mofSimplify Manual DRAFT

Gianmarco Terrones*

HJK Group, MIT

1 Overview

The mofSimplify website, available at `mofsimplify.mit.edu`, serves as an interface through which one can interact with the CoRE MOF data curated by Aditya Nandy. Specifically, a user can attain property predictions on new MOFs by using mofSimplify. These predictions are generated by artificial neural networks (ANNs; models, like linear regression but more complicated and not linear) trained on the curated MOF data. In addition to these predictions, the user can see which CoRE MOFs are most similar to the new MOF as determined by the ANNs.

The reason why ANNs are suitable for these predictions is that MOF stability is very difficult to determine through simulation.

2 Queueing up a MOF

The user has three ways to queue up a MOF for analysis on the website.

The first way is to click on the **Load example MOF** button, which loads a UiO-66 cif file. UiO-66 is loaded onto the website by default when the site is first opened.

The second way is to click on the **Upload MOF cif file** button, which will allow the user to select a file from their computer to upload. The file must be a `cif` file; the website will alert you if you accidentally upload something else. The rest of the functionality on the website has a better probability of success if the MOF you upload is "clean." In other words, the cif file should not have solvent molecules.

The third way is to click on the **Make bb-generated MOF** button, which generates a MOF out of building blocks. Before clicking the button, however, a linker, SBU (secondary building unit), and MOF net must be selected. The code behind MOF building block construction comes from `https://github.com/tobacco-mofs/tobacco_3.0`. This repository is affiliated with recent work [1] which builds off of the Topologically Based Crystal Constructor (ToBaCCo) [2]. The linkers and SBUs are pieced together according to the chosen net. Not all combinations of linker, SBU, and net can generate a MOF, and if the user selects an invalid combination and attempts generation, an alert is triggered. An example of a valid combination that you can try is 3B_4H_Ch as the linker, 6c_Al_1 as the SBU, and acs as the net. mofSimplify does not have the capability to make ToBaCCo MOFs with more than one type of linker or SBU, but you can download the ToBaCCo 3.0 code for this purpose if you wish to do so. The ToBaCCo code comes with a manual.

Whenever a new MOF is queued up in any of the three possible ways, all visualizations, status messages, and predictions are cleared. The name of the queued up MOF is displayed as the first line in the **Status messages and MOF predictions** section of mofSimplify.

3 MOF visualization

The user can visualize the entire queued MOF using the "Visualize MOF" button. The code behind the visualization comes from `https://github.com/snurr-group/web-mofid`, which is affiliated with recent work [3]. Visualization works the majority of the time for most MOFs, but if it fails (usually due to an out-of-memory error after repeated MOF visualization) an alert is presented to the user which instructs the user to refresh the page, as this tends to fix the problem. However, if a MOF has too large a unit cell, visualization on mofSimplify will not work. I would recommend trying a program like VESTA in this case.

*gterrone@mit.edu

4 MOF component identification

The user can determine the linkers and SBUs that comprise the queued MOF by clicking the **Get MOF components** button. An alert is displayed if this operation fails. The operation tends to fail, or at least take longer, if the queued MOF has a very large unit cell. Once the MOF components have been identified, the dropdowns "Linkers" and "sbus" are populated. The user can select a linker (sbu) from the appropriate dropdown menu and click on the **Visualize linker** (**Visualize sbu**) button to visualize it. The SMILES string of the visualized component will also appear next to the visualization. In addition, there is a button **Only components with unique connectivities** which eliminates duplicate components using molecular graph determinants. This button can be useful when many non-unique linkers or SBUs in a MOF are incorrectly found to be distinct by the molSimplify code. However, this button filters out isomers as well, so upon being clicked it is replaced with an **All components** button which gives the user the option to undo the filtering in the dropdowns. The **Reset zoom level** button under the component visualization pane resets the zoom level of the visualized component.

The code behind this operation is from `molSimplify.Informatics.MOF.MOF_descriptors`. molSimplify is available on GitHub at `https://github.com/hjkgrp/molSimplify` and is affiliated with recent work [4]. As you may have guessed, mofSimplify's name was inspired by molSimplify.

5 MOF property prediction

The user can predict the queued MOF's stability using the **Predict stability upon solvent removal** or the **Predict thermal stability** buttons. Both of these operations generate features by invoking code from `molSimplify.Informatics.MOF.MOF_descriptors` to get chemical information in the form of RACs and by making Zeo++ calls to attain geometric information (for more about RACs, see this paper [5]; for more on Zeo++, see this paper [6]). A prediction is then made using an ANN trained by Aditya Nandy on the data he compiled. To be clear, there is one ANN trained for solvent removal stability predictions, and there is one ANN trained for thermal stability predictions. Data was compiled using a widescale search and natural language processing; there is a Scientific Data paper about this published by now hopefully.

For solvent removal stability predictions, the model is performing a classification task but quantifies its certainty in its prediction by how close the output is to zero (unstable) or one (stable).

For the thermal stability prediction, in addition to a temperature prediction, a percentile for the current MOF's predicted thermal breakdown temperature relative to the training data experimental breakdown temperatures is displayed. Furthermore, a plot is displayed showing the current MOF's predicted breakdown temperature relative to the experimental breakdown temperatures of the CoRE MOF database MOFs used to train the utilized ANN.

