Research Article

Towards Automated Tools for Characterization of Molecular Porosity

Ismael Gómez García,[a,b] Marco Bernabei,[a] Maciej Haranczyk[a,c*]

[a] IMDEA Materials Institute, C/Eric Kandel 2, 28906 - Getafe, Madrid, Spain

[b] Universidad Carlos III de Madrid, Avda. Universidad 30, 28911 Leganés, Spain

[c] Lawrence Berkeley National Laboratory, One Cyclotron Rd, Berkeley, CA 94720, USA

* Corresponding Author; Email: maciej.haranczyk@imdea.org

Journal of Chemical Theory and Computation, 2019, 15, 1, 787–798

Abstract. The emerging advanced porous materials, e.g. extended framework materials and porous molecular materials, offer an unprecedented level of control of their structure and function. The enormous possibilities for tuning these materials by changing their building blocks mean that, in principle, optimally performing materials for a variety of applications can be systematically designed. However, the process of finding a set of optimal structures for a given application requires computational high-throughput tools to analyze and sieve through many candidate materials. In particular, in the case of porous molecular materials, the analysis and selection of a molecule is one of key aspects as the structure of the molecule determines the structure of the resulting material, and very often the porosity of the molecule significantly contributes to the porous properties of the resulting material. In this work, we introduce definitions and algorithms to characterize porosity at the molecular level, along with a software implementation of these algorithms. We demonstrate applications of the software tool in the discovery and characterization of porous molecules among ca. 94 million molecules currently enlisted in PubChem database.

1. Introduction

Advanced porous materials, e.g. extended framework materials and porous molecular materials, are being prototyped, and in some cases used, in various applications including gas separations[1,2], gas storage[3,4], sensing[5] and catalysis[6]. The modular chemistry underlying these materials has given the enormous possibilities for tuning their properties by changing their building blocks. Although, optimally performing materials for a variety of applications can be systematically designed, the process of finding a set of optimal structures for a given application is challenging due to the enormous search space of candidate materials, e.g. 10^{60} small organic molecules could serve as candidates for porous molecular material phases[7]. There is a substantial payoff for developing novel tools and approaches to accelerate the search and discovery processes. In particular, molecular simulations, crystal[8,9] and amorphous structure prediction[10], automated workflows[11–13], dedicated descriptors[14,15] and machine learning tools[16,17] have been investigated as components of such accelerated discovery efforts. Although a number of such tools and approaches have been combined into complex discovery workflows in applications to porous framework materials[3,4,18] similar tools and approaches dedicated to porous molecular material have been lagging behind.

Porous molecular materials (PMMs) are solids built from discrete, likely rigid, molecules interacting with each other through non-covalent bonds (Fig. 3-1). Porosity of PMM emerges as a combined effect of the geometry of the molecule itself and the molecular packing in the solid. Porous molecular materials[19] have been discussed in literature for about half-century, i.e. since the gas absorption capacity of molecule (tris(o-phenylenedi-oxy)cyclophosphazene, TPP) molecular crystal was reported in 1964[20]. Since then, the number of known porous molecular materials has been raising though the porosity of the vast majority of them[19], e.g. measured by internal surface area, has been rather limited when compared with framework materials. A recent survey of Cambridge Structure Database

identified 481 porous molecular materials and this number should be treated as lower bound due to the limitations of the methodology.[16]

In the last decade, the field of advanced porous molecular materials has entered a new era with the introduction of porous organic cage (POC) materials (Fig. 3-1b, 3-1d). POC is an organic molecule with an internal cavity and at least two windows granting access to it. POCs rely on modular chemistries such as imine condensation reaction[21], boronic ester condensation[22], or alkyne metathesis[23], which allow to achieve materials with high surface areas and large pores, e.g. up to mesoporous levels[24]. The major advantage of POCs is their solution processability. The resulting materials can be either crystalline or amorphous while the molecular porosity can be also exploited in solution[6].

Porosity in porous molecular materials can arise in multiple ways. First, the material porosity may be the result of its molecules being porous themselves, as is the case of cage molecules such as the family imine cages[21]. This case is often referred as *intrinsic porosity*, where there is a porous network connecting internal molecular pores. Second, the material porosity may be the result of inefficient packing of molecules in space, such that the voids between molecules are present even if the molecules themselves are not porous. This case is often referred as *extrinsic porosity* (Fig. 3-1a, 3-1c). Both intrinsic and extrinsic material porosity can coexist in the same solid as in, for example, porous organic cage material CC13[25]. Furthermore, there are other, less specific terms in use when referring to porous molecules of various chemistries, e.g. belt-like molecules, such as noria[26] or molecular squares[27–29]; or cup-like molecules such as calixarene molecules[30].

With such a variety in porous molecules and the corresponding porous solid phases, computational techniques can be used to predict properties of any materials ahead of synthesis, accelerating the discovery process. A discovery pipeline may consist of mainly three steps: (1) Molecule selection; (2) Solid-state structure prediction; (3) Properties and performance prediction for the materials from step (2) in the context of a given application.

Computational tools to perform Steps (2) and (3), e.g. electronic structure methods, molecular simulations and crystal structure predictions, are available and a recent review captures their state of the art in the context of PMMs[7]. Step (1) includes identification and characterization of porous molecules, which can serve as suitable candidates for porous materials. In some cases, molecular structure may even reflect material properties (for instance, the window size of a cage molecules may correlate with diffusion of a given species into the material or the internal cavity of a cage molecule can be a strong guest binding site). Automation of Step (1) with appropriate computational tools can streamline this step and thus enable high-throughput discovery of porous molecular materials adapted to different applications.

Until very recently, characterization of porous molecules was limited to standard molecular descriptors, e.g. mass, outside molecular dimensions (e.g. length, width), surface area, and ad hoc-developed descriptors that address porosity, e.g. pore diameter, which had to be taken 'by hand' with help from a visualization package. Only recently, the first effort to develop basic molecular porosity descriptors was presented by Miklitz *et al.* in the context of POCs for xenon/krypton separations[31]. Specifically, a methodology to calculate several molecular descriptors was introduced, in particular: 1) the maximum diameter of the molecule, understood as the largest distance between any two atoms of the molecule; 2) Intrinsic void diameter, defined as twice the distance between the center of mass of the molecule and its closest atom (corrected with Van der Waals radii); 3) Spherical void volume, defined as the volume of the sphere placed at the center of mass with the radius equal to the half of the intrinsic void diameter; 4) Number of windows for the molecule; 5) Window sizes. The number of windows and window sizes are defined through an algorithm that distributes probing rays uniformly from the center of mass of the molecule towards an outer sphere, then checks whether these rays come close enough to any atom, discarding those. Afterwards, the remaining rays are clustered with a density-based

clustering algorithm. The number of windows equals the number of clusters. Each cluster is then used to reconstruct a plane that represents the entry window, giving an estimate of window size. The approach by Miklitz et al. represents the first attempt to provide molecular porosity descriptors. As such, it offers a rather simple description, where one molecular pore is assumed per molecule with its center coaligned with the center of mass of the molecule; similarly, windows are assumed to be regular, having their size determined by the average of the set of rays that determines them.

In a recent publication, our group introduced a method to identify porous cages, along with their pore sizes[32]. In this work, we aim to extend our methods to establish definitions of molecular porosity and its numerical descriptors as well as present the corresponding algorithms. Our aim is to provide robust algorithms, which can detect and characterize diverse cases of molecular porosity.

Our algorithms and tools: 1) detect and characterize the internal voids of a molecule. Void characterization includes information about pore size and pore exposure; 2) detect and characterize windows and entry paths connecting the external space with the molecule's internal voids. As a result, the number of windows and entry paths with their corresponding sizes are provided; 3) calculate the diameter of a spherical probe that can access and/or occupy pores; as a result of solving this problem, the largest cavity diameter (LCD) and the pore limiting diameter (PLD) are calculated; 4) compute the internal surface area of the molecule. The algorithms and their implementation are demonstrated by performing high-throughput screening of PubChem 3D database to identify and characterize the porous molecules it contains.

2. Methods

2.1 Molecular representations

Formally, we define a molecule, M as a set of bonds and atoms, i.e.:

$$\mathrm{M} \equiv A_M \cup B_M \quad (1)$$

Where $A_M = \{a_1, \ldots, a_n\}$ is the set of atoms, and $B_M = \{b_1, \ldots, b_m\}$ is the set of chemical bonds. Each atom, $a_i = (x_i, y_i, z_i, r_i)$, is defined by its coordinates (i.e. the geometrical center of the atom) and its radius. In this representation, atoms are assumed to be hard spheres. Each bond is defined as a pair $b_j = (k, l)$, by the indices of the atoms it connects (k, l in {1, …, n}, where n is the number atoms; $k \neq l$).

Molecular information (i.e. atom positions and type, chemical bonds) is provided as input, and molecules can origin from any chemical repository such as such as PubChem[33] or the Cambridge Structural Database[34]. Molecules are treated as rigid objects. Molecular probes (i.e., candidates to enter the cavities of a studied molecule) are also assumed to be hard spheres with user-specified radii.

2.2 Identification of void space: Voronoi tessellation

In order to analyze the space around atoms, we use Voronoi tessellation[14], a computational geometry technique that, given a set of points, constructs cells, the boundaries of which are defined by points being in equal distance to the neighboring points. Vertices and edges of the Voronoi cells form a graph (set of nodes connected by edges), which represents the space between the points. In particular, given the set of atoms, A_M, we obtain a graph, named Voronoi graph, defined as:

$$\mathrm{V}(A_M) \equiv V_N \cup V_E \quad (2)$$

where $V_N = \{n_1, \ldots, n_k\}$ is the set of Voronoi nodes, and $V_E = \{e_1, \ldots, e_j\}$ is the set of Voronoi edges. Each node $n_i = (x_i, y_i, z_i, d_i)$ is defined by its 3D coordinates and the

distance d_i to the surface of closest atom. Each edge $e_i = (n_a, n_b, D_i)$ is defined by the indices of the two Voronoi nodes connected by the edge (n_a, n_b, with a,b in {1, …, k}, a ≠ b), and the distance D_i to the surface of the closest atom. In order to take into account atom radii to estimate the distances d_i and D_i, two approaches can be used: 1) Radical Voronoi tessellation, which modifies the distance function to correct nodes positions in an approximate manner, or 2) "High accuracy" Voronoi tessellation[35], which substitutes each atom by a set of smaller spheres of fixed radii. The former is faster, whereas the latter is more precise. Details about different versions of Voronoi tessellation can be found in the Supporting Information (SI) file. In this work, we used the Voro++ library[36] to compute Voronoi tessellation.

2.3 Point Exposure Map

Our approach is based on the observation that a molecular pore is created by atom-made boundary(ies). In particular, the boundaries not only need to limit the space but also surround the space in order to create an internal void – the pore. Alternatively, in order to determine if a given point in space is inside a molecule, we need to determine how exposed the point is to the surroundings of the molecule. Intuitively, this can be done as follows: a sphere, centered at the point of interest, large enough to surround the molecule, is defined. Over that sphere, we define the point exposure map (with respect to the molecule) as the complement of the shadow of the molecule taking the studied point as a non-directional light source. Next, we outline how to formally define this idea.

Let M be the molecule defined as in Eq. (1), and x the point to be studied. Let S be a sphere with center x and radius $R = \max(\{\text{dist}(x, a) : a \in A_M\})$, and Ps = {$s_1$, …, s_n} a set of points over S, distributed according to Vogel's method[37]. Vogel's method allows to distribute a set of points over the surface of a sphere in an even manner (any such method would serve to our purpose). A set of N points with Vogel's method is distributed

constructing a spiral over the surface of the sphere. Using cylindrical coordinates each point i is defined as follows: the angle $\rho_i = \theta_i$, where θ_i is a multiple of the golden angle; the radius $\tau_i = \sqrt{1 - z_i^2}$; and $z_i = (1 - \frac{1}{2})(1 - \frac{2i}{N-1})$.

Let T be a triangulation of P_s. In T, all triangles have similar surface areas (see Fig. 3-2). We want to erase triangles from T to obtain a surface that reflects how much the molecule M surrounds the point x. Let H_s be the set of tetrahedrons defined as:

$$H_s \equiv \{(x, t) : t \in T\} \quad (3)$$

The set of "triangles to be erased", T_e, is defined as:

$$T_e \equiv \{ t: (x, t) \cap M \neq \emptyset, (x, t) \in H_s\} \quad (4)$$

Finally, the set of remaining triangles, to which we will refer as the *exposure map* for point x, is defined as the subtraction of triangles:

$$E_{map}(x, M) \equiv T - T_e \quad (5)$$

An example of the exposure map for three different molecules representative of different molecular motifs can be seen in Fig. 3-2. This set of triangles may have several connected components (two triangles are connected if they share an edge). Also, we refer to the surface of a triangle t as S(t), and to the surface of a component C as S(C). The surface of the initial triangular grid is S(T).

