Autonomous Exploration and Identification of High Performing Adsorbents using Active Learning

Gaël Donval *[1], Calum Hand *[1,2], James Hook *[3], Emiko Dupont[3], Malena Sabaté Landman[3], Melina A. Freitag[3], Matthew J. Lennox[1,2], and Tina Düren[1,2]

[1] *Centre for Advanced Separations Engineering, Department of Chemical Engineering, University of Bath, Bath, BA2 7AY, United Kingdom*
[2] *EPSRC Centre for Sustainable and Chemical Technologies (CSCT), University of Bath, Bath, BA2 7AY, United Kingdom*
[3] *EPSRC Centre for Doctoral Training in Statistical Applied Mathematics (SAMBa) and Department of Mathematical Sciences, University of Bath, Bath, BA2 7AY, United Kingdom*

** These authors contributed equally.*

Abstract

MOFs and COFs are porous materials with a large variety of applications including gas storage and separation. Synthesised in a modular fashion from distinct building blocks, a near infinite number of structures can be constructed and the properties of the material can be tailored for a specific application. While this modularity is a very attractive feature it also poses a challenge. Attempting to identify the best performing material(s) for a given application is experimentally intractable. Current research efforts combine molecular simulations and machine learning techniques to evaluate the simulated performance of hundreds of thousands of materials to identify top performing MOFs and COFs for a given application. These approaches typically rely on moderated brute-force screening which is still resource-intensive as typically between 70 - 100 % of the hundreds of thousands of materials must be simulated to create a training set for the machine learning models used, restricting screening to relatively simple molecules. In this work we demonstrate our novel Bayesian mining approach to materials screening which allows 62 - 92 % of the top 100 porous materials for a range of applications to be readily identified from large materials databases after only assessing less than one percent of all materials. This is a stark contrast to the 0 - 1 % achieved by conventional brute-force screening where porous materials are just chosen at random during a high throughput screening. Through this accelerated virtual screening process, the identification of high performing materials can be used to more rapidly inform experimental efforts and hence lead to an acceleration of the entire research and development pipeline of porous materials.

1 Introduction

Metal-organic frameworks (MOFs) and covalent organic frameworks (COFs) are highly modular, porous materials which have been studied for a range of applications due to their modularity allowing for development of bespoke materials [1, 2]. This modularity is a double-edged sword however, as their large number presents a bottleneck in the research and development pipeline. For instance, tens of thousands of experimental structures have been published in the Cambridge structural database [3] and hundreds of thousands of hypothetical structures have been created computationally — e.g. in the hypothetical Metal Organic Framework (hMOF) [4] and the hypothetical Covalent Organic Framework (hCOF) [5] databases. If the goal is to identify the highest performing material for a given application, then a vast number of MOFs or COFs must be studied in order to suitably explore the material space and identify top performing materials.

The systematic evaluation of materials via experimental high throughput screening (HTS) is highly resource-intensive both in terms of time and money. To mitigate these costs, computational based HTS has become prevalent since it allows for MOF and COF structures to be screened against a target application on a drastically reduced timescale without the associated costs of synthesising

and characterising the materials. Computational HTS for adsorption applications typically makes use of grand-canonical Monte Carlo (GCMC) based molecular simulations [6] and has been used in the screening of MOFs and COFs for numerous gas storage and separation applications [5, 7–15].

Computational HTS has allowed for the systematic evaluation of hundreds of thousands of structures found in materials databases like the hMOF database and hCOF database. The computational costs associated with routinely performing GCMC screenings on such a large scale has however become prohibitive, due to the length of time and computational power required to simulate each material. To cut down on the number of materials which must be simulated, many researchers are beginning to incorporate machine learning (ML) into their materials screenings as summarised in the recent review by Jablonka *et al.* [16].

ML uses a set of mathematical tools to approximate the relationship between independent input variables $\mathbf{x}$ and corresponding dependent output variable y. By observing prior training samples, it determines a suitable "approximate function" $\hat{f}$ that maps each input $\mathbf{x}$ to a corresponding approximate output $\hat{y} = \hat{f}(\mathbf{x})$. Ideally $\hat{y} = y$ but in practice $\hat{y}$ is close to but not equal to y due to the approximate nature of $\hat{f}$. Once developed, this approximate function allows for the prediction of the dependent value even for previously unseen instances of $\mathbf{x}$.

As an example, in the context of finding the most suitable porous materials for methane storage, a ML model could be trained to relate the amount of methane adsorbed by a MOF (output y) from several descriptors such as void fraction and surface area of that MOF (inputs $\mathbf{x}$). Making a prediction with a trained ML model is quicker by several orders of magnitude than molecular simulations, facilitating even larger HTS.

Most ML based screening studies use 70 - 80 % of the available materials to train the model [16] corresponding to tens of thousands of simulations. The remaining 20 - 30 % of the data is then used to validate the trained model. This approach does not scale to scenarios where the required simulations are computationally expensive in the first instance, for example due to the examination of complex and/or mixed adsorbates as part of the HTS. In order to investigate these scenarios then, a new approach is required to evaluating large material databases.

Finding an optimal MOF or COF for an application is not an easy task. Molecular simulations are "black box" functions in terms of parameters optimisation: there is no analytical expression trivially linking a given structure (or features of those structures) to a quantifiable property such as gas adsorption. Conducting molecular simulations is also expensive enough that it is not tractable on databases containing tens of thousands of structures; moreover these kind of simulations provide noisy data. An elegant approach would be to replace the non-analytical black box simulations by a well-behaved analytical function whose statistical properties and derivatives are known. Such a "surrogate" function can then be used as a good general interpolator in place of the black box function. One of the more popular approaches to do exactly that is Bayesian optimisation (BO) which uses Gaussian processes as the surrogate model, supplemented by an acquisition function to select materials for testing [17–19]. The Gaussian processes provide interpolated values with uncertainties and the acquisition function represents the typical exploration/exploitation trade-off.

A further advantage of BO is that it allows the model to learn as a scientist would: in a continuous, iterative process capable of autonomously selecting which material to investigate next. As such it is a good example of active learning (AL), a type of ML where the specific goal is to rapidly train a reliable ML model in an iterative fashion rather than a batch process requiring large volumes of data upfront [17–19].

The application of BO in materials science is limited to a small number of publications applying the technique to materials screening [20–23]. Indeed, in a recently published review on the application of ML in the study of porous materials by Jablonka *et al.* only four paragraphs out of 64 pages discussed the application of AL [16]. For HTS of over 100,000 materials for various applications, BO based active learning represents an attractive approach for materials screening [23–25].

In this paper, we demonstrate the successful application of BO assisted material screenings for several adsorption targets using our recently developed AL framework, the Autonomous Materials

Investigator (AMI). To demonstrate the robustness of the AMI, we performed three separate autonomous HTS on different porous material targets:

1. Methane storage in COFs, evaluated using the deliverable capacity (working volume of methane which can be adsorbed by a COF) [5]
2. Methane storage in COFs, evaluated via the Wiersum adsorption performance indicator (API) [26]
3. Carbon dioxide/nitrogen separation (carbon capture) in MOFs, evaluated by a bespoke metric based on work by Wilmer *et al.* [14]

For all three HTS, the features used to describe the materials are investigated and the rate of acquisition of top performers by the AMI in each study is assessed.

2 Methodology

2.1 The Autonomous Materials Investigator (AMI)

The AMI conducts a HTS by first selecting a number of randomly sampled materials from the given material database and determines their performance with full-fledged molecular simulations. After this initial sampling, the AMI trains its surrogate model, a Gaussian processes regressor, from the features of the sampled materials (descriptive input parameters) and their target scores (their performance in the given adsorption application). The surrogate model then provides Bayesian predictions of the performance scores of the remaining untested structures using existing data from the already-tested structures. These predictions are then used to identify the most-likely top performers from the remaining materials, which are then sampled using one of several acquisition functions, and tested by running a molecular simulation on the sampled material. The results of the molecular simulation are finally added to the training set for a next model fitting. The process of conducting Bayesian predictions and sampling likely top performing materials then repeats until a predetermined number of materials have been sampled and investigated (figure 1).

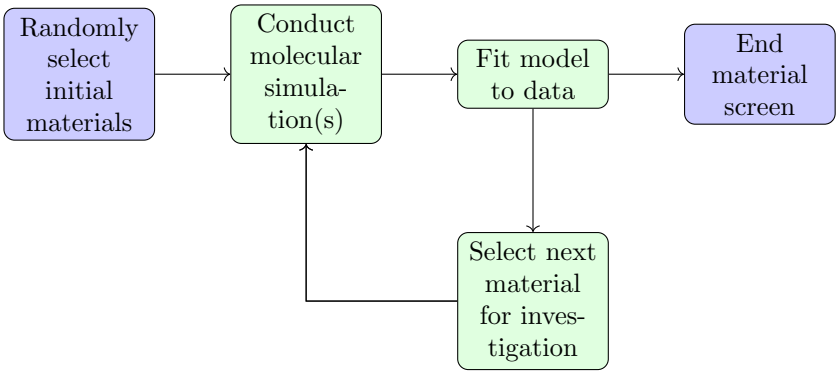

Figure 1: Overview of AMI screening using molecular simulations to investigate materials selected by model. The iterative sampling loop is highlighted green.

