

PoreMatMod.jl: Julia Package for *in Silico* Postsynthetic Modification of Crystal Structure Models

E. Adrian Henle, Nickolas Gantzler, Praveen K. Thallapally, Xiaoli Z. Fern, and Cory M. Simon*



Cite This: *J. Chem. Inf. Model.* 2022, 62, 423–432



Read Online

ACCESS |



Metrics & More

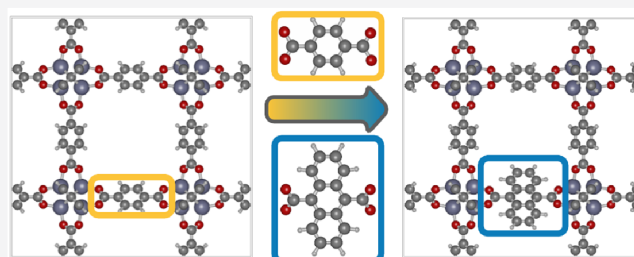


Article Recommendations



Supporting Information

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. For example, one can install functional groups on the linkers of a parent MOF, mimicking postsynthetic modification. Other applications of PoreMatMod.jl to modify crystal structure models include introducing defects with precision and correcting artifacts of X-ray structure determination (adding missing hydrogen atoms, resolving disorder, and removing guest molecules). The find-and-replace operations implemented by PoreMatMod.jl can be applied broadly to diverse atomistic systems for various *in silico* structural modification tasks.



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 ions or metal-oxo clusters (“secondary building units” or 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 metals¹⁹ 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. Molassembler³⁵ generates molecules from graph representations and allows modification of them but does not pertain to crystalline systems. An in-house code FunctionalizeThis^{36,37} also achieved automatic functionalization of MOFs by (i, pattern matching) comparing interatomic distances and atom types between an input MOF and input fragment and then (ii, functionalization) installing a new fragment on the MOF using three manually specified anchor atoms for alignment. ChemSpaX³⁸ is an open source Python package to automatically append functional groups onto 3D molecular structures. molSimplify³⁹ and QChASM⁴⁰ implement a replacement routine for monovalent functional groups; QChASM can also automatically replace ligands and catalysts in ligand-catalyst complexes.

Received: October 9, 2021

Published: January 14, 2022



In this work, we present `PoreMatMod.jl`, a new, free, open-source, tested, and documented software package, written in the Julia programming language,⁴¹ 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 algorithm⁴² to search for all subgraphs of the parent graph that are isomorphic to the query graph,⁴³ (2) uses orthogonal Procrustes⁴⁴ to align the replacement fragment on the matching substructure of the parent structure, and 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. In contrast to prior structural modification tools, `PoreMatMod.jl` can handle query and replacement fragments split across the periodic boundary. We demonstrate that `PoreMatMod.jl` is useful for the following:

- tuning the chemistry of existing crystal structure models of porous materials (e.g., curated databases of experimentally reported structures in refs 45–51) to generate a library of hypothetical materials for computational screening (e.g., refs 36 and 52–55)
- generating heterogeneous, multilinker and/or multi-SBU MOFs with precise control of functional group and/or SBU placement (e.g., ref 56)
- 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., refs 46, 57, and 58)
- introducing missing-linker and missing-SBU defects into MOFs to enable computational studies on the influence of such defects on properties (e.g., refs 59–61)
- searching for subgraphs in libraries of crystal structure models to, e.g., filter structures, characterize chemical diversity,⁶² or search for MOFs with certain building blocks.^{63,64}

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. For coding novices, we provide a graphical interface to `PoreMatMod.jl` in the form of a Pluto notebook.⁶⁵

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 and then (2, replace) replace specified instances of the query fragments with properly aligned replacement fragments to produce a child crystal structure. For example, suppose we wish to replace all 1,4-benzodicarboxylate (BDC) linkers in

IRMOF-1⁶⁶ with trifluoromethyl-BDC, mimicking BBR. The following `PoreMatMod.jl` code produces a crystal structure model of a hypothetical, functionalized IRMOF-1 child shown in Figure 1.

```
# 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)
```

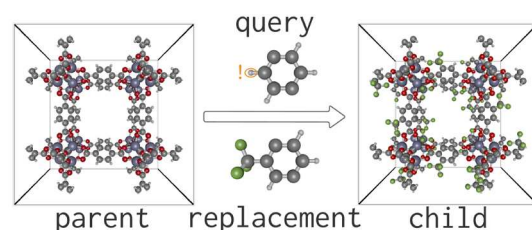


Figure 1. Overview of find-and-replace operations in `PoreMatMod.jl` to, for example, construct a hypothetical MOF. Here, we replace *p*-phenylene query fragments in the IRMOF-1 parent⁶⁶ with trifluoromethyl-*p*-phenylene replacement fragments (in random orientations) to give a functionalized child structure. The masked atom of the query fragment to be replaced is annotated with !.

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.

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 the replacement fragment onto the parent structure for installation.

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 67 and 68. `PoreMatMod.jl` allows (i) alteration of the bonding rules and/or (ii) manual bond assignment.

