

PoreMatMod.jl: Julia package for *in silico* post-synthetic modification of crystal structure models

E. Adrian Henle¹, Nickolas Gantzler², Praveen K. Thallapally³, Xiaoli Z. Fern⁴, and Cory M. Simon^{1*}

¹School of Chemical, Biological, and Environmental Engineering. Oregon State University. Corvallis, OR. USA.

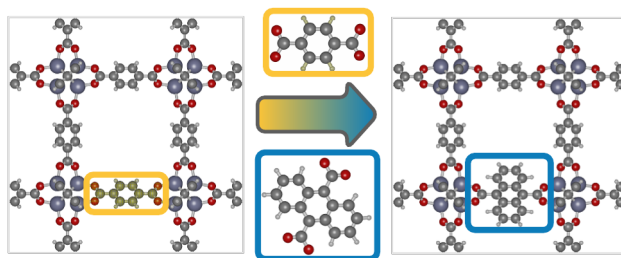
²Department of Physics. Oregon State University. Corvallis, OR. USA.

³Pacific Northwest National Lab. Richland, WA. USA.

⁴School of Electrical Engineering and Computer Science. Oregon State University. Corvallis, OR. USA.

*Cory.Simon@oregonstate.edu

October 6, 2021



Abstract

PoreMatMod.jl is a free, open-source, user-friendly, and documented Julia package for modifying crystal structure models of porous materials such as metal-organic frameworks (MOFs). PoreMatMod.jl functions as a find-and-replace algorithm on crystal structures by leveraging (i) Ullmann's algorithm to search for subgraphs of the crystal structure graph that are isomorphic to the graph of a query fragment and (ii) the orthogonal Procrustes algorithm to align a replacement fragment with a targeted substructure of the crystal structure for installation. The prominent application of PoreMatMod.jl is to generate libraries of hypothetical structures for virtual screenings via molecular simulations. For example, one can install functional groups on the linkers of a parent MOF, mimicking post-synthetic modification. Other applications of PoreMatMod.jl to modify crystal structure models include introducing defects and correcting artifacts of X-ray structure determination (adding missing hydrogen atoms, resolving disorder, and removing guest molecules).

Keywords: porous materials, crystal structure models, high-throughput computational screening, substructure search

1 Introduction

Nanoporous materials have adsorption-based applications for the separation [1–4], storage [5, 6], detection [7, 8], and production [9–11] of chemicals. Advanced classes of nanoporous materials, such as metal-organic frameworks (MOFs) [12], offer highly tunable pore environments for guest adsorbates. A critical task is to determine the chemical structure that endows the material with an optimal property.

The adjustability of MOFs largely stems from their modular synthesis: metal atoms or metal-oxo clusters [“secondary building units” (SBUs)] [13] coordinate to organic linker molecules to form an extended network. The abundance of compatible metals/SBUs and linkers, assembling within different topologies, gives many possible structures. Postsynthetic modification (PSM) [14], building block replacement (BBR) [15, 16], and mixing linkers [17, 18] and/or metals [19] grant further structural and chemical tunability. E.g., in covalent PSM, a reaction on the MOF can install a new chemical functionality on its linkers.

Among the practically unbounded number of possible structures, high-throughput computational screenings can accelerate the discovery of novel porous materials with optimal properties for a given application. By (i) constructing a library of atomistic models of material structures, then (ii) using molecular models and simulations to predict the property of each structure in the library, we can guide experimental campaigns by (a) providing a ranked list of materials and (b) elucidating structure-property relationships [20–23].

The computation-aided discovery of novel materials demands methods to construct predicted crystal structure models of hypothetical porous materials. Several algorithms to generate structure models of porous materials [24–34] mimic *de novo* synthesis by computationally stitching together/ assembling molecular building blocks into a topological network/template. Molassembler [35] generates molecules from graph representations and allows modification of them, but does not pertain to crystalline systems.

In this work, we present PoreMatMod.jl, a new, free, open-source, tested, and documented software package, written in the Julia programming language [36], to modify crystal structure models of porous materials. In contrast to prior structure generation algorithms [26–34] that mimic *de novo* synthesis, PoreMatMod.jl produces hypothetical structures by modifying existing crystal structure models, mimicking PSM or BBR. PoreMatMod.jl functions, effectively, as a find-and-replace algorithm on labeled (by the chemical species) graph and point cloud representations of porous crystals. The user specifies a parent crystal structure, a query molecular fragment/moiety, and a replacement molecular fragment/moiety. PoreMatMod.jl then (1) applies Ullman’s algorithm [37] to search for all subgraphs of the parent graph that are isomorphic to the query graph [38], (2) uses orthogonal Procrustes [39] to align the replacement fragment on the matching substructure of the parent structure, then (3) removes the query fragment and installs the replacement fragment in an optimal geometry, giving the child structure. When multiple subgraphs of the parent

are isomorphic to the query, `PoreMatMod.jl` grants precise control over the distribution of the replacements. We demonstrate that `PoreMatMod.jl` is useful for:

- tuning the chemistry of existing crystal structure models of porous materials (e.g., curated databases of experimentally-reported structures in Refs. [40–46]) to generate a library of hypothetical materials for computational screening (e.g., [47–50])
- generating heterogeneous, multi-linker MOFs with precise control of functional group placement (e.g., [51])
- repairing artifacts in crystal structures determined from X-ray diffraction (XRD) patterns, such as missing hydrogen atoms, disorder, and the presence of solvents (e.g., [41])
- introducing missing-linker and missing-SBU defects into MOFs to enable computational studies on the influence of such defects on properties (e.g., [52–54])
- searching for subgraphs in libraries of crystal structure models to e.g., filter structures or characterize chemical diversity (e.g., [55]).

See github.com/SimonEnsemble/PoreMatMod.jl for the source code and documentation for `PoreMatMod.jl`. Registered as an official Julia package under an MIT license, `PoreMatMod.jl` is easy to install and runs on Windows, Mac, and Linux operating systems.