As more structures are tested, the AMI's knowledge is regularly updated which refines the feature-score relationship within the surrogate model. Thus, the AMI continually updates the approximate function $\hat{f}$ as it iteratively samples individual materials. The AMI's major advantage is that it autonomously samples materials from the database, while training itself, allowing to identify the absolute top performers after probing only a small fraction of the input database. Therefore, it does not require an entire, pre-computed, database (in contrast to other ML approaches as per Jablonka *et al.* [16]). In doing so, full simulations are performed for each sampled material, always providing reliable data instead of ML approximates.

In the search for the top performers, it is vital for the AMI to strike a balance between *exploitation*

(testing of materials similar to those already sampled with the highest predicted scores), and *exploration* (testing of materials which the AMI is more uncertain about in terms of their predicted performance) when sampling materials for investigation. Testing materials with dissimilar features (exploration) is essential for exploring the whole materials space and ensuring that the AMI is not stuck testing only similar materials while ignoring other high performing materials with very different features. Equally however, exploration can result in testing a high proportion of low performing materials.

In traditional Bayesian optimisation, the objective is to find the single highest value corresponding to the material with the largest performance indicator for the application considered. However, in screening for adsorption applications we are interested in identifying a subset of high performing materials to study in more detail (e.g. using more computationally expensive simulations or experiments) to determine feature-performance relationships more generally and find a collection of best performing materials. This broadened identification of top performing materials is demonstrated through the use of two novel approaches we refer to as "Bayesian mining" [27]. Our Bayesian mining algorithms - Greedy tau, and Greedy N - either target all structures with a performance score above some threshold (e.g. the methane storage target set by the US Department of Energy) or all of the structures in some chosen fraction of the full database (e.g. the top 100). This latter target has the advantage that it requires no user input or prior knowledge of the range of expected performance scores. Here, we compare our two Greedy algorithms against two conventional Bayesian acquisition functions, expected improvement (EI) and Thompson sampling, which both search for the single best material in the entire database [28].

In order to provide a proof of principle of the method, we used pre-existing brute-force screening results from a range of different screening scenarios as detailed in section 2.3. We looked up the corresponding material performance value from the published data, which replaced conducting a molecular simulation on the sampled material as this allowed for repeated runs of the AMI to be conducted in significantly less time than with full simulations. It is worth noting that even though we focus on molecular simulations of adsorption here, the framework can be easily adapted to any process where autonomous screening would be beneficial.

2.2 Features

As our aim is to predict the adsorption performance of MOFs and COFs, we used a set of physical and chemical properties as descriptive features that are typically used to characterise porous materials such as chemical composition, surface area, void fraction and pore diameter. In addition, we computed topological features, which provide a characterisation of the shape of the pore space. A full list and description of the features are given in table S1.

The topological features were calculated following the persistent homology method first used by Lee *et al.* [29] with the notable difference that we performed calculations on the material structure only. This makes our topological features an intrinsic property of the structure that is independent from the choice of adsorbate, allowing it to be re-used as a feature in different adsorption screening scenarios. For simplicity, we applied the method to a $3 \times 3 \times 3$ supercell. For a handful of structures the use of that fixed number of cells led to the violation of the minimum image convention (maximum persistent homology length scale of radius 4 Å, see section S6). These structures were simply discarded. Full details are given in supporting information S6.

Prior to being used in the AMI, for each target assessed, all feature values were standardised by subtracting the mean value of the feature within the target database and dividing this value by the standard deviation of the feature. This is a standard pre-processing step in ML so that all features are provided to the model in a similar numerical range.

2.3 Scores

We used the following three adsorption quantities as ML *scores* throughout this paper *i.e.* the numerical values used to assess the performance in each application. They were chosen as they

represent application focused adsorption challenges currently being investigated and also require increasingly more nuanced features for the AMI to suitably model the feature-score relationship.

Deliverable capacity Firstly, we use the volumetric deliverable capacity, given in equation 1 for methane stored at 65 bar and released at 5.8 bar at 298 K as simulated by Mercado *et al.* [5] for 69,840 hypothetical COFs:

$$\mathrm{DC}_{\mathrm{CH_4}} = N_{65\,\mathrm{bar}} - N_{5.8\,\mathrm{bar}}. \tag{1}$$

DC is the deliverable capacity and N is the volumetric amount of methane adsorbed in v(STP)/v.

Wiersum adsorbent performance indicator (API) Secondly, we used the same dataset from Mercado *et al.* and considered a modified version of Wiersum *et al.* 's adsorbent performance indicator (equation 2) which was originally developed for assessing the adsorption of mixtures [26]. The Wiersum API is an engineering performance metric which recognises that the optimum material for an adsorption process will be a compromise between the deliverable capacity and the heat of adsorption (ΔH_{ads}) which indicates the energy required to release stored methane for utilisation. The heat of adsorption is sometimes used as a *feature* in gas adsorption predictions [30, 31] but is intentionally included as a component of the target in this paper to ensure that the feature input for each material contains only properties that can be calculated *a priori* for the whole database and can be reused for other applications. Hence, all data that rely on GCMC simulations and are application-specific are only considered as scores:

$$\mathrm{API}_{\mathrm{Wiersum}} = \frac{\mathrm{DC}_{\mathrm{CH_4}}}{\Delta H_{\mathrm{ads}}}. \tag{2}$$

Selectivity API To test the AMI on a more demanding application, we used data published by Wilmer *et al.* for N_2 and CO_2 mixture adsorption in the hMOF database [14]. Here we chose a mixture containing mole fractions of $y_{\mathrm{CO_2}} = 0.1$ and $y_{\mathrm{N_2}} = 0.9$ adsorbed at 1 bar and desorbed at 0.1 bar at 298 K; the conditions used by Bae and Snurr to represent flue gas separation by vacuum swing adsorption [32]. Note that the data, although used to screen MOFs for mixture separation, are based on pure component simulations which are much quicker to run than mixture simulations [14]. While it has been noted that the the original force fields used by Wilmer *et al.* [14] potentially caused over-prediction of adsorption in fluorine containing MOFs [33], the hMOF data was used in this work without any further treatment.

In order to identify high-performing MOFs for carbon dioxide capture from flue gas, we used an adsorption performance indicator based on the CO_2/N_2 selectivity of the MOF. Selectivity is in general defined as per equation 3

$$S_{i,j} = \frac{x_i/x_j}{y_i/y_j}. \tag{3}$$

where x_i is the mole fraction of component i in the pore and y_i is the mole fraction in the bulk. For our case looking at a mixture of 10 mole-percent CO_2 and 90 mole-percent N_2 this becomes equation 4, with a slight modification:

$$\hat{S}_{\mathrm{CO_2,N_2}} = 9 \times \frac{N_{\mathrm{CO_2}}}{N_{\mathrm{N_2}} + 1}. \tag{4}$$

We added 1 to the denominator to avoid division by zero in cases where the MOF did not take up any N_2 during the simulation. It is worth noting that this change does not alter the relative rankings of selectivities.

The heat of adsorption was not included in the data of Wilmer *et al.* [4]. Our selectivity API (API_S) given in equation 5 therefore only combines two important evaluation criteria for adsorbents in pressure swing applications: the selectivity under adsorption conditions, and the working (or deliverable) capacity of CO_2

$$\mathrm{API}_S = \log\left(1 + (\hat{S} \times \mathrm{DC}_{\mathrm{CO}_2})\right) \tag{5}$$

The resultant distribution of values inside the log is monomodal and tailed towards high values [34, 35]. As Gaussian processes assume normally distributed targets, we took the logarithm of those values API_S to reduce the effect of tailing and increase the symmetry of their distribution.

2.4 Experimental Design

The efficiency of four acquisition functions *Greedy tau*, *Greedy N*, *Thompson*, and *EI* were assessed for each of the previous three performance scores until a given percentage of the overall database had been sampled. We chose to stop our sampling at about 1 % of the entire database in each case. This corresponds to 500 COFs from the entire hCOF database [5] and 950 MOFs from the hMOF database [14]. These totals include the 50 samples which were selected randomly to initialise the AMI, after which the AMI updated itself with each further sample taken.

