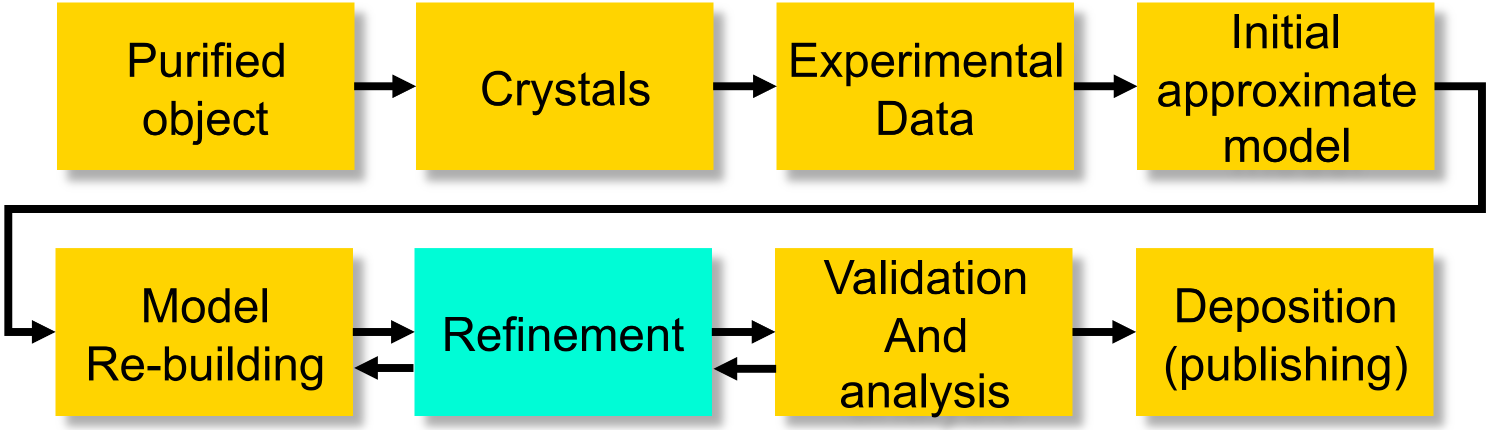
Automated structure refinement with *phenix.refine*

Pavel Afonine

Computation Crystallography Initiative
Physical Biosciences Division
Lawrence Berkeley National Laboratory, Berkeley CA, USA

Australasian Crystallography school
17th-24th July, 2010

Structure refinement workflow



Purified object

Crystals

Experimental Data

Initial approximate model

Model Re-building

Refinement

Validation And analysis

Deposition (publishing)

phenix.refine

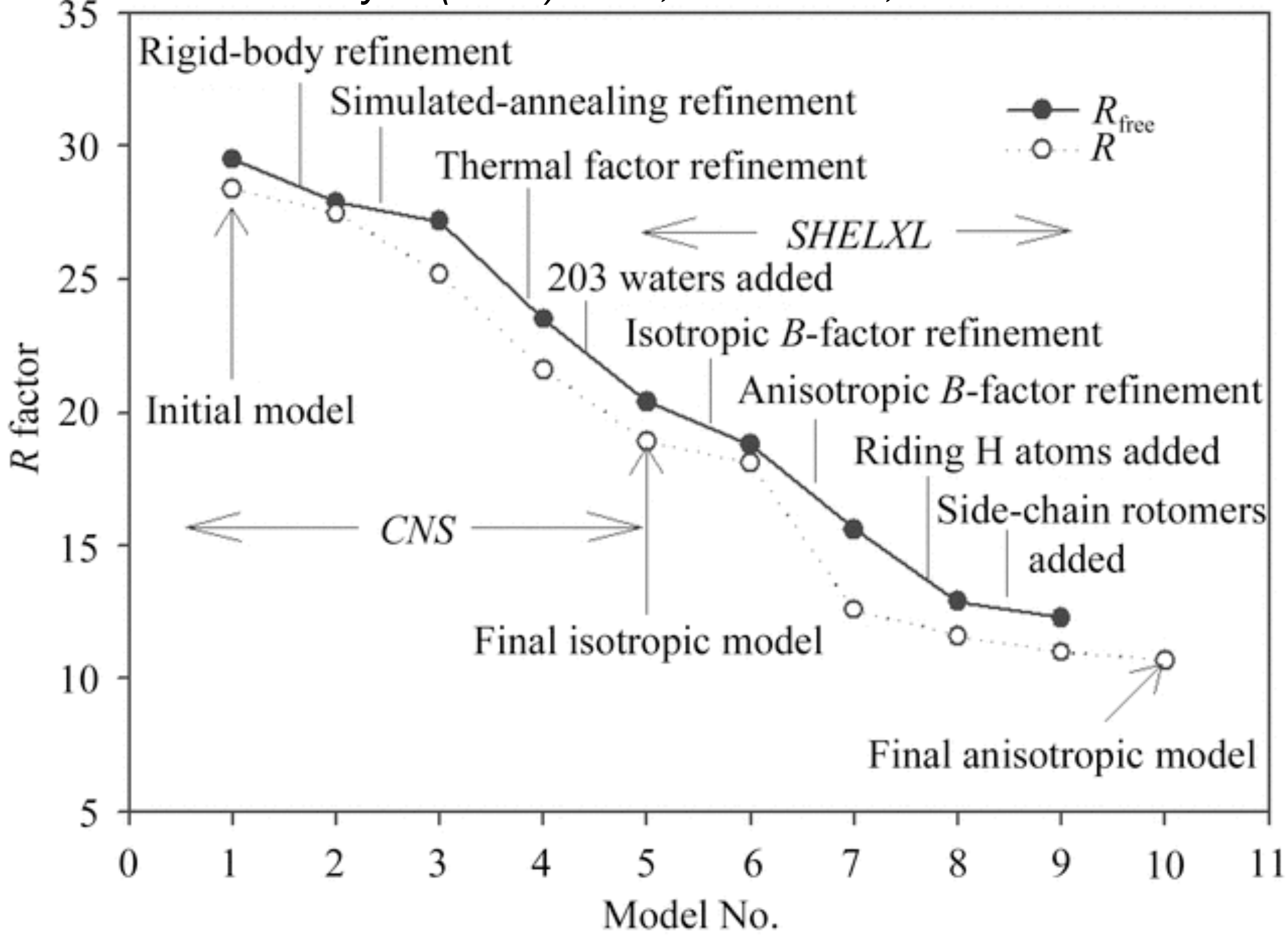
- **Highly-automated state-of-the-art structure refinement tool of PHENIX**
- Active development mainly at Lawrence Berkeley National Lab (USA):
  - *Paul Adams*
  - *Pavel Afonine*
  - *Nathaniel Echols*
  - *Ralf Grosse-Kunstleve*
  - Jeff Headd
  - *Nigel Moriarty*
  - Peter Zwart

+ *valuable scientific support by many others (*Marat Mustyakimov*, Sasha Urzhumtsev, Vladimir Lunin, …)*

Automation of structure refinement

- What used to be in the past … and often still the case nowadays

Acta Cryst. (2002). D58, 2009-2017, Yousef et al.

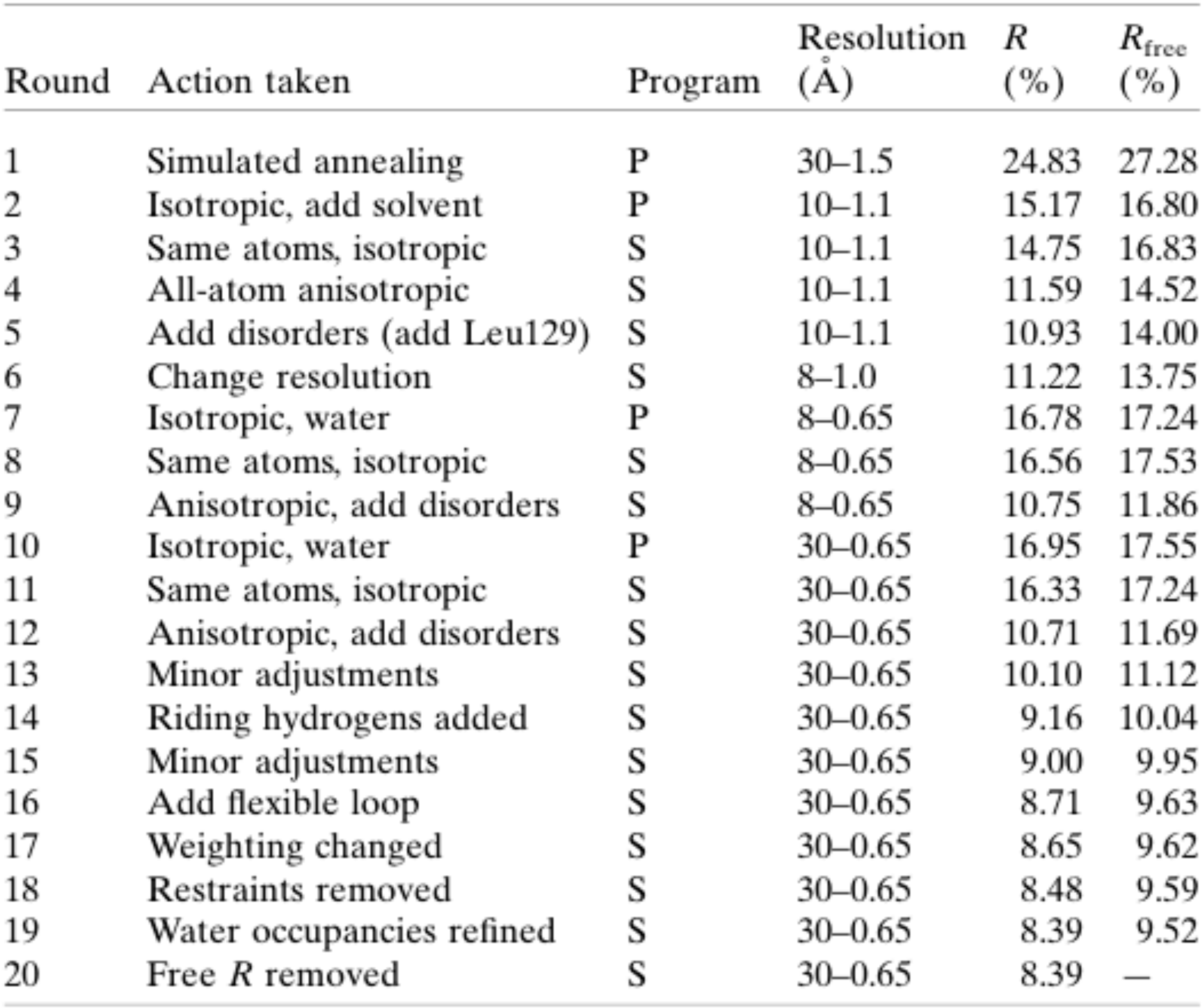


- Clearly, the modern software should do all these steps automatically

- PHENIX is making a good progress in achieving this goal

Automation of structure refinement

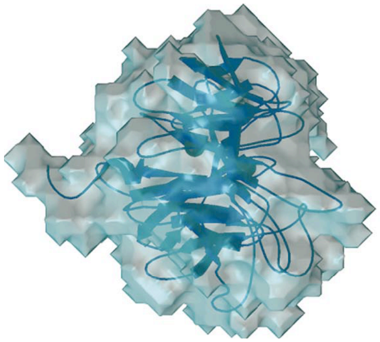
Wang *et al., Acta Cryst. (2007). D63, 1254-1268*



Round	Action taken	Program	Resolution (Å)	R (%)	R_{free} (%)
1	Simulated annealing	P	30–1.5	24.83	27.28
2	Isotropic, add solvent	P	10–1.1	15.17	16.80
3	Same atoms, isotropic	S	10–1.1	14.75	16.83
4	All-atom anisotropic	S	10–1.1	11.59	14.52
5	Add disorders (add Leu129)	S	10–1.1	10.93	14.00
6	Change resolution	S	8–1.0	11.22	13.75
7	Isotropic, water	P	8–0.65	16.78	17.24
8	Same atoms, isotropic	S	8–0.65	16.56	17.53
9	Anisotropic, add disorders	S	8–0.65	10.75	11.86
10	Isotropic, water	P	30–0.65	16.95	17.55
11	Same atoms, isotropic	S	30–0.65	16.33	17.24
12	Anisotropic, add disorders	S	30–0.65	10.71	11.69
13	Minor adjustments	S	30–0.65	10.10	11.12
14	Riding hydrogens added	S	30–0.65	9.16	10.04
15	Minor adjustments	S	30–0.65	9.00	9.95
16	Add flexible loop	S	30–0.65	8.71	9.63
17	Weighting changed	S	30–0.65	8.65	9.62
18	Restraints removed	S	30–0.65	8.48	9.59
19	Water occupancies refined	S	30–0.65	8.39	9.52
20	Free *R* removed	S	30–0.65	8.39	—

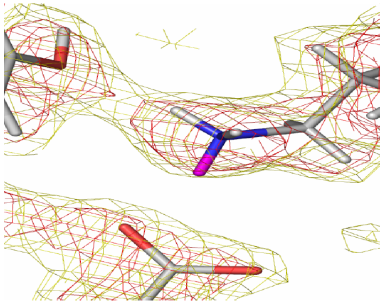
phenix.refine: single program for a very broad range of resolutions

Low



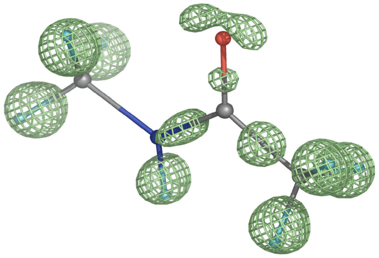
- Group ADP refinement
- Rigid body refinement
- Torsion Angle dynamics