2 Overview of the find-and-replace task in `PoreMatMod.jl`

Given a parent crystal structure, we wish to (1, find) search for all query fragments in the parent structure, then (2, replace) replace specified instances of the query fragments with the replacement fragments to produce a child crystal structure. For example, suppose we wish to replace all 1,4-benzodicarboxylate (BDC) linkers in IRMOF-1 with trifluoromethyl-BDC, mimicking BBR. The `PoreMatMod.jl` code:

```
# read crystal structure of the parent MOF
parent_xtal = Crystal("IRMOF-1.cif")

# read query fragment. masked atoms, to be replaced, marked with !
query_fragment = moiety("p-phenylene.xyz")
# read replacement fragment
replacement_fragment = moiety("tfm-p-phenylene.xyz")

# (1) search parent structure for query fragment
# (2) replace occurrences of query fragment with replacement fragments
# (with randomly chosen orientations)
child_xtal = replace(parent_xtal, query_fragment => replacement_fragment)
```

produces a crystal structure model of a hypothetical, functionalized IRMOF-1 child shown in Fig. 1.

Below, we explain the implementation of `PoreMatMod.jl` and illustrate more use cases. The source code, links to detailed documentation, and code for all use cases illustrated below are hosted at github.com/SimonEnsemble/PoreMatMod.jl.

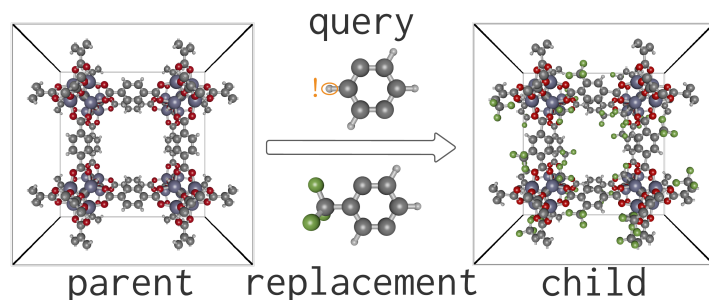


Figure 1: Overview of find-and-replace operations in PoreMatMod.jl to, for example, construct a hypothetical MOF. Here, we replace *p*-phenylene query fragments in the IRMOF-1 parent with trifluoromethyl-*p*-phenylene replacement fragments (in random orientations) to give a functionalized child structure. The masked atom of the query fragment to be replaced is annotated with !.

3 The inner-workings of PoreMatMod.jl

PoreMatMod.jl functions as a find-and-replace tool for crystal structures.

3.1 Representing crystal structures and chemical fragments

During the “find” stage, PoreMatMod.jl works with node-labeled, simple graph representations (nodes: atoms, edges: bonds, labels: chemical species) of crystal structures and chemical fragments. Edges for (periodic) crystal structures also join atoms bonded across the unit cell boundary. During the “replace” stage, PoreMatMod.jl works with labeled point cloud representations, where each atom is represented by a point in 3D space, labeled by the chemical species. In the case of crystal structures, periodic boundary conditions are imposed over the 3D space composed of the unit cell. The 3D coordinates are needed to align and install the replacement fragment onto the parent structure in a reasonable geometry.

By default, we infer the graph representation of crystal structures and chemical fragments from their labeled point cloud representations provided in a crystallographic information file (.cif) and XYZ file (.xyz), respectively. We assign a bond between a pair of atoms if the (periodic, in the case of crystal structures) distance between them is less than the sum of their covalent radii taken from Refs. [56, 57]. PoreMatMod.jl allows (i) alteration of the bonding rules and/or (ii) manual bond assignment.

N.b., one must construct the query and replacement chemical fragments for the find-and-replace task at hand by (i) cutting them from the parent structure and/or (ii) building them in an atom editor (e.g., Avogadro [58]).

3.2 “Find”: search for subgraphs of the parent graph that match the query graph

The goal of the “find” stage is to search for all subgraphs of the labeled graph representing the parent crystal structure that are isomorphic to the labeled graph representing the query fragment. Colloquially, two graphs are [exactly] *isomorphic* if they have the same number of nodes (atoms) and can be overlaid with each other to “match” in terms of both the connectivity of the nodes (bonding pattern) and the node labels (atomic species).

To precisely describe the [exact] subgraph isomorphism problem [38], let $G_p = (\mathcal{V}_p, \mathcal{E}_p)$ and $G_q = (\mathcal{V}_q, \mathcal{E}_q)$ be the graph representations of the parent crystal structure and query fragment, respectively, with \mathcal{V} and \mathcal{E} the set of nodes and edges, respectively. Let $s : \mathcal{V}_p \cup \mathcal{V}_q \rightarrow \{\text{H, He, Li, Be, ... , No, Lr}\}$ be the node-labeling function that maps a node to the chemical species it represents. A subgraph $G'_p = (\mathcal{V}'_p, \mathcal{E}'_p)$ of the parent graph G_p ($\text{subgraph} \implies \mathcal{V}'_p \subseteq \mathcal{V}_p$ and $\mathcal{E}'_p \subseteq \mathcal{E}_p$) is *isomorphic* to the query graph G_q if there exists a bijection $\theta : \mathcal{V}'_p \rightarrow \mathcal{V}_q$ such that $\{v'_{p,i}, v'_{p,j}\} \in \mathcal{E}'_p \iff \{\theta(v'_{p,i}), \theta(v'_{p,j})\} \in \mathcal{E}_q$ and $s(v'_p) = s(\theta(v'_p))$ for all $v'_p \in \mathcal{V}'_p$. The subgraph isomorphism problem is to find all subgraphs G'_p of G_p isomorphic to G_q .

PoreMatMod.jl searches for all subgraphs G'_p of the parent crystal structure graph G_p that are isomorphic to the query graph G_q using Ullmann’s subgraph isomorphism algorithm [37]. The set of subgraph isomorphisms often includes *symmetry-equivalent* subsets of bijections with the same domain G'_p . Each bijection in a symmetry-equivalent set maps the nodes in \mathcal{V}'_p to the nodes in \mathcal{V}_q via a different permutation. E.g., there are four symmetry-equivalent isomorphisms between a subgraph of the BDC linker and the *p*-phenylene fragment (see Fig. 2a). PoreMatMod.jl groups the search results—the set of isomorphisms between subgraphs $\{G'_{p,i}\}$ and G_q —by symmetry-equivalence.