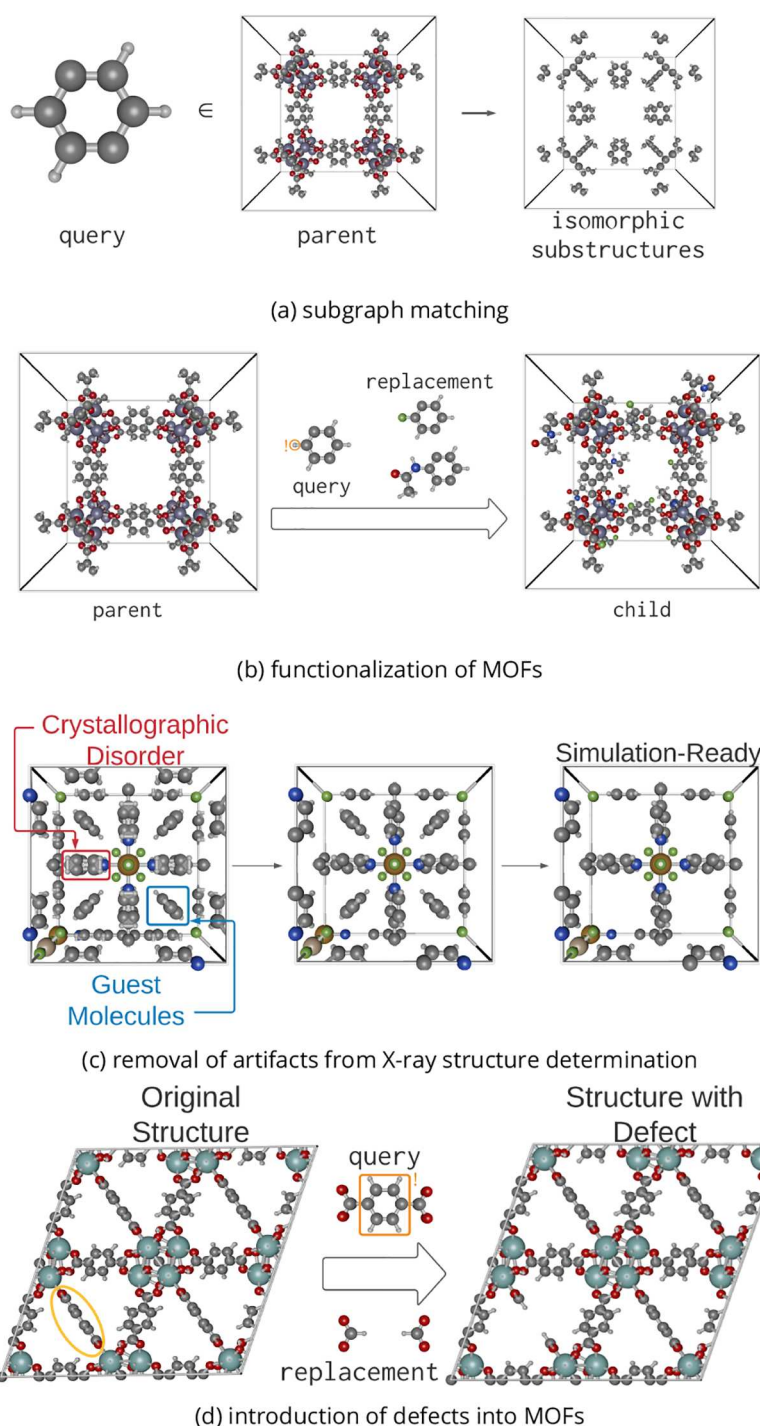


Figure 2. Use cases of PoreMatMod.jl. (a) Searching for subgraphs of IRMOF-1⁶⁶ that match *p*-phenylene, including those crossing the periodic boundary. (b) Creating a hypothetical, multivariate MOF, by finding *p*-phenylene fragments in IRMOF-1 and partially replacing them at randomly selected positions 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 structure⁷² to afford a simulation-ready structure. (d) Introducing missing-BDC-linker defects in UiO-66, replacing them with two capping groups.⁷³

The query and replacement chemical fragments for the find-and-replace task at hand may be constructed (i) manually, by (a) cutting them from the parent structure or (b) building them in a molecule editor (e.g., Avogadro⁶⁹), or (ii) programmatically, by generating them from bonding graphs via a structure prediction tool (e.g., Open Babel⁷⁰).

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,⁴³ 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 (subgraph $\Rightarrow \mathcal{V}'_p \subseteq \mathcal{V}_p$ 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 $\{v_{p,i}, v_{p,j}\} \in \mathcal{E}'_p \Leftrightarrow \{\theta(v_{p,i}), \theta(v_{p,j})\} \in \mathcal{E}_q$ 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.⁴² 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 Figure 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, and then install the replacement fragment on the parent in its place to give the child crystal structure (e.g., in Figure 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., C $\rightarrow$ 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 Figure 1, the H atom of the *p*-phenylene is a masked atom. The atom property viewer in iRASP⁷¹ 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 $G'_r = (\mathcal{V}'_r, \mathcal{E}'_r)$ of the replacement graph that is

isomorphic to the induced subgraph of the query graph containing the nonmasked 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 nonmasked 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 translate (i) the substructure of the replacement fragment having correspondence with the parent substructure and (ii) the corresponding substructure of the parent structure to be centered at the origin of a Cartesian coordinate system. If the substructure of the parent is split across a periodic boundary, we, prior to centering, first conglomerate the atoms into a nonperiodic, Euclidean space by (i) finding the connected components of the graph of the parent substructure isomorphic to the query after bonds traversing the periodic boundary are deleted, then (ii) shifting its atoms, component-by-component, to occupy the position of the nearest periodic image that places them near a reference (the largest) component. Let $\mathbf{x}_v \in \mathbb{R}^3$ be the centered (on the basis of the atoms involved in the correspondence) 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 problem⁴⁴

$$Q_{\text{opt}} := \underset{Q: Q^T Q = I}{\operatorname{argmin}} \sum_{v_r \in \mathcal{V}'_r} \|Q \mathbf{x}_{v_r} - \mathbf{x}_{\theta^{-1}(\phi(v_r))}\|^2 \quad (1)$$

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 eq 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

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

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 and then (ii) augmenting the parent with the replacement fragment at its aligned coordinates in eq 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 introduce a new edge $\{v_p, v_r = \phi^{-1}(\theta(v'_p))\}$ with the installed replacement fragment.

When replacing multiple query-matching substructures of a parent with the same replacement fragment, PoreMatMod.jl first finds the optimal alignments onto all of the matches, then installs all of the replacement fragments, and then finally removes all of the query-matching substructures of the parent. This allows PoreMatMod.jl to replace two substructures of a parent structure even if there is a nonempty intersection between them (see Figure S1).

