Automated Crystallographic Structure Solution in PHENIX

Pavel Afonine

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

The PHENIX project

- **PHENIX is a (relatively) new package for automated structure solution** that incorporates handling of both: X-ray and neutron data
- PHENIX is not a pipe-line made of existing programs, but a highly integrated software
- Library based development (Python, C++) and new or re-designed algorithms
- Designed to be used by both novices and experienced users
- Long-term development and support
- Large experience of crystallographic software and methods development (previous experience of CNS development – Paul Adams and Ralf Grosse-Kunsteve)

PHENIX

Lawrence Berkeley National Laboratory

Paul Adams, Pavel Afonine, Nat Echols, Jeff Headd, Ralf Grosse-Kunstleve, Nigel Moriarty, Nicholas Sauter, Peter Zwart

Los Alamos National Laboratory

Tom Terwilliger, Li-Wei Hung

Cambridge University

Randy Read, Airlie McCoy, Laurent Storoni, Gabor Bunkoczi, Robert Oeffner

Duke University

Jane Richardson & David Richardson, Ian Davis, Vincent Chen, Jeff Headd, Chris Williams, Bryan Arendall, Laura Murray

Funding

- NIH/NIGMS: *P01GM063210, P50GM062412, P01GM064692, R01GM071939*
- Lawrence Berkeley Laboratory
- PHENIX Industrial Consortium

Paul Adams – project director

PHENIX and Neutron crystallography

Macromolecular Neutron Crystallography Consortium (MNC)

Los Alamos National Lab
Paul Langan, Marat Mustyakimov, Benno Schoenborn

Lawrence Berkeley National Lab (LBNL)
Paul Adams, Pavel Afonine

http://mnc.lanl.gov/

Some milestones (publications)

- 2002:

  Grosse-Kunstleve RW et al. **The Computational Crystallography Toolbox: crystallographic algorithms in a reusable software framework.**

  Grosse-Kunstleve RW, Adams PD. **On the handling of atomic anisotropic displacement parameters.**

  Adams PD et al. **PHENIX: building new software for automated crystallographic structure determination.**

- 2003:

  Grosse-Kunstleve RW et al. **Substructure search procedures for macromolecular structures.**

  Abrahams D et al. **Building Hybrid Systems with Boost.Python**.

- 2004:

  Adams PD et al. **Recent developments in the PHENIX software for automated crystallographic structure determination**.

  Grosse-Kunstleve RW et al. **cctbx news: Geometry restraints and other new features.**

- 2005:

  McCoy AJ et al. **Likelihood-enhanced fast translation functions**

  Afonine PV et al. **A robust bulk-solvent correction and anisotropic scaling procedure**

  Afonine PV et al. **The Phenix refinement framework**.

  Zwart PH et al. **Xtriage and Fest: automatic assessment of X-ray data and substructure structure factor estimation**.

Some milestones (publications)

- 2007:

  Terwilliger TC et al. **Interpretation of ensembles created by multiple iterative rebuilding of macromolecular models**.

  McCoy AJ et al. **Phaser crystallographic software**.

  Afonine PV et al. **On macromolecular refinement at subatomic resolution with interatomic scatterers**.

- 2008

  Terwilliger TC et al. **Iterative model building, structure refinement and density modification with the PHENIX AutoBuild wizard**

  Terwilliger TC et al. **Iterative-build OMIT maps: map improvement by iterative model building and refinement without model bias**.

- 2009:

  Grosse-Kunstleve RW et al. **Experience converting a large Fortran-77 program to C++**

  Grosse-Kunstleve RW et al. **Torsion Angle Refinement and Dynamics as a Tool to Aid Crystallographic Structure Determination**

  Moriarty NW et al. **electronic Ligand Builder and Optimization Workbench (eLBOW): a tool for ligand coordinate and restraint generation**

  Afonine PV et al. **Automatic multiple-zone rigid-body refinement with a large convergence radius**.

  Terwilliger TC et al. **Decision-making in structure solution using Bayesian estimates of map quality: the PHENIX AutoSol wizard**.

  Urzhumtseva L. et al. **Crystallographic model quality at a glance**.

Some milestone publications

- 2010

  Afonine PV et al. **Joint X-ray and neutron refinement with phenix.refine**

  Grosse-Kunstleve RW et al. **cctbx PDB handling tools**

  Afonine PV et al. **phenix.model_vs_data: a high-level tool for the calculation of crystallographic model and data statistics**

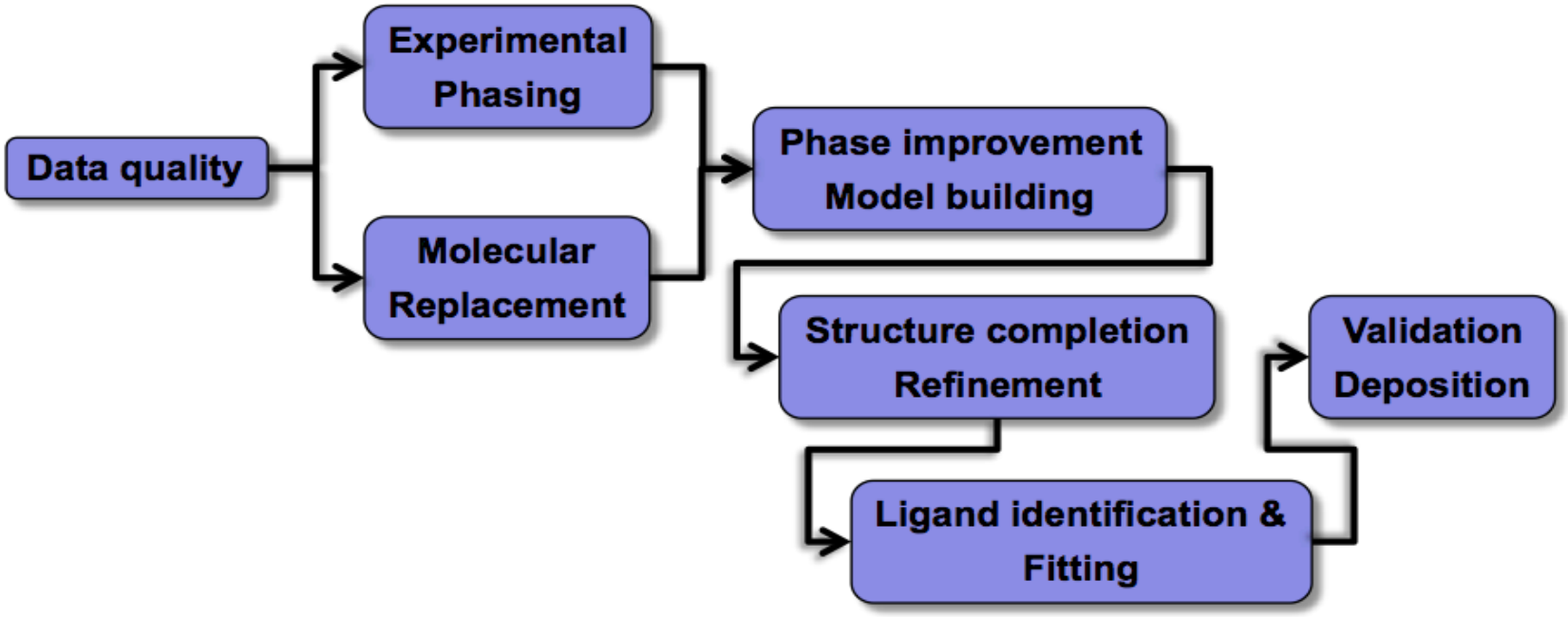
  Adams PD et al. **PHENIX: a comprehensive Python-based system for macromolecular structure solution**

  Afonine PV et al. **Atomic Displacement Parameters (ADPs), their parameterization and refinement in PHENIX**. Computational Crystallography Newsletter. 1

- 2011

  Urzhumtsev A, et al. **TLS for dummies**. Computational Crystallography Newsletter. 2.

Why Automation ?



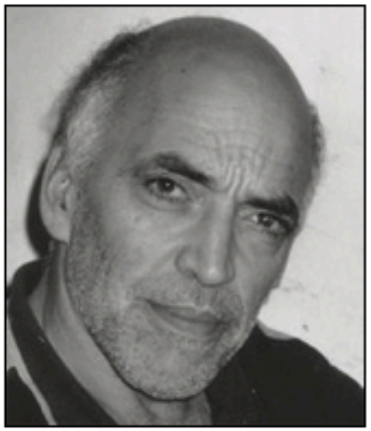
- ✓ Automation can increase efficiency, and reduce human error (especially for non-expert crystallographers)

Why Automation ?

- Makes difficult cases more feasible for experts
- Routine structure solution cases are accessible to a wider group of structural biologists
- Software can try more possibilities than we are typically willing to bother with
- Multiple trials or use of different parameters can be used to estimate uncertainties
- If a task is modular and automated, you can run it many times…
  - … checking different space groups, datasets to use
  - … checking if your model is biasing the map
  - … checking if you always get the same model

- What is required:
  - Software carrying out individual steps
  - Seamless connection between steps
  - A way to decide what is good
  - Strategies for structure determination and decision-making

Idea of automation is not new



"In the field of macromolecular structure determination, AMoRe was the first programme aiming at solving a crystal structure in an automated way."

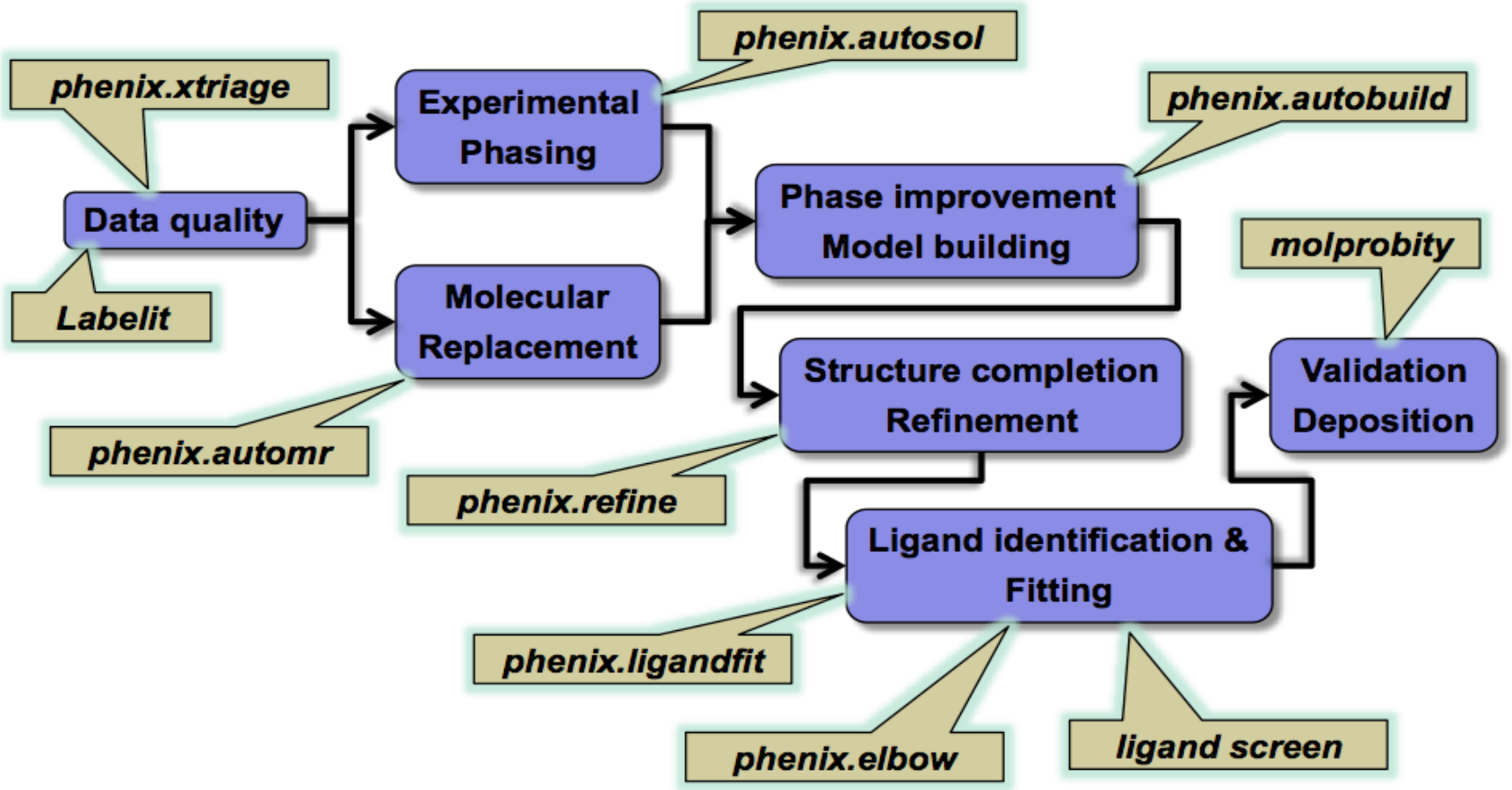
J.Navaza, Acta Cryst. A50, 157-163 (1994)

