Supporting Information

Geometrical Properties Can Predict CO_2 and N_2 Adsorption Performance of Metal-organic Frameworks (MOFs) at Low Pressure

Michael Fernandez* and Amanda S. Barnard

CSIRO Virtual Nanoscience Laboratory

343 Royal Parade, Parkville 3052

Victoria, Australia.

Contents

*michael.fernandezllamosa@csiro.au

1 Selection of principal components, clusters and archetypes

The selection of the optimum number of principal components (PC), clusters and archetypes was performed by analysing the amount of explained data variance as a function of the number of components in each analysis in Figure S1.

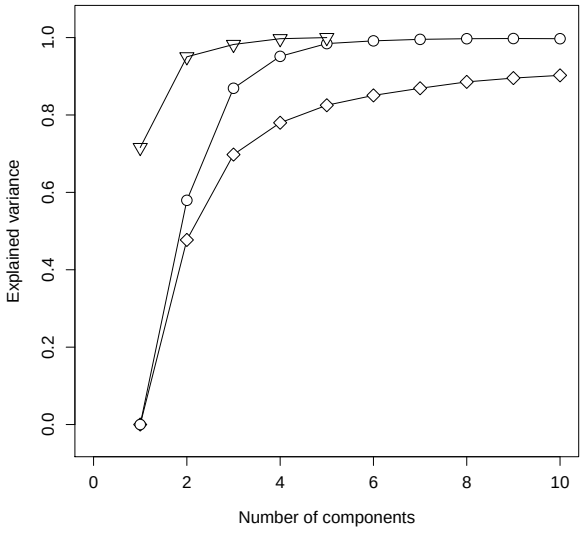
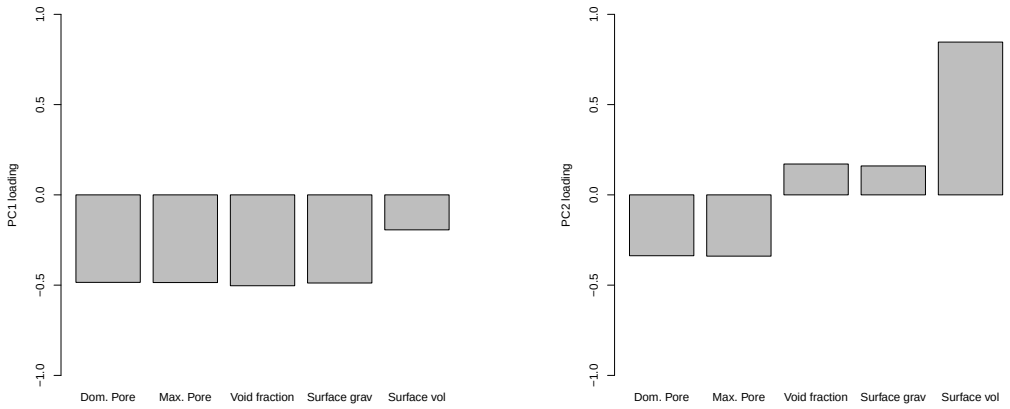


Figure S1: Amount of described variance as function of the number of PCA ($\bigtriangledown$), *k*-means ($\diamond$) and AA ($\circ$).

2 Contribution of the geometrical features to the principal components (PCs)

The contributions of each geometrical features to the two main PCs of the MOFs appear in Figure S2.



(a) (b)

Figure S2: Loading of each geometrical features in the a) PC1 and b) PC2.

3 Machine learning

3.1 Linear multiple regression (MLR)

Linear regression represents a regression that is linear in the unknown parameters used in the fit. We used the most common form of linear regression known as least squares fitting. [1]

3.2 Decision tree (DTree)

Decision trues are binary rule-based modeling technique that typically uses an attribute selection search to construct binary rules of different combinations of attributes. A decision tree model approximate the electronic properties of graphene as rudimentary decision rules based on the

values of a number of attributes, with the number and specific types of attributes can vary to suit the needs of the task. Despite their simplicity, decision tree has shown to perform fairly well with the added value of ease interpretation given the number of rules are not very large.[2]

3.3 K-nearest-neighbour(kNN)

K-nearest-neighbour (kNN) modeling uses normalized Euclidean distance to predict each instance as the average value of a certain number of closest training instance. It eliminates the need for building models but the zero training time comes at the expense of a large amount of time needed to compute the distances between all the data instances in the training data.[3]

3.4 Support vector machines (SVM)

Support vector machines (SVMs)[4] are a machine learning method of broad applicability to many types of pattern recognition problems based on the structural risk minimization principle from statistical learning theory. A detailed description of SVMs and SRM is available in.[4] The input vectors are first mapped onto one feature space (possibly with a higher dimension) by means of a kernel function. Then, a hyperplane is built to separate the positive and negative examples within this feature space. Only relatively low-dimensional vectors in the input space and dot products in the feature space will evolve by a mapping function. Different types of kernels can be used in SVM models, including linear, polynomial, radial basis function (RBF), and sigmoid. Of these, the RBF kernel is the most recommended and popularly used, since it has finite response across the entire range of the real x-axis. There are several important parameters in SVM, including the kernel parameters and the regularization parameters that must be externally tuned.

