MRC Laboratory of Molecular Biology

ACEDRG

Restraint generator for monomers/ligands

Garib N Murshudov
MRC-LMB, Cambridge

Contents

- Introduction
- AceDRG: two functions
- Validation of entries in the DB and derived data
- Generation of new ligand description
- Covalent link description
- Identification and dealing with metals
- Conclusions

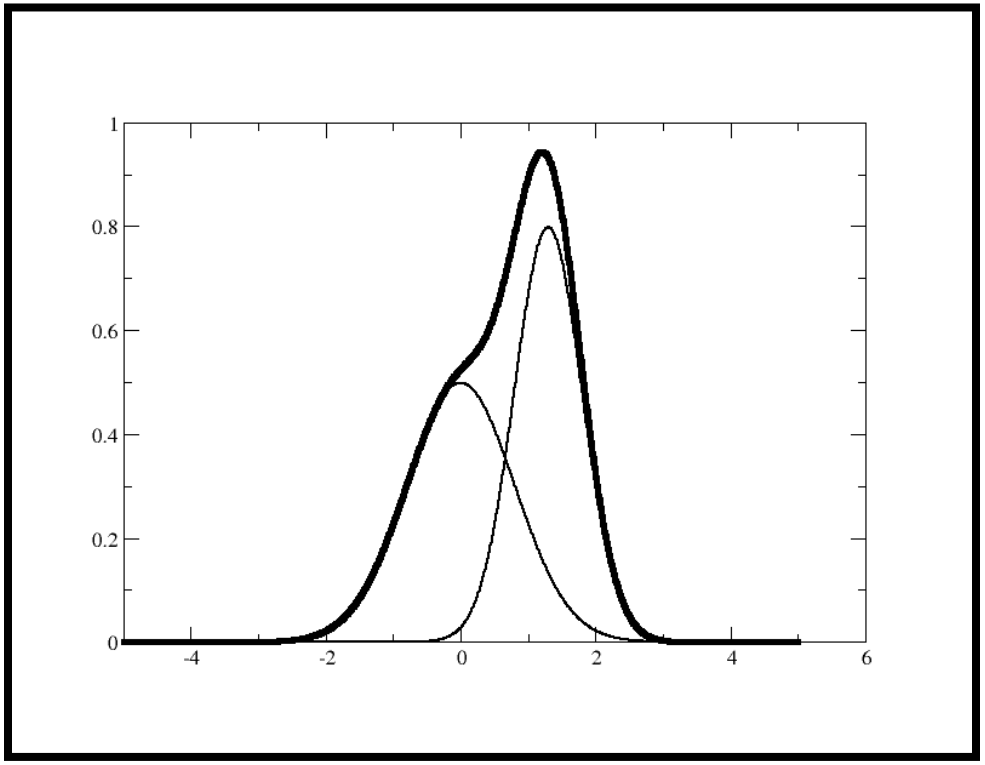
Introduction

- Information about internal structure of molecules is used to complement experimental data
- Dictionary is used for model building and fitting into the data
- Dictionary is used to position missing atoms (e.g. hydrogens)
- Properties of atoms and functional groups are used for non-bonding interactions
- Dictionary of monomers can also be used for MM calculations

Why Restraints?

Example: two-atom ideal case

Distance between atoms is 1.3Å. B-factors are 20 and 50



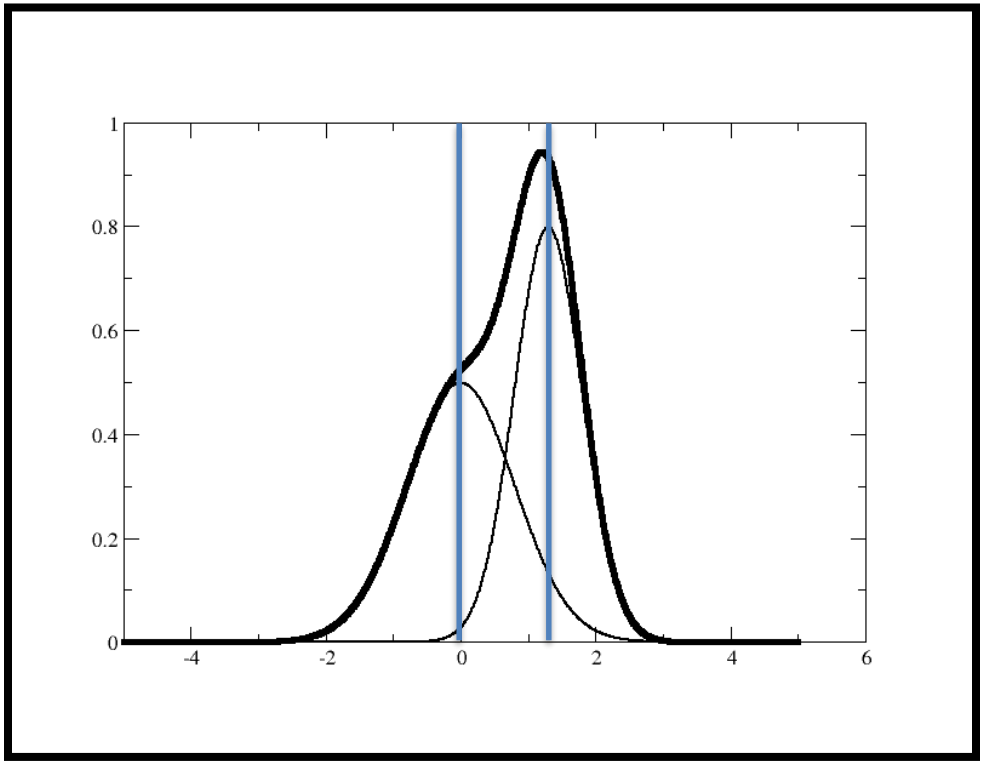
Thin lines – single atoms

Bold line - sum of the two atoms

Why Restraints?

Example: two-atom ideal case

Distance between atoms is 1.3Å. B-factors are 20 and 50



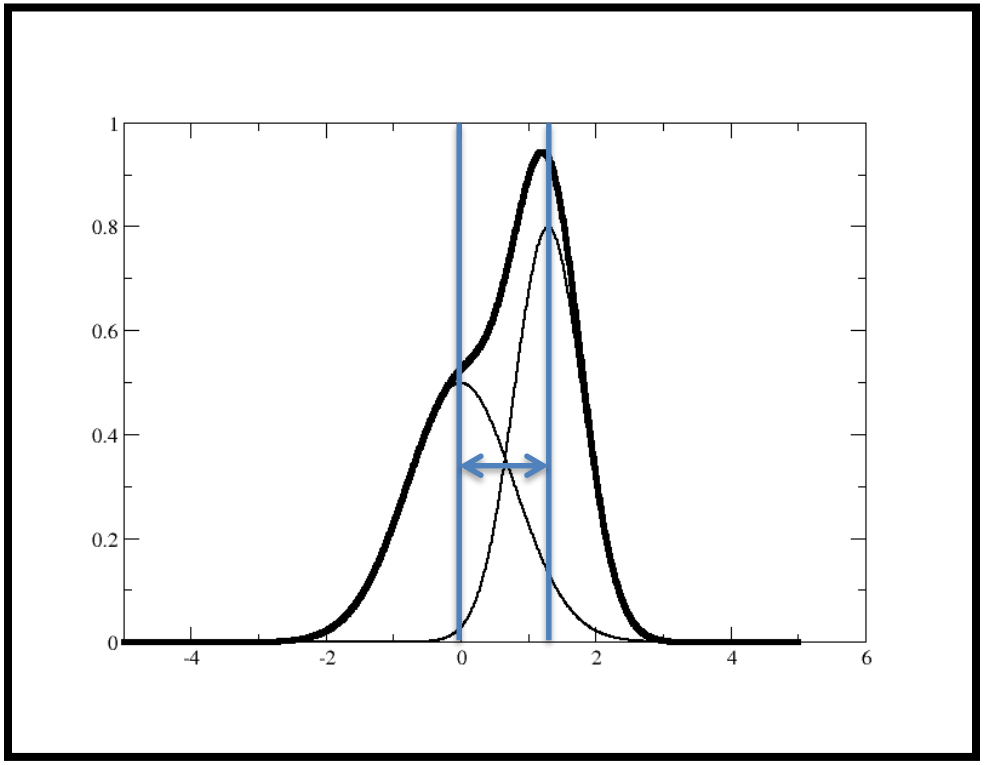
Thin lines – single atoms

Bold line - sum of the two atoms

Why Restraints?

