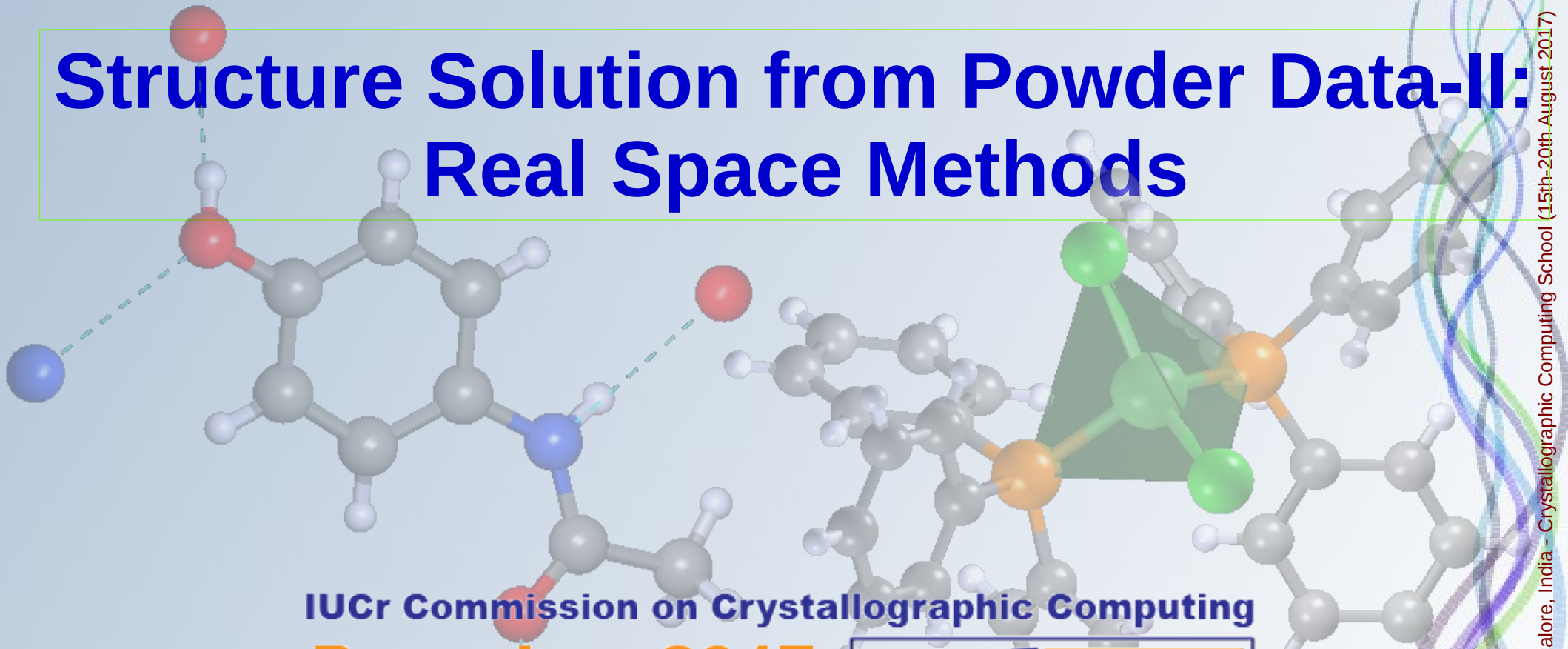
Structure Solution from Powder Data-II: Real Space Methods



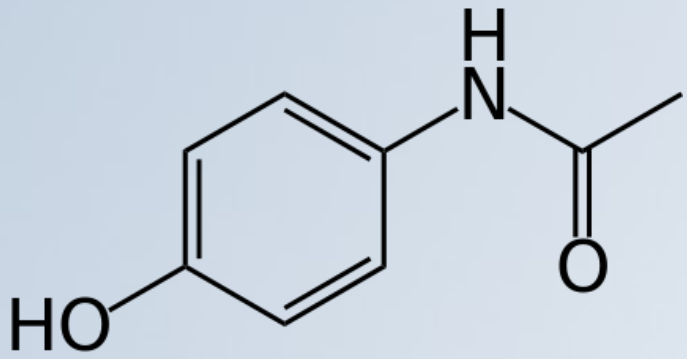
IUCr Commission on Crystallographic Computing

Bangalore 2017

Crystallographic Computing School

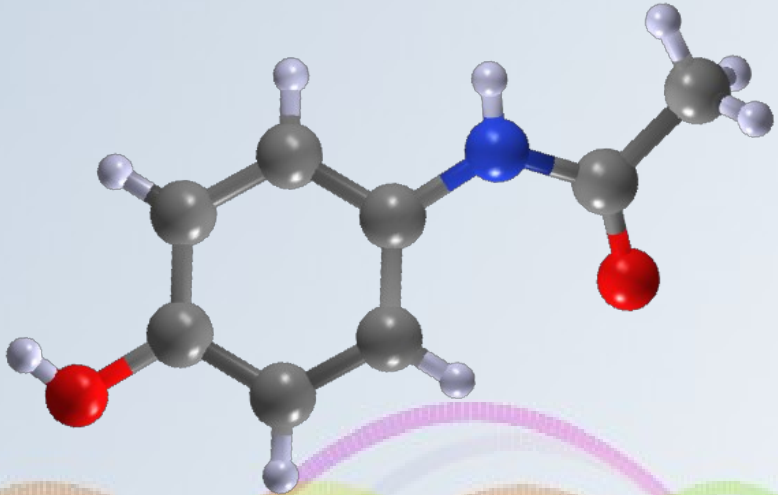
Building starting model

- It is necessary to know the molecular connectivity. Spectroscopic techniques (MS, NMR) can be useful

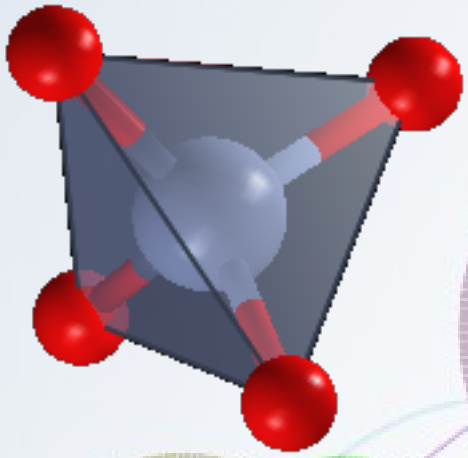


- Crystal structure can be described as a combination of building blocks

atom



molecule



polyhedron

Building starting model

- **Check for similar molecules** in databases or in the literature

- **Optimize molecular geometry** by computational chemistry programs

http://www.ba.ic.cnr.it/ochemdb/
Login: CCS2017
Password: IUCR2017

Crystal Structure Databases

Non-commercial database are in red

- **CSD** (Cambridge Structural Database) (organics & organometallics): http://www.ccdc.cam.ac.uk/
- **ICSD** (Inorganic Crystal Structure Database) (inorganics, elements, minerals & intermetallics): http://icsd.ill.fr/
- **COD** (Crystallography Open Database) (general database): http://www.crystallography.net/

Other databases: ICDD PDF-4+, American Mineralogist Crystal Structure Database, MINCRYST, Zeolite Structures Database, ...

File format: CIF (Crystallographic Information File)

Free Chemistry Databases

- PubChem: https://pubchem.ncbi.nlm.nih.gov/
- NIST Chemistry WebBook: http://webbook.nist.gov/chemistry/
- Drugbank: http://www.drugbank.ca/

Other databases: ZINC, eMolecules, ChEBI, NMRShiftDB, ...

Chemical file formats: *sdf, mol, mol2, cml, SMILES*

2D molecular structures must be optimized before being used for structure solution

Geometry optimization

Three levels of theory

- Molecular-mechanics force fields (**MM**)
- Semi-empirical methods (**SE**)
- *Ab initio* methods: Hartree–Fock methods, density functional theory (**DFT**)

Strategy: **MM** → **SI** → **DFT**

Programs: MOPAC, Gamess, NWChem, Gaussian, ABINIT, ORCA, Molpro, Q-Chem, octopus, etc.

Molecule editor

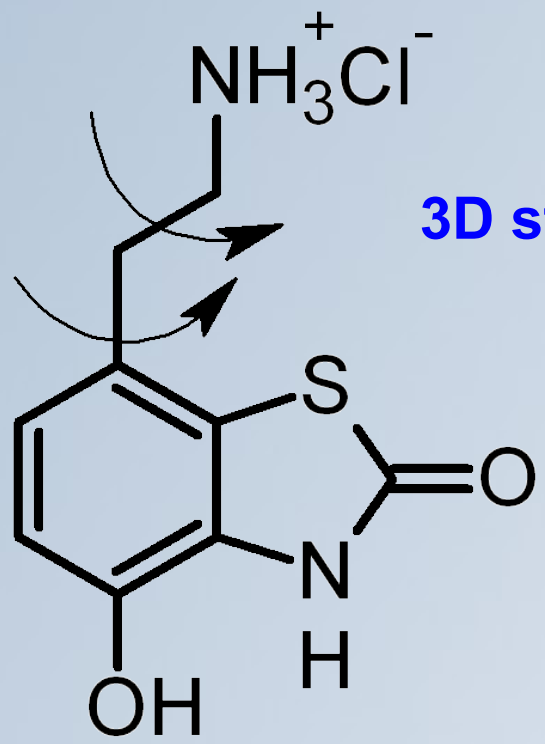
A molecule editor allows

- Sketch molecules in 2D or 3D format
- Optimize the geometry by force field method
- Create input file for the quantum-chemistry calculations
- Read output files of the most common computational packages

Some free available software

- ACD/ChemSketch - http://www.acdlabs.com
- Avogadro - http://avogadro.openmolecules.net/wiki/Main_Page
- MarvinSketch - http://www.chemaxon.com/products/marvin/
- Gabedit: http://gabedit.sourceforge.net/

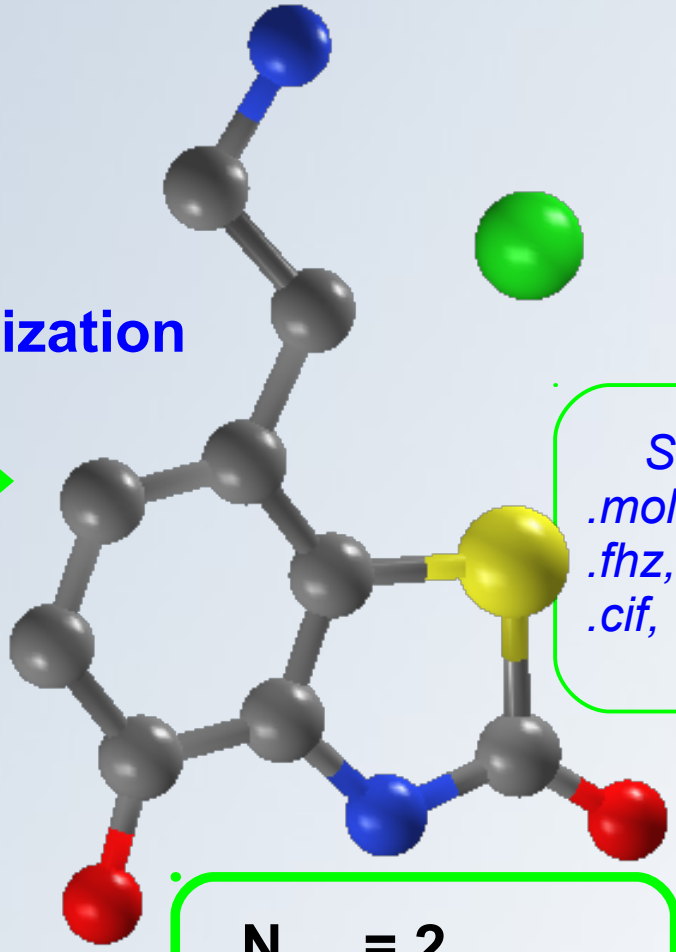
Building starting model



2-(4-Hydroxy-2-oxo-2,3-dihydro-1,3-benzothiazol-7-yl)
ethylammonium chloride

P 21/a

3D structure optimization

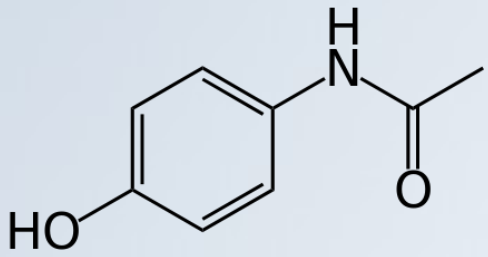


Save as
.mol, .mol2,
.fhz, .mop,
.cif, .pdb, .frac

$N_{frag} = 2$

$N_{dof} = 6 + 3 + 2$

```
program example4
!
!  Crystal structure solution of paracetamol (form I) by real space method.
!
   use iso_fortran_env, only: stdout => OUTPUT_UNIT,ERROR_UNIT
   use crystal_phase
   use gen_frm
   use errormod
   use datasetmod
   use prog_constants
   use variables, only: cryst,dataset
   use general, only: lo
   use molcom, only: kscreen
   use sannel
   implicit none
   type(error_type)              :: err
   type(dataset_type)            :: datas
   integer                       :: ier
   type(SimAnn_Conditions_type)  :: sc
   type(bb_bc_condition)         :: bcond
!
!  Initialize libexpo
   call InitExpo2002(0)
   lo = stdout
   kscreen = 0
   call load_chemical_tables('../files/',err)
   if (err%signal) then
       write(ERROR_UNIT,'(a)')' Message: '//err%msg()
       stop 1
   endif
!
!  Set up crystal phase
   call new_phases(cryst,1)
   call cryst(1)%set_symmetry(init_spaceg_type('P 21/n'), set_cell_type([7.100,9.380,11.708,90.0,97.42,90.0]))
   call crystal_file_import(cryst(1),'paracetamol.mol',has_symmetry=.true.,err=err)
   if (err%signal) stop 2
!
!  Set up XRPD data set
   call datas%open_file('paracetamol.xy',ier,wavel=CU_WAVE)
   if (ier /= 0) stop 3
   call push_back_dataset(dataset,datas)
   call dataset_to_expo(dataset(1))
!
!  Run global optimization
   call sa_prelim(goptim,sc,bcond,.true.,saerr=err); if (err%signal) stop 4
   call sa_run(goptim,sc,bcond)
!
   end program example4
```