Medium and High



- Restrained refinement (xyz, ADP: isotropic, anisotropic, mixed)
- Automatic water picking

Subatomic

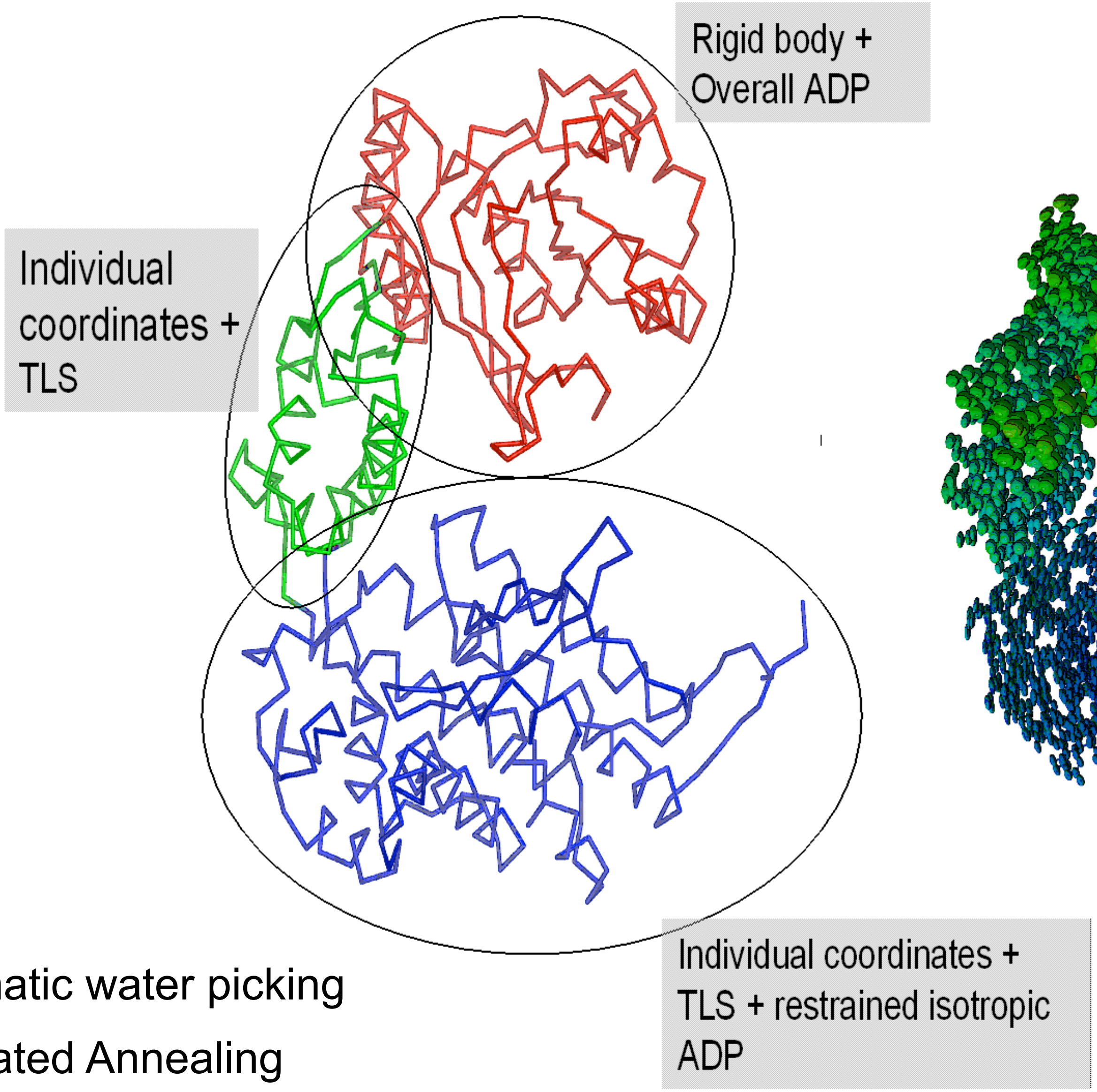


- Bond density model
- Unrestrained refinement
- FFT or direct
- Explicit hydrogens

- Automatic NCS restraints
- Simulated Annealing
- Automatic side chain rotamer fixing
- Occupancies (individual, group, automatic constrains for alternative conformations)
- Various targets: LS, ML, MLHL,…

- TLS refinement
- Use hydrogens at any resolution
- Refinement with twinned data
- X-ray, Neutron, joint X-ray + Neutron
- Built-in water picking and refinement

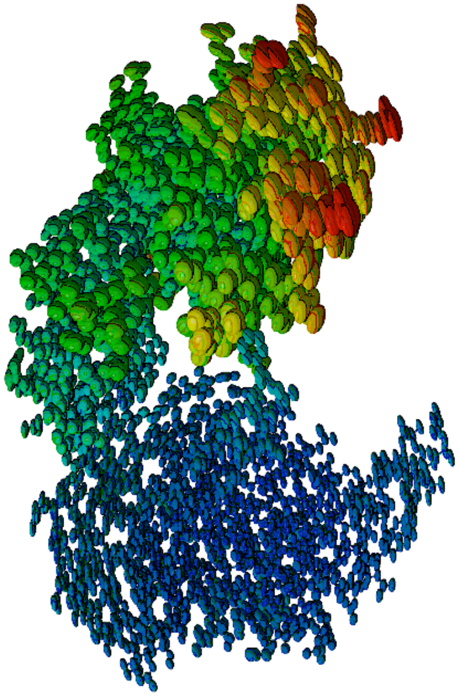
Refine any part of a model with any strategy: ***all in one run***



Rigid body + Overall ADP

Individual coordinates + TLS

Individual coordinates + TLS + restrained isotropic ADP



+ Automatic water picking

+ Simulated Annealing

+ Add and use hydrogens

Running phenix.refine

- Designed to be very easy to use

- Several ways of running:

1. Command line version:

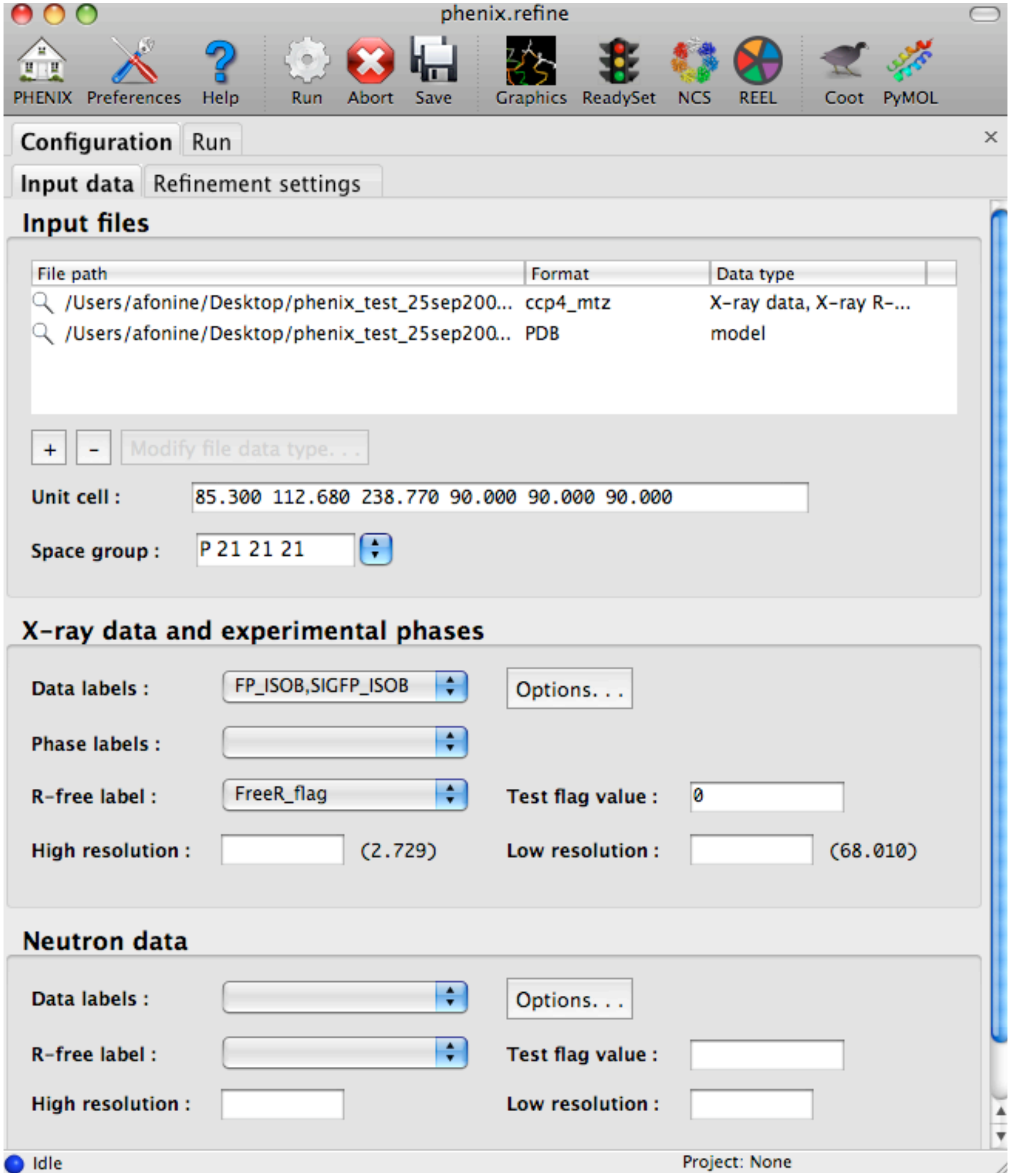
   ```
   phenix.refine model.pdb data.hkl [parameters]
   ```

   - Highly customizable (more than 300 parameters available to change)

2. Can be called from a Python script allowing to run it within different contexts

3. GUI

phenix.refine GUI



Input files

File path	Format	Data type
/Users/afonine/Desktop/phenix_test_25sep200...	ccp4_mtz	X-ray data, X-ray R-...
/Users/afonine/Desktop/phenix_test_25sep200...	PDB	model

Unit cell : 85.300 112.680 238.770 90.000 90.000 90.000

Space group : P 21 21 21

X-ray data and experimental phases

Data labels : FP_ISOB,SIGFP_ISOB

Phase labels :

R-free label : FreeR_flag

Test flag value : 0

High resolution : (2.729)

Low resolution : (68.010)

Neutron data

Data labels :

R-free label :

Test flag value :

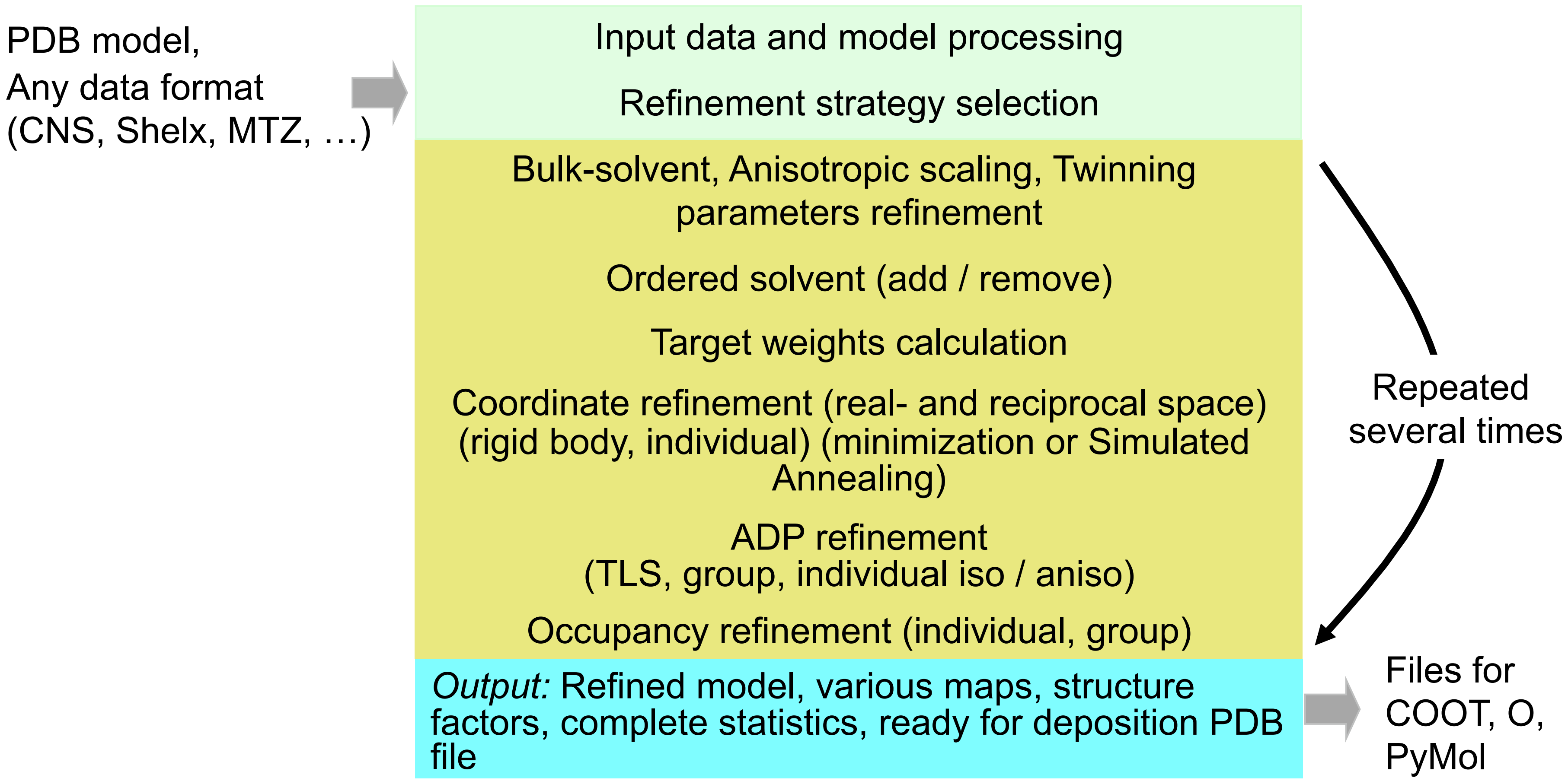
High resolution :