Example: two-atom ideal case

Distance between atoms is 1.3Å. B-factors are 20 and 50



Thin lines – single atoms

Bold line - sum of the two atoms

Restraints

Standard restraints (used by default) include:

- Bond lengths
- Angles
- Chirals
- Planes
- Some torsion angles
- B-values
- VDW repulsions

These help to ensure that the model is chemically sensible

Note – we generally deal with restraints, not constraints

Introduction

Two ways of generating chemistry of molecules:

1) Tabulated atom types and mapping of these atom types to current molecule

   Problem: The number of atom types can be very large (potentially infinite)

1) Tabulated "ideal" values for building blocks of macromolecules
   1) Monomers (e.g. AA. NA, sugars) and ligands
   2) Links between monomers (peptide links, sugar links, NA links)

   Problem: Not all building blocks have high resolution experimental structure

   QM may take long time if environment need to be accounted for

List of existing dictionary generators

GRADE	Global Phasing
ProDRG	Dundee group
Libcheck	CCP4
Phenix.elbow	Phenix group
Pyrogen	COOT (Paul Emsley)
AceDRG	**CCP4**

Grade, Phenix.elbow and pyrogen use MOGUL from CSD.

AceDRG at the moment uses COD as a source

Source of atom types and “ideal” values

There are two options:

1. High Level QM calculations
    1. Expensive
    2. It is hard to account for effects of environment
2. Small molecular crystallographic database
    1. Crystallography Open Database - COD
    2. Cambridge Structure Database - CSD

Note 1: CSD already offers very good resource – MOGUL. It is used by the programs GRADE, phenix and coot’s pyrogen.

Note 2: All small molecular databases have their own problems: data must be validated before use

AceDRG: two functions

We need to solve two problems

1. Derive atom types from small molecular database
2. Apply derived parameters to generate new ligand description

Both of these problems are dealt with AceDRG. It generates parameter tables and applies them to new ligands.

Atom types are dynamic: if database is extended or new database is used then AceDRG can be applied to generate new parameter tables

AceDRG applied to COD

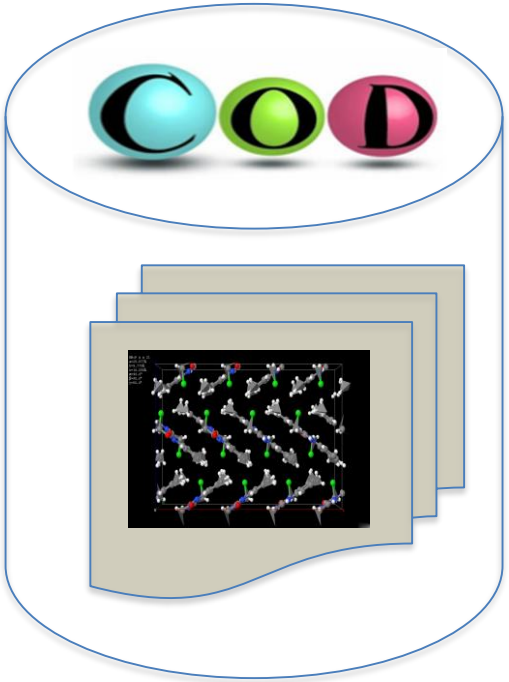
ACEDRG currently uses small molecule structures from Crystallography Open Database(COD) as a source for:

- ❑ Atom types that characterize atom's local environments.
- ❑ Derivation the values of bond lengths and angles

Data Source

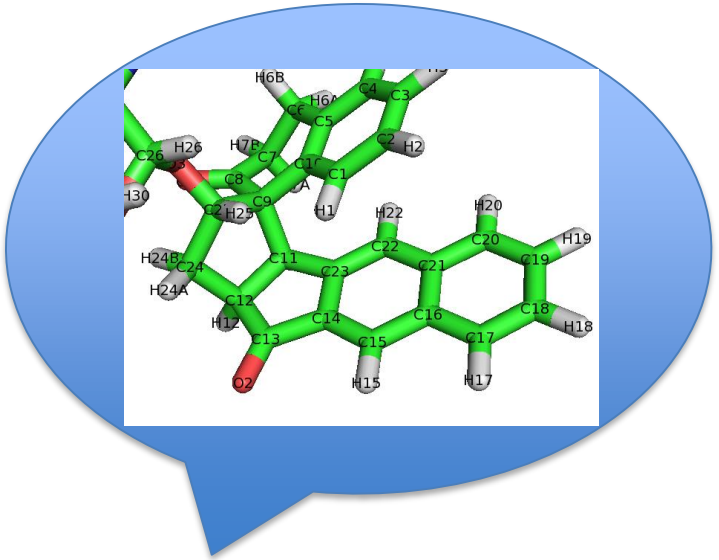
Brief summary of COD

- COD contains more than 300,000 crystal structures of organic, inorganic and metal-organic compounds. (for comparison, number of ligands in PDB : ~23000)
- All structures are published in peer-reviewed journals.
- It is freely accessible and updated daily.



Rules for atom types

1. Element name
2. The number of bonds
3. Belongness to rings (up to 6 member)
4. Aromaticity of these rings
5. Information about immediate neighbours
6. If atoms are in aromatic rings then some information about their third neighbours



Atom C23: C[5,6a](C[5,5]C[5,5]C[5,6]H)(C[5,6a]C[6a]C[5])(C[6a]C[6a,6a]H)
Atom C18: C[6a](C[6a]C[6a,6a]H)(C[6a]C[6a]H)(H)

Rules for hybradisation

sp^3:

1. C, N, B with four bonds
2. S with three or four bonds
3. O with two bonds and none of the bonded atoms is sp^2

sp^2:

1. C with three bonds
2. N or B with three bonds and one of the attached atoms is sp^2
3. O with two bonds and one of the attached atoms is sp^2
4. O with one bond

AceDRG applied to COD

Curating and Validating data from COD

- Total structures in COD: **307987**
- Rejected structures : **175923**
  - Resolution is low (>0.88A) or Rfactors are high (> 0.10) and no symmetry operator, occupancies ≠ 1 : **72067**
  - Rejected by basic chemical sanity check:
    - the number of bonds to an atom should be consistent with its valence
    - Difference between same type of bonds within molecule should not be very different: **103856**

AceDRG applied to COD

Curating and Validating data from COD

Extended organic Set
Includes: H, C,N, O, B,P, S, F Cl, I, Br, Se

- Structures with metals as first or second neighbour: **7286**

Number of COD structures used by AceDRG:	**124778**
40% of COD	

Validation & Quality Control

COD:

Crystallographic & chemical validation

ACEDRG:

Crystallographic & chemical validation
(use CIF files from COD)

Separate Validation Tool:

Systematic validation
Statistical analysis

Coot:

Final ligand validation

– checks consistency against reference
- can use multiple sources – acedrg (COD), pyrogen (Mogul/CSD)
- resource comparison

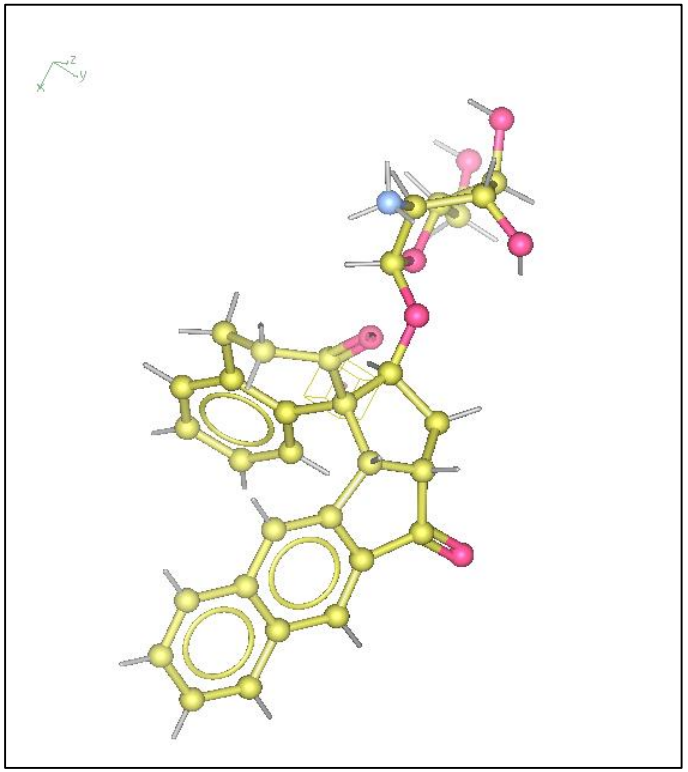
AceDRG as a dictionary generator

Monomer/Ligand/restraint generator

- Input files of SMILES, SDF/MDL, mmCIF formats
- Output Dictionary files of mmCIF format: contain atom types, bond lengths and angles, torsion angles, planes and chiral centers
- Output coordinate files of PDB format, which represent one of the conformations.

Using Acedrg

- RDKit is used to accept input files of different formats: SMILES, MDL/SDF mol, SYBL/MOL2. MMCIF files are converted to mol files first.



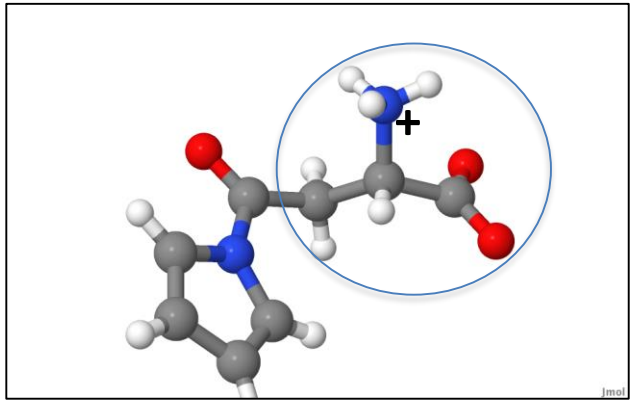
N[C@@H]1[C@@H](O)[C@H](O)[C@@H(CO)O[C@H]1O[C@H]2C[C@H]3[C@H](c4cc5ccccc5cc4C3=O)[C@@]26C(=O)CCc7ccccc67

Functional groups in AceDRG

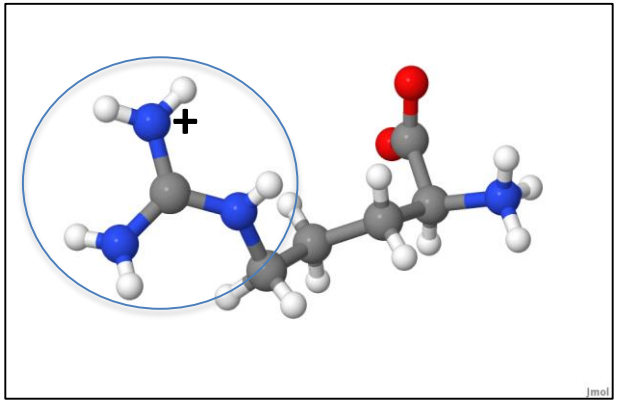
There are 21 functional groups with their assosiated pKa values. Functional groups are used to make decision about protonation state of a ligand. Current assumption is that pH is 7.0. But it can be changed by a user or software running acedrg.

The list of functional groups can be extended.

Functional Groups in Acedrg: example



Substructure in ASM
pKa : 2.0 PH : 7.0

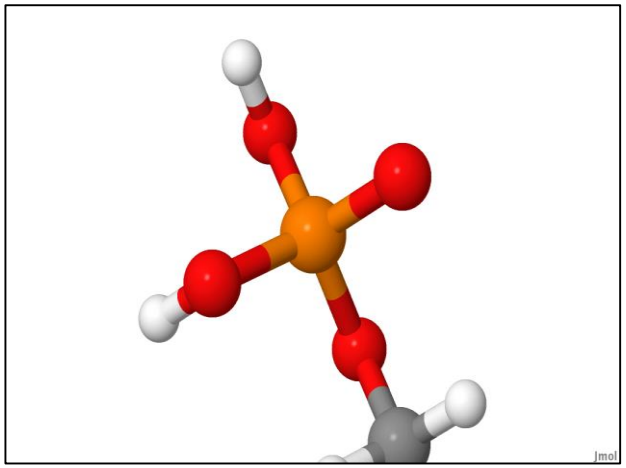


Substructure in ARG
pKa : 12.5 PH : 7.0

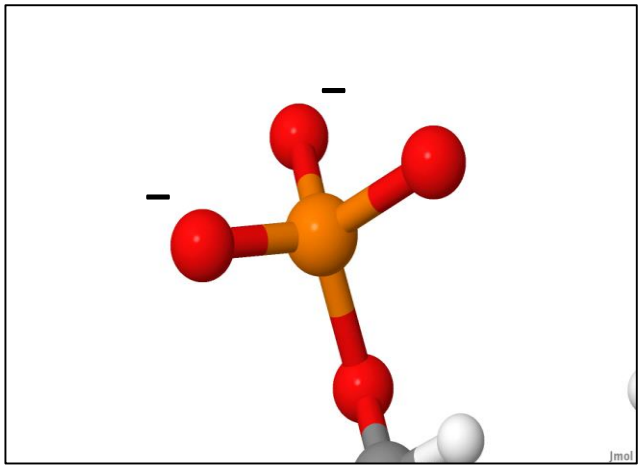
The default pH is 7.0

Functional Groups in Acedrg: example

Phosphate group PO4 in e.g. PYO



In PDB: in vacuum?