Paracetamol (form I polymorph)
$C_8H_9NO_2$

H atoms

H atoms do not contribute significantly to X-ray diffraction, they can be ignored during the structure solution

Eliminating the H atoms reduces the number of atoms and DoFs, decreasing the time to evaluate CF for each trial structure

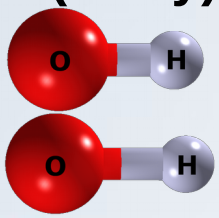
Delete H atoms

```
subroutine remove_atoms_from_list(atoms,veta,val,[bonds],[iord])
```

e.g. `call remove_atoms_from_list(atoms,atoms%z(),H_at,bonds)`

Add H atoms at X-ray distances

X-H (X-ray) < X-H (neutrons)

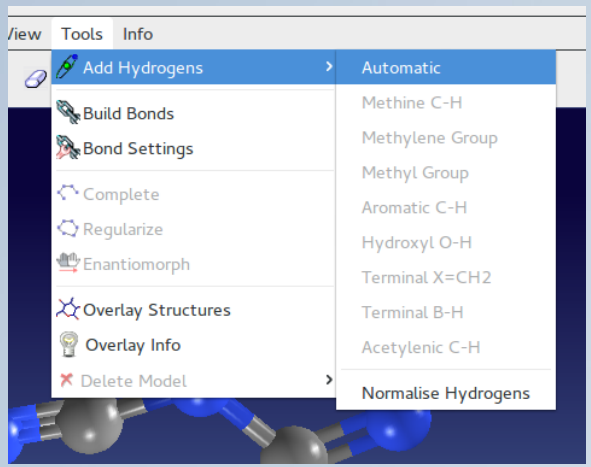


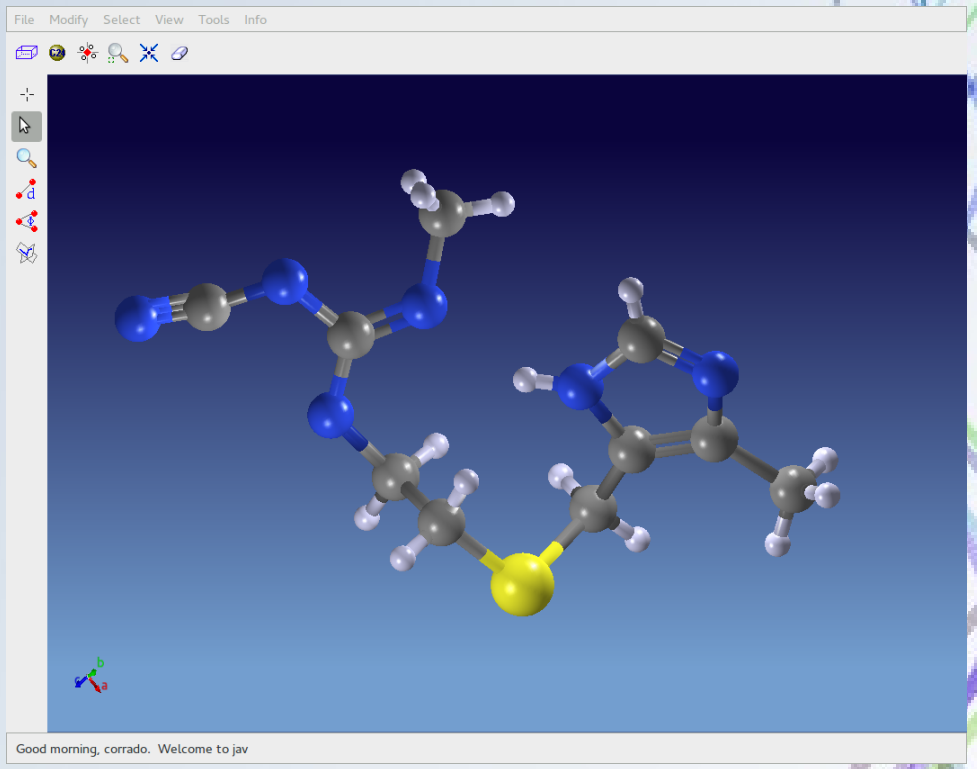
0.82 Å (X-ray)

0.95 Å (neutrons)

```
subroutine add_hydrogens(atoms,bonds,elem,cell,spg,[vat],[htype],ier)
```

Hydrogen calculation



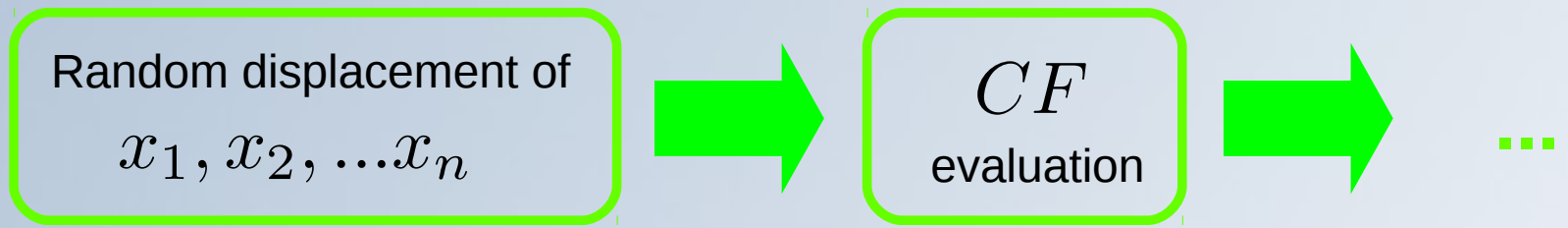


Hydrogen atoms are positioned geometrically

Program organization

$$CF = f(\mathbf{X}) = f(x_1, x_2, ...x_n)$$

- Parameters are adjusted **simultaneously**



- Parameters are adjusted **individually and sequentially**

In the calculation of the structure factors only the contributes of parameter x_i will be update

Program organization

$$|\mathbf{F}_{hkl}|^2 = |\mathbf{A}_{hkl}|^2 + |\mathbf{B}_{hkl}|^2$$

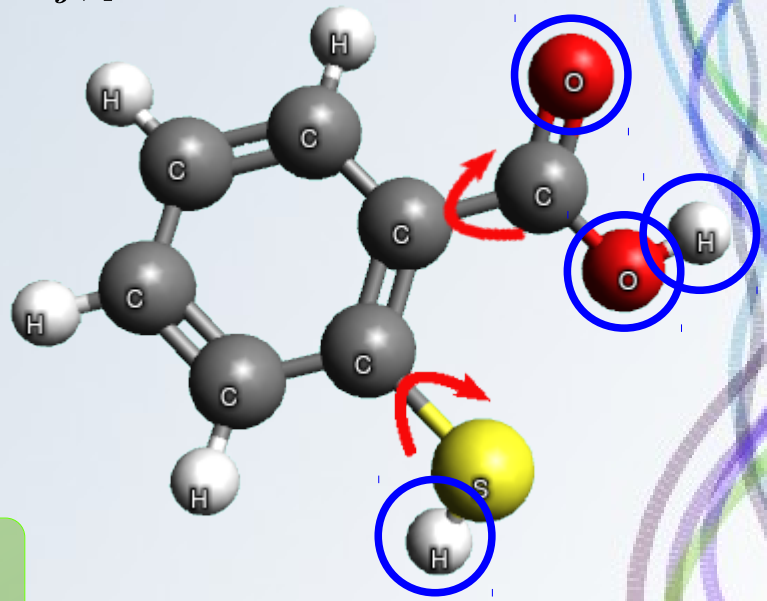
$$\mathbf{A}_{hkl} = \sum_{j=1}^{N} f_j cos[2\pi(hx_j + ky_j + lz_j)]$$

$$\mathbf{B}_{hkl} = \sum_{j=1}^{N} f_j sin[2\pi(hx_j + ky_j + lz_j)]$$

Centrosymmetric crystal structure:

$$\mathbf{F}_{hkl} = \sum_{j=1}^{N} f_j cos[2\pi(hx_j + ky_j + lz_j)]$$

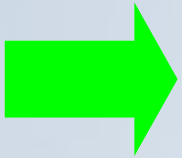
In the calculation of the structure factors only the contributes of the displaced atoms will be update.



Rotated atoms are circled in blue

Simulated annealing options

- Cost function
- Resolution
- Random seed
- **Number of moves**
- **Number of SA runs**
- Starting temperature
- Temperature reduction factor



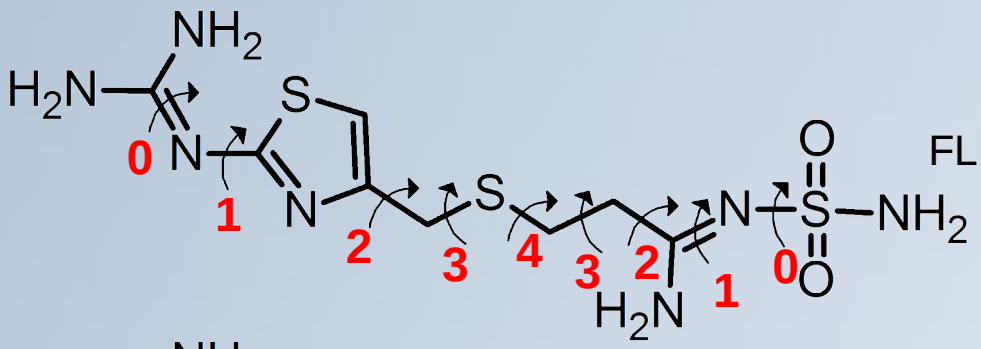
- No. of molecular fragments
- No. of external DoFs
- No. of internal DoFs
- The flexibility of the molecule

```
!
!   Run global optimization
    call sa_prelim(goptim,sc,bcond,.true.,saerr=err); if (err%signal) stop 4
    goptim%nrun = 100
    sc%nt=100; sc%autont=.false.
    call sa_run(goptim,sc,bcond)
```

FLEX parameter

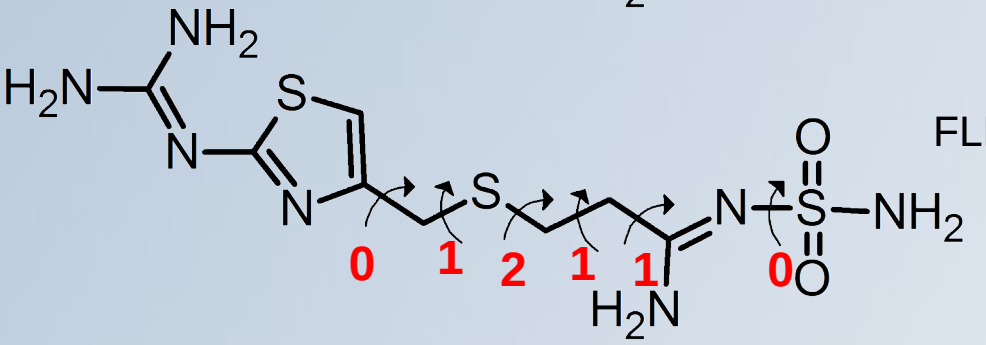
$$FLEX = nDoF + \sum_{i}^{nDoF} f_i$$

$nDoF$ = number of internal DoFs
f_i = number of coupled DoFs

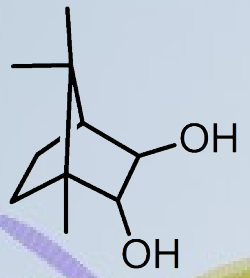


FLEX = 9 + (0 + 1 + 2 +3 + 4 + 3 + 2 + 1 + 0) = 25

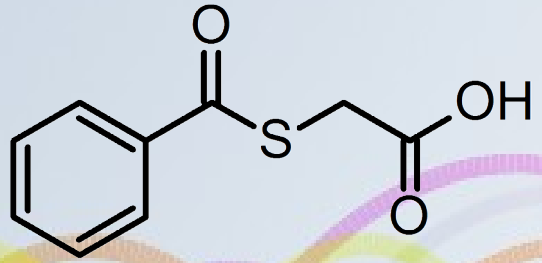
In red the values of f for each internal DoF in the famotidine. H-atoms were ignored



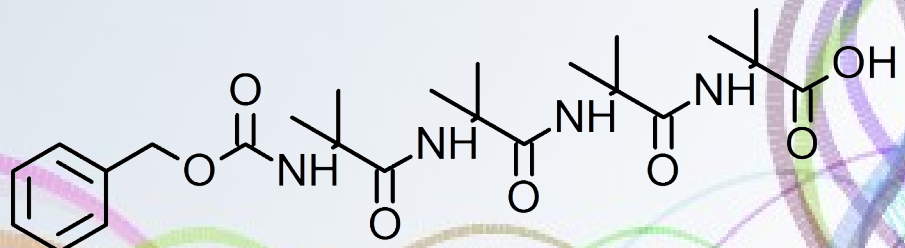
FLEX = 6 +(0 + 1 + 2 +1 + 1 + 0) = 11



FLEX=0



FLEX=7



FLEX=64

Cost Functions

Whole profile R factors: R_p, R_{wp}, χ^2

$$R_{wp} = \sqrt{\frac{\sum_i w_i (y_{exp}(\theta_i) - y_{calc}(\theta_i))^2}{\sum_i w_i y_{exp}(\theta_i)^2}}$$

$$2\theta_0 - f * FWHM < y(\theta_i) < 2\theta_0 + f * FWHM \;\; f = 1$$

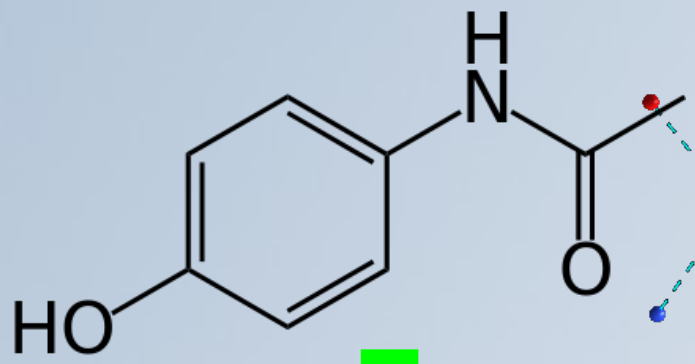
Integrated intensities R factors: R_B, R_{Bg}

