This is a list of instructions for applying the search workflow based on combining CSD information on conformational preferences with *ab initio* calculation. A background and a detailed explanation of the method can be found in:

**Iuzzolino, L**.; Reilly, A. M.; McCabe, P.; Price, S. L., Use of Crystal Structure Informatics for Defining the Conformational Space Needed for Predicting Crystal Structures of Pharmaceutical Molecules. Journal of Chemical Theory and Computation 2017, 13 (10), 5163-5171.

Before starting, make sure you have these paths in your .bashrc file:

export CSDHOME=/opt/CCDC/CSD\_2018

export CCDC\_MOGUL\_DATA=/opt/CCDC/CSD\_2018/data/

export LD\_LIBRARY\_PATH=/usr/local/lib/python2.7/dist-packages/ccdc/\_lib:$LD\_LIBRARY\_PATH

export PYTHONPATH='/home/luca/.local/lib/python2.7/site-packages':$PYTHONPATH

The CSDHOME and CCDC\_MOGUL\_DATA paths should be changed for every release.

Before running the scripts, run these commands:

module add miniconda2

source activate {user\_name}-env (*e.g.* source activate luca-env)

I suggest logging out and in again after doing these analyses, as staying in this environment can create problems with other python scripts.

Part 1: Deciding how to treat each torsion angle

1. Get the input molecule in a directory, possibly in a .mol2 format
2. Apply the script Analyse\_molecule. This script should be run as:

./Analyse\_molecule XXXX.mol2

Make sure than when you run this script Xming is on.

1. The main output files that you should consider are:
	1. The histograms and the probability density functions (PDFs), which can be found in the directory called Plots. They are named according the torsion angle whose distribution is plotted. The red lines are the PDFs, plotted in terms of the function f(θ)
	2. The text file Ranges.txt. The initial section describes in what ranges the main torsion angles exist with a PDF value above 0.1 (this is related to their likelihood). The sections headed by ‘Maxima:’ is more Important. Each value after the equal sign corresponds to a peak in the torsion angle distribution. The value among brackets describe the maximum in the peak, while the values after the ‘+-‘ describe the half width at half maxima.

 *E.g.* 84(0.65)+-21 indicates that for a certain torsion angle there

is a peak in the distribution at 84°, with a PDF f(θ) value of 0.65 and a half width at half maximum of 21°.

* 1. The text file called Shape\_matches.txt. For each torsion angle the shape effect of modifying its value in 30° steps is indicated. The first column describes the value of the torsion angle, the second column the percentage shape match with the previous torsion angle value and the third the percentage shape match with the initial conformation.

 *E.g.* 97.31.mol 97.32 93.26 means that a given torsion angle, when taking a value of 97.31°, has a 97% shape match with the previous step (in which the same torsion angle was 67.31°) and of 93% with the initial conformation (whatever the initial value in the input molecule was).

1. Use the decision tree in Figure 3 in the paper to decide how to treat each torsion angle. The information required is the PDF f(θ) peaks, their values, the half width and half maxima and the shape matches.

 Part 2: Selecting conformational regions with representative values for the constrained torsion angles

Since it is now possible to actually treat a given torsion angle as rigid in the CSD Conformer Generator (CG), I will write down two possible approaches. Note that I have never personally tested the second one, but I am confident it should give comparable answers. For very flexible molecules (*e.g.* ritonavir) I would actually strongly suggest to use the 2nd one, as the CG struggles to cover the whole search space.

 Approach 1 (based on paper):

1. Apply the CSD Conformer Generator on the molecule. Use the script Do\_CG.py:

 python Do\_CG.py XXXX.mol2

This will generate a comprehensive set of conformations, located in a file called conformers.mol2

1. Use the decision tree in Figure 4 in the paper to decide the separation thresholds to pick the representative value for each constrained torsion angle. Once again the information you need is is the PDF f(θ) peaks, their values, the half width and half maxima and the shape matches.
2. Create a file (the name doesn’t matter, although I’d suggest calling it something like ‘Torsions\_constrained\_thresholds’ for clarity) containing a list of the torsion angles you want to constrain followed by the separation threshold. The formatting is important: it needs to contain four atom labels (labels corresponding to atoms in XXXX.mol2) followed by the selected separation thresholds. Each torsion angle should be in a new line.