3.3 “Replace”: align and install a replacement fragment on a substructure of the parent crystal structure

The goal in the “replace” stage is to align a replacement fragment with a targeted substructure of the parent crystal structure matching the query fragment, remove this substructure, then install the replacement fragment on the parent in its place to give the child crystal structure (e.g., in Fig. 1).

At this juncture, we have identified a subgraph G'_p of the graph of the parent crystal structure that is isomorphic to the graph of the query G_q via the bijection $\theta : \mathcal{V}'_p \rightarrow \mathcal{V}_q$. Now, to translate and align the replacement fragment onto this substructure of the parent crystal, we must find an injective mapping from a subset of the nodes on the graph of the replacement fragment G_r to corresponding nodes of the parent subgraph G'_p .

First, a subset of the atoms of the query fragment must be (manually) flagged as *masked* atoms in the XYZ input file by appending an exclamation mark ! to their atomic species labels (e.g., $\text{C} \rightarrow \text{C}!$ for a carbon atom). A masked atom in the query fragment implies that the corresponding atom of the parent crystal structure (i) must be removed [e.g., to make room for replacement with a different functionality] but (ii) does not correspond with an atom on the replacement fragment and thus

cannot be used in the process of aligning the replacement fragment onto the parent crystal. E.g., in Fig. 1, the H atom of the *p*-phenylene is a masked atom. The atom property viewer in iRASPA [59] facilitates finding which atom(s) in the XYZ file to annotate with ! to label as masked. Let $\mathcal{V}_{q,!} \subseteq \mathcal{V}_q$ denote the subset of nodes of the query graph that are masked.

Then, `PoreMatMod.jl` automatically searches for each subgraph G'_r of the replacement graph that is isomorphic to the induced subgraph of the query graph containing the non-masked nodes $\mathcal{V}_q \setminus \mathcal{V}_{q,!}$. Each is a bijection $\phi : \mathcal{V}'_r \rightarrow \mathcal{V}_q \setminus \mathcal{V}_{q,!}$ indicating correspondence between a subset of the atoms on the replacement fragment and the non-masked atoms of the query fragment. Finally, the composition of the bijections $v'_p = \theta^{-1}(\phi(v'_r))$ (with \cdot^{-1} indicating the inverse of a mapping) gives the node of the parent graph $v'_p \in \mathcal{V}'_p$ that corresponds with node $v'_r \in \mathcal{V}'_r$ of the replacement graph—this directly informs the alignment of the replacement fragment onto the parent structure.

Using the mapping $\theta^{-1}(\phi(v'_r))$, we now aim to determine the optimal alignment (rotation and translation) of the replacement fragment onto the substructure of the parent crystal that matched the query fragment. For this task, we rely on labeled point cloud representations to specify the geometry in which the replacement fragment is installed. First, we center the 3D coordinates of the (i) replacement fragment and (ii) the corresponding substructure of the parent structure at the origin of a Cartesian coordinate system. If the substructure of the parent is split across a periodic boundary, we reconstruct it using the nearest images of atoms so that Euclidean distance is equal to periodic distance. Let $\mathbf{x}_v \in \mathbb{R}^3$ be the centered 3D coordinates of the atom represented by node $v \in \mathcal{V}'_p \cup \mathcal{V}_r$ of the graph of the parent substructure or replacement fragment. Second, we find the 3×3 orthogonal (rotation) matrix Q_{opt} that optimally rotates the replacement fragment about the origin to align it with the matching parent substructure by solving the orthogonal Procrustes problem [39]:

$$Q_{opt} := \arg \min_{Q: Q^T Q = I} \sum_{v'_r \in \mathcal{V}'_r} \|Q\mathbf{x}_{v'_r} - \mathbf{x}_{\theta^{-1}(\phi(v'_r))}\|^2, \quad (1)$$

where I is the identity matrix. Here, $Q\mathbf{x}_{v'_r}$ is the position of atom v'_r of the rotated (by Q) replacement fragment, and $\mathbf{x}_{\theta^{-1}(\phi(v'_r))}$ is the position of the corresponding atom of the parent crystal structure. The objective in eqn. 1 expresses the closeness of the atoms of the rotated replacement fragment to their corresponding atoms of the parent structure. The coordinates of each atom $v_r \in \mathcal{V}_r$ of the *aligned* replacement fragment are then:

$$\mathbf{x}_{v_r, \text{aligned}} := Q_{opt}\mathbf{x}_{v_r} + \mathbf{x}_0 \quad (2)$$

where \mathbf{x}_0 is the center (accounting for periodic boundary conditions) of the coordinates of the substructure of the parent having correspondence with atoms in the replacement fragment (specifically, atoms $\theta^{-1}(\phi(\mathcal{V}'_r))$).

Finally, we *install* the replacement fragment on the parent crystal structure by (i) removing the substructure of the parent that matched the query fragment, then (ii) augmenting the parent with the replacement fragment at its aligned coordinates in eqn. 2. We also introduce bonds between the installed replacement fragment and the parent structure: if we removed a bond $\{v_p, v'_p\} \in \mathcal{E}_p$ such that $v_p \in \mathcal{V}_p \setminus \mathcal{V}'_p$ and $v'_p \in \mathcal{V}'_p$, we may introduce a new edge $\{v_p, v'_r = \phi^{-1}(\theta(v'_p))\}$ with the installed replacement fragment.

There may exist multiple, symmetry-equivalent (from the graph perspective) bijections $\theta^{-1} \circ \phi$ from the subset of atoms of the `replacement` fragment to a subset of the atoms on the substructure of the parent matching the query fragment. In this case, we solve the orthogonal Procrustes problem for all of them and select the bijection for the installation procedure that gives the lowest alignment error in eqn. 1. As opposed to choosing one of the graph-symmetry-equivalent bijections at random, this procedure to select the bijection with the optimal alignment can be important. E.g., when replacing a two-fold disordered pyridine rotamer with a corrected pyridine ligand, there are a total of twelve bijections between the pyridine replacement and the two-fold disordered pyridine query, four of which describe spatially correct [flat] rings and eight of which describe spatially incorrect [taco-shaped] rings.

The unit cell of the parent crystal is preserved after replacement. It is currently not possible to replace a query fragment in the parent with a replacement fragment that requires, to accommodate it, (i) expansion/contraction of the unit cell and/or (ii) parent atoms not belonging to but surrounding the query fragment to move.