In Acedrg, PH=7.0

Use of Acedrg

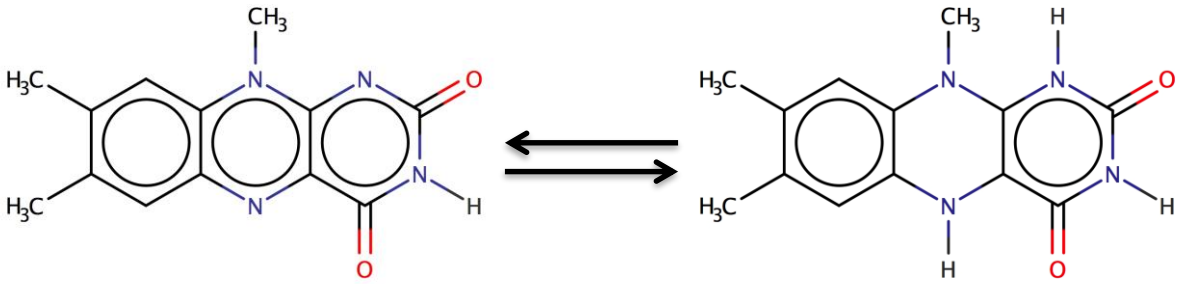
Ligand/restraint generator

- Input files of SMILES, SDF/MDL, mmCIF, SYBYL/Mol2 formats
- Output monomer/ligand files of mmCIF format, which contain atom types, bond lengths and angles, torsion angles, planes and chirality centers, and are used as restraint files for refinements.
- Output coordinate files of PDB format, which represent one of lower energy conformations for the ligands under consideration.