2.4 Identification of internal space: Pore Exposure Ratio

In order to decide whether a given point in space is either internal or external with respect to the molecule, we introduce a technique to assign any given point x a number between 0 and 1. We define this as the *Pore Exposure Ratio* (PER). Briefly speaking, PER is the ratio between the surface area of largest connected component on the exposure map and the surface of the initial surrounding sphere. This number will never be larger than 1 (as all connected components are subsets of the initial spherical grid), and it will tend to be

smaller when the point is surrounded from many different directions, approximating our intuition of what is being "inside the molecule". To compute the PER, we make use of the Point Exposure Map, defined in the previous section. Computing the connected components of PEM is done through a standard depth-first algorithm.

Let x be any point, and M the reference molecule. Let $E_{map}(x, M)$ be the exposure map for x. The exposure map can be defined as the union of its n connected components, i.e.:

$$E_{map}(\mathrm{x}, \mathrm{M}) = \bigcup_{i=1}^{n} C_i \quad (6)$$

We consider two triangles to be connected if they share at least an edge. For simplicity, we assume the components to be inversely sorted by surface (i.e., $S(C_i) < S(C_j)$ if $j>i$). Then, the component with maximum surface is C_1, and Pore Exposure Ratio for x is defined by the formula:

$$\mathrm{PER}(x) \equiv \frac{S(C_1)}{S(T)} \quad (7)$$

PER(x) is a number between 0 and 1 as C_1 is a subset of T, and it defines "how much the molecule M surrounds the point x".

The pseudocode of PER algorithm is described in Scheme 1 included in the SI. The Pore Exposure Ratio provides an insight on how much exposed the x point is with respect to the molecule. The PER when combined with a criterion to classify the points as internal (described in next section), and with Voronoi tessellation, allow us to detect molecular cavities.

2.5 Threshold selection

In order to decide if a point is to be considered 'internal to the molecule', we set a threshold for the PER value, such that any point with PER below that threshold will be considered 'internal'. To select a threshold, PER value was computed for the center of mass of 337 molecules extracted from PubChem and Cambridge Structural Database.

Visual inspection led us to set PER threshold at 0.45, in order to include molecular cages, belts and cups as porous molecules.

Selection of PER threshold provides us with a uniform criterion to decide whether any point in space is inside or outside a molecule. This is crucial for the task of finding molecular cavities, and has other applications, discussed in section 4.

2.6 Cavity detection and characterization

Characterization of the internal space of a molecule is a key task in identifying porous molecules, and characterizing their porosity in terms of pore size, entry paths, windows, and internal surface area. Our algorithm for characterization of the molecular internal space combines the different techniques discussed in the previous sections: 1) Voronoi tessellation to find empty space; 2) PER to determine which Voronoi nodes (i.e. pore candidates) are internal.

Our objective is to find a set of internal Voronoi nodes, i.e., best candidates to be molecular pores. The set of internal Voronoi nodes, V_{IN}, is defined as follows:

$$V_{IN} \equiv \{n \in V_N : \mathrm{PER}(n) \leq 0.45\} \quad (8)$$

An algorithm to compute this set is described in Scheme 2 in the SI and depicted as part of Fig. 3-3.

2.6.1 Largest molecular cavity and its two descriptors. Characterization of molecular cavities permits identification of the molecule's largest cavity. In summary, the largest cavity will correspond to the internal Voronoi node with the largest distance to atoms. Let $V_{IN} = \{n_1, \ldots, n_p\}$, and $d_{max} = \max(\{d_1, \ldots, d_p\})$. Let $n_{max} = (x_{max}, y_{max}, z_{max}, d_{max})$ be the internal Voronoi node with $d = d_{max}$ (i.e., with maximum distance to atoms). n_{max} is the largest cavity of the molecule. Now we can define Largest Cavity Diameter (LCD) for the molecule as the distance from the largest cavity to its closest atom (corrected by atom radii). Formally:

$$\mathrm{LCD(M)} \equiv 2d_{max} \quad (9)$$

We can also define the molecule's PER as the PER value assigned to the largest cavity:

$$\mathrm{PER(M)} \equiv \mathrm{PER}(n_{max}) \quad (10)$$

LCD(M) and PER(M) are two molecular descriptors of interest in this study. LCD(M) provides general information about molecule's porosity (i.e. size of largest probe that can lie inside the molecule). The molecule's PER gives an intuition of the shape of the molecule: lower PER vales are associated to cage-like molecules, and with more constrained pores, whereas higher PER values are associated with belt-like molecules and less constrained pores. Algorithm for LCD and molecular PER is described in Scheme 2 in the SI.

2.7 Access to molecular pores

In many applications of porous molecular materials, molecular cavities have to be accessed by a guest molecule, which is represented by a spherical probe of a given radius. Due to the molecular shape, there may be several access routes, and they may restrict the size of the probe. In order to describe pore accessibility, we introduce two definitions. A *chemical window* is a set of connected atoms and the corresponding chemical bonds that form an opening connecting the outside of the molecule with its internal cavity. An entry path is a possible trajectory of entering the internal cavity that corresponds to locally-largest probe. Entry paths help with the detection of irregular windows that may allow access of molecular probes in multiple ways, as described in Fig. 3-4. Chemical windows and entry paths are related concepts. Every entry path will cross through exactly one chemical window, and every chemical window must have at least one entry path crossing it. Irregular windows are expected to have more than one entry path.

2.8 Detection of chemical windows

We want to identify chemical windows, i.e., bonds and atoms restricting access to the molecule. To detect chemical windows, we rely on the definition of exposure map (see Section 2.3) of a special point.

Let M be the molecule, and c the "center of the molecule" ($c = \frac{1}{n_a}\sum_{i=1}^{n_a} a_i$, a_i being coordinates of atoms, n_a being the number of atoms). We can write the exposure map as the union of its n connected components as in formula (6). Each connected component is a set of triangles, and can also be expressed in terms of the edges of those triangles:

$$C_i = \bigcup_{j=1}^{n} e_{ij} \tag{11}$$

Where $\{e_{i1}, ..., e_{in}\}$ are the edges associated to C_i. For each edge of the component, we define the associated bond b_{ij}, as follows:

$$b_{ij} = \{b : dist(b, e_{ij}) = \min_{b \in B_M} dist(b, e_{ij})\} \tag{12}$$

In words, b_{ij} is the chemical bond with least distance to e_{ij}. Due to molecular geometry, this bond is unique. The chemical window associated to the component C_i is then defined as the set of chemical bonds associated to its edges, along with the atoms. We define the set of bonds for window i (with n edges) as:

$$B_i = \bigcup_{j=1}^{n} b_{ij} \tag{13}$$

Notice that two component edges could share a bond. Thus, B_i does not necessarily have n elements (this will very rarely happen). The set of atoms for window i corresponds to those atoms connected by the bonds retrieved in previous step. Formally:

$$A_i = \{a : \exists b \in B_i \; such \; that \; a \in b\} \tag{14}$$

The window associated to C_i, W_i, is defined as:

$$W_i \equiv A_i \cup B_i \tag{15}$$

The set of windows of the molecule, W, is defined as the union of the windows associated to each component of the exposure map of the center of mass, i.e., if the molecule has n windows:

$$W \equiv \cup_{i=1}^{n} W_i \tag{16}$$

The pseudocode for chemical window detection algorithm is described in Scheme 4 in the SI. This process is also depicted as part of Fig. 3-3.

2.9 Detection of entry paths

We want to identify entry paths, i.e. ones that lead from the surroundings of the molecule to its internal cavities that not cross any chemical bonds or atoms, and move across the largest openings in the molecular structure. To identify entry paths, we first consider the set of Voronoi edges connecting internal nodes with external nodes (i.e., nodes with PER > 0.45). We call this the set of "potential entries", V_{EPE}.

$$V_{EPE} \equiv \{(n_a, n_b) \in V_E : (a \in V_{IN}) \nabla (b \in V_{IN})\} \tag{17}$$

Potential entries are then grouped by the window they traverse through. This is decided by a total distance criterion, i.e., e in V_{EPE} crosses through W_i if:

$$\sum_{a \in A_i \subset W_i} dist(e, a) = \min_{W_j \in W} \{\sum_{a \in A_j \subset W_j} dist(e, a)\} \tag{18}$$

A_i defined as in section 2.8.1. For each window, the set of potential entries traversing through that window is defined as W_{EPE}. Consider this set to be sorted by edge distance to atoms (i.e., edge with maximum D_i first), and let e_{W1} the first edge of the list and D_{W1} its distance value. A subset of the previous one, called "relevant entries", W_{RE}, is defined as:

$$W_{RE} \equiv \{ (n_a, n_b, \mathrm{D}) \in W_{EPE} : D \geq 0.8 \cdot D_{W1}\} \tag{19}$$

Another subset, called "far enough entries", W_{FEE} is defined as:

$$W_{FEE} \equiv \{ \mathrm{e} \in W_{EPE} : dist(\mathrm{e}, E_{W1}) \geq D_{W1}\} \tag{20}$$

The set of "entry paths for Window W_i", $EP(W_i)$, is defined as:

$$\mathrm{EP}(W_i) \equiv E_{W1} \cup W_{RE} \cup W_{FEE} \quad (21)$$

This construction obeys to the need of deleting an excess of Voronoi edges, keeping only the one furthest to atoms, and those far enough from atoms (at least 0.8 times the distance of the most relevant) and far enough from the most relevant to be accounted for. The 0.8 threshold was chosen empirically.

The set of molecule's entry paths corresponds to the union of entry paths of all its windows, i.e.:

$$\mathrm{EP(M)} \equiv \cup_{W_i \in W} EP\,(W_i) \quad (22)$$

Information on the number of paths across a given window can provide information on how irregular its shape is (regular windows will have only one entry path, whereas elongated or bending windows may have more). More generally, the number of entry paths to the molecule corresponds to the cardinality of the EP(M) set. The algorithm to compute the entry paths is described in Scheme 5 of the SI. This process is also depicted as part of Fig. 3-3. Entry paths are also used to compute Pore Limiting Diameter of the molecule, as it will be discussed in Section 2.11.

2.10 Window size

To determine the size of the largest molecular probe that can pass through a chemical window, we define window size. It corresponds to twice the largest distance from any entry path obtained for this window to its atoms. It is given by the expression:

$$\mathrm{WS(W)} \equiv 2 \max_{(\{n_a, n_b, \mathrm{D}) \in \mathrm{W}\}} D \quad (23)$$

Window size allows introducing a molecular descriptor called the maximum window size, defined as the largest size among all window sizes in the molecule. Formally:

$$MWS(M) \equiv \max_{W_i \in W} WS(W_i) \quad (24)$$

2.11 Pore limiting diameter

Pore Limiting Diameter (PLD) is defined as the twice the radius of the largest (spherical) probe that can enter any molecular cavity, and lie within it. Such value is determined by both the size of the cavities inside the molecule, and the size of the entry paths used to reach those cavities.

To compute PLD, we consider a subset of Voronoi graph. First, consider the set of internal Voronoi nodes (V_{IN}, see Eq. 8) and the set of entry paths (M_{EP}, see Section 2.8.2). The restricted Voronoi graph, V_R, is defined as:

$$V_R \equiv V_{IN} \cup M_{EP} \cup V_{IE} \tag{25}$$

Where V_{IE} is the set of internal edges, i.e., edges connecting internal nodes. Formally:

$$V_{IE} \equiv \{ (n_a, n_b, D) \in V_E : (n_a \in V_{IN}) \wedge (n_b \in V_{IN}) \} \tag{26}$$

This restricted graph contains all the relevant information about accessibility to molecular cavities: restrictions to access the molecule are provided by distances associated to entry paths (contained in M_{EP}), whereas restrictions associated with cavities are provided by internal pores (V_{IN}), and restrictions in movements inside the molecule are provided by internal edges (V_{IE}). To compute PLD, a modified Dijkstra algorithm is applied to V_R. Classical Dijkstra algorithm[38] explores a graph, and aims to find a path (i.e., a list of edges) with minimum total 'weight'. In our case, weight for each node and edge is the associated distance to atoms (d for nodes, D for edges). Our variant of the algorithm aims to find the path with largest opening (i.e., the path that allows the largest probe to get in). Formally, let $p = \{e_1, \ldots, e_n\}$ be a path of length n, the restriction of p, r(p), is defined as:

$$r(p) \equiv \min(\{D_1, d_{1a}, d_{1b}, \ldots, D_n, d_{na}, d_{nb}\}) \tag{27}$$

Where D_i is the distance associated to the edge e_i, and d_{ia}, d_{ib} are the distances associated to the nodes a_i and b_i connected by the edge e_i (for i in {1, … , n}). Now, if P is the set of all possible paths in V_R (i.e., connected sequences of edges), we want to find p_{max}, the path with maximum restriction, I.e.:

$$p_{max} \equiv \{ p \in P : r(p) = \max_{p \in P}(r(p)) \} \quad (28)$$

Then, the pore limiting diameter for the molecule, PLD(M), is defined as the restriction of p_{max}, i.e.:

$$PLD(M) = 2\, r(p_{max}) \quad (29)$$

Details about modified Dijkstra algorithm are provided in Scheme 6 of the SI. More information about how Dijkstra algorithm works can be found in SI.

