Lithium Interstitial Code

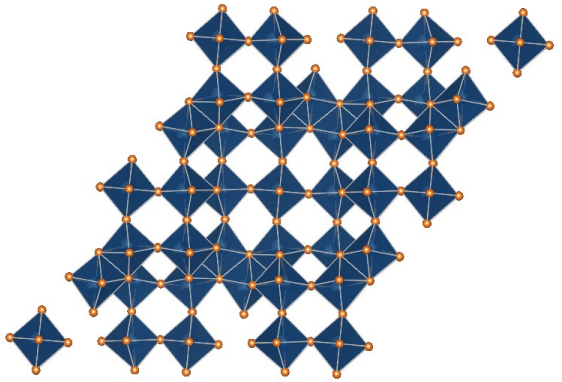
Muna Saber and Farnaz Kaboudvand

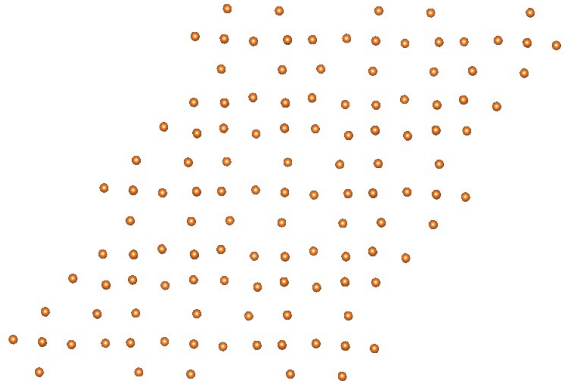
Overview

Start with indexing Chalcogens in Chalcogens sublattice (use Orbittree.hh/.cc)

1. Keep clusters within a certain cutoff number (1A to 7 Angstroms)
2. Now that you have all these clusters, filter out clusters with less than three anions or more than six (Can make it input condition)
3. Find center of position
4. If other atom is within 1A of center position, remove cluster
5. Now that you have remaining clusters, if there are competing clusters, choose the cluster with the highest Chalcogens coordination
6. Potentially can generate all combinations with other oxygen coordinations chosen as preferred (but may be too many)

TM_xO_y





PseudoCode Part 1

Read in Poscar

Make an ordered map of Chalcogens with coordinates and index

For (first index to the last one) :

Set cut off for orbit distance

(Consider Boundarys)

For (clusters within specified orbit):

Find center of coordinates

For (clusters where center of coordinates does not overlap atoms):

Check if there are competing clusters for same indicies and choose the highest coordinated cluster (ie if cluster 1 is within cluster 2, choose cluster 2)

Put intercalant at the remaining center of coordinates

Specifications

– tetrahedral (only consider clusters of size 4 at last four loop)

- octahedral (only consider clusters of size 6 at last four loop)

Potential functions

Read_poscar(poscar)

get_species(poscar)

Ordered_map my_chalcogens(indicies, coords)

Double set_cutoff(my_chalcogens)

Vector<Vector3f> find_center_coords(my_chalcogens)

ordered_map find_coordination_of_orbits(my_chalcogens, center_coords)

Vector3f return_larger_coordination_number(my_chalcogens, orbit_coords)

Vector3f place_intercalant(center_coords, species)

print_new_poscar(intercalant_positions, poscar)