Scripts for XR postprocessing with USPEX

Michele Galasso

December 16, 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`

1

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 fixcomp_split_CIFs.py

It splits all the structures of a fixed composition USPEX run into multiple CIF files. The script 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_ID_fitness_enthalpy_iupacformula_pressure_symmetry.cif`, where:

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

ID is the structure ID from the USPEX run

fitness is the `real_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 a tolerance of 0.2

Example: `python fixcomp_split_CIFs.py Individuals gatheredPOSCARS 50GPa`

4   xr_screening.py

It performs a screening of USPEX results, looking for the structures that best match an experimental X-ray spectrum. For the theory behind this script, see appendix A. In a few words, given an input experimental spectrum, the theoretical spectrum is calculated for each structure in the USPEX run and a value F is computed. The smaller is F, the better is the agreement between theoretical and experimental spectra. The script contains the following input parameters:

1. the importance coefficients $f(h_i)$ used in the computation of F

2. the experimental pressure

3. the pressure of the USPEX run

4. the start and end angles for the computation of theoretical spectra

5. the experimental wavelength

6. the value of σ for the gaussian smearing of peaks, used for generating output pictures

7. the name of the *spectrum file*, containing angles and intensities of the experimental spectrum

8. the name of the file `extended_convex_hull` from USPEX

9. the name of the file `extended_convex_hull_POSCARS` from USPEX

10. the parameter *match_tol*, that is the tolerance for matching experimental peaks with theoretical peaks, in degrees

The script performs the following operations:

1. for each structure, it reads the parameter *fitness* from `extended_convex_hull` and the geometry from `extended_convex_hull_POSCARS`

2. it computes the theoretical X-ray spectrum

3. it computes the agreement F between the spectra

4. it outputs a CIF file with the symmetrized structure (tolerance 0.2)

5. it outputs a PNG graph with the theoretical and experimental spectra superimposed for comparison

The name of each CIF and PNG file have the format `F_ID_fitness_iupacformula_pressure_symmetry`, where:

F is the agreement F between theoretical and experimental spectra

ID is the structure ID from the USPEX run

fitness is the `fitness` of the structure from the USPEX run

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 a tolerance of 0.2

Example: `python xr_screening.py`

5 exclusion.py

It allows to quickly filter a multitude of CIF files. 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`

Appendices

A   Spectra comparison

We developed a code which computes, from the experimental spectrum and the USPEX output, the degree of agreement (fitness) of each relaxed structure with the experimental data. The USPEX calculation and the experimental spectrum do not need to be exactly at the same pressure, but the two pressures need to be *close*, that is, no more than $20\,\mathrm{GPa}$ apart.

Since we have a pressure difference, we first translate each calculated structure to the experimental pressure by using the Birch-Murnaghan equation

$$\Delta P = \frac{3B_0}{2}\left[\left(\frac{V_0}{V}\right)^{\frac{7}{3}} - \left(\frac{V_0}{V}\right)^{\frac{5}{3}}\right]\left\{1 + \frac{3}{4}\left(B_0' - 4\right)\left[\left(\frac{V_0}{V}\right)^{\frac{2}{3}} - 1\right]\right\}$$

where ΔP is the pressure difference, V_0 is the volume of the unit cell at the calculated pressure, V is the volume at the experimental pressure, B_0 is the bulk modulus and B_0' is the derivative of the bulk modulus with respect to pressure. For B_0 and B_0' we take the average values $B_0 = 300$ and $B_0' = 3$.

Assuming that, for small pressure variations, also V_0/V will be small, we approximate the Birch-Murnaghan equation to a second order Taylor expansion in V_0/V and we get the volume V at the experimental pressure:

$$V = \frac{300}{150 + \sqrt{22500 + 300\Delta P}}V_0$$

Then we define the following scaling factor

$$k = \sqrt[3]{\frac{300}{150 + \sqrt{22500 + 300\Delta P}}}$$

that will be used to rescale the lattice parameters of all the calculated structures.

After this rescaling, the relaxed structures are symmetrized with a tolerance of 0.2 and the theoretical XRD spectra are computed. Both the theoretical and the experimental spectra are in the form of a series of peaks. For each peak we know the diffraction angle and the relative intensity (while the intensity of the highest peak in each spectrum has been conventionally given the value 100). We define a *match* between a theoretical and an experimental peak if the two peaks are less than *match_tol* degrees apart, regardless of their intensities.

The fitness between a calculated XRD spectrum and the experimental spectrum is defined by the following fitness function

$$F = \sum_{i,j}^{match} \frac{(x_i^{exp} - x_j^{th})^2}{\Delta\alpha^2} f(h_i^{exp}) + \sum_{i}^{rest} \frac{(x_i^{exp})^2}{\Delta\alpha^2} f(h_i^{exp}) + \sum_{i}^{rest} \frac{(x_i^{th})^2}{\Delta\alpha^2} f(h_i^{th})$$

$$\sum_{i,j}^{match} \frac{(h_i^{exp} - h_j^{th})^2}{100^2} f(h_i^{exp}) + \sum_{i}^{rest} \frac{(h_i^{exp})^2}{100^2} f(h_i^{exp}) + \sum_{i}^{rest} \frac{(h_i^{th})^2}{100^2} f(h_i^{th})$$

where the x_i are diffraction angles, the h_i are intensities, $\Delta\alpha$ is the total width of the spectra and $f(h_i)$ is an importance coefficient given by the following piecewise-defined function

$$f(t) = \begin{cases} 5 & \text{if } t > 90 \\ 1 & \text{if } 50 < t \leq 90 \\ 0.25 & \text{if } 10 < t \leq 50 \\ 0.02 & \text{if } 1 < t \leq 10 \\ 0 & \text{if } t \leq 1 \end{cases}$$

The first term of the fitness is a sum over the matched peaks, and each addend of this sum will be smaller the more the two peaks are close to each other. At the denominator we find the total width $\Delta\alpha$ of the spectra, which is the maximum value for an angle and it allows us to get dimensionless addends. The importance coefficient just tells us that higher peaks will give a more significant contribution to the fitness than smaller peaks, and that peaks with intensity less than 1 will give no contribution. The second term is a sum over the experimental peaks that are left after our matching, we call these peaks *experimental rest*. It has a similar fashion to the first term, but its addends will have a much higher value since there is no subtraction at the numerator. The third term, analogously, is a sum over the *theoretical rest*. The remaining three terms are very similar to the first three, but angles are substituted by intensities.

It is clear that a low value of F gives a good agreement between calculated and experimental spectra, allowing a quick identification of promising candidates in the USPEX output.