Information about PLD for a molecule provides information about the maximum intrinsic storage capacity of that molecule. Comparing PLD with LCD, for example, it can be seen if intrinsic pores are accessible.

2.12 Internal Surface Area

To compute the internal surface area, we introduce a (Monte Carlo)-based method, divided in four steps. First, a fixed number of random points are sampled over each atom's surface. Points are sampled to guarantee uniform distribution per surface unit. To do so, for each point, two random numbers u and v are uniformly sampled in the range $[0,1]$. The spherical coordinates for the random point are then obtained as $\theta = 2\pi u$, $\phi = arccos(2v - 1)$ and $r = r_{atom}$. Second, each point is tested to be outside the intersection of atoms, checking distance from the point to every other atom: if it's lesser than atom radius, point is not in atom's overlap region. Third, each point is tested to be internal or external. To avoid overload due to computation of PER value for so many points, sampled points are classified as internal if closest Voronoi node from Voronoi graph is internal, and as external otherwise (see Fig. 3-5 for more details). Fourth, the ratio of internal points is calculated (i.e., the fraction of internal points divided by total number of sampled points) for each atom. Internal surface area of that atom is defined as atom's surface multiplied by ratio of internal points sampled. Internal surface area of the molecule

is defined as the sum of all internal surface areas of its atoms. The pseudocode description of this algorithm is shown in Scheme 7 of the SI.

2.13 Implementation

All the previously outlined algorithms were implemented in Molipor tool (www.nanoporousmaterials.org/programs/), which has been written in C++ language, with help of the Eclipse IDE for C/C++ developers (version 1.3.2)[39]. The g++ compiler (clang-700.1.76) was used for compilation tasks. Debugging tasks were performed with help of gdb[40]. For the task of Voronoi tessellation computing, Voro++ library was integrated with our code[36].

2.14 Validation

Validation of the presented algorithms was performed using a set of ca. 337 molecules, 286 of which can be found in the PubChem database, and 51 of which can be found in CSD. These molecules represent four groups based on their structure: cage-like, belt-like, cup-like and non-porous molecules, respectively. The results obtained from applying our tool over these molecules were evaluated against visual inspection.

2.14.1 PER Validation. In Fig. 3-6, six example molecules are presented along with the PER values associated with either their pores (if any) or the center of the box (for non-porous molecules). The box is defined when computing Voronoi tessellation, as the minimum rectangular box parallel to axis that contains the molecule. Porous cages have PER values under 0.36, with lesser values for molecules with more constrained internal space. For cup-like and belt-like molecules, PER values are close to 0.45. Differentiation between cup-like and belt-like is not achieved solely by studying PER. For the non-porous molecule shown, PER value goes far beyond, reaching 0.8. This is a common feature of such molecules.

2.14.2 Chemical window validation. To confirm that the chemical windows are being correctly identified, a visual inspection was conducted over the set of 337 molecules. The chemical window algorithm computed the correct number of windows for each molecule. Results for five example molecules are shown in Fig. 3-7.

2.15 Dataset for high-throughput analysis

Dataset construction was performed starting from the PubChem Compound Database. Several filters were applied, keeping only molecules that appeared alone (e.g. excluding salts), were organic, non-charged, non-radical, with less than 8 rotatable bonds (to favor rigid molecules) and with more than 48 atoms (estimated minimum size of porous molecule). A set of 1 258 975 candidates for porous molecules was identified. The database construction process is discussed with more details in a recent publication by our group[32]

3. Results

We demonstrate the developed algorithms by statistical analysis of an initial set of 1 258 975 molecules, out of which 6020 showed porosity levels of interest (i.e. pore size > 1.4 A). We analyzed this set of 6020 molecules, observing structural and porosity features of the molecules present in the set, including: 1) Histograms with number of molecules within a given range, for the six developed molecular porosity descriptors (see Fig. 3-8); 2) Graphical representation of PER value versus other four parameters: LCD, PLD, the number of windows and surface area (see Fig. 3-9); 3) Pairwise correlations of the six molecular descriptors (see Fig. 3-10).

3.1 Molecular descriptors distribution

Our molecular database study revealed several patterns among the different descriptors developed in this work, as highlighted in Figs. 3-8 and 3-9. Interestingly, the number of

molecules featuring exactly 2 windows outnumbers the others, indicating a high number of belt-like molecules. These molecules are, potentially, building blocks for porous materials. Our analysis revealed more than 4000 of these belt-like molecules.

3.2 Molecules and PER values

We also represented PER values against different molecular descriptors (see Fig. 3-9), expecting that PER would help identifying different types of molecules. By observing PER histogram (see Fig. 3-8), most molecules have values close to 0.4, which corresponds to the intuition of belt-like molecules having such value. When comparing number of windows with PER, this intuition gets reinforced: molecules with exactly 2 windows accumulate after 0.28 (with few exceptions corresponding to bending belts). It can also be seen that, the higher the number of windows, the lesser the PER values around which molecules accumulate, although for 3 and 4 windows PER values are widespread. When studying PER compared to LCD and PLD, there can be multiple pore sizes around one given PER value, although there's a tendency of higher PER to be associated with higher pore size. Molecules with pores bigger than 10 Å are accumulated PER values over 0.4. Visual inspection revealed that this corresponds to belt-like molecules.

3.3 Correlations among descriptors

We studied the pairwise correlations between the six descriptors developed during this work (PER, LCD, PLD, number of windows, number of entries and internal surface area) plus maximum window size (see eq. 24). Correlation values are presented in Fig 3-10.

Several strong positive correlations between several parameters were found:

- LCD, PLD and maximum window size are strongly correlated with each other (all three pairwise Spearman correlations are higher than 0.89, with 0.99 Spearman correlation for PLD and maximum window size), indicating that pore

size and restriction to pore access (determined by PLD) are typically of similar magnitude. High correlation between PLD and the maximum window size indicates that the window size is playing an important role on determining PLD.

- Number of windows and number of entries also show a high positive correlation (0.92 Spearman correlation). This indicates that, in most cases, windows are regular enough to consider them to have just one entry path (recall that every so-called chemical window has at least one entry path).
- Moderate positive correlation was found between PER and PLD and the maximum window size (>0.4 Spearman correlation), although not between PER and LCD, indicating that low PER is expected to be associated with more restricted access routes, although not with smaller cavities. Internal surface area correlates weakly (absolute value <0.3 Spearman correlation) with all the other parameters.

4. Discussion

Molecular descriptors developed in this work are based on a geometry-based approach. A different approach would be the use of topology-based techniques such as graph-theory[41] and/or homology groups[42]. Graph theory, for instance, would allow studying the molecule as if it was a graph, thus reducing the problem of window detection to a cycle detection. Homology groups represent another way to approach the same problem, as they are a natural way to describe holes in the surface of an object. Although topology-based techniques may seem attractive at first sight, we see a major drawback to their use. Under topological approaches, any deformation of the structure that does not imply breaking the structure (i.e., that leaves bonds untouched) would remain undetectable, whereas it is obvious that deformations on molecular structures affect critical aspects such as window

size, pore size, number of entries, or even number of windows. Thus, we based our approaches on geometry-based techniques, limiting the involvement of graph theory to a modified version of Dijkstra algorithm is used to find the least restricting path towards the insides of the molecule.

Compared to the recent publication of Miklitz et al.[31], our approach has the advantage of not assuming molecules to be porous at the first place. In Miklitz's work, the molecular pore is assumed to be in the center of mass of the molecule, which works for cage molecules but is not generally true for porous molecules and/or molecules in porous molecular materials. In our work, we avoid this assumption by first computing the Voronoi graph (which gives a list of potential cavities) and then studying nodes of this graph to see which ones are inside the molecular pore, using the largest (i.e., the one placed furthest away from atoms) as the largest cavity of the molecule. In the case of detection of molecular windows, our approach is similar to the one of Miklitz *et al.* as it requires a reference point (geometrical center) to perform analysis. However, our approach goes into greater level of detail when it comes to window characterization. All geometrically restricting atoms and bonds (and thus detailed geometry of the window) are recovered, and allow accurate characterization of non-symmetric windows, i.e. by analyzing the Voronoi graph, in particular its nodes and edges going through the to-be-analyzed window, it is possible to detect a number of geometrically-relevant, independent entry paths and the corresponding largest spherical probe diameters that can pass through the window following these paths.

Finally, we note that two of our algorithms, one for detection of porous molecules using PER and one for identification of entry paths, rely on threshold selection. Through the presented work, i.e. analysis of hundreds of examples, we have come up with thresholds that in our view best represent the chemical intuition required in high-throughput studies

of large databases. However, future studies that may focus on specific aspects of molecular porosity may require a refinement of these thresholds.

5. Conclusions

In this work, we presented a set of definitions and algorithms to describe porosity at the molecular level. Six descriptors of molecular porosity were introduced the Largest Cavity Diameter (LCD), the molecular PER, the Pore Limiting Diameter (PLD), the number of windows, the number of entry paths, and the Internal Surface Area (ISA). Calculation thereof relies on the geometry representation of a molecule, i.e. the molecule is defined as a set of hard sphere atoms with a given coordinates and radii. An implementation of our set of algorithms was used to identify porous molecules in PubChem database, and subsequently characterize a set of 6020 molecules. It was observed that majority of porous molecules in this repository have 2 windows, and that their PLDs and window sizes are highly correlated. Also, high correlation between the number of windows and the number of entry paths indicates that most windows are regular in shape.

The presented contributions can be integrated with high-throughput discovery workflows for porous molecular materials[43] as well as employed in detailed characterized on particular porous molecules of interest.

6. Figures

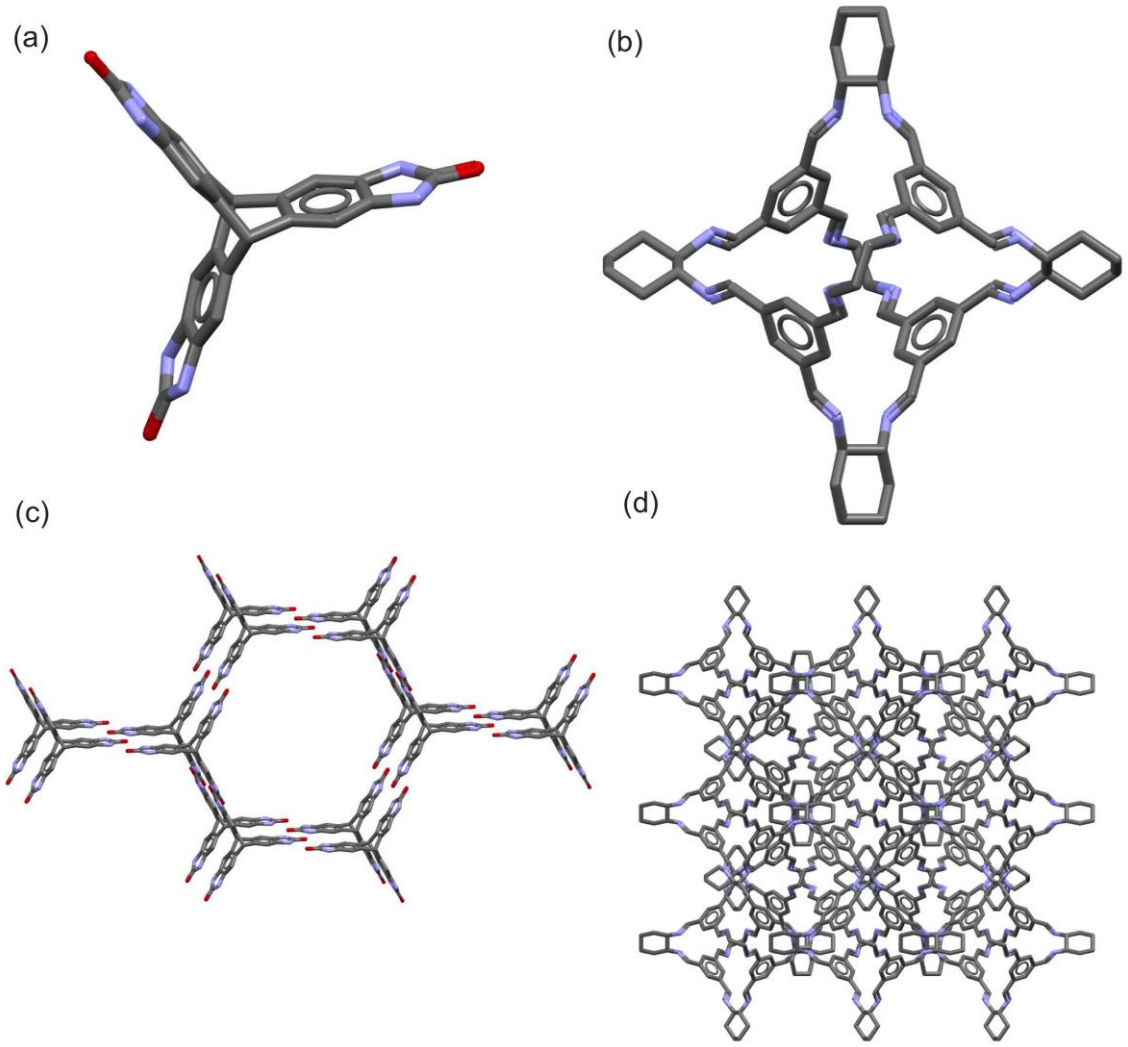


Figure 3-1. Two molecules that build up into porous molecular materials. (a) Trypticene OMIM molecule with extrinsic porosity[44]. (b) Covalent cage 3, cage molecule with intrinsic porosity[21]. (c) Crystal structure for tryticene OMIM presented in (a), showing porosity at material level. (d) Crystal structure for Covalent cage 3 presented in (b), showing porosity at material level.