Low resolution :

Idle

Project: None

Refinement flowchart



PDB model,
Any data format
(CNS, Shelx, MTZ, …)

Input data and model processing

Refinement strategy selection

Bulk-solvent, Anisotropic scaling, Twinning parameters refinement

Ordered solvent (add / remove)

Target weights calculation

Coordinate refinement (real- and reciprocal space) (rigid body, individual) (minimization or Simulated Annealing)

ADP refinement
(TLS, group, individual iso / aniso)

Occupancy refinement (individual, group)

Repeated several times

Output: Refined model, various maps, structure factors, complete statistics, ready for deposition PDB file

Files for COOT, O, PyMol

Bulk-solvent modeling and anisotropic scaling

- **Search for optimal U$_{\mathbf{CRYSTAL}}$, *k*$_{\mathbf{SOL}}$ and *B*$_{\mathbf{SOL}}$:**

$$\mathbf{F}_{\mathrm{MODEL}} = k_{\mathrm{OVERALL}} e^{-\mathbf{s}\mathbf{U}_{\mathrm{CRYSTAL}}\,\mathbf{s}^{t}} \left(\mathbf{F}_{\mathrm{CALC\_ATOMS}} + k_{\mathrm{SOL}} e^{-\frac{B_{\mathrm{SOL}}\;s^{2}}{4}} \mathbf{F}_{\mathrm{MASK}} \right)$$

A robust bulk-solvent correction and anisotropic scaling procedure.
Acta Cryst. (2005). P.V. Afonine, R.W. Grosse-Kunstleve & P.D. Adams

- **In case of twinning, twin fraction (α) is refined too**

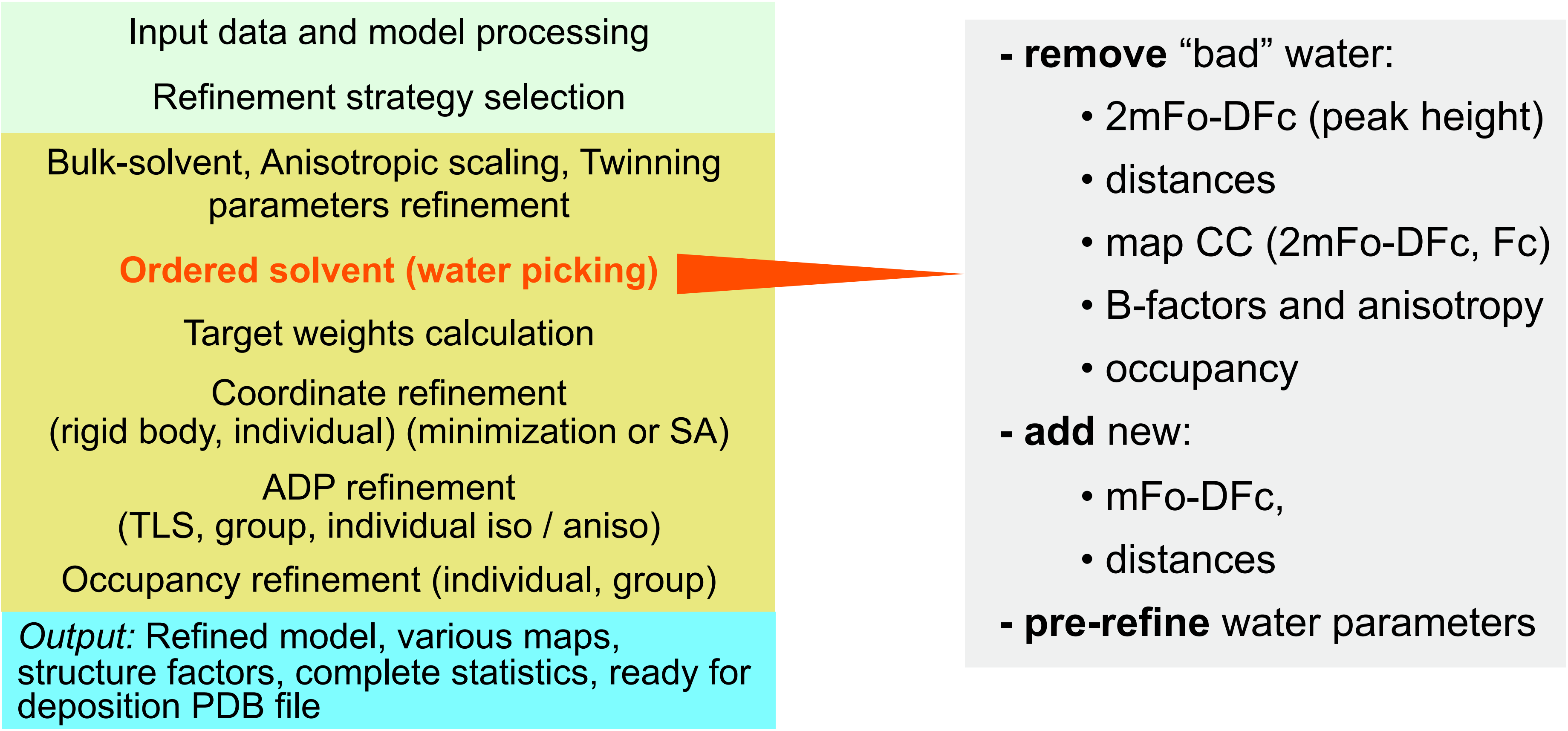
$$F_{\mathrm{MODEL}} = \left|\mathbf{F}_{\mathrm{MODEL}}\right| = k_{\mathrm{OVERALL}} \sqrt{\alpha\left|\mathbf{F}_{\mathrm{M}}(\mathbf{h})\right|^{2} + (1-\alpha)^{2}\left|\mathbf{F}_{\mathrm{M}}(\mathbf{Th})\right|^{2}}$$

$$\mathbf{F}_{\mathrm{M}}(\mathbf{h}) = e^{-\mathbf{s}\mathbf{U}_{\mathrm{CRYSTAL}}\,\mathbf{s}^{t}} \left(\mathbf{F}_{\mathrm{CALC\_ATOMS}} + k_{\mathrm{SOL}} e^{-\frac{B_{\mathrm{SOL}}\;s^{2}}{4}} \mathbf{F}_{\mathrm{MASK}} \right)$$

Automatic Water Picking

✓ Water is updated (add/remove/refine) automatically as part of refinement run:

- No need to do it as a separate step using external tools



Input data and model processing

Refinement strategy selection

Bulk-solvent, Anisotropic scaling, Twinning parameters refinement

Ordered solvent (water picking)

Target weights calculation

Coordinate refinement (rigid body, individual) (minimization or SA)

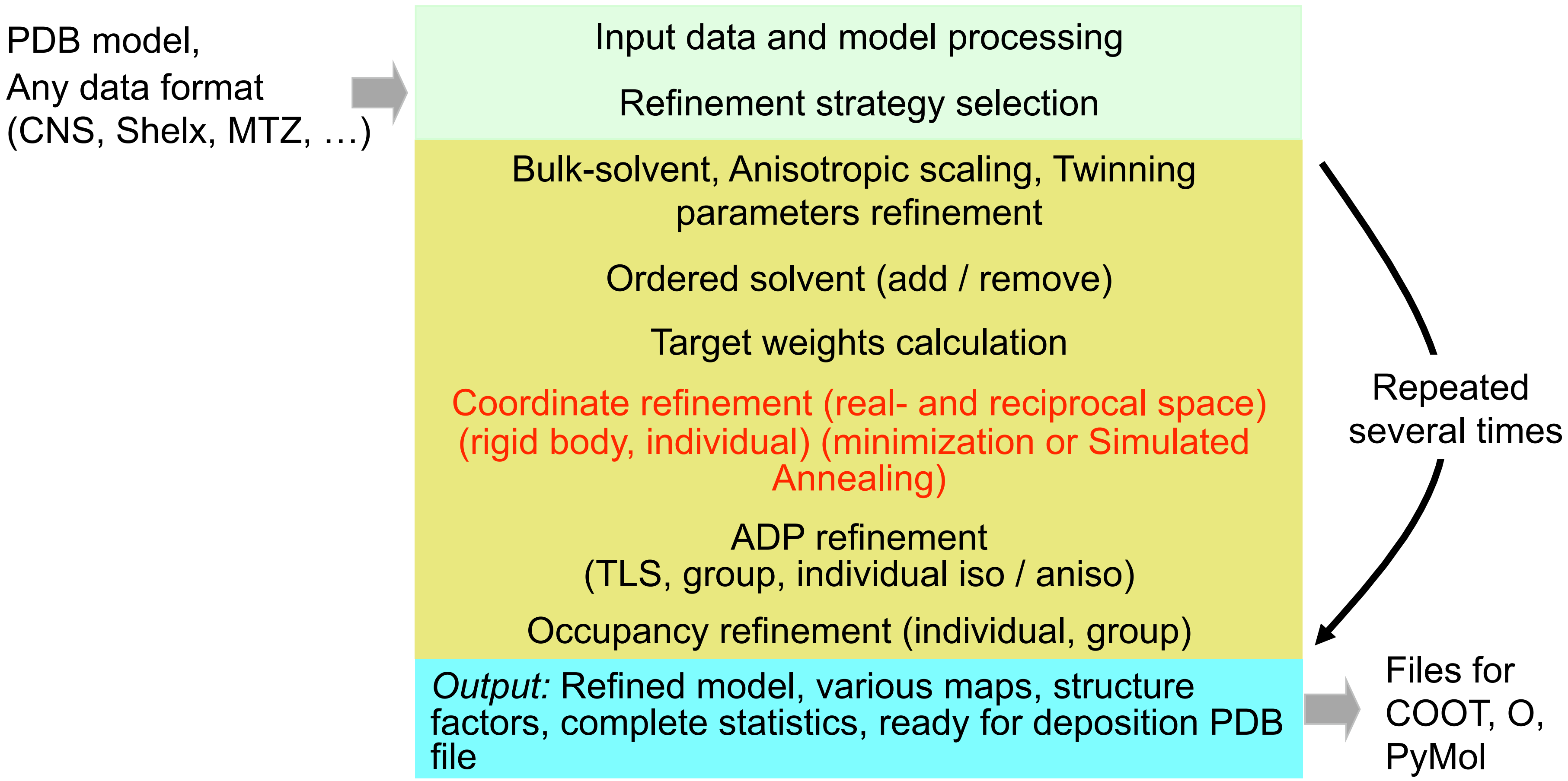
ADP refinement (TLS, group, individual iso / aniso)

Occupancy refinement (individual, group)

Output: Refined model, various maps, structure factors, complete statistics, ready for deposition PDB file

- **remove** “bad” water:
  - 2mFo-DFc (peak height)
  - distances
  - map CC (2mFo-DFc, Fc)
  - B-factors and anisotropy
  - occupancy
- **add** new:
  - mFo-DFc,
  - distances
- **pre-refine** water parameters

Refinement flowchart



PDB model,
Any data format
(CNS, Shelx, MTZ, …)

Input data and model processing

Refinement strategy selection

Bulk-solvent, Anisotropic scaling, Twinning parameters refinement

Ordered solvent (add / remove)

Target weights calculation

Coordinate refinement (real- and reciprocal space) (rigid body, individual) (minimization or Simulated Annealing)

ADP refinement
(TLS, group, individual iso / aniso)

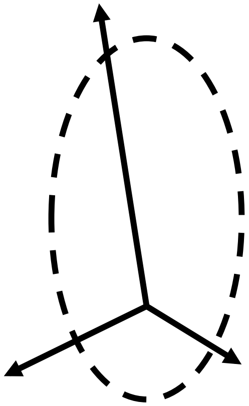
Occupancy refinement (individual, group)

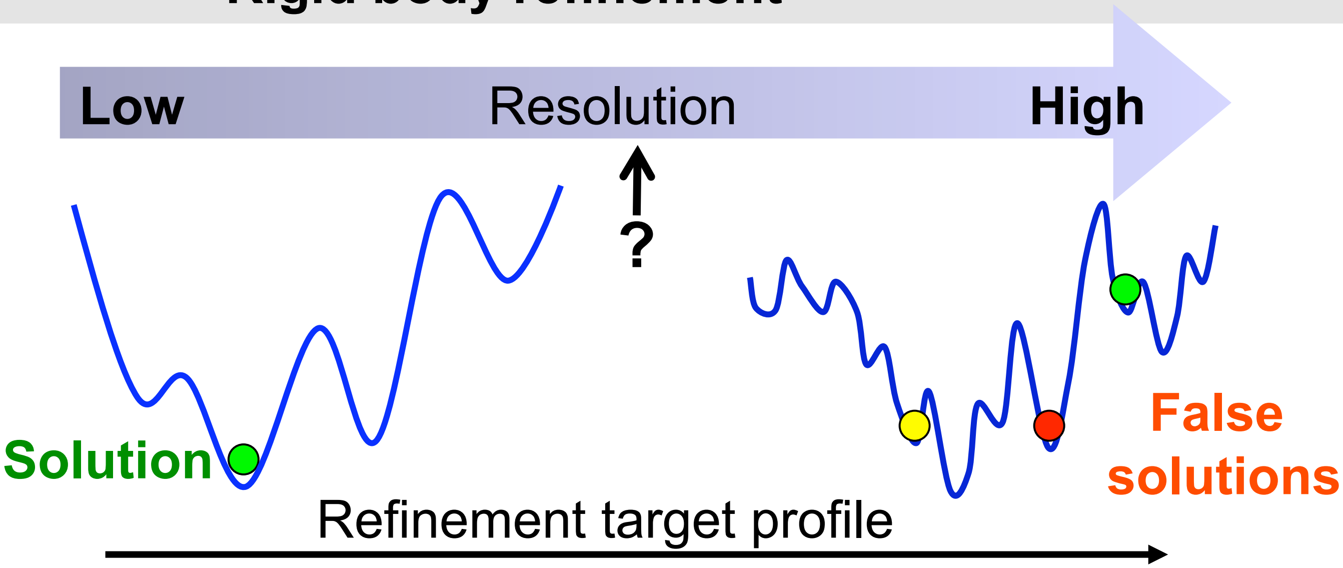
Output: Refined model, various maps, structure factors, complete statistics, ready for deposition PDB file

Repeated several times

Files for
COOT, O,
PyMol

Rigid body refinement





Low

Resolution

High

?

