Contents

1 Zeolite Simulation Environment

Jerry Crum
Updated: 10/5/2020

2 Description

This is a package for automating structure generation and characterization of zeolites for computational chemistry.

3 Requirements

This package is built around the ASE atoms object, and thus requires installation of ASE, available at: https://wiki.fysik.dtu.dk/ase/install.html.

Ring finding uses graph theory implemented by NetworkX, available at: https://networkx.github.io.

Numpy is also required.

4 Installation

Clone this git repository into your PYTHONPATH (i.e. anaconda3/lib/python3.X/site-packages/), and you are good to go. Also, you can clone this git repository anywhere you want, and then add the location to your PYTHONPATH.

5 Current Modules

5.1 zse.cation

5.1.1 divalent(atoms, M, path= None)

This function will place one divalent cation into a zeolite framework that contains two negative charge centers. The cation is placed at each of 6 rings around each of the aluminum atoms. This creates 12 structures. Depending on the placement of your Al, some of the strucutres may not be unique.

The structures will be placed in the path provided as an input. Format of the structures folder names are:

D-aluminum_index-oxygen1_index,oxygen2_index.

The original version of this code was written by Sichi Li in 2017.

atoms: zeolite framework in the form of an ASE atoms object, with two Al T-sites.

M: Symbol of the divalent cation you would like added to the structure. Must be a string, i.e. 'Cu'.

path: If path is provided, structures will be saved to path location.

returns: Trajectory of atom objects for the structures created.

5.1.2 monovalent(atoms, symbol, path=None)

This function will place one monovalent cation into a zeolite framework that contains one aluminum atoms. The cation is placed at each of the 6 rings around the aluminum atoms. This creates 6 structures.

The structures will be placed in the path provided as an input. Format of the structures folder names are: D-aluminum_index-oxygen1_index,oxygen2_index.

atoms: zeolite framework in the form of an ASE atoms object, with one Al T-site.

symbol: Symbol of the monovalent cation you would like added to the structure. Must be a string, i.e. 'Na'.

path: If path is provided, structures will be saved to path location.

returns: Trajectory of atom objects for the structures created.

5.2 zse.collections

5.2.1 framework('code')

This command will load an atoms object of the zeolite code you input. Every zeolite structure listed on iza-strcutres.org/databases/ as of 3/7/2020 are contained in this package.

code: string of the 3 character IZA framework code (i.e. 'CHA' or 'MFI').

returns: Atoms object of that zeolite framework with the lattice constants and atom positions from the IZA.

5.2.2 get_all_fws()

returns: List of all the zeolite frameworks included in ZSE.

5.3 zse.protonate

5.3.1 isolated(atoms,tatom,path='.')

This command will enumerate all the possible structures of where a proton can bind around an isolated hetero atom inside the zeolite framework. Returns a trajectory of the possible structures and a list of all the oxygen indices used (see examples for usage).

atoms: zeolite framework in the form of an ASE atoms object.

tatom: index of the hetero atom you would like to protonate.

path: path to the location where you would like the structures saved. Currently, the structures can only be saved in the VASP POSCAR format.

returns: traj, oxygens

traj: A trajectory of atom objects for the structures created.

oxygens: List of the oxygen indices that the proton was bound to.

5.3.2 paired(atoms,tatoms,path='.')

This command will enumerate all the possible structures of where a proton can bind around a set of paired hetero atoms inside the zeolite framework. Returns a trajectory of the possible structures, and a list of all the oxygen indices used (see examples for usage).

atoms: zeolite framework in the form of an ASE atoms object.

tatom: if both hetero atoms are of the same species, use the symbol (i.e. 'Al'). Otherwise use a list of the indices for the hetero atoms.

path: path to the location where you would like the structures saved. Currently, the structures can only be saved in the VASP POSCAR format.

returns: traj, oxygens

traj: A trajectory of atom objects for the structures created.

oxygens: List of the oxygen indices that the proton was bound to.

5.4 zse.rings

5.4.1 get_fwrings(code)

code: Zeolite framework code in string form, i.e. 'CHA'.

returns: Array containing the size of rings present in that zeolite framework.

