**SAPA Tutorial**

**Requirements for SAPA**

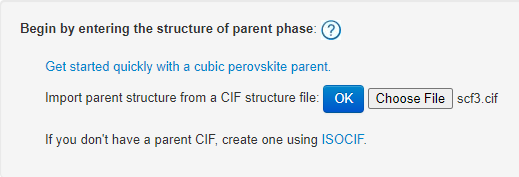
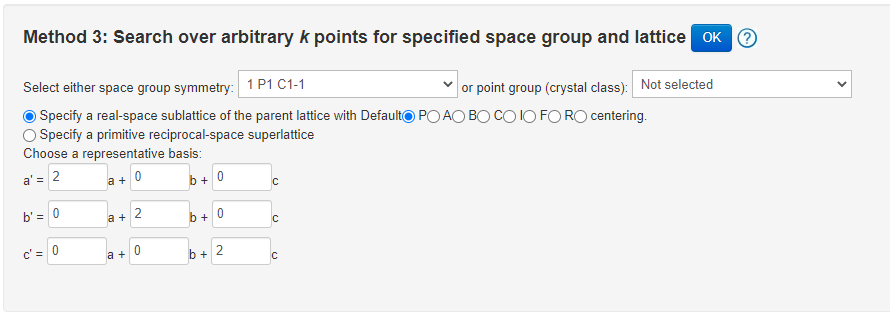
* Python version > 3.6
* HDF5 – can be obtained [here](https://www.hdfgroup.org/solutions/hdf5/) or by installing [DAWN](https://dawnsci.org/)
* TOPAS Academic version 6 or 7

**Generating ISODISTORT CIF**

* Choose input parent structure to be used
* Obtain cif of said parent structure

There are two ways to obtain the ISODISTORT cif:

**Method 1 – Online Method**

* Navigate browser to <https://stokes.byu.edu/iso/isodistort.php>
* Upload cif file here
* Scroll down to method 3
* Select space group symmetry P1
* ****Input supercell transformation
* Click OK
* Click OK again on the new window
* Choose “CIF file” option and click OK

**Method 2 – Python Method**

* Navigate to <https://github.com/FrandsenGroup/isopydistort>
* Follow installation instructions – requires “requests” library
* Use method 3 to obtain the desired supercell down to P1
* This will involve using var\_dict and setting the elements of the transformation matrix, e.g. to use a 2x2x2 supercell, var\_dict = {“basis11”: “2”, “basis22”: “2”, “basis33”: “3”}, since the other elements remain at their default. Default is the identity matrix
* Set format=”structurefile”

**MODIFYING PATH**

SAPA isn’t a python module, so it needs to know where to look for the scripts when importing. By default, it looks in the working directory, so you could copy sapa.py into the working directory for each SAPA run. However, this can be cumbersome if you intend to use SAPA regularly. There are two ways to change the places python looks for files – the first is a permanent solution, but likely requires an admin account. The second will have to be run each time.

**Setting PYTHONPATH variable**

* Save the SAPA scripts (sapa.py and sapa\_utils\_hdf.py) in a new folder on your machine. If you have other scripts you regularly import, you can also place them in this folder. For example, I have my scripts saved in C:\lib\_sapa
* Using the windows search bar, search for “environment variables”, which should give you an option “edit the system environment variables” – click this, which should give you a “system properties” window
* Click “Environment Variables” in the bottom right
* A screenshot of a computer program

  Description automatically generatedIn the Environment Variables window, make a new user variable (the uppermost “New” button)
* For variable name, enter PYTHONPATH
* For variable value, enter the absolute path of the folder you placed the SAPA scripts in (e.g. C:\lib\_sapa) – you can use the “Browse Directory” button to do this
* Click OK on the Environment Variables window, then Apply in the System Properties Window

**Using sys module**

If for whatever reason you are unable to set the PYTHONPATH variable, you can use the sys python module to append to the list of places python looks for scripts. This needs to be done each time you start a new session (or at the start of a script, before importing the dependent scripts). As with the previous method, you need to save the SAPA scripts in a new folder (preferably one you can easily remember). Then, before you import sapa in your interactive python environment, or script, do the following:

* Import sys
* sys.path.append(“path to sapa.py”)

The path needs to either be a raw string (r preceding the opening quotation mark of the string), or needs to escape the \ characters indicating a subfolder, e.g.

* r”C:\\Users\username\scripts\sapa”
* or “C:\\Users\\username\\scripts\\sapa”

**Running SAPA**

The SAPA object is created using the ISODISTORT CIF as an input. It contains all the information within the CIF as class variables, and also has functions to perform SAPA analysis. The basic use and functionality of this is demonstrated in the jupyter notebook tutorials

SAPA can be ran in two different ways. The typical way refines all mode amplitudes for a single irrep at a time, from randomised starting values. The second way, the “single mode” way, refines each mode on its own. These require slightly different input files, so use different functions to generate the input files. Input file generation is demonstrated in the following notebooks:

* Tutorial1\_BaTiO3.ipynb
* Singlemode\_ScF3.ipynb
* Groupmode\_ScF3.ipynb

**Supplying Fixed Variables to SAPA**

Variables that you want to fix per dataset, such as lattice parameters or occupancies, can be supplied via the execute command. This is detailed in the “Passing Fixed Parameters” notebook in the “Additional Examples” folder.

**Skipping Irreps**

Particular irreps can be skipped e.g. if you want to refine the modes from that irrep alongside those from each other irrep. This is detailed in the Skipping Irreps notebook in the “Additional Examples” folder