Solution

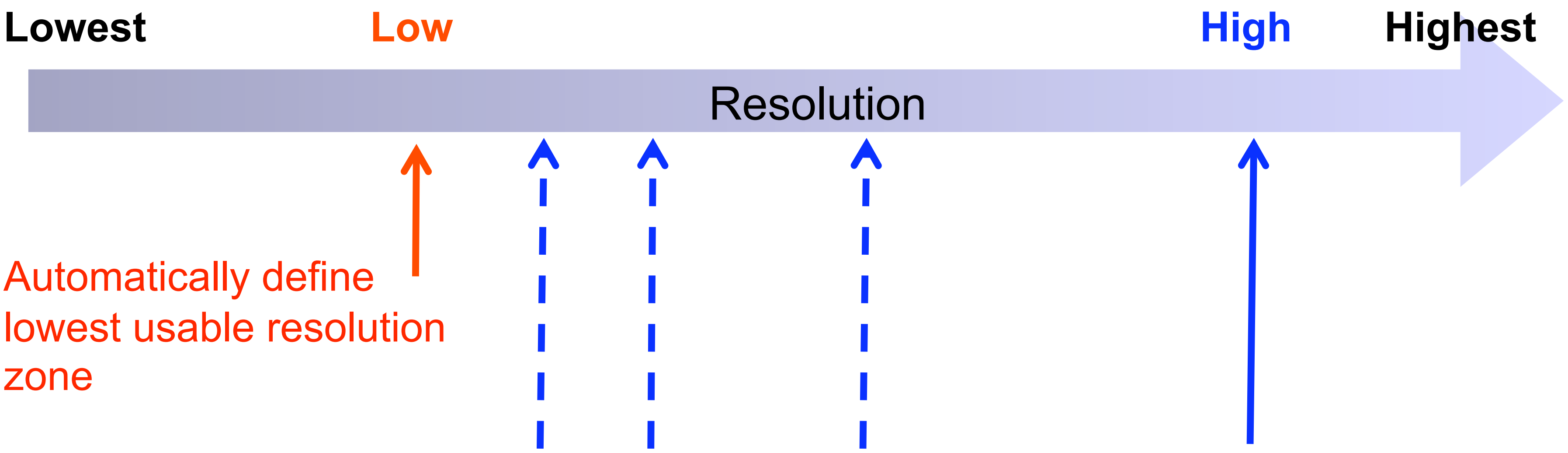
False solutions

Refinement target profile

- **Rigid body refinement challenges:**
  - Need to use *low resolution* reflections to achieve a solution
    - Using too low resolution may not be good
    - Need to use higher resolution data to assure better solution
    - How to define low-high resolution border (3…4…6A)?
- **PHENIX MZ protocol** makes all these decisions automatically

Automatic multiple-zone rigid-body refinement with a large convergence radius.
P. V. Afonine, R. W. Grosse-Kunstleve, A. Urzhumtsev and P. D. Adams. J. Appl. Cryst. 42, 607-615 (2009)

Automated Rigid Body Refinement in PHENIX (MZ protocol)



Automatically define lowest usable resolution zone

This insures quick and reliable convergence

Gradually add higher resolution reflections. This supports convergence and assures higher precision of the solution.

During rigid body refinement some large model movements are expected.
This invalidates the solvent mask, so the bulk-solvent model is updated at each step.

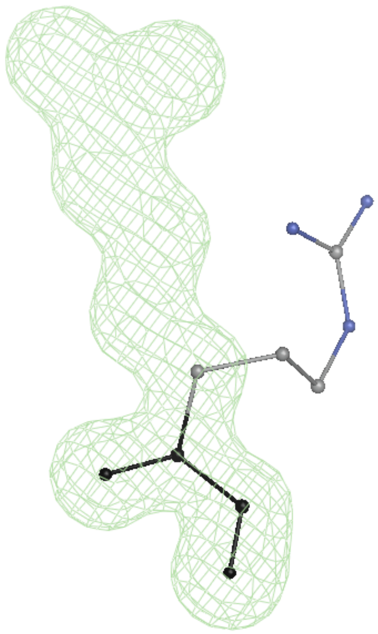
- All parameters used in the protocol are optimized to achieve the highest convergence radius at minimal runtime.
  - This is done by the grid search over ~100000 trial refinements using more than 100 different structures.

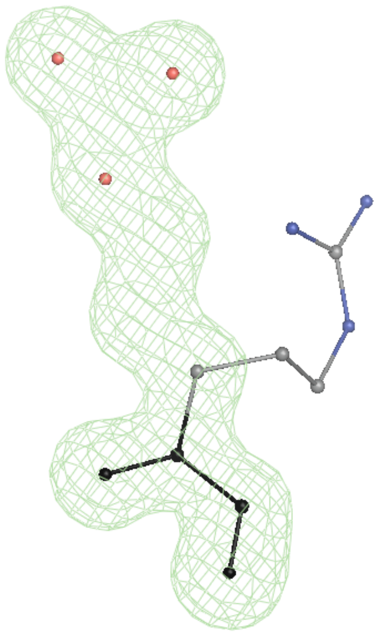
Dual-space refinement: combining real and reciprocal space refinement

Why real-space refinement ?

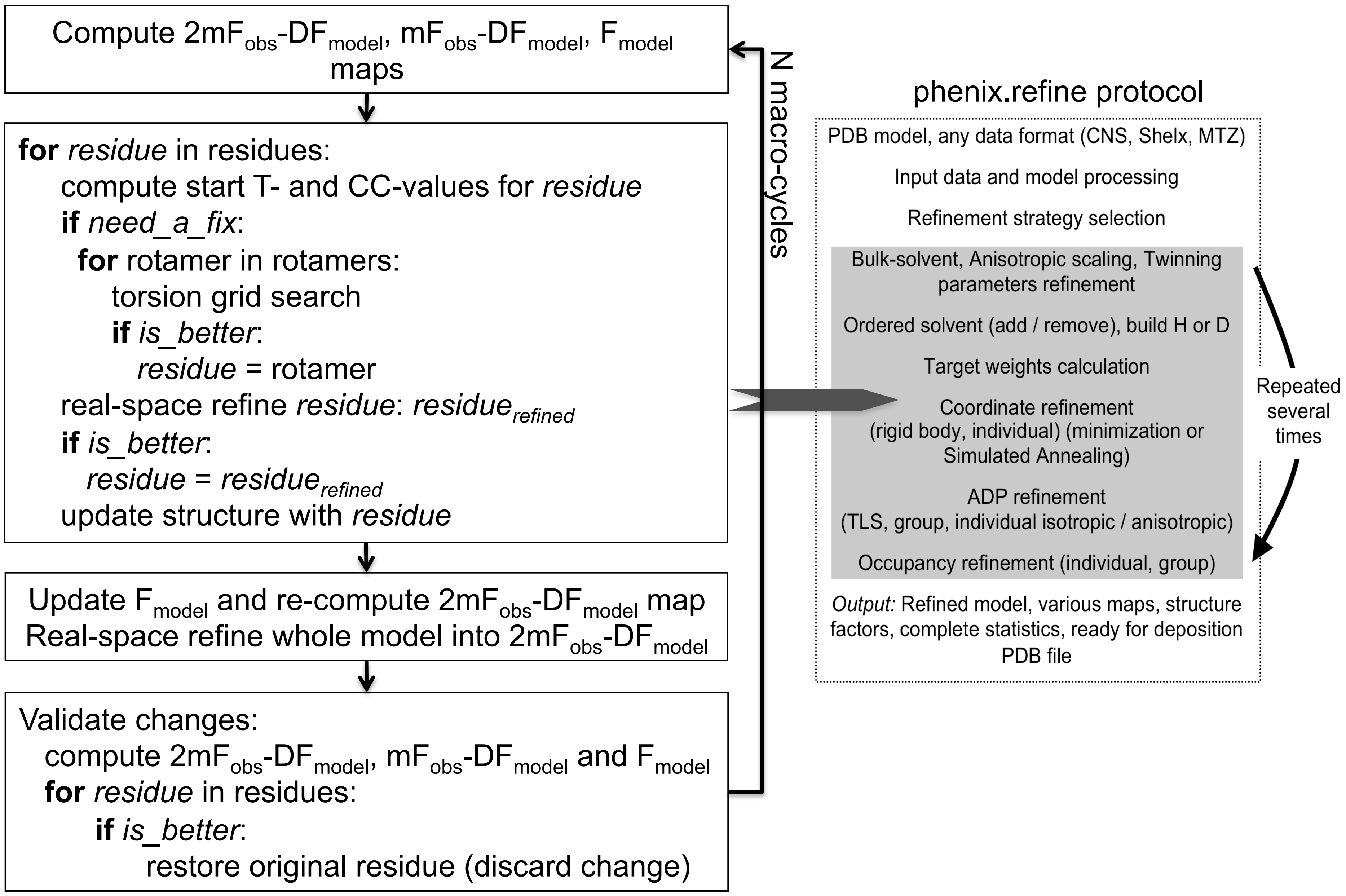
- Can be done locally (for example, for a residue or ligand)
- Grid search can be used -> Convergence radius can be dramatically increased compared to gradient driven-refinement or SA
- Ordered solvent update can be enabled at earlier stage

✓ **Eliminate the tedium of manual work on fixing side chains on graphics**





Local real-space refinement



Compute $2mF_{obs}$-DF_{model}, mF_{obs}-DF_{model}, F_{model} maps

```
for residue in residues:
    compute start T- and CC-values for residue
    if need_a_fix:
      for rotamer in rotamers:
          torsion grid search
          if is_better:
            residue = rotamer
    real-space refine residue: $residue_{refined}$
    if is_better:
      residue = $residue_{refined}$
    update structure with residue
```

Update F_{model} and re-compute $2mF_{obs}$-DF_{model} map

Real-space refine whole model into $2mF_{obs}$-DF_{model}

```
Validate changes:
  compute $2mF_{obs}$-$DF_{model}$, $mF_{obs}$-$DF_{model}$ and $F_{model}$
  for residue in residues:
      if is_better:
          restore original residue (discard change)
```

phenix.refine protocol

PDB model, any data format (CNS, Shelx, MTZ)

Input data and model processing

Refinement strategy selection

Bulk-solvent, Anisotropic scaling, Twinning parameters refinement

Ordered solvent (add / remove), build H or D

Target weights calculation

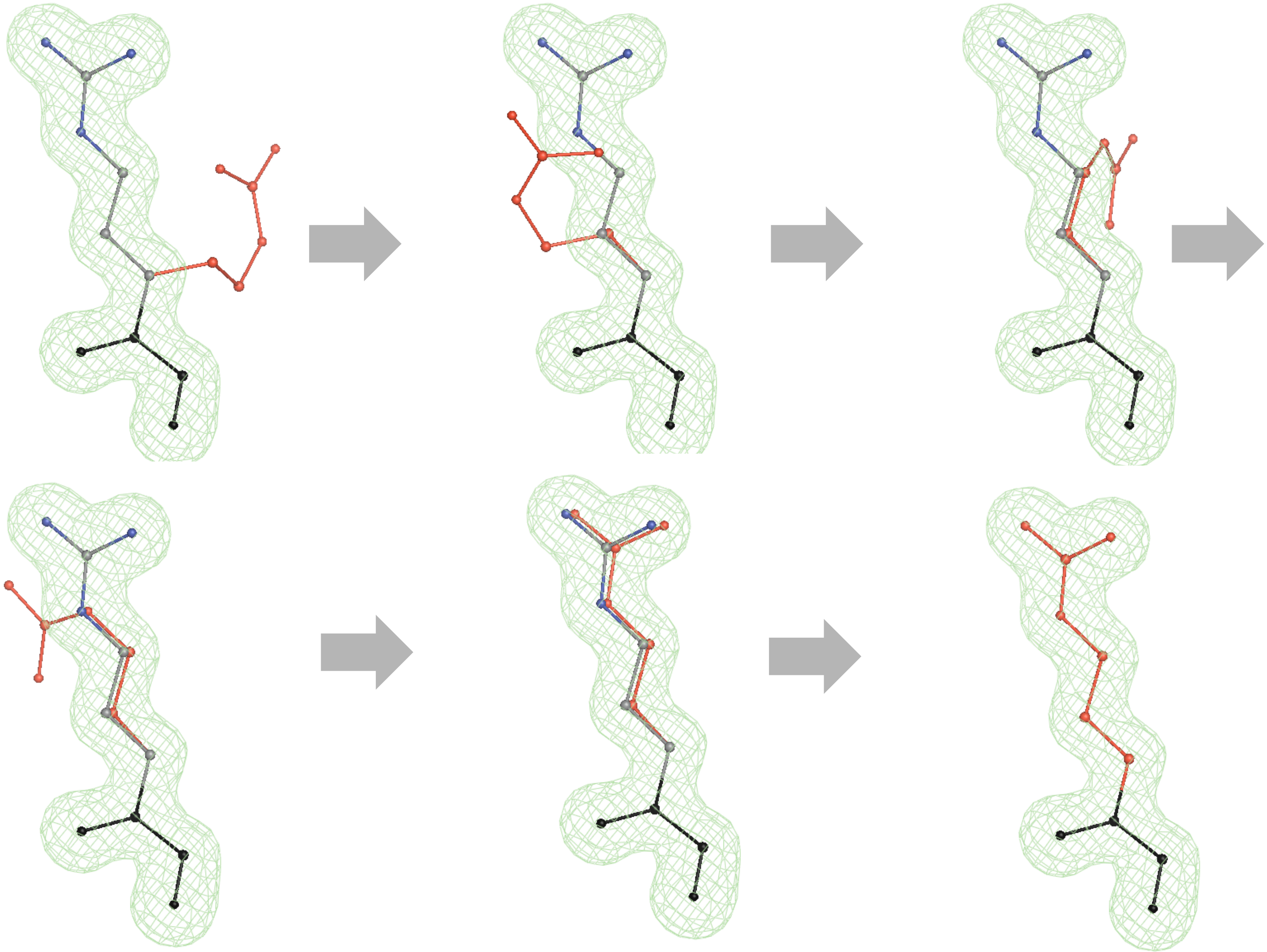
Coordinate refinement (rigid body, individual) (minimization or Simulated Annealing)