An example: Effect of aromaticity definition

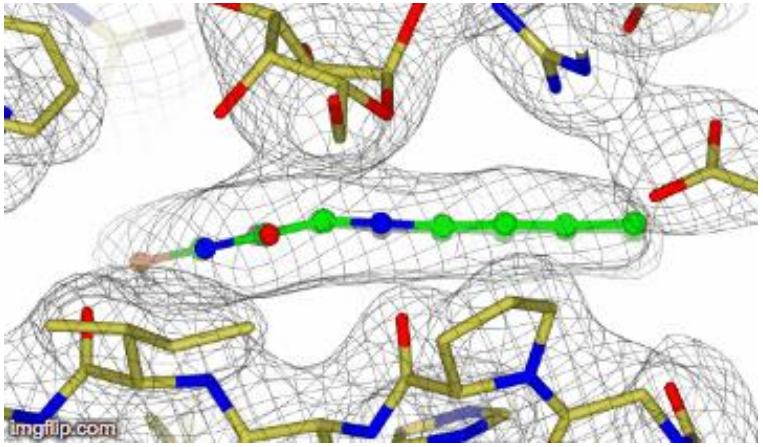
AceDRG

- An example of $FADH_2$ with two different aromaticity definition



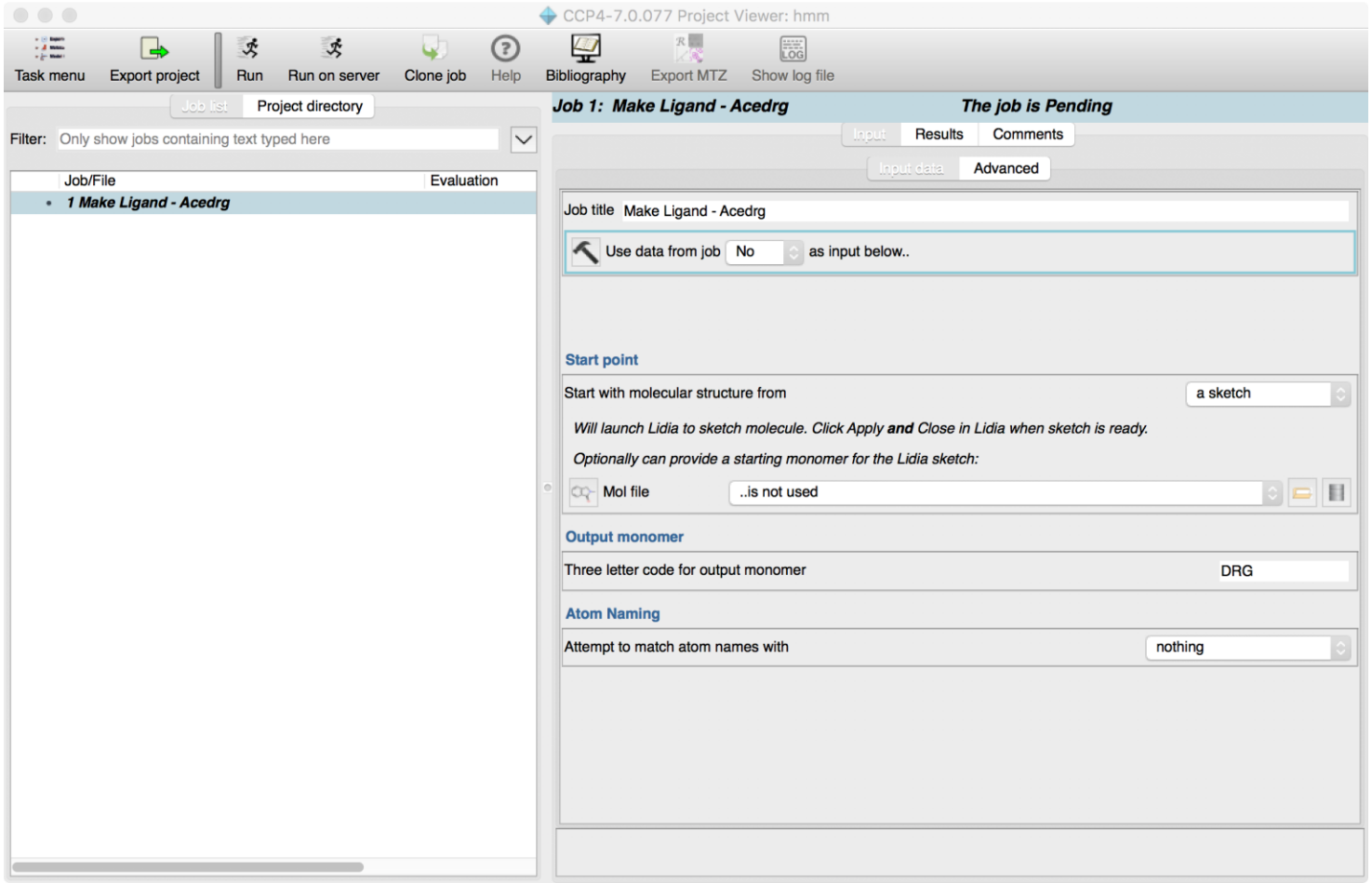
FAD $\quad\quad$ $FADH_2$

PDB 3HDY
$FADH_2$ before and after refinement
with AceDRG-generated restraints



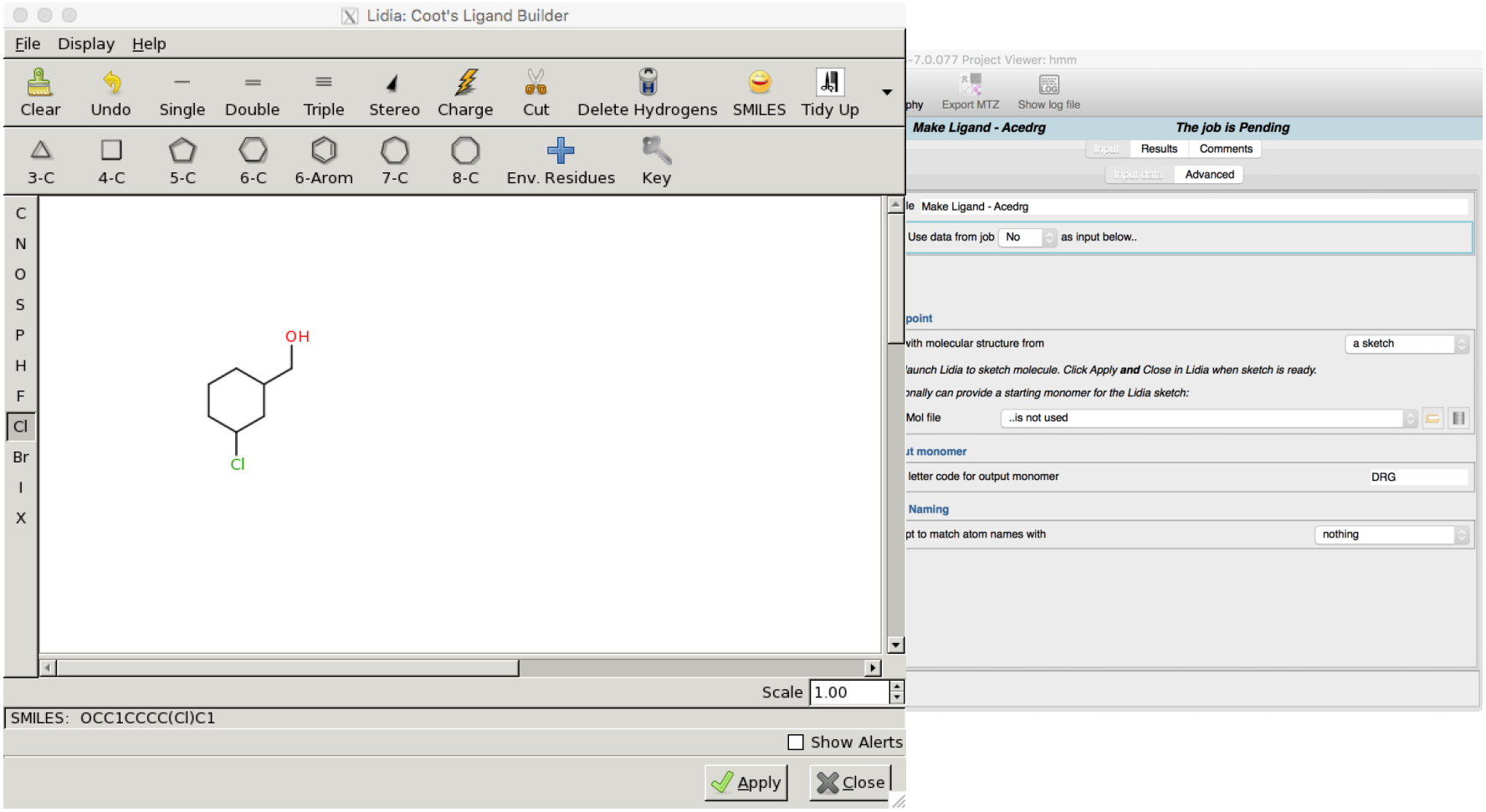
Using Acedrg in ccp4i2

In CCP4i2 interface



Using Acedrg in ccp4i2

In CCP4i2 interface



Covalent Links in AceDRG

Description how to use AceDRG can be found here:

https://www2.mrc-lmb.cam.ac.uk/groups/murshudov/content/acedrg/acedrg.html

Example: Create a file containing something like this:

LINK: RES-NAME-1 2OP FILE-1 2OP_acedrg.cif ATOM-NAME-1 C RES-NAME-2 VAL ATOM-NAME-2 N

Write to a file. For example – myInsts.txt

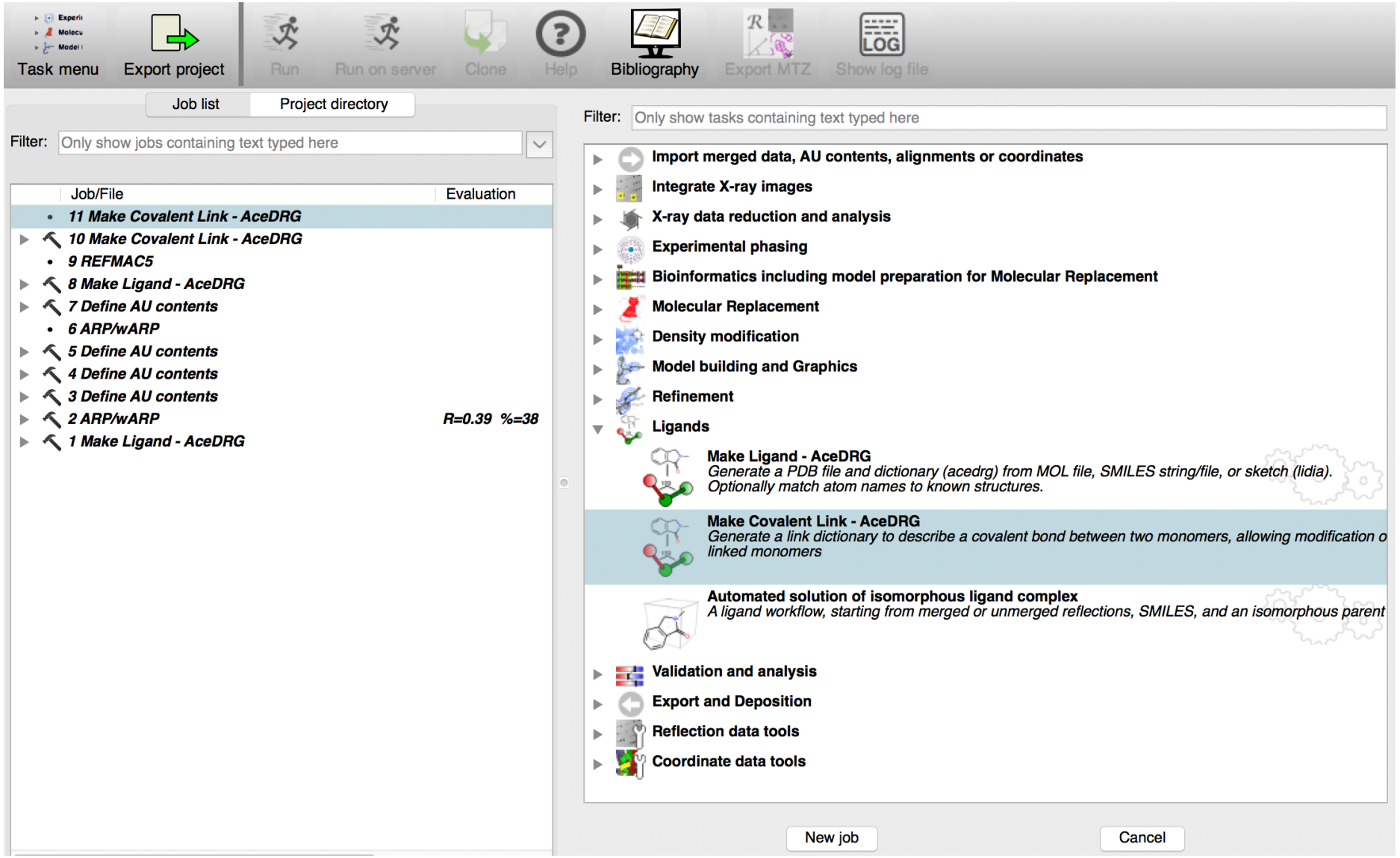
acedrg -L myInsts.txt

It will create files corresponding covalent links.