4 Use cases of `PoreMatMod.jl`

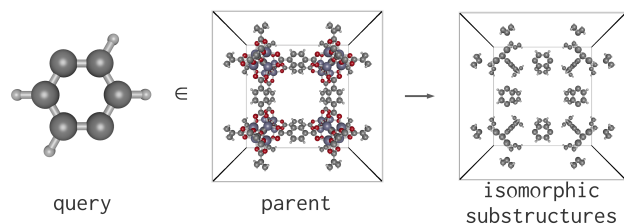
We provide illustrative examples of several practical uses of `PoreMatMod.jl`.

4.1 Subgraph matching

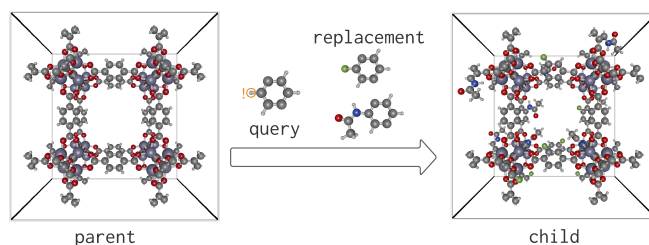
Suppose we wish to search for subgraphs of the IRMOF-1 parent crystal structure that are isomorphic to a *p*-phenylene query fragment. The `PoreMatMod.jl` code:

```
# read in crystal structure of parent MOF
parent_xtal = Crystal("IRMOF-1.cif")
# read in query fragment
query_fragment = moiety("p-phenylene.xyz")
# search for isomorphic subgraphs
search = query_fragment ∈ parent_xtal
```

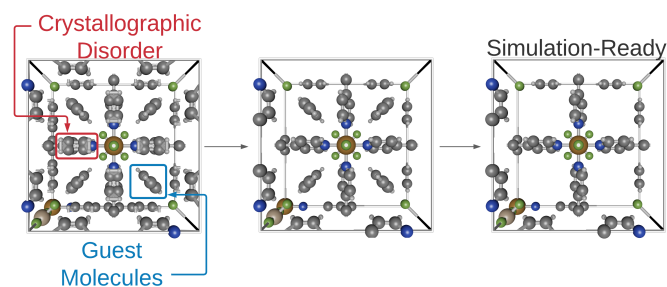
finds a set of 96 isomorphisms between subgraphs of the parent IRMOF-1 graph and the *p*-phenylene query graph, composed of 24 groups of four symmetry-equivalent isomorphisms with 24 distinct subgraphs in the IRMOF-1 unit cell (one on each BDC linker). See Fig. 2a. Notably, `PoreMatMod.jl` finds the query fragments that are split across the periodic boundaries. `search.isomorphisms` is a nested array, where the sets of symmetry-equivalent bijections between a particular \mathcal{V}'_p and \mathcal{V}_q are grouped together. We provide helper functions to count the total number of isomorphisms found, the number of unique subgraphs \mathcal{V}'_p involved in the isomorphisms, and the number of bijections for each distinct \mathcal{V}'_p .



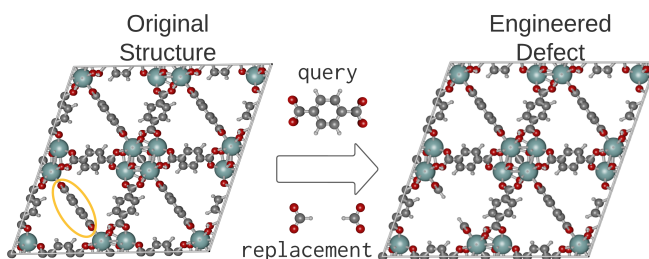
(a) subgraph matching



(b) functionalization of MOFs



(c) removal of artifacts from X-ray structure determination



(d) introduction of defects into MOFs

Figure 2: Use cases of PoreMatMod.jl. (a) Searching for subgraphs of IRMOF-1 that match *p*-phenylene, including those crossing the periodic boundary. (b) Creating a hypothetical, multi-variate MOF, by finding *p*-phenylene fragments in IRMOF-1 and partially replacing them at randomly selected positions with fluoro-*p*-phenylene or acetylamido-*p*-phenylene fragments. (c) Resolving disordered ligands and removing acetylene adsorbates in an XRD-determined SIFSIX-2-Cu-i structure [60] to afford a simulation-ready structure. (d) Introducing missing-BDC-linker defects in UiO-66, replacing them with two capping groups.

4.2 Appending functional groups to a MOF

Suppose we wish to construct a hypothetical MOF derived from a parent MOF by appending functional groups to its linkers. To accomplish this, we construct a query fragment matching a fragment of the incumbent linker in the parent structure and a replacement fragment as the functionalized version of it. Via a find-and-replace operation, `PoreMatMod.jl` produces a child structure with a new chemical functionality installed on the linker of the parent, in a reasonable geometry for (i) a warm-start for a geometry optimization and/or (ii) molecular simulations to predict its properties.

Illustrated in the overview in Sec. 2, Fig. 1 shows the child IRMOF-1 structure produced by effectively appending one trifluoromethyl functionality on each BDC linker of the parent IRMOF-1 structure. With four possible substitution sites for the trifluoromethyl group on each BDC linker—reflected by the four symmetry-equivalent isomorphisms with the *p*-phenylene query fragment for each BDC linker of the parent structure—`PoreMatMod.jl` by default chooses the substitution site on each linker which minimizes the root-mean-square deviation of coordinates for nodes $v'_r = \phi^{-1}(\theta(v'_p))$. However, `PoreMatMod.jl` grants us precise control over (i) which linkers are functionalized and (ii) which substitution site on each linker is functionalized. Specifically, the `replace` function accepts keyword arguments to control the distribution of replacement fragment installations: the functional group can be installed: (i) at random substitution sites on (a) all linkers, (b) a specified number of randomly-chosen linkers, or (c) a specified subset of the linkers or (ii) at specified substitution sites on specified linkers.