Source:

http://www.in-cites.com/papers/JorgeNavaza.html

PHENIX: principal tools

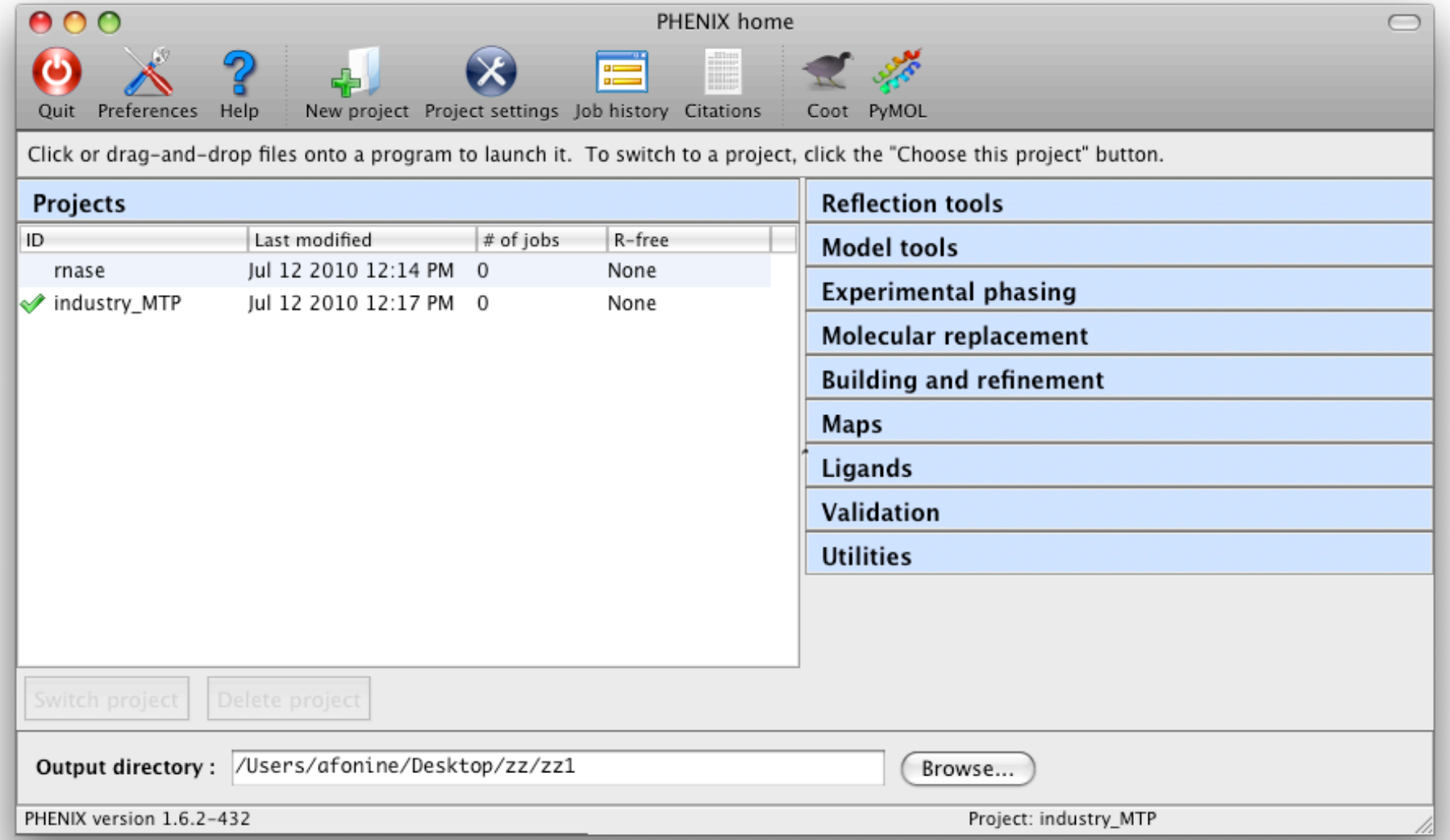
Complete set of tools for crystallographic structure determination: from experimental data to PDB deposited structure



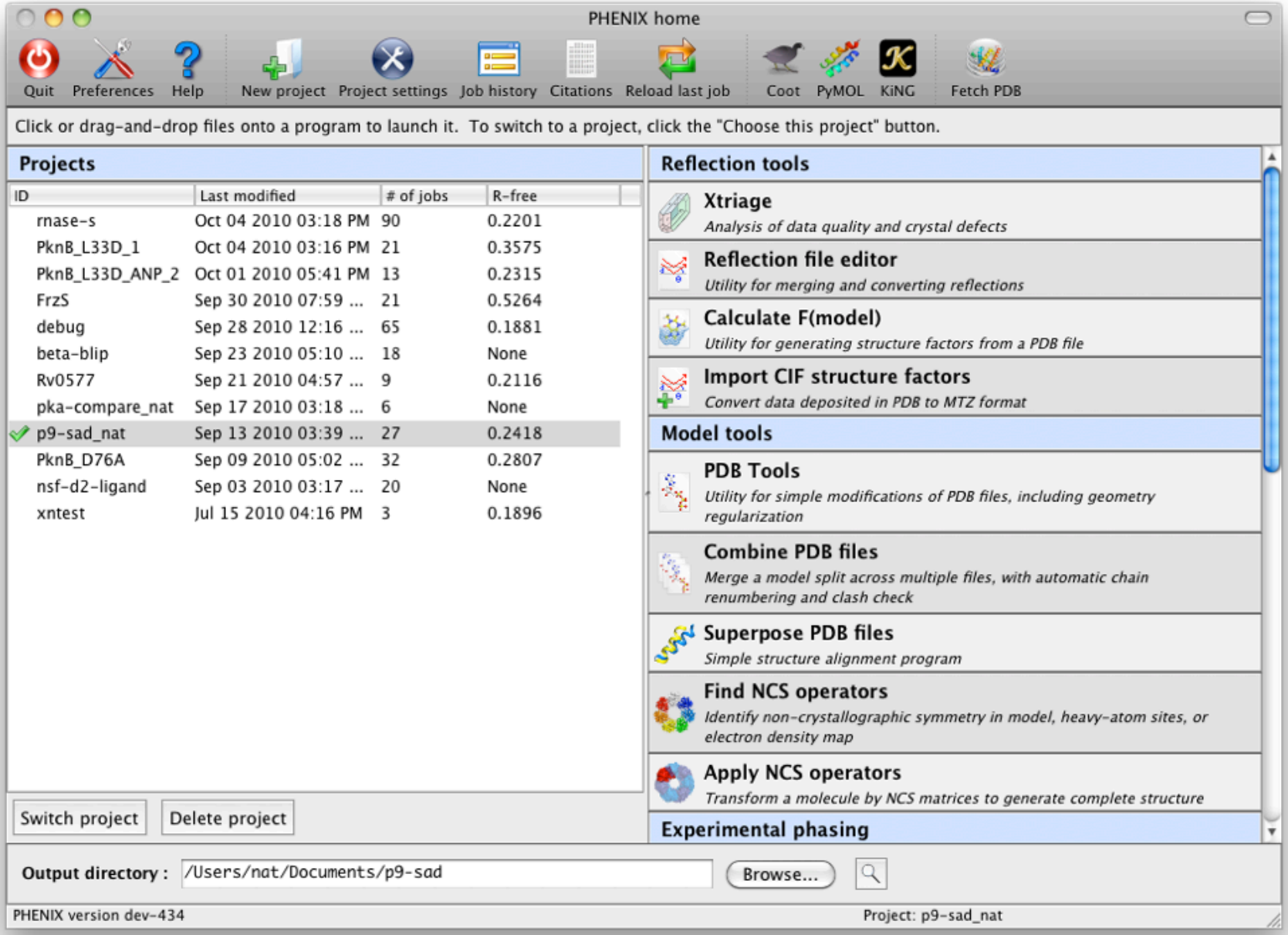
PHENIX: principal tools

- PHENIX programs can be run using:
  - **GUI**: easy for beginners, guided process - less chance of errors
  - **Command line**: convenient for scripting of multiple and large scale tasks
  - Expert developers can run certain tools from their own programs

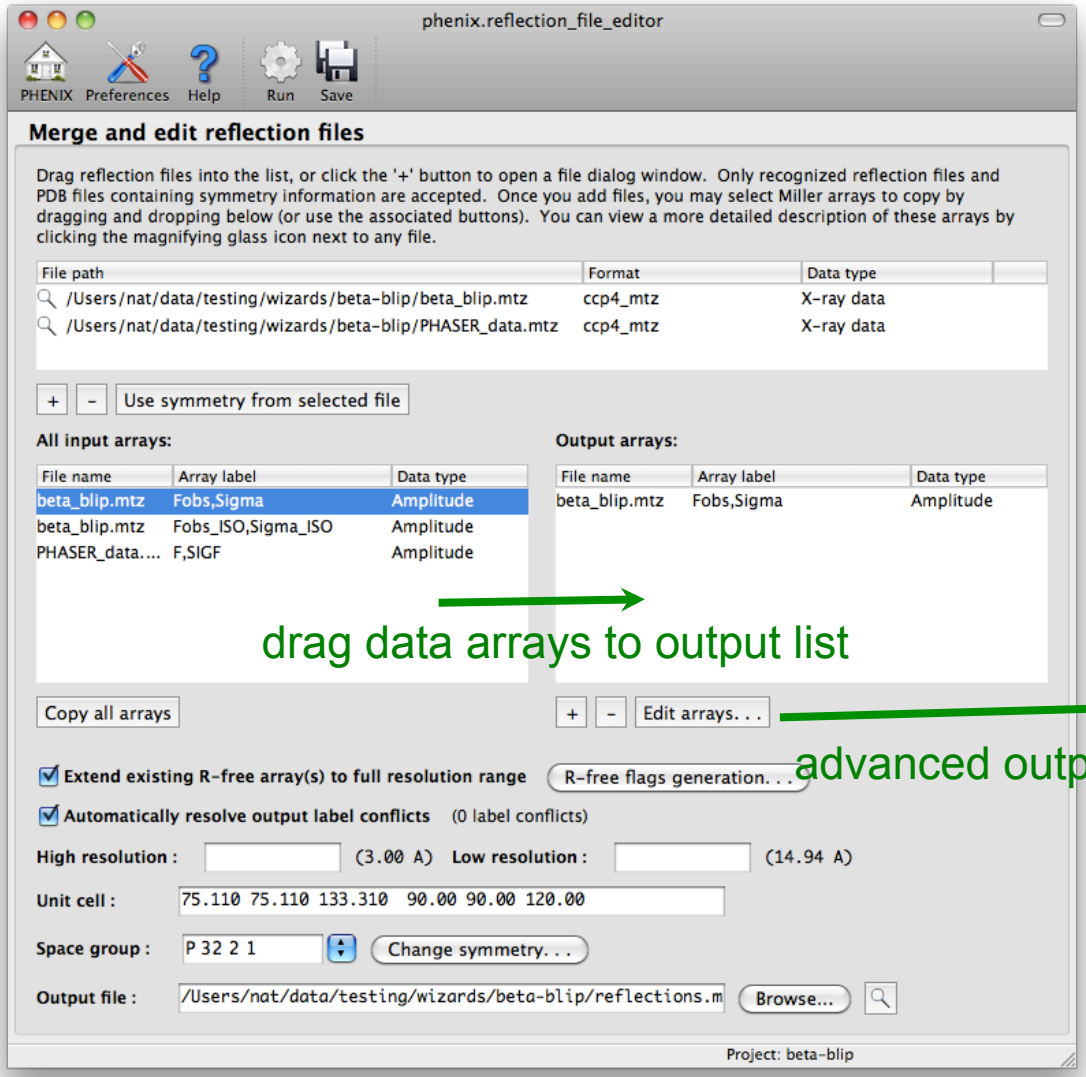
GUI examples

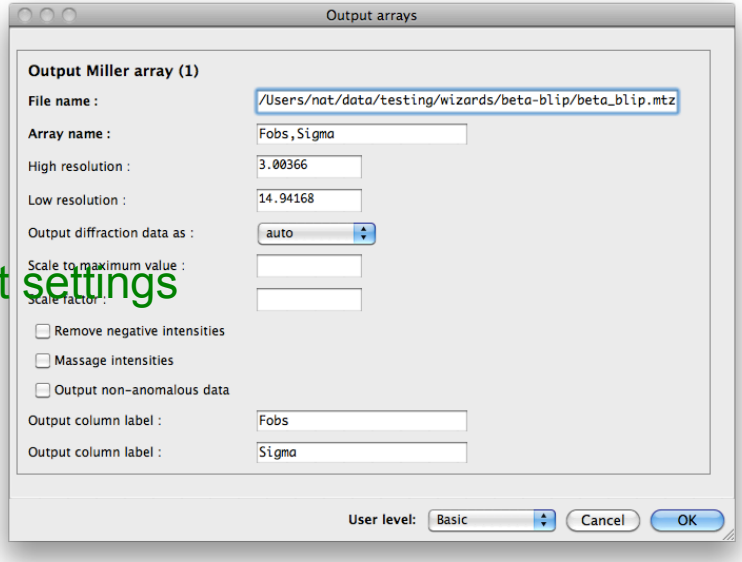


GUI examples



GUI examples: Reflection file editor



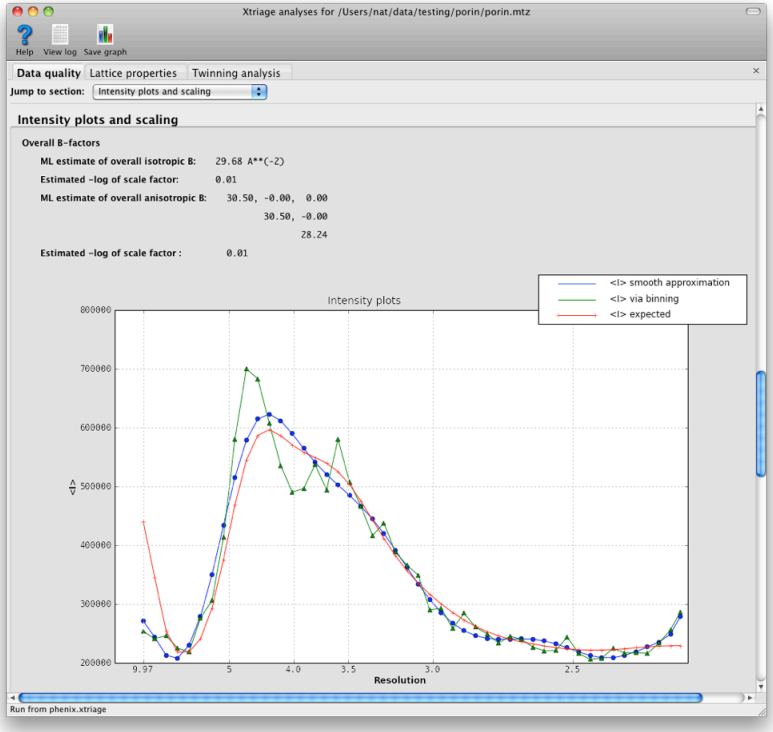


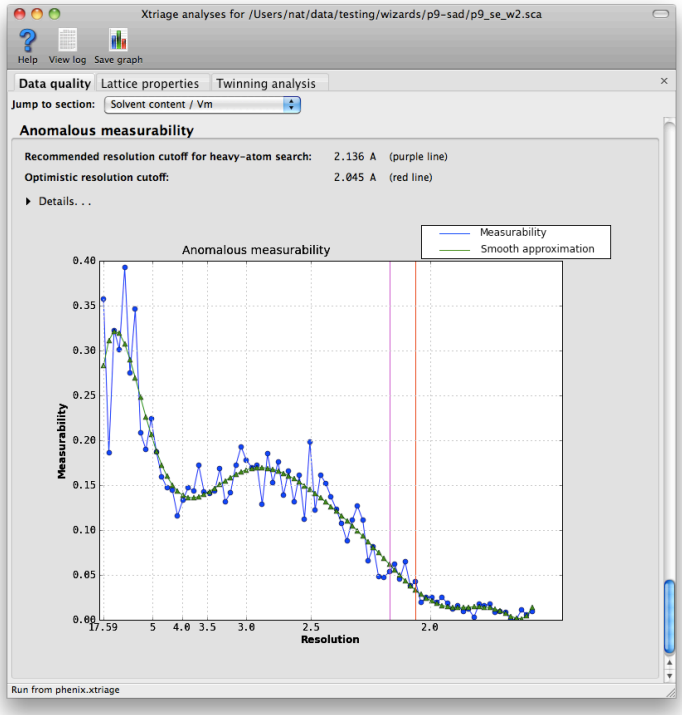
Combine and manipulate reflection files in any format, output as MTZ.