ADP refinement (TLS, group, individual isotropic / anisotropic)

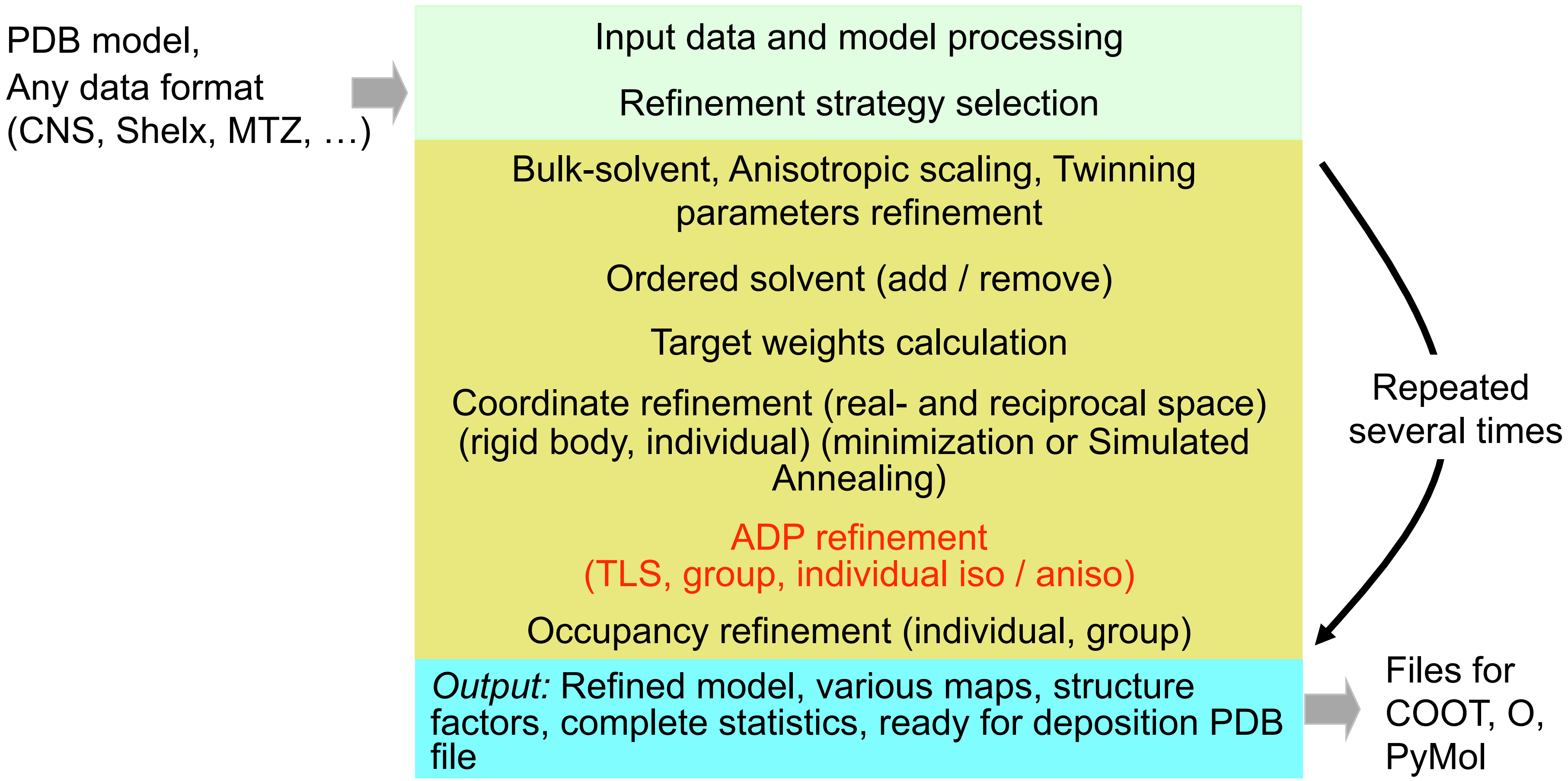
Occupancy refinement (individual, group)

Output: Refined model, various maps, structure factors, complete statistics, ready for deposition PDB file

Real-space refinement: torsion grid search



Refinement flowchart



PDB model,
Any data format
(CNS, Shelx, MTZ, …)

Input data and model processing

Refinement strategy selection

Bulk-solvent, Anisotropic scaling, Twinning parameters refinement

Ordered solvent (add / remove)

Target weights calculation

Coordinate refinement (real- and reciprocal space) (rigid body, individual) (minimization or Simulated Annealing)

ADP refinement
(TLS, group, individual iso / aniso)

Occupancy refinement (individual, group)

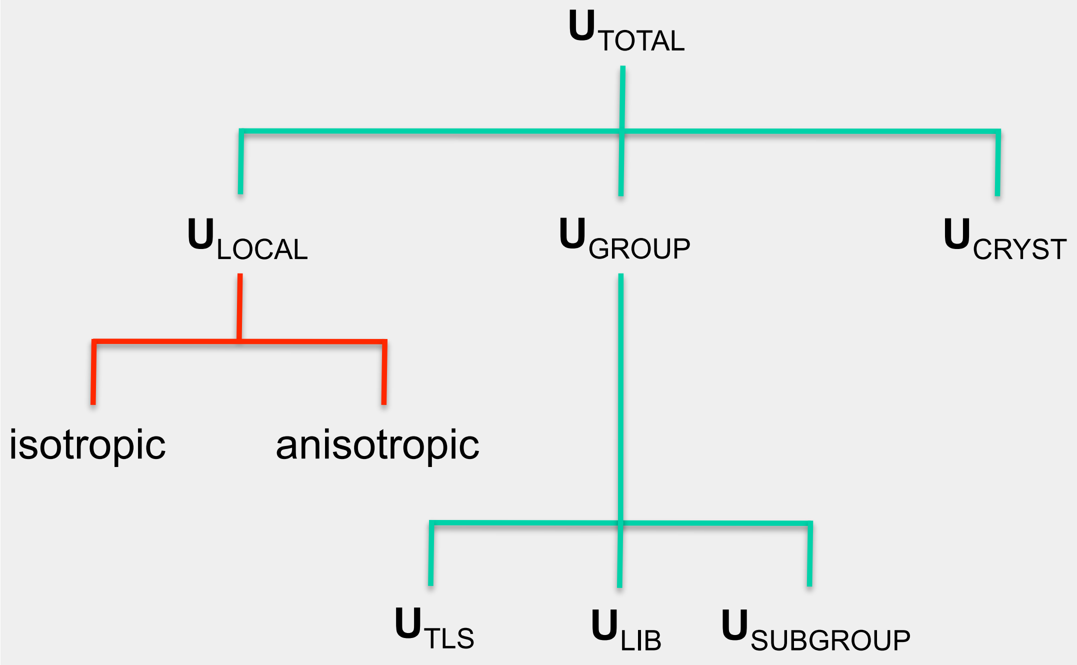
Repeated several times

Output: Refined model, various maps, structure factors, complete statistics, ready for deposition PDB file

Files for
COOT, O,
PyMol

Atomic Displacement Parameters (ADP or "B-factors")

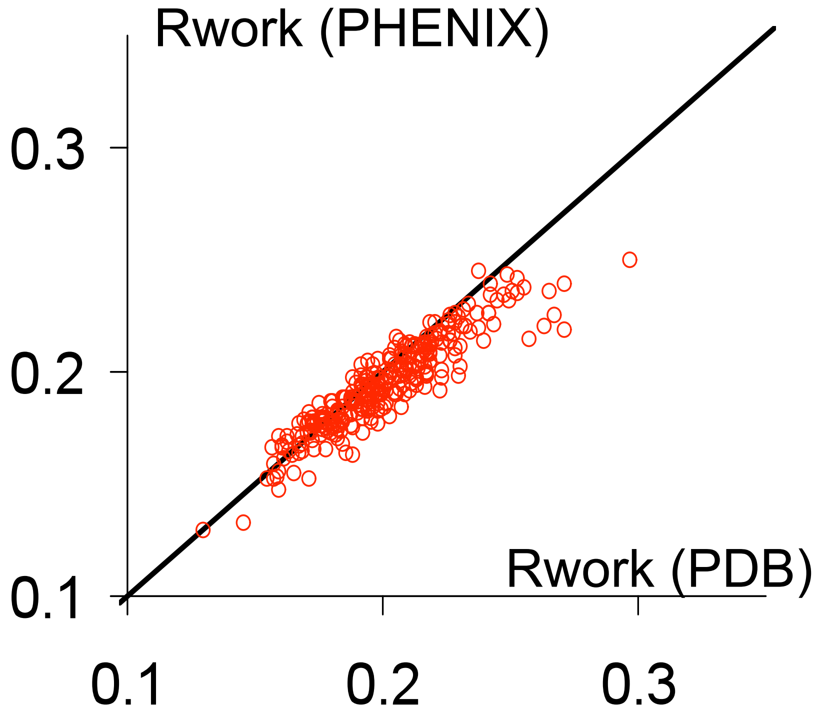
- **Total ADP** $U_{TOTAL} = U_{CRYST} + U_{GROUP} + U_{LOCAL}$

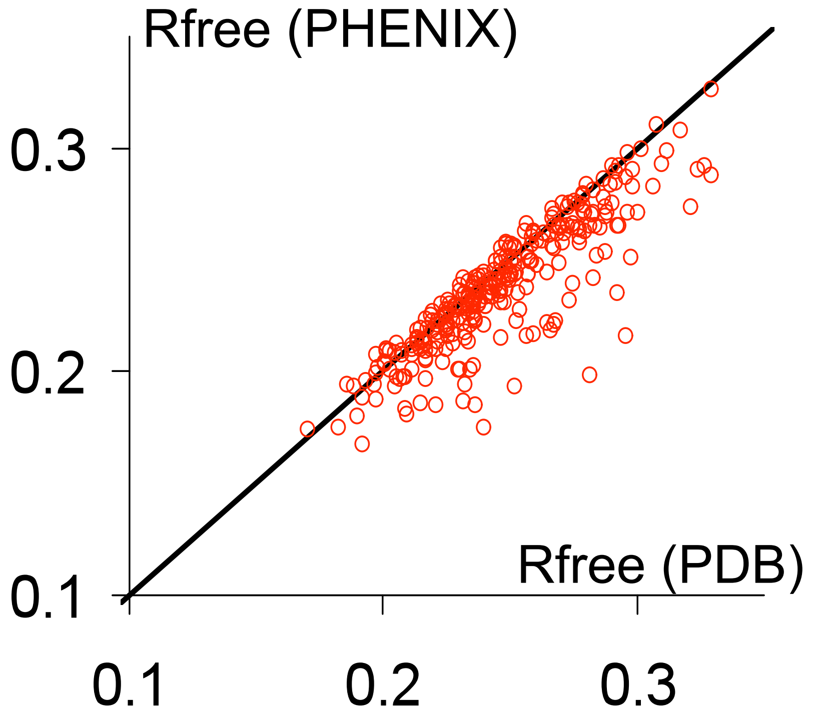


- $\mathbf{U_{CRYST}}$ – overall anisotropic scale (6 parameters).
- $\mathbf{U_{TLS}}$ – rigid body displacements of molecules, domains, secondary structure elements. $U_{TLS} = T + ALA^t + AS + S^tA^t$ (20 TLS parameters per group).
- $\mathbf{U_{LOCAL}}$ – local vibration of individual atoms.
- U_{LIB} – librational motion of side chain around bond vector.

