Calculated with pywindow for Different Molecular Pores

system	D_{max} [Å]	D_{void_opt} [Å]	V_{void_opt} [Å^3]	$n_{windows}$	mean($D_{win.x}$) [Å]
YAQHOQ	10.5	3.6	25.0	0	–
BATVUP	14.8	5.0	63.6	2	3.5
NUXHIZ	18.6	9.0	377.0	3	7.2
PUDXES	22.2	5.4	82.3	4	3.6
REYMAL	34.0	13.8	1363.1	6	9.1

Analysis of Molecular Dynamics Trajectories. MD trajectories of molecular pores can be analyzed to study breathing effects by determining the pore-limiting envelope (PLE), a distribution of window diameters over the trajectory, the fluctuations of D_{void_opt} and the fluctuations of D_{max}. Results for PUDXES are presented in Figure 4. We recently published a study in which the PLE was used to characterize the possible application of molecular pores in xenon/krypton separation.[22]

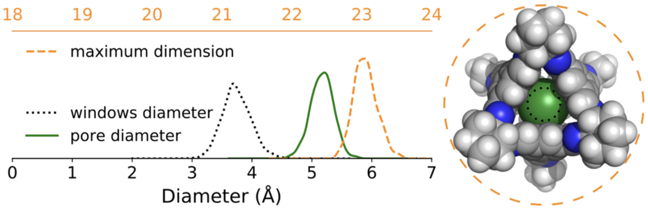


Figure 4. Histograms of the window diameter, pore diameter, and maximum molecular dimension from an MD trajectory for PUDXES (**CC3**), shown at the right. The window diameter and pore diameter histograms share the bottom *X* axis, whereas the maximum molecular dimension histogram is plotted with respect to the top *X* axis.

Other Material Classes. As pywindow depends only on Cartesian coordinates and atom types, host molecules need not be fully organic. For example, we analyzed a metal–organic cage (CSD reference: SAYGOR). Systems that are not inherently discrete, such as framework materials, can also be analyzed if the user provides a representative fragment to mimic an isolated molecule with no boundary conditions for void and window diameters to be calculated. As an example, we analyzed a fragment of a metal–organic framework (CSD reference: MIBQAR). The analyzed materials are shown in Figure 5, and the pywindow output is given in Table 3.

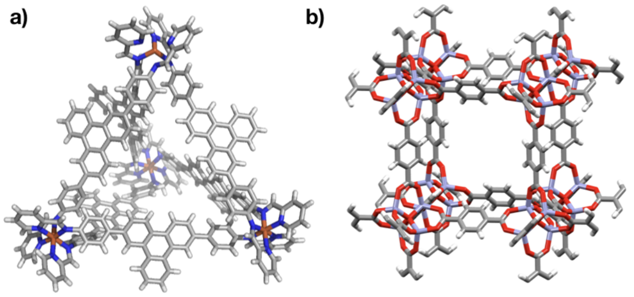


Figure 5. Examples of the transferability of pywindow to other materials: (a) a metal–organic cage (CSD reference: SAYGOR) and (b) a representative fragment of a metal–organic framework (CSD reference: MIBQAR).

Table 3. Structural Features Calculated for a Metal–Organic Cage (SAYGOR) and a Representative Fragment of a Metal–Organic Framework (MIBQAR) with pywindow

system	D_{max} [Å]	D_{void_opt} [Å]	V_{void_opt} [Å^3]	$n_{windows}$	mean($D_{win.x}$) [Å]
MIBQAR	–	12.3	968.9	6	7.9
SAYGOR	32.5	9.4	435.6	4	7.2

CONCLUSIONS

pywindow enables the structural characterization of molecular pores. The possibility of using pywindow to determine the number of windows and their diameters in molecules of various shapes and chemistry has been demonstrated, as has the applicability to metal–organic cages and, in certain circumstances, framework materials. The object-oriented character and modularity of the code make it easily extendable, and the code is freely available. pywindow can be used to determine whether a material has a predisposition to be porous and whether the windows are big enough to allow for guest diffusion. We have presented the functionality to analyze molecular dynamics trajectories in an automated and straightforward fashion. The pore-limiting envelope allows one to relate the dynamic change of window diameter to the diffusion of guests in flexible materials. The possibility that guest molecules will bind within the pores can be assessed with the calculation of the cavity size. There is no requirement to visually assess the molecules prior to the analysis, and no predefined information about the molecule is required. Thus, the software is easily applied for high-throughput simulation. The planned future extension of pywindow is to include methods for further structural properties, such as the host molecule's shape, in terms of asphericity and acylindricity, and the shapes of the cavities and windows.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jcim.8b00490.