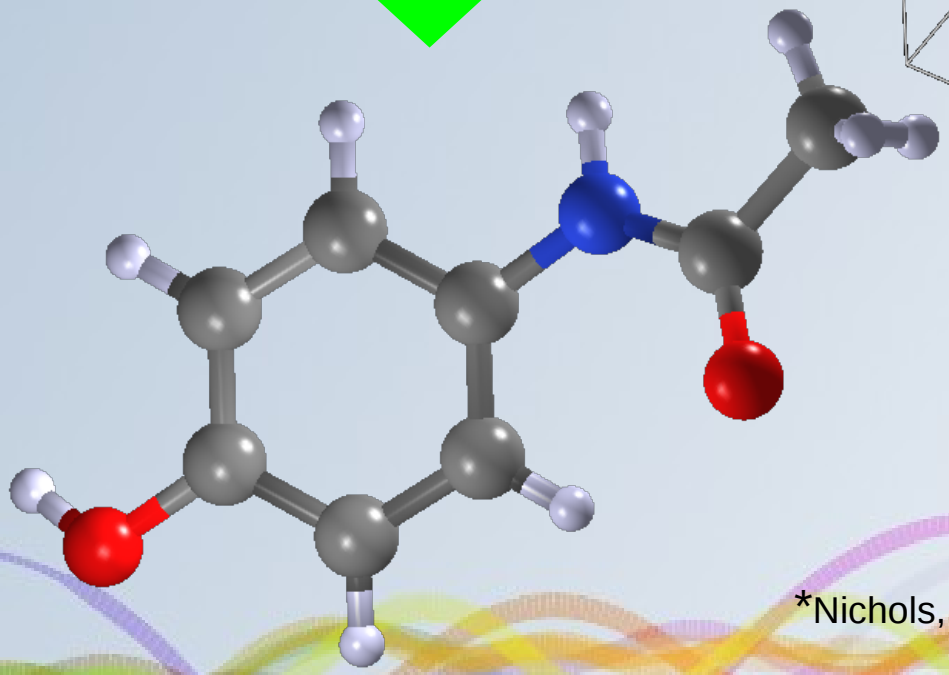
$$R_B = \frac{\sum_h |I_h^{exp} - I_h^{calc}|}{\sum_h I_h^{exp}}$$

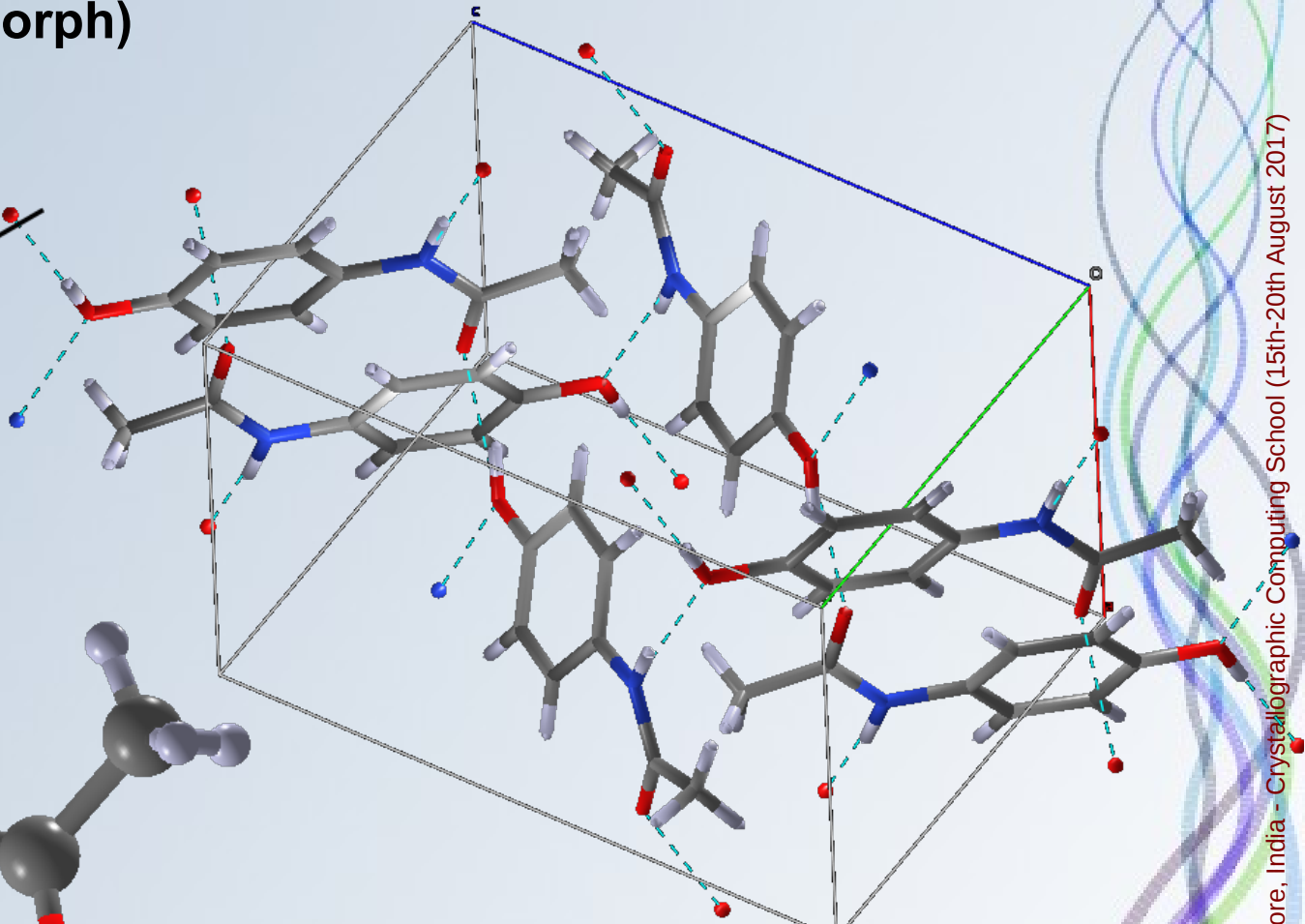
Other cost functions: $CF_{\text{potential energy}}, CF_{\text{geometry restraints}},$
$CF_{\text{bond valecence}}, CF_{\text{antibumping}}$

Molecular compounds

Paracetamol (form I polymorph)
($C_8H_9NO_2$) *

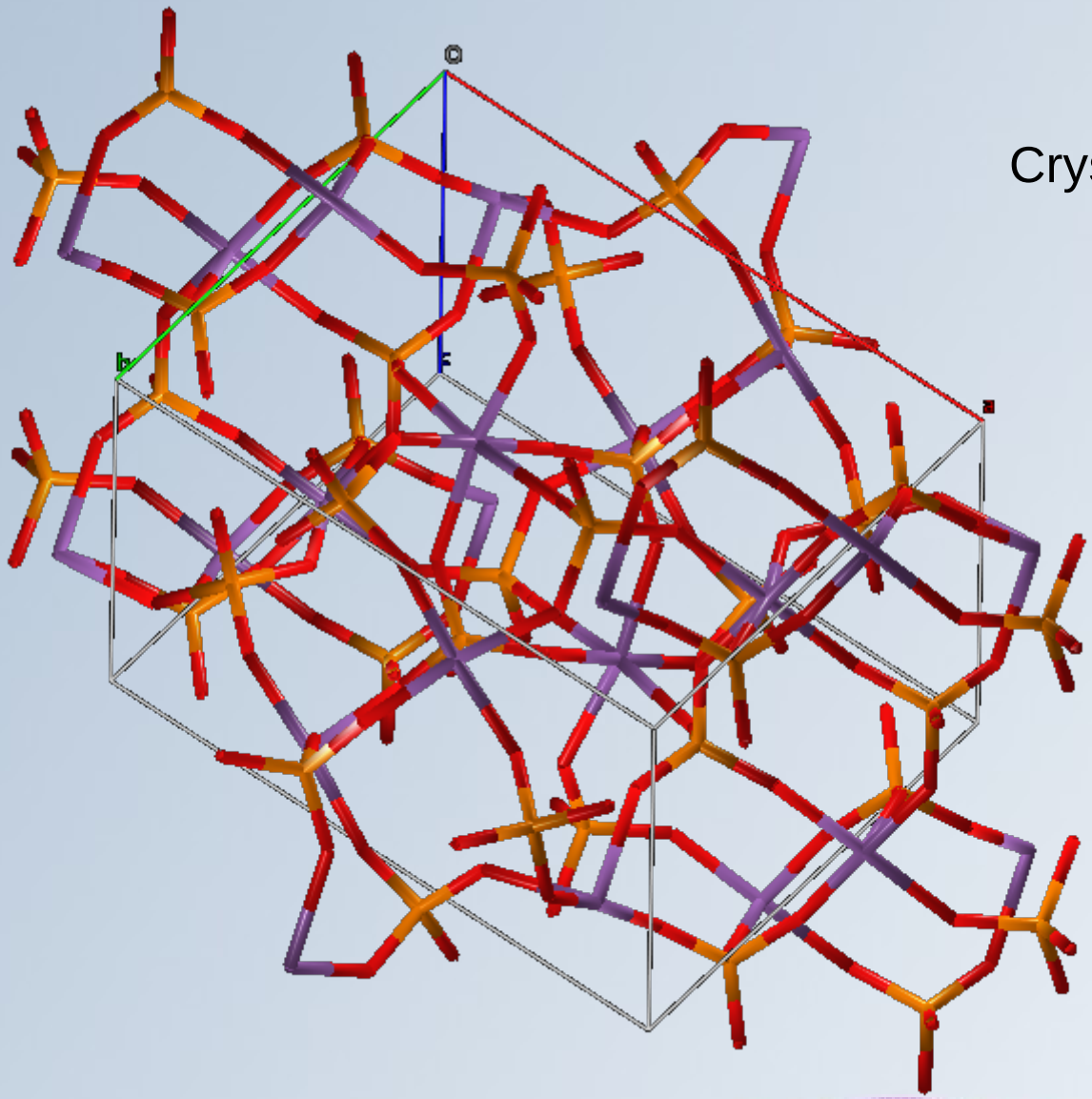






*Nichols, C. & Frampton, C. S. (1998). J. Pharm. Sci. 87, 684–693.

Non-molecular compounds

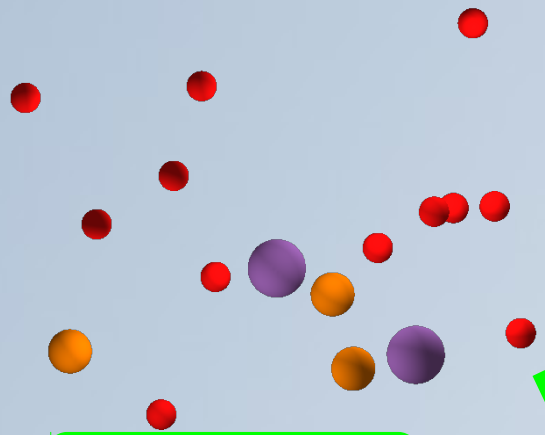


Crystal structure of **$Sb_2(PO4)_3$** *

*Jouanneaux, A., Verbaere, A., Guyomard, D., Piffard, Y., Oyetola, S. & Fitch, A. N. (1991). *Eur. J. Solid State Inorg. Chem.* **28**, 755-765.