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 eq 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 2-fold disordered pyridine rotamer with a corrected pyridine ligand, there are a total of 12 bijections between the pyridine replacement and the 2-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 isomorphic 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. For all examples in Figures 1–3, the advantage of using PoreMatMod.jl, as opposed to manually editing these structures in a graphical molecular editor like Avogadro,⁶⁹ is that repetitive find-and-replace operations may be performed programmatically, in high-throughput settings.

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 loads the parent and query

```
# 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
```

structures and finds a set of 96 isomorphisms between subgraphs of the parent graph and the query graph, composed of 24 groups of four symmetry-equivalent isomorphisms involving 24 distinct subgraphs of the IRMOF-1 unit cell (one on each BDC linker). See Figure 2a. Notably, PoreMatMod.jl finds the query fragments that are split across the periodic boundaries. `search.isomorphisms`

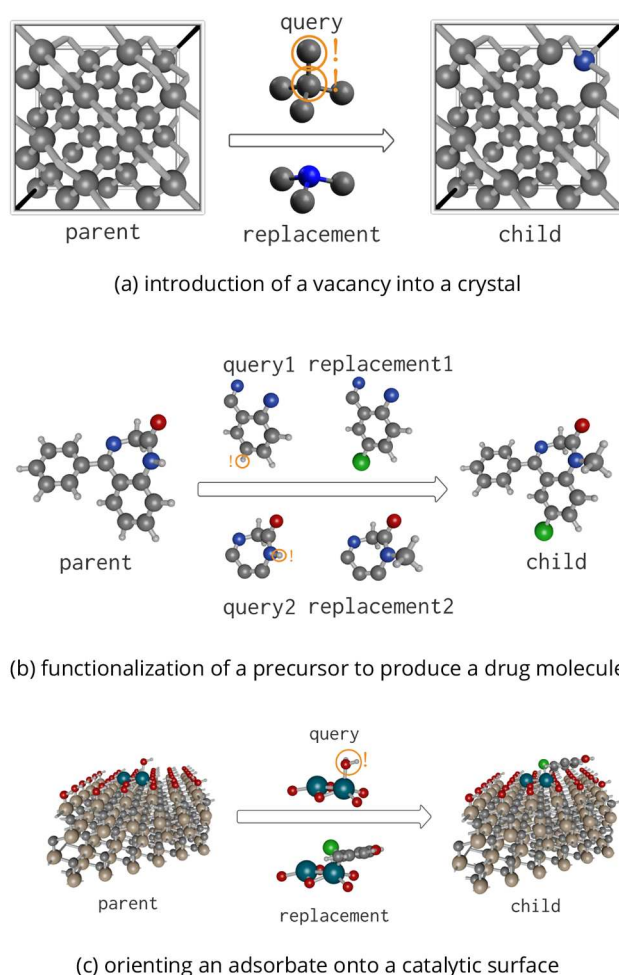


Figure 3. PoreMatMod.jl use cases beyond MOFs. (a) Introduction of a nitrogen-vacancy defect into a replicated diamond crystal.⁸³ (b) Generation of an active pharmaceutical ingredient (diazepam) by two-step functionalization of a molecular precursor (benzodiazepine).⁸⁴ (c) Orienting an adsorbate (4-chlorophenol) onto catalytic sites (Pd_2) of an inorganic surface (silicon carbide) as an initial condition for a DFT calculation.⁸⁶ Masked atoms annotated with !.

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 $\mathcal{V}'_p$.

4.2. Appending Functional Groups to a Structure. 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 Section 2, Figure 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 between the *p*-phenylene query fragment and each BDC linker of the parent—PoreMatMod.jl by default chooses the substitution site on each linker which minimizes the alignment error. 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 (multilinker) MOFs¹⁷ with precise distributions of functional groups on the linkers. E.g., suppose we wish for partial functionalization, with acetamido and fluoro groups, of the BDC linkers of IRMOF-1. 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 multivariate MOF in Figure 2b.

Just as the subgraph matching routine identified the *p*-phenylene query fragment split across the periodic boundaries of the unit cell in Figure 2a, note PoreMatMod.jl also handles the installation of replacement fragments split across periodic boundaries (illustrated in Figure 2b). If the length of the replacement fragment exceeds the length of the unit cell, however, the unit cell of the parent crystal structure must first be replicated (PoreMatMod.jl is capable of this).

4.3. Construction of a MOF with Mixed-Metal SBUs.

The SBUs of a MOF could share connectivity and geometry but be composed of different metal ions.⁷⁴ PoreMatMod.jl enables facile *in silico* metal exchange in MOFs with pure-metal SBUs to grant mixed-metal SBUs with precisely controlled ratios and configurations of metals within each SBU. Figure S2 illustrates the modification of UiO-66 to create an analog structure with mixed-metal SBUs via a find-and-replace operation.

4.4. Repairing Artifacts of Experimentally Resolved Crystal Structures.

Crystal structures experimentally determined from XRD studies often exhibit artifacts such as missing hydrogen atoms, crystallographic disorder, and unwanted guest molecules.^{21,47,48} These artifacts are obstacles to molecular simulations and electronic structure calculations involving MOF structures, which require a clean and chemically valid structure. PoreMatMod.jl can repair these artifacts of XRD to give a simulation-ready crystal structure, as illustrated in Figure 2c.

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 and 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.