The controls over the distribution of the replacement fragment installations enable `PoreMatMod.jl` to generate multivariate (multi-linker) MOFs [17] with precise distributions of functional groups on the linkers. E.g., suppose we wish for partial functionalization of the BDC linkers of IRMOF-1 with acetamido and fluoro groups. We accomplish this by defining a *p*-phenylene query fragment, tagged with `!` at one of its hydrogen atoms, and two replacement fragments, acetamido-*p*-phenylene and fluoro-*p*-phenylene. Executing two sequential replacement operations with incomplete replacement gives the multi-variate MOF in Fig. 2b.

4.3 Repairing artifacts of experimentally-resolved MOF crystal structures

Crystal structures of MOFs experimentally determined from XRD studies often exhibit artifacts such as missing hydrogen atoms, crystallographic disorder, and unwanted guest molecules in the pores [21,42]. These artifacts are obstacles to conducting molecular simulations of gas adsorption, which require a chemically valid structure resembling the activated MOF. `PoreMatMod.jl` can repair these artifacts of XRD to give a simulation-ready crystal structure. Fig. 2c illustrates.

Unwanted solvent molecules or adsorbates in the pores of a MOF structure can be removed by searching for the graph of the solvent/adsorbate and replacing it with `nothing` (i.e., deleting it). To prevent necessary structural components from also being removed, the user may specify an option in the substructure search to return subgraphs of the parent graph that are isomorphic to the query fragment *and* are disconnected from the remainder of the nodes in the parent graph.

Disordered moieties, such as components of organic linkers with rotational freedom, manifest in crystal structures by presenting a multiplicity of conformations. To convert such a multiplicity of

conformers into a single conformer to give a simulation-ready structure, we can (i) extract the disordered substructure from the MOF and define it to be the query fragment then (ii) execute a find-and-replace with a replacement fragment constructed as a single conformation of the disordered moiety.

We can append missing hydrogen atoms to a parent structure via a find-and-replace operation with (i) the query fragment as the fragment of the parent with the missing hydrogen atom(s) and (ii) the replacement fragment as the query fragment with hydrogen atoms appropriately appended to it.

4.4 Introducing defects into MOFs

Missing-SBU/metal and missing-linker defects can be engineered into MOFs, which affect their properties [61–63]. To use molecular models and simulations to study how defects influence the properties of MOFs [52, 64], we wish to construct structure models of MOFs featuring defects. `PoreMatMod.jl` can introduce both missing-linker/SBU defects into MOFs with precision by searching for and deleting the linker/SBU, then replacing it with `nothing` or a capping group. Fig. 2d illustrates the introduction of a missing-linker defect in UiO-66 [65]. The precise controls over replacement operations in `PoreMatMod.jl` enables computational studies on how the number, location, and distribution of defects in the MOF structure influence its properties.

5 Discussion

We presented `PoreMatMod.jl` as a free, open source, user-friendly, tested, and documented Julia package for modifying crystal structures. `PoreMatMod.jl` functions as a find-and-replace algorithm on crystal structures: given a parent crystal structure, `PoreMatMod.jl` (1, via Ulmann’s subgraph isomorphism algorithm) searches for subgraphs of the parent that match the graph of a query fragment then (2, via orthogonal Procrustes) aligns and installs replacement fragments in place of the substructures of the parent matching the query fragment, giving a child crystal structure. `PoreMatMod.jl` grants precise control over the distribution and orientation of the replacement fragments on the parent. We demonstrated `PoreMatMod.jl` as a useful tool for: (1) installing functional groups on the ligands of MOF structures to generate a library of hypothetical MOFs for virtual screening (Figs. 1 and 2b); (2) repairing artifacts of XRD structure determination, such as missing hydrogen atoms, disorder, and solvent in the pores (Fig. 2c); (3) introducing missing-node/linker defects in MOFs to facilitate studies on how defects influence their properties (Fig. 2d); and (4) filtering databases of crystal structures by searching the structures for subgraphs (Fig. 2a).

In addition to MOFs, `PoreMatMod.jl` can modify other atomistic systems, whether periodic, such as semi-conductors [25], or non-periodic, such as drug molecules. The latter is achieved by assigning an arbitrary unit cell box.

Depending on the parent and query structures, run times for subgraph matching range from milliseconds to tens of seconds. E.g., the subgraph search for BDC in the replicated UiO-66 unit cell shown in Fig. 2d runs in ca. 20 s (subsequent replacement operations, however, require only ca. 5 ms). In Fig. S1, we analyze the run times for an example subgraph matching task over a large database

of MOFs. To reduce computational expense in the subgraph matching routine, (1) the parent crystal structure, where possible, should be provided as the minimal representation of the unit cell and (2) the query fragments should be provided as the minimal needed to effect the find-and-replace operation.

We now discuss limitations of the (1) subgraph matching algorithm and (2) replacement routine as implemented in `PoreMatMod.jl`. First, the subgraph matching algorithm is unable to distinguish among stereoisomers or conformations of flexible moieties, as the graph representation of a fragment is invariant to stereoisomerism and conformation changes. By imposing other geometry-based constraints on matches, such a budget on the sum of pairwise distances between the corresponding atoms after aligning the query with its matching substructure of the parent—thereby quantifying the quality of the spatial overlay—it is possible to distinguish among stereoisomers and conformations. Second, replacement of a query fragment with a replacement fragment bearing no structural commonalities (“overlap”) with it is presently not possible because there is no basis for automatically defining the optimality of the alignment of the replacement fragment with the parent crystal structure. Third, as the child structure inherits the unit cell of the parent structure, `PoreMatMod.jl` cannot currently handle replacements that require expansion/contraction of the unit cell. E.g., it cannot replace a linker of a MOF with a shorter or longer linker, as then the unit cell would need to contract/expand to accommodate it.

Future work to improve `PoreMatMod.jl` includes: improve the efficiency of the subgraph matching algorithm by parallelization and/or implementation of faster algorithms [66]; implement inexact subgraph matching (subgraph similarity searching) [67]; handle replacement fragments which (a) require unit cell expansion/contraction and/or (b) have no overlap with the query fragment; consider edge metadata (e.g., bond order) in the definition of a subgraph match; include spatial overlay criteria in the matching algorithm to discern among spatial isomers and conformations; and leverage crystallographic point group symmetries to, when enumerating e.g. functionalized MOF analogues, prevent generation of symmetry-degenerate structures.