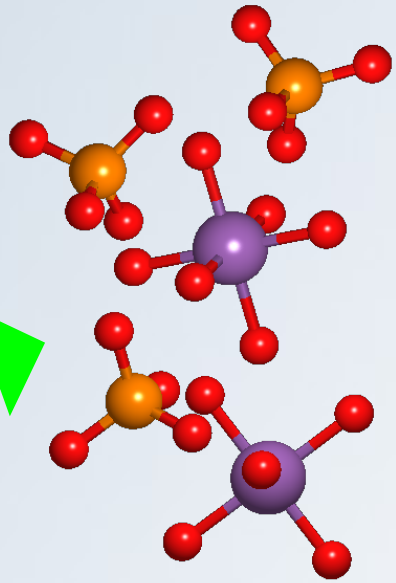
Non-molecular compounds

$$\frac{N_{obs}}{N_{dof}} > 8$$

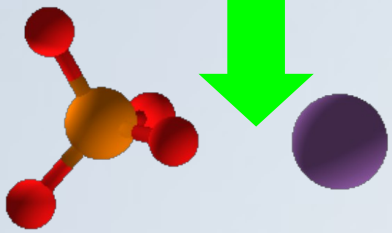


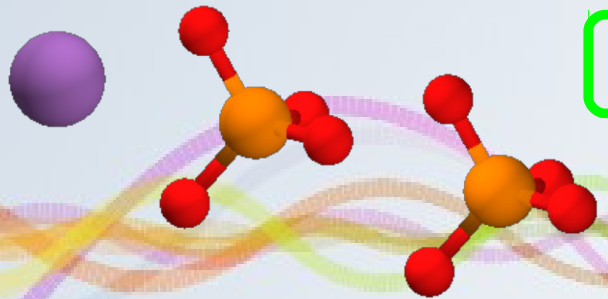
$N_{dof} = 51$

$Sb_2(PO_4)_3$



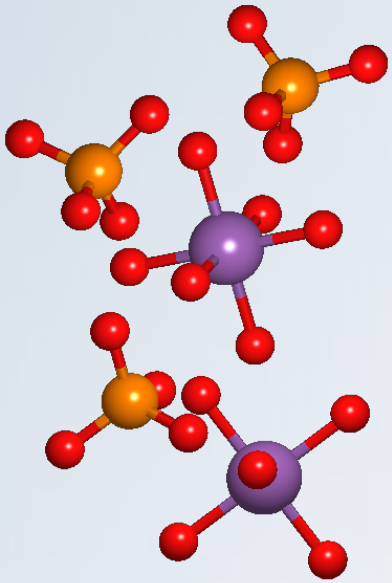
$N_{dof} = 30$



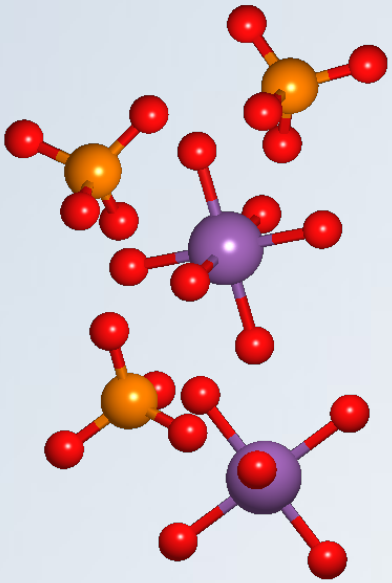


$N_{dof} = 24$

```
  program example5
!
!  Crystal structure solution of Sb2(PO4)3 by real space method.
!
   use iso_fortran_env, only: stdout => OUTPUT_UNIT,ERROR_UNIT
   use crystal_phase
   use gen_frm
   use errormod
   use datasetmod
   use prog_constants
   use variables, only: cryst,dataset
   use general, only: lo,alambda
   use molcom, only: kscreen
   use sannel
   implicit none
   type(error_type)              :: err
   type(dataset_type)            :: datas
   integer                       :: ier
   type(SimAnn_Conditions_type)  :: sc
   type(bb_bc_condition)         :: bcond
!
!  Initialize libexpo
   call InitExpo2002(0)
   lo = stdout
   kscreen = 0
   call load_chemical_tables('../files/',err)
   if (err%signal) then
       write(ERROR_UNIT,'(a)')' Message: '//err%msg()
       stop 1
   endif
!
!   Set up crystal phase
    call new_phases(cryst,1)
    call cryst(1)%set_symmetry(init_spaceg_type('P 21/n'),set_cell_type([11.936,8.7354,8.3185,90.0,91.12,90.0]))
    call crystal_fragment_import(cryst(1),'tetra P O',RX_SOURCE,err); if (err%signal) stop 2
    call crystal_fragment_import(cryst(1),'tetra P O',RX_SOURCE,err); if (err%signal) stop 2
    call crystal_fragment_import(cryst(1),'tetra P O',RX_SOURCE,err); if (err%signal) stop 2
    call crystal_fragment_import(cryst(1),'octa Sb O',RX_SOURCE,err); if (err%signal) stop 2
    call crystal_fragment_import(cryst(1),'octa Sb O',RX_SOURCE,err); if (err%signal) stop 2
!
!   Set up XRPD data set
    call datas%open_file('sbpo.dat',ier,wavel=1.45072,sync=.true.)
    if (ier /= 0) stop 3
    call push_back_dataset(dataset,datas)
    call dataset_to_expo(dataset(1))
!
!   Run global optimization with d.o.c
    call sa_prelim(goptim,sc,bcond,.true.,saerr=err)
    call cryst(1)%set_doc(.true.)
    call sa_run(goptim,sc,bcond)
!
    end program example5
```

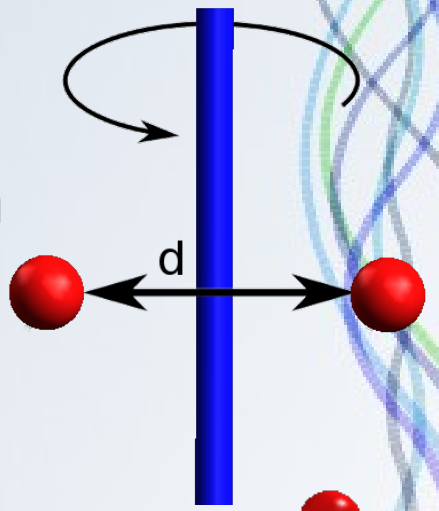


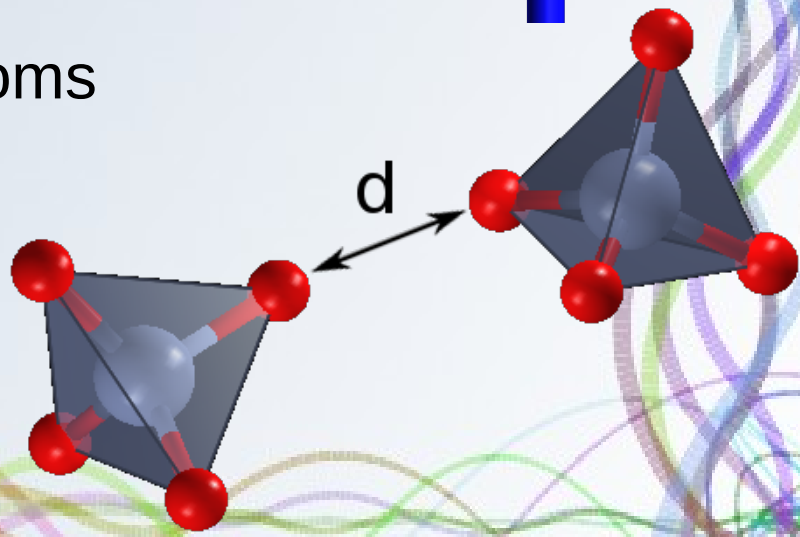
```
  program example5
!
!  Crystal structure solution of Sb2(PO4)3 by real space method.
!
   use iso_fortran_env, only: stdout => OUTPUT_UNIT,ERROR_UNIT
   use crystal_phase
   use gen_frm
   use errormod
   use datasetmod
   use prog_constants
   use variables, only: cryst,dataset
   use general, only: lo,alambda
   use molcom, only: kscreen
   use sannel
   implicit none
   type(error_type)              :: err
   type(dataset_type)            :: datas
   integer                       :: ier
   type(SimAnn_Conditions_type)  :: sc
   type(bb_bc_condition)         :: bcond
!
!  Initialize libexpo
   call InitExpo2002(0)
   lo = stdout
   kscreen = 0
   call load_chemical_tables('../files/',err)
   if (err%signal) then
       write(ERROR_UNIT,'(a)')' Message: '//err%msg()
       stop 1
   endif
!
!  Set up crystal phase
   call new_phases(cryst,1)
   call cryst(1)%set_symmetry(init_spaceg_type('P 21/n'),set_cell_type([11.936,8.7354,8.3185,90.0,91.12,90.0]))
   call crystal_fragment_import(cryst(1),'tetra P O',RX_SOURCE,err); if (err%signal) stop 2
   call crystal_fragment_import(cryst(1),'tetra P O',RX_SOURCE,err); if (err%signal) stop 2
   call crystal_fragment_import(cryst(1),'tetra P O',RX_SOURCE,err); if (err%signal) stop 2
   call crystal_fragment_import(cryst(1),'octa Sb O',RX_SOURCE,err); if (err%signal) stop 2
   call crystal_fragment_import(cryst(1),'octa Sb O',RX_SOURCE,err); if (err%signal) stop 2
!
!  Set up XRPD data set
   call datas%open_file('sbpo.dat',ier,wavel=1.45072,sync=.true.)
   if (ier /= 0) stop 3
   call push_back_dataset(dataset,datas)
   call dataset_to_expo(dataset(1))
!
!  Run global optimization with d.o.c
   call sa_prelim(goptim,sc,bcond,.true.,saerr=err)
   call cryst(1)%set_doc(.true.)
   call sa_run(goptim,sc,bcond)
!
   end program example5
```



Non-molecular compounds

- You cannot know the number and the type of the polyhedra
- Some atoms are expected to fall on special position
- Different building blocks share some atoms





Dynamical occupancy correction (DOC)

- Falcioni, M. & Newsam, J. M. (1989). *Nature* **342**, 260-262.
- Favre-Nicolin, V. & Černỳ, R. (2002). *J. Appl. Cryst.*. **35**, 734-743

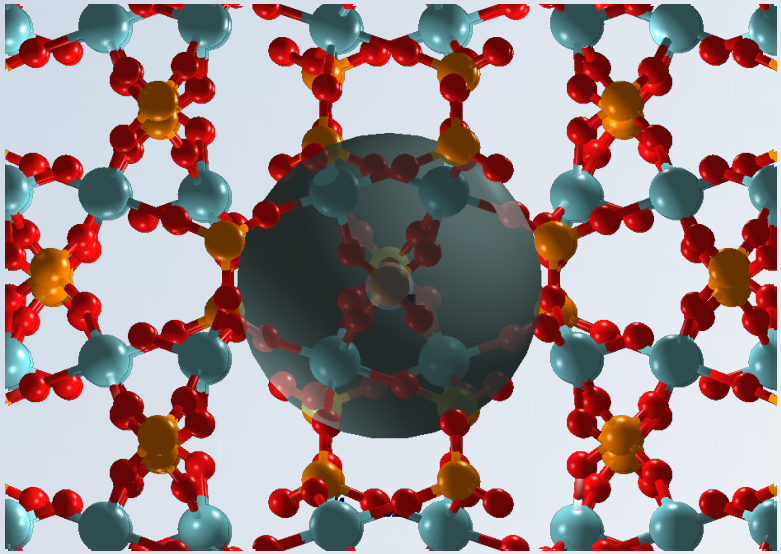
$$occupancy = \frac{1}{1 + \sum_{neighbour} |d_{min} - d_i|} \qquad d_{min} = 1\mathring{A}$$

```
USE atom_type_util
call compute_doc_atoms(atom,cell,spg,distmin)
```

DOC is able to merge the excess atoms automatically

Range searching of atoms

Given a set of atoms in the space, which of those atoms fall within some specified area?

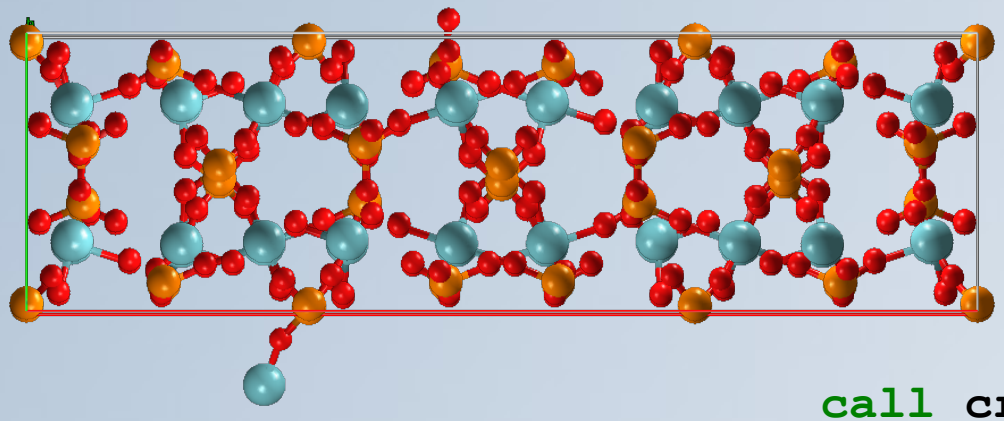


Methods available *

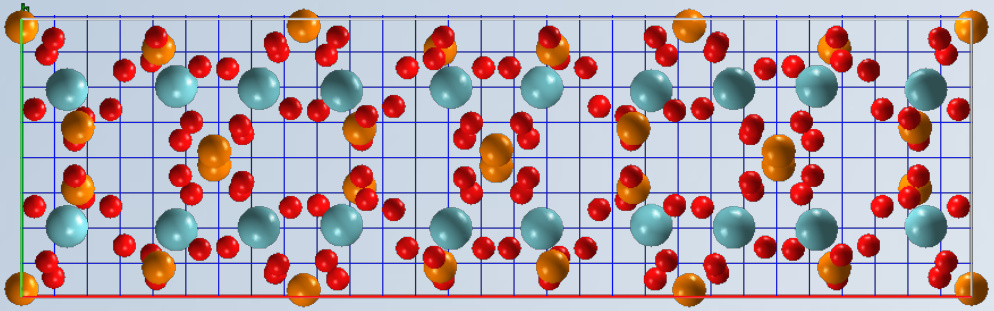
- Sequential search
- Grid method

* R. Sedgewick, *Algorithms in C*, Addison-Wesley publishing company, 1983, chap. 26

Range searching of atoms



```
call crystal%make_symmetry_cell()
```



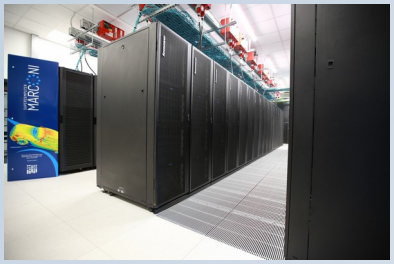
```
call crystal%allocate_grid(cutoff=1.0)

call crystal%fill_grid()

call crystal%compute_doc(1.0)
```

Parallel Machines

Supercomputers



Marconi by CINECA (Italy)
244.800 cores in total

Workstations

Typically 4-18 cores

Notebooks

Typically 2-4 cores

Smartphones

Typically 2-4 cores

Graphical Processing Units

Up to 3000 cores

Three Protocols

- **Message Passing Interface (MPI)**

  Distributed-memory systems

- **Open MultiProcessing (OpenMP)**

  Shared-memory systems

- **Compute Unified Device Architecture (CUDA)**
  Open Computing Language (OpenCL)

  Coprocessor architecture

Three Protocols

- **Message Passing Interface (MPI)**

  Distributed-memory systems

- **Open MultiProcessing (OpenMP)**

  Shared-memory systems

- **Compute Unified Device Architecture (CUDA)**
  Open Computing Language (OpenCL)