4.5. Introducing Defects. Missing-SBU and missing-linker defects can be engineered into MOFs, which affect their properties.^{75–77} PoreMatMod.jl can introduce 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. Figure 2d illustrates the introduction of a missing-linker defect in UiO-66.⁷³ The precise control over replacement operations in PoreMatMod.jl enables computational studies on how the number, location, and distribution of defects in the MOF structure influence the properties of the MOF.^{59,78}

4.6. Non-MOF Use Cases. In addition to MOFs, PoreMatMod.jl can modify other atomistic systems, whether periodic, such as covalent organic frameworks⁷⁹ or semiconductors,²⁵ or nonperiodic, such as porous organic cages,⁸⁰ metal–organic polyhedra,⁸¹ or drug molecules,⁸² to generate libraries for virtual screening. Nonperiodic systems may be modified by assigning an arbitrary unit cell box.

Figure 3 demonstrates three non-MOF PoreMatMod.jl use cases: (a) introducing a nitrogen-defect into a (replicated) diamond crystal structure,⁸³ (b) generating a drug molecule through two-step functionalization of a precursor,⁸⁴ and (c) orienting adsorbates onto catalytic sites for density functional theory (DFT) calculations.^{77,85}

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 and 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 (Figures 1 and 2b); (2) repairing artifacts of XRD structure determination, such as missing hydrogen atoms, disorder, and solvent in the pores (Figure 2c); (3) introducing

missing-node/linker defects in MOFs to facilitate studies on how defects influence their properties (Figure 2d); (4) generating MOFs with mixed-metal SBUs (Figure S2); and (5) filtering databases of crystal structures by searching the structures for subgraphs (Figure 2a). Intriguingly, `PoreMatMod.jl` could be used to mutate the chemistry of a structure as part of an evolutionary algorithm to optimize a property of a material (e.g., refs 87 and 88).

Launch a graphical interface, based on a Pluto notebook, to `PoreMatMod.jl` via `PoreMatModGO()` in Julia.

For crystal structures represented in a space group with higher symmetry than the P_1 space group, `PoreMatMod.jl` by default applies symmetry operations to convert them into the P_1 space group. Optionally, `PoreMatMod.jl` can maintain the crystal structure in the higher-symmetry space group and perform find-and-replace operations on it, provided it contains the full query fragment. This would guarantee preservation of the higher-level symmetry after installing a replacement fragment. E.g., see Figure S3.

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 Figure 2d runs in ca. 20 s (Subsequent replacement operations, however, require only ca. 5 ms.). In Figure S4, 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 as 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. Finally, `PoreMatMod.jl` is not currently capable of detecting changes in the crystal space group due to find-and-replace operations, and, by default, it applies symmetry operations to put crystals in the P_1 space group.

Future work to improve `PoreMatMod.jl` includes the following: improve the efficiency of the subgraph matching algorithm by parallelization and/or implementation of faster algorithms;⁸⁹ implement inexact subgraph matching (subgraph similarity searching);⁹⁰ 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; leverage crystallographic space group symmetries to, when enumerating, e.g., functionalized MOF analogues, prevent generation of symmetry-degenerate structures; and implement a post-(find-and-replace) geometry optimization with classical force fields.

Data and Software Availability. Our `PoreMatMod.jl` software is free and open, under an MIT License, and is hosted on Github at github.com/SimonEnsemble/PoreMatMod.jl with a link to the documentation. All data and code to reproduce the examples in this paper are available in the same Github repository. This paper pertains to v0.2.9 of `PoreMatMod.jl`.

■ ASSOCIATED CONTENT

SI Supporting Information

The Supporting Information is available free of charge at <https://pubs.acs.org/doi/10.1021/acs.jcim.1c01219>.

Analysis of substructure search algorithm performance and additional use case illustrations (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Cory M. Simon – School of Chemical, Biological, and Environmental Engineering, Oregon State University, Corvallis, Oregon 97331, United States; orcid.org/0000-0002-8181-9178; Email: Cory.Simon@oregonstate.edu

Authors

E. Adrian Henle – School of Chemical, Biological, and Environmental Engineering, Oregon State University, Corvallis, Oregon 97331, United States; orcid.org/0000-0002-3331-7173

Nickolas Gantzler – Department of Physics, Oregon State University, Corvallis, Oregon 97331, United States

Praveen K. Thallapally – Pacific Northwest National Laboratory, Richland, Washington 99354, United States; orcid.org/0000-0001-7814-4467

Xiaoli Z. Fern – School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, Oregon 97331, United States

Complete contact information is available at: <https://pubs.acs.org/doi/10.1021/acs.jcim.1c01219>

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We acknowledge support from the National Science Foundation (NSF) (award 1920945) (A.H., C.M.S., X.F.) and the U.S. Department of Defense (DoD) Defense Threat Reduction Agency (DTRA) (award HDTRA-19-31270) (N.G., C.M.S., P.T.).

■ REFERENCES

(1) Li, J.-R.; Sculley, J.; Zhou, H.-C. Metal-Organic Frameworks for Separations. *Chem. Rev.* **2012**, *112* (2), 869–932.