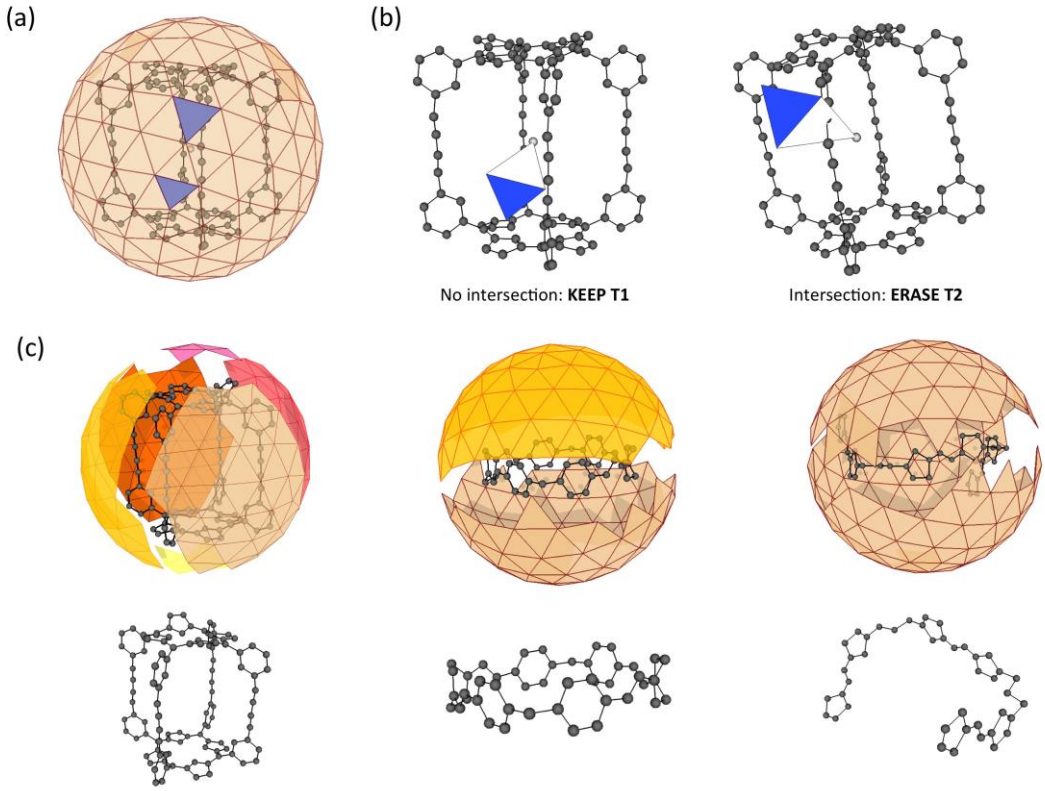


Figure 3-2. Point exposure map. (a) Triangular grid of a sphere (orange) with two triangles (blue) selected as examples for map generation. (b) Study of the two triangles: left, tetrahedron formed by reference point (grey dot) and the triangle T1 does not intersect the molecule, thus T1 is kept; rightm tetrahedron formed by the point and the triangle T2 does intersect the molecule, thus is erased. (c) Three examples of exposure map for the center of mass of three different molecules: left, a cage molecule; center, a belt molecule; right, a linear molecule. Left, map is formed by six connected components, each represented in a different color (orange, yellow, pink, light brown, light yellow, dark pink); middle, map is formed by two components, in two colors (yellow and light brown); right, map is formed by a single component (brown). Molecules PubChem[33] ID (left to right): 101377082, 235007 and 15505942.

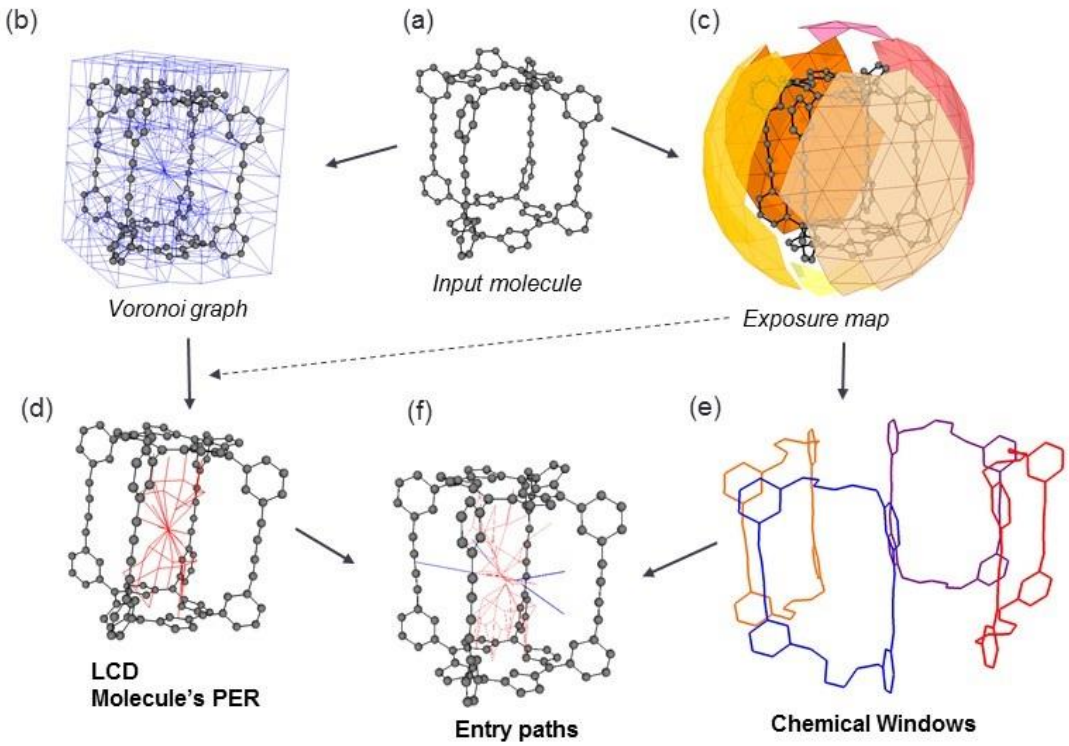


Figure 3-3. Molecular porosity analysis overview. Molecular porosity analysis to compute molecular pores and access routes can be sumarized in few steps (a) Example molecule (grey spheres as atoms, lines as bonds). PubChem[33] ID: 101377082 (b) Voronoi tessellation of the molecule is an intermediate step (blue lines) to obtain molecular pores and entry paths. (c) Exposure of map of the geometrical center of the molecule is an intermediate step to compute molecular windows. Colors as in Fig. 2c left (d) Combining PER and Voronoi graph, molecular cavities can be identified. This allows to compute LCD and molecular PER. In the figure, internal Voronoi nodes and edges are shown in red. (e) Chemical windows can be identified from exposure map of molecule's center. In the figure, each window is shown in a different color (red, orange blue and purple). (f) Combining information from Voronoi tessellation, internal cavities, and molecular windows, entry paths can be identified. In the figure, entry paths are shown as blue lines.

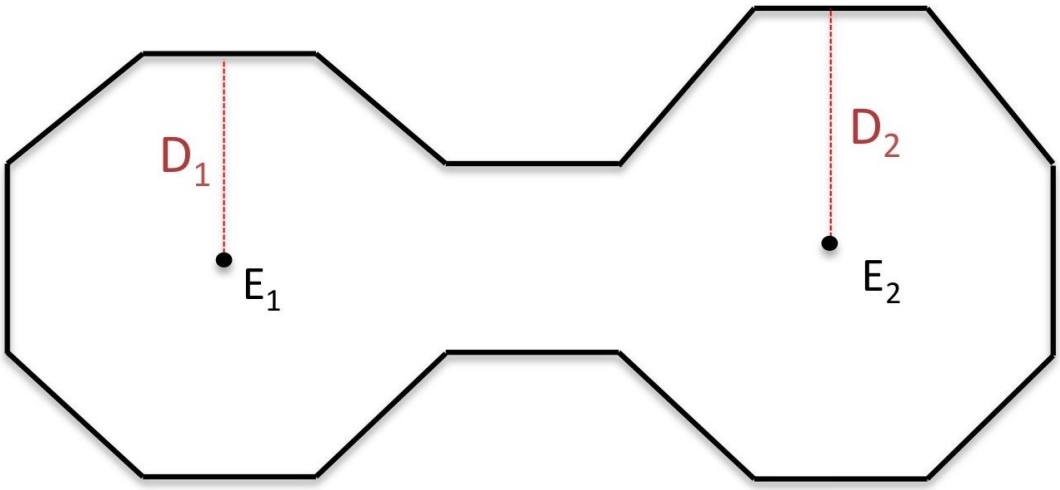


Figure 3-4. Schematic view of an irregular window. Window (bonds only) viewed from the top (black lines). Black dots represent entry paths for the irregular window. Two entry paths (E_1 and E_2) with different restriction distances (D_1 and D_2, in red) are available for this window.

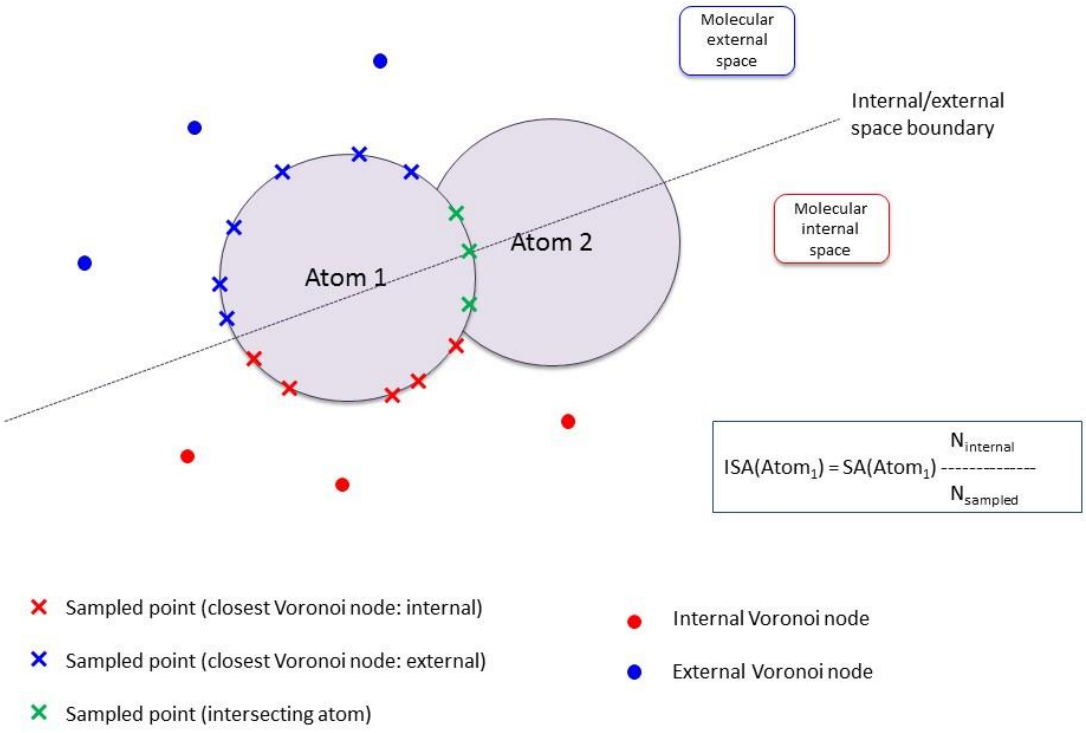


Figure 3-5. Internal surface area of an atom in a porous molecule. To compute internal surface area for one atom, random points on its surface are sampled. Each sampled point is accounted as internal if: 1) it's not placed inside another atom and 2) the closest Voronoi node from Voronoi graph is an internal node (according to Section 2.6 definition) – internal Voronoi nodes represented as red dot. Step 2 is taken instead of computing PER for each sampled point for efficiency reasons. In the figure, example random points are represented as crosses further hlighted by colors: green for those placed inside another atom (i.e., not fullfiling 1), red for internal points (i.e. fullfiling 1 and 2) and blue for external points (i.e. fulfilling 1 but not 2). Voronoi nodes are represented as circles: red (internal Voronoi nodes) and blue (external Voronoi nodes). The ratio of internal points over sampled points, multiplied by atom's surface area, provides the value of the internal surface area of the atom.