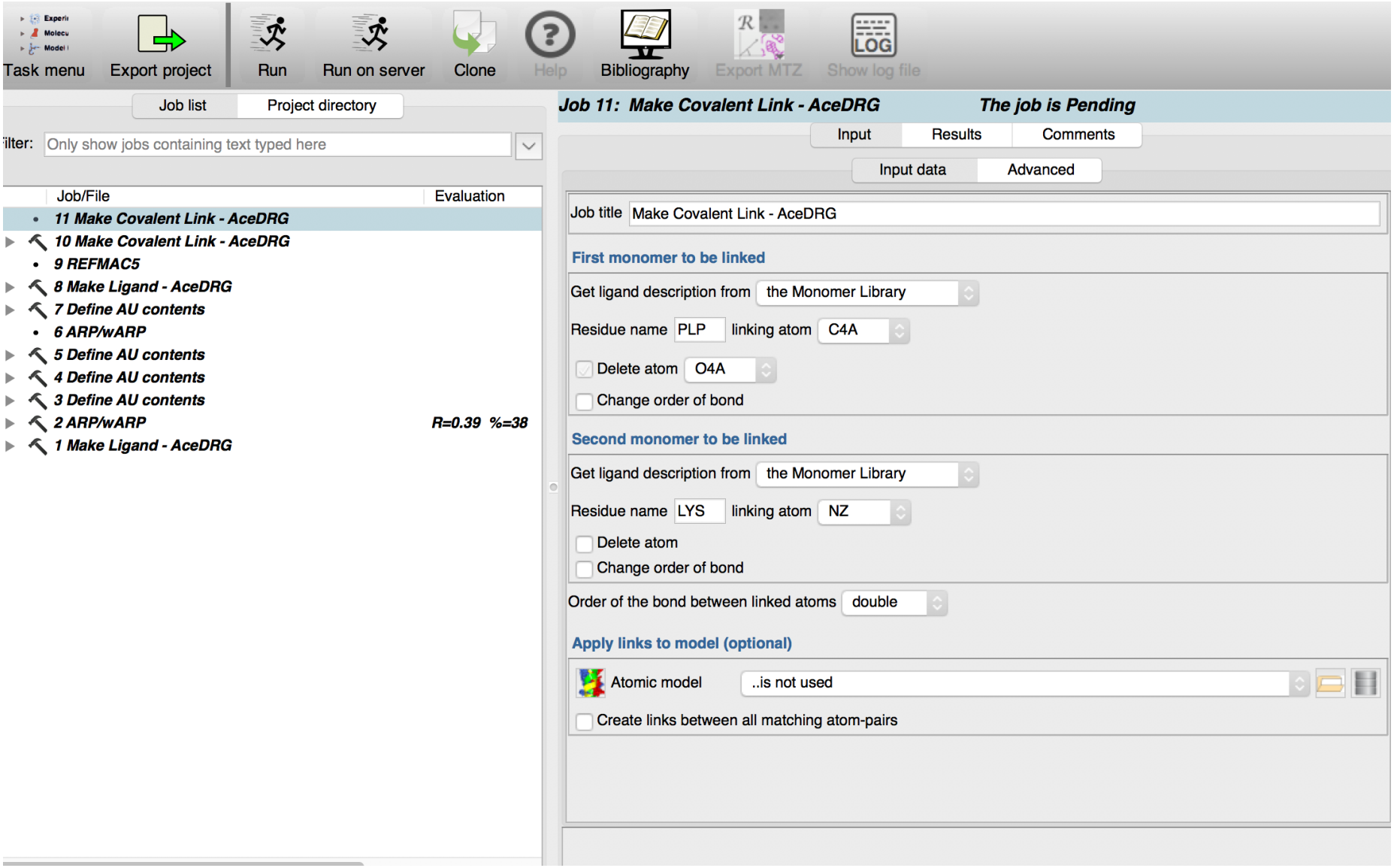
This option is not yet available on the interface. It will be available soon.

Covalent links using ccp4i2

From task menu select “Make Covalent Link – AceDRG”



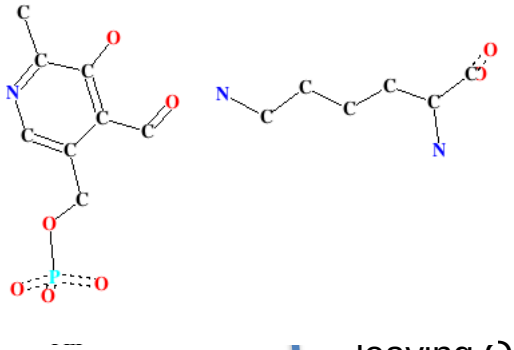
Covalent links using ccp4i2



Creating a new link, as seen in JLigand GUI

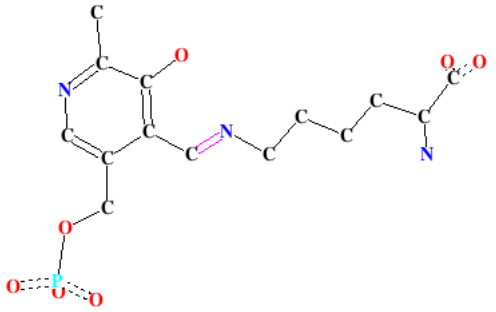
The two monomers are in effect reacted in silico
Hydrogen atoms are dealt with automatically*)

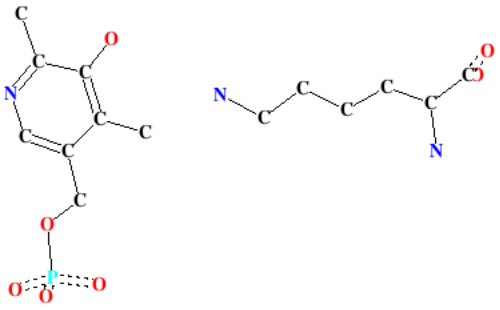
*) it is also possible to visualise H-atoms and deal with them explicitly