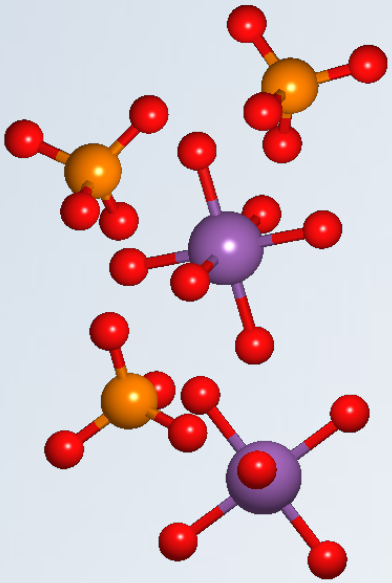
  Coprocessor architecture

Running the Parallel Version of Expo2014

- Computer with multi-core CPUs and Linux environment.
- Open MPI installed.
- Compiling Expo2014 from source and linking with MPI libraries
- Run Expo2014 by using the launcher **`mpirun`** with the appropriate options.

```
mpirun -np 10 expo input_file.exp
```

```
   program example5
!
!  Crystal structure solution of Sb2(PO4)3 by real space method.
!
   use iso_fortran_env, only: stdout => OUTPUT_UNIT,ERROR_UNIT
   use crystal_phase
   use gen_frm
   use errormod
   use datasetmod
   use prog_constants
   use variables, only: cryst,dataset
   use general, only: lo
   use molcom, only: kscreen
   use sannel
   use bb_bc
   use mpi_prog
   implicit none
   type(error_type)              :: err
   type(dataset_type)            :: datas
   integer                       :: ier
   type(SimAnn_Conditions_type)  :: sc
   type(bb_bc_condition)         :: bcond
   integer                       :: mpi_err
!
   call mpi_prog_init()
!
!  Initialize libexpo
   call InitExpo2002(0)
   lo = stdout
   kscreen = 0
   call load_chemical_tables('../files/',err)
   if (err%signal) then
       write(ERROR_UNIT,'(a)')' Message: '//err%msg(); stop 1
   endif
!
!  Set up crystal phase
   call new_phases(cryst,1)
   cryst(1)%cr_name='sbpo'
   call cryst(1)%set_symmetry(init_spaceg_type('P 21/n'),set_cell_type([11.936,8.7354,8.3185,90.0,91.12,90.0]))
   call crystal_fragment_import(cryst(1),'tetra P O',RX_SOURCE,err); if (err%signal) stop 2
   call crystal_fragment_import(cryst(1),'tetra P O',RX_SOURCE,err); if (err%signal) stop 2
   call crystal_fragment_import(cryst(1),'tetra P O',RX_SOURCE,err); if (err%signal) stop 2
   call crystal_fragment_import(cryst(1),'octa Sb O',RX_SOURCE,err); if (err%signal) stop 2
   call crystal_fragment_import(cryst(1),'octa Sb O',RX_SOURCE,err); if (err%signal) stop 2
!
!  Set up XRPD data set
   call datas%open_file('sbpo.dat',ier,wavel=1.45072,sync=.true.)
   if (ier /= 0) stop 3
   call push_back_dataset(dataset,datas)
   call dataset_to_expo(dataset(1))
!
!  Run global optimization with d.o.c
   call sa_prelim(goptim,sc,bcond,.true.,saerr=err)
   call cryst(1)%set_doc(.true.)
   call sa_run(goptim,sc,bcond)
   call MPI_Finalize(mpi_err)
!
   end program example5
```



1) ./configure FC=mpif90
2) mpirun -np 4 ./example5

DS with Low Quality Diffraction Pattern

- **Bond valence restraints**
- **Anti-bumping restraints**
- **Molecular geometry restraints**

Bond Valence Restraints

Atomic valence V_i of atom *i* in crystal structure is the sum of individual bond valences S_{ij}

$$V_i = \sum_j S_{ij} \qquad S_{ij} = exp(\frac{R_0 - R_{ij}}{B})$$

R_{ij} distance between atoms *i* and *j*

R_0, B bond valence parameters (**`bvparmyyyy.cif`** maintened by I.D. Brown and available from http://www.iucr.org/resources/data/datasets/bond-valence-parameters)

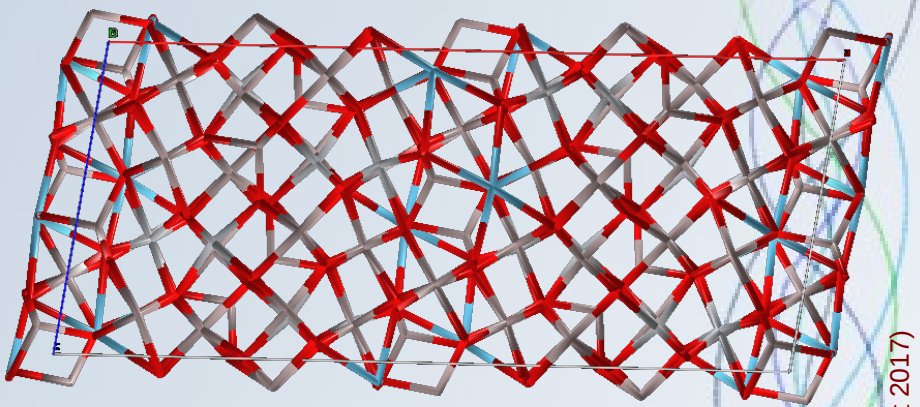
$$G_{ii} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(V_i - V_i^0)^2}$$ **global-instability index**

The estimated values V_i can be incorporated as restraints in the cost function (*):

$$CF_{VB} = \sum_i w_i (V_i - V_i^0)^2$$

* J. Pannetier, J. Bassas-Alsina, J. Rodriguez-Carvajal & V. Caignaert, (1990). Nature 346, 343 - 345

```
   program example6
   use iso_fortran_env, only: stdout => OUTPUT_UNIT, ERROR_UNIT
   use crystal_phase
   use gen_frm
   use errormod
   use symmgrid
   implicit none
   type(crystal_phase_t)                          :: cryst
   type(error_type)                               :: err
   integer                                        :: ia,ncoord
   real, dimension(:), allocatable                :: bvs
   type(bond_info_t), dimension(:), allocatable :: dinfo
!
   call load_chemical_tables('../files/',err)
   if (err%signal) then
       write(ERROR_UNIT,'(a)')' Message: '//err%msg(); stop 1
   endif
!
   call crystal_file_import(cryst,'LaTi_true.cif',err=err)
   if (err%signal) stop 2
!
!  Bond valence parameters
   call cryst%load_bvparam()
   call cryst%print_bvparam(stdout)
!
!  Initialization for bond valence
   call cryst%make_symmetry_cell()
   call cryst%allocate_grid(cutoff=maxval(cryst%bvpar(:,:)%rmax))
   call cryst%fill_grid()
!
!  Bond valence sum and additional info
   allocate(bvs(cryst%natoms()))
   do ia=1,cryst%natoms()
      call cryst%compute_bvs(ia,bvs(ia))
      write(stdout,'(/2x,a,a8,a,f10.3)') 'Atom: ',cryst%at(ia)%glab(),' BVS: ',bvs(ia)
      ncoord = cryst%coord_number(ia)
      if (ncoord > 0) then
          call new_bond_info(dinfo,ncoord)
          call cryst%valence_atom_info(ia,dinfo,bvs(ia),ncoord)
          call print_bond_info(cryst%at,cryst%spg,dinfo,stdout)
      endif
   enddo
!
   write(stdout,'(a,f10.3)')'GII=',cryst%gii_index(bvs)
   end program example6
```



$LaTi_2Al_9O_{19}$ *

```
corrado@Z420-1: ~/libexpo/test/example3
File  Edit  View  Search  Terminal  Help

 ------------------------------------------------------------------------
            Bond Valence Parameters and bond windows
 Atom1          Atom2                Ro        B       rmin      rmax
 ------------------------------------------------------------------------
 O2-            O2-                1.406     0.370    1.000     0.000
 Al3+           O2-                1.651     0.370    1.318     2.270
 Ti4+           O2-                1.815     0.370    1.704     2.304
 La3+           O2-                2.086     0.450    2.100     3.000

  Atom:      La1 BVS:     2.709
Bonds for atom:
    1      La1       La  0.4458  0.8695  0.0652 x, y, z + (0,0,0)
 ------------------------------------------------------------------------
   27      O14        O  0.3564  0.7510 -0.0470              -x, -y, -z + (1,0,0)   2.513
   20      O7         O  0.3544  0.8730  0.1926                 x, y, z + (0,0,0)   2.562
   22      O9         O  0.3497  1.0020 -0.0590              -x, -y, -z + (0,2,0)   2.742
   25      O12        O  0.4497  0.6280  0.0591                 x, y, z + (0,0,0)   2.659
   28      O15        O  0.4537  0.7540 -0.2050                x, y, z + (0,0,-1)   2.949
   15      O2         O  0.5457  0.7660  0.1670                x, y, z + (0,0,-1)   2.590
   16      O3         O  0.5473  0.9990  0.1690                x, y, z + (0,0,-1)   2.760
   15      O2         O  0.4543  0.7660  0.3330           -x, y, -z+1/2 + (0,0,1)   2.823
   16      O3         O  0.4527  0.9990  0.3310           -x, y, -z+1/2 + (1,0,1)   2.936
   16      O3         O  0.4527  1.0010 -0.1690              -x, -y, -z + (1,1,1)   2.723
   29      O16        O  0.4455  1.1300  0.0800              -x, -y, -z + (1,1,0)   2.869

  Atom:      Ti1 BVS:     4.499
Bonds for atom:
    2      Ti1       Ti  0.6045  0.8814  0.1285 x, y, z + (0,0,0)
 ------------------------------------------------------------------------
   29      O16        O  0.5545  0.8700 -0.0800                x, y, z + (0,0,-1)   2.174
   15      O2         O  0.5457  0.7660  0.1670                x, y, z + (0,0,-1)   1.914
   16      O3         O  0.5473  0.9990  0.1690                x, y, z + (0,0,-1)   1.911
   22      O9         O  0.6503  0.9980  0.0590                 x, y, z + (0,0,0)   1.839
   32      O19        O  0.6409  0.7510  0.0380                 x, y, z + (0,0,0)   1.930
   20      O7         O  0.6456  0.8730  0.3074           -x, y, -z+1/2 + (1,0,0)   1.849

  Atom:      Ti2 BVS:     4.222
Bonds for atom:
    3      Ti2       Ti  0.6895  0.6180  0.1308 x, y, z + (0,0,0)
 ------------------------------------------------------------------------
   25      O12        O  0.6487  0.4950  0.0510                x, y, z + (0,0,-1)   1.752
   31      O18        O  0.6526  0.6280  0.3012             x, -y, z+1/2 + (0,0,0)   1.967
   32      O19        O  0.6409  0.7510  0.0380                 x, y, z + (0,0,0)   1.966
   21      O8         O  0.7443  0.6090 -0.0024       -x+1/2, -y+1/2, -z + (0,0,1)   1.924
   26      O13        O  0.7519  0.5070  0.2405   -x+1/2, y+1/2, -z+1/2 + (1,-1,1)   2.042
   18      O5         O  0.7452  0.7420  0.2602                x, y, z + (0,0,-1)   2.135

  Atom:      Al1 BVS:     2.948
Bonds for atom:
    4      Al1       Al  0.7717  0.8725  0.1791 x, y, z + (0,0,0)
 ------------------------------------------------------------------------
   18      O5         O  0.7452  0.7420  0.2602                x, y, z + (0,0,-1)   1.784
   21      O8         O  0.7557  0.8910  0.0024                x, y, z + (0,0,-1)   1.715
                                                                          1,0-1         Top
```

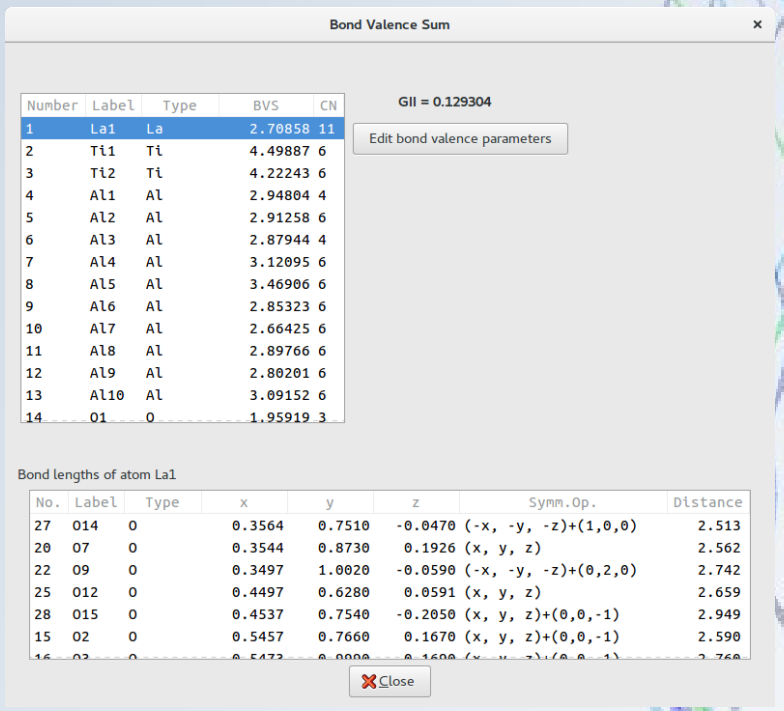

* Kasunič, M., Meden, A., S.D. Škapin, Suvorov, D. and Golobič, A. (2011). Acta Cryst. B67, 455–460

```
program example6
use iso_fortran_env, only: stdout => OUTPUT_UNIT, ERROR_UNIT
use crystal_phase
use gen_frm
use errormod
use symmgrid
implicit none
type(crystal_phase_t)                          :: cryst
type(error_type)                               :: err
integer                                        :: ia,ncoord
real, dimension(:), allocatable                :: bvs
type(bond_info_t), dimension(:), allocatable :: dinfo
!
call load_chemical_tables('../files/',err)
if (err%signal) then
    write(ERROR_UNIT,'(a)')' Message: '//err%msg(); stop 1
endif
!
call crystal_file_import(cryst,'LaTi_true.cif',err=err)
if (err%signal) stop 2
!
!  Bond valence parameters
   call cryst%load_bvparam()
   call cryst%print_bvparam(stdout)
!
! Initialization for bond valence
call cryst%make_symmetry_cell()
call cryst%allocate_grid(cutoff=maxval(cryst%bvpar(:,:)%rmax))
call cryst%fill_grid()
!
!  Bond valence sum and additional info
allocate(bvs(crys
do ia=1,cryst%nat
   call cryst%com
   write(stdout,'
   ncoord = cryst
   if (ncoord > 0
       call new_b
       call cryst
       call print
   endif
enddo
!
write(stdout,'(a,
end program examp
```