Extend old R-free sets, and generates new sets as thin shells (for refinement in presence of NCS).

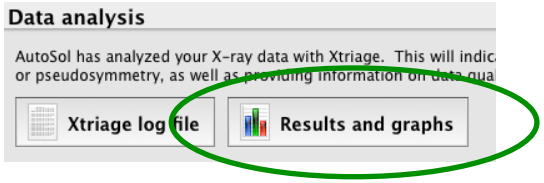
Data analysis with *phenix.xtriage*

- Analysis of signal-to-noise, data quality, Wilson plot, translational NCS, twinning, symmetry issues, and more





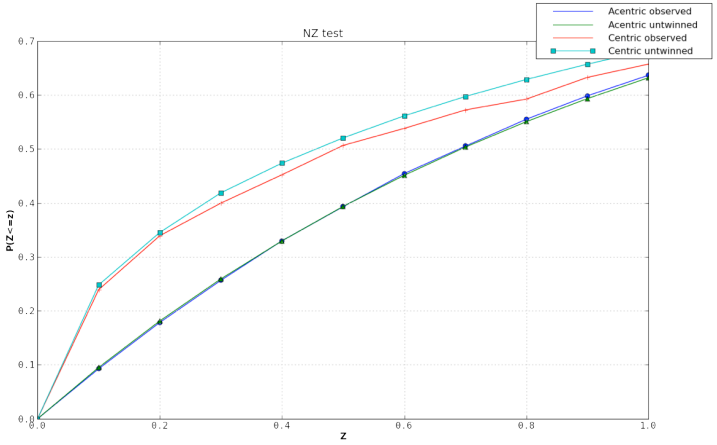
AutoSol and AutoBuild run Xtriage almost immediately, and results can be viewed from those GUIs. However, it may save time and effort to run Xtriage yourself first.

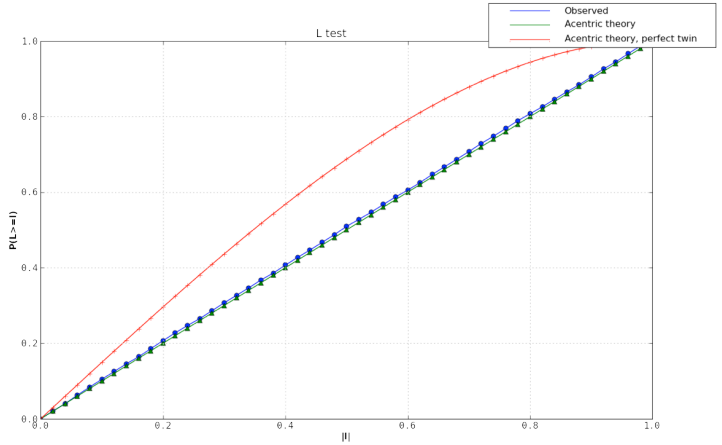


Identifying twinned structures in Xtriage

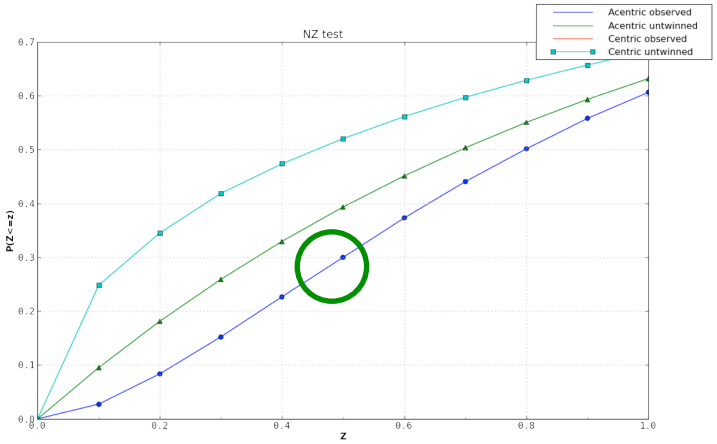
- Twinning changes the distribution of intensity values in predictable ways

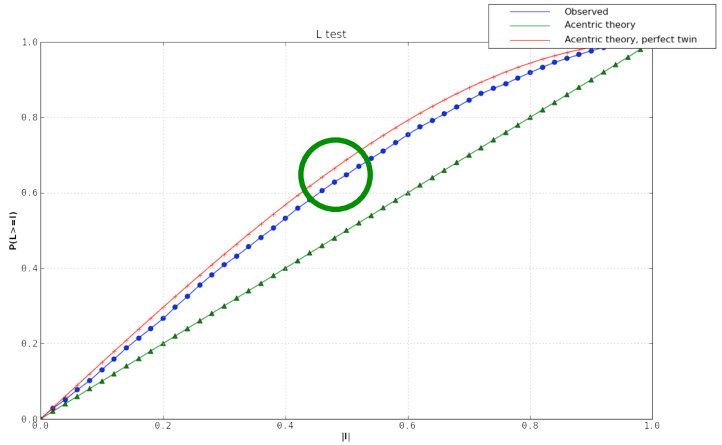
Good data (p9-sad example): observed intensity distributions are close to expected values





Twinned data (porin-twin example): NZ test curve is sigmoidal, L test curve is shifted upwards

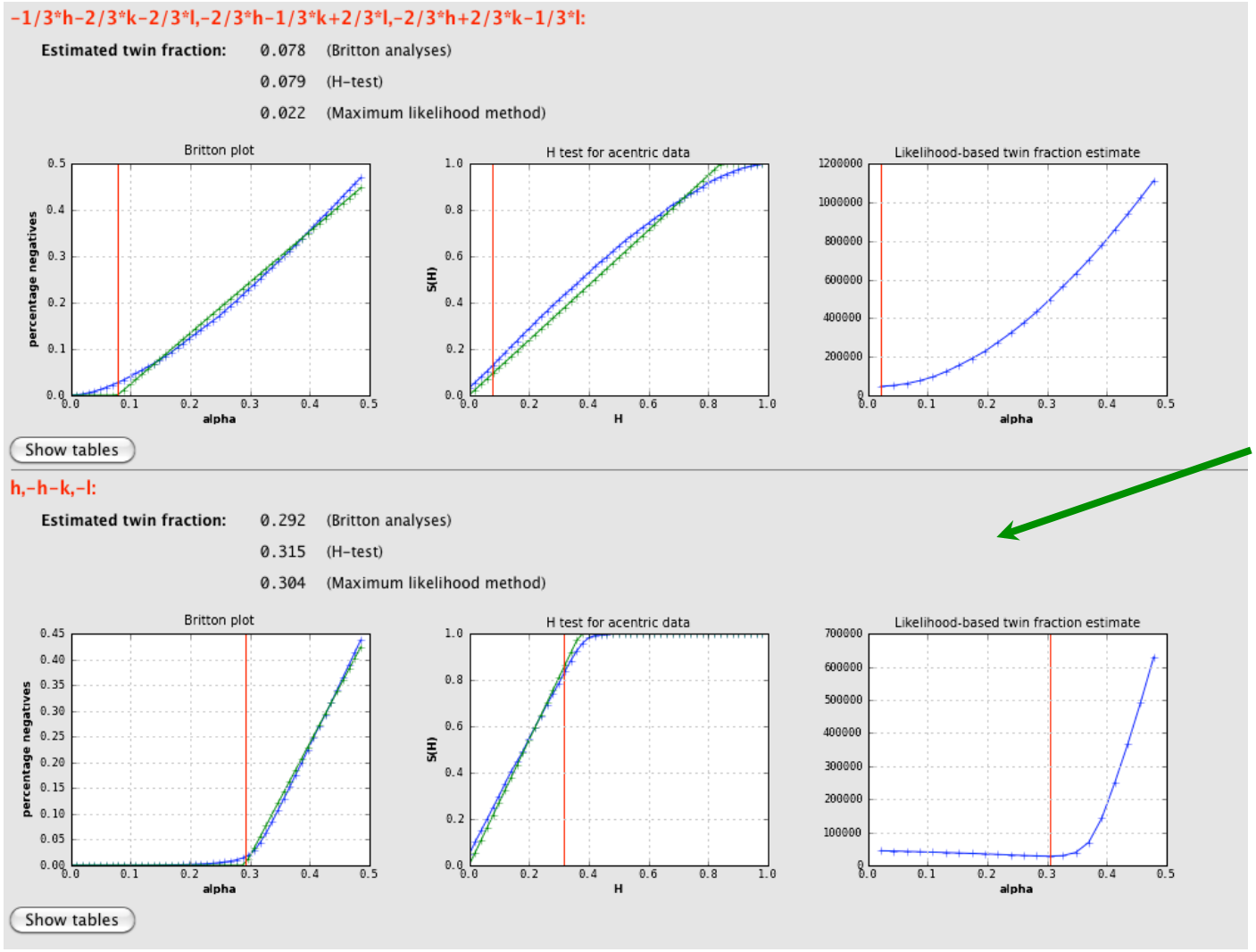




Intensity distributions can also be affected by pseudotranslation (especially NZ test); make sure you look at all of the evidence for twinning!

Identifying twinned structures in Xtriage

- The twin fraction for all possible twin laws will be estimated; usually one of these is obviously different



Two twin laws from the porin-twin example are shown; in this case *h,-h-k,-l* is the actual twin law for this crystal. This can be used in phenix.refine, which will determine the true twin fraction based on the refined model.

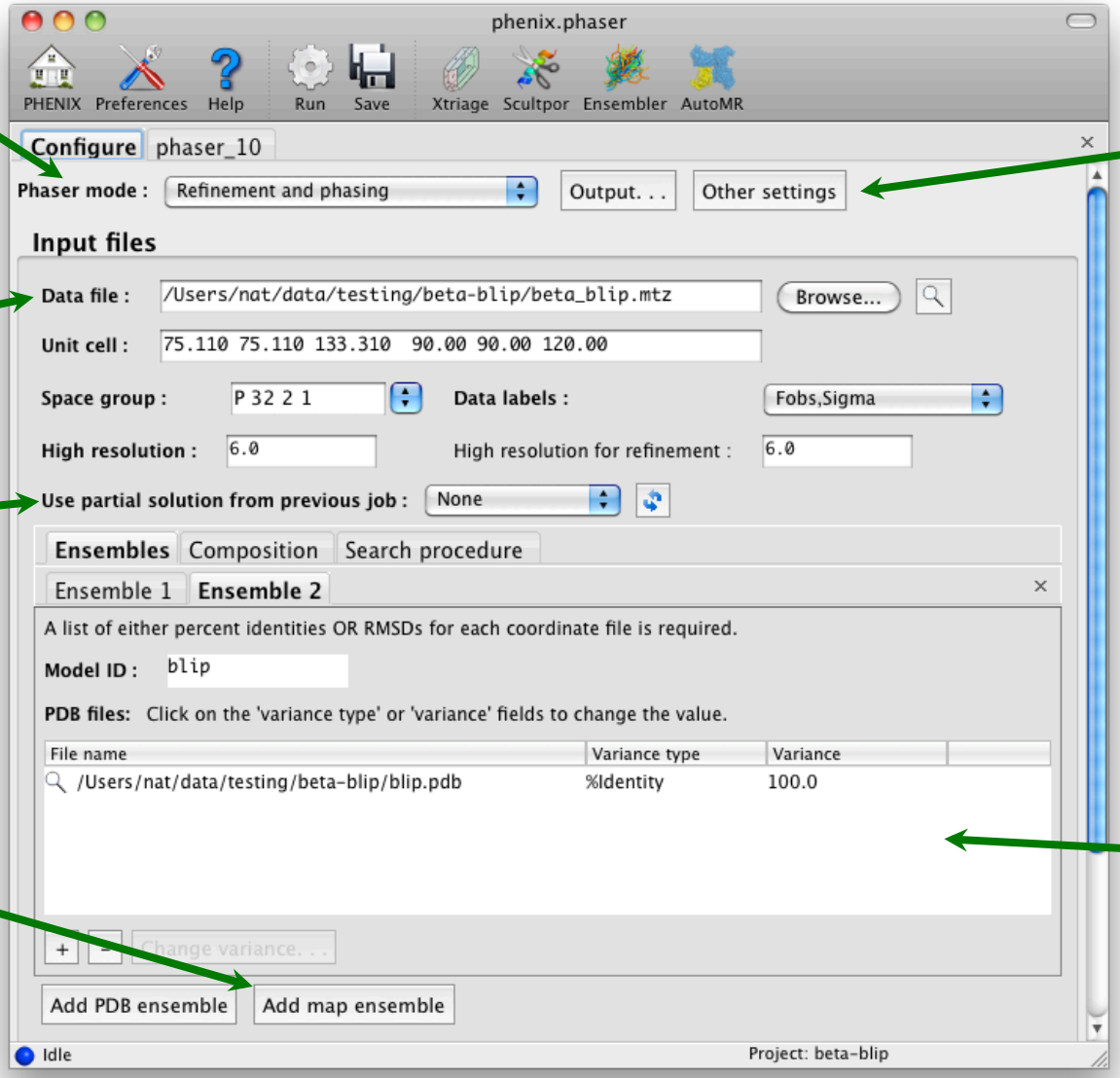
GUI examples: PHASER

Supports modes MR_AUTO, MR_FRF, MR_BRF, MR_FTF, MR_BTF, MR_PAK, MR_RNP

Any reflection file format permitted

One-click re-use of partial solutions from past runs

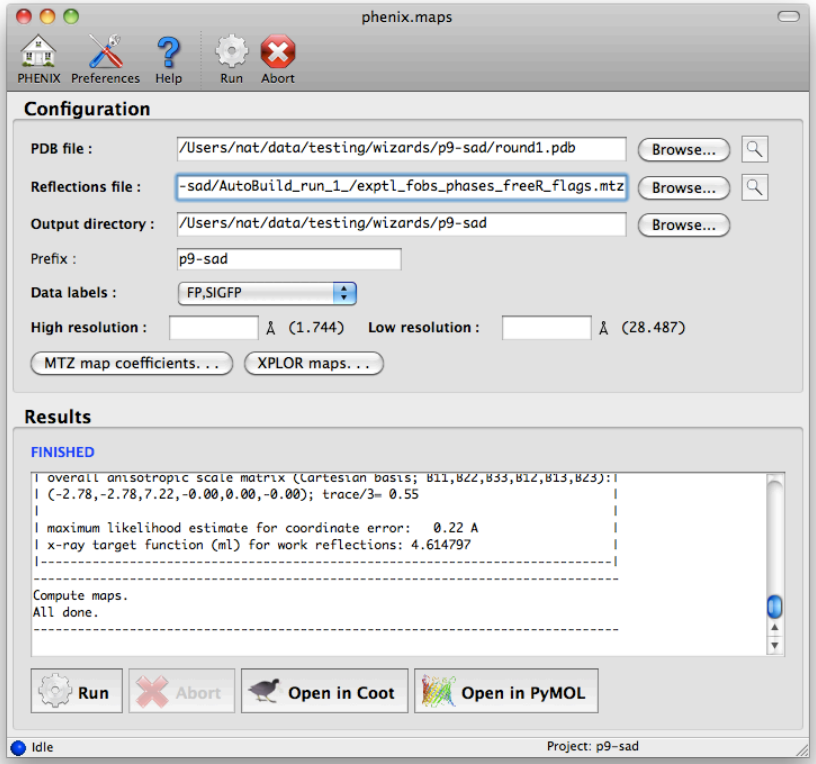
Can use a low-resolution map as a search model