If the user happens to upload a MOF that is in the training data of the solvent (thermal) ANN, and the **Predict stability upon solvent removal** (**Predict thermal stability**) button is pressed, mofSimplify will return the ground experimental truth for that MOF instead of making an ANN prediction.

Predictions (or ground truths) are color coded. For the solvent case, green corresponds to the model being confident that the MOF is stable upon solvent removal (or ground truth stability). Red corresponds to the model being confident the MOF is unstable upon solvent removal (or ground truth instability). Yellow corresponds to the model being less sure of its prediction. For the thermal case, green indicates the current MOF's predicted or ground truth thermal breakdown temperature is high relative to the training data. Red indicates the current MOF's predicted or ground truth thermal breakdown temperature is low relative to the training data. Yellow indicates an intermediate thermal breakdown temperature.

If a prediction fails at any point, an alert is presented to the user. The code may take a long time to run or even fail to complete if the queued MOF has a very large unit cell. Many of the buttons on mofSimplify generate an alert when they are clicked informing the user that the selected operation may take a few seconds. A prediction operation can take around 12 seconds for some MOFs. The code is probably running after you click a button, please be patient! Do not click on another button if you just clicked on a button and its operation has not yet completed.

5.1 ANN nearest neighbors

Recent work has evaluated ANN latent space distance as a method for quantifying uncertainty in a prediction [7]. In brief, latent space distance quantifies the similarity between two model inputs after they have passed through most of the neural network. In our case, a model input is a MOF which is translated into a bunch of numbers that describe it. Two MOFs which have a small ANN latent space distance can be expected to have a similar property, regarding the property that that ANN predicts for.

When a solvent removal stability or thermal stability prediction is made in mofSimplify, the corresponding ANN nearest neighbors drop down is populated with five MOFs. These nearest neighbors are the CoRE MOFs that have the smallest latent space to the queued MOF.

Information about a neighbor can be displayed by clicking the **Show solvent neighbor info** or **Show thermal neighbor info** button. The information displayed includes the neighbor's experimental ground truth and the DOI of the paper associated with the neighbor. By clicking the **Download solvent neighbor info** button, the user can download a file containing this information, as well as the `cif` file for the neighbor. The first file is a `.out` file and can be opened using a text editor.

The **Visualize solvent neighbor** and **Visualize thermal neighbor** buttons are analogs of the **Visualize MOF** button. The difference is that they do not visualize the queued MOF, but rather the neighbor currently selected in the solvent or thermal neighbor drop down. The last line of **Status messages and MOF predictions** indicates which the name of the MOF that is currently visualized.

Likewise, the **Get solvent neighbor components** and **Get thermal neighbor components** buttons are analogs of the **Get MOF components** button. There is a line above the Linkers drop down which indicates the name of the MOF for which components are listed.

If a ground truth is returned instead of a prediction (see MOF property prediction), no ANN nearest neighbors are gathered.

Clicking on the **Show thermal neighbor info** button displays, in addition to the things mentioned previously, a simplified TGA plot. The dot on the plot indicates the thermal breakdown temperature of the MOF. The simplified TGA plot comes from a TGA plot in the paper on the neighbor. For copyright reasons, the original TGA plots could not be displayed. Simplified plots were generated by choosing four points total, two on each segment of different slope on the TGA plot. A rapid decrease in mass indicates breakdown.

References

[1] R. Anderson and D. A. Gómez-Gualdrón, "Increasing topological diversity during computational "synthesis" of porous crystals: how and why," *CrystEngComm*, vol. 21, no. 10, pp. 1653–1665, 2019.

[2] Y. J. Colón, D. A. Gómez-Gualdrón, and R. Q. Snurr, "Topologically guided, automated construction of metal–organic frameworks and their evaluation for energy-related applications," *Crystal Growth & Design*, vol. 17, no. 11, pp. 5801–5810, 2017.

[3] B. J. Bucior, A. S. Rosen, M. Haranczyk, Z. Yao, M. E. Ziebel, O. K. Farha, J. T. Hupp, J. I. Siepmann, A. Aspuru-Guzik, and R. Q. Snurr, "Identification schemes for metal–organic frameworks to enable rapid search and cheminformatics analysis," *Crystal Growth & Design*, vol. 19, no. 11, pp. 6682–6697, 2019.

[4] E. I. Ioannidis, T. Z. Gani, and H. J. Kulik, "molsimplify: A toolkit for automating discovery in inorganic chemistry," 2016.

[5] J. P. Janet and H. J. Kulik, "Resolving transition metal chemical space: Feature selection for machine learning and structure–property relationships," *The Journal of Physical Chemistry A*, vol. 121, no. 46, pp. 8939–8954, 2017.

[6] T. F. Willems, C. H. Rycroft, M. Kazi, J. C. Meza, and M. Haranczyk, "Algorithms and tools for high-throughput geometry-based analysis of crystalline porous materials," *Microporous and Mesoporous Materials*, vol. 149, no. 1, pp. 134–141, 2012.

[7] J. P. Janet, C. Duan, T. Yang, A. Nandy, and H. J. Kulik, "A quantitative uncertainty metric controls error in neural network-driven chemical discovery," *Chemical science*, vol. 10, no. 34, pp. 7913–7922, 2019.