TLS refinement in PHENIX: robust and efficient

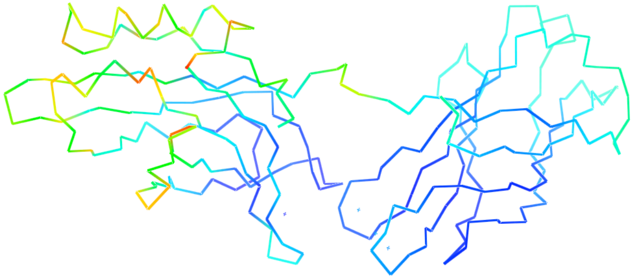
- Highly optimized algorithm based on systematic re-refinement of ~350 PDB models
- In most of cases *phenix.refine* produces better R-factors compared to published
- Don’t crash or get “unstable”





ADP refinement: example

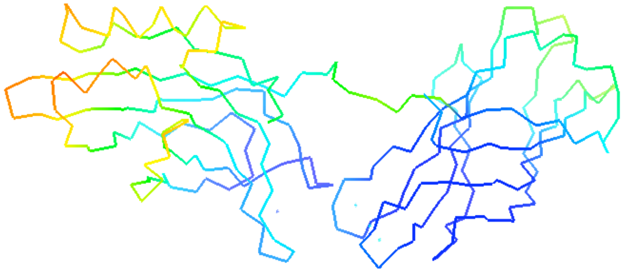
Synaptotagmin refinement at 3.2 Å (PDB code: 1DQV)



CNS (original refinement)

R-free = ***34*** *%*

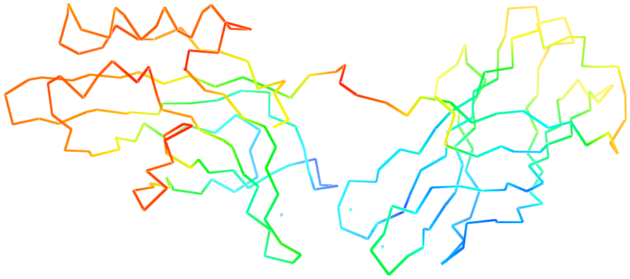
R = ***29*** *%*



PHENIX – Isotropic restrained ADP

R-free = ***28*** *%*

R = ***23*** *%*



PHENIX – TLS + Isotropic ADP

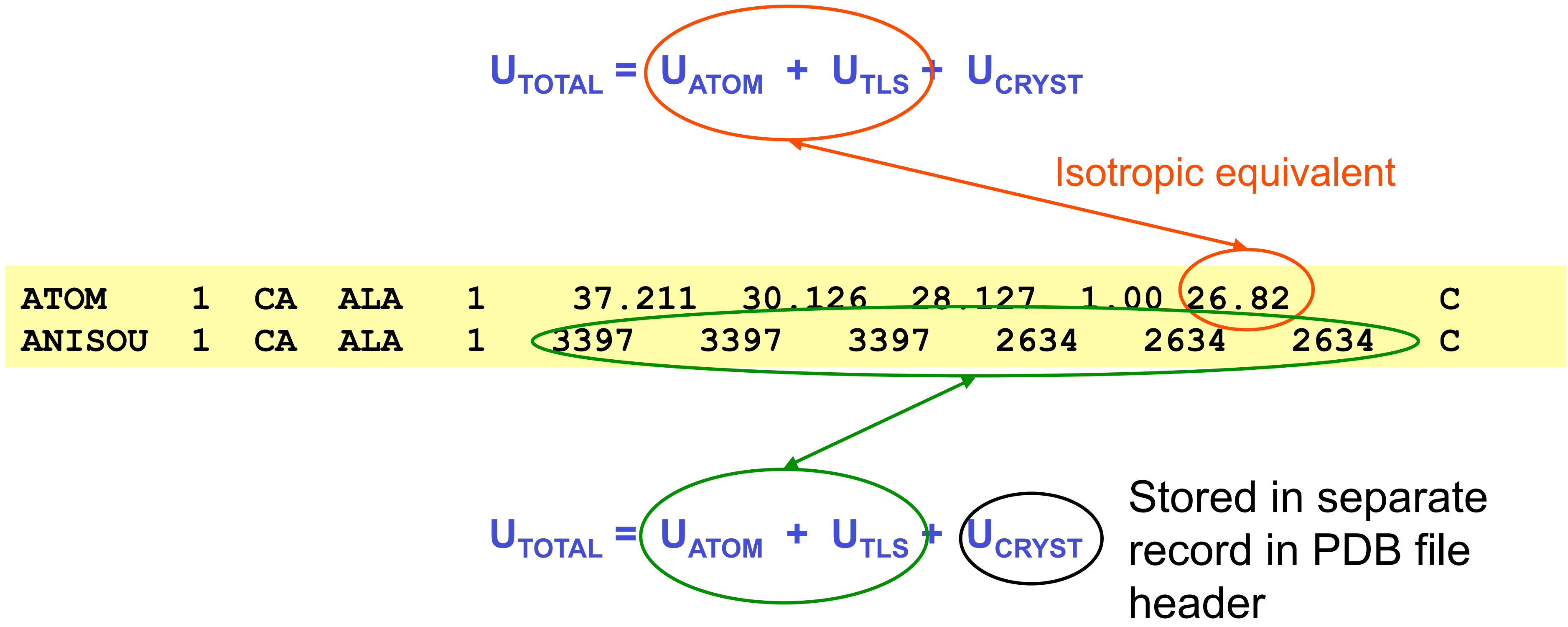
R-free = ***25*** *%*

R = ***20*** *%*

9% improvement in both Rwork and Rfree !

ADP refinement: what goes to PDB

phenix.refine outputs TOTAL B-factor (iso- and anisotropic):



$$U_{TOTAL} = U_{ATOM} + U_{TLS} + U_{CRYST}$$

Isotropic equivalent

```
ATOM      1  CA  ALA   1    37.211  30.126  28.127  1.00 26.82        C
ANISOU    1  CA  ALA   1   3397   3397   3397   2634   2634   2634    C
```

$$U_{TOTAL} = U_{ATOM} + U_{TLS} + U_{CRYST}$$

Stored in separate
record in PDB file
header

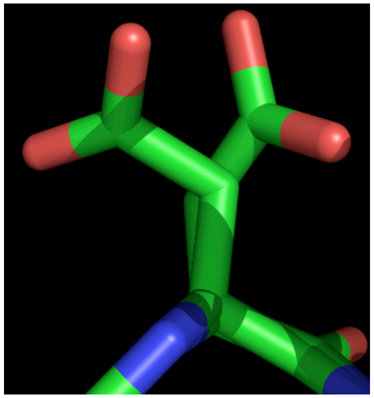
Atom records are self-consistent:

✓ Straightforward visualization (color by B-factors, or anisotropic ellipsoids)

✓ Straightforward computation of other statistics (R-factors, etc.) – no need to use external helper programs for any conversions.

Occupancy refinement

- Automatic constraints for occupancies of atoms in alternate locations



- Any user defined selections for individual and/or group occupancy refinement can be added on top of the automatic selection.

```
ATOM      1  N  AARG A 192      -5.782  17.932  11.414  0.72  8.38           N
ATOM      2  CA AARG A 192      -6.979  17.425  10.929  0.72 10.12           C
ATOM      3  C  AARG A 192      -6.762  16.088  10.271  0.72  7.90           C
ATOM      7  N  BARG A 192     -11.719  17.007   9.061  0.28  9.89           N
ATOM      8  CA BARG A 192     -10.495  17.679   9.569  0.28 11.66           C
ATOM      9  C  BARG A 192      -9.259  17.590   8.718  0.28 12.76           C
```

```
ATOM    549  AU        A  34     -23.064   7.146 -23.942  0.78 15.44          Au
```

```
ATOM    549  HA3 ARG A  34     -23.064   7.146 -23.942  1.00 15.44           H
ATOM    550  H  AARG A  34     -24.447   7.644 -21.715  0.15  8.34           H
ATOM    551  D  BARG A  34     -24.447   7.644 -21.715  0.85  7.65           D
ATOM    552  N   ARG A  35     -22.459   9.801 -22.791  1.00  8.54           N
```

```
ATOM      6  S   SO4     1       1.302   1.419   1.560  0.70 13.00
ATOM      7  O1  SO4     1       1.497   1.295   0.118  0.70 11.00
ATOM      8  O2  SO4     1       1.098   0.095   2.140  0.70 10.00
ATOM      9  O3  SO4     1       2.481   2.037   2.159  0.70 14.00
ATOM     10  O4  SO4     1       0.131   2.251   1.823  0.70 12.00
```

Occupancy refinement – more examples

```
ATOM   3690  O2 AEDO C   1      23.106  -3.999  -8.239  0.58 15.69           O
ATOM   3691  C2 AEDO C   1      21.710  -4.102  -8.630  0.58 15.43           C
ATOM   3692  C1 AEDO C   1      20.965  -2.841  -8.282  0.58 16.78           C
ATOM   3693  O1 AEDO C   1      21.111  -2.587  -6.901  0.58 19.33           O
ATOM   3687  I  BIOD C   1      21.798  -3.596  -7.915  0.42 34.88           I
```

Refinement with twinned data

- Two steps to perform twin refinement:

  - run phenix.xtriage to get twin operator (twin law):

```
% phenix.xtriage data.mtz
```

  - run phenix.refine:

```
% phenix.refine model.pdb data.mtz twin_law="-h-k,k,-l"
```

- Taking twinning into account makes (big) difference:

Interleukin mutant (PDB code: 1l2h)

	R/R-free (%)
PHENIX (no twinning):	24.9 / 27.4
PHENIX (twin refinement):	15.3 / 19.2

Hydrogen atoms in refinement

- **Some facts about hydrogen atoms:**

  - H atoms are not visible in X-ray maps at “typical macromolecular” resolutions, that is ~1Å and lower. This is because:
    - H atom is a weak scatterer (much weaker than C, N or O atoms)
    - models contain too much noise so the H contribution is hidden in it. Ideally (nearly error free model) one would see H even at ~2Å resolution.

  - Some or most of H atoms can be seen in maps at ultra-high resolutions (~1Å and higher):
    - The resolution itself is not the sufficient condition to see H: the noise level should be low (small *R*-factor).

  - Hydrogen atoms constitute nearly 50% of the total atoms in protein structures. Typical example: Fab structure (PDB code: 1f8t): 3593 non-H atoms, 3269 H atoms.

  - Since H is a weak scatterer, it mostly contributes to the low resolution (and not to the high!). The reason why we see H atoms only in structures corresponding to high resolution data is because these structures are typically accurate enough and complete so the noise level is small (small *R*-factor).

Hydrogen atoms in refinement

- Expected benefits from using H atoms in refinement:

  - Improve R-factors (typically reduces R-factor by 1-2%)
  - Improve model geometry (eliminate bad clashes)
  - Model residual density at high resolution or in neutron maps

- phenix.refine offers various options for handling H atoms at ***any*** resolution:

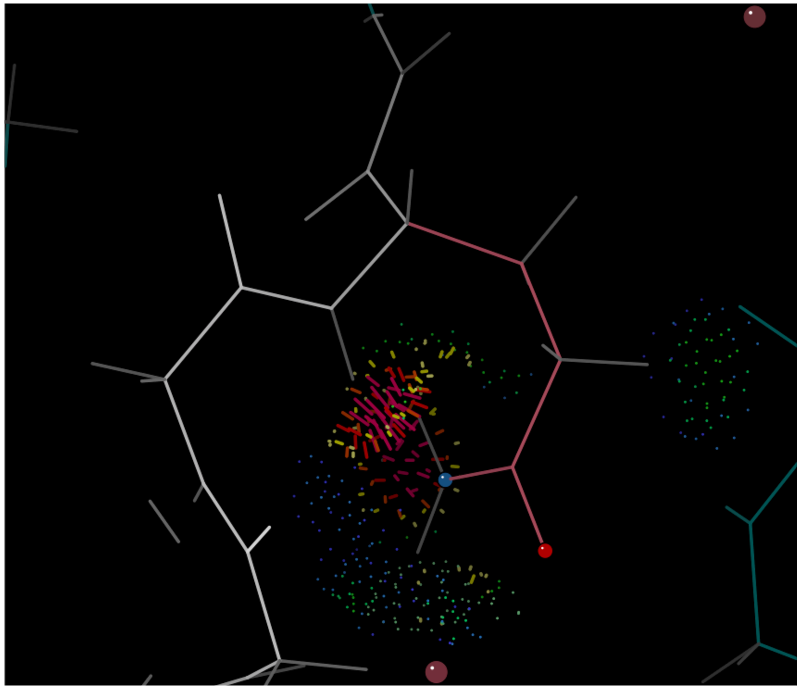
  - Riding model (low-high resolution)
  - Individual atoms (ultrahigh resolution or neutron data)
  - Account for scattering contribution or just use to improve the geometry

- Example: automatic re-refinement of 1000 PDB models with and without H:

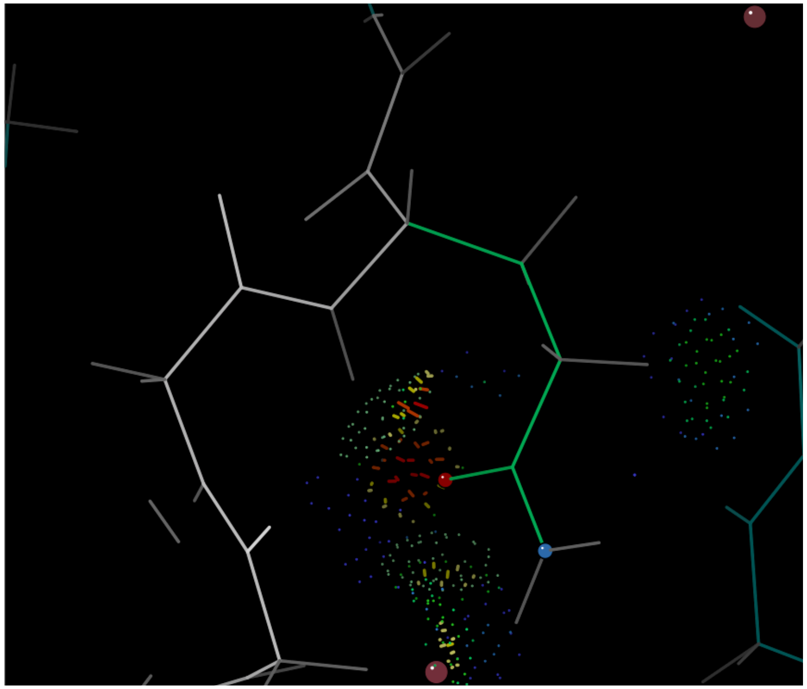
pdb	**resolution**	**Rfree(no H) – Rfree(with H)**
1akg	1.1	1.9
1byp	1.75	1.41
1dkp	2.3	0.93
1rgv	2.9	0.50

