MS42 Solving Structures Through Combination of Reciprocal and Direct Space Methods

MS42-02

Automated and meta-structure solution from powder diffraction in Python using pyobjcryst

V. Favre-Nicolin [1]

[1]ESRF-The European Synchrotron - Grenoble (France)

Abstract

Twenty years ago the Fox software [1,2] was written to solve crystal structures from powder diffraction using global optimisation methods. The application was since improved to include all steps used in structure solution, including indexing and profile fitting. Despite the availability of powerful algorithms, solving structures can remain a tedious tasks with the lack of *a priori* knowledge, because the actual building blocks as well as the spacegroup (and thus the multiplicity and the number of independent atoms) are often ambiguous: solving a structure may require to test a large number of possible parametrisations, usually by hand.

A few years ago the objcryst library behind Fox was interfaced in python as part of the diffpy [3] suite in the pyobjcryst library, and more recently all the library components which can be used to solve structures (indexing, profile fitting, Monte Carlo, parallel tempering algorithms...) have been added to pyobjcryst. This allows to use most of the functionality of Fox inside a python notebook and a web browser, as shown in Figure 1.

We will also show that beyond the ability to solve structure solutions using a python script or notebook, it is possible to automatically loop through possible crystal configurations like building blocks (molecules, polyedra) and spacegroups, including parallelising the search using python's multiprocessing module, as is shown in the example notebooks available from pyobjcryst's online documentation [4].

References

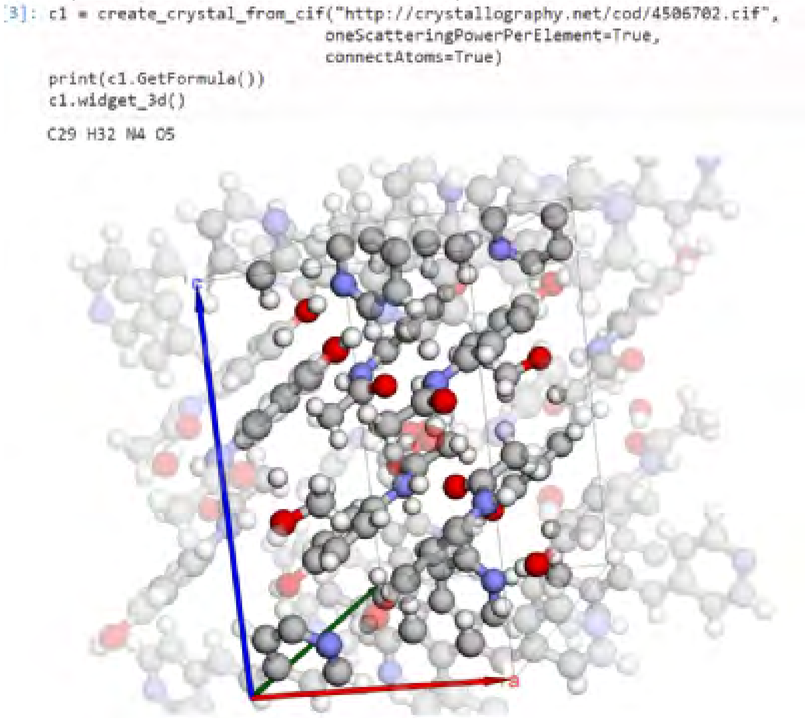
[1] V. Favre-Nicolin and R. Cerný, J. Appl. Cryst. (2002). 35, 734-743

[2] V. Favre-Nicolin and R. Cerný, Z. Kristallogr. 219 (2004) 847–856

[3] P. Juhas, C. L. Farrow, X. Yang, K. R. Knox, and S. J. L. Billinge, Acta Crystallogr. A 71, 562 (2015)

[4] https://pyobjcryst.readthedocs.io/

3D crystal structure from COD in a python notebook



```
[3]: c1 = create_crystal_from_cif("http://crystallography.net/cod/4506702.cif",
                                 oneScatteringPowerPerElement=True,
                                 connectAtoms=True)
     print(c1.GetFormula())
     c1.widget_3d()

     C29 H32 N4 O5
```