This is a list of instructions for optimizing search-generated crystal structures with DFTB3-D3. Details and background about the methods can be found in:

**Iuzzolino, L**.; McCabe, P.; Price, S. L.; Brandenburg, J. G., Crystal structure prediction of flexible pharmaceutical-like molecules: density functional tight-binding as an intermediate optimisation method and for free energy estimation. *Faraday Discussions* 2018, *Advance Article.*

Copy all the files from /home/cposs/DFTB-opt to your working directory.

Part 1: Optimise the search-generated crystal structures

1. Have all the crystal structures two directories down, like unique\_pool/str-1. This is quite normal in a search. The input structures should be called act.res
2. Make a list of structures you want to optimize called DFTB\_list
3. If you want to change the convergence criteria, change the lines after TOLDEG, TOLDEE and TOLDEX in cif2fort34.sh. Check the Crystal manual for their meaning. I suggest they are not changed though.
4. Run the script setup\_DFTB. This will setup the jobs.
5. Run the runscript in each directory; the runscript is called job\_rundftb

qsub job\_rundftb

I have written DFTB\_manag for this purpose, which requires a num\_cor like all job managers we use.

1. To rank the crystal structures in terms of energy, run the script Rank\_DFTB. You need to specify how many atoms the molecule has. It will output a file called DFTB\_ranking, in which the first column is the structure identifier, the second the DFTB3 absolute energy, the third the relative energy in kJ/mol compared to the global minimum and the fourth one the density in g/cm3.
2. The output files are not really readable. To get a .res file for further calculations (*e.g.* DMACRYS jobs) you need to run the script Get\_res\_files. This will put in each directory a starting.res containing the DFTB3-D3 optimised crystal structures with the symmetry identified by Platon (it can be different from the one in the search-generated crystal structures)

Part 2: Perform phonon calculations

This is quite a slow calculation, and so I don’t advise performing it on many structures. As such, I will explain how to perform an individual calculation, but this could be easily scripted

1. Move the input structure to a .cif format. You can use the script convert\_format.py. The structure name must be of only one world (modify if needed, otherwise the script will fail).

E.g. python convert\_format.py XXXX.res XXXX.cif

1. Apply the cif2fort34\_phonons.sh script

./cif2fort34\_phonons.sh XXXX.cif

I suggest you don’t change the convergence criteria. The script as it is calculates the phonon modes only at the Γ point of a 1x1x1 cell. If you want to increase the cell size you need to modify the line in cif2fort34\_phonons.sh:

sed -i "s/EXTERNAL/EXTERNAL\nSCELPHONO\n1 0 0\n0 1 0\n0 0 1\nFREQCALC\nNUMDERIV\n2\nDISPERSION/" INPUT

Changing the various 1s as appropriate. I don’t really know how to decide this, but it requires some expertise.

1. This will create a runscript called job\_rundftb\_phon, which you can run with qsub
2. Once the job has run, you’ll get two output files: crystal\_phon1.out, which contains the results of the tight optimisation (not important) and crystal\_phon.out, which contains the results of the phonon calculation. At the bottom of crystal\_phon.out you can find the various contributions to free energy: EL is lattice energy, E0 the zero point energy, ET the thermal contribution to internal energy, TS the entropic contribution to free energy, and PV the pressure-volume term (this should be negligible)

Note that they are absolute values, so they should be normalised to find free energy differences.