6 Acknowledgements

We acknowledge support from the National Science Foundation (NSF) (award 1920945) (AH, CMS, XF) and the U.S. Department of Defense (DoD) Defense Threat Reduction Agency (DTRA) (award HDTRA-19-31270) (NG, CMS, PT).

References

- [1] Jian-Rong Li, Julian Sculley, and Hong-Cai Zhou. Metal-organic frameworks for separations. *Chemical Reviews*, 112(2):869–932, October 2011.
- [2] Qihui Qian, Patrick A. Asinger, Moon Joo Lee, Gang Han, Katherine Mizrahi Rodriguez, Sharon Lin, Francesco M. Benedetti, Albert X. Wu, Won Seok Chi, and Zachary P. Smith. MOF-based membranes for gas separations. *Chemical Reviews*, 120(16):8161–8266, July 2020.

- [3] Xiang Zhao, Yanxiang Wang, Dong-Sheng Li, Xianhui Bu, and Pingyun Feng. Metal-organic frameworks for separation. *Advanced Materials*, 30(37):1705189, March 2018.
- [4] Marco Taddei and Camille Petit. Engineering metal-organic frameworks for adsorption-based gas separations: from process to atomic scale. *Molecular Systems Design & Engineering*, 2021.
- [5] Bin Li, Hui-Min Wen, Wei Zhou, and Banglin Chen. Porous metal-organic frameworks for gas storage and separation: What, how, and why? *The Journal of Physical Chemistry Letters*, 5(20):3468–3479, September 2014.
- [6] Hao Li, Kecheng Wang, Yujia Sun, Christina T. Lollar, Jialuo Li, and Hong-Cai Zhou. Recent advances in gas storage and separation using metal-organic frameworks. *Materials Today*, 21(2):108–121, March 2018.
- [7] Hai-Yang Li, Shu-Na Zhao, Shuang-Quan Zang, and Jing Li. Functional metal-organic frameworks as effective sensors of gases and volatile compounds. *Chemical Society Reviews*, 49(17):6364–6401, 2020.
- [8] Lin-Tao Zhang, Ye Zhou, and Su-Ting Han. The role of metal-organic frameworks in electronic sensors. *Angewandte Chemie International Edition*, 60(28):15192–15212, February 2021.
- [9] Qi Wang and Didier Astruc. State of the Art and Prospects in Metal-Organic Framework (MOF)-Based and MOF-Derived Nanocatalysis. *Chemical Reviews*, 120:1438–1511, 2020.
- [10] Anastasiya Bavykina, Nikita Kolobov, Il Son Khan, Jeremy A. Bau, Adrian Ramirez, and Jorge Gascon. Metal-organic frameworks in heterogeneous catalysis: Recent progress, new trends, and future perspectives. *Chemical Reviews*, 120(16):8468–8535, March 2020.
- [11] Hannelore Konnerth, Babasaheb M. Matsagar, Season S. Chen, Martin H.G. Pechtl, Fa-Kuen Shieh, and Kevin C.-W. Wu. Metal-organic framework (MOF)-derived catalysts for fine chemical production. *Coordination Chemistry Reviews*, 416:213319, August 2020.
- [12] Hiroyasu Furukawa, Kyle E. Cordova, Michael O’Keeffe, and Omar M. Yaghi. The chemistry and applications of metal-organic frameworks. *Science*, 341(6149):1230444, August 2013.
- [13] Markus J. Kalmutzki, Nikita Hanikel, and Omar M. Yaghi. Secondary building units as the turning point in the development of the reticular chemistry of MOFs. *Science Advances*, 4:9180, 2018.
- [14] Mark Kalaj and Seth M. Cohen. Postsynthetic modification: An enabling technology for the advancement of metal-organic frameworks. *ACS Central Science*, 6:1046–1057, 2020.
- [15] Pravas Deria, Joseph E. Mondloch, Olga Karagiari, Wojciech Bury, Joseph T. Hupp, and Omar K. Farha. Beyond post-synthesis modification: evolution of metal-organic frameworks via building block replacement. *Chemical Society Reviews*, 43(16):5896–5912, April 2014.
- [16] Olga Karagiari, Wojciech Bury, Joseph E Mondloch, Joseph T Hupp, and Omar K Farha. Solvent-assisted linker exchange: an alternative to the de novo synthesis of unattainable metal-organic frameworks. *Angewandte Chemie International Edition*, 53(18):4530–4540, 2014.