In addition to the acquisition functions, the impact on the AMI of using distinct feature groups physical (P), chemical (C) and topological (T), and combinations of these groups (PT, PC, CT) relative to the whole available feature space (PCT) was also assessed. A full list of the features in the different feature groups available in each database is given in table S1. The feature matrix resultant from these different feature usages were all brought up to the same total number of features as the full PCT feature matrix for each target by the inclusion of "white noise" features, containing random values for each material, sampled from a standard normal distribution. This ensured the same number of features would be present in each tested group, allowing for fairer comparison between results. To benchmark the performance of each feature group combination, a feature matrix containing only white noise (R) was also created, to which the AMI attempted to model the relationship with the given target values. Feature groups rather than singular features were assessed here as the AMI requires a lengthy, iterative process of sampling and fitting in order to "fully" train. This notably increased the training time and therefore prohibited conventional, univariate feature selection techniques used in typical ML scenarios [16].

Each combination of acquisition function and feature group was tested 16 times, each time starting with a different set of randomly sampled materials in the initialisation process, so as to obtain reliable average performance values and demonstrate the consistency of the AMI's ability. Confidence intervals were generated by bootstrap sampling — with replacement — the maximum number of top performing structures identified and taking the 0.5% and 99.5% quantiles of the resulting sample means (i.e. confidence level of 99%).

3 Results and Discussion

3.1 Feature Comparison

Selecting the features which describe the training examples in a data set is an important process in any ML workflow. Different problems require different features to suitably model the relationship, however the number of features should ideally be constrained due to the negative impacts of the number of features on model training times [16]. This is especially true when using a Gaussian process surrogate model as its training time is more negatively impacted than typical ML models with large numbers of features [36]. Hence the first stage of demonstrating the AMI in the three target scenarios is the identification of features which allow for suitable mapping of the COF and MOF structures to the target values. Due to the AL approach employed by the AMI, however, the relative importance of the features were expected to change throughout the screening as different groups of materials were sampled by the AMI in the discussed exploration/exploitation trade-off.

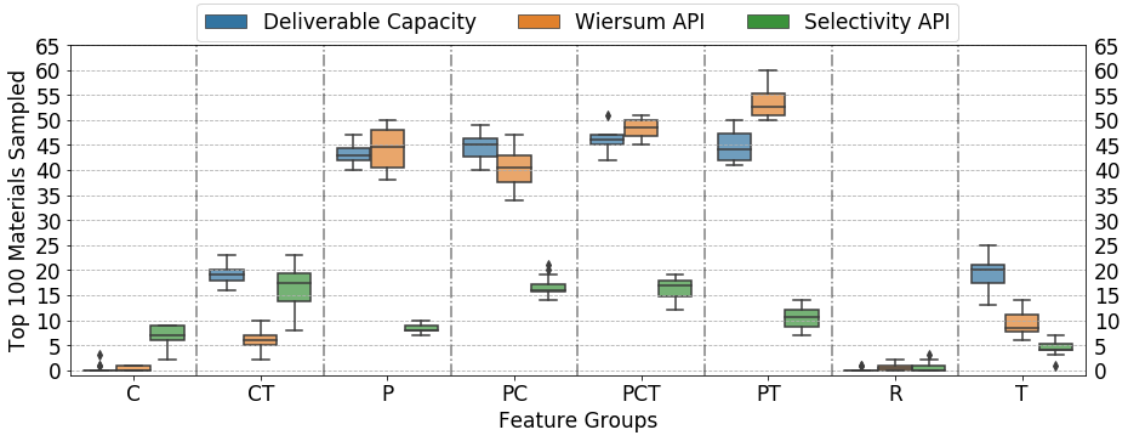

Figure 2: Box plots of AMI performance for each target scenario using different feature groups. Each box plot is constructed from 16 repeats with all materials sampled using the EI acquisition function. Each repeat consists of 500 AL iterations for COFs or 950 AL iterations for MOFs ($\sim$ 1% of the respective databases).

As the purpose of this work is to sample highly-performant materials from large databases, the metric used to assess the impacts of different features was the number of top 100 performing materials sampled in the total number of materials sampled by the AMI using conventional EI sampling. As the performance values of each screening scenario are taken from prior studies and are therefore known, it was possible to use this benchmark. The 50 materials randomly sampled to initialise the AMI were discarded from the assessment so that only materials intentionally sampled by the AMI would be considered.

From the results of the feature assessment in figure 2, a general trend was observed for the methane targets determined for the hCOF database, where physical features dominated with some benefit from the inclusion of topological features and a negligible contribution from the chemical features which performed as well as random noise. Overall, for the deliverable capacity target, the feature group making use of all features (figure 2 PCT) had allowed the AMI to sample a median of 46 (44-47 at 99% confidence level) top 100 performers after only 500 materials sampled. As each AMI sampling represents an unknown material being simulated in order to obtain its deliverable capacity, the power of the AMI is immediately apparent with almost 1 in 10 materials sampled being in the top 100 COFs from a database containing over 69,000 entries. The AMI sampling of the hCOF database for the Wiersum API target was even more successful, with a median of 52 (51-55 *idem*) top 100 performers sampled. Here the best performance of the AMI was achieved in the absence of chemical features only taking into account physical and topological features (figure 2 PT). The preference for the AMI in using feature group PT over PCT for the Wiersum API sampling was likely due to the limited ability of binary chemical features (element present or not in the structure) to convey sufficient information on the target and hence "misled" the AMI in this instance. The notably higher number of top performers sampled was likely due to fewer COFs satisfying the physical requirements of being top performers, and hence providing narrower distributions of values which allowed the AMI to sample these materials more easily. This can be seen in figure S1 where the distributions of physical features are much tighter for the top performing Wiersum API and deliverable capacity targets.

In contrast to the trend noted for the hCOF adsorption targets, the feature group resulting in the highest AMI performance for the hMOF Selectivity API target was the combination of chemical and topological features (figure 2 CT). This group had earlier lead the AMI to under-perform compared to physical feature groups but now allowed for 17 (13-18 *idem*) top 100 MOFs to be sampled. Compared to the results obtained by the AMI for sampling COFs for different methane based adsorption targets, the success of the AMI in this screening scenario is still notable as 17 top 100 performing MOFs were sampled from over 130,000 in only 950 AMI iterations. This significantly outperforms conventional brute-force screenings, particularly when the complexity of

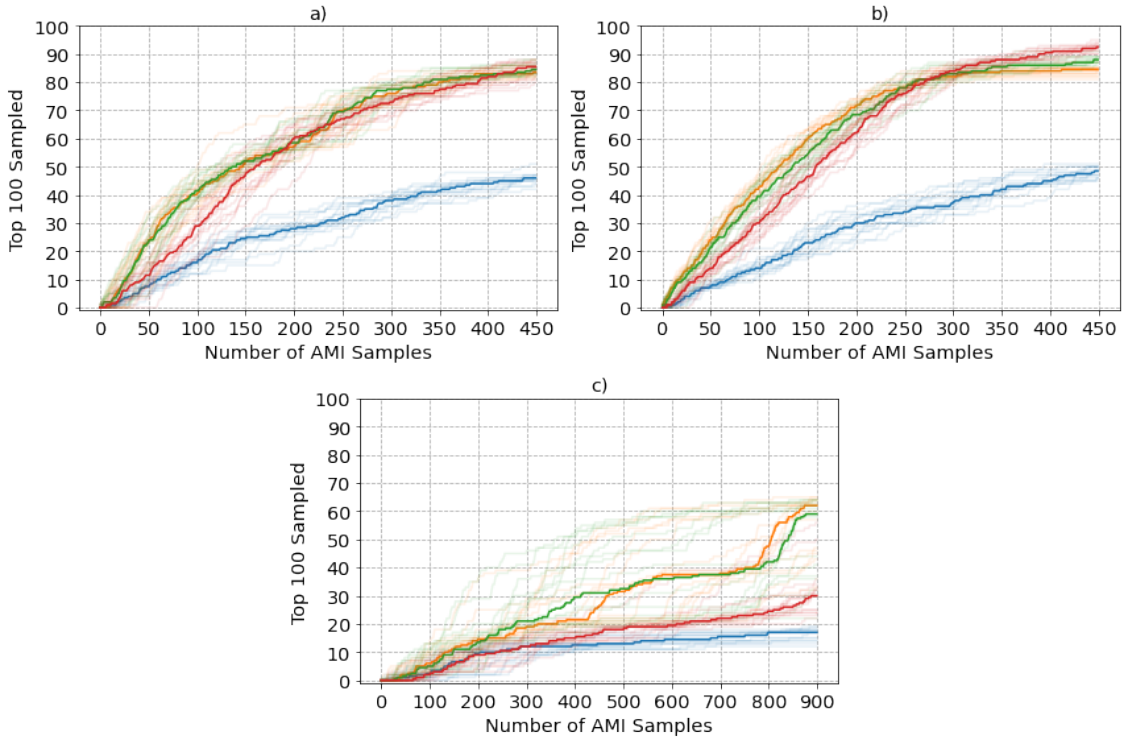

Figure 3: Comparison of AMI acquisition rate of top 100 performing materials for different acquisition functions: EI (blue), Thompson (red), Greedy N (green), Greedy tau (orange). The repeats for each acquisition function assessment are shown as faint lines, with the median sampling as a function of AMI iteration shown by the bold line. Targets shown are: a) methane deliverable capacity (hCOF), b) Wiersum API for methane (hCOF), c) Selectivity API for CO_2 and N_2 (hMOF).

the target (selective adsorption of CO_2 and N_2) is considered.