> Description of the reconstruction of the periodic unit cell process, definitions of the structural properties of molecular pores, method of calculating the average molecular diameter, results for the examples from the main text, and discussion of the limitations of pywindow (PDF)

AUTHOR INFORMATION

Corresponding Author

*E-mail: k.jelfs@imperial.ac.uk. Phone: +44 (0)20 7594 3438.

ORCID

Marcin Miklitz: 0000-0002-0144-3116
Kim E. Jelfs: 0000-0001-7683-7630

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

We acknowledge the Royal Society (University Research Fellowship to K.E.J.), the EPSRC (EP/M017257/1 and EP/N004884/1), and the ERC (Grant Agreement 758370, ERC-StG-PE5-CoMMaD) for funding. We thank Dr. Enrico Berardo and Lukas Turcani for useful discussions.

■ REFERENCES

(1) Martin, R. L.; Smit, B.; Haranczyk, M. Addressing Challenges of Identifying Geometrically Diverse Sets of Crystalline Porous Materials. *J. Chem. Inf. Model.* **2012**, *52*, 308−18.

(2) Jelfs, K. E.; Cooper, A. I. Molecular Simulations to Understand and to Design Porous Organic Molecules. *Curr. Opin. Solid State Mater. Sci.* **2013**, *17*, 19−30.

(3) Thornton, A. W.; Winkler, D. A.; Liu, M. S.; Haranczyk, M.; Kennedy, D. F. owards Computational Design of Zeolite Catalysts for CO2 Reduction. *RSC Adv.* **2015**, *5*, 44361−44370.

(4) Haldoupis, E.; Nair, S.; Sholl, D. S. Pore Size Analysis of > 250,000 Hypothetical Zeolites. *Phys. Chem. Chem. Phys.* **2011**, *13*, 5053−5060.

(5) Simon, C. M.; Kim, J.; Gomez-Gualdron, D. A.; Camp, J. S.; Chung, Y. G.; Martin, R. L.; Mercado, R.; Deem, M. W.; Gunter, D.; Haranczyk, M.; Sholl, D. S.; Snurr, R. Q.; Smit, B. The Materials Genome in Action: Identifying the Performance Limits for Methane Storage. *Energy Environ. Sci.* **2015**, *8*, 1190−1199.

(6) Smart, O. S.; Neduvelil, J. G.; Wang, X.; Wallace, B. A.; Sansom, M. S. P. HOLE: A Program for the Analysis of the Pore Dimensions of Ion Channel Structural Models. *J. Mol. Graphics* **1996**, *14*, 354−360.

(7) First, E. L.; Gounaris, C. E.; Wei, J.; Floudas, C. A. Computational Characterization of Zeolite Porous Networks: An Automated Approach. *Phys. Chem. Chem. Phys.* **2011**, *13*, 17339.

(8) First, E. L.; Floudas, C. A. MOFomics: Computational Pore Characterization of Metal Organic Frameworks. *Microporous Mesoporous Mater.* **2013**, *165*, 32−39.

(9) Sarkisov, L.; Harrison, A. Computational Structure Characterisation Tools in Application to Ordered and Disordered Porous Materials. *Mol. Simul.* **2011**, *37*, 1248−1257.

(10) 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.

(11) Pinheiro, M.; Martin, R. L.; Rycroft, C. H.; Jones, A.; Iglesia, E.; Haranczyk, M. Characterization and Comparison of Pore Landscapes in Crystalline Porous Materials. *J. Mol. Graphics Modell.* **2013**, *44*, 208−19.

(12) Heimbach, I.; Rhiem, F.; Beule, F.; Knodt, D.; Heinen, J.; Jones, R. O. pyMolDyn: Identification, Structure, and Properties of Cavities/Vacancies in Condensed Matter and Molecules. *J. Comput. Chem.* **2017**, *38*, 389−394.

(13) Hasell, T.; Cooper, A. I. Porous Organic Cages: Soluble, Modular and Molecular Pores. *Nat. Rev. Mater.* **2016**, *1*, 16053.

(14) Zwijnenburg, M. A.; Berardo, E.; Peveler, W. J.; Jelfs, K. E. Amine Molecular Cages as Supramolecular Fluorescent Explosive Sensors: A Computational Perspective. *J. Phys. Chem. B* **2016**, *120*, 5063−5072.

(15) Evans, J. D.; Huang, D. M.; Hill, M. R.; Sumby, C. J.; Thornton, A. W.; Doonan, C. J. Feasibility of Mixed Matrix Membrane Gas Separations Employing Porous Organic Cages. *J. Phys. Chem. C* **2014**, *118*, 1523−1529.