- [17] Hexiang Deng, Christian J. Doonan, Hiroyasu Furukawa, Ricardo B. Ferreira, John Towne, Carolyn B. Knobler, Bo Wang, and Omar M. Yaghi. Multiple functional groups of varying ratios in metal-organic frameworks. *Science*, 327:846–850, 2010.
- [18] Liang Feng, Kun-Yu Wang, Gregory S Day, and Hong-Cai Zhou. The chemistry of multi-component and hierarchical framework compounds. *Chemical Society Reviews*, 48(18):4823–4853, 2019.
- [19] Lisa J. Wang, Hexiang Deng, Hiroyasu Furukawa, Felipe Gándara, Kyle E. Cordova, Dani Peri, and Omar M. Yaghi. Synthesis and characterization of metal-organic framework-74 containing 2, 4, 6, 8, and 10 different metals. *Inorganic Chemistry*, 53(12):5881–5883, May 2014.
- [20] Yamil J. Colón and Randall Q. Snurr. High-throughput computational screening of metal-organic frameworks. *Chemical Society Reviews*, 43:5735–5749, 2014.
- [21] Arni Sturluson, Melanie T. Huynh, Alec R. Kaija, Caleb Laird, Sunghyun Yoon, Feier Hou, Zhenxing Feng, Christopher E. Wilmer, Yamil J. Colón, Yongchul G. Chung, Daniel W. Siderius, and Cory M. Simon. The role of molecular modelling and simulation in the discovery and deployment of metal-organic frameworks for gas storage and separation. *Molecular Simulation*, 45(14-15):1082–1121, August 2019.
- [22] Hilal Daglar and Seda Keskin. Recent advances, opportunities, and challenges in high-throughput computational screening of MOFs for gas separations. *Coordination Chemistry Reviews*, 422:213470, November 2020.
- [23] Jack D. Evans, Guillaume Fraux, Romain Gaillac, Daniela Kohen, Fabien Trouselet, Jean-Mathieu Vanson, and François-Xavier Coudert. Computational chemistry methods for nanoporous materials. *Chemistry of Materials*, 29(1):199–212, September 2016.
- [24] Tom A. Young, Razvan Gheorghe, and Fernanda Duarte. cgbind: A python module and web app for automated metallogage construction and host-guest characterization. *Journal of Chemical Information and Modeling*, 60(7):3546–3557, June 2020.
- [25] Julia A. Schmidt, Joseph A. Weatherby, Isaac J. Sugden, Alejandro Santana-Bonilla, Francesco Salerno, Matthew J. Fuchter, Erin R. Johnson, Jenny Nelson, and Kim E. Jelfs. Computational screening of chiral organic semiconductors: Exploring side-group functionalization and assembly to optimize charge transport. *Crystal Growth & Design*, August 2021.
- [26] Christopher E. Wilmer, Michael Leaf, Chang Yeon Lee, Omar K. Farha, Brad G. Hauser, Joseph T. Hupp, and Randall Q. Snurr. Large-scale screening of hypothetical metal-organic frameworks. *Nature Chemistry*, 4:83–89, 2012.
- [27] Caroline M. Draznieks, John M. Newsam, Alan M. Gorman, Clive M. Freeman, and Gérard Férey. De Novo Prediction of Inorganic Structures Developed through Automated Assembly of Secondary Building Units (AASBU Method). *Angewandte Chemie International Edition*, 39:2270–2275, 2000.
- [28] Richard L. Martin and Maciej Haranczyk. Optimization-based design of metal-organic framework materials. *Journal of Chemical Theory and Computation*, 9:2816–2825, 2013.

- [29] Richard L. Martin and Maciej Haranczyk. Construction and characterization of structure models of crystalline porous polymers. *Crystal Growth and Design*, 14:2431–2440, 2014.
- [30] Peter G. Boyd and Tom K. Woo. A generalized method for constructing hypothetical nanoporous materials of any net topology from graph theory. *Crystal Engineering Communications*, 18:3777–3792, 2016.
- [31] Yamil J. Colon, Diego A. Gomez-Gualdro, and Randall Q. Snurr. Topologically guided, automated construction of metal-organic frameworks and their evaluation for energy-related applications. *Crystal Growth and Design*, 17:5801–5810, 2017.
- [32] Lukas Turcani, Andrew Tarzia, Filip T. Szczypiński, and Kim E. Jelfs. stk: An extendable python framework for automated molecular and supramolecular structure assembly and discovery. *The Journal of Chemical Physics*, 154(21):214102, June 2021.
- [33] Sangwon Lee, Baekjun Kim, Hyun Cho, Hooseung Lee, Sarah Yunmi Lee, Eun Seon Cho, and Jihan Kim. Computational screening of trillions of metal-organic frameworks for high-performance methane storage. *ACS Applied Materials & Interfaces*, 13(20):23647–23654, May 2021.
- [34] Matthew Witman, Sanliang Ling, Samantha Anderson, Lianheng Tong, Kyriakos C. Stylianou, Ben Slater, Berend Smit, and Maciej Haranczyk. In silico design and screening of hypothetical MOF-74 analogs and their experimental synthesis. *Chemical Science*, 7(9):6263–6272, 2016.
- [35] Jan-Grimo Sobez and Markus Reiher. Molassembler: Molecular graph construction, modification, and conformer generation for inorganic and organic molecules. *Journal of Chemical Information and Modeling*, 60(8):3884–3900, July 2020.
- [36] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.
- [37] Julian R. Ullmann. An algorithm for subgraph isomorphism. *Journal of the ACM*, 23(1):31–42, 1976.
- [38] Hans-Christian Ehrlich and Matthias Rarey. Maximum common subgraph isomorphism algorithms and their applications in molecular science: a review. *WIREs Computational Molecular Science*, 1(1):68–79, January 2011.
- [39] Peter H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31:1–10, 1966.
- [40] Ch. Baerlocher and L.B. McCusker. Database of zeolite structures. <http://www.iza-structure.org/databases/>, July 2021.
- [41] Yongchul G. Chung, Jeffrey Camp, Maciej Haranczyk, Benjamin J. Sikora, Wojciech Bury, Vaiva Krungleviciute, Taner Yildirim, Omar K. Farha, David S. Sholl, and Randall Q. Snurr. Computation-ready, experimental metal-organic frameworks: A tool to enable high-throughput screening of nanoporous crystals. *Chemistry of Materials*, 26(21):6185–6192, October 2014.
- [42] Yongchul G. Chung, Emmanuel Haldoupis, Benjamin J. Bucior, Maciej Haranczyk, Seulchan Lee, Konstantinos D. Vogiatzis, Sanliang Ling, Marija Milisavljevic, Hongda Zhang, Jeff S. Camp, Ben