For both hCOF and hMOF targets, the trends observed using the EI acquisition function with different feature groups were also observed with the other acquisition functions with the exception of some distributions being notably wider or tighter (see supporting information S5).

3.2 Bayesian Mining with AMI

The different AMI screening functions EI, Thompson, Greedy N, and Greedy tau were compared with the AMI able to access all features available in the PCT group for the relevant target. Again, each acquisition function study was repeated 16 times, the median result of which was found in order to visualise the non-skewed number of top 100 performing materials sampled by the AMI as a function of AMI iterations (materials sampled exclusively by the AMI), presented in figure 3. Any top performing materials sampled during the initial random sampling were omitted as they do not reflect the abilities of the acquisition functions.

For both of the COF screening targets (a and b) it can be seen in figure 3 that, with the exception of EI, the AMI was able to consistently sample 84 (83-86 at 99% confidence level) and 92 (91-93, *idem*) of the top 100 performing COFs for deliverable capacity and Wiersum API respectively after only 450 AMI-guided samples (500 total). This is a stark contrast with conventional brute-force screenings where most of the 69,000 COFs have to be simulated to identify the top range of materials.

For the selectivity API target studied on the hMOF database (figure 3c), overall lower numbers of top performing materials were identified compared to the methane targets but using the Greedy acquisition functions still resulted in an average of 56 (49-62 at 99% confidence level) top performing materials being sampled by the AMI after 900 iterations (950, including initialisation). Given

the selectivity API describes the selectivity of both CO_2 and N_2 within a MOF, it is altogether a much more complex target than the methane based COF targets, and hence demonstrates an equally if not more impressive achievement. The greater spread of individual curves seen in figure 3c compared to the two other cases is likely caused by the the additional uncertainties on the MOF features (section 3.1). Overall it seems that the chemical features do provide some insight into the hMOF database, but do not fully capture the significance and hence yields inconsistent results for MOFs and COFs.

Across all four of the AMI acquisition functions and three target scenarios assessed as part of this work, the Greedy acquisition functions and the conventional Thompson were found to consistently outperform EI. This is likely due to EI's specific exploration/exploitation balancing skewed towards exploration. We would expect EI to perform better in more diverse datasets where exploration would be better rewarded.

Both Greedy functions behaved very similarly to each other in all cases presented here. They consistently performed better than the other two acquisition functions at low sample counts and were seen to be overtaken by Thompson at larger sample counts. The better behaviour at lower sample counts can be explained by their propensity to exploitation, as hinted by the steps observed for those functions around 200 samples in figure 3a as well a 400 and 800 in figure 3c. Those steps typically occur as the local pool of well-performing materials gets depleted and the algorithm starts to explore again.

Thompson always exhibited a slower onset and only outperformed the Greedy algorithms at high sample count in the case of methane (hCOF) on the ranges we studied. Thompson is based on ML model sampling: with few samples, it is heavily biased towards exploration but as the number of samples increases, it leans more and more towards exploitation. This both explains the slower uptake at the start but the steeper slope in figures 3a/b at the highest sample counts. It is also more sensitive to noisy data and non-normal data distributions which likely explains its worse performance in figure 3c compared to the Greedy approach.

At this stage, it is important to acknowledge that counting the number of top 100 performers, while very helpful to prove the AMI works, is both arbitrary and impossible to do in a real run, when data is calculated on-the-fly, and thus, the performance across the entire dataset is not known. To solve this problem, we compared the distributions of target values obtained through ML to a reference distribution (here calculated on the full dataset but in real runs, could be obtained by random sampling of the dataset). The data, shown in figure 4, were presented as cumulative distribution function (CDF) (i.e. the cumulative integral of the more usual probability density functions frequently seen as histograms). At a given target value (i.e. performance level), the CDF is the proportion of the sampled dataset with performance at most that target value. The advantage of this representation is that the lower and the more to the right the curve is, the more performant the acquisition function is overall for the number of samples considered. Said differently, these graphs show how well the different ML algorithms skew their selection towards better performing materials.

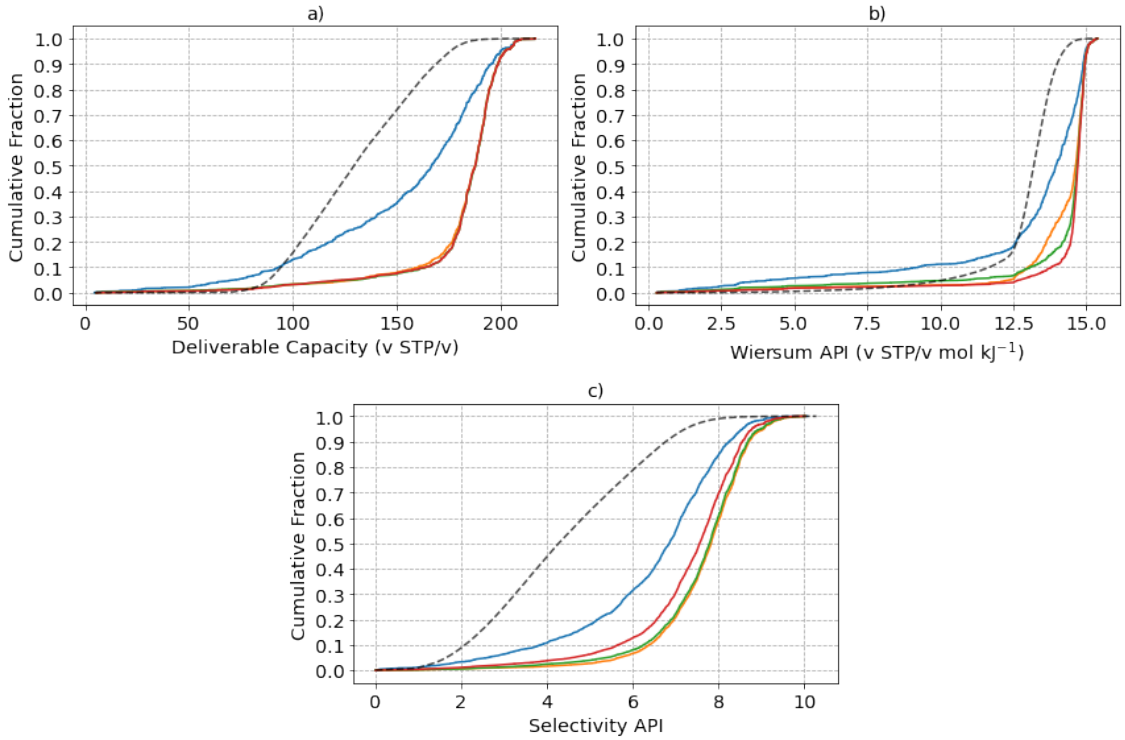

Figure 4: Cumulative distribution function of target values of materials sampled by the AMI using different acquisition functions: expected improvement (blue), Thompson (red), Greedy N (green), Greedy tau (orange). The CDF of the target values for the entirety of each data set is also presented for comparison (black dash). Targets shown are: a) methane deliverable capacity (hCOF), b) Wiersum API for methane (hCOF), c) Selectivity API for CO_2 and N_2 (hMOF). Each plain curve is the total CDF of all 16 repetitions over 450 samples each for hCOF and 900 samples each for hMOF.

As noted earlier, the EI acquisition function is consistently performing the worst among acquisition functions, with some oversampling of the lowest 10 - 15 % (compared to our reference) due to the more exploratory nature of EI. Once again, there might be datasets where EI's approach could turn out to be an advantage though it is not the case here.

For COF deliverable capacity target (figures 4a/b), the Greedy and Thompson acquisition functions explored nearly identical regions of the database. Some reproducible discrepancies are observed between Greedy and Thompson when considering the Wiersum API. Though small, those differences illustrate both Greedy's propensity to exploitation and the bias we introduced in choosing the top 100 in 3c.

The selectivity API data (figure 4c) illustrates the same point: though Thompson eventually identified only half of the top 100 materials that Greedy methods identified at the end of the 900 samples, this way of looking at the data show that Thompson actually stays very competitive: it did not identify the absolute best performing materials as well but still performed well at identifying best-performing materials in general. Just like with EI, it is very probable that a more varied dataset would favour Thompson.