- (2) Qian, Q.; Asinger, P. A.; Lee, M. J.; Han, G.; Rodriguez, K. M.; Lin, S.; Benedetti, F. M.; Wu, A. X.; Chi, W. S.; Smith, Z. P. MOF-Based Membranes for Gas Separations. *Chem. Rev.* **2020**, *120* (16), 8161–8266.
- (3) Zhao, X.; Wang, Y.; Li, D.-S.; Bu, X.; Feng, P. Metal-Organic Frameworks for Separation. *Adv. Mater.* **2018**, *30* (37), 1705189.
- (4) Taddei, M.; Petit, C. Engineering metal-organic frameworks for adsorption-based gas separations: from process to atomic scale. *Mol. Syst. Des. Eng.* **2021**, *6*, 841–875.
- (5) Li, B.; Wen, H.-M.; Zhou, W.; Chen, B. Porous Metal-Organic Frameworks for Gas Storage and Separation: What, How, and Why? *J. Phys. Chem. Lett.* **2014**, *5* (20), 3468–3479.
- (6) Li, H.; Wang, K.; Sun, Y.; Lollar, C. T.; Li, J.; Zhou, H.-C. Recent advances in gas storage and separation using metal-organic frameworks. *Mater. Today* **2018**, *21* (2), 108–121.
- (7) Li, H.-Y.; Zhao, S.-N.; Zang, S.-Q.; Li, J. Functional metal-organic frameworks as effective sensors of gases and volatile compounds. *Chem. Soc. Rev.* **2020**, *49* (17), 6364–6401.
- (8) Zhang, L.-T.; Zhou, Y.; Han, S.-T. The Role of Metal-Organic Frameworks in Electronic Sensors. *Angew. Chem., Int. Ed.* **2021**, *60* (28), 15192–15212.
- (9) Wang, Q.; Astruc, D. State of the Art and Prospects in Metal–Organic Framework (MOF)-Based and MOF-Derived Nanocatalysis. *Chem. Rev.* **2020**, *120*, 1438–1511.
- (10) Bavykina, A.; Kolobov, N.; Khan, I. S.; Bau, J. A.; Ramirez, A.; Gascon, J. Metal–Organic Frameworks in Heterogeneous Catalysis: Recent Progress, New Trends, and Future Perspectives. *Chem. Rev.* **2020**, *120* (16), 8468–8535.
- (11) Konnerth, H.; Matsagar, B. M.; Chen, S. S.; Precht, M. H.; Shieh, F.-K.; Wu, K. C.-W. Metal-organic framework (MOF)-derived catalysts for fine chemical production. *Coord. Chem. Rev.* **2020**, *416*, 213319.
- (12) Furukawa, H.; Cordova, K. E.; O’Keeffe, M.; Yaghi, O. M. The Chemistry and Applications of Metal-Organic Frameworks. *Science* **2013**, *341* (6149), 1230444.
- (13) Kalmutski, M. J.; Hanikel, N.; Yaghi, O. M. Secondary building units as the turning point in the development of the reticular chemistry of MOFs. *Sci. Adv.* **2018**, *4*, eaat9180.
- (14) Kalaj, M.; Cohen, S. M. Postsynthetic Modification: An Enabling Technology for the Advancement of Metal–Organic Frameworks. *ACS Cent. Sci.* **2020**, *6*, 1046–1057.
- (15) Deria, P.; Mondloch, J. E.; Karagiari, O.; Bury, W.; Hupp, J. T.; Farha, O. K. Beyond post-synthesis modification: evolution of metal-organic frameworks via building block replacement. *Chem. Soc. Rev.* **2014**, *43* (16), 5896–5912.
- (16) Karagiari, O.; Bury, W.; Mondloch, J. E.; Hupp, J. T.; Farha, O. K. Solvent-assisted linker exchange: an alternative to the de novo synthesis of unattainable metal-organic frameworks. *Angew. Chem., Int. Ed.* **2014**, *53* (18), 4530–4540.
- (17) Deng, H.; Doonan, C. J.; Furukawa, H.; Ferreira, R. B.; Towne, J.; Knobler, C. B.; Wang, B.; Yaghi, O. M. Multiple Functional Groups of Varying Ratios in Metal-Organic Frameworks. *Science* **2010**, *327*, 846–850.
- (18) Feng, L.; Wang, K.-Y.; Day, G. S.; Zhou, H.-C. The chemistry of multi-component and hierarchical framework compounds. *Chem. Soc. Rev.* **2019**, *48* (18), 4823–4853.
- (19) Wang, L. J.; Deng, H.; Furukawa, H.; Gándara, F.; Cordova, K. E.; Peri, D.; Yaghi, O. M. Synthesis and Characterization of Metal-Organic Framework-74 Containing 2, 4, 6, 8, and 10 Different Metals. *Inorg. Chem.* **2014**, *53* (12), 5881–5883.
- (20) Colón, Y. J.; Snurr, R. Q. High-throughput computational screening of metal–organic frameworks. *Chem. Soc. Rev.* **2014**, *43*, 5735–5749.
- (21) Sturluson, A.; Huynh, M. T.; Kaija, A. R.; Laird, C.; Yoon, S.; Hou, F.; Feng, Z.; Wilmer, C. E.; Colón, Y. J.; Chung, Y. G.; Siderius, D. W.; Simon, C. M. The role of molecular modelling and simulation in the discovery and deployment of metal-organic frameworks for gas storage and separation. *Mol. Simul.* **2019**, *45* (14–15), 1082–1121.