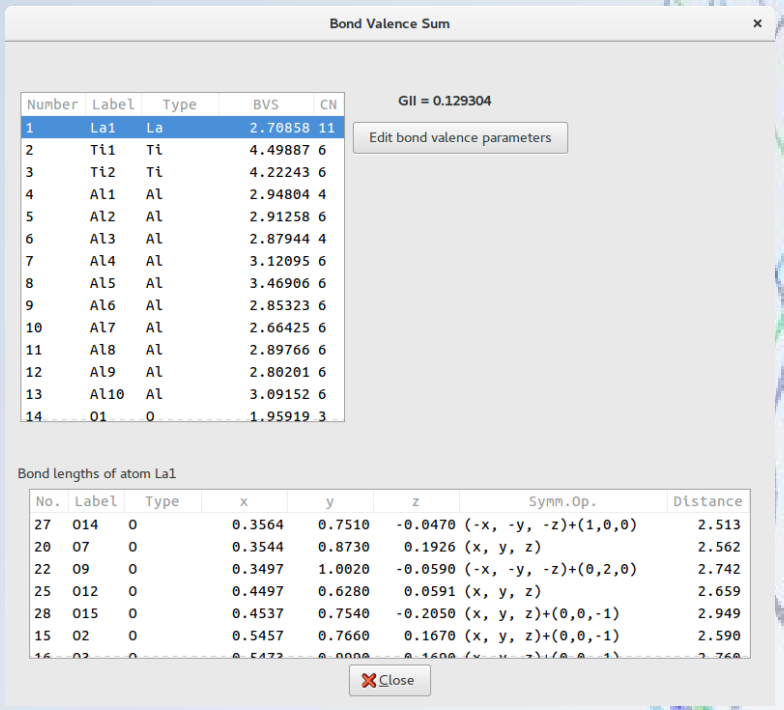
Bond Valence Parameters and bond windows

Atom1	Atom2	Ro	B	rmin	rmax
O2-	O2-	1.406	0.370	1.000	0.000
Al3+	O2-	1.651	0.370	1.318	2.270
Ti4+	O2-	1.815	0.370	1.704	2.304
La3+	O2-	2.086	0.450	2.100	3.000

```
   program example6
   use iso_fortran_env, only: stdout => OUTPUT_UNIT, ERROR_UNIT
   use crystal_phase
   use gen_frm
   use errormod
   use symmgrid
   implicit none
   type(crystal_phase_t)                          :: cryst
   type(error_type)                               :: err
   integer                                        :: ia,ncoord
   real, dimension(:), allocatable                :: bvs
   type(bond_info_t), dimension(:), allocatable :: dinfo
!
   call load_chemical_tables('../files/',err)
   if (err%signal) then
       write(ERROR_UNIT,'(a)')' Message: '//err%msg(); stop 1
   endif
!
   call crystal_file_import(cryst,'LaTi_true.cif',err=err)
   if (err%signal) stop 2
!
!  Bond valence parameters
   call cryst%load_bvparam()
   call cryst%print_bvparam(stdout)
!
!  Initialization for bond valence
   call cryst%make_symmetry_cell()
   call cryst%allocate_grid(cutoff=maxval(cryst%bvpar(:,:)%rmax))
   call cryst%fill_grid()
!
!  Bond valence sum and additional info
   allocate(bvs(cryst%natoms()))
   do ia=1,cryst%natoms()
      call cryst%compute_bvs(ia,bvs(ia))
      write(stdout,'(/2x,a,a8,a,f10.3)') 'Atom: ',cryst%at(ia)%glab(),' BVS:',bvs(ia)
      ncoord = cryst%coord_number(ia)
      if (ncoord > 0) then
          call new_bond_info(dinfo,ncoord)
          call cryst%valence_atom_info(ia,dinfo,bvs(ia),ncoord)
          call print_bond_info(cryst%at,cryst%spg,dinfo,stdout)
      endif
   enddo
!
   write(stdout,'(a,f10.3)')'GII=',cryst%gii_index(bvs)
   end program example6
```



```
program example6
use iso_fortran_env, only: stdout => OUTPUT_UNIT, ERROR_UNIT
use crystal_phase
use gen_frm
use errormod
use symmgrid
implicit none
type(crystal_phase_t)                          :: cryst
type(error_type)                               :: err
integer                                        :: ia,ncoord
real, dimension(:), allocatable                :: bvs
type(bond_info_t), dimension(:), allocatable :: dinfo
!
call load_chemical_tables('../files/',err)
if (err%signal) then
    write(ERROR_UNIT,'(a)')' Message: '//err%msg(); stop 1
endif
!
call crystal_file_import(cryst,'LaTi_true.cif',err=err)
if (err%signal) stop 2
!
! Bond valence parameters
call cryst%load_bvparam()
call cryst%print_bvparam(stdout)
!
! Initialization for bond valence
call cryst%make_symmetry_cell()
call cryst%allocate_grid(cutoff=maxval(cryst%bvpar(:,:)%rmax))
call cryst%fill_grid()
!
! Bond valence sum and additional info
allocate(bvs(cryst%natoms()))
do ia=1,cryst%natoms()
   call cryst%compute_bvs(ia,bvs(ia))
   write(stdout,'(/2x,a,a8,a,f10.3)') 'Atom: ',cryst%at(ia)%glab(),' BVS:',bvs(ia)
   ncoord = cryst%coord_number(ia)
   if (ncoord > 0) then
       call new_bond_info(dinfo,ncoord)
       call cryst%valence_atom_info(ia,dinfo,bvs(ia),ncoord)
       call print_bond_info(cryst%at,cryst%spg,dinfo,stdout)
   endif
enddo
!
write(stdout,'(a,f10.3)')'GII=',cryst%gii_index(bvs)
end program example6
```



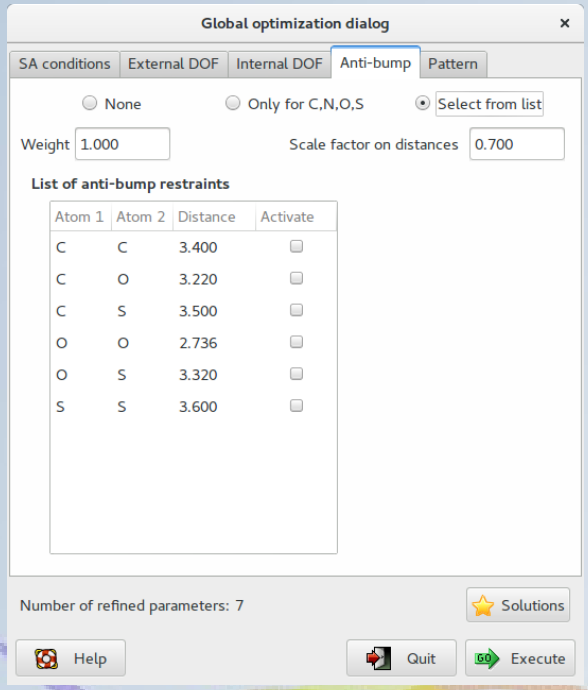
Imposing anti-bumping restraints

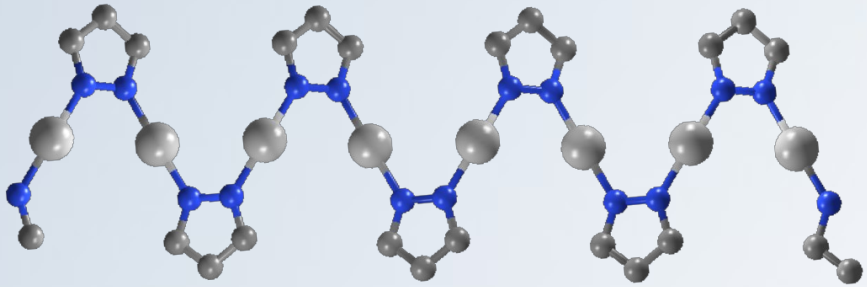
$$CF_{bump} = \sum_{ij}^{n} w_{ij}(d_{ij}^{min} - d_{ij}^{model})^{2k}$$

$$k = 2$$

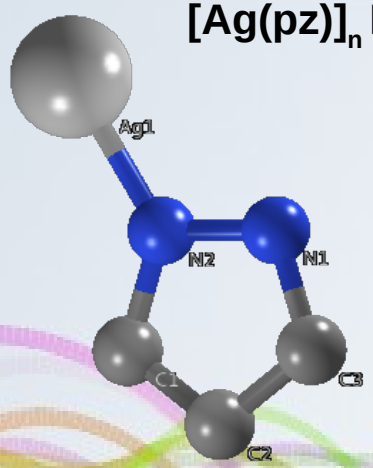
$$d_{ij}^{model} < d_{ij}^{min}$$

$$d_{ij}^{min} = \epsilon(R_i^{vdW} + R_j^{vdW})$$





$[Ag(pz)]_n$ Hpz = pyrazole




```
%sannel
bump * *
nobump Ag1 N1
nobump Ag1 N2
bscale 0.9
```

Feng, Z.J., Jia, R.R., Dong, C., Cao, S.X. & Zhang, J.C. (2010). J.Appl. Cryst. 43, 179—180

Molecular Geometry Restraints

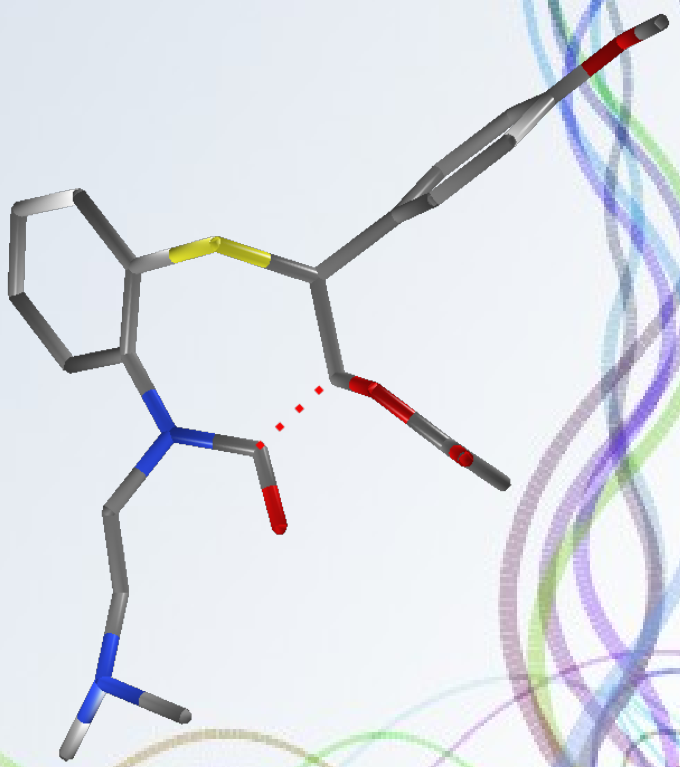
$$CF_{restraints} = \sum_i w_i MAX(0.0, |d_{target_i} - d_{AB_i}| - tol_i)^2$$

d_{AB_i} = distance between two atoms A and B

d_{target_i} = ideal distance

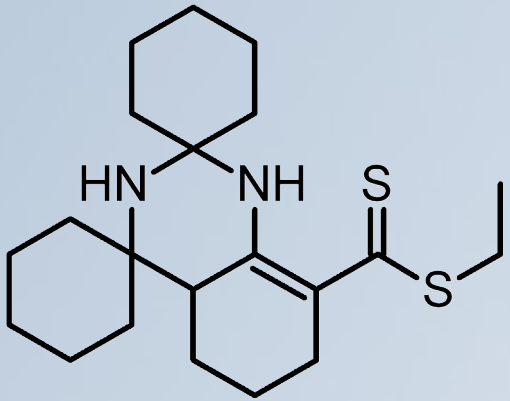
tol_i = permitted tolerance

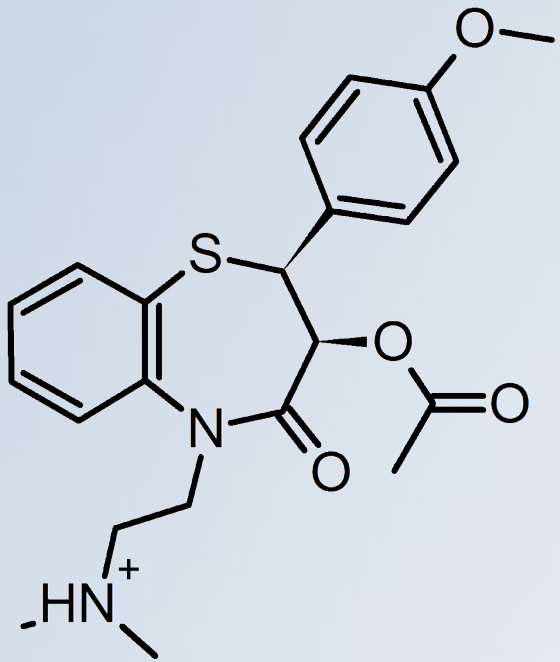
w_i = user supplied weight

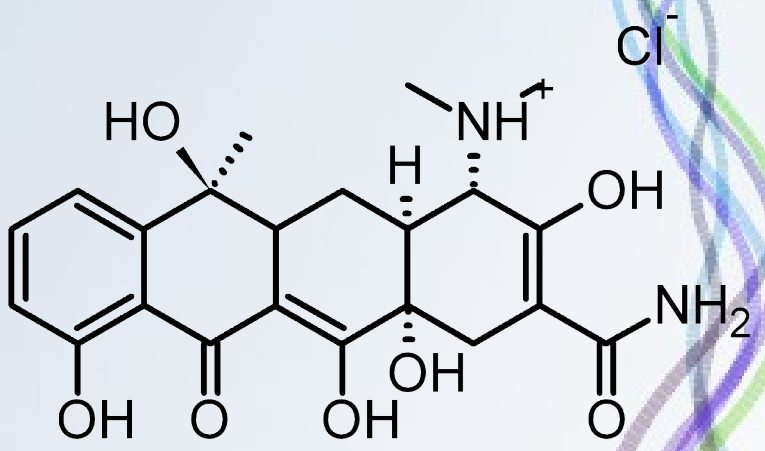


Non planar ring systems

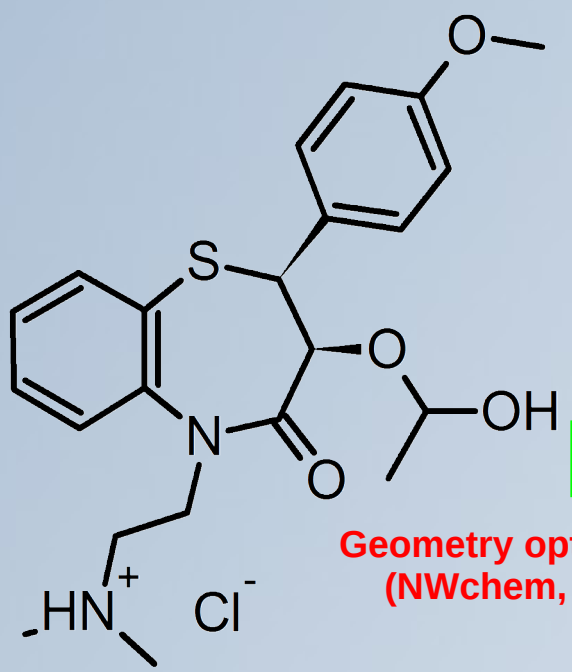
Attention to non planar ring systems or unusual combinations of elements in functional groups