 *E.g.* if you want to apply a separation threshold of 45° to a torsion angle C1\_C2\_C3\_C4 your file should contain one line with C1 C2 C3 C4 45

The entire list of torsion angles picked by the CG is contained in the file ‘Torsions\_shapes’ produced in Part 1. You may modify that file adding the separation thresholds and removing the explicitly flexible torsion angles

1. Apply the Select\_conformers.py script:

 python Select\_conformers.py conformers.mol2 Torsions\_constrained\_thresholds

It will ask whether you want to overlay all the conformers on a common reference, to avoid including mirror images. I’d suggest writing always 1, unless you only want to generate enantiopure crystal structures. In that case I reckon you need to choose a conformer with the right handness (I never had that problem, it may require some testing).

1. This will outuput a file called selected\_conformers.mol2, which contains the conformational regions. The value taken by the considered torsion angles in each selected conformer are in ‘selected\_conformers’. ‘similar conformers’ lists that conformers that were eliminated by the selection script.

 Approach 2:

1. Create a file (the name doesn’t matter, although I’d suggest calling it something like ‘Locked\_torsions’ for clarity) containing a list of bonds around which the CG should not look for conformations. This list should include the torsion angles treated as explicitly flexible. The formatting is important, and each line should contain the labels of the two atoms that define that bond.

 *E.g.* if in Part 1 you have selected a torsion angles C1\_C2\_C3\_C4 for explicitly treatment then the file should contain a line with C2 C3

1. Apply the CSD Conformer Generator on the molecule. Use the script Do\_CG.py:

 python Do\_CG.py XXXX.mol2 --locked Locked\_torsions

1. The output conformers.mol2 should already contain the conformational regions. I suggest still applying the Select\_conformers.py script with dummy separation thresholds for the constrained torsion angles (*e.g.* 10°) to obtain their actual values.

The file containing the conformational regions can then be extracted using the ‘Extract\_conformers’ script that I have put in the ‘Calculate\_energy\_CRs’ directory. This wil create a conformers\_poolwith a set of directories each containing a specific conformer. There are several other scripts in that directory, they are quite self explanatory.

Part 3: Generate ΔEintra grids for the explicitly flexible torsion angles.

There are various ways to generate the grids, like you would do in a normal CrystalPredictor 1 run. Surrogate molecules can also be used.

 Part 4: Analyse torsion angles with polar hydrogen atoms

The methods listed thus far ignore polar H atoms. The CG tends to give them silly values at times. So they should be scanned *ab initio*, possibly with Gaussian, and dealt with according to the result of the scan. If they are very flexible they should be treated as search variables, otherwise they should be deemed to define conformational regions and constrained. Constraining them can be a bit tricky, and some scripting to modify the conformations in the conformers\_pool directory will be needed. I have nothing transferable for this aim.

 Part 5: Calculate the energy of the CRs

Optimise all the conformations in conformers\_pool, constraining the torsion angles that will be constrained in the search (*i.e.* those that actually define the conformational regions). Modifying ‘min\_input’ in the ‘Calculate\_energy\_CRs’ directory and using some of the scripts can help, but this can be modified. It should be quite easy.

 Part 6: Perform the searches

All the CRs that after the optimisations have a low enough energy should be searched. I suggest 26 kJ/mol in the paper as a threshold, but I reckon this could be changed in particular if there is competition between internal and intermolecular H-bonds. In the searches use the point charges from the optimisations and the grids calculated in part 3 for the explicitly flexible torsion angles. I have put a directory called Do\_searches that contains some of the scripts I used to set up the searches in CrystalPredictor 1.8, in particular ‘settingup’ and ‘Get Starting CP’. Explaining all of them here would take too long, but they are quite self explanatory and easy to understand and modify. I guess this procedure can be automated in other ways.