- (22) Daglar, H.; Keskin, S. Recent advances, opportunities, and challenges in high-throughput computational screening of MOFs for gas separations. *Coord. Chem. Rev.* **2020**, *422*, 213470.
- (23) Evans, J. D.; Fraux, G.; Gaillac, R.; Kohen, D.; Trouselet, F.; Vanson, J.-M.; Coudert, F.-X. Computational Chemistry Methods for Nanoporous Materials. *Chem. Mater.* **2017**, *29* (1), 199–212.
- (24) Young, T. A.; Gheorghe, R.; Duarte, F. cgbind: A Python Module and Web App for Automated Metalloclage Construction and Host-Guest Characterization. *J. Chem. Inf. Model.* **2020**, *60* (7), 3546–3557.
- (25) Schmidt, J. A.; Weatherby, J. A.; Sugden, I. J.; Santana-Bonilla, A.; Salerno, F.; Fuchter, M. J.; Johnson, E. R.; Nelson, J.; Jelfs, K. E. Computational Screening of Chiral Organic Semiconductors: Exploring Side-Group Functionalization and Assembly to Optimize Charge Transport. *Cryst. Growth Des.* **2021**, *21* (9), S036–S049.
- (26) 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.
- (27) Mellot Draznieks, C.; Newsam, J. M.; Gorman, A. M.; Freeman, C. M.; Férey, G. De Novo Prediction of Inorganic Structures Developed through Automated Assembly of Secondary Building Units (AASBU Method). *Angew. Chem., Int. Ed.* **2000**, *39*, 2270–2275.
- (28) Martin, R. L.; Haranczyk, M. Optimization-Based Design of Metal–Organic Framework Materials. *J. Chem. Theory Comput.* **2013**, *9*, 2816–2825.
- (29) Martin, R. L.; Haranczyk, M. Construction and Characterization of Structure Models of Crystalline Porous Polymers. *Cryst. Growth Des.* **2014**, *14*, 2431–2440.
- (30) Boyd, P. G.; Woo, T. K. A generalized method for constructing hypothetical nanoporous materials of any net topology from graph theory. *Cryst. Eng.* **2016**, *18*, 3777–3792.
- (31) Colon, Y. J.; Gomez-Gualdro, D. A.; Snurr, R. Q. Topologically Guided, Automated Construction of Metal-Organic Frameworks and Their Evaluation for Energy-Related Applications. *Cryst. Growth Des.* **2017**, *17*, S801–S810.
- (32) Turcani, L.; Tarzia, A.; Szczypiński, F. T.; Jelfs, K. E. stk: An extendable Python framework for automated molecular and supra-molecular structure assembly and discovery. *J. Chem. Phys.* **2021**, *154* (21), 214102.
- (33) Lee, S.; Kim, B.; Cho, H.; Lee, H.; Lee, S. Y.; Cho, E. S.; Kim, J. Computational Screening of Trillions of Metal-Organic Frameworks for High-Performance Methane Storage. *ACS Appl. Mater. Interfaces* **2021**, *13* (20), 23647–23654.
- (34) Witman, M.; Ling, S.; Anderson, S.; Tong, L.; Stylianou, K. C.; Slater, B.; Smit, B.; Haranczyk, M. In silico design and screening of hypothetical MOF-74 analogs and their experimental synthesis. *Chem. Sci.* **2016**, *7* (9), 6263–6272.
- (35) Sobez, J.-G.; Reiher, M. Molassembler: Molecular Graph Construction, Modification, and Conformer Generation for Inorganic and Organic Molecules. *J. Chem. Inf. Model.* **2020**, *60* (8), 3884–3900.
- (36) Bae, Y.-S.; Liu, J.; Wilmer, C. E.; Sun, H.; Dickey, A. N.; Kim, M. B.; Benin, A. I.; Willis, R. R.; Barpaga, D.; LeVan, M. D.; Snurr, R. Q. The effect of pyridine modification of Ni-DOBDC on CO₂ capture under humid conditions. *Chem. Commun.* **2014**, *50* (25), 3296–3298.
- (37) McDaniel, J. G.; Li, S.; Tylianakis, E.; Snurr, R. Q.; Schmidt, J. R. Evaluation of Force Field Performance for High-Throughput Screening of Gas Uptake in Metal-Organic Frameworks. *J. Phys. Chem. C* **2015**, *119* (6), 3143–3152.
- (38) Kalikadien, A.; Pidko, E. A.; Sinha, V. ChemSpaX: Exploration of Chemical Space by Automated Functionalization of Molecular Scaffold. *ChemRxiv* **2021**, in press.
- (39) Ioannidis, E. I.; Gani, T. Z. H.; Kulik, H. J. molSimplify: A toolkit for automating discovery in inorganic chemistry. *J. Comput. Chem.* **2016**, *37* (22), 2106–2117.
- (40) Ingman, V. M.; Schaefer, A. J.; Andreola, L. R.; Wheeler, S. E. QChASM: Quantum chemistry automation and structure manipulation. *WIREs Comput. Mol. Sci.* **2021**, *11* (4), e1510.