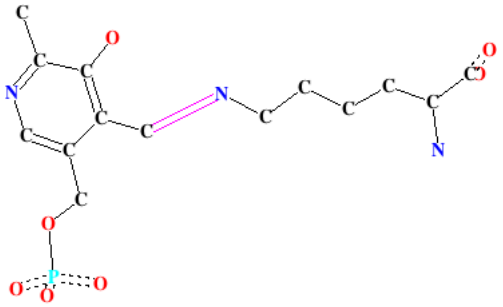
leaving O is removed

the new compound
is regularised





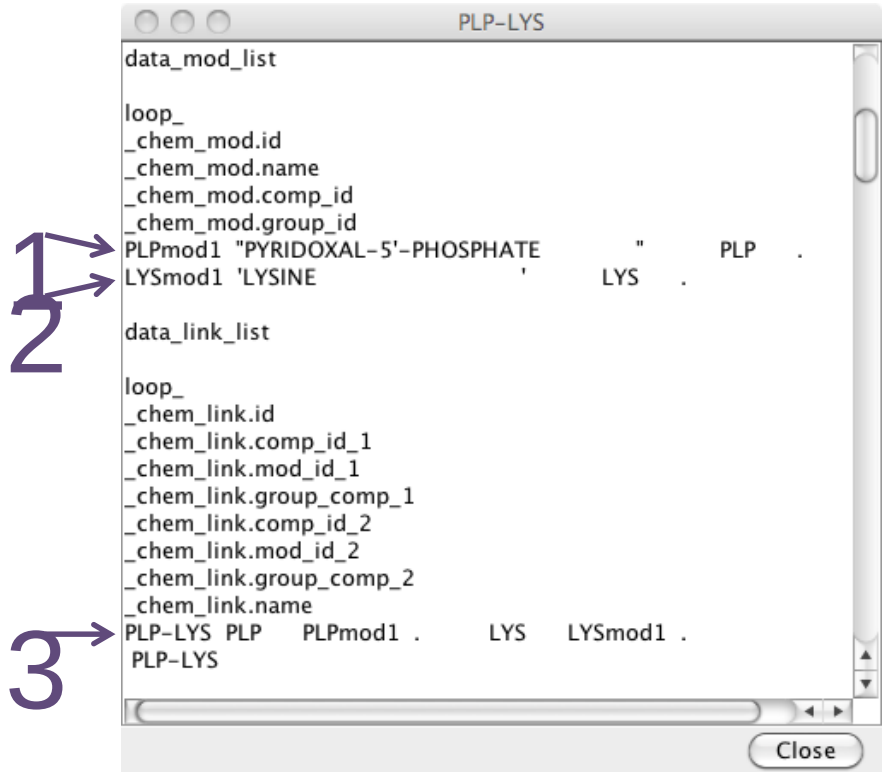
double bond C=N is added



Both monomers are modified and link is created

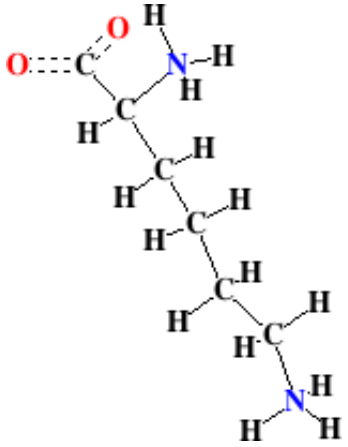
Contents:

(1) modification "PLPmod1"
(2) modification "LYSmod1"
(3) link "PLP-LYS"

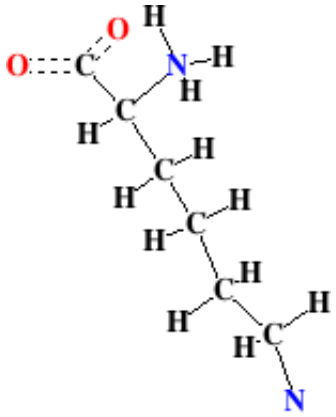


The new link, "file view"

LYS



LYSmod1



Modification "LYSmod1": changes to LYS

```
data_mod_LYSmod1

loop_
_chem_mod_atom.mod_id
_chem_mod_atom.function
_chem_mod_atom.atom_id
_chem_mod_atom.new_atom_id
_chem_mod_atom.new_type_symbol
_chem_mod_atom.new_type_energy
_chem_mod_atom.new_partial_charge
 LYSmod1  change NZ    .    .  N      0.000
 LYSmod1  delete HZ1   .    .  .       .
 LYSmod1  delete HZ2   .    .  .       .
 LYSmod1  delete HZ3   .    .  .       .
```

Atoms

```
loop_
_chem_mod_bond.mod_id
_chem_mod_bond.function
_chem_mod_bond.atom_id_1
_chem_mod_bond.atom_id_2
_chem_mod_bond.new_type
_chem_mod_bond.new_value_dist
_chem_mod_bond.new_value_dist_esd
 LYSmod1  change CE   NZ     .      1.455   0.020
 LYSmod1  delete NZ   HZ3    .        .      .
 LYSmod1  delete NZ   HZ2    .        .      .
 LYSmod1  delete NZ   HZ1    .        .      .
```

Bonds

.....

Angles

Utilising new link description

Three remaining steps:

– docking monomer(s) into electron density

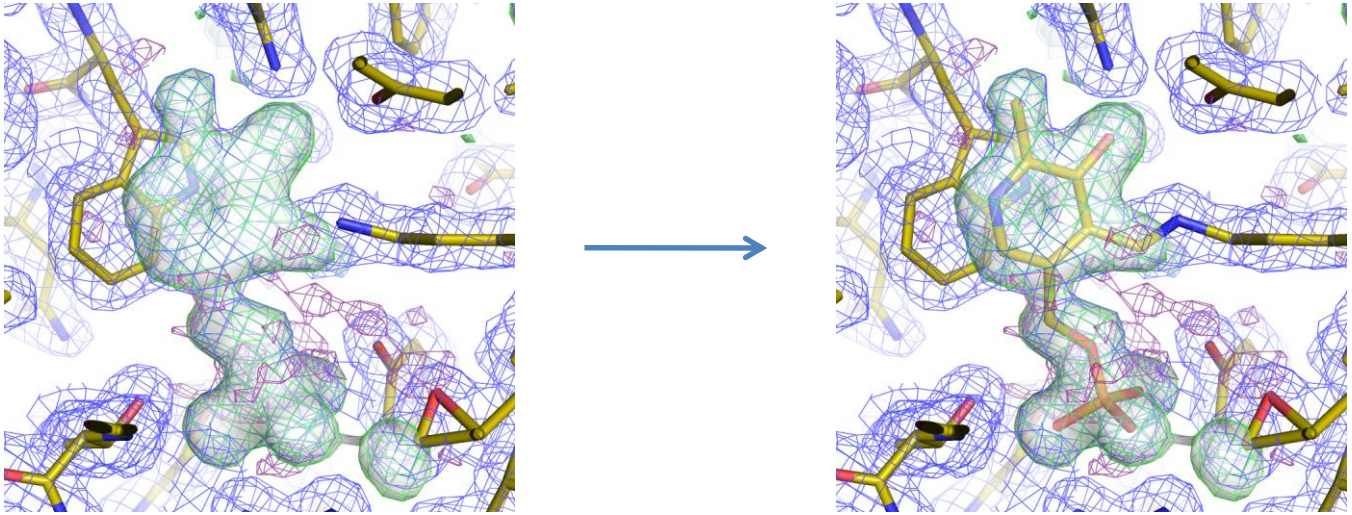
– defining link in the pdb-file

– refinement of the structure with linked ligand using additional library

(1) Docking into the electron density

In our example, this is completely independent step: the additional library is not used.