Option for automatic side chain flips to avoid clashes

- Apply side chain flips if necessary (Asn/Gln/His)



Bad

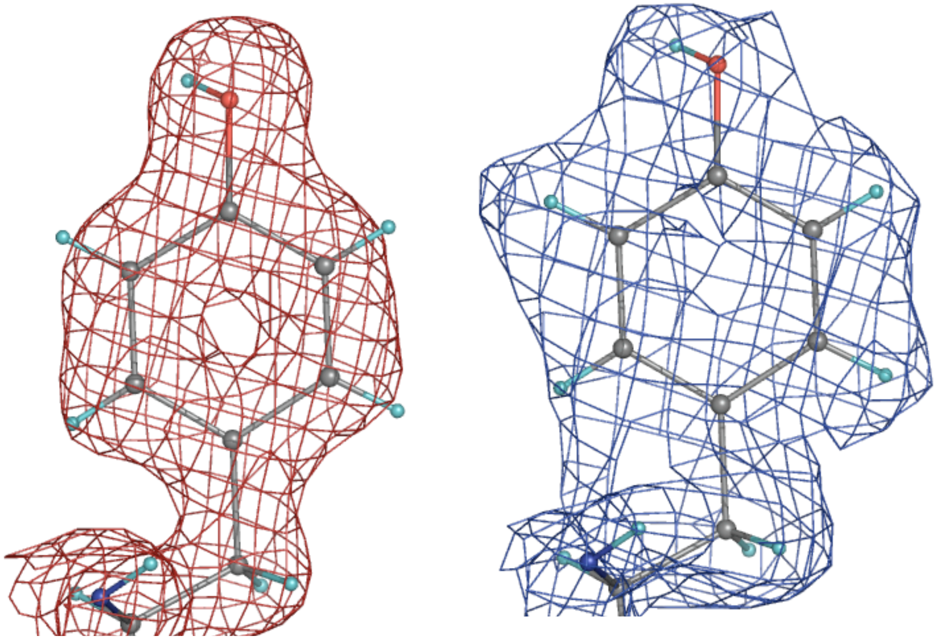


Good

Refinement using X-ray and Neutron diffraction data

2mFo-DFc maps

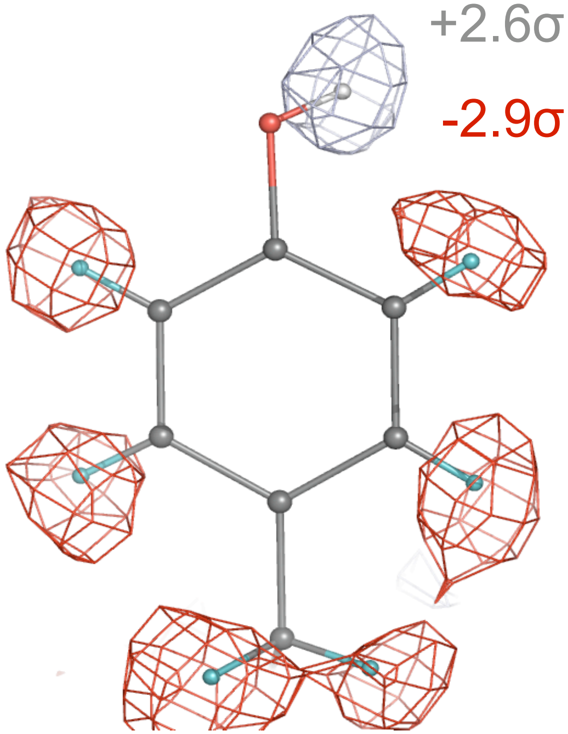
X-ray (1.8 Å) **Neutron (2.2 Å)**



Fo-Fc, (H-, D-omit **neutron map**), **1.6 Å** resolution

+2.6σ, D atoms

-2.9σ, H atoms



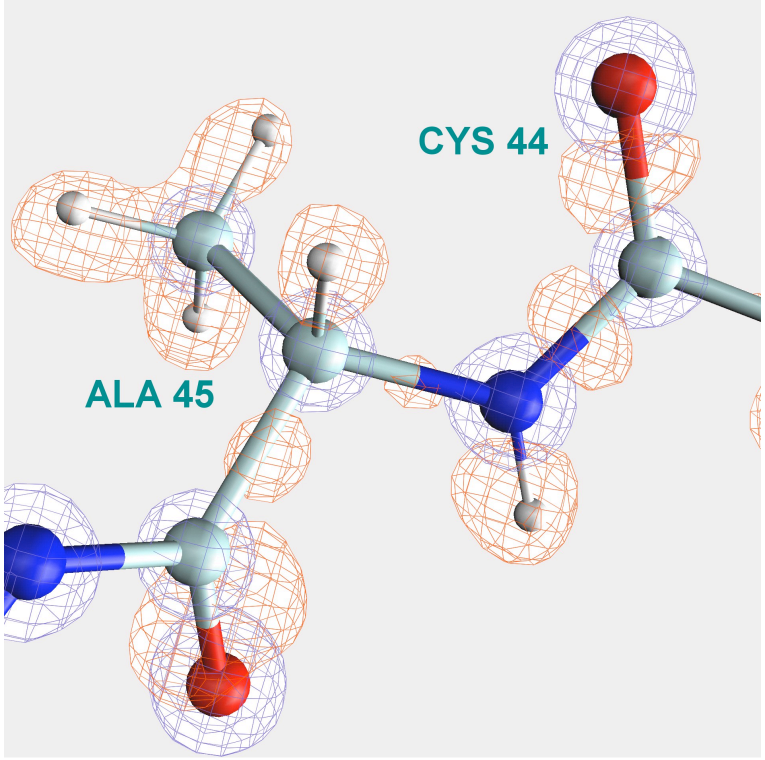
✓ Unlike typical resolution X-ray maps, neutron maps show hydrogen atoms

✓ *phenix.refine* can refine structures using neutron or both X-ray and neutron data simultaneously (Joint XN refinement)

Refinement at subatomic resolution

~340 structures in PDB at resolution higher than 1.0 Å

Aldose Reductase (0.66 Å resolution)



Fo-Fc (orange) 2Fo-Fc (blue)

✓ phenix.refine has unique set of tools to correctly refine such structures

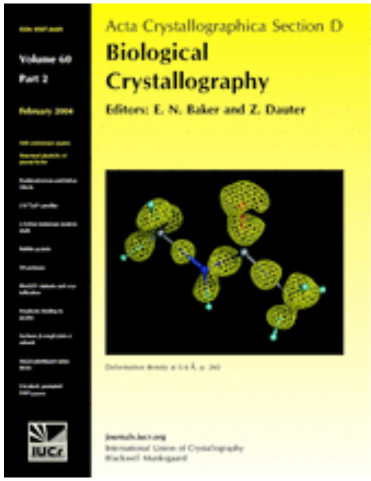
Modeling at subatomic resolution: IAS model

- Basics of IAS model:

  Afonine et al, Acta Cryst. D60 (2004)

- First practical examples of implementation and use in PHENIX:

  Afonine et al, Acta Cryst. D63, 1194-1197 (2007)

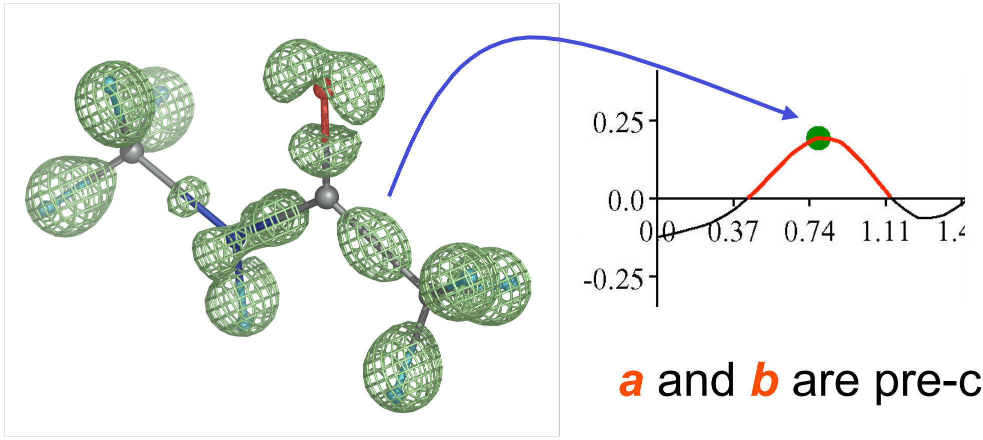


Acta Crystallographica Section D
Biological Crystallography
Editors: E. N. Baker and Z. Dauter

Volume 60
Part 2
February 2004

IUCr

IAS modeling in PHENIX



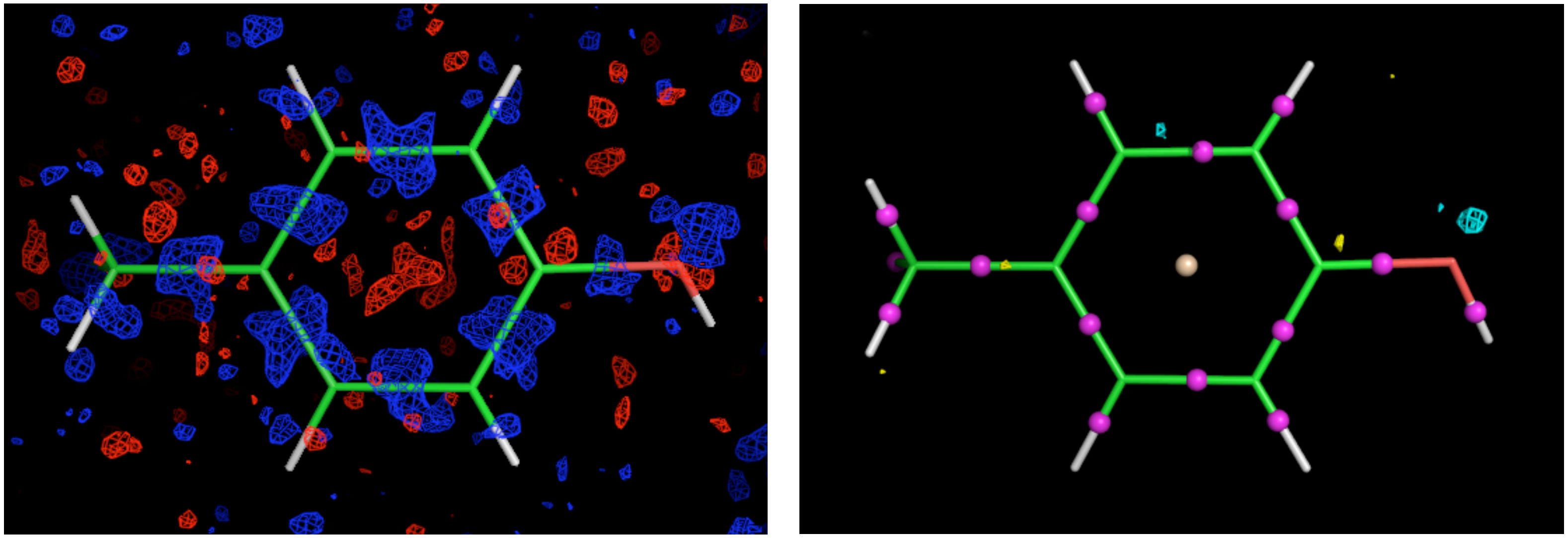
Simple Gaussian is good enough:

$$f_{bond\_scatterer}(\mathbf{s}) = \mathbf{a}\,\exp(\ \mathbf{b}\,\mathbf{s}^2)$$

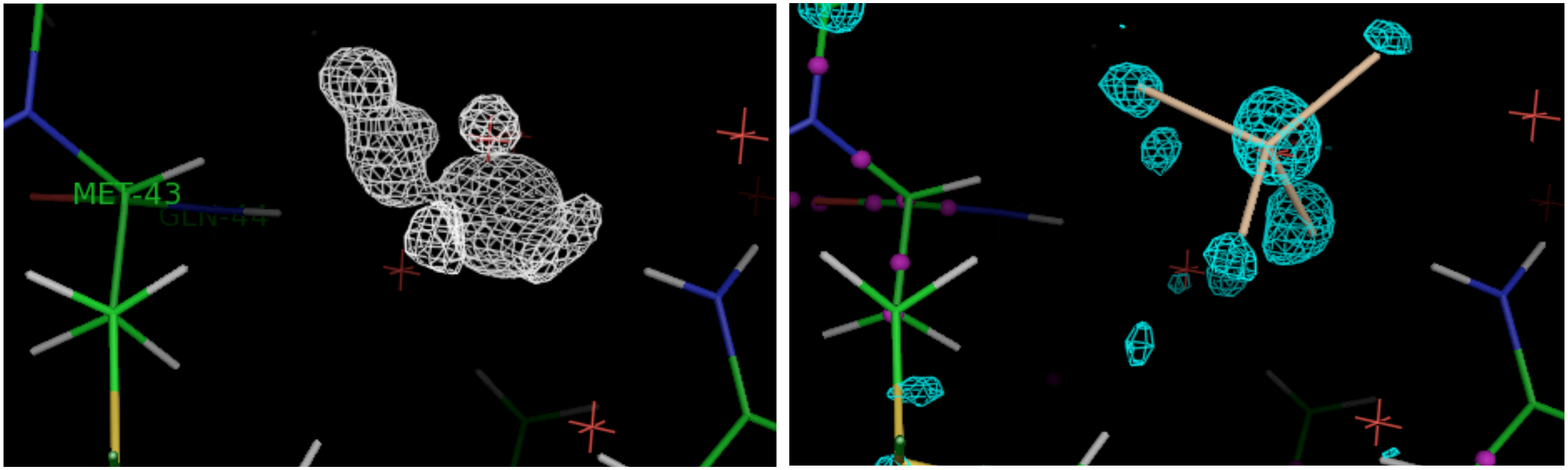
a and *b* are pre-computed library for most bond types

IAS modeling: benefits

- Improve maps: reduce noise. Before (left) and after (right) adding of IAS.