5.4.2 get_orings(atoms, index, possible)

This function will find all the rings associated with a specific oxygen atom in the zeolite framework.

atoms: Atoms object containing the zeolite framework. Works best if there are no extra framework atoms present.

index: Index of the oxygen atom in question.

possible: The possible sizes of rings for the framework type in question. Use get_fwrings() for this information.

returns: Class, paths, atoms

Class: List of the size of rings found to be associated with the oxygen.

paths: Indices of the atoms that make up each of the rings.

atoms: Atoms object that shows only the atoms in the rings associated with the oxygen.

5.4.3 get_rings(atoms, index)

Warning: This an old implementation of get_orings that does not work consistently for every framework type.

This command will list the size of the rings associated with an oxygen atom. For CHA and other zeolite frameworks with only one unique T Site, the oxygens are only associated with three rings. This may be not be true for more complicated zeolites, and further testing is required.

atoms: zeolite framework in the form of an ASE atoms object.

index: index of the oxygen you want to get the rings about, integer.

Note: With latest update, speeds are much improved for ring classification. Also, there is no longer a need to provide the possible rings to the function. Please see example workbook for usage.

5.4.4 get_trings(atoms, index, possible)

This function will find all the rings associated with a specific T-site in the zeolite framework. This function works by using get_orings() for each oxygen connected to the T-site, and then removing duplicate rings.

atoms: Atoms object containing the zeolite framework. Works best if there are no extra framework atoms present.

index: Index of the oxygen atom in question.

possible: The possible sizes of rings for the framework type in question. Use get_fwrings() for this information.

returns: Class, paths, atoms

Class: List of the size of rings found to be associated with the oxygen.

paths: Indices of the atoms that make up each of the rings.

atoms: Atoms object that shows only the atoms in the rings associated with the oxygen.

5.4.5 find_orings(G, index, possible)

This is a helper function for get_trings(), it is not really meant to be used on it's own.

5.4.6 remove_dups(rings)

This is a helper function that removes duplicate rings found during use of get_orings(), get_trings(), and unique_rings(). It is not really intended to be used on it's own.

5.4.7 remove_sec(rings)

This is a helper function that removes secondary rings (random paths through the framework) found during use of get_orings(), get_trings(), and unique_-rings(). It is not really intended to be used on it's own.

5.4.8 tring_driver(code)

This is a helper function for get_trings(), and unique_rings(). It is not intended to be used on it's own.

5.4.9 unique_rings(code)

This function will find all of the unique rings in a framework of sizes contained in the possible rings from get_fwrings().

code: Zeolite framework code in string form, i.e. 'CHA'.

returns: unique_tsites, unique_full, trajectories

unique_tsites: Dictionary containing all the unique rings of each size present in the framework, listed by the T-sites in the ring. If you have used this code on CHA, and you would like to see the list of 8-MR T-sites, type: // unique_tistes[8].

unique_full: Dictionary containing all the unique rings of each size present in the framework, listed by all the atom indices that make up each ring. If you have used this code on CHA, and you would like to the list of 8-MR indices, type: // unique_full[8].

trajectories: Dictionary containing all the atoms objects for each ring found. If you have this used this code on CHA, and you want to visualize the 8-MR, type: */ from ase.visualize import view /* view trajectories[8].

5.5 zse.substitute

5.5.1 tsub(atoms,index,new_atom)

This command will replace one atom in the zeolite framework with a new of your choosing.

atoms: zeolite framework in the form of an ASE atoms object.

index: index of the atom you would like to change into a new element.

new_atom: symbol of the element you would like to change index into, must be string.

returns: Atoms object of the structure with the substitution.

6 Future Inclusions

T Site Pair Identification

7 Contributions

If there is something you would like to see added to this package, or if you would like to contribute, please email me at jcrum@nd.edu.

8 Acknowledgments

8.1 Contributors

Jerry Crum
Justin Crum
Sichi Li
Yujia Wang
William Schneider

8.2 Testers

Craig Waitt
Jian Ren Lim
Elsa Koninckz