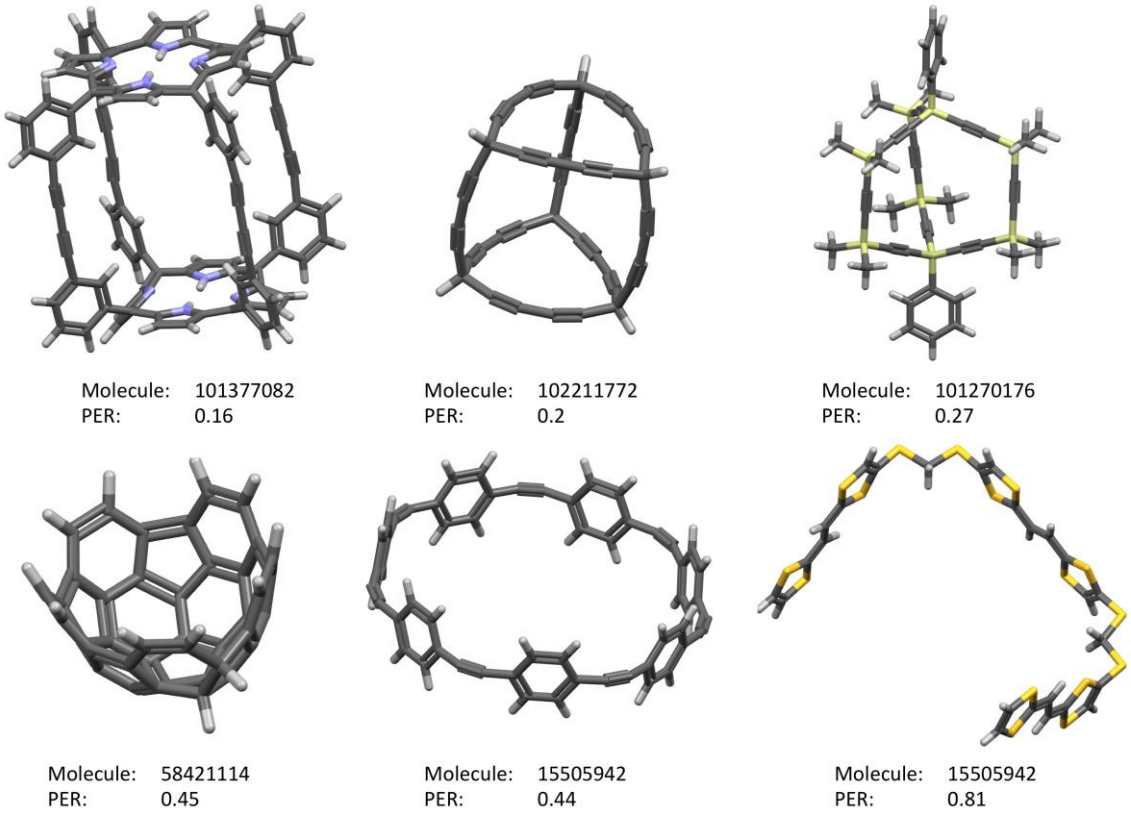


Figure 3-6. PER examples. Representative molecules are shown, together with their PubChem[33] IDs. Top three are cage molecules. Bottom molecules represent, from left-to-right, cup-like, belt-like and fully non-porous geometries. PER values are computed for the center of the box containing the molecule in the case of the non-porous molecule and for the largest pore in the case of the cage molecules.

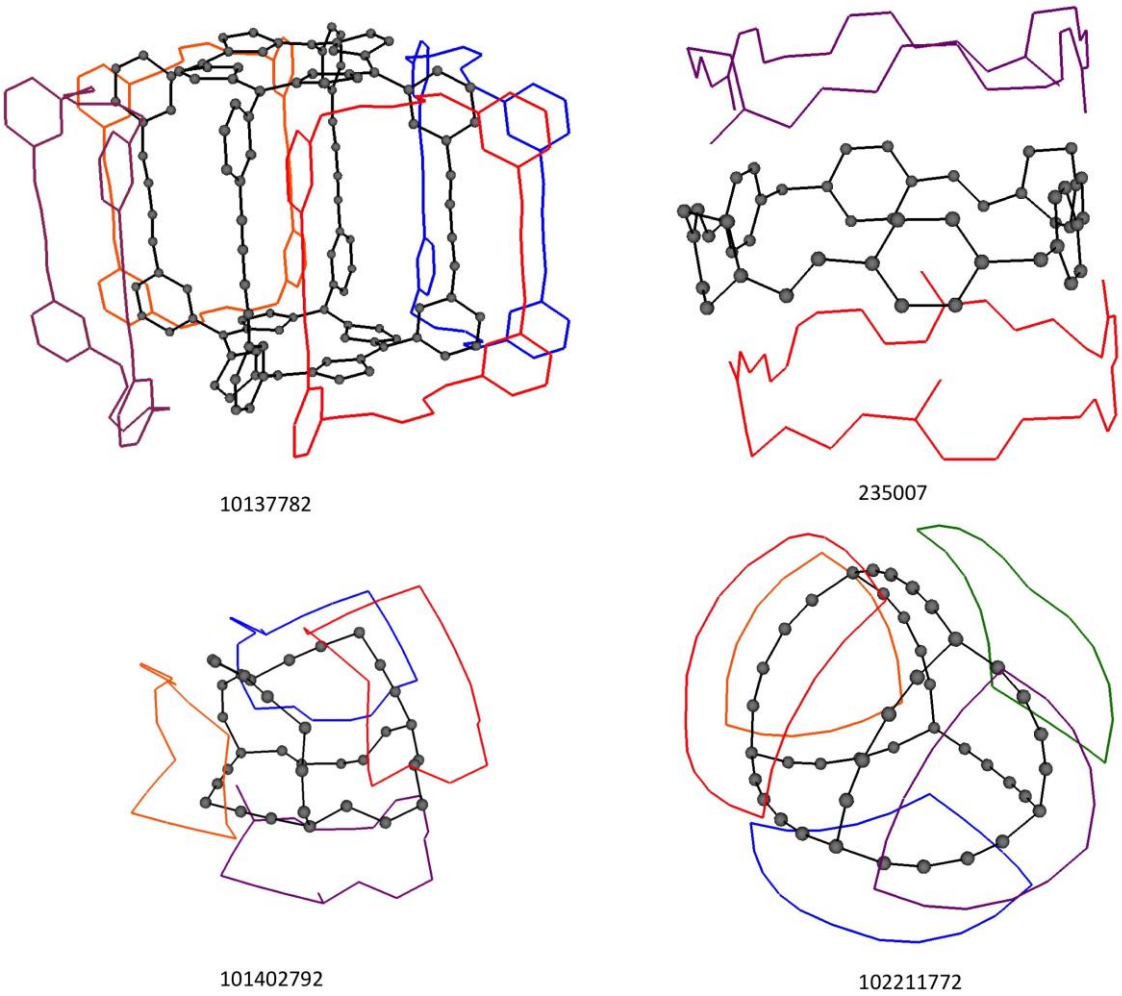


Figure 3-7. Chemical window computing for four example molecules. Each window is presented in a different color, as a set of bonds (straight lines), omitting atoms. PubChem CIDs are shown below each molecule.

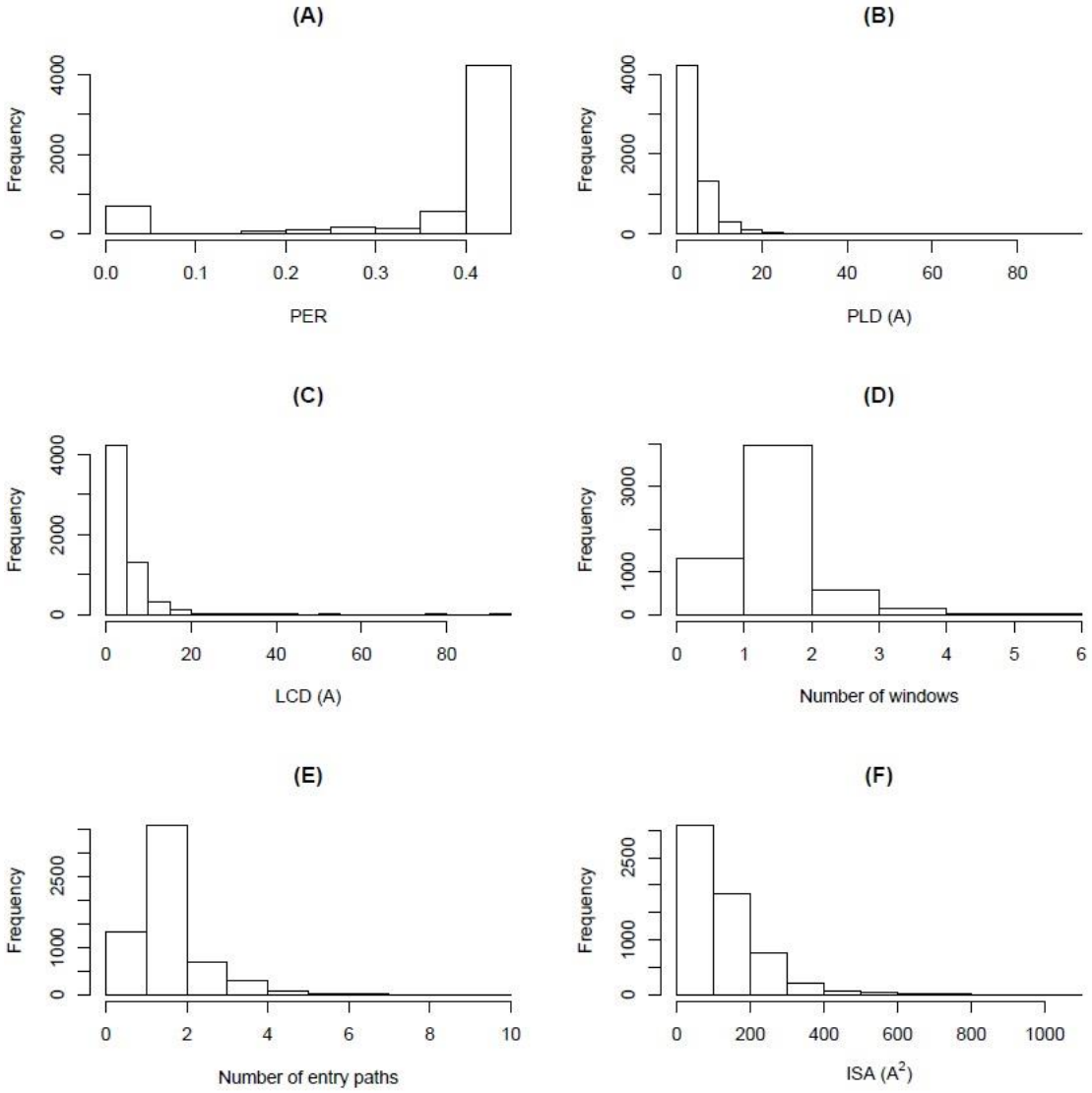


Figure 3-8. Number of appearances in the database. Graphics above show the number of molecules with PER < 0.45 placed in different ranges for the six parameters computed by the software, namely: (A) Pore exposure ratio; (B) Pore limiting diameter; (C) Largest cavity diameter; (D) Number of windows; (E) Number of entry paths; (F) Internal surface area.