4 Conclusions

In this work, three target applications for the hCOF and hMOF databases were assessed for the first time using a novel Bayesian mining approach. Our approach with the AMI was able to consistently sample 84 (83-86 at 99% confidence level) and 92 (91-93, *idem*) percent of top 100 performing COFs for deliverable capacity and Wiersum API, respectively, and 56 (49-61, *idem*) percent of top 100 performing MOFs after assessing less than one percent of the database in each

target instance (500, and 950 materials, respectively). This undoubtedly demonstrates the power of our Bayesian mining approach over brute-force high throughput screening or conventional ML assisted high throughput screening, both requiring tens or hundreds of thousands of materials to be simulated for each target investigated. This dramatic reduction in materials that need to be assessed for their performance opens the possibility to conduct screening of large databases for complex applications. We anticipate that our approach will enable the investigation of applications of porous materials which currently require complex molecular simulations to assess performance and are simply infeasible using current approaches. While we used previously published data as a proof of principle in this paper, the AMI can be readily combined with different simulation programs to predict the performance of a material or even with experimental results. To encourage the adoption of our Bayesian mining approach within porous materials research, the AMI is to be released as an open source software package, designed to integrate with the "Raspa" molecular simulations software [37] (see supporting information). By combining the power of our Bayesian mining approach to material screening and the opportunity for other members of the research community to adapt the AMI to their needs, it is hoped that AMI will revolutionise how high throughput screening of porous materials is conducted.

5 Conflicts of Interest

The authors declare no conflict of interest.

6 Acknowledgements

Funding was received from the Engineering and Physical Sciences Research Council EP/L016354/1. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 648283 GROW-MOF). This work was also supported by a scholarship from the EPSRC Centre for Doctoral Training in Statistical Applied Mathematics at Bath (SAMBa), under the project EP/L015684/1. This research made use of the Balena High Performance Computing (HPC) Service at the University of Bath.

References

(1) Z. Ajoyan, P. Marino and A. J. Howarth, *CrystEngComm*, 2018, **20**, 5899–5912.
(2) S. Cao, B. Li, R. Zhu and H. Pang, *Chemical Engineering Journal*, 2019, **355**, 602–623.
(3) J. Goldsmith, A. G. Wong-Foy, M. J. Cafarella and D. J. Siegel, *Chemistry of Materials*, 2013, **25**, 3373–3382.
(4) C. E. Wilmer, M. Leaf, C. Y. Lee, O. K. Farha, B. G. Hauser, J. T. Hupp and R. Q. Snurr, *Nature Chemistry*, 2012, **4**, 83–89.
(5) R. Mercado, R.-S. Fu, A. V. Yakutovich, L. Talirz, M. Haranczyk and B. Smit, *Chemistry of Materials*, 2018, **30**, 5069–5086.
(6) D. Frenkel and B. Smit, *Understanding Molecular Simulation*, Elsevier, Croydon, 2nd, 2002.
(7) R. Babarao, R. Custelcean, B. P. Hay and D.-E. Jiang, *Crystal Growth & Design*, 2012, **12**, 5349–5356.
(8) M. Tong, Y. Lan, Q. Yang and C. Zhong, *Chemical Engineering Science*, 2017, **168**, 456–464.
(9) T. Yan, Y. Lan, M. Tong and C. Zhong, *ACS Sustainable Chemistry & Engineering*, 2019, **7**, 1220–1227.
(10) W. Li, Y. Pang and J. Zhang, *Journal of Molecular Modeling*, 2014, **20**, 2346.
(11) Z. Qiao, C. Peng, J. Zhou and J. Jiang, *Journal of Materials Chemistry A*, 2016, **4**, 15904–15912.
(12) N. S. Bobbitt, J. Chen and R. Q. Snurr, *The Journal of Physical Chemistry C*, 2016, **120**, 27328–27341.

(13) D. A. Gomez, J. Toda and G. Sastre, *Physical Chemistry Chemical Physics*, 2014, **16**, 19001–19010.
(14) C. E. Wilmer, O. K. Farha, Y.-S. Bae, J. T. Hupp and R. Q. Snurr, *Energy & Environmental Science*, 2012, **5**, 9849.
(15) C. Altintas and S. Keskin, *ACS Sustainable Chemistry & Engineering*, 2019, **7**, 2739–2750.
(16) K. M. Jablonka, D. Ongari, S. M. Moosavi and B. Smit, *Chemical Reviews*, 2020, **120**, 8066–8129.
(17) P. I. Frazier, *arXiv*, 1807.02811, 2018, 1–22.
(18) D. Packwood, *Bayesian Optimization for Materials Science*, Springer Singapore, Singapore, 2017, vol. 3.
(19) T. Lookman, P. V. Balachandran, D. Xue, J. Hogden and J. Theiler, *Current Opinion in Solid State and Materials Science*, 2017, **21**, 121–128.
(20) D. Xue, P. V. Balachandran, R. Yuan, T. Hu, X. Qian, E. R. Dougherty and T. Lookman, *Proceedings of the National Academy of Sciences of the United States of America*, 2016, **113**, 13301–13306.
(21) T. Lookman, P. V. Balachandran, D. Xue and R. Yuan, *npj Computational Materials*, 2019, **5**, 21–38.
(22) D. Xue, P. V. Balachandran, J. Hogden, J. Theiler, D. Xue and T. Lookman, *Nature Communications*, 2016, **7**, 1–9.
(23) P. V. Balachandran, D. Xue, J. Theiler, J. Hogden and T. Lookman, *Scientific Reports*, 2016, **6**, 1–9.
(24) Y. G. Chung, D. A. Gómez-Gualdrón, P. Li, K. T. Leperi, P. Deria, H. Zhang, N. A. Vermeulen, J. F. Stoddart, F. You, J. T. Hupp, O. K. Farha and R. Q. Snurr, *Science Advances*, 2016, **2**, 1–9.
(25) H. Ohno and Y. Mukae, *The Journal of Physical Chemistry C*, 2016, **120**, 23963–23968.
(26) A. D. Wiersum, J.-S. Chang, C. Serre and P. L. Llewellyn, *Langmuir*, 2013, **29**, 3301–3309.
(27) J. Hook, C. Hand and E. Whitfield, *arXiv*, 2009.05418, 2020, 1–14.
(28) J. M. Hernández-Lobato, J. Requeima, E. O. Pyzer-Knapp and A. Aspuru-Guzik, 2017.
(29) Y. Lee, S. D. Barthel, P. Dłotko, S. M. Moosavi, K. Hess and B. Smit, *Nature Communications*, 2017, **8**, 15396.
(30) H. Liang, W. Yang, F. Peng, Z. Liu, J. Liu and Z. Qiao, *APL Materials*, 2019, **7**, 091101.
(31) Z. Shi, H. Liang, W. Yang, J. Liu, Z. Liu and Z. Qiao, *Chemical Engineering Science*, 2020, **214**, 115430.
(32) Y.-S. Bae and R. Q. Snurr, *Angewandte Chemie International Edition*, 2011, **50**, 11586–11596.
(33) M. Fernandez and A. S. Barnard, *ACS Combinatorial Science*, 2016, **18**, 243–252.
(34) L. A. Aroian, *The Annals of Mathematical Statistics*, 1947, **18**, 265–271.
(35) D. V. Hinkley, *Biometrika*, 1969, **56**, 635–639.
(36) C. E. Rasmussen and C. K. I. Williams, in *Gaussian Processes for Machine Learning*, The MIT Press, Boston, 2006, ch. Regression, pp. 7–30.
(37) D. Dubbeldam, S. Calero, D. E. Ellis and R. Q. Snurr, *Molecular Simulation*, 2016, **42**, 81–101.

Supporting Information - Autonomous Exploration and Identification of High Performing Adsorbents using Active Learning

Gaël Donval *[1], Calum Hand *[1,2], James Hook *[3], Emiko Dupont[3], Malena Sabaté Landman[3], Melina A. Freitag[3], Matthew J. Lennox[1,2], and Tina Düren[1,2]

[1] *Centre for Advanced Separations Engineering, Department of Chemical Engineering, University of Bath, Bath, BA2 7AY, United Kingdom*
[2] *EPSRC Centre for Sustainable and Chemical Technologies (CSCT), University of Bath, Bath, BA2 7AY, United Kingdom*
[3] *EPSRC Centre for Doctoral Training in Statistical Applied Mathematics (SAMBa) and Department of Mathematical Sciences, University of Bath, Bath, BA2 7AY, United Kingdom*

* *These authors contributed equally.*

Contents

S1 AMI Model and Availability

The AMI uses a Gaussian process regressor with an RBF kernel and constant mean function as its surrogate model. The Gaussian process regressor, kernel, and mean function were implemented in the GPy python library [1] and used as part of the AMI back end with no modification made to the GPy code.