- Find new features: originally wrong water (left) replaced with SO4 ion (right) clearly suggested by improved map after adding IAS

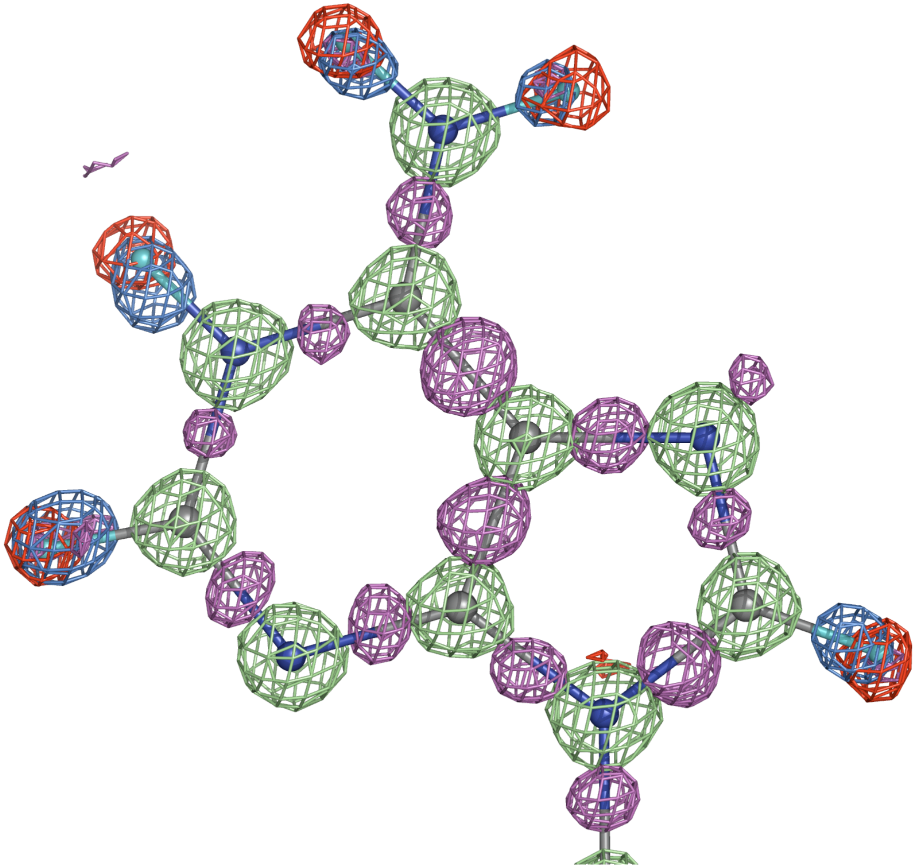


X-ray and Neutron Crystallography: Complimentary Methods

- Still complimentary even at subatomic resolution (NAD structure)

Neutron 2mFo-DFc map at 0.65 Å resolution, ±2.4σ, green (positive), red (negative)

X-ray mFo-DFc map at 0.6 Å resolution, blue: H omit, 5σ, magenta: 2.8σ all atoms included



Running phenix.refine (command line)

Model refinement

- Designed to be very easy to use

```
phenix.refine model.pdb data.hkl [parameters]
```

Some basic examples of running phenix.refine from the command line

- Refinement of individual coordinates, B-factors, and occupancies for some atoms:

```
phenix.refine model.pdb data.hkl
```

- Add water picking and Simulated Annealing to default run above:

```
phenix.refine model.pdb data.hkl simulated_annealing=true
ordered_solvent=true
```

- Refinement of individual coordinates and B-factors using neutron data:

```
phenix.refine model.pdb data.hkl
main.scattering_dictionary=neutron
```

- To see all parameters (more than 300):

```
phenix.refine --show_defaults=all
```

Running phenix.refine

```
% phenix.refine model.pdb data.hkl parameter_file
```

where **parameter_file** contains following lines:

```
refinement.main {
    high_resolution = 2.0
    low_resolution = 15.0
    simulated_annealing = True
    ordered_solvent = True
    number_of_macro_cycles = 5
}
refinement.refine.adp {
     tls = chain A
     tls = chain B
}
```

For typing enthusiasts, the equivalent command line run would be:

```
% phenix.refine model.pdb data.hkl xray_data.high_resolution=2
xray_data.low_resolution=15 simmulated_annealing=true
ordered_solvent=True adp.tls=“chain A” adp.tls=“chain B”
main.number_of_macro_cycles=5
```

Typical way of phenix.refine run from the command line

1. Get the file with all parameters:

```
% phenix.refine --show-defaults=all > parameter_file
```

2. Edit the file `parameter_file`:
   - Remove all parameters that you are not planning to change (make sure to have all { } matched)
   - Change the rest of parameters

3. Run phenix.refine as following:

```
% phenix.refine model.pdb data.hkl parameter_file
```

or (If `model.pdb` and `data.hkl` are included into `parameter_file` file)

```
% phenix.refine parameter_file
```

Useful tip: to compare the set of parameters in your `parameter_file` file against the set of all default parameters, type:

```
% phenix.refine --diff-params parameter_file
```

Some refinement runs require two steps: hydrogens and ligands

- When running:

```
% phenix.refine model.pdb data.hkl
```

each item in model.pdb is matched against the CCP4 Monomer Library to extract the topology and parameters and to automatically build corresponding restraints.

- If model.pdb contains an item not available in CCP4 Monomer Library, e.g. a novel ligand, use ReadySet! program to generate topology and parameter definitions for refinement:

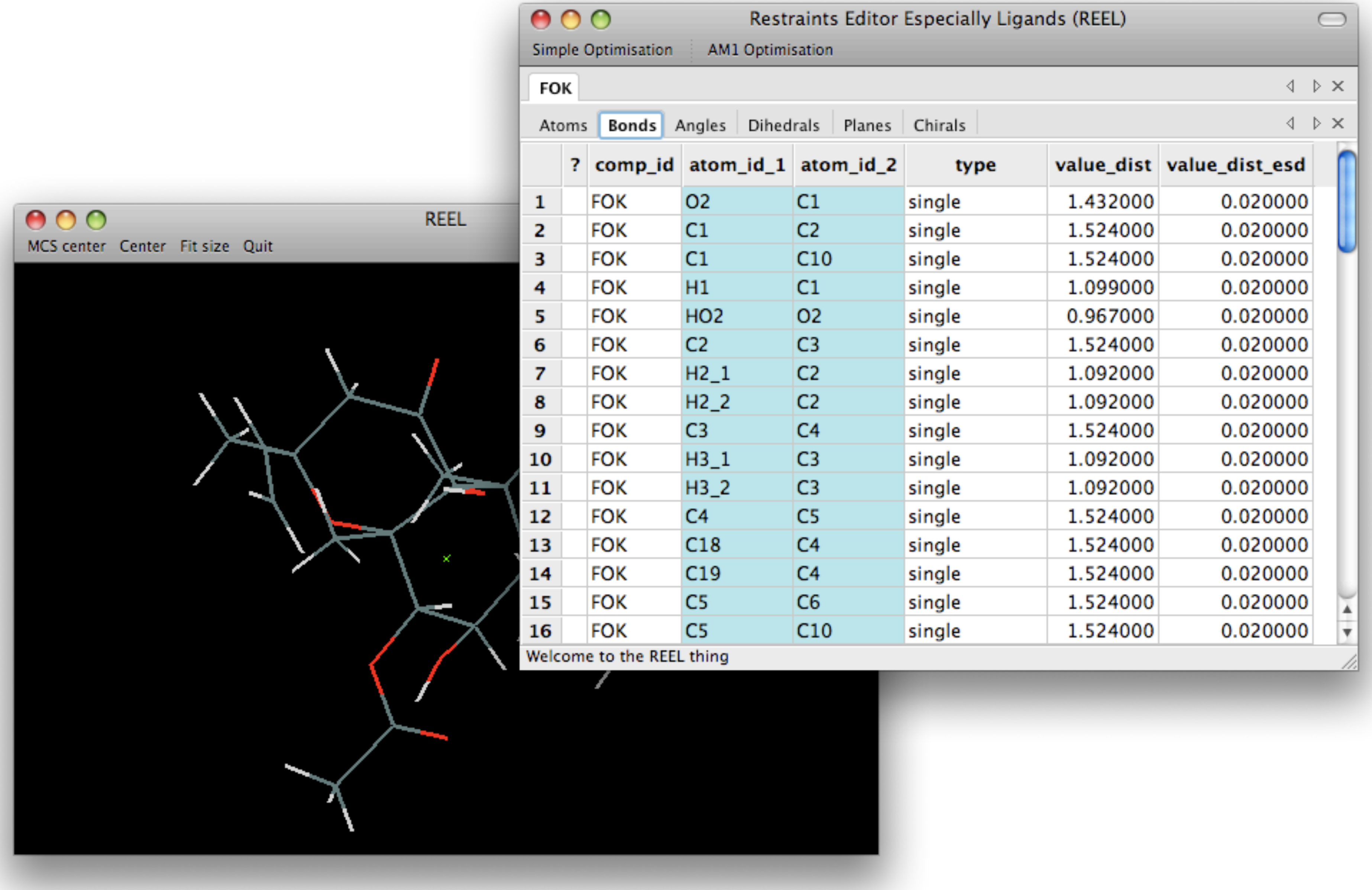
```
% phenix.ready_set model.pdb
```

This will produce the file LIG.cif and updated PDB file `model.updated.pdb` with all H atoms added which can be used for refinement:

```
% phenix.refine model.pdb data.hkl LIG.cif
```

Restraints and novel ligands: REEL

```
% phenix.reel LIG.cif
```



	?	comp_id	atom_id_1	atom_id_2	type	value_dist	value_dist_esd
1		FOK	O2	C1	single	1.432000	0.020000
2		FOK	C1	C2	single	1.524000	0.020000
3		FOK	C1	C10	single	1.524000	0.020000
4		FOK	H1	C1	single	1.099000	0.020000
5		FOK	HO2	O2	single	0.967000	0.020000
6		FOK	C2	C3	single	1.524000	0.020000
7		FOK	H2_1	C2	single	1.092000	0.020000
8		FOK	H2_2	C2	single	1.092000	0.020000
9		FOK	C3	C4	single	1.524000	0.020000
10		FOK	H3_1	C3	single	1.092000	0.020000
11		FOK	H3_2	C3	single	1.092000	0.020000
12		FOK	C4	C5	single	1.524000	0.020000
13		FOK	C18	C4	single	1.524000	0.020000
14		FOK	C19	C4	single	1.524000	0.020000
15		FOK	C5	C6	single	1.524000	0.020000
16		FOK	C5	C10	single	1.524000	0.020000

Welcome to the REEL thing

Some refinement runs require two steps: twinning

- Two steps to perform twin refinement:

  - run *phenix.xtriage* to get twin operator (twin law):

```
% phenix.xtriage data.mtz
```

  - run *phenix.refine*:

```
% phenix.refine model.pdb data.mtz twin_law="-h-k,k,-l"
```

Model refinement - output

- Input command

```
phenix.refine model.pdb data.mtz [parameters]
```

- Output files

`model_refine_001.eff`	summary of all input parameters
`model_refine_001.geo`	summary of all restraints used
`model_refine_001.log`	complete information about refinement
`model_refine_001.pdb`	refined structure
`model_refine_001_map_coeffs.mtz`	Fourier map coefficients
`model_refine_002.def`	parameters for the next run

If data file is not in MTZ format, or there are multiple data files at input (example: one with Fobs and the other one with free-R flags), then phenix.refine will combine them into one MTZ data file called: `model_data.mtz` and this file should be used in all subsequent runs.

Example of a complex refinement run

- Do the following:
  - refine individual coordinates for all atoms using minimization and Simulated Annealing
  - refine coordinates of three rigid body groups:
    - chain A
    - chain B and chain C
    - chain D
  - individual anisotropic ADP for all Uranium atoms
  - individual isotropic ADP for all other atoms
  - three TLS groups:
    - atoms in residues from 1 to 300 of chain A and whole chain B
    - atoms from 301 to 500 in chain A
    - whole chain D
  - update water during refinement
  - use NCS in refinement
  - output everything into a files with prefix *test*

```
% phenix.refine model.pdb data.hkl parameters.eff
```

where **`parameters.eff`** contains following lines: see next slide…

Example of a complex parameter file

```
refinement {
  output {
    prefix = test
  }
  refine {
    strategy=*individual_sites individual_sites_real_space *rigid_body \
             *individual_adp group_adp *tls *occupancies group_anomalous
    sites {
      rigid_body = chain A
      rigid_body = chain B or chain C
      rigid_body = chain D
    }
    adp {
      individual {
        isotropic = not (element U)
        anisotropic = element U
      }
      tls = chain A and resseq 1:300 or chain B
      tls = chain A and resseq 301:500
      tls = chain D
    }}
  main {
    simulated_annealing = True
    ordered_solvent = True
    ncs = True
}}
```