Supplementary Information

Efficiently mapping structure-property relationships of gas adsorption in porous materials: Application to Xe adsorption

Alec R. Kaija,[a] Christopher E. Wilmer[a]*

[a]Department of Chemical & Petroleum Engineering,
University of Pittsburgh, Pittsburgh, PA 15261, USA

*To whom correspondence should be addressed. E-mail: wilmer@pitt.edu

Table of contents

Section S1. Randomly-generating pseudomaterials

Section S2. Simulation details

Section S3. Parent-selection algorithm

Section S4. Mutation algorithm

S1. Randomly-generating pseudomaterials

Pseudomaterials are hypothetical material-analogues consisting of Lennard-Jones (LJ) spheres randomly positioned within a unit cell. We developed an algorithm to generate these pseudomaterials by randomly-selecting values to define their structural characteristics. For each parameter, a value was selected from the ranges presented Table S1.1:

Table S1.1 Ranges of values used to define pseudomaterial structural parameters.

Parameter	Minimum	Maximum
ε/k_b (K)	1.258	513.264
σ (Å)	1.052	6.549
Number density (atoms/Å^3)	0.0000149	0.02122
Lattice constant (Å)	25.6	51.2

To generate a pseudomaterial, first three crystallographic lattice parameters (*a*, *b*, *c*) were selected in each of the *x*, *y*, and *z*-directions. All crystal lattices were cubic, so all corresponding angles (*α*, *β*, *γ*) were 90°. Once the unit cell volume could be calculated, a number density was randomly-selected to determine the number of atom-sites with which to populate the unit cell. A lower limit was imposed so that each unit cell would be populated with at minimum two atom-sites or that the number density was at least 0.0000149 atoms/Å^3. Each of these atom-sites was then assigned a random position within the unit cell. Chemical species, or pseudoatom-types, were defined by randomly-selected LJ parameters, *σ* and *ε*. In this study each pseudomaterial was comprised of at most four different pseudoatom-types. Once all of the atom-sites were positioned within the unit cell, each was assigned a pseudoatom-type from one of the four available.

S2. Simulation details

Xenon loading. Grand canonical Monte Carlo (GCMC) simulations were used to calculate Xe loading according to previously established methods.[1,2] In three separate runs, GCMC simulations were performed at 298K at various pressures (1 bar, 5 bar, and 10 bar). Each calculation used a 1000 cycle equilibration period followed by a 1000 cycle production run. A cycle consists of n Monte Carlo steps, where n equals the number of molecules in the system (which fluctuates). Simulations included insertion, deletion, and translation moves of molecules with equal probabilities. Fugacities were calculated using the Peng-Robinson equation of state. Xe loadings were reported as the absolute volumetric amount of gas in the pseudomaterial's void space.

Void fraction. GCMC simulations were also used to calculate void fraction in each pseudo material according to established methods.[2] Each simulation consisted of a 1000 cycle production run. Simulations used the Widom particle insertion method with a helium atom probe (σ=2.96Å). Average Rosenbluth weight was recorded for the void fraction of each pseudomaterial.

Interaction potential. Interaction energies from non-bonded atoms were computed through the Lennard-Jones potential:

$$V_{ij} = 4\epsilon_{ij}\left(\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right)$$

where i and j are interacting atoms, r_{ij} is the distance between atoms i and j, σ_{ij} is the LJ diameter, and ε_{ij} is the LJ well depth. The LJ cutoff was 12.8 Å. Coloumbic interactions were ignored because of the electrostatic nature of the adsorbate, Xe. LJ parameters for framework pseudoatoms were selected at random from a range of values 50% larger than the UFF; values for Xe also come from the UFF. Table S2.1 presents the LJ parameters used for simulations in this study:

Table S2.1 LJ parameters for framework pseudoatoms and Xe adsorbate atoms.

Atom type	**σ (Å)**	**ε/k_B (K)**
pseudoatoms	1.052 – 6.549	1.258 - 513.264
Xe	3.69	110.704

S3. Parent selection algorithm

In an attempt at encouraging new pseudomaterials with unique structure property combinations, rather than randomly-generating new pseudomaterials, unique pseudomaterials (already present in the library) are selected as parents and mutated. To select unique parents, first, the parameter-space was divided into 100 (10x10) congruent bins, which were weighted per the number of pseudomaterials in each bin:

$$w_i = \frac{\sum c_i}{c_i} = \text{bin-specific weight}$$

$$w_i^* = \frac{w_i}{\sum c_i} = \text{normalised bin-specific weight}$$

where c_i is the bin-count, or number of pseudomaterials in a particular bin. The normalized bin specific weights, w_i, were used then in the weighted random selection of the parent-bin. From the parent-bin one pseudomaterial was selected as a parent, where any pseudomaterial may be selected with equal probability. For more information on the random selection function used see the official Numpy documentation.[3]

S4. Mutation algorithm

Mutating pseudomaterials creates new materials by perturbing the values used to define the structural characteristics of some parent-pseudomaterial. As describe previously (Section S1), a pseudomaterial is defined by a collection of crystal lattice parameters (a, b, c), some number of atom-site positions, and four pseudoatom-types (LJ parameters, σ and ε), one assigned to each atom-site. Mutation simply refers to the process of systematically perturbing each of these values:

$$x_{new} = x_{old} \pm M * U([x_{min}, x_{max}])$$

where M is the mutation strength parameter and $U([x_{min},x_{max}])$ is a random variable from a uniform distribution across the limits for the value being perturbed. Perturbations are performed by selecting a random value and then selecting a new value between the original and random values. The greater the mutation strength, the closer the new value will be to the random value, such that a mutation strength of 0% creates a clone of the parent pseudomaterial and 100% creates a completely random pseudomaterial with no structural similarity to its parent. For more information on the uniform selection function used see the official Numpy documentation.[4]

References:

1 A. Ö. Yazaydın, R. Q. Snurr, T.-H. Park, K. Koh, J. Liu, M. D. LeVan, A. I. Benin, P. Jakubczak, M. Lanuza, D. B. Galloway, J. J. Low and R. R. Willis, *J. Am. Chem. Soc.*, 2009, **131**, 18198–18199.

2 C. E. Wilmer, O. K. Farha, Y.-S. Bae, J. T. Hupp and R. Q. Snurr, *Energy Environ. Sci.*, 2012, **5**, 9849.

3 https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.choice.html

4 https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.uniform.html