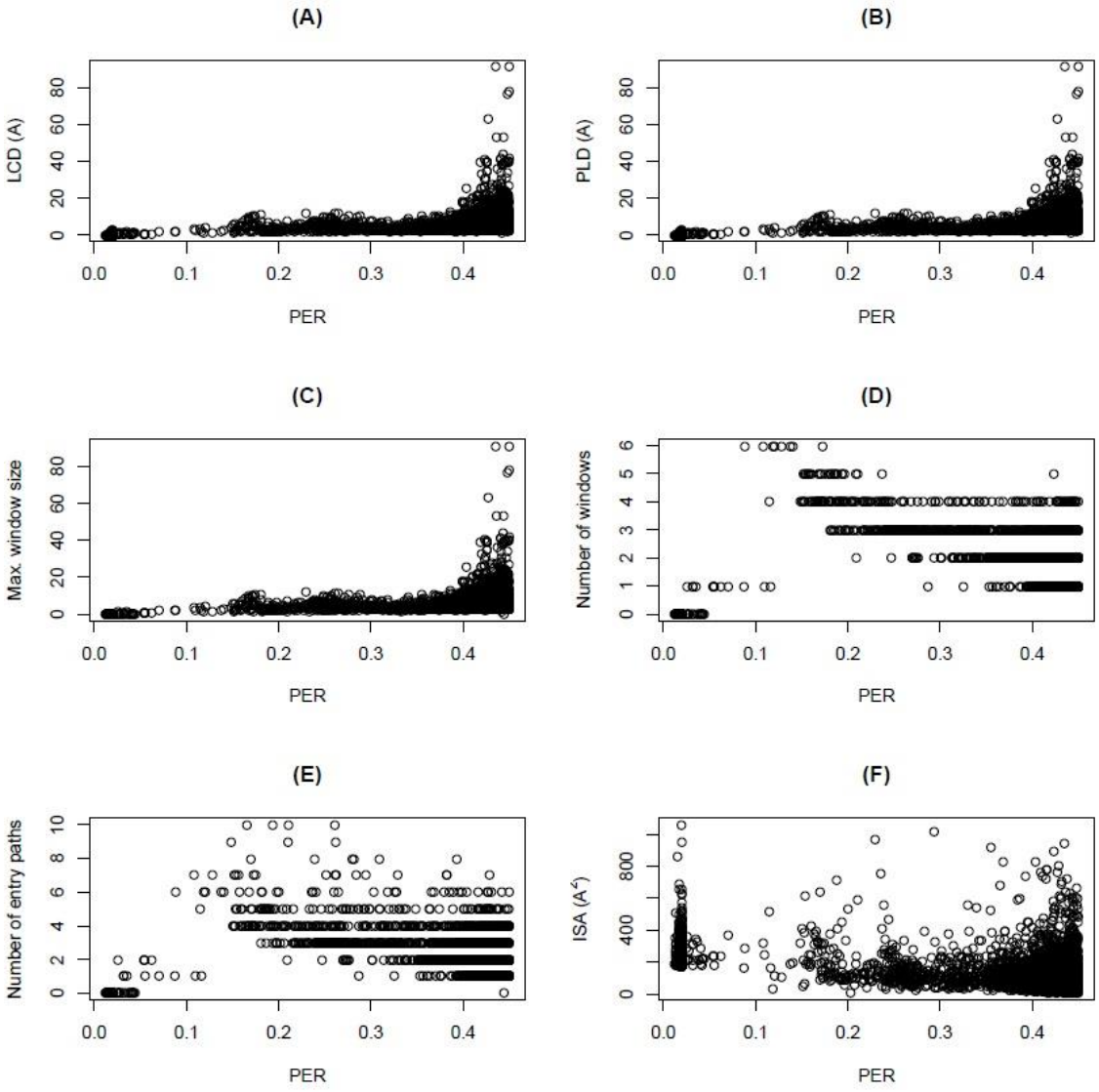


Figure 3-9. PER compared with molecular descriptors. Graphics above show a description of the population distributions for different PER values compared with LCD, PLD, number of windows and ISA. (A) PER vs largest cavity diameter (A); (B) PER vs pore limiting diameter (A); (C) PER vs maximum window size; (D) PER vs number of windows; (E) PER vs number of entries; (F) PER vs internal surface area (A^2).

Figure 3-10. Correlations among molecular descriptors Blue colors indicate positive correlation, red colors indicate negative correlation. Color intensity is associated with strength of correlation.

TABLES

Molecule	**LCD (A)**	**PLD (A)**	**PER**	**NW**	**NEP**	**ISA (A^2)**
10137782	9.06	5.64	0.16	4	4	45.51
101402792	3.04	2.54	0.26	4	5	15.25

102211772	5.34	4.02	0.2	5	5	17.61
70680111	5.84	4.04	0.17	4	4	24.63
235007	7.4	7.18	0.4	2	2	14.34

Table 3-1. Parameter values for 5 molecules from PubChem.

7. Notes and References

Corresponding Author

maciej.haranczyk@imdea.org

Author Contributions

The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

Acknowledgements

Authors acknowledge support from the Spanish Ministry of Economy and Competitiveness (RYC-2013-13949) and the resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Supporting Information

As an extension of the work presented in this document, a Supporting Information file has been provided. Supporting information file is divided in six sections, and contains information about complementary methods (sections 1-4), detailed pseudocodes for algorithms (section 5) and the set of molecules for methods validation (section 6).

References

(1) Sumida, K.; Rogow, D. L.; Mason, J. A.; McDonald, T. M.; Bloch, E. D.; Herm, Z. R.; Bae, T. H.; Long, J. R. Carbon Dioxide Capture in Metal-Organic Frameworks. *Chem. Rev.* **2012**, *112*, 724–781.

(2) Herm, Z. R.; Wiers, B. M.; Mason, J. A.; Van Baten, J. M.; Hudson, M. R.; Zajdel, P.; Brown, C. M.; Masciocchi, N.; Krishna, R.; Long, J. R. Separation of Hexane Isomers in a Metal-Organic Framework with Triangular Channels. *Science (80-.).* **2013**, *340*, 960–964.

(3) 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.; et al. The Materials Genome in Action: Identifying the Performance Limits for Methane Storage. *Energy Environ. Sci.* **2015**, *8*, 1190–1199.

(4) Thornton, A. W.; Simon, C. M.; Kim, J.; Kwon, O.; Deeg, K. S.; Konstas, K.; Pas, S. J.; Hill, M. R.; Winkler, D. A.; Haranczyk, M.; et al. Materials Genome in Action: Identifying the Performance Limits of Physical Hydrogen Storage. *Chem. Mater.* **2017**, *29*, 2844–2854.

(5) Kreno, L. E.; Leong, K.; Farha, O. K.; Allendorf, M.; Van Richard P., D.; Hupp, J. T. 2-40 Metal-Organic Framework Materials as Chemical Sensors. *Chem. Rev. (Washington, DC, United States)* **2012**, *112*, 1105–1125.

(6) Cooper, A. I. Porous Molecular Solids and Liquids. *ACS Cent. Sci.* **2017**, 3, 544-553.

(7) Evans, J. D.; Jelfs, K. E.; Day, G. M.; Doonan, C. J. Application of Computational Methods to the Design and Characterisation of Porous Molecular Materials. *Chem. Soc. Rev.* **2017**, *46*, 3286–3301.

(8) Day, G. M. Current Approaches to Predicting Molecular Organic Crystal Structures. *Crystallogr. Rev.* **2011**, *17*, 3–52.

(9) Price, S. L. Predicting Crystal Structures of Organic Compounds. *Chem. Soc. Rev.* **2014**, *43*, 2098–2111.

(10) Evans, J. D.; Huang, D. M.; Hill, M. R.; Sumby, C. J.; Sholl, D. S.; Thornton, A. W.;

Doonan, C. J. Molecular Design of Amorphous Porous Organic Cages for Enhanced Gas Storage. *J. Phys. Chem. C* **2015**, *119*, 7746–7754.

(11) Jain, A.; Ong, S. P.; Chen, W.; Medasani, B.; Qu, X.; Kocher, M.; Brafman, M.; Petretto, G.; Rignanese, G.-M.; Hautier, G.; et al. FireWorks: A Dynamic Workflow System Designed for High- Throughput Applications. *Concurr. Comput. Pract. Exp.* **2015**, *27*, 5037–5059.

(12) Matito-Martos, I.; Moghadam, P. Z.; Li, A.; Colombo, V.; Navarro, J. A. R.; Calero, S.; Fairen-Jimenez, D. Discovery of an Optimal Porous Crystalline Material for the Capture of Chemical Warfare Agents. *Chem. Mater.* **2018**, 30, 4571-4579.

(13) Berthold, M. R.; Cebron, N.; Dill, F.; Gabriel, T. R.; Kötter, T.; Meinl, T.; Ohl, P.; Thiel, K.; Wiswedel, B. KNIME - the Konstanz Information Miner. *ACM SIGKDD Explor. Newsl.* **2009**, *11*, 26.

(14) 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.

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

(16) Evans, J. D.; Huang, D. M.; Haranczyk, M.; Thornton, A. W.; Sumby, C. J.; Doonan, C. J. Computational Identification of Organic Porous Molecular Crystals. *CrystEngComm* **2016**, *18*, 4133–4141.

(17) Simon, C. M.; Mercado, R.; Schnell, S. K.; Smit, B.; Haranczyk, M. What Are the Best Materials to Separate a Xenon/Krypton Mixture? *Chem. Mater.* **2015**, *27*, 4459–4475.

(18) Wilmer, C. E.; Leaf, M.; Lee, C. Y.; Farha, O. K.; Hauser, B. G.; Hupp, J. T.; Snurr, R. Q. Large-Scale Screening of Hypothetical Metal-Organic Frameworks. *Nat. Chem.* **2012**, *4*, 83–89.

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

(20) Allcock, H. R.; Siegel, L. A. Phosphonitrilic Compounds . 111 Molecular Inclusion Compounds of Tris (O-pheny1enedioxy) Phosphonitrile Trimer. *J. Am. Chem. Soc.* **1964**, *80*, 5140–5144.

(21) Tozawa, T.; Jones, J. T. a; Swamy, S. I.; Jiang, S.; Adams, D. J.; Shakespeare, S.; Clowes, R.; Bradshaw, D.; Hasell, T.; Chong, S. Y.; et al. Porous Organic Cages. *Nat. Mater.* **2009**, *8*, 973–978.

(22) Ono, K.; Johmoto, K.; Yasuda, N.; Uekusa, H.; Fujii, S.; Kiguchi, M.; Iwasawa, N. Self-Assembly of Nanometer-Sized Boroxine Cages from Diboronic Acids. *J. Am. Chem. Soc.* **2015**, *137*, 7015–7018.

(23) Wang, Q.; Yu, C.; Long, H.; Du, Y.; Jin, Y.; Zhang, W. Solution-Phase Dynamic Assembly of Permanently Interlocked Aryleneethynylene Cages through Alkyne Metathesis. *Angew. Chemie - Int. Ed.* **2015**, *54*, 7550–7554.

(24) Zhang, G.; Presly, O.; White, F.; Oppel, I. M.; Mastalerz, M. A Permanent Mesoporous Organic Cage with an Exceptionally High Surface Area. *Angew. Chemie - Int. Ed.* **2014**, *53*, 1516–1520.

(25) 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.; et al. 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.

(26) Kudo, H.; Hayashi, R.; Mitani, K.; Yokozawa, T.; Kasuga, N. C.; Nishikubo, T. Molecular Waterwheel (Noria) from a Simple Condensation of Resorcinol and an Alkanedial. *Angew. Chemie - Int. Ed.* **2006**, *45*, 7948–7952.

(27) Fujita, M.; Yazaki, J.; Ogura, K. Preparation of a Macrocyclic Polynuclear Complex,[(en)Pd(4,4'-bpy)]4(N03)8,Which Recognizes an Organic Molecule in Aqueous Media. *J. Am. Chem. Soc.,* **1990**, *112*, 5646–5648.

(28) Stang, P. J. Molecular Architecture: Coordination as the Motif in the Rational Design and Assembly of Discrete Supramolecular Species - Self-Assembly of Metallacyclic Polygons and Polyhedra. *Chem. - A Eur. J.* **1998**, *4*, 19–27.

(29) Williams, M. E.; Hupp, J. T. Scanning Electrochemical Microscopy Assessment of Rates of Molecular Transport through Mesoporous Thin-Films of Porphyrinic "molecular Squares." *J. Phys. Chem. B* **2001**, *105*, 8944–8950.

(30) Thallapally, P. K.; Wirsig, T. B.; Barbour, L. J.; Atwood, J. L. Crystal Engineering of Nonporous Organic Solids for Methane Sorption. *Chem. Commun.* **2005**, No. 35, 4420–4422.