3.5 Artificial Neural Networks (ANNs)

ANNs is a computer-based model in which a number of processing elements, also called neurons, units or nodes are interconnected by links in a netlike structure forming "layers".[5] A variable value is assigned to every neuron. The neurons can be one of three different kinds. The input

neurons form the input layer, which receives their values by direct assignation and are associated with independent variables, with the exception of the bias neuron. The hidden neurons collect values from the input neurons, giving a result that is passed to a non-input neuron. Finally, the output neurons collect values from other units and correspond to different dependent variables, forming the output layer. The links between units have associated values, named weights that condition the values assigned to the neurons. There exist additional weights assigned to bias values that act as neuron value offsets. The weights are adjusted through a training process in order to minimize network error. Neural networks are adjusted, or trained, so that a particular input leads to a specific target output.

3.6 Random forests (RF)

Random forests (RF) are a ensemble learning method that train a multitude of decision trees and output the mode of the classes in classification or mean prediction in regression of the individual trees. RF improve over decision trees with respect to overfitting to the training set. The algorithm for inducing a random forest combines "bagging" and the random selection of features in order to construct a collection of decision trees with controlled variance.[6]

4 List of prototypes and archetypes structures

Table S1: List of MOFs prototype structures

ID	Dominant Pore	Maximum Pore	Void Fraction	$Surface_{grav}$	$Surface_{vol}$	CO_2	N_2
5024922	18.75	20.25	0.890	4891.41	1442.86	low	low
5040209	6.25	7.75	0.621	1730.32	1810.40	low	low
24205	4.25	5.25	0.399	690.56	871.70	low	high
5020072	12.75	14.25	0.831	4548.25	1953.73	low	low
5013797	8.25	9.75	0.741	3368.52	2239.85	low	low

Table S2: List of MOFs archetype structures.

ID	Dominant Pore	Maximum Pore	Void Fraction	$Surface_{grav}$	$Surface_{vol}$	CO_2	N_2
8372	4.75	6.25	0.663	3250.00	3116.19	low	low
3001288	24.75	24.75	0.934	5806.41	906.50	low	low
1001792	3.25	4.75	0.182	166.71	256.01	low	low
34903	5.75	6.25	0.880	813.60	1762.05	high	low
5036495	9.75	10.75	0.855	6947.27	2350.62	low	low

5 Test set prediction accuracy as a function of the size of the training set

The test set prediction accuracies of the optimum RF models for increasing size of the training set appear in Figure S3.

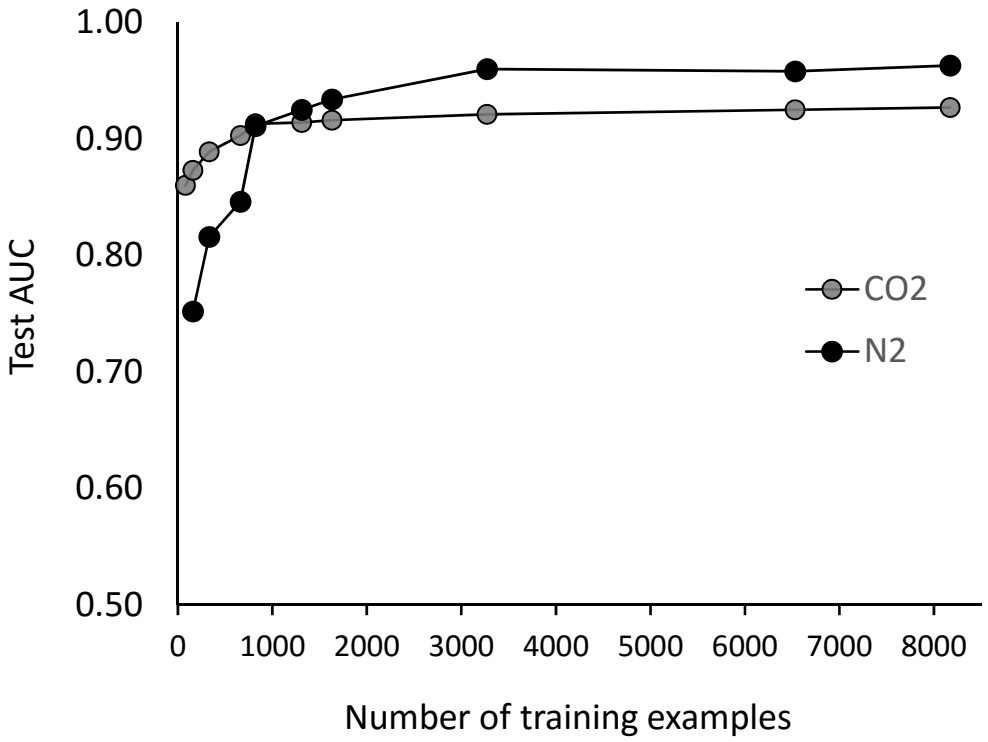


Figure S3: Test set prediction accuracy as a function of the number of training examples.

References

[1] Edwards, A. *An introduction to linear regression and correlation*; W. H. Freeman and Comp: San Francisco, 1977.

[2] Quinlan, J. R. *C4.5: Programs for Machine Learning*; California: Morgan Kaufmann Publishers: San Mateo, 1993.

[3] Aha, D. W.; Kibler, D.; Albert, M. K. Instance-based learning algorithms. *Mach. Learn.* **1991**, *6*, 37–66.

[4] Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297.

[5] Bishop C, *Neural Networks for Pattern Recognition*; Oxford University Press, USA, 1995.

[6] Breiman, L. Random forests. *Mach. Learn.* **2001**, 5–32.