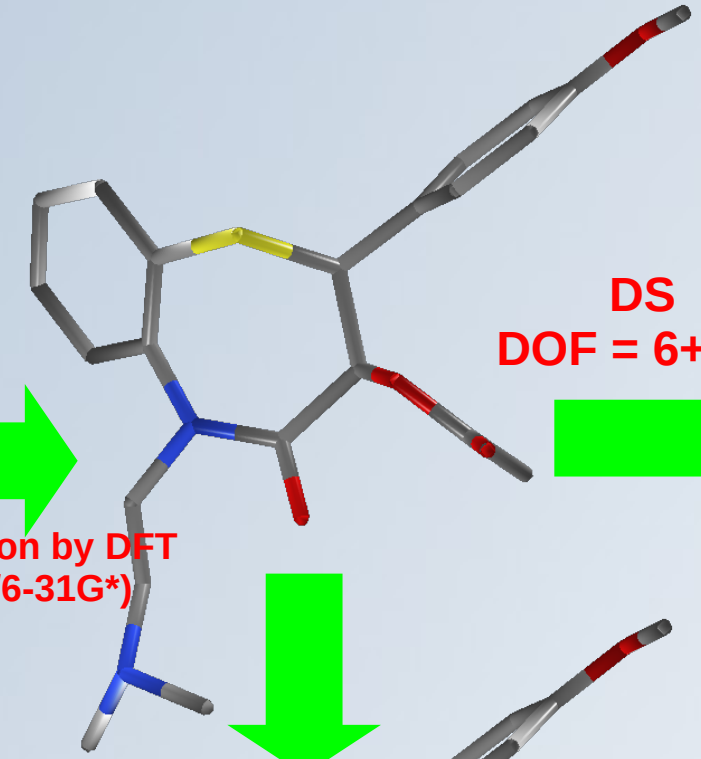




Structure Solution of Diltiazem Hydrochloride

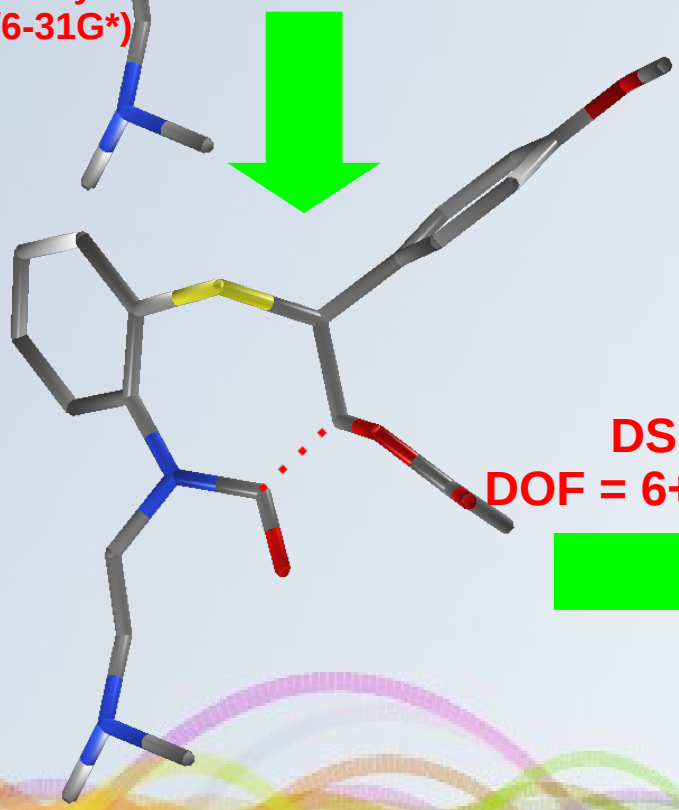


Geometry optimization by DFT
(NWchem, B3LYP/6-31G*)

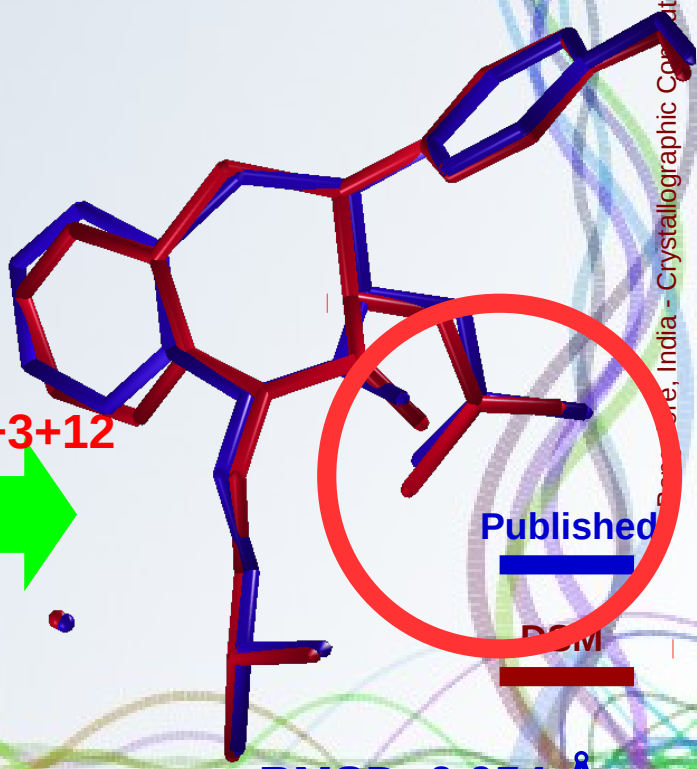


DS
DOF = 6+3+7

NO SOLUTION!



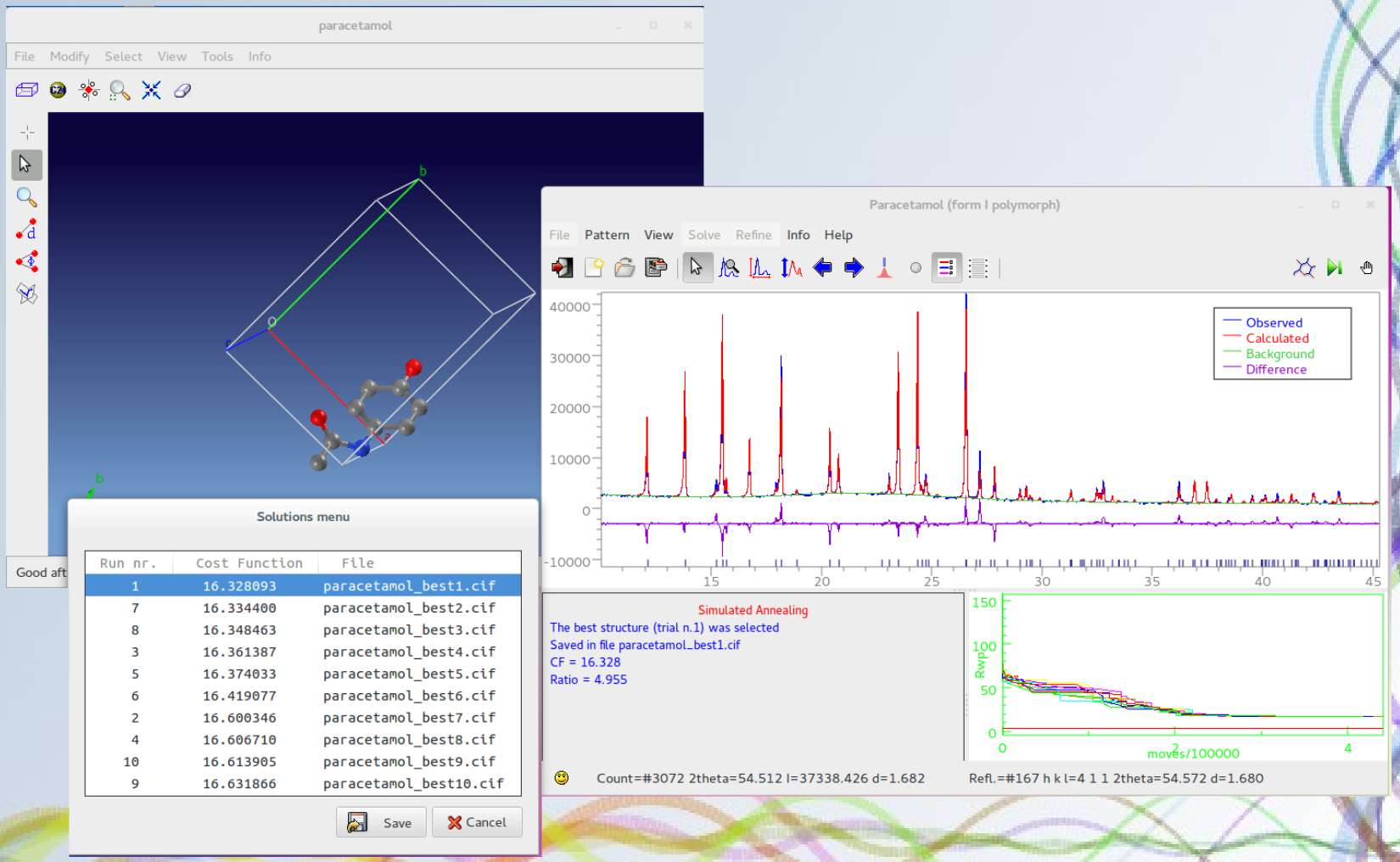
DS
DOF = 6+3+12



RMSD=0.054 Å

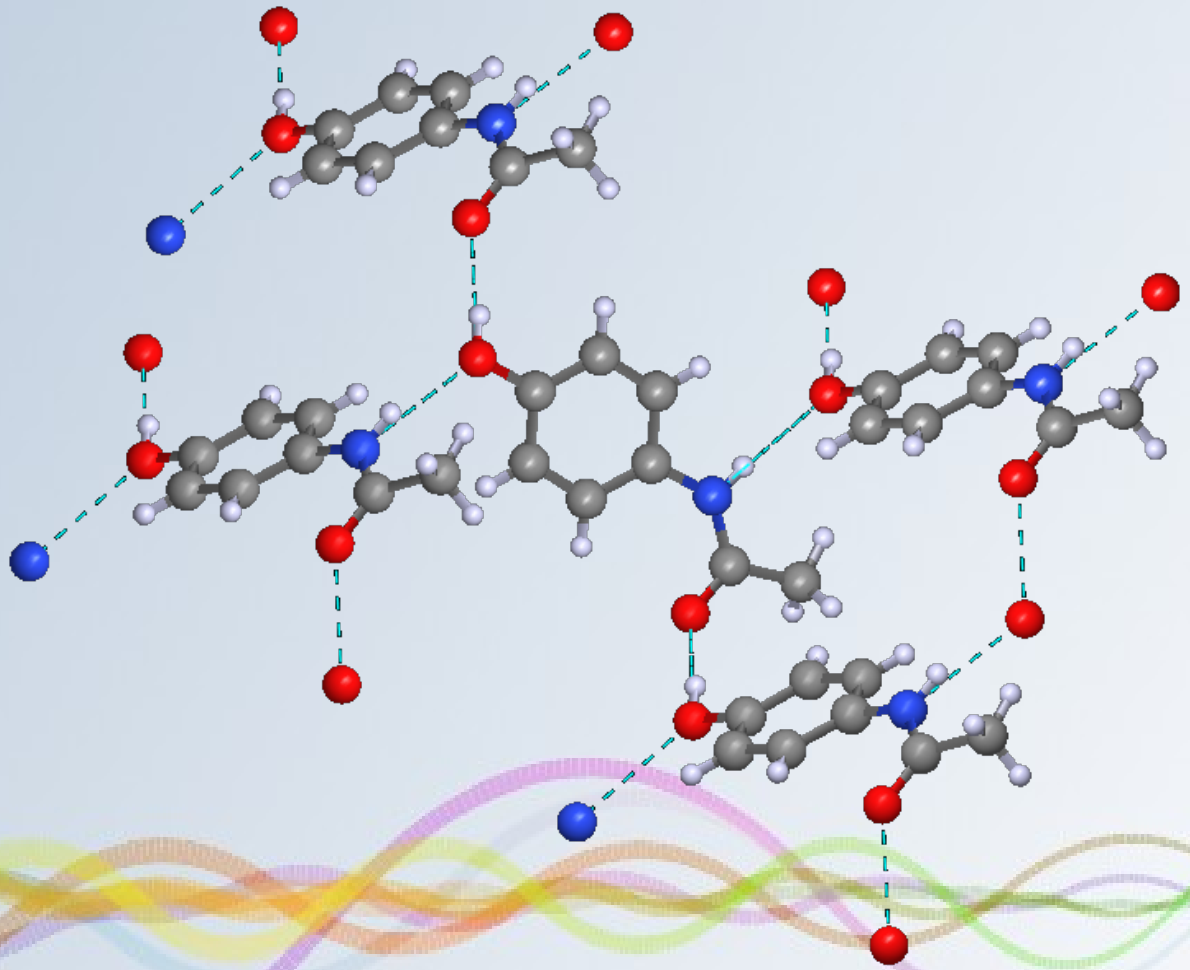
Assessing the solution

- Agreements factors
- Visual match between calculated and observed profile
- Reproducibility of solution