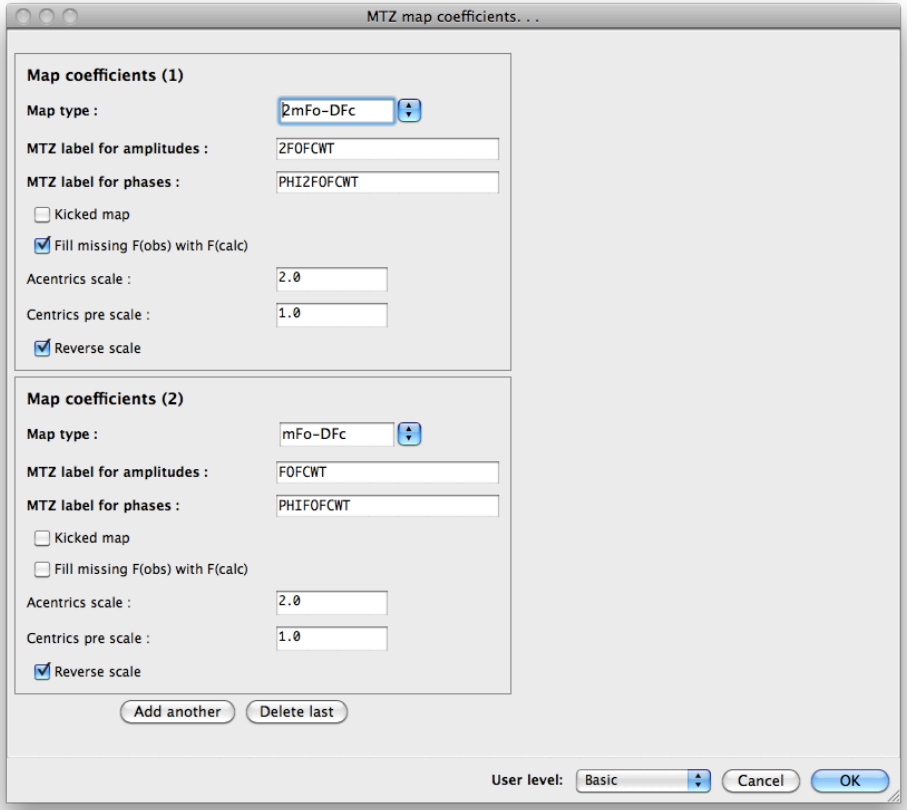


Most keywords found here

Drag-and-drop supported

GUI examples: phenix.maps

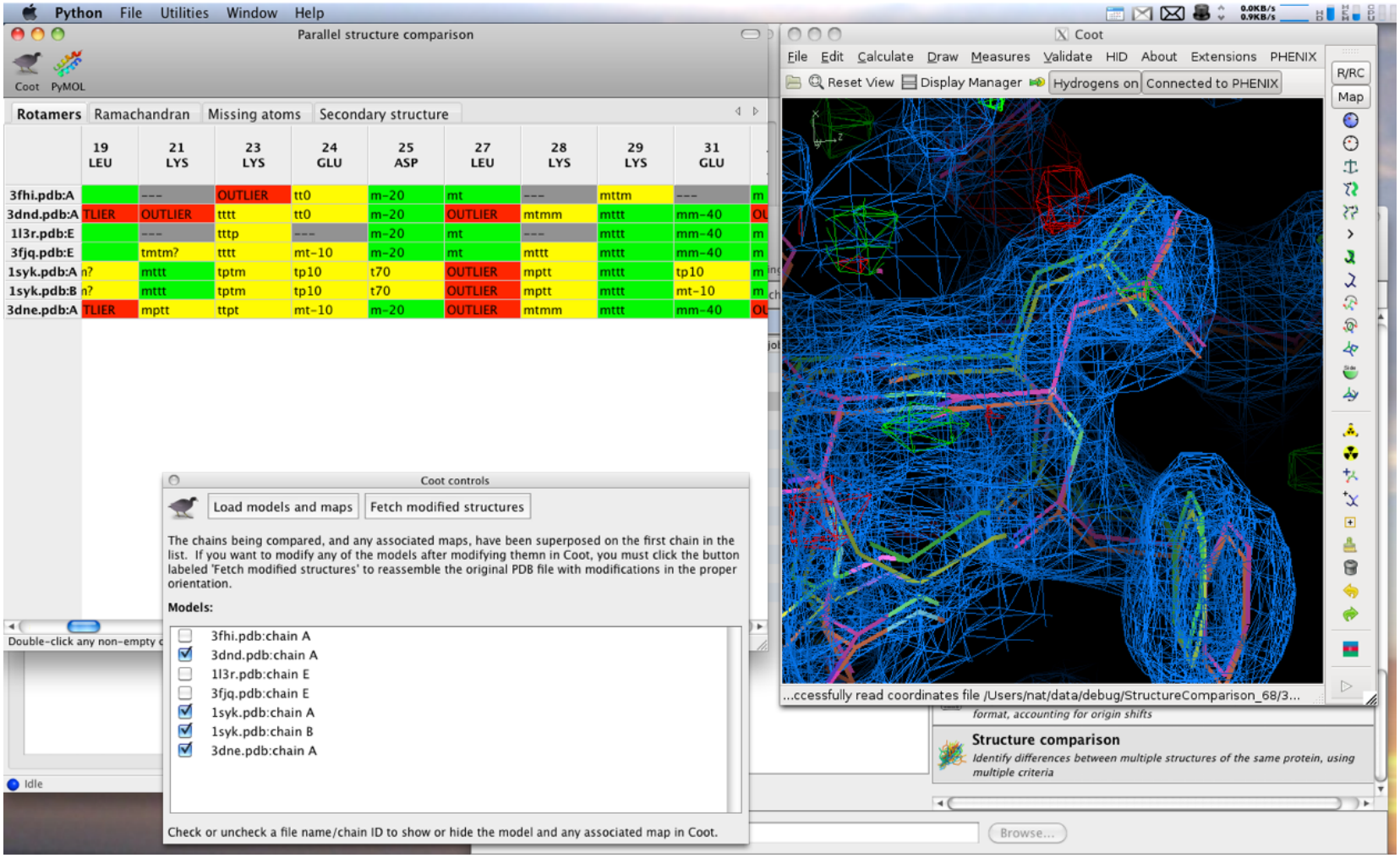




- Any kind of map defined as [p][m]Fo+[q][D]Fc (Example: 3.2Fo-1.7Fc)
- "kicked" map: removes bias by averaging maps calculated with shaken coordinates
- Fill missing F(obs) with F(calc): often improves 2mFo-DFc maps, but watch out for bias! (phenix.refine and Refmac both do this)
- B-factor sharpened maps (with automatic Bsharp determination)
- Anomalous difference maps

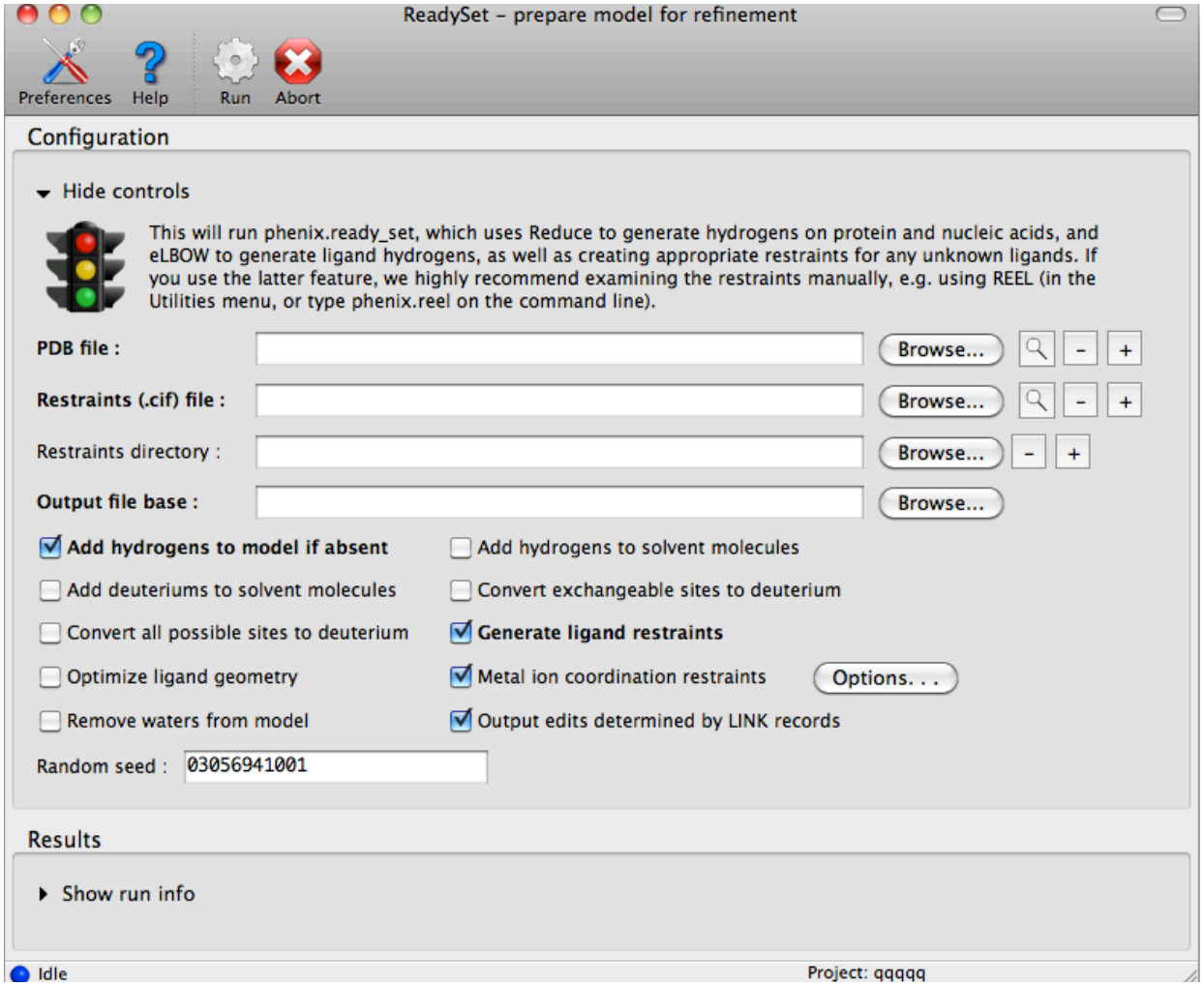
GUI examples: Parallel Structure Validation/Comparison

- Identifies points of difference between structures of the same protein, with optional map superpositioning



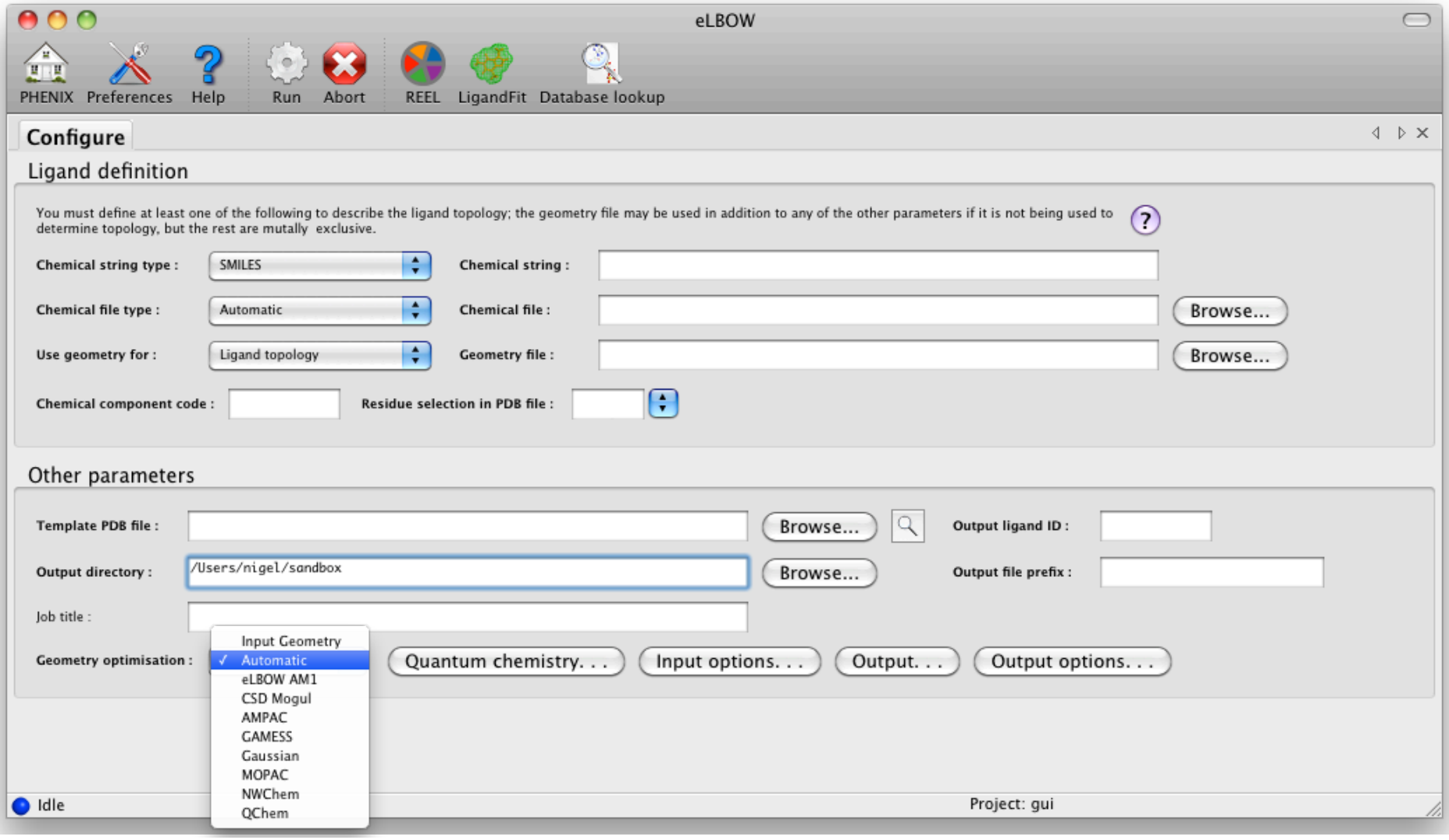
Ligands: ReadySet! - One-stop preparation for your refinement needs

- Generates files for refinement
  - Adds hydrogens, deuteriums, metal-coordination, CIF file
  - Uses Reduce for protein hydrogens and eLBOW for ligands
  - Command line **`phenix.ready_set model.pdb`** will do this all!



