Scripts for XR postprocessing with USPEX

Michele Galasso

December 14, 2019

1 split_CIFs.py

It splits the structures with lowest enthalpy of a variable composition USPEX run into multiple CIF files. The script takes the following arguments from command line:

1. the output file `extended_convex_hull` from USPEX
2. the output file `extended_convex_hull_POSCARS` from USPEX
3. the value of the external pressure used for the USPEX run

The script performs the following operations:

1. for each structure, it reads the parameters *enthalpy* and *fitness* from `extended_convex_hull` and the geometry from `extended_convex_hull_POSCARS`
2. it selects, for each reduced formula, the 5 structures with lowest enthalpy
3. it outputs the selected structures as CIF files in a new folder *results*

The name of each CIF file has the format `i_ID_fitness_enthalpy_iupacformula_pressure_symmetry.cif`, where:

i is a natural number which orders the output with increasing *fitness*

ID is the structure ID from the USPEX run

fitness is the *fitness* of the structure

enthalpy is the *enthalpy* of the structure

iupacformula is the *IUPAC formula* of the structure

pressure is the pressure used for the USPEX run

symmetry is the space group number, determined with tolerance 0.2

Example: `python split_CIFs.py extended_convex_hull extended_convex_hull_POSCARS 50GPa`

2 sublattice_split_CIFs.py

It reads the results of a variable composition USPEX run, it removes all hydrogen atoms, and then it splits the structures with lowest enthalpy into multiple CIF files. The script takes the following arguments from command line:

1. the output file `extended_convex_hull` from USPEX
2. the output file `extended_convex_hull_POSCARS` from USPEX
3. the value of the external pressure used for the USPEX run

The script performs the following operations:

1. for each structure, it reads the parameters *enthalpy* and *fitness* from `extended_convex_hull` and the geometry from `extended_convex_hull_POSCARS`
2. it deletes all hydrogen atoms
3. it selects, for each reduced formula, the 5 structures with lowest enthalpy
4. it outputs the selected structures as CIF files in a new folder *results*

The name of each CIF file has the format `i_ID_fitness_enthalpy_iupacformula_pressure_symmetry.cif`, where:

i is a natural number which orders the output with increasing *fitness*

ID is the structure ID from the USPEX run

fitness is the *fitness* of the structure

enthalpy is the *enthalpy* of the structure

iupacformula is the *IUPAC formula* of the structure, with hydrogens

pressure is the pressure used for the USPEX run

symmetry is the space group number, determined with a tolerance of 0.2 and without hydrogens

Example: `python sublattice_split_CIFs.py extended_convex_hull extended_convex_hull_POSCARS 50GPa`

3 fix_comp.py

The script performs postprocessing for USPEX calculations with *fixed composition.* It takes the following arguments from command line:

1. the output file `Individuals` from USPEX
2. the output file `gatheredPOSCARS` from USPEX

3. the value of the external pressure used for the USPEX run

The script performs the following operations:

1. for each structure, it reads the parameter *enthalpy* from `Individuals` and the geometry from `gatheredPOSCARS`

2. it computes `real_fitness` = `enthalpy` / `total_number_of_atoms`

3. it outputs the structures as CIF files in a new folder *results*

The name of each CIF file has the format `i_realfitness_iupacformula_symmetry_ID_pressure.cif`, where:

i is a natural number which orders the output with increasing `real_fitness`

realfitness is the `real_fitness` of the structure

iupacformula is the *IUPAC formula* of the structure

symmetry is the space group number, determined with a tolerance of 0.2

ID is the *structure ID* read from the input files

pressure is the pressure used for the USPEX run

Example: `python3 fix_comp.py Individuals gatheredPOSCARS 50GPa`

4 xr_screening.py

The script performs a screening of USPEX results, looking for the structures that best match an experimental X-ray spectrum. It contains a number of input parameters, like the name of the input files, the theoretical and experimental pressure, and so on. These parameters are all explained by comments, and some of them need to be tuned for your specific problem. The script performs the following operations:

1. for each structure, it reads the geometry from `gatheredPOSCARS`

2. it computes the theoretical X-ray spectrum

3. it computes a fitness, describing how much the theoretical spectrum agrees with the experimental one

4. it outputs a graph with the theoretical and experimental spectra, where the file name starts with the value of the computed fitness

5 exclusion.py

The script takes the following arguments from command line:

1. the wavelength of the incident radiation in Å
2. the cut-off in % of the maximum intensity
3. a number of exclusion regions in degrees, expressed as two angles separated by a hyphen (-)

The script works in a folder with many CIF files, and performs the following:

1. it opens, one by one, all CIF files and it predicts the XRD pattern of the structure according to the given wavelength
2. if the predicted pattern contains any peak in the exclusion regions that is bigger than the given cut-off, it deletes the CIF file

Example: `python3 exclusion.py 0.6199 25 25-28 31-32`