- Slater, J. Ilja Siepmann, David S. Sholl, and Randall Q. Snurr. Computation-Ready Experimental Metal-Organic Framework (CoRE MOF) 2019 Dataset (Version 1.1.3) [Data set]. *Zenodo*, 2020.
- [43] Peyman Z. Moghadam, Aurelia Li, Seth B. Wiggin, Andi Tao, Andrew G. P. Maloney, Peter A. Wood, Suzanna C. Ward, and David Fairen-Jimenez. Development of a cambridge structural database subset: A collection of metal-organic frameworks for past, present, and future. *Chemistry of Materials*, 29(7):2618–2625, March 2017.
- [44] Minman Tong, Youshi Lan, Qingyuan Yang, and Chongli Zhong. Exploring the structure-property relationships of covalent organic frameworks for noble gas separations. *Chemical Engineering Science*, 168:456–464, August 2017.
- [45] Daniele Ongari, Aliaksandr V Yakutovich, Leopold Talirz, and Berend Smit. Building a consistent and reproducible database for adsorption evaluation in covalent-organic frameworks. *ACS Central Science*, 5(10):1663–1675, 2019.
- [46] Marcin Miklitz, Shan Jiang, Rob Clowes, Michael E. Briggs, Andrew I. Cooper, and Kim E. Jelfs. Computational screening of porous organic molecules for xenon/krypton separation. *The Journal of Physical Chemistry C*, 121(28):15211–15222, July 2017.
- [47] Christopher E. Wilmer, Michael Leaf, Chang Yeon Lee, Omar K. Farha, Brad G. Hauser, Joseph T. Hupp, and Randall Q. Snurr. Large-scale screening of hypothetical metal-organic frameworks. *Nature Chemistry*, 4(2):83–89, November 2011.
- [48] Andrew Tarzia, James E. M. Lewis, and Kim E. Jelfs. High-throughput computational evaluation of low symmetry Pd₂L₄ cages to aid in system design. *Angewandte Chemie International Edition*, August 2021.
- [49] Zhijie Chen, Penghao Li, Ryther Anderson, Xingjie Wang, Xuan Zhang, Lee Robison, Louis R. Redfern, Shinya Moribe, Timur Islamoglu, Diego A. Gómez-Gualdrón, Taner Yildirim, J. Fraser Stoddart, and Omar K. Farha. Balancing volumetric and gravimetric uptake in highly porous materials for clean energy. *Science*, 368(6488):297–303, April 2020.
- [50] Peter G. Boyd, Arunraj Chidambaram, Enrique García-Díez, Christopher P. Ireland, Thomas D. Daff, Richard Bounds, Andrzej Gładysiak, Pascal Schouwink, Seyed Mohamad Moosavi, M. Mercedes Maroto-Valer, Jeffrey A. Reimer, Jorge A. R. Navarro, Tom K. Woo, Susana Garcia, Kyriakos C. Stylianou, and Berend Smit. Data-driven design of metal-organic frameworks for wet flue gas CO₂ capture. *Nature*, 576(7786):253–256, 2019.
- [51] Song Li, Yongchul G. Chung, Cory M. Simon, and Randall Q. Snurr. High-throughput computational screening of multivariate metal-organic frameworks (MTV-MOFs) for CO₂ capture. *The Journal of Physical Chemistry Letters*, 8(24):6135–6141, December 2017.
- [52] Sanggyu Chong, Günther Thiele, and Jihan Kim. Excavating hidden adsorption sites in metal-organic frameworks using rational defect engineering. *Nature Communications*, 8:1539, 2017.
- [53] Arthur De Vos, Kevin Hendrickx, Pascal Van Der Voort, Veronique Van Speybroeck, and Kurt Lejaeghere. Missing linkers: An alternative pathway to UiO-66 electronic structure engineering. *Chemistry of Materials*, 29(7):3006–3019, March 2017.

- [54] Pritha Ghosh, Yamil J. Colón, and Randall Q. Snurr. Water adsorption in UiO-66: the importance of defects. *Chem. Commun.*, 50(77):11329–11331, 2014.
- [55] Seyed Mohamad Moosavi, Aditya Nandy, Kevin M. Jablonka, Diele Ongari, Jon P. Janet, Peter G. Boyd, Yongjin Lee, Berend Smit, and Heather J. Kulik. Understanding the diversity of the metal-organic framework ecosystem. *Nature Communications*, 11:4068, 2020.
- [56] Beatriz Cordero, Verónica Gómez, Ana E. Platero-Prats, Marc Revés, Jorge Echeverría, Eduard Cremades, Flavia Barragána, and Santiago Alvarez. Covalent radii revisited. *Dalton Transactions*, pages 2832–2838, 2008.
- [57] Taoyi Chen and Thomas A. Manz. A collection of forcefield precursors for metal–organic frameworks. *RSC Advances*, 9(63):36492–36507, 2019.
- [58] Marcus D Hanwell, Donald E Curtis, David C Lonie, Tim Vandermeersch, Eva Zurek, and Geoffrey R Hutchison. Avogadro: an advanced semantic chemical editor, visualization, and analysis platform. *Journal of Cheminformatics*, 4(1):1–17, 2012.
- [59] David Dubbeldam, Sofía Calero, and Thijs J.H. Vlugt. iRASP: GPU-accelerated visualization software for materials scientists. *Molecular Simulation*, 44:653–676, 2018.
- [60] Xili Cui, Kaijie Chen, Huabin Xing, Qiwei Yang, Rajamani Krishna, Zongbi Bao, Hui Wu, Wei Zhou, Xinglong Dong, Yu Han, Bin Li, Qilong Ren, Michael J. Zaworotko, and Banglin Chen. Pore chemistry and size control in hybrid porous materials for acetylene capture from ethylene. *Science*, 353:141–144, 2016.
- [61] Zhenlan Fang, Bart Bueken, Dirk E. De Vos, and Roland A. Fischer. Defect-engineered metal-organic frameworks. *Angewandte Chemie International Edition*, 54:7234–7254, 2015.
- [62] Jianwei Ren, Mpho Ledwaba, Nicholas M. Musyoka, Henrietta W. Langmi, Mkhulu Mathe, Shijun Liao, and Wan Pang. Structural defects in metal-organic frameworks (MOFs): Formation, detection and control towards practices of interests. *Coordination Chemistry Reviews*, 349:169–197, 2017.
- [63] Andrew S. Rosen, Justin M. Notestein, and Randall Q. Snurr. Realizing the data-driven, computational discovery of metal-organic framework catalysts. *arXiv*, page 2108.06667, 2021.
- [64] Hoeyeon Kim, Sangwon Lee, and Jihan Kim. Computational analysis of linker defective metal-organic frameworks for membrane separation applications. *Langmuir*, 35:3917–3924, 2019.
- [65] Yi Feng, Qian Chen, Minqi Jiang, and Jianfeng Yao. Tailoring the properties of UiO-66 through defect engineering: A review. *Industrial & Engineering Chemistry Research*, 58(38):17646–17659, 2019.
- [66] Jinsoo Lee, Wook-Shin Han, Romans Kasperovics, and Jeong-Hoon Lee. An in-depth comparison of subgraph isomorphism algorithms in graph databases. *Proceedings of the VLDB Endowment*, 6(2):133–144, December 2012.

- [67] Xifeng Yan, Philip S Yu, and Jiawei Han. Substructure similarity search in graph databases. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 766–777, 2005.