As Gaussian processes become notably slower when using larger data sets [2] we used a sparse sampling strategy to mitigate this. Our strategy was to condense the entire feature matrix into only 500 data points (referred to as "inducing points"):

- 300 points were the centroids found when performing K-Means clustering on a sample of 5000 data points from the full feature matrix (as implemented in scikit-learn [3]). The distance calculations when clustering were scaled using the length scales determined so far.
- 100 points were the features of the data points with the highest predicted means (predicted using the GPy backend model).
- 100 points were the features of the data points with the highest predicted variance (predicted using the GPy backend model).

This combination of data points allowed us to capture the main regions of the data base being investigated (centroids), the region of database to exploit (high mean), and the region of database to explore (high variance). When a prediction is needed for the entire feature matrix, the euclidean distances between the points in the feature matrix and the inducing matrix are calculated and moderated by applying a squared exponential. The calculated distance matrices were then transformed to allow predictions to be made by the AMI more rapidly for the full feature matrix.

The AMI library is being finalised before release but can be found on gitlab "AMInvestigator/ame".

S2 Features

The physical features used were taken from the hMOF [4] and hCOF [5] databases respectively . Chemical features were determined from the provided element density in both databases as shown:

$$P_{\mathrm{x}} = \begin{cases} 1, & \rho_x > 0 \\ 0, & \rho_x = 0 \end{cases}$$

Table S1: Features used by the AMI with units shown where appropriate. Binary chemical features are denoted as P_{x}. The feature groups are also specified.

Group	HCOF	HMOF
Physical	void fraction	void fraction
	density ($kg\ m^{-3}$)	density ($t\ m^{-3}$ normalised by 1 $t\ m^{-}$
	surface area ($m^2\ g^{-1}$)	gravimetric surface area ($m^2\ g^{-1}$)
	largest included sphere diameter (Å)	volumetric surface area ($m^2\ cm^{-3}$)
	largest free sphere diameter (Å)	max pore diameter (Å)
	largest included sphere along free sphere path diameter (Å)	dominant pore diameter (Å)
Chemical	P_{F}	P_{F}
	P_{H}	P_{H}
	P_{N}	P_{N}
	P_{O}	P_{Cl}
	P_{S}	P_{Br}
	P_{Si}	P_{V}
	-	P_{Cu}
	-	P_{Zn}
	-	P_{Zr}
Topological	topological feature 1	topological feature 1
	topological feature 2	topological feature 2
	topological feature 3	topological feature 3
	topological feature 4	topological feature 4
	topological feature 5	topological feature 5

S3 Feature Analysis

Tabulated values of the median number of top 100 materials sampled using Expected Improvement.

Table S2: Median number of top 100 materials sampled for each target after 500 sampled materials for deliverable capacity and Wiersum API targets, and 950 for Selectivity API target. Materials were sampled using "Expected Improvement".

Features	Deliverable Capacity	Wiersum API	Selectivity API
PCT	46	48	17
PC	45	40	16
PT	44	52	10
P	43	44	8
T	20	8	4
CT	19	6	17
C	0	0	7
R	0	0	0

S4 Distributions of Physical Features for hCOF Database

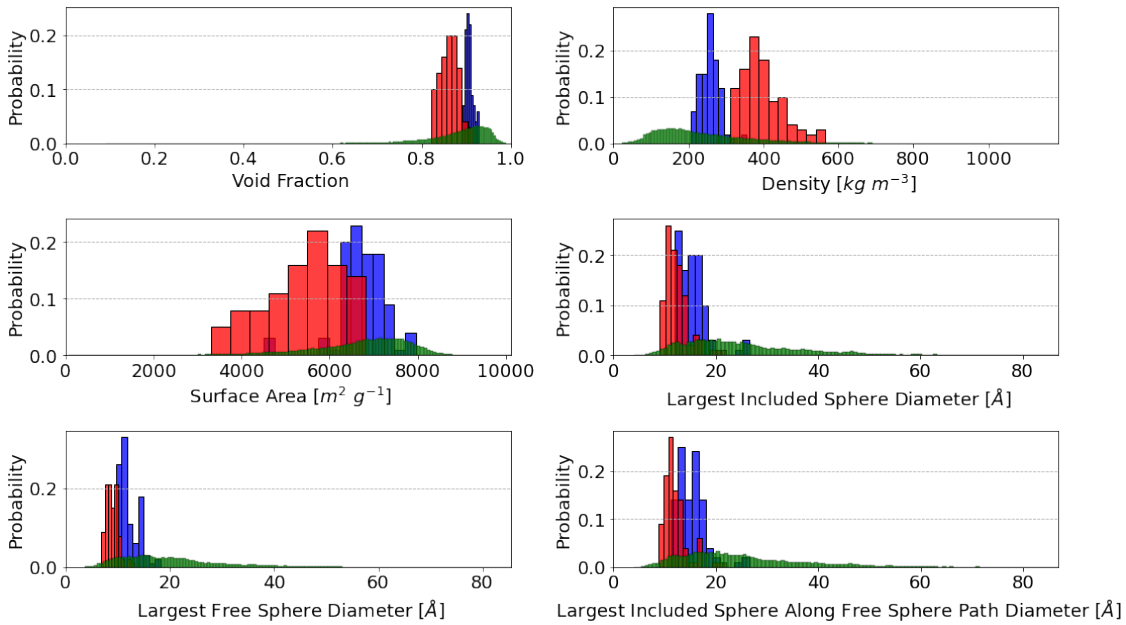

Figure S1: Histograms of individual physical features for the hCOF database. Values for COFs with the top 100 Wiersum API values (blue). COFs with the top 100 deliverable capacity values (red). The entire hCOF database (green). Each histogram is normalised such that the total heights of the bars is one.

S5 AMI Performance with Different Feature Groups and Acquisition Functions

Box plots showing the distribution of number of top 100 materials sampled using a given set of features (x axis) and acquisition function (box colours). Each target explored in this work is presented as a separate box plot.

hCOF - Deliverable Capacity

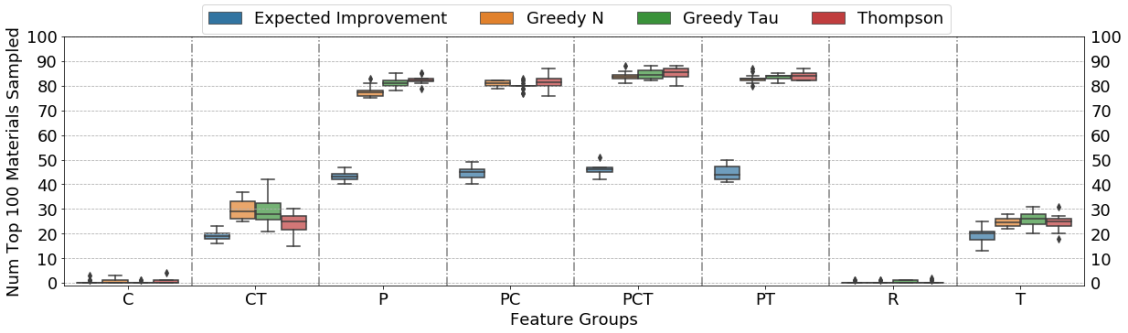

Figure S2: Box plots showing the number of sampled COFs with top 100 Deliverable Capacity values sampled by the AMI. Each feature group shows the maximum sampling achieved by each acquisition function and is constructed from 16 repeats.

hCOF - Wiersum API

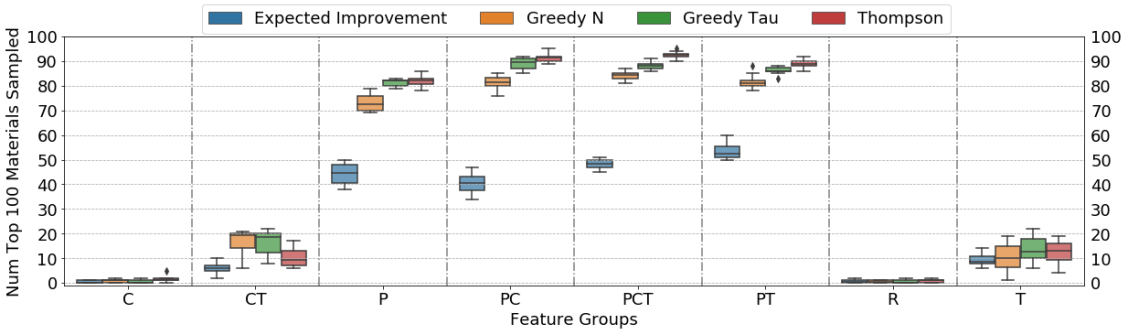

Figure S3: Box plots showing the number of sampled COFs with top 100 Wiersum API values sampled by the AMI. Each feature group shows the maximum sampling achieved by each acquisition function and is constructed from 16 repeats.

