Zeoran: Generating zeolite frameworks with aluminum substitutions

Here we describe a simple software to generate an arbitrary number of zeolite structures where the position of the Al atoms are sampled from four possible distributions. The source code and all the required files to run it can be found in the following GitHub repository: `zeoran`.

Generating zeolite frameworks

Here we consider zeolites as rigid molecules. Hence, a configuration is simply given by a boolean vector of size $\mathcal{T}$ (the amount of T-atoms of the zeolite). A 0 (1) in the n-th position of this vector corresponds to having a Si (Al) atom in the n-th T site. The methods used to generate these structures are fully based on graph theory. Namely, we read the structure of a given zeolite and keep only the bonds between the atoms, loosing in principle all the spatial information. However, since the edges in our graph represent chemical bonds, there is actually a relation between closeness in the graph and in the real space.

Since the oxygen positions are common in all the possible frameworks (no matter the Si/Al ratio), we delete the oxygen atoms from the graph, keeping only the T-atoms. Simply put, we turn the bonds T-O-T into T-T, and we refer to the remaining graph as T-graph. Then, from the T-graph we can generate specific configurations starting from the pure silica framework.

We have developed four different algorithms to generate these structures, each of them imposing different restraints on the framework. First, we focused on creating frameworks with multiple chains of consecutive aluminum atoms. For each of these chains we first find the base atom, corresponding to the first atom of the chain. Then, the chain is completed by adding aluminum atoms in T-sites that are bonded to previous atoms of the chain. Atoms belonging to different chains cannot be first neighbours in the T-graph, since that would mean that the two chains that they belong to would merge in a longer one. Notice that all Lowensteinian configurations are composed only by chains of length 1.

Second, we have also developed an algorithm to generate zeolite frameworks with local spatial clusters of aluminum atoms. Even though in our T-graph the spatial position of the atoms is not explicitly encoded, we mentioned that there is a relation between distance within the graph and distance in the real space. Therefore, an easy but effective way to create clusters from a given root atom, is to put aluminum atoms in all its neighbours, and then in the second neighbours, third neighbours, etc. until we had introduced the desired number of aluminum atoms.

Third, we can also consider completely random Al configurations, meaning that the Al sites are sampled from a uniform distribution among all available T-sites. The algorithm to implement this method is straightforward.

Lastly, given a certain number of aluminum atoms N, we want to stochastically generate configurations that are opposed to the clustered structures: structures where the aluminum substitutions are spread within the framework. In physics or in information theory, we would informally say that we aim to find the most disordered structures (that is, the structures that maximizes the entropy of the system). To this end, we developed an algorithm based on the so-called stochastic process maximal entropy random walk (MERW). In the next sections, we briefly explain this process and how it can be used to generate structures that maximizes the entropy of the system.

Maximal entropy random walks

A MERW is a type of biased random walk on a graph where the transition probabilities are chosen accordingly to the principle of maximal entropy, which in that context states that all paths in the graph are equally probable. Formally, let $\mathcal{G}$ be a simple connected graph and let A be its adjacency matrix. Since A is symmetric, it can be diagonalized and its eigenvalues are real. In the base where A is diagonal, we can write

$$A = \sum_{i=1}^{n} \lambda_i \psi_i \psi_i^T \tag{0.1}$$

where λ_i are the eigenvalues of A and $\psi_i^T = (\psi_{i,1}, \ldots, \psi_{i,n})$ the associated eigenvectors. Furthermore, from the Perron-Frobenius theorem, we may assum that the dominant (the greatest) eigenvalue of A is unique. Hence, it is clear that for big values of l, we can approximate $A^l \approx \lambda^l \psi \psi^T$. For the reasoning coming, we will use the following property (which can be proved by doing induction on n).

Proposition 0.1. *Let A be the adjacency matrix of a graph $\mathcal{G}$ and i and j two vertices of $\mathcal{G}$. Then, the element (i, j) of the matrix A^n corresponds to the number of different paths of length n from the vertex i to the vertex j.*

Now, our aim is to compute the probability S_{ij} of going from the vertex i to the vertex j in one step.[1] To this end, let us compute the number of paths m_{il} centered in the vertex i and having length of $2l$ for big values of l. If the extremes j and k of those paths where given, by Proposition 0.1 the number of paths would be $(A^l)_{ji} \cdot (A^l)_{ik}$. Thus, adding for all possible values of j and k and using Eq. (0.1), we have that

$$m_{il} = \sum_{j=1}^{n} \sum_{k=1}^{n} (A^l)_{ji} \cdot (A^l)_{ik} \approx \lambda^{2l} \psi_i^2 \left(\sum_{j=1}^{n} \psi_j \right)^2 . \tag{0.2}$$

[1]Formally, we can write that $S_{ij} = P(X_n = j | X_{n-1} = i)$, where X is the random walk and n the discrete time variable. We will see that S_{ij} depends only on the graph and not explicitly on the time.

Next, let us denote by ρ_i the probability of a given path of infinite length to be centered in the vertex i. Surely, ρ_i can be computed as the quotient between the number of paths of infinite length centered in i over the number of all paths of infinite length

$$\rho_i = \lim_{l\to\infty} \frac{m_{il}}{\sum_{k=1}^{n} m_{kl}} = \frac{\psi_i^2}{\psi^2}. \tag{0.3}$$

Notice that by considering that the center of the path corresponds to the present time step, then ρ_i stands for the probability of being at the vertex i.