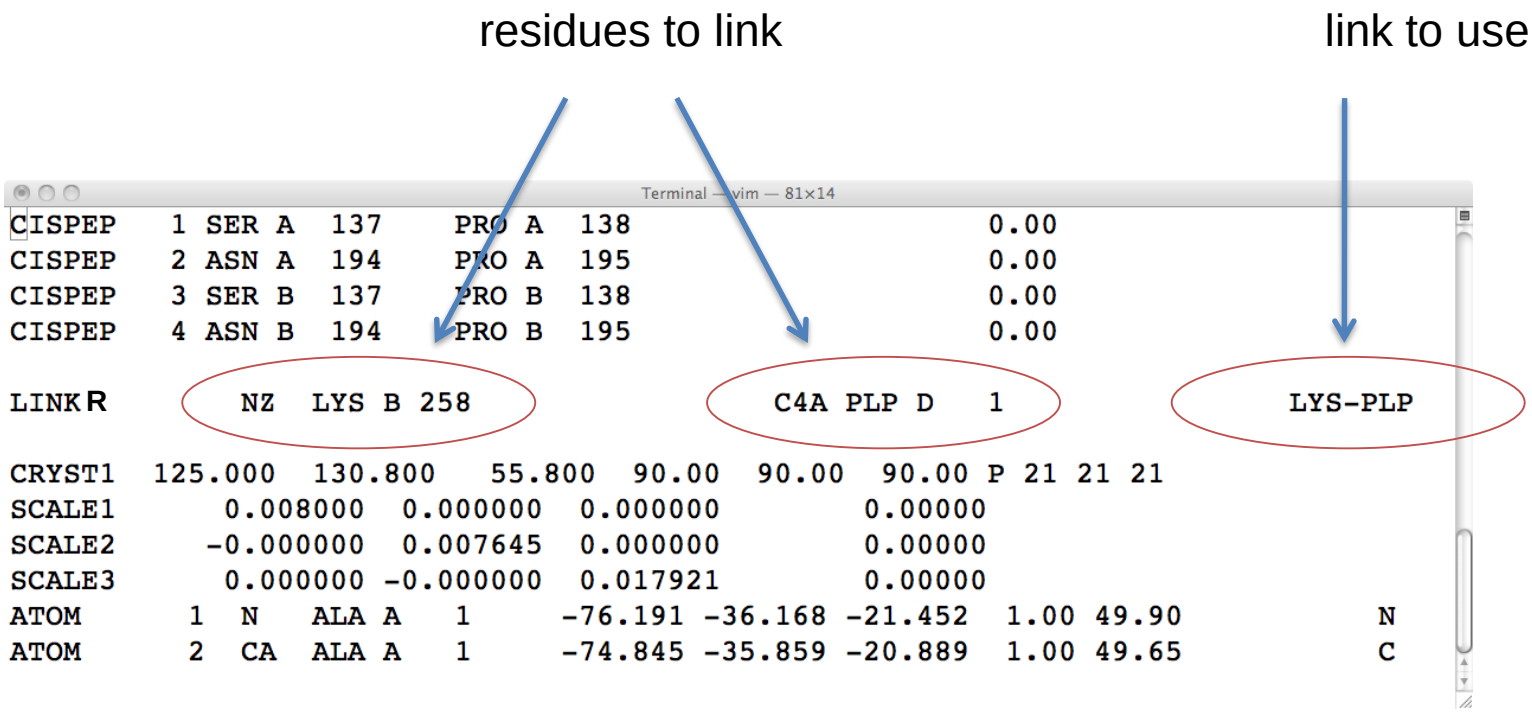
– non-modified monomer is taken from the standard library
– docking is performed, e.g. using coot:



– leaving atoms (O4A of PLP in this example) are removed

(2) Defining link in the pdb-file

Ccp4i2 should be able to edit the pdb/mmCIF file and add link record



Metals

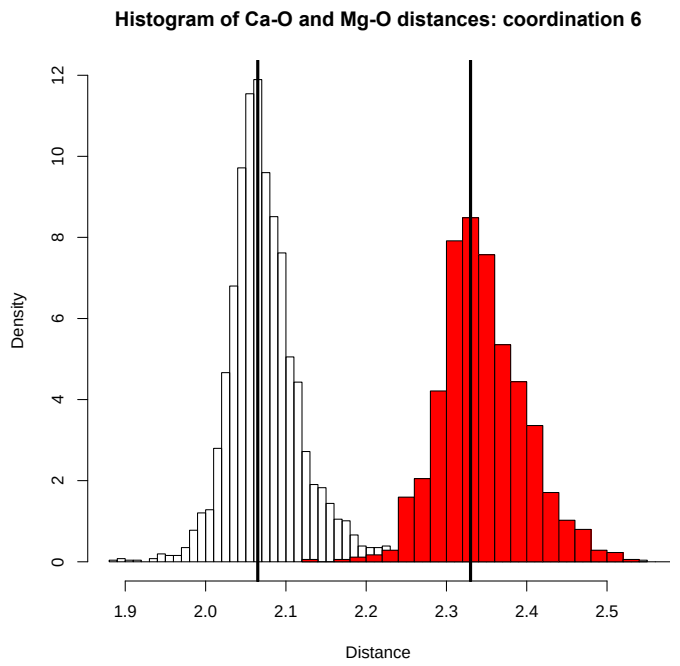
Metal identification

1. Distance to coordinating atoms
2. Coordination numbers
3. B values
4. Difference maps
5. Anomalous difference maps

Metals

Mg (black histogram) and Ca have definite coordination distance to O.
For Ca: distance is around 2.34
For Mg: distance is around 2.07

Actual distance will depend on surrounding stoms



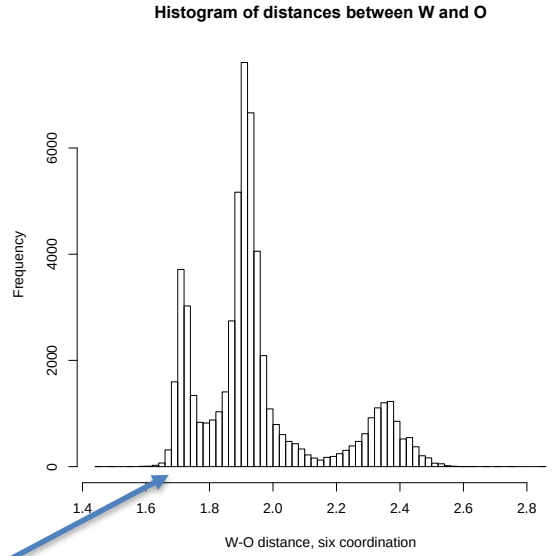
Red – Ca O distance: coordination 6
Black – Mg O distance coordination 6

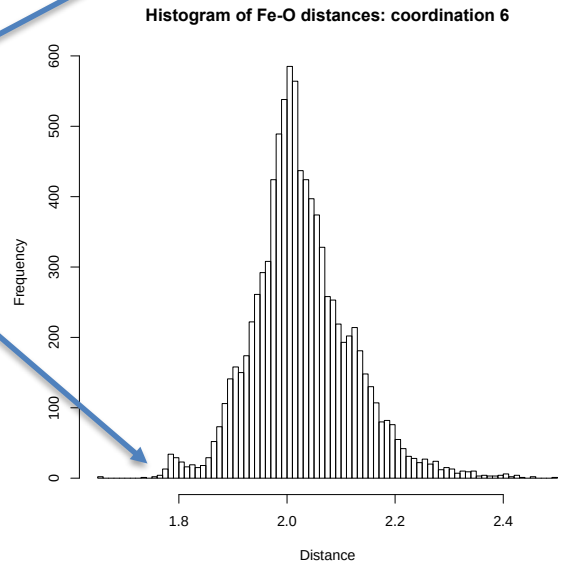
Not all metals have potential distances

W and Fe and O distances are not unique.
One of these distances correspond to double bond covalent.

It would help if you would know properties of the metal in your protein.

Double bond





Some of the links related to metals

ZN-CYS

ZN-HISND

ZN-HISNE

FE-CYS

FE-HISND

FE-HISNE

All these and many other links are available from the monomer library. You can access them using and editor and view the file

$CLIBD_MON/ list/mon_lib_list.cif

There are few more pretabulated links related to HEC-CYS and others

Conclusions and future perspectives

- Local graph based dynamic atom typing was designed
- Crystal structures from COD are validated
- AceDRG can be used to derive data from any small molecule database
- COD is currently used, but CSD or other databases can be used
- Derived data and AceDRG are available from CCP4
- Metals can be identified using distances, B values, difference maps, anomalous difference maps

Future

- Use AceDRG tables for ligand validation
- Deal with metal containing ligands (proteins)
- CCP4i2 interface for links and metals

Acknowledgement

Fei Long
Paul Emsley
Rob Nicholls

Saulius Grazulis
Andrius Merkys

Macin Wojdyr
Andrey Lebedev
Other members

Jon Agirre