- (41) Bezanson, J.; Edelman, A.; Karpinski, S.; Shah, V. B. Julia: A fresh approach to numerical computing. *SIAM Review* **2017**, *59* (1), 65–98.
- (42) Ullmann, J. R. An algorithm for subgraph isomorphism. *J. Assoc. Comput. Mach.* **1976**, *23* (1), 31–42.
- (43) Ehrlich, H.-C.; Rarey, M. Maximum common subgraph isomorphism algorithms and their applications in molecular science: a review. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2011**, *1* (1), 68–79.
- (44) Schönemann, P. H. A generalized solution of the orthogonal procrustes problem. *Psychometrika* **1966**, *31*, 1–10.
- (45) Baerlocher, Ch.; McCusker, L. B. Database of Zeolite Structures. 2021. <http://www.iza-structure.org/databases/> (accessed 2021-12-23).
- (46) Chung, Y. G.; Camp, J.; Haranczyk, M.; Sikora, B. J.; Bury, W.; Krungleviciute, V.; Yildirim, T.; Farha, O. K.; Sholl, D. S.; Snurr, R. Q. Computation-Ready, Experimental Metal-Organic Frameworks: A Tool To Enable High-Throughput Screening of Nanoporous Crystals. *Chem. Mater.* **2014**, *26* (21), 6185–6192.
- (47) Chung, Y. G.; Haldoupis, E.; Bucior, B. J.; Haranczyk, M.; Lee, S.; Vogiatzis, K. D.; Ling, S.; Milisavljevic, M.; Zhang, H.; Camp, J. S.; Slater, B.; Siepmann, J. I.; Sholl, D. S.; Snurr, R. Q. *Computation-Ready Experimental Metal-Organic Framework (CoRE MOF) 2019 Dataset (Version 1.1.3)*; 2020.
- (48) Moghadam, P. Z.; Li, A.; Wiggin, S. B.; Tao, A.; Maloney, A. G. P.; Wood, P. A.; Ward, S. C.; Fairen-Jimenez, D. Development of a Cambridge Structural Database Subset: A Collection of Metal–Organic Frameworks for Past, Present, and Future. *Chem. Mater.* **2017**, *29* (7), 2618–2625.
- (49) Tong, M.; Lan, Y.; Yang, Q.; Zhong, C. Exploring the structure-property relationships of covalent organic frameworks for noble gas separations. *Chem. Eng. Sci.* **2017**, *168*, 456–464.
- (50) Ongari, D.; Yakutovich, A. V.; Talirz, L.; Smit, B. Building a consistent and reproducible database for adsorption evaluation in covalent-organic frameworks. *ACS Cent. Sci.* **2019**, *5* (10), 1663–1675.
- (51) 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* (28), 15211–15222.
- (52) 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* (2), 83–89.
- (53) Tarzia, A.; Lewis, J. E. M.; Jelfs, K. E. High-Throughput Computational Evaluation of Low Symmetry Pd₂L₄ Cages to Aid in System Design. *Angew. Chem., Int. Ed.* **2021**, *60*, 20879.
- (54) Chen, Z.; Li, P.; Anderson, R.; Wang, X.; Zhang, X.; Robison, L.; Redfern, L. R.; Moribe, S.; Islamoglu, T.; Gómez-Gualdrón, D. A.; Yildirim, T.; Stoddart, J. F.; Farha, O. K. Balancing volumetric and gravimetric uptake in highly porous materials for clean energy. *Science* **2020**, *368* (6488), 297–303.
- (55) Boyd, P. G.; Chidambaram, A.; García-Díez, E.; Ireland, C. P.; Daff, T. D.; Bounds, R.; Gladysiak, A.; Schouwink, P.; Mohamad Moosavi, S.; Maroto-Valer, M. M.; Reimer, J. A.; Navarro, J. A. R.; Woo, T. K.; Garcia, S.; Stylianou, K. C.; Smit, B. Data-driven design of metal-organic frameworks for wet flue gas CO₂ capture. *Nature* **2019**, *576* (7786), 253–256.
- (56) Li, S.; Chung, Y. G.; Simon, C. M.; Snurr, R. Q. High-Throughput Computational Screening of Multivariate Metal-Organic Frameworks (MTV-MOFs) for CO₂ Capture. *J. Phys. Chem. Lett.* **2017**, *8* (24), 6135–6141.
- (57) Altintas, C.; Avci, G.; Daglar, H.; Azar, A. N. V.; Erucar, I.; Velioglu, S.; Keskin, S. An extensive comparative analysis of two MOF databases: high-throughput screening of computation-ready MOFs for CH₄ and H₂ adsorption. *J. Mater. Chem. A* **2019**, *7* (16), 9593–9608.
- (58) Velioglu, S.; Keskin, S. Revealing the effect of structure curations on the simulated CO₂ separation performances of MOFs. *Mater. Adv.* **2020**, *1* (3), 341–353.
- (59) Chong, S.; Thiele, G.; Kim, J. Excavating hidden adsorption sites in metal-organic frameworks using rational defect engineering. *Nat. Commun.* **2017**, *8*, 1539.
- (60) De Vos, A.; Hendrickx, K.; Voort, P. V. D.; Speybroeck, V. V.; Lejaeghere, K. Missing Linkers: An Alternative Pathway to UiO-66 Electronic Structure Engineering. *Chem. Mater.* **2017**, *29* (7), 3006–3019.
- (61) Ghosh, P.; Colón, Y. J.; Snurr, R. Q. Water adsorption in UiO-66: the importance of defects. *Chem. Commun.* **2014**, *50* (77), 11329–11331.
- (62) Moosavi, S. M.; Nandy, A.; Jablonka, K. M.; Ongari, D.; Janet, J. P.; Boyd, P. G.; Lee, Y.; Smit, B.; Kulik, H. J. Understanding the diversity of the metal-organic framework ecosystem. *Nat. Commun.* **2020**, *11*, 4068.
- (63) Halder, P.; Prerna; Singh, J. K. Building Unit Extractor for Metal-Organic Frameworks. *J. Chem. Inf. Model.* **2021**, DOI: 10.1021/acs.jcim.1c00547.
- (64) Bucior, B. J.; Rosen, A. S.; Haranczyk, M.; Yao, Z.; Ziebel, M. E.; Farha, O. K.; Hupp, J. T.; Siepmann, J. I.; Aspuru-Guzik, A.; Snurr, R. Q. Identification Schemes for Metal-Organic Frameworks To Enable Rapid Search and Cheminformatics Analysis. *Cryst. Growth Des.* **2019**, *19* (11), 6682–6697.
- (65) Fons Van der Plas. Pluto.jl. 2020. <https://github.com/fonsp/Pluto.jl> (accessed 2021-12-23).
- (66) Li, H.; Eddaoudi, M.; O’Keeffe, M.; Yaghi, O. M. Design and synthesis of an exceptionally stable and highly porous metal-organic framework. *Nature* **1999**, *402* (6759), 276–279.
- (67) Cordero, B.; Gómez, V.; Platero-Prats, A. E.; Revés, M.; Echeverría, J.; Cremades, E.; Barragán, F.; Alvarez, S. Covalent radii revisited. *Dalton Trans.* **2008**, 2832–2838.
- (68) Chen, T.; Manz, T. A. A collection of forcefield precursors for metal–organic frameworks. *RSC Adv.* **2019**, *9* (63), 36492–36507.
- (69) Hanwell, M. D.; Curtis, D. E.; Lonie, D. C.; Vandermeersch, T.; Zurek, E.; Hutchison, G. R. Avogadro: an advanced semantic chemical editor, visualization, and analysis platform. *J. Cheminf.* **2012**, *4* (1), 17.
- (70) Yoshikawa, N.; Hutchison, G. R. Fast, efficient fragment-based coordinate generation for Open Babel. *J. Cheminf.* **2019**, *11* (1), 49.
- (71) Dubbeldam, D.; Calero, S.; Vlugt, T. J. iRASP: GPU-accelerated visualization software for materials scientists. *Mol. Simul.* **2018**, *44*, 653–676.
- (72) Cui, X.; Chen, K.; Xing, H.; Yang, Q.; Krishna, R.; Bao, Z.; Wu, H.; Zhou, W.; Dong, X.; Han, Y.; Li, B.; Ren, Q.; Zaworotko, M. J.; Chen, B. Pore chemistry and size control in hybrid porous materials for acetylene capture from ethylene. *Science* **2016**, *353*, 141–144.
- (73) Feng, Y.; Chen, Q.; Jiang, M.; Yao, J. Tailoring the properties of UiO-66 through defect engineering: A review. *Ind. Eng. Chem. Res.* **2019**, *58* (38), 17646–17659.
- (74) Kalmutzki, M. J.; Hanikel, N.; Yaghi, O. M. Secondary building units as the turning point in the development of the reticular chemistry of MOFs. *Sci. Adv.* **2018**, *4* (10), eaat9180.
- (75) Fang, Z.; Bueken, B.; De Vos, D. E.; Fischer, R. A. Defect-Engineered Metal-Organic Frameworks. *Angew. Chem., Int. Ed.* **2015**, *54*, 7234–7254.
- (76) Ren, J.; Ledwaba, M.; Musyoka, N. M.; Langmi, H. W.; Mathe, M.; Liao, S.; Pang, W. Structural defects in metal-organic frameworks (MOFs): Formation, detection and control towards practices of interests. *Coord. Chem. Rev.* **2017**, *349*, 169–197.
- (77) Rosen, A. S.; Notestein, J. M.; Snurr, R. Q. Realizing the Data-Driven, Computational Discovery of Metal-Organic Framework Catalysts. *arXiv* **2021**, 2108.06667.
- (78) Kim, H.; Lee, S.; Kim, J. Computational Analysis of Linker Defective Metal-Organic Frameworks for Membrane Separation Applications. *Langmuir* **2019**, *35*, 3917–3924.
- (79) Diercks, C. S.; Yaghi, O. M. The atom, the molecule, and the covalent organic framework. *Science* **2017**, *355* (6328), eaal1585.
- (80) 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.; Tang, C.; Thompson, S.; Parker, J.; Trewin, A.; Bacsá, J.; Slawin, A.