(16) Mao, H.; Zhang, S. Mixed-Matrix Membranes Incorporated with Porous Shape-Persistent Organic Cages for Gas Separation. *J. Colloid Interface Sci.* **2017**, *490*, 29−36.

(17) Mastalerz, M. Permanent Porous Materials from Discrete Organic Molecules−towards Ultra-High Surface Areas. *Chem. - Eur. J.* **2012**, *18*, 10082−10091.

(18) Hasell, T.; Culshaw, J. L.; Chong, S. Y.; Schmidtmann, M.; Little, M. A.; Jelfs, K. E.; Pyzer-knapp, E. O.; Shepherd, H.; Adams, D. J.; Day, G. M.; Cooper, A. I. Controlling the Crystallization of Porous Organic Cages: Molecular Analogs of Isoreticular Frameworks Using Shape-Specific Directing Solvents. *J. Am. Chem. Soc.* **2014**, *136*, 1438−1448.

(19) Little, M. A.; Chong, S. Y.; Schmidtmann, M.; Hasell, T.; Cooper, A. I. Guest Control of Structure in Porous Organic Cages. *Chem. Commun.* **2014**, *50*, 9465−9468.

(20) Jiang, S.; Jelfs, K. E.; Holden, D.; Hasell, T.; Chong, S. Y.; Haranczyk, M.; Trewin, A.; Cooper, A. I. Molecular Dynamics Simulations of Gas Selectivity in Amorphous Porous Molecular Solids. *J. Am. Chem. Soc.* **2013**, *135*, 17818−17830.

(21) Pyzer-Knapp, E. O.; Thompson, H. P. G.; Schiffmann, F.; Jelfs, K. E.; Chong, S. Y.; Little, M. A.; Cooper, A. I.; Day, G. M. Predicted Crystal Energy Landscapes of Porous Organic Cages. *Chem. Sci.* **2014**, *5*, 2235−2245.

(22) Miklitz, M.; Jiang, S.; Clowes, R.; Briggs, M. E.; Cooper, A. I.; Jelfs, K. E. Computational Screening of Porous Organic Molecules for Xenon/Krypton Separation. *J. Phys. Chem. C* **2017**, *121*, 15211−15222.

(23) Garcia, I. G.; Bernabei, M.; Soto, R. P.; Haranczyk, M. Out-of-Oblivion Cage Molecules and Their Porous Crystalline Phases. *Cryst. Growth Des.* **2017**, *17*, 5614−5619.

(24) Holden, D.; Jelfs, K. E.; Cooper, A. I.; Trewin, A.; Willock, D. J. Bespoke Force Field for Simulating the Molecular Dynamics of Porous Organic Cages. *J. Phys. Chem. C* **2012**, *116*, 16639−16651.

(25) RDKit: Open-Source Cheminformatics Software. http://www.rdkit.org (accessed June 26, 2017).

(26) Greenaway, R. L.; Santolini, V.; Bennison, M. J.; Alston, B. M.; Pugh, C. J.; Little, M. A.; Miklitz, M.; Eden-Rump, E. G. B.; Clowes, R.; Shakil, A.; Cuthbertson, H. J.; Armstrong, H.; Briggs, M. E.; Jelfs, K. E.; Cooper, A. I. High-Throughput Discovery of Organic Cages and Catenanes Using Computational Screening Fused with Robotic Synthesis. *Nat. Commun.* **2018**, *9*, 2849.

(27) Groom, C. R.; Bruno, I. J.; Lightfoot, M. P.; Ward, S. C. The Cambridge Structural Database. *Acta Crystallogr., Sect. B: Struct. Sci., Cryst. Eng. Mater.* **2016**, *72*, 171−179.

(28) Smith, W.; Yong, C. W.; Rodger, P. M. DL_POLY: Application to Molecular Simulation. *Mol. Simul.* **2002**, *28*, 385−471.

(29) Jorgensen, W. L.; Maxwell, D. S.; Tirado-Rives, J. Development and Testing of the OLPS All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids. *J. Am. Chem. Soc.* **1996**, *118*, 11225−11236.

(30) Yong, C. W. Descriptions and Implementations of DL_F Notation: A Natural Chemical Expression System of Atom Types for Molecular Simulations. *J. Chem. Inf. Model.* **2016**, *56*, 1405−1409.

(31) van Gunsteren, W. F.; Berendsen, H. J. C. A Leap-Frog Algorithm for Stochastic Dynamics. *Mol. Simul.* **1988**, *1*, 173−185.

DOI: 10.1021/acs.jcim.8b00490