hMOF - Selectivity API

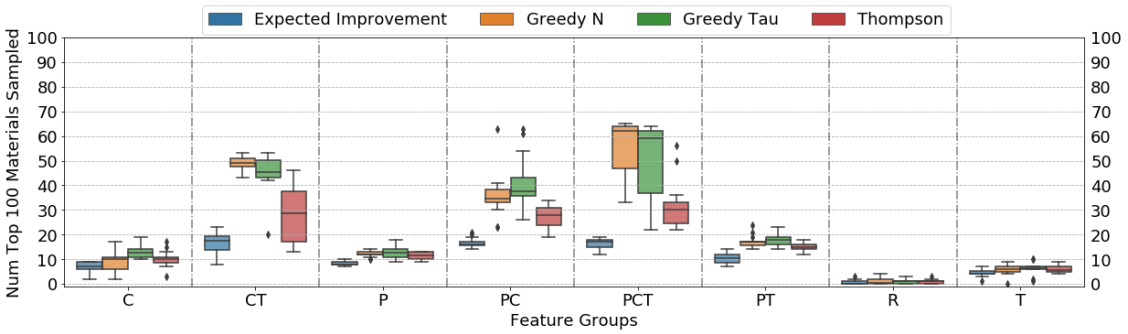

Figure S4: Box plots showing the number of sampled MOFs with top 100 Selectivity API values sampled by the AMI. Each feature group shows the maximum sampling achieved by each acquisition function and is constructed from 16 repeats.

Tabulated Values

Tabulated values of plots shown in section S5

Table S3: Median values of the number of top 100 materials sampled for each target with the associated acquisition function and feature group combination.

Acquisitor	Features	Deliverable Capacity	Wiersum API	Selectivity API
Thompson	**PCT**	85	92	30
	PC	81	91	28
	CT	25	9	28
	PT	84	89	15
	P	82	82	11
	C	0	1	10
	T	25	13	5
	R	0	1	1
Greedy Tau	**PCT**	84	88	59
	CT	28	18	45
	PC	80	89	37
	PT	84	86	18
	P	81	82	12
	C	0	0	12
	T	26	12	6
	R	0	0	1
Greedy N	**PCT**	83	84	62
	CT	29	19	49
	PC	81	81	34
	PT	83	81	17
	P	77	72	12
	C	0	1	10
	T	24	10	6
	R	0	0	0
Expected Improvement	**PCT**	46	48	17
	CT	19	6	17
	PC	45	40	16
	PT	44	52	10
	P	43	44	8
	C	0	0	7
	T	20	8	4
	R	0	0	0

S6 Persistent Homology

The aim of topological data analysis is to capture the "shape" or, more precisely, the topological features which include the connectivity and higher-dimensional holes of a collection of data points (also known as a "point cloud"). In this context, the point cloud is viewed as a 0-dimensional representation of a higher-dimensional object. An example of a point cloud is shown in figure S5. Persistent homology is a tool that is used to record the topological features of the higher-dimensional object which will depend on the resolution (or length scale) at which we consider the data. For example, at a higher resolution (or smaller scale), the data points may represent something relatively disconnected and with fewer higher-dimensional topological features than at a coarser resolution (or larger scale). The features which exist over a long range of length scales are called "persistent".

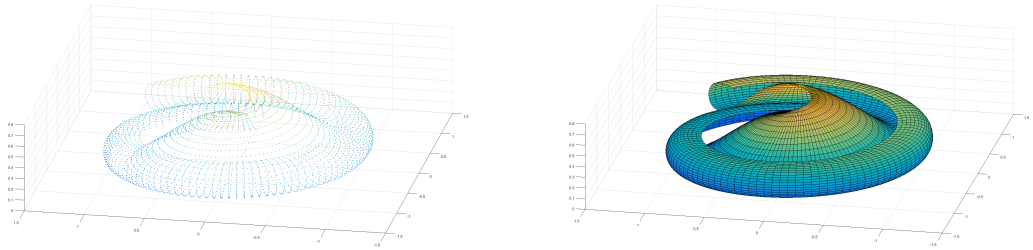

Figure S5: A point cloud (left) representing the pasta type "cappelletti" (shown as a surface plot on the right) [6].

Given a point cloud, that is, a finite collection of points in an ambient space of some dimension n, its persistent homology is computed as follows. For each fixed length scale $r > 0$, we build a higher-dimensional object called a Rips complex on the points in the point cloud by the following rule: Put n-dimensional balls of radius r centred at each data point; whenever $k+1$ such balls intersect, create the k-dimensional simplex defined by these points. So for $k = 1$ we get edges between data points, for $k = 2$ we get surfaces etc. The collection of these simplices as well as the relationships between them form the Rips complex. An example of this is given in figure S6. Note that, unlike the perhaps better known simplicial complex from algebraic topology, this is a purely abstract combinatorial construction that need not have a geometric realisation. However, as with a conventional simplicial complex, the algebraic structure of the Rips complex allows us to define what is known as the homology of the complex. This is a sequence of groups $\{H_k\}_{k=0,1,\ldots}$ that, loosely speaking, counts the number of non-contractible loops in each dimension k. H_0 counts the number of connected components and H_k is trivial for $k \geq n$. In other words, the homology of the Rips complex at length scale r is a measure of the connectivity and the number of holes in different dimensions in the data at that length scale. The persistent homology of the point cloud is the collection of all the homologies $\{H_k\}_{k=0,1,\ldots}$ across all the length scales $r > 0$.

The information given in the persistent homology can be stored in the form of a barcode where $2r$ (the diameter of the balls in the construction of the Rips complex) goes along the x-axis and each horizontal line corresponds to a connected component or a non-contractible loop in the Rips complex [7]. Figure S7 shows an example of this. Starting with r close to 0, none of the radius r balls will intersect and, therefore, the only non-trivial homology is in dimension $k = 0$ and each point in the data set will represent a connected component in H_0. But as we increase the length scale r, more and more simplices are added to the Rips complex and as a result, we get fewer and fewer connected components. At the same time, non-contractible loops in higher dimensions appear. As we increase r further, more non-contractible loops may appear and some existing loops will become contractible. In this way, the barcode represents a kind of topological fingerprint of the point cloud in which more persistent features are represented by longer horizontal lines.

However, while the barcode is a useful graphical representation of the persistent homology, it has the problem that it is not unique. For example, changing the order in which the non-contractible loops are recorded may change the barcode entirely. Instead we use a summary of the barcode

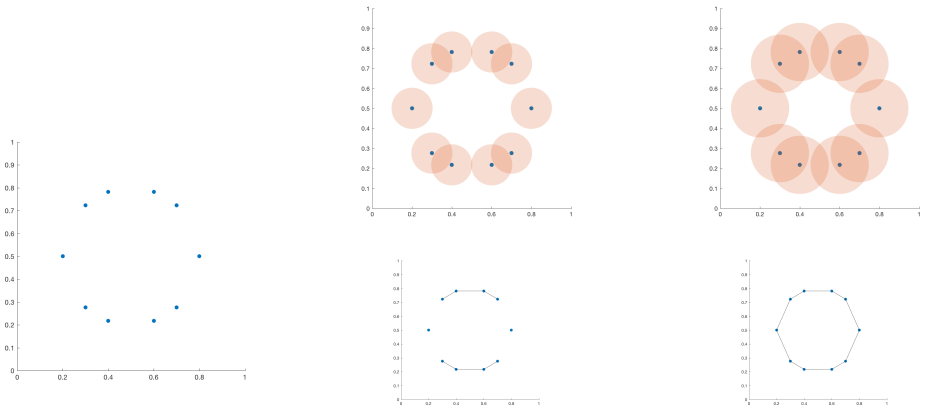

Figure S6: A point cloud in an ambient space of dimension 2 (left) and its Rips complex at two different length scales $r = r_1$ (middle) and $r = r_2$ (right). For each length scale we have shown the point cloud with balls of radius r (top) and the resulting Rips complex (bottom).