M. Z.; Steiner, A.; Cooper, A. I. Porous organic cages. *Nat. Mater.* **2009**, 8 (12), 973–978.

(81) Gosselin, A. J.; Rowland, C. A.; Bloch, E. D. Permanently Microporous Metal-Organic Polyhedra. *Chem. Rev.* **2020**, 120 (16), 8987–9014.

(82) Liota, E.; Spyrou, G.; Vassilatis, D. K.; Cournia, Z. Structure-based virtual screening for drug discovery: principles, applications and recent advances. *Curr. Top. Med. Chem.* **2014**, 14 (16), 1923–1938.

(83) Gali, A.; Fyta, M.; Kaxiras, E. Ab initio supercell calculations on nitrogen-vacancy center in diamond: Electronic structure and hyperfine tensors. *Phys. Rev. B* **2008**, 77 (15), 155206.

(84) Calcatera, N. E.; Barrow, J. C. Classics in Chemical Neuroscience: Diazepam (Valium). *ACS Chem. Neurosci.* **2014**, 5 (4), 253–260.

(85) Bernales, V.; Ortuño, M. A.; Truhlar, D. G.; Cramer, C. J.; Gagliardi, L. Computational design of functionalized metal-organic framework nodes for catalysis. *ACS Cent. Sci.* **2018**, 4 (1), 5–19.

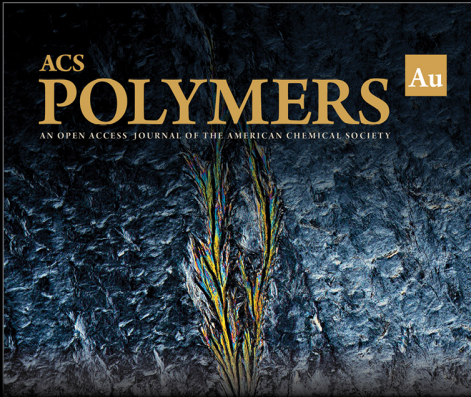
(86) Chu, C.; Huang, D.; Gupta, S.; Weon, S.; Niu, J.; Stavitski, E.; Muhich, C.; Kim, J.-H. Neighboring Pd single atoms surpass isolated single atoms for selective hydrodehalogenation catalysis. *Nat. Commun.* **2021**, 12, 5179.

(87) Chung, Y. G.; Gómez-Gualdrón, D. A.; Li, P.; Leperi, K. T.; Deria, P.; Zhang, H.; Vermeulen, N. A.; Stoddart, J. F.; You, F.; Hupp, J. T.; Farha, O. K.; Snurr, R. Q. In silico discovery of metal-organic frameworks for precombustion CO₂ capture using a genetic algorithm. *Sci. Adv.* **2016**, 2 (10), e1600909.

(88) Collins, S. P.; Daff, T. D.; Piotrkowski, S. S.; Woo, T. K. Materials design by evolutionary optimization of functional groups in metal-organic frameworks. *Sci. Adv.* **2016**, 2 (11), e1600954.

(89) Lee, J.; Han, W.-S.; Kasperovics, R.; Lee, J.-H. An in-depth comparison of subgraph isomorphism algorithms in graph databases. *Proc. VLDB Endow.* **2012**, 6 (2), 133–144.

(90) Yan, X.; Yu, P. S.; Han, J. Substructure similarity search in graph databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*; 2005; pp 766–777, DOI: 10.1145/1066157.1066244



ACS
POLYMERS Au
AN OPEN ACCESS JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

Editor-in-Chief: **Prof. Shelley D. Minteer**, University of Utah, USA

Deputy Editor:
Prof. Arthi Jayaraman
University of Delaware, USA

Open for Submissions 

pubs.acs.org/polymerau

 **ACS Publications**
Most Trusted. Most Cited. Most Read.