Analogously, we can compute the number of paths of length $2l+1$ for big values of l having the adjacent vertices i and j in the center:

$$m_{(i,j)l} = \sum_{r=1}^{n}\sum_{k=1}^{n} (A^l)_{ri} \cdot A_{ij} \cdot (A^l)_{jk} \approx \lambda^{2l} \psi_i A_{ij} \psi_j \left(\sum_{k=1}^{n} \psi_k\right)^2 . \tag{0.4}$$

Then, the probability ρ_{ij} of being at j and going to i in the subsequent step reads

$$\rho_{ij} = \lim_{l\to\infty} \frac{m_{(i,j)l}}{\sum_{k=1}^{n}\sum_{r=1}^{n} m_{(k,r)l}} = \frac{\psi_i A_{ij} \psi_j}{\sum_{k=1}^{n}\sum_{r=1}^{n} \psi_k A_{kr} \psi_r} = \frac{\psi_i A_{ij} \psi_j}{\lambda \psi^2}. \tag{0.5}$$

Finally, recalling Eqs. (0.3, 0.5) and the definition of conditioned probability, we obtain that

$$S_{ij} = \frac{\rho_{ij}}{\rho_i} = \frac{A_{ij}}{\lambda} \frac{\psi_j}{\psi_i}. \tag{0.6}$$

Notice that the transition probability matrix of the random walk S_{ij} is stochastic matrix, that can be obtained by computing the eigenvalues of the adjacent matrix of the graph. To illustrate how this matrix maximizes the entropy of the system, we consider the following example (see Figure 1). Consider a 2D random walk in a closed region of the $\mathbb{Z}^2$ space. Furthermore, consider that this close region has some defects, which can be understood as walls that the walker cannot cross. These walls are represented in green in Figure 1. Now, consider that the walker initially starts in the center position (i.e. in the region that is almost closed to the outside by the walls). Let us analyse the behaviour of the walker when two different models are used.

First, consider a random walk where in every position, it is equally probable to make the next move in all the cardinal directions available (North, East, South and West). We call this the standard random walk (SRW). In this case, it is clear that the walker will require very long times in average to leave from the region of the middle, since there is only one tiny exit. Indeed, in Figure 1 (a) we see that before leaving, the walker already visited all the other positions, and most likely more than once. Actually, we found that in average the random walker needs 3652.7 steps to get to the outside of the green region.

Now, consider that the walker is performing a MERW. Clearly, the vast majority of the paths that can be followed in this graph are outside from the green region. Hence, since all paths are now equally probable, the transition probability will be such that the walker is pushed to the outside of this region. This can be seen in Figure 1 (b), which shows that the walker is able to find its way out relatively fast. Indeed, it takes him on average just 19.4 steps to get out, which is significantly a smaller number as in the case of the SRW.

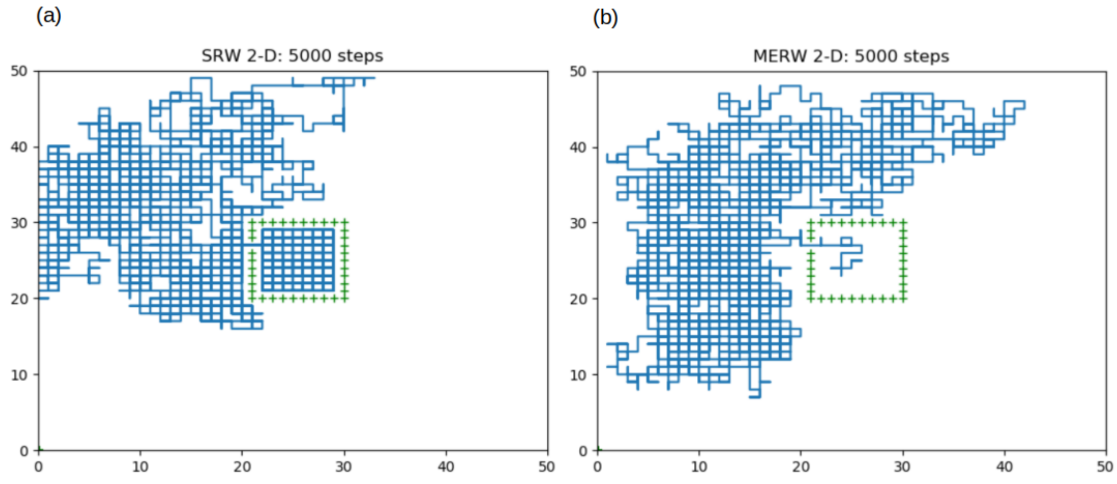

Figure 1: *5000 steps random walk simulations for both SRW and MERW in a lattice with defects. Defects are shown in green and the path of the random walk is shown in blue. The random walks start at the position (25,25), which corresponds to the center of the whole graph.*

Maximal entropy zeolite structures

In this section we explain how the MERWs can be used to define random structures with a given amount of aluminum atoms in such a way that these atoms are spread along the structure. We consider the all silica framework as starting point, and we first choose randomly one T-site, where we will put the first aluminum substitution. Then, we start an iterative process, that we will carry out until the number of substitutions equals the input value:

1. We remove the T site where we have an aluminum now from the graph. In the notation introduced in the last section, this corresponds to putting a wall in the corresponding vertex of the graph.

2. We compute the new transition probability matrix.

3. We select randomly an available vertex (not a wall) and start a maximal entropy random walk from there for a certain number of equilibration steps `Neq`.

4. Once the equilibration has finished, the random walker keeps wlaking until a vertex is visited for a certain number of times `Nvisits`. This vertex is where we add the next aluminum atom.