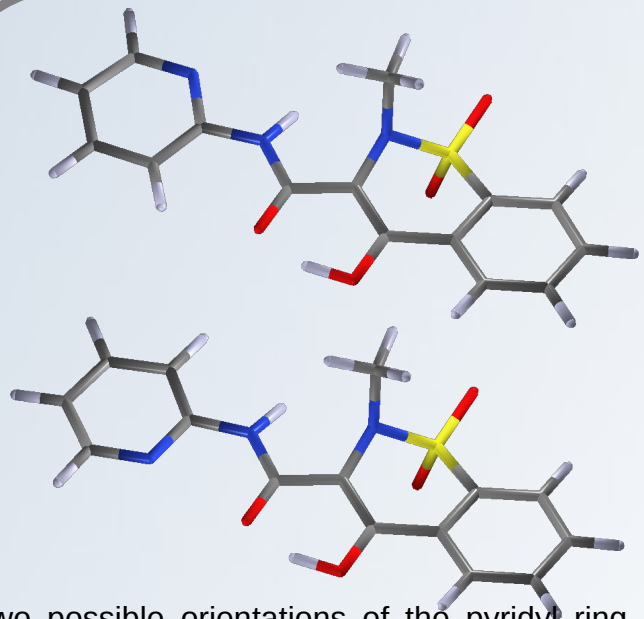
Assessing the solution

- Crystal packing
- Check close contacts, void spaces, likely iterations
- Network of interactions: hydrogen bonds and short contacts



Combined powder X-ray diffraction data and quantum-chemical calculations

- **Optimization of the molecular geometry** to obtain accurate starting models
- **Restraints** in the Rietveld refinement
- **H atoms**
- Solve **ambiguities**
  (*e.g.*, space groups, torsion angles)
- **Refinement** of crystal structure
- **Validation** of experimental crystal structures



Two possible orientations of the pyridyl ring in the piroxicam molecule (Naelapää, K., van de Streek, J., Rantanen, J., and Bond, A. D. (2012), J.Pharm. Sci. 101, 4214–4219)

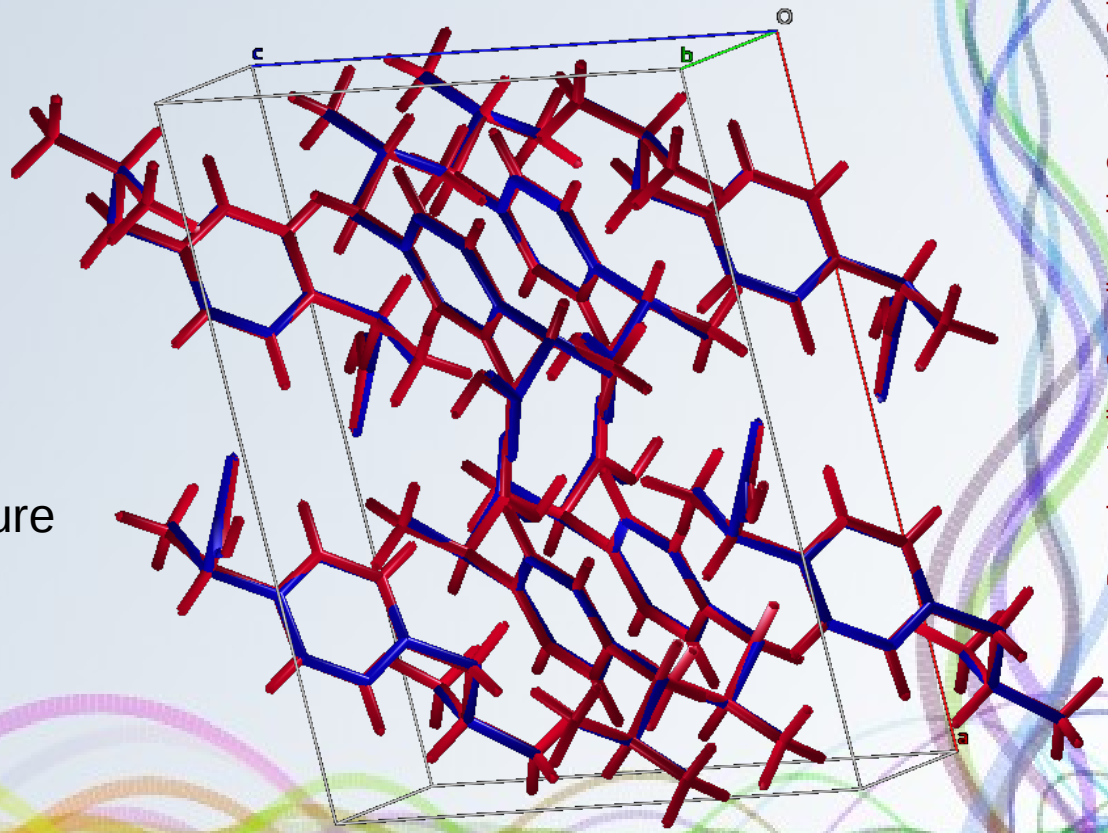
Assessing the solution with DFT-D

Theoretical approach: plane wave (PW) density functional theory with dispersion correction (DFT-D)

RMSD for non H-atoms above 0.25 Å could indicate incorrect experimental crystal structure *

Ibuprofen

RMSD=0.023 Å



Experimental crystal structure

DFT-D3 with NWChem

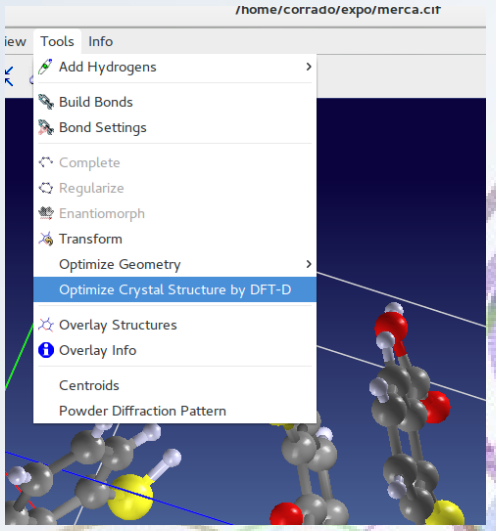
*Jacco ven de Streek *et al.* Validation of molecular crystal structures *Acta Cryst.* (2010). B**66**, 544–558

DFT-D: Howto

Software	Academic price (€)	Link
VASP	4,000	www.vasp.at
CASTEP	1,800	www.castep.org
CRYSTAL	1,000	www.crystal.unito.it
Quantum ESPRESSO	free	www.quantum-espresso.org
NWChem	free	www.nwchem-sw.org
Abinit	free	www.abinit.org

Hardware: multi-core Linux Workstation

Time: approx. 100 hrs for small molecules on single CPU



- **Quantum espresso**

```
subroutine write_qe_file(filename,atom,cell,spg,structname)
subroutine read_qe_file(filename,atom,cell,spg,errc)
```

- **CRYSTAL**

```
subroutine write_crystal_file(filename,atom,cell,spg,structname)
subroutine read_crystal_file(filename,atom,cell,spg,errc)
```

- **NWCHEM**

```
subroutine write_nw_file(filename,atom,cell,spg,structname,nwtheory)
subroutine OBConversionReadFile(filename,atom,bond,err)
```

- **Abinit**

```
subroutine write_ab_file(filename,atom,cell,spg)
subroutine OBConversionReadFile(filename,atom,bond,err)
```

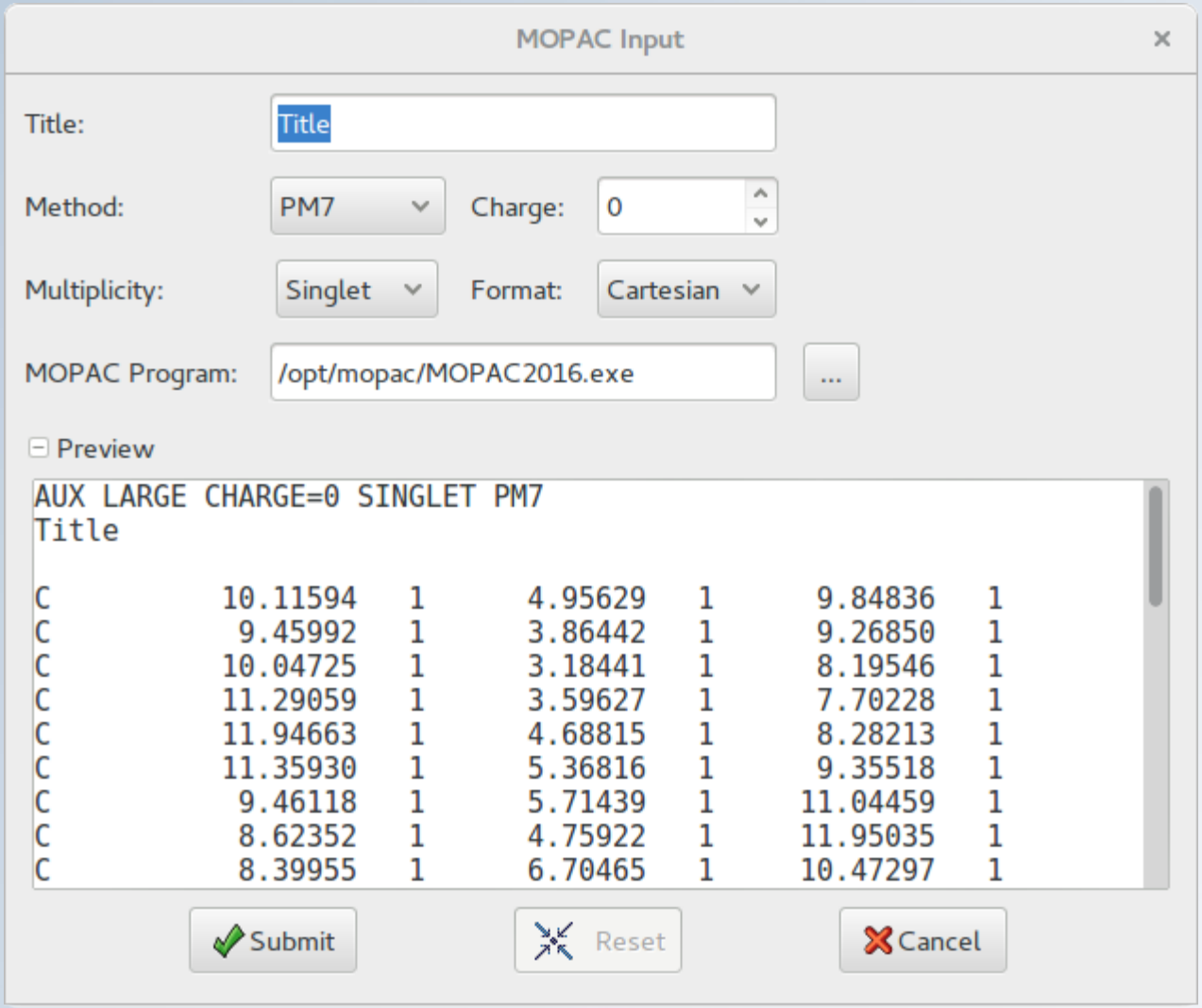
- **Gaussian, MOPAC**

```
subroutine OBConversionWriteFile(filename,atom,bond,cell,spg)
subroutine OBConversionReadFile(filename,atom,bond,err)
```

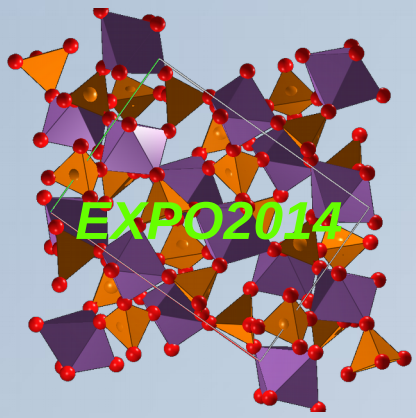
General procedures:

```
subroutine crystal_file_import(crystal,filename,[ftype],[radtype],[has_symmetry],err)
subroutine crystal_file_export(crystal,filename,[ftype],[radtype],[wave],[comm])
```

Graphical User Interface for MOPAC2016



The Use of Open Babel



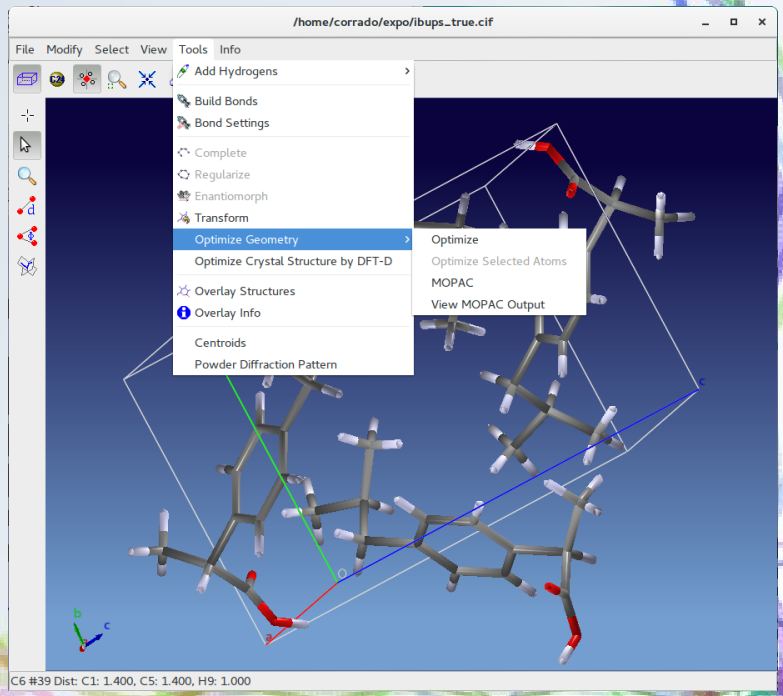
Fotran/C++

obmodule.f90

C++

- Molecular-mechanics force fields (MMFF99 and UFF provided by **Open Babel library)**

- Able to process input and output files of the most common quantum-chemistry packages



Download this lecture

http://www.ba.ic.cnr.it/softwareic/expo/tutorials-and-lectures/

Contact, software download and info

http://www.ba.ic.cnr.it/softwareic/expo/

Acknowledgements

Colleagues of the research team

A. Altomare, A. Moliterni, R. Rizzi, N. Corriero and A. Falcicchio

Other collaborators

G. Cascarano, R. Mallamo, F. Ciriaco