(31) 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.

(32) Gómez García, I.; Bernabei, M.; Pérez Soto, R.; Haranczyk, M. Out-of-Oblivion Cage Molecules and Their Porous Crystalline Phases. *Cryst. Growth Des.* **2017**, *17*, 5614–5619.

(33) Kim, S.; Thiessen, P. A.; Bolton, E. E.; Chen, J.; Fu, G.; Gindulyte, A.; Han, L.; He, J.; He, S.; Shoemaker, B. A.; et al. PubChem Substance and Compound Databases.

Nucleic Acids Res. **2016**, *44*, D1202–D1213.

(34) 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.

(35) Pinheiro, M.; Martin, R. L.; Rycroft, C. H.; Haranczyk, M. High Accuracy Geometric Analysis of Crystalline Porous Materials. *CrystEngComm* **2013**, *15*, 7531.

(36) Rycroft, C. H. VORO++: A Three-Dimensional Voronoi Cell Library in C++. **2009**, *19*, 1–16.

(37) Vogel, H. A Better Way to Construct the Sunflower Head. *Math. Biosci.* **1979**, *44*, 179–189.

(38) Dijkstra, E. W. A Note on Two Probles in Connexion with Graphs. *Numer. Math.* **1959**, *1*, 269–271.

(39) Https://www.eclipse.org/. Accessed Oct 1st 2018.

(40) Https://www.gnu.org/software/gdb/. Accessed Oct 1st 2018.

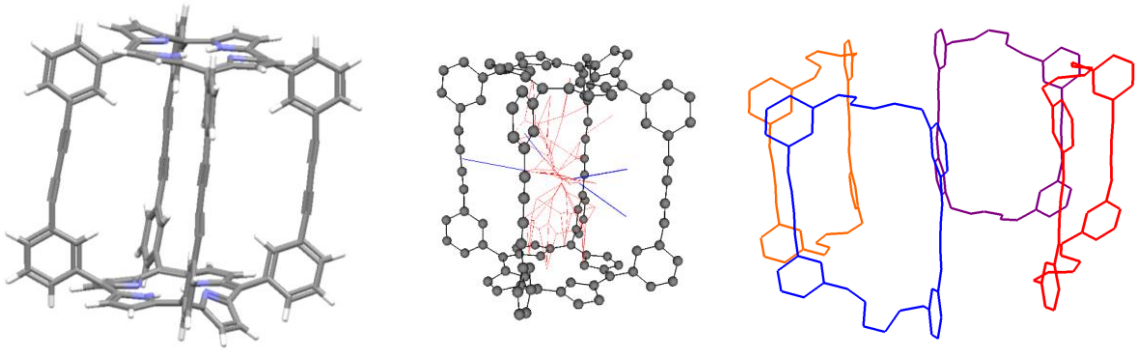
(41) Bondy, A.; M.R.Murty. Graph Theory. **2008**.

(42) Edelsbrunner, H.; Morozov, D. Persistent Homology: Theory and Practice. *6th Eur. Congr. Math.* **2012**, 123–142.

(43) Bernabei, M.; Pérez-Soto, R.; Gomez Garcia, I.; Haranczyk, M. In Silico Design and Assembly of Cage Molecules into Porous Molecular Materials. *Mol. Syst. Des. Eng.* **2018**, *3*, 942-950.

(44) Taylor, R. G. D.; Carta, M.; Bezzu, C. G.; Walker, J.; Msayib, K. J.; Kariuki, B. M.; McKeown, N. B. Triptycene-Based Organic Molecules of Intrinsic Microporosity. *Org. Lett.* **2014**, *16*, 1848–1851.

TOC Figure



8. Supporting information

8.1 Voronoi tessellation (detailed algorithm)

The algorithm for construction of Voronoi graph starts with a set of points S. For each point s in S, a Voronoi cell (see Fig. 3-S1, bottom right) is computed as the region of space that is closer to that point than to anyone else in S. That is:

$$x \text{ in Voro\_cell}(s) \text{ if } dist(x, s) \leq dist(x, s') \text{ for every } s' \text{ in } S\text{-}\{s\}$$

where dist(x, s) stands for the distance between point x and atom s. The points in the boundary of the Voronoi cell are those at the exact same distance to some neighbor, thus being part of another point cell (see Fig. 3-S1). This process constructs a graph in space that covers regions of maximal separation between atoms. Thus, is to be expected to represent voids between atoms, and as such also around and within the molecule.

To consider atom radii, we used radical Voronoi tessellation[1]. This technique works similar as Voronoi tessellation does, but it changes the distance function to add radii information. Atoms are assumed to be hard spheres with a given radius. When using radical Voronoi tessellation, cells are given by the equation:

$$\text{x in Voro\_cell(s) if } \text{dist}^2(\text{x, s})\text{-rad}^2(\text{s}) \leq \text{dist}^2(\text{x, s'})\text{-rad}^2(\text{s'}) \text{ for every s' in S-\{s\}}$$

Where $rad^2(s)$ stands for the square radius of atom s, and $dist^2(x, s)$ stands for the square of the distance from the atom s to the point x.

Several open source implementations of Voronoi tessellation are available and well described in the scientific literature [2,3]. In this work, we used Voro++ library.

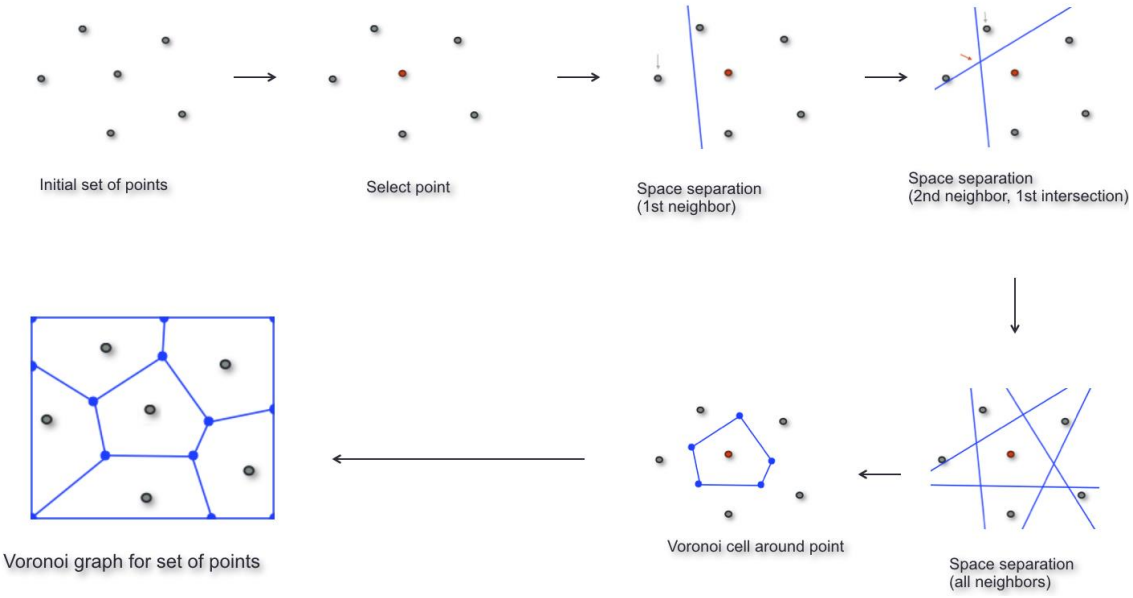


Figure 3-S1. Voronoi tessellation detail. Detail of the construction of a Voronoi cell (blue lines) around a given point (red dot). The diagram shows the entire process for a single point (top, bottom right), and then shows the result of constructing the full Voronoi diagram for a set of points (bottom left).

8.2 High accuracy in Voronoi tessellation

Radical Voronoi tessellation approximates Voronoi tessellation considering atom radii. Unfortunately, this approach is not exact, and may cause computed nodes to be out-of-place for a certain margin of error. In the case of materials, it has been observed that use of radical Voronoi tessellation may cause pores to be out of place for distances as big as 0.1 Å^4. To address this problem, "high accuracy Voronoi tessellation" can be applied instead of radical Voronoi tessellation. High accuracy tessellation consists on substituting each atom's rigid sphere by a fixed number of smaller spheres around its surface. All these spheres have the same radius. After this replacement, regular Voronoi tessellation is applied over the new set of points (made of the replacement spheres). Regular Voronoi tessellation is described in the previous section. In our work, we used N=50 spheres per atom

replacement. For efficiency reasons, high accuracy Voronoi tessellation should not always be applied (especially when exploring a big dataset). In this work, we used radical Voronoi tessellation to perform high throughput database analysis and high accuracy Voronoi tessellation for the studies of molecules shown in results section. In the software tool, both options are available.

8.3 Spherical grid construction and selection of number of triangles

Construction of spherical grid for point exposure map (see Section 2.3) is guided by two criteria: triangles should be as regular as possible, and their surfaces should be similar. To achieve this goal, a three-step algorithm is applied: 1) An icosahedron (20-sided polyhedron), centered at the reference point, is constructed; 2) each facet of the icosahedron is subdivided in a number of self-similar triangles; and 3) vertices of the self-similar triangles are projected over a sphere of the desired radius. Due to the nature of this particular triangulation, the spherical grid can only have a number of triangles N dependent on the number of self-similar divisions, k:

$$N(k) = 20 \cdot 4^k$$

Valid numbers of triangles include 80, 320, 1280 and 5120. Number of triangles influences the values computed from exposure map, in particular Pore Exposure Ratio. However, this number is bounded when $k \to \infty$. In Fig. 3-S2 we show how this number behaves for six example molecules, when grid triangles equal 80, 320, 1280 and 5120 respectively. Higher numbers of triangles forego computational capacities of regular computers due to memory issues and are not shown here. In this work, triangles were set at 320 to compute PER. This number is a compromise between accuracy at PER calculation and computation time.

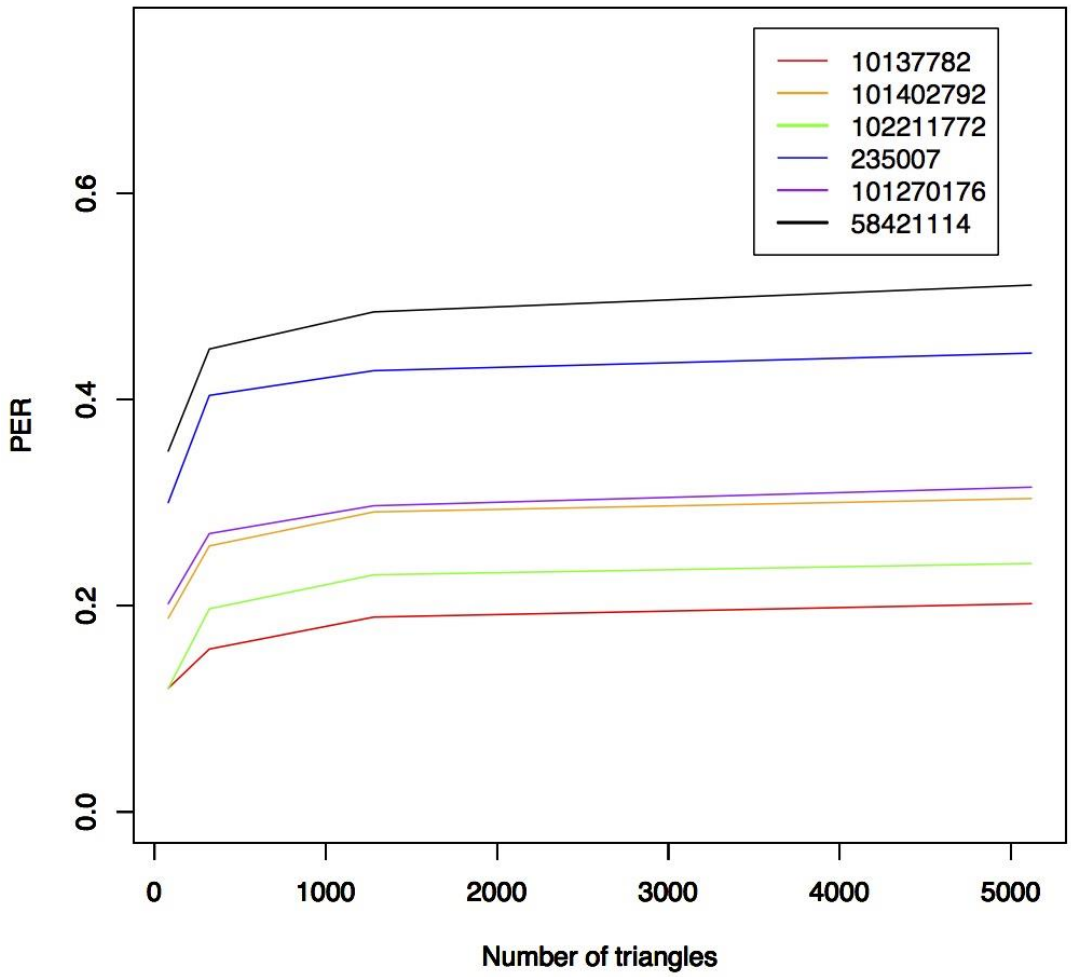


Figure 3-S2. PER values for largest molecular cavity (calculated at k=2, N=320 triangles) for different number of triangles in construction of point exposure map. Six different molecules are shown (CIDs from PubChem dataset used as molecular identifiers). PER values are typically underestimated with lower number of triangles. Small relative changes happen between the k=3 (N=1280 triangles) and k=4 (N=5120 triangles). This sort of convergence-like behavior indicates stability of PER at high number of triangles.