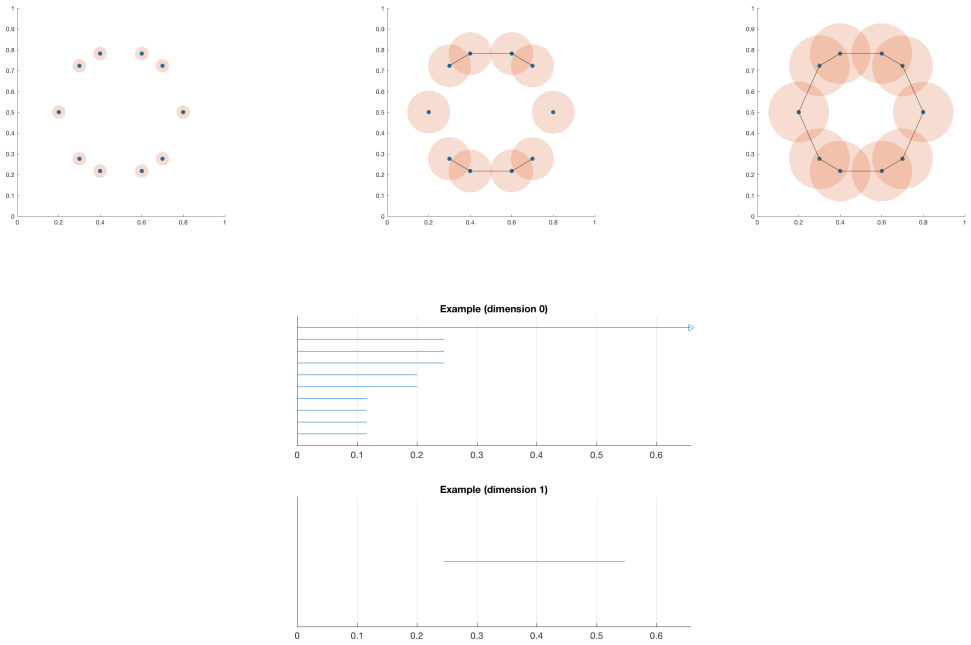

Figure S7: The Rips complex defined by a point cloud at three different length scales (top) and the barcode for this point cloud (bottom).

data called a *persistence diagram* which is invariant under reordering of loops. Each horizontal line in the barcode has a starting point and an endpoint corresponding to the birth (at $2r = b$) and death (at $2r = d$) of a connected component/non-contractible loop in the data. The persistence diagram summarises the barcode as a plot of all the birth-death pairs (b, d) (with separate plots for each dimension k of the homology). So for a given point cloud, its persistence diagram consists of n plots, namely, the birth-death pairs of $H_0, H_1, \ldots, H_{n-1}$ respectively. The further a birth-death pair is away from the diagonal $y = x$ in the persistence diagram, the more persistent is the non-contractible loop that it corresponds to [8]. An example of a persistence diagram is shown in figure S8.

The atoms in a MOF/COF form a periodic structure of repeated unit cells. The (x, y, z)-coordinates of the atoms in such a unit cell define a point cloud in 3-dimensional space and

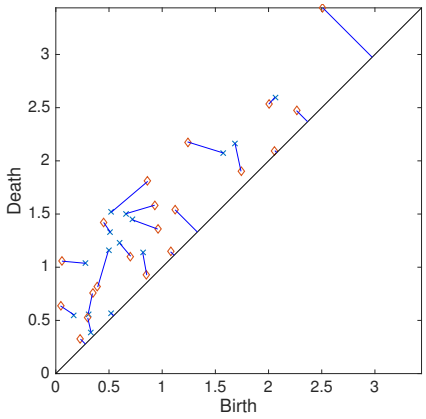

Figure S8: Illustration of a distance measure using persistence diagrams.

the length scale in this case measures the proximity of atoms to each other. However, if we simply computed the persistent homology of a unit cell, it would not accurately reflect the topology of the MOF/COF in the areas near the atoms in the boundary of the unit cell as the radius-r balls from atoms in adjacent unit cells would be ignored when constructing the Rips complex. We can get around this problem with a simple adjustment. We note that the construction of the Rips complex relies only on the distances between the data points rather than their positions. To reflect the periodic boundary conditions, we can simply define the distance between two atoms A_1 and A_2 to be the minimum distance between the atom A_1 in the chosen unit cell and all the copies of the atom A_2 in this unit cell as well as the 26 unit cells that surround it (see S9 for a two-dimensional illustration).

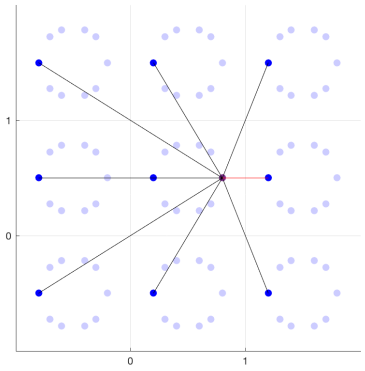

Figure S9: A two-dimensional illustration of the distance measure used between the atoms in a MOF/COF. The unit cell in the middle has 8 surrounding unit cells. The distance between the atoms A_1 (in red) and A_2 (in blue) is the minimum distance (shown in red) of the 9 possible distances between A_1 and all the copies of A_2.

There are several existing software packages that can be used to compute the persistent homology of a given point cloud, including the Java library JavaPlex [9]. For each hypothetical MOF/COF in our dataset we computed the adjusted distances described above for the atoms in a unit cell and used these as input to JavaPlex to compute the birth-death pairs for the homologies H_0 and H_1 of the persistence diagram. In practice we can only compute the homologies for finitely many values of r. In JavaPlex this is done by only considering values of r that are multiples of a pre-specified filtration value. This means that birth values b are rounded down to the nearest such value of r and death values d are rounded up. We have chosen a filtration value of 0.05 to balance the need for accuracy versus computational efficiency.

In order to include the information from the persistence diagram in the predictive model for MOF/COF performance, we convert it into a vector that we call the persistent homology feature vector. For simplicity, we have only included the homologies H_1 in the model (that is, the information from the plot in the persistence diagram corresponding to loops in dimension 1) but the same method could be applied in general. The purpose of the feature vector is to capture the essential information from the persistence diagram in a format that can be input as a predictor in the model and in a way that is consistent between different MOFs/COFs. The underlying structure of the feature vector is a histogram. We partition the area covered by the persistence diagram into horizontal strips (or "bins") and for each MOF/COF, the histogram counts the number of points in each bin. The histogram is therefore a summary of the persistence diagram which can be made more or less coarse by the choice of bin size. Each of the points (b, d) in the persistence diagram corresponds to a non-contractible loop in the MOF/COF and, loosely speaking, the death value d measures the size of the corresponding pore while its persistence $d - b$ measures how well defined the pore is. The topological features with higher relative persistence $\frac{d-b}{d}$ are likely to be the more important ones and we reflect this in the histogram by applying the relative persistence of each point as a weight. Finally, we scale the histogram according to "surface area/volume" of the corresponding MOF/COF. This scaling ensures a level of consistency between the way that different MOFs/COFs are included in the model.

In summary, the feature vector is defined as follows. Let $\{[D_i, D_{i+1}[\}_{i=1,\ldots,l}$ be a partition into l intervals of the possible death values in the persistence diagrams. This defines l horizontal strips. For a given MOF/COF with "surface area/volume" of S, let $\{(b_i, d_i)\}_{i=1,\ldots,N}$, denote the points in its persistence diagram. The elements of the feature vector $(x_1, \ldots, x_l)$ for this MOF/COF are then given by

$$x_j = \frac{1}{S} \sum_{D_j \leq d_i < D_{j+1}} \frac{d_i - b_i}{d_i}$$

where x_j is 0 if there are no death values in the interval $[D_j, D_{j+1}]$.

References

(1) GPy, *GPy: A Gaussian process framework in python*, `http://github.com/SheffieldML/GPy`, since 2012.

(2) C. E. Rasmussen and C. K. I. Williams, in *Gaussian Processes for Machine Learning*, The MIT Press, Boston, 2006, ch. Regression, pp. 7–30.

(3) F. Pedregosa, G. Varoquanx, A. Gramfort, V. Michel, B. Thirion, G. O, B. M, P. Prettenhofer, R. Weiss, V. Duborg, J. Vanderplas, A. Passos and D. Cournapeau, *Scikit-learn: Machine Learning in Python*, tech. rep., 2011, pp. 2825–2830.

(4) C. E. Wilmer, O. K. Farha, Y.-S. Bae, J. T. Hupp and R. Q. Snurr, *Energy & Environmental Science*, 2012, **5**, 9849.

(5) R. Mercado, R.-S. Fu, A. V. Yakutovich, L. Talirz, M. Haranczyk and B. Smit, *Chemistry of Materials*, 2018, **30**, 5069–5086.

(6) G. L. Legendre, *Architectural Design*, 2011, **81**, 100–101.

(7) R. Ghrist, *Bulletin of the American Mathematical Society*, 2008, **45**, 61–75.

(8) Y. Mileyko, S. Mukherjee and J. Harer, *Inverse Problems*, 2011, **27**, 124007.

(9) A. Tausz, M. Vejdemo-Johansson and H. Adams, Proceedings of ICMS 2014, ed. H. Hong and C. Yap, 2014, pp. 129–136.