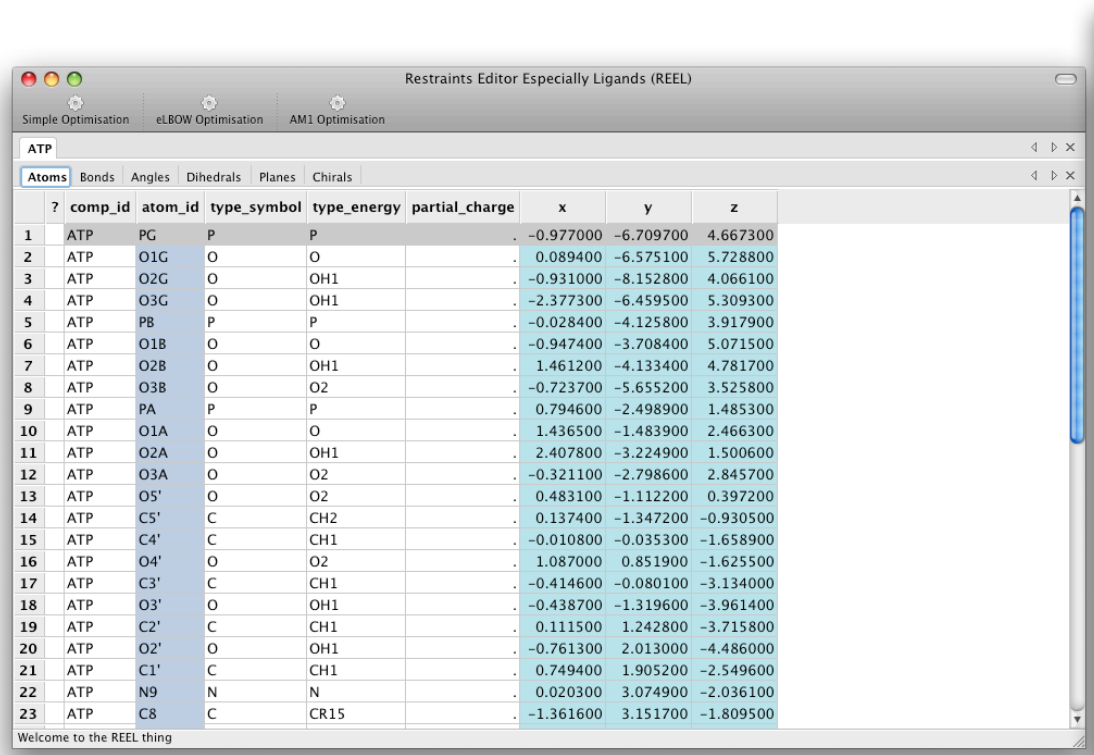
Ligands: eLBOW

- eLBOW - electronic Ligand Builder & Optimisation Workbench: uses a semi-empirical method to generate atomic coordinates from a chemical topology, then calculates restraint values

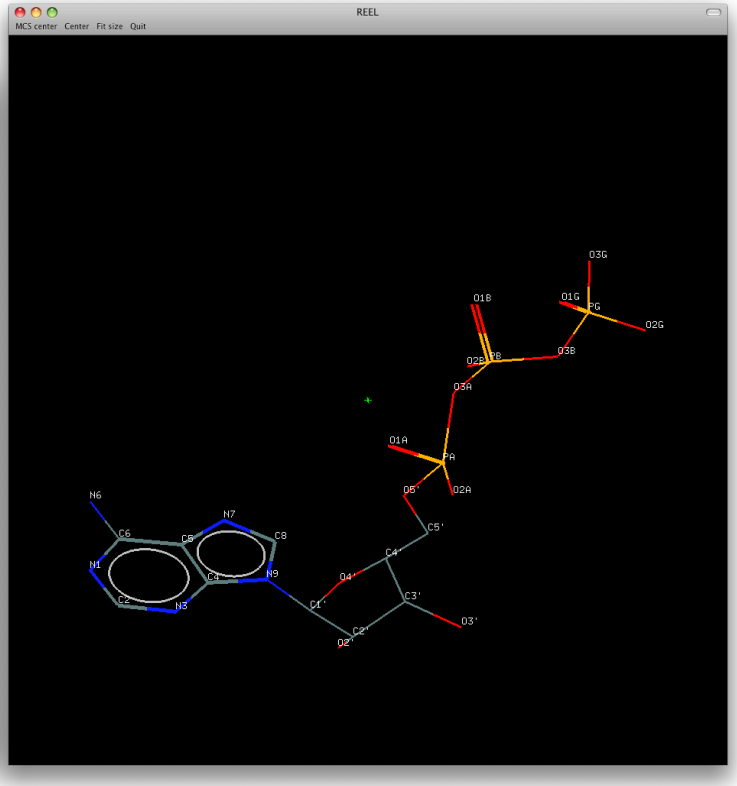


Ligands: REEL (Restraints Editor, Especially Ligands)

- phenix.reel provides a graphical interface to manipulate restraints
  - Editing of restraints using a molecule & table view
  - Two-way and set-intersection highlighting
  - Comparison of ligand geometries
  - eLBOW interface
  - Generate files required to link a ligand to a protein

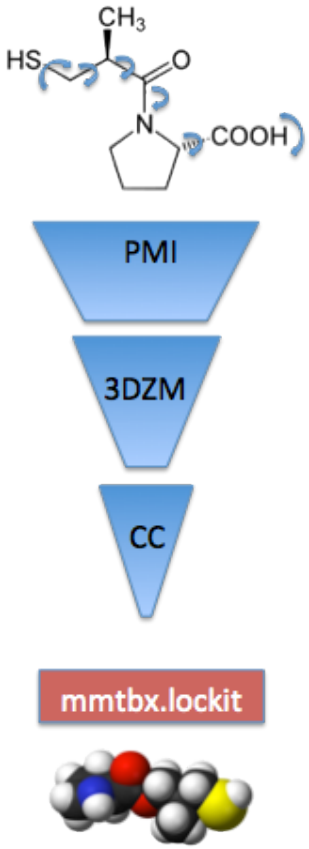


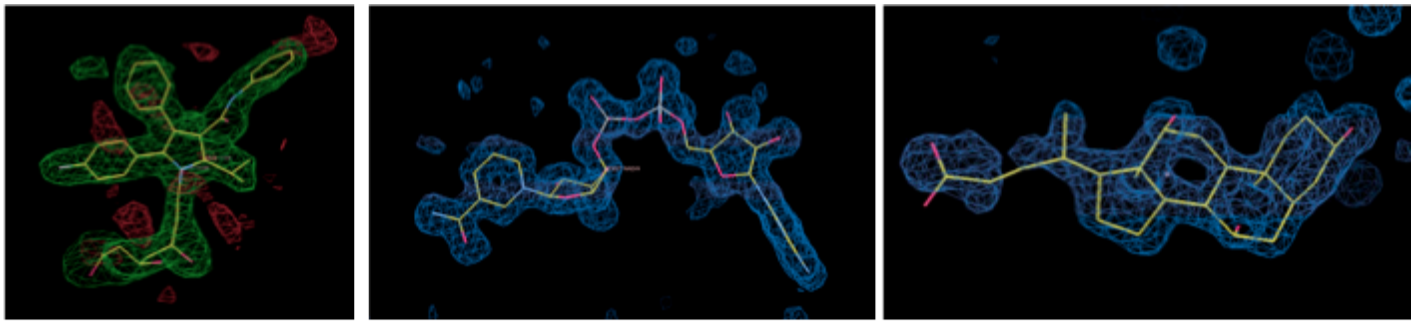
?	comp_id	atom_id	type_symbol	type_energy	partial_charge	x	y	z
1	ATP	PG	P	P	.	-0.977000	-6.709700	4.667300
2	ATP	O1G	O	O	.	0.089400	-6.575100	5.728800
3	ATP	O2G	O	OH1	.	-0.931000	-8.152800	4.066100
4	ATP	O3G	O	OH1	.	-2.377300	-6.459500	5.309300
5	ATP	PB	P	P	.	-0.028400	-4.125800	3.917900
6	ATP	O1B	O	O	.	-0.947400	-3.708400	5.071500
7	ATP	O2B	O	OH1	.	1.461200	-4.133400	4.781700
8	ATP	O3B	O	O2	.	-0.723700	-5.655200	3.525800
9	ATP	PA	P	P	.	0.794600	-2.498900	1.485300
10	ATP	O1A	O	O	.	1.436500	-1.483900	2.466300
11	ATP	O2A	O	OH1	.	2.407800	-3.224900	1.500600
12	ATP	O3A	O	O2	.	-0.321100	-2.798600	2.845700
13	ATP	O5'	O	O2	.	0.483100	-1.112200	0.397200
14	ATP	C5'	C	CH2	.	0.137400	-1.347200	-0.930500
15	ATP	C4'	C	CH1	.	-0.010800	-0.035300	-1.658900
16	ATP	O4'	O	O2	.	1.087000	0.851900	-1.625500
17	ATP	C3'	C	CH1	.	-0.414600	-0.080100	-3.134000
18	ATP	O3'	O	OH1	.	-0.438700	-1.319600	-3.961400
19	ATP	C2'	C	CH1	.	0.111500	1.242800	-3.715800
20	ATP	O2'	O	OH1	.	-0.761300	2.013000	-4.486000
21	ATP	C1'	C	CH1	.	0.749400	1.905200	-2.549600
22	ATP	N9	N	N	.	0.020300	3.074900	-2.036100
23	ATP	C8	C	CR15	.	-1.361600	3.151700	-1.809500



Ligands: ez-Ligand – fast ligand building into density map (in progress)

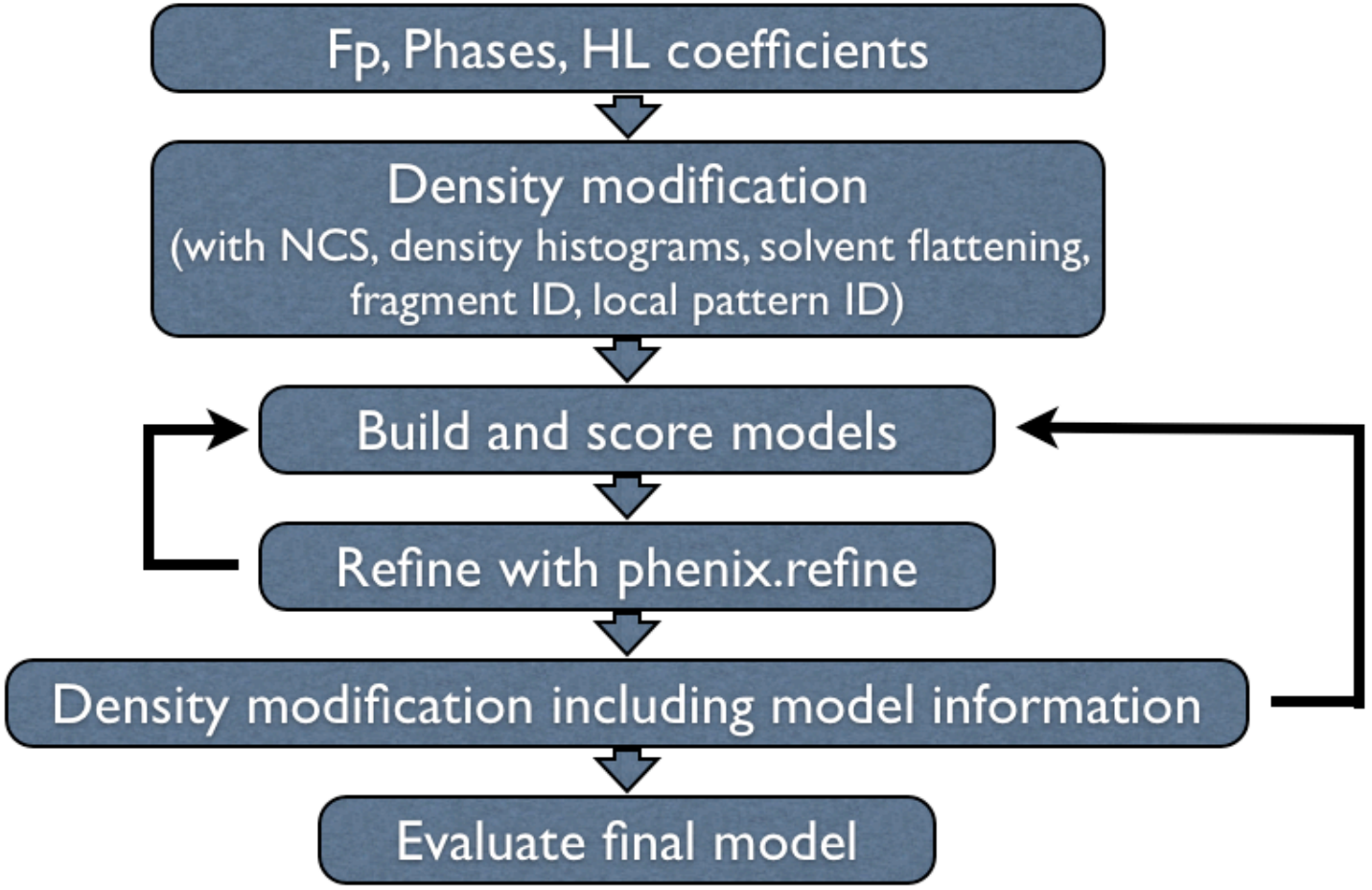
- General procedure
  - Find blob in difference density
  - Generate lots (>1e5) random conformations of a ligand via ultra-fast torsion angle sampling
  - For each conformation apply a series of ‘shape filters’ of increasing complexity
  - Principal moment of Inertia match (PMI)
  - Orientation independent shape features (3DZM)
  - Low resolution Real Space Correlation (CC)
  - FFT based fast rotational matching
  - Real Space Refinement





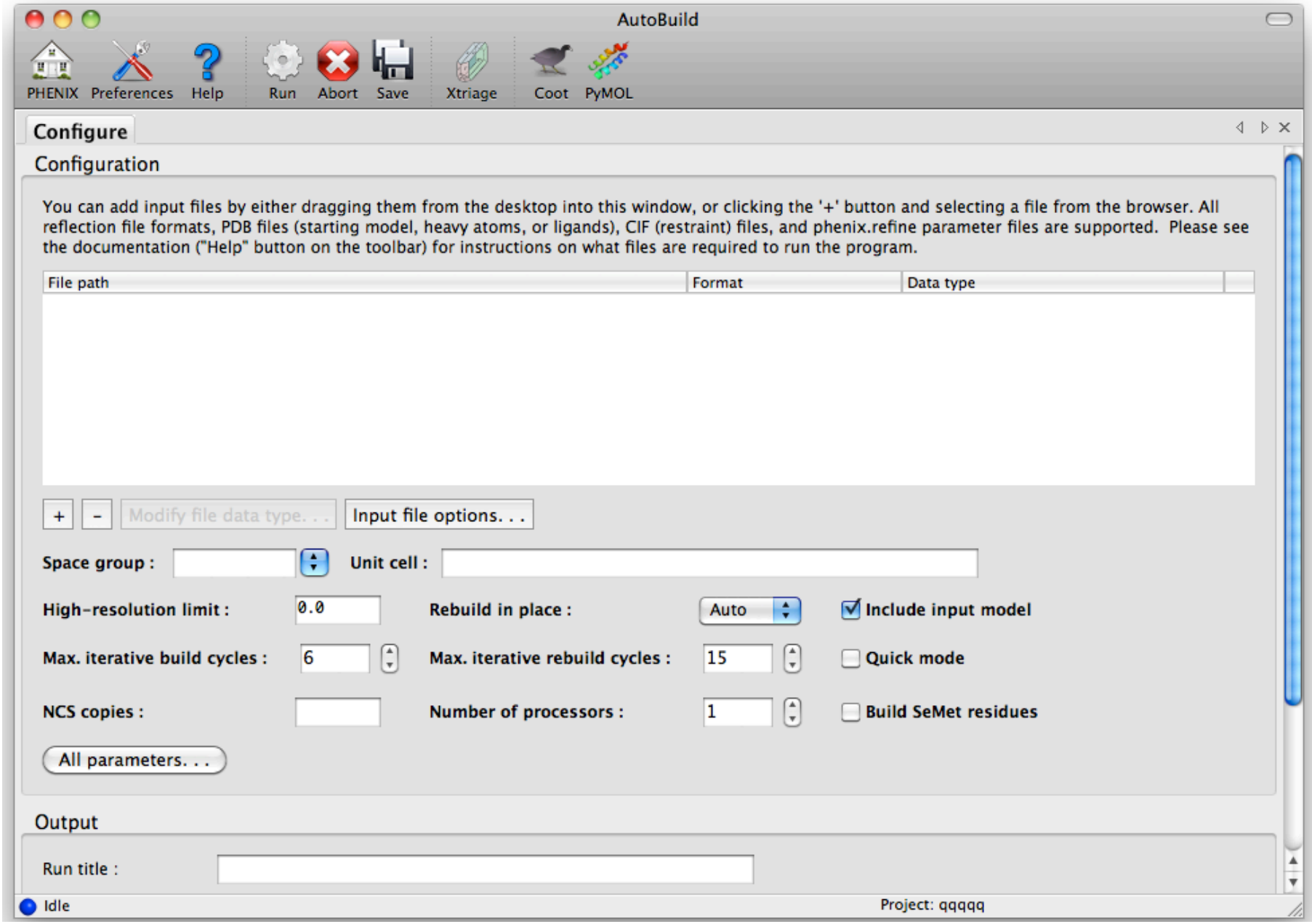
Lipitor: 50 seconds NAD: 2 minutes Cholic Acid: 50 seconds

Automated model building and rebuilding with AutoBuild

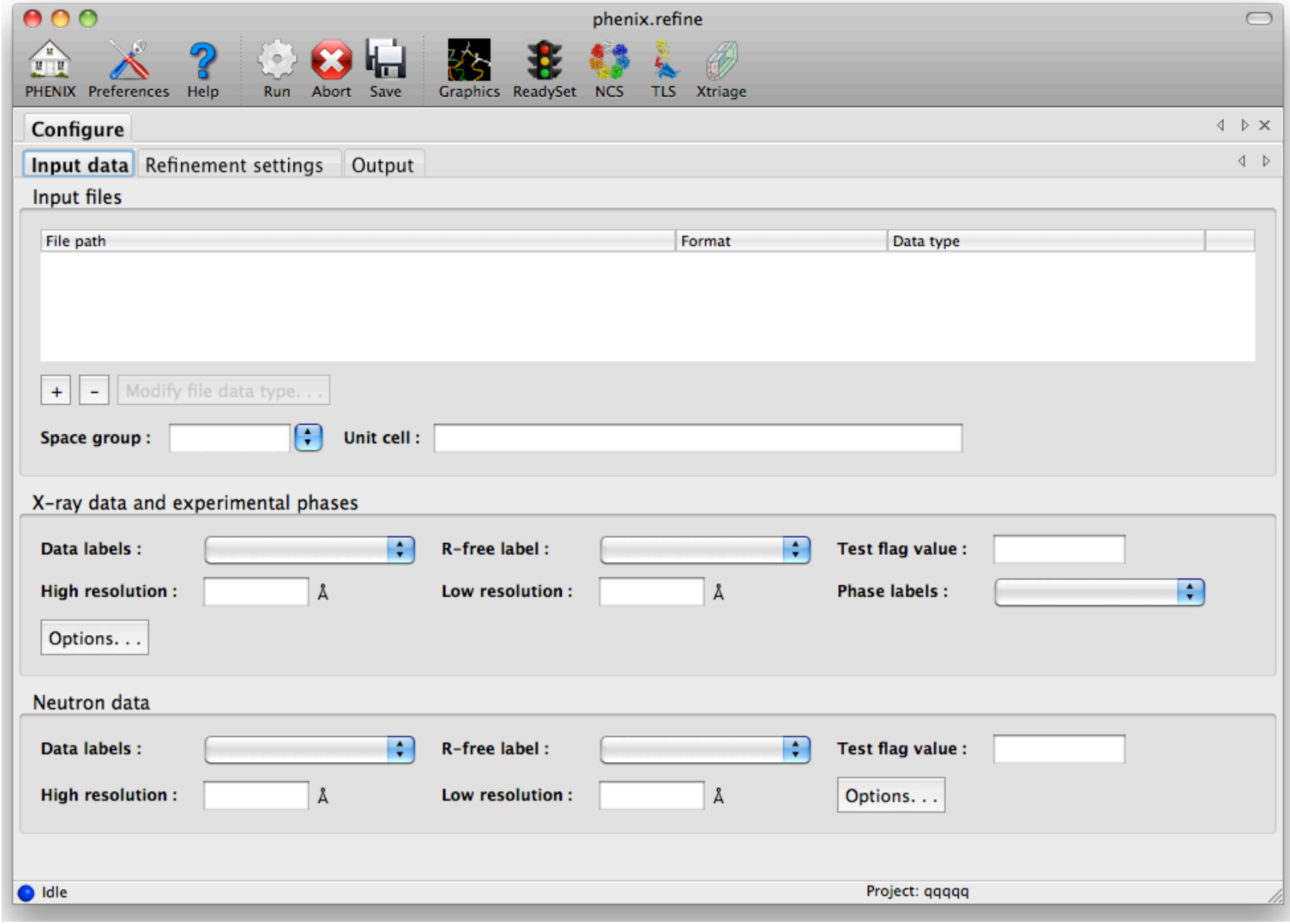


- Proteins and nucleic acids
- General model building: low to high resolution (3.5Å or better)
- Location of secondary structure elements (few seconds, tested at ~4Å)
- Loop building and extension and side chain docking

Automated model building and rebuilding with AutoBuild



phenix.refine GUI

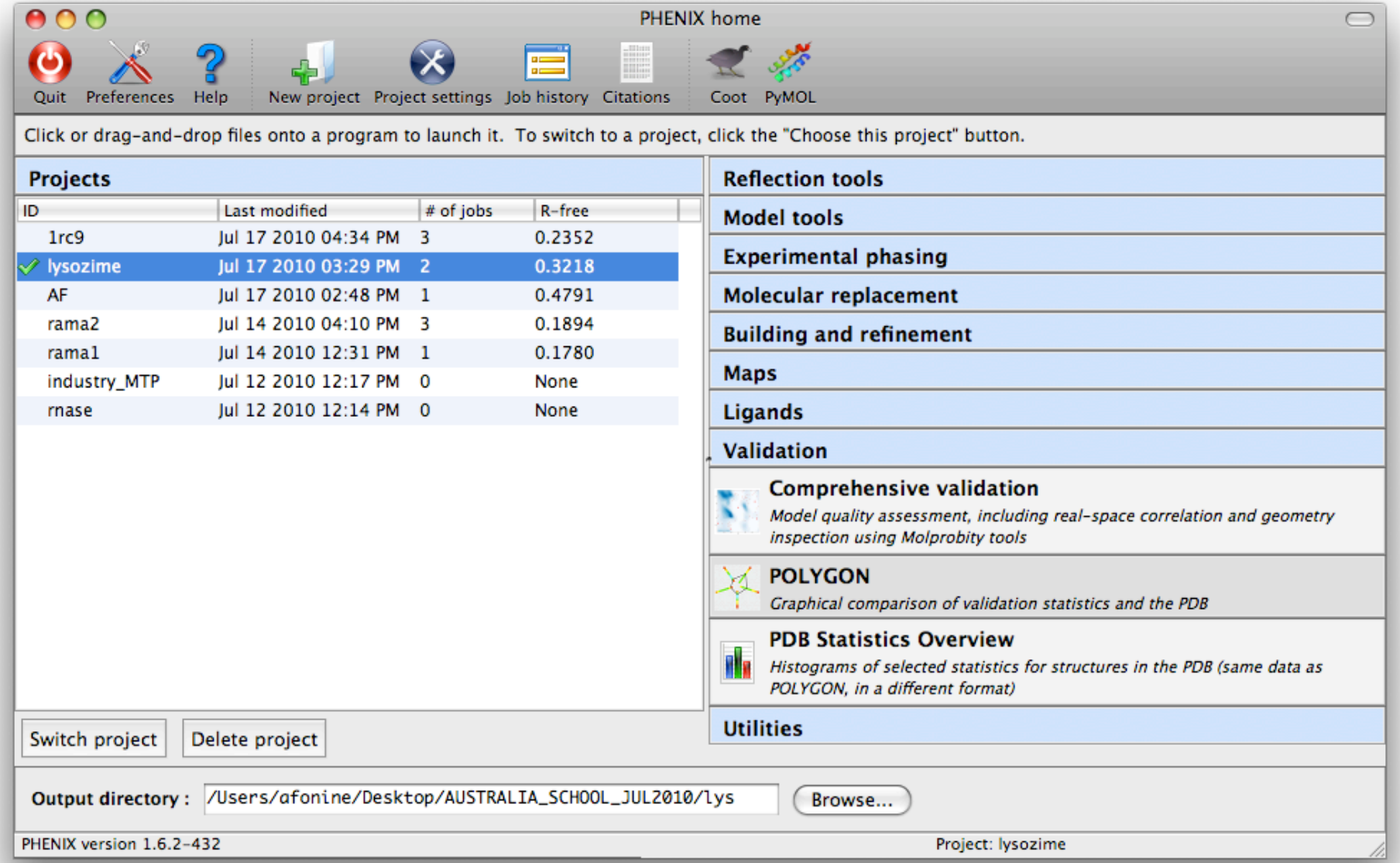


PHENIX tools for model validation

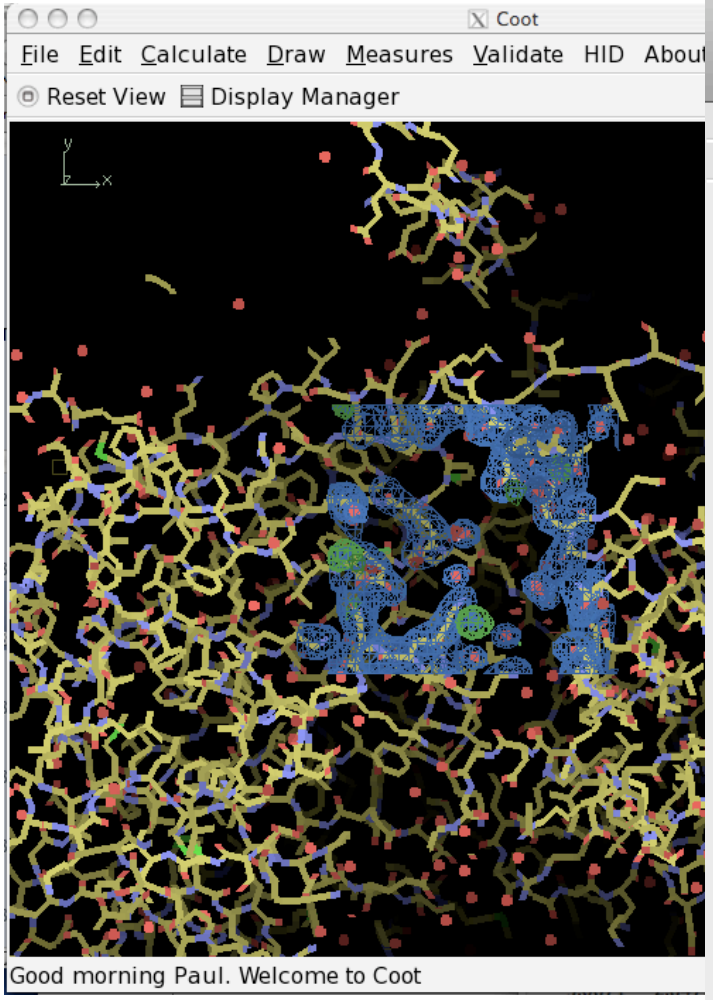
- **`Comprehensive validation`** option available from PHENIX GUI:
  - MolProbity scores;
  - Real-space correlation (map CC), 2mFo-DFc and mFo-DFc listed for each atom or residue;
  - Basic geometry statistics (rmsd and max deviation for bonds, angles, …)
  - phenix.model_vs_data report;
  - POLYGON.
- phenix.refine .log file contains lots of information.
- Tools to create various maps (iterative build omit maps, SA omit maps, Average kick maps, i*mFo-j*DFc maps)…
- Getting uncertainties by building multiple models.

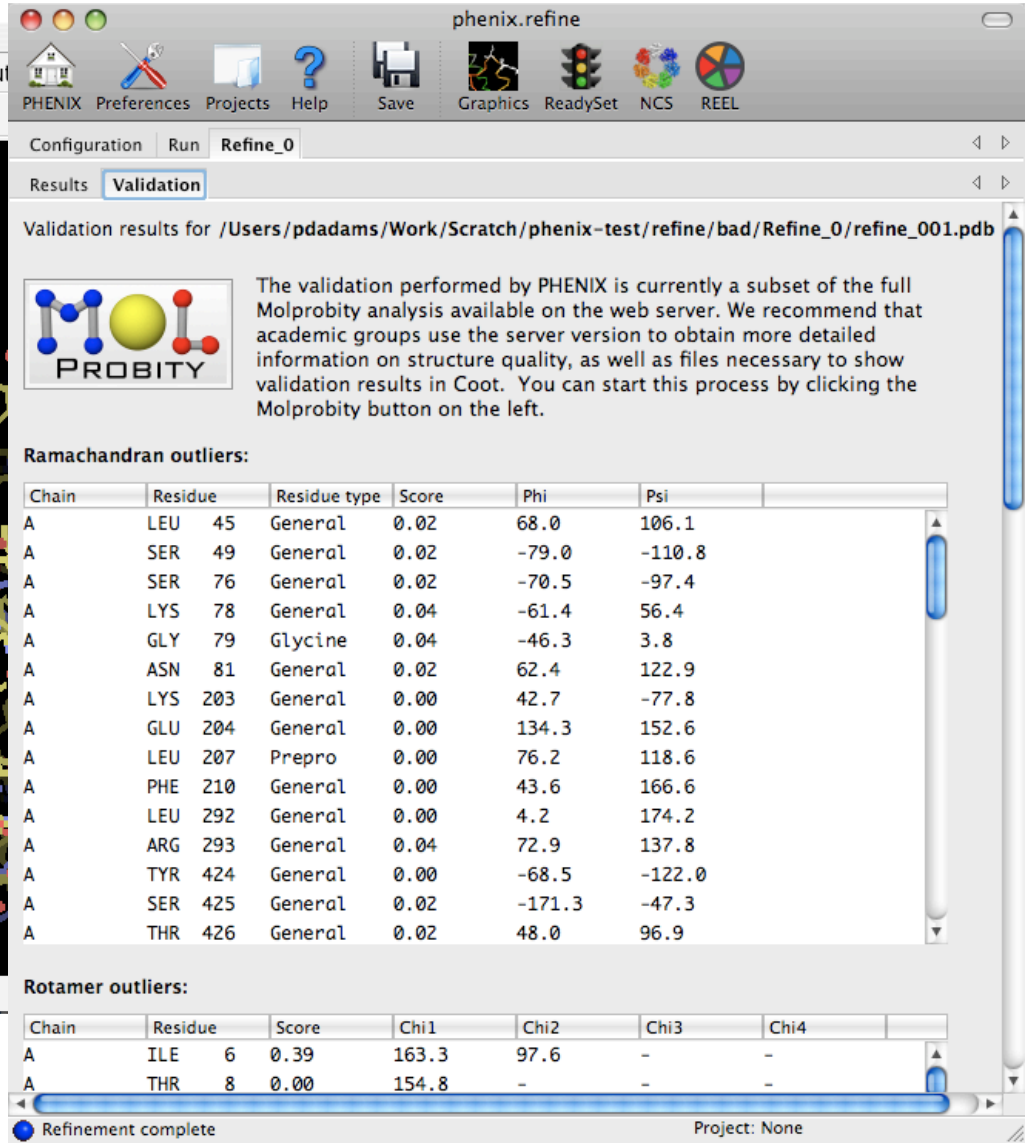
PHENIX tools for model validation

- **`Comprehensive validation`** option available from PHENIX GUI:

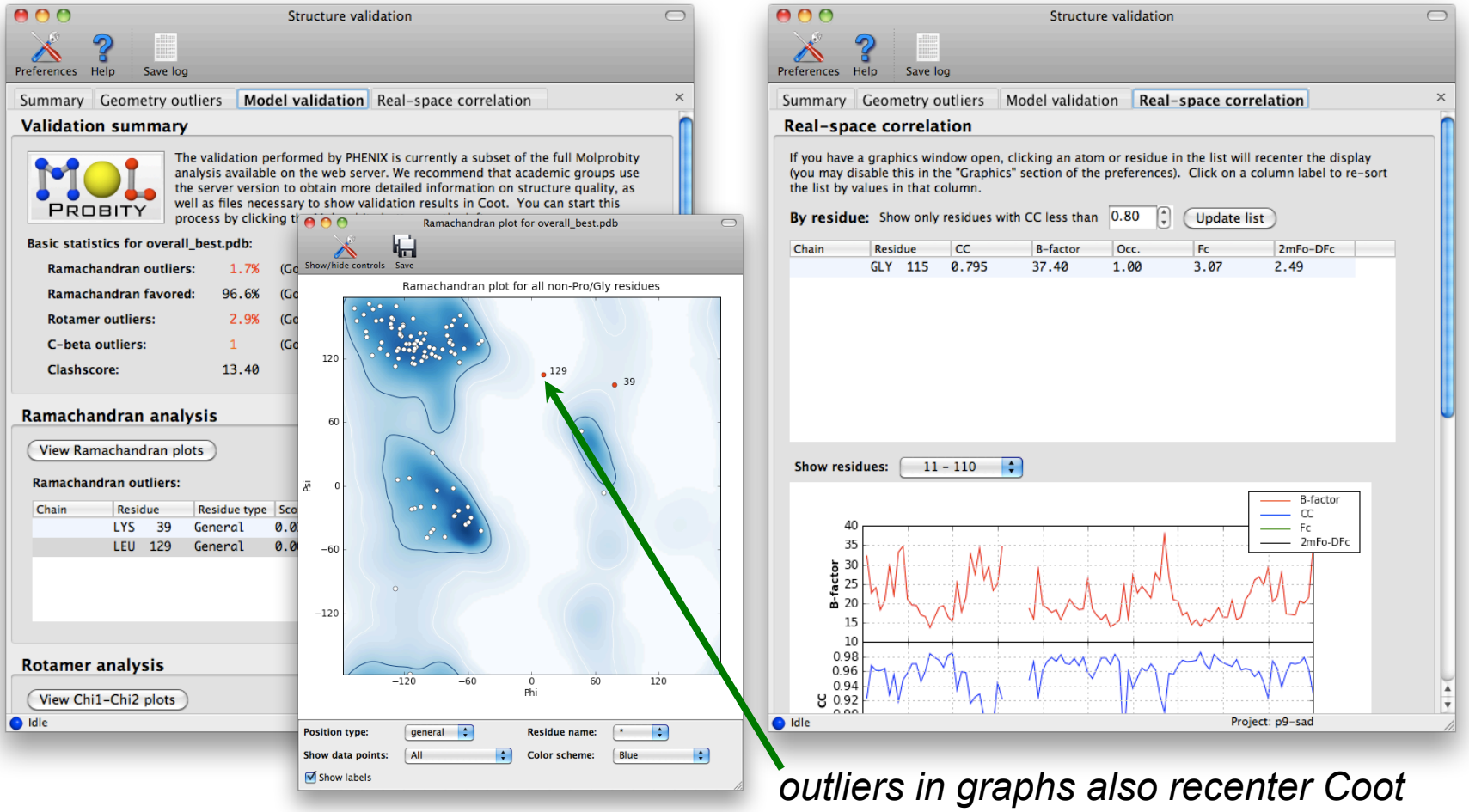


PHENIX tools for model validation





PHENIX tools for model validation



outliers in graphs also recenter Coot

New developments: integration of PHENIX with Rosetta

Frank DiMaio et al. Nature (2011)
Improved molecular replacement by density- and energy-guided protein structure optimization.

- Molecular replacement or the subsequent rebuilding usually fail with divergent starting models based on remote homologues with less than 30% sequence identity.

- It is demonstrated that this limitation can be substantially reduced by combining algorithms for protein structure modelling with those developed for crystallographic structure determination.

- Integrating *Rosetta structure modelling* with *PHENIX Autobuild chain tracing* yielded for 8 of 13 X-ray diffraction data sets that could not be solved in the laboratories of expert crystallographers and that remained unsolved after application of an extensive array of alternative approaches.

- An estimate is that the new method should allow rapid structure determination without experimental phase information for over half the cases where current methods fail, given diffraction data sets of better than 3.2 Å resolution, four or fewer copies in the asymmetric unit, and the availability of structures of homologous proteins with >20% sequence identity.

PHENIX command line tools

- Most of PHENIX command-line tools are invoked as

  phenix.command_name

  Example: phenix.refine, phenix.maps, etc.

- To see all available commands and quick hint about what it is:

  phenix.commands

- Typically, running a command without arguments will give a quick help message

- Currently there are 276 commands

PHENIX: principal tools

- Command line tools are still easy to run:
  - Autobuild *(from starting phases to complete and refined model)*:

    ```
    phenix.autobuild data=scale.mtz model=mr.pdb seq_file=correct.seq
    ```

  - Ligandfit (automatically find and build ligands into density):

    ```
    phenix.ligandfit data=nsf.mtz model=noligand.pdb ligand=atp.pdb
    ```

  - AutoMR (molecular replacement with Phaser + Autobuild = refined model):

    ```
    phenix.automr native.sca search.pdb RMS=0.8 mass=23000 copies=1
    ```

  - phenix.refine (highly automated structure refinement, X-ray, Neutron):

    ```
    phenix.refine nsf-d2.mtz nsf.pdb
    ```

  - phenix.xtriage (complete data analisys):

    ```
    phenix.xtriage porin_fp.mtz
    ```

Some commands

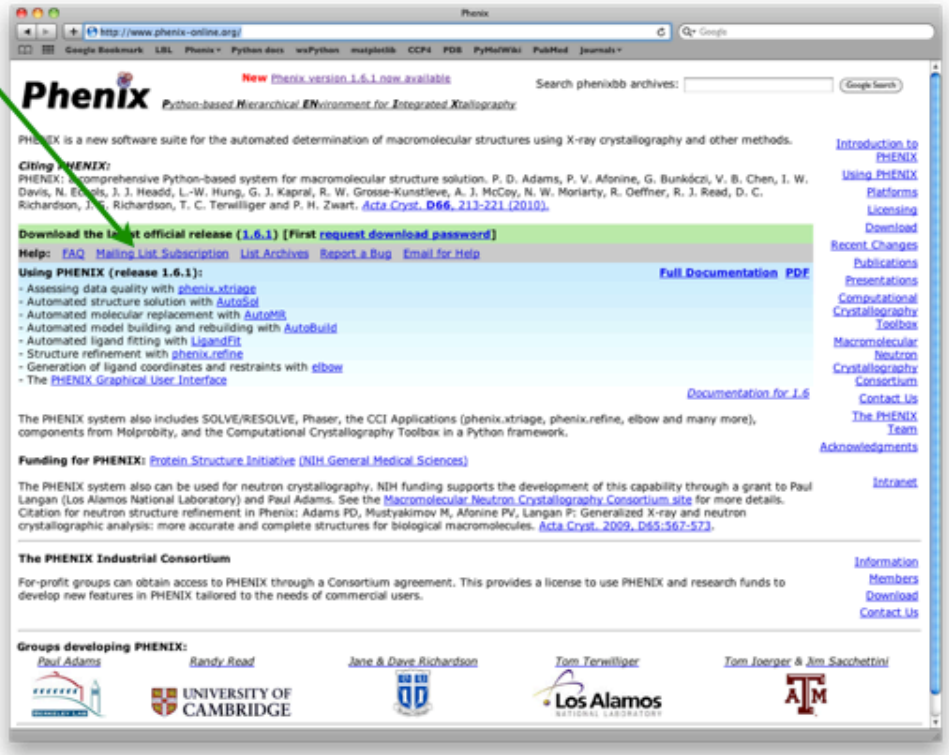
phenix:	Run Phenix graphical user interface
phenix.about:	Summarize contributors, packages, and info for phenix
phenix.acknowledgments:	Summarize third-party components of Phenix
phenix.autobuild:	Iterative model-building density modification and refinement
phenix.automr:	Automated MR and model-building
phenix.autosol:	Automated structure solution by MR/MAD/SAD
phenix.cif_as_mtz:	Convert CIF to MTZ
phenix.clashscore:	Evaluate model based on all-atom contacts
phenix.cns_as_mtz:	Convert CNS to MTZ
phenix.commands:	List command line Phenix methods
phenix.elbow:	Ligand builder (CIF from PDB, SMILES etc)
phenix.ensembler:	Superpose PDB files to create ensemble for MR
phenix.fetch_pdb:	Download a model from PDB (specify PDB code)
phenix.find_all_ligands:	Find ligands from a list in a map
phenix.find_helices_strands:	Build helices and strands into a map
phenix.find_tls_groups:	Automatic identification of appropriate TLS groups in model
phenix.fit_loops:	Fit missing loops in a model
phenix.fmodel:	Calculate structure factors from model
phenix.form_factor_query:	f' and f" table lookup given element and wavelength
phenix.ksdssp:	Identify secondary structure in a model
phenix.map_value_at_point:	Get map value at given coordinates
phenix.maps:	Create maps from PDB and MTZ files
phenix.metal_coordination:	Generate restraints for metal coordination sites
phenix.model_vs_data:	Evaluate model using experimental data
phenix.mtz.dump:	Dump MTZ file contents
phenix.mtz2map:	Convert MTZ file to map (superseded by phenix.maps)

Some commands

phenix.multi_crystal_average:	Multi-crystal averaging
phenix.pdb.hierarchy:	Quick summary of PDB file content
phenix.pdb_atom_selection:	Extract selected atoms from PDB file
phenix.pdb_interpretation:	Read PDB file and build restraints for refinement
phenix.pdbtools:	Manipulate PDB files
phenix.phaser:	Run PHASER
phenix.polygon:	Compare model statistics to expected distributions
phenix.print_sequence:	Print sequence from PDB file
phenix.r_factor_statistics:	R-factor statistics at given resolution
phenix.ramalyze:	Validate protein backbone Ramachandran dihedral angles
phenix.ready_set:	Prepare for refinement
phenix.reduce:	Run REDUCE, software for addition or trimming of hydrogens
phenix.refine:	Carry out refinement of a model
phenix.resolve:	Run resolve
phenix.rotalyze:	Validate protein sidechain rotamers
phenix.solve:	Run SOLVE
phenix.superpose_ligands:	Superimpose two ligands
phenix.superpose_maps:	Superimpose PDB files and transform map to match
phenix.superpose_pdbs:	Superimpose PDB files using aligned sequences
phenix.table_one:	Prepare Table 1 for publicaiton
phenix.tls:	Extract/Combine TLS from PDB file
phenix.version:	Print version of Phenix
phenix.xtriage:	Analyze data files for quality and unusual conditions

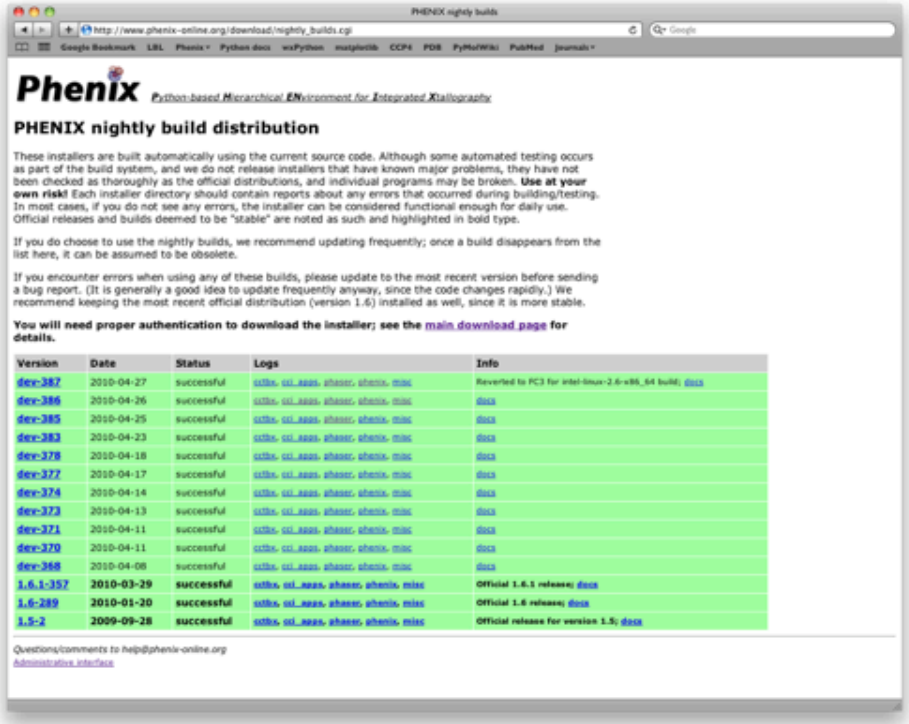
PHENIX resources online

- help@phenix-online.org: user support
- bugs@phenix-online.org: bug reports
- phenixbb@phenix-online.org: message board (subscribers only)



Obtaining PHENIX

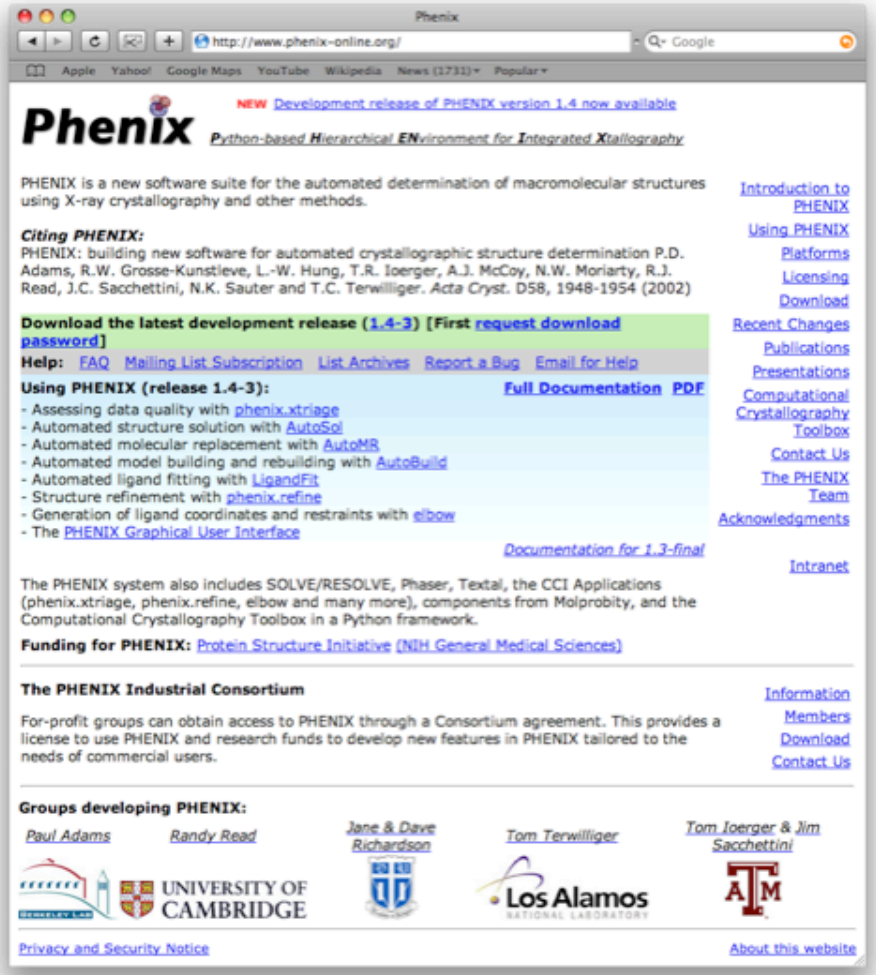
- Free to academic users; simple online registration required *(please use your academic email address!)*
- Regular official releases (typically 2-8 months)
- Nightly builds



http://www.phenix-online.org/download/nightly_builds.cgi

PHENIX Distribution

- Regular releases
- Supported on:
  - Linux (RedHat, Fedora)
  - Mac OSX
- Regular development releases:
  - Nightly builds
  - **http://www.phenix-online.org/**
- Extensive documentation

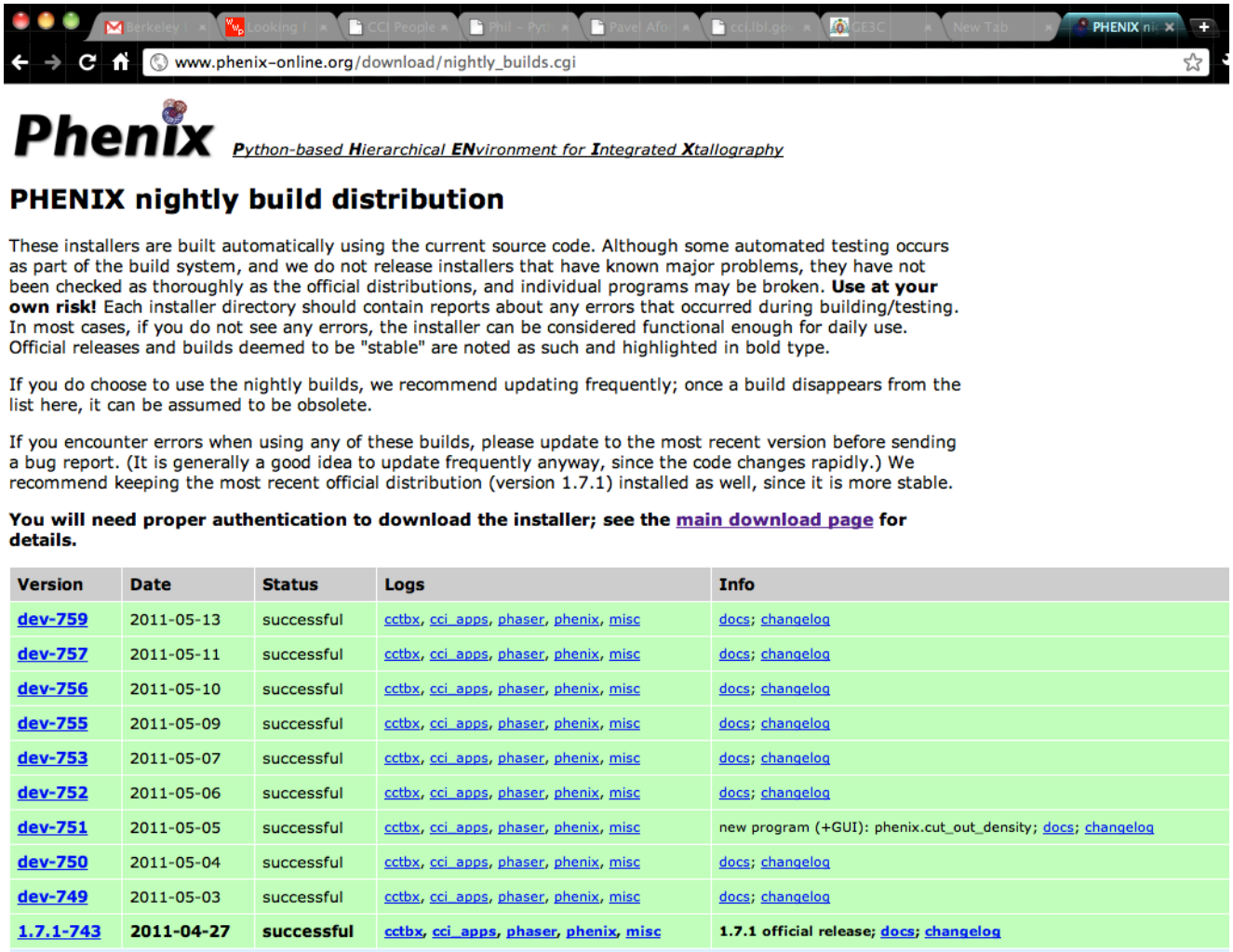


BERKELEY LAB

Phenix

PHENIX: installation

- **You can do it:** you don't have to have administrative privileges to install PHENIX: you can always install it under your own account in the place of your choice.
- **Use** the latest version from nightly builds: it has latest bug fixes, newest features, …
- **Keep** the latest official release in case the nightly build is broken (you can have several versions of PHENIX installed on your computer).



Reporting bugs, problems, asking questions

- **Something didn't work as expected?... program crashed?... missing feature?...**

  Not Good: silently give up and run away looking for alternative software (or write your own program).

  Good: report us a problem, ask a question, request a feature (explain why it's good to have), ask for help.

- **Reporting a bug:**

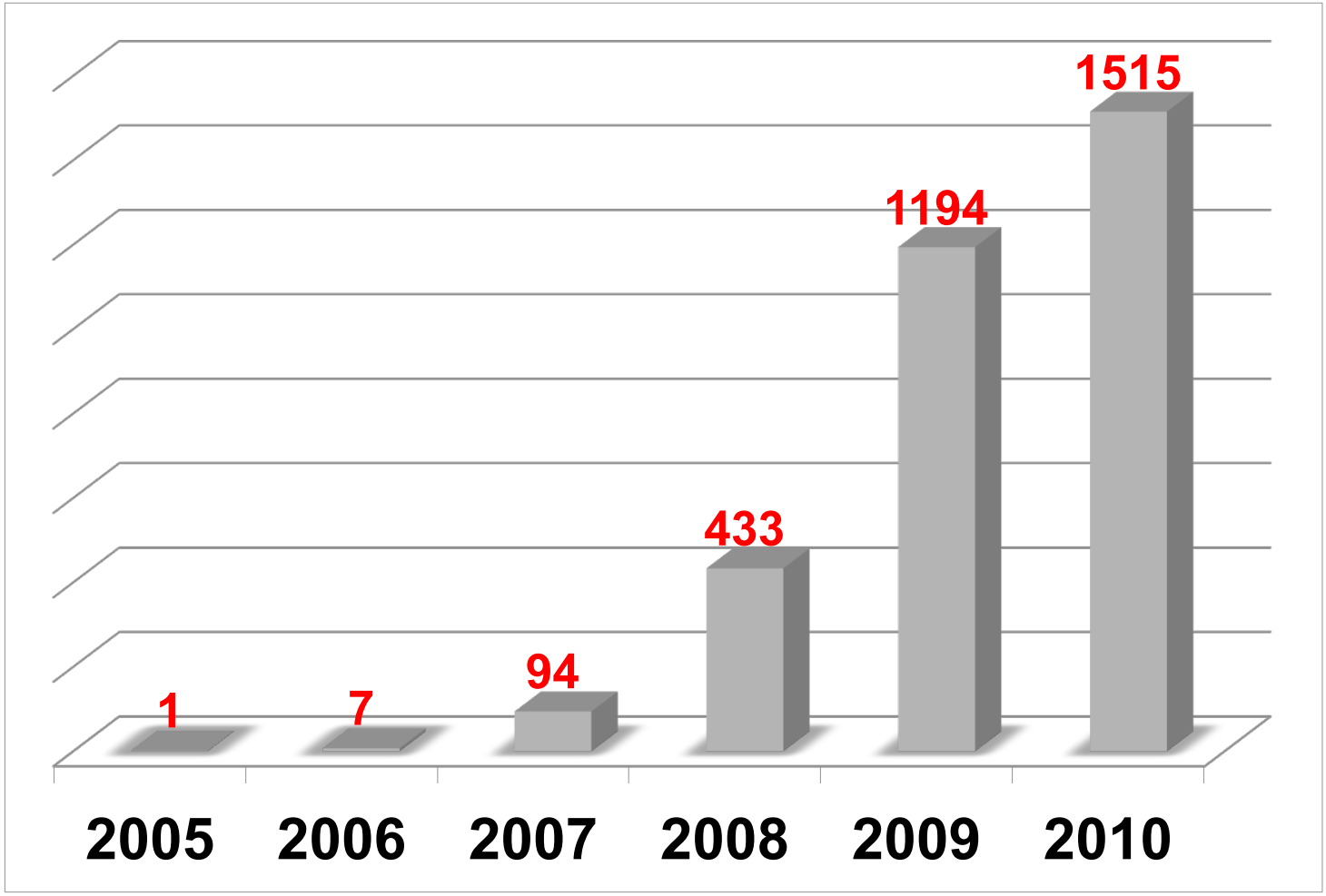
  Not good: "Hi! PHENIX crashed, I don't know what to do."