8.4 Modified Dijkstra algorithm

Our modified Dijkstra algorithm works in a similar manner as the classical Dijkstra algorithm[5]. It takes a given internal node as a starting point, and finds the sequence of nodes and edges that a probe would have to travel through to move from that node to the outside of the molecule. To do so: First, it assigns a weight equal to zero to every node, except for the target node, that gets a weight equal to its radius; second, the initial node is visited. Visiting the node implies exploring its neighbors, marking the node as visited, and assigning it a weight that is computed as the maximum value among the access routes to the node, or the size of the node (if all the routes are bigger). This process is repeated for every neighbor, prioritizing those that provide routes with larger openings. As a result, a list of nodes and a given weight are provided. The weight corresponds to the PLD value for that pore. To compute molecule's PLD, this algorithm is applied to all internal Voronoi nodes, and the maximum value obtained is kept.

8.5 Schemes and baseline functions.

In this section, we present algorithms in a pseudocode manner (see Schemes 1-7).

Scheme 1. Pore Exposure Ratio algorithm

```
PoreExposureRatio(Point)
        G = spheric_grid_around(Molecule, Point)
        GridSurface = surface(G)
        Subgrid = erase_triangles(Molecule, Point, G)
        CCs = connected_components(Shadow)
        for (i in 1 to number(CCs))
        SurfacesList[i] = surface(CCs[i])
        end for
        SurfaceMax = max(SurfacesList)
        Rate = SurfaceMax/GridSurface
```

Scheme 2. Cavity detection algorithm

```
V = voro_tessellation(Molecule)
for (v_node in V)
        PER[v_node] = pore_exposure_ratio(v_node)
        if (PER[v_node] < PER_Threshold)
                v_node.type = internal
        else
                v_node.type = external
        end if
end for
```

Scheme 3. LCD and molecular PER computing algorithm

```
GetInternalPore(VoroGraph)
        InternalNodes = internal_nodes(VoroGraph)
        N = node_with_max_radius(InternalNodes)
        MoleculePoreSize = N.radius
        MolecularPER = N.PER
        return(PoreSize, MoleculePER)
```

Scheme 4. Chemical window detection algorithm

```
ChemicalWindowDetection(Molecule)
        Point = get_center(Molecule)
        G = spheric_grid_around(Molecule, Point)
        Subgrid = erase_triangles(Molecule, Point, G)
        CCs = connected_components(Shadow)
        for (CC in CCs)
                B = get_closest_bonds(CC, Molecule, Point)
                Win = window_reconstruct(B)
```

```
            Molecule.insert_window(Win)
        end for
```

Scheme 5. Entry path computing algorithm

```
EntryPathDetection(Molecule, VoroGraph, CCs)
        EC = entry_candidates(VoroGraph)
        for (Window in Molecule.windows())
            EC_Win = get_window_entries(Window, EC, CCs)
            EC_Win = cluster_entries(EC_Win)
            Window.insert_entries(EC_Win)
        end for
```

Scheme 6. Pore limiting diameter algorithm

```
PoreLimitingDiameter(VoroGraph, Pore)
        PQ = priority_queue()
        Visited = visited_list(VoroGraph.nodes().external())
        W = assign_weights(W, VoroGraph.nodes()-Pore, 0)
        W = assign_weights(W, Pore, Pore.radius())
        visit_node(Pore, PQ, W, Visited)
        while(!empty(PQ))
            NextP = p_with_max_weight(W, Visited)
            PQ = extract_from_pq(PQ, NextP)
            visit_node(NextP, PQ, W, Visited)
        end while
        return(Maximum(W))
```

Scheme 7. Internal surface area algorithm

```
InternalSurfaceArea(Molecule, NP)
        Surface = 0
        for (Atom in Molecule)
```

```
        R = Atom.radius()
        Center = Atom.center()
        AtomSurf = 2*PI*R^2
        Points = sample_points(R, Center, NP)
        Points = PER_classify(Points)
        RatioInternal = ratio_of_internal(Points)
        Surface = Surface + AtomSurf*RatioInternal
    end for
    return Surface
```

Algorithms depend on a set of baseline functions, described also in this section. These functions include computational geometry, graph and topological techniques that allow performing the analysis described above. In this section, we briefly describe such functions to provide a better understanding on them:

- spheric_grid(Molecule, Point): Constructs a spherical grid around the point and the molecule. The grid is centered at the point, and has radius equal to 1.5 times the maximum of the distances from the point to the atoms. It is guaranteed that the spherical grid will fully cover the molecule. The grid is constructed in such a way that triangles in the surface are as regular as possible. This is done using the Vogel's method[34].
- surface(Grid): Computes the surface of a set of triangles, by computing each triangle surface individually and then summing them all.
- erase_triangles(Molecule, Point, G): Computes subgrid for the given point and molecule, erasing triangles from the grid surface if there is a bond in between them and the reference point.

- connected_components(Grid): For a given set of triangles in space, computes the set of connected components of those triangles. Two triangles are considered to be connected if they share an edge. Result is returned in form of a linked list.
- get_center(Molecule): Computes the center of a box containing the molecule.
- get_closest_bonds(CC, Molecule, Point): Returns those bonds that are closer to the connected component than to the point given as reference.
- window_reconstruct(B): Reconstructs the window, adding potentially missing bonds by checking if they connect two atoms already present in the window.
- entry_candidates(VoroGraph): Collects all the edges from Voronoi graph that connect nodes classified as internal with nodes classified as external.
- get_window_entries(Window, EC, CCs): Associates entry candidates with their window, using the connected component associated to that window. The entry edge is linked to the window with the closest connected component to edge's mid point.
- cluster_entries(EC_Win): Clusterizes edges associated with the given window. Algorithm described in more detail below.
- priority_queue(): Creates an empty priority queue, where the nodes to be visited are inserted in a way that their weight is taken into account, so the best candidate is popped first.
- visited_list(Points): Creates a list of visited nodes. Nodes in this list will act as search limits (they will be taken into account, but search won't continue from them).
- assign_weights(W, Points, Radius): Creates or modifies a weights list, by adding elements to it and assigning the given weight to all those elements.

- visit_node(Pore, PQ, W, Visited): The core function in this algorithm. Visits a node, exploring all its neighbors and assigning a new weight to it. The way this weight is assigned is explained below. This function also adds the node to the list of visited nodes.
- p_with_max_weight(W, Visited): Returns the point with maximum weight from the list of weights, and removes that value from it.
- extract_from_pq(PQ, NextP): Extracts a given point from priority queue.
- sample_points(R, Center, NP): Samples NP random points around the atom's center, at distance R. These points are sampled in such a way that they get evenly distributed around the surface.
- PER_classify(Points): Classifies the list of points given as an argument as internal or external using the PER method.
- ratio_of_internal(Points): Returns the ratio of internal by total number of points, as an estimate of the percentage of the atom's surface to be accounted as internal.

8.6 Set of molecules for validation

The set of molecules for validation of the tool are extracted from both PubChem and Cambridge Structural Database. Identifiers for these molecules are provided in table 3-S1.

Table 3-S1. Identifiers for PubChem and CSD molecules utilized as validation set for the tool.

PubChem	CSD
015505942 044224624 058350525 10008508	CSD_ABINOP CSD_ACDMFM
100988307 100997889 101011409 101011411	CSD_ACEZEP CSD_ADOPUG
101017099 101025558 101031605 101032051	CSD_AFUVAZ CSD_ATATIZ
101053160 101079888 101124917 101161596	CSD_BAGTAG CSD_BALNIM
101171364 101199196 101243160 101249402	CSD_BATVUP CSD_BIJYUP
101249403 101257027 101270176 101272100	CSD_CECDAR CSD_CIYVOW
101358818 101369893 101377081 101377082	CSD_COFKIS CSD_DIHGOR
101402791 101402792 101402793 101408774	CSD_DUCMUL CSD_EPIRUR
101408775 101415269 101429981 101457313	CSD_ERUFAY CSD_FAQWOJ
101465341 101504996 101510168 101569102	CSD_FEQXAC CSD_FIFTAR

101569103 101569105 101569107 101570056 101580482 101771978 101838948 101843402 101894046 101929323 101949060 101949067 101964660 102030692 102047080 102047081 102151215 102168016 102177430 102188020 102188100 102211772 102211773 102217538 102218044 102221570 102221571 102284634 102319015 102374965 102430937 102444682 102503341 102580403 10508428 10510568 10604658 10671385 10675159 10897925 11018236 11072634 11123511 11125941 11181815 11192627 11193642 11238464 11262018 11354378 11377685 11408158 11455902 1152252 11635702 11786744 117959245 11966537 11981753 1401986 14342511 1440050 1627906 162966 1727514 19010533 1936358 194283 21204034 2245469 2309856 235006 235007 2377096 24879595 25139831 262197 2628943 2751912 2847949 3015217 3127689 3133374 316664 3178126 320089 3206470 3212855 325973 3288857 3302848 3321396 335130 3355454 3363079 336926 3397505 3399625 3430237 3440068 3447911 3448986 3463783 3463811 3471501 3475323 3539556 3552088 3607042 3616454 3633032 3633034 3644803 3649866 3656660 3667731 3702418 3708959 3714638 3726133 3751837 3757803 3771103 3784932 3785796 388227 393564 394061 3941478 3985941 4033423 403709 4046549 4088964 4091745 4098246 4106264 4117009 4117256 4119225 4132867 4137416 4162866 4177481 4185072 4196425 4208441 4218223 4221128 4232100 4243367 4256554 4262335 4280375 4282568 4289933 4290317 4291730 4361048 4363723 437681 4383060 4385097 4410067 4412015 4433099 4441323 44516222 44534936 44534938 4457151 44598704 4486432 4575069 4586720 4587761 4589286 4599659 4623661 4624122 4625518 4681551 4681928 4686419 4688503 4691007 4703365 471312 4749974 4749975 4773429 4861346 4868273 4872414 489601 4921811 4960924 5017719 50229583 5041139 50416199 50417518 5046678 5048702 5054207 5055060 5055061 5058767 5079641 5087983 5092887 50940581 5121841 5190517 5192190 5223577 5226940 5234922 5251414 53309855 53392482 53392483 542442 542566 57587102 57731024 57878841 58224486 58224491 58421114 58843877 59248899 60055778 635857 635933 636323 640060 653801 6696405 69868224 71573529 72423321 85871078 87120 88543151 88963072	CSD_FIFTEV CSD_FIFTIZ CSD_FIFTOF CSD_FOMLUQ CSD_FOQTEM CSD_FOQTOW CSD_FUYHEN CSD_FUYHIR CSD_GANDAC CSD_GANDUW CSD_GANKIR CSD_GUMCIB CSD_LUXVAB CSD_MAVVAI CSD_MESTUA CSD_NOVNAP CSD_NUNRIX CSD_NUXHIZ CSD_OJITOR CSD_OZECAY03 CSD_PAQFES CSD_PUDWUH CSD_PUDXAO CSD_PUDXES CSD_REQXES CSD_SATJAA CSD_SATJEE CSD_UTEVOF CSD_VOLZON CSD_VOMPAQ

References

1. 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.* **149,** 134–141 (2012).

2. Rycroft, C. H. VORO++: A three-dimensional Voronoi cell library in C++. **19,** 1–16 (2009).

3. Jamin, C., Pion, S. & Teillaud, M. 3D Triangulation Data Structure, in: CGAL User and Reference Manual, 4.6.1 ed. *CGAL Editor. Board.* 2015 (2015).

4. Pinheiro, M., Martin, R. L., Rycroft, C. H. & Haranczyk, M. High accuracy geometric analysis of crystalline porous materials. *CrystEngComm* **15,** 7531 (2013).

5. Dijkstra, E. W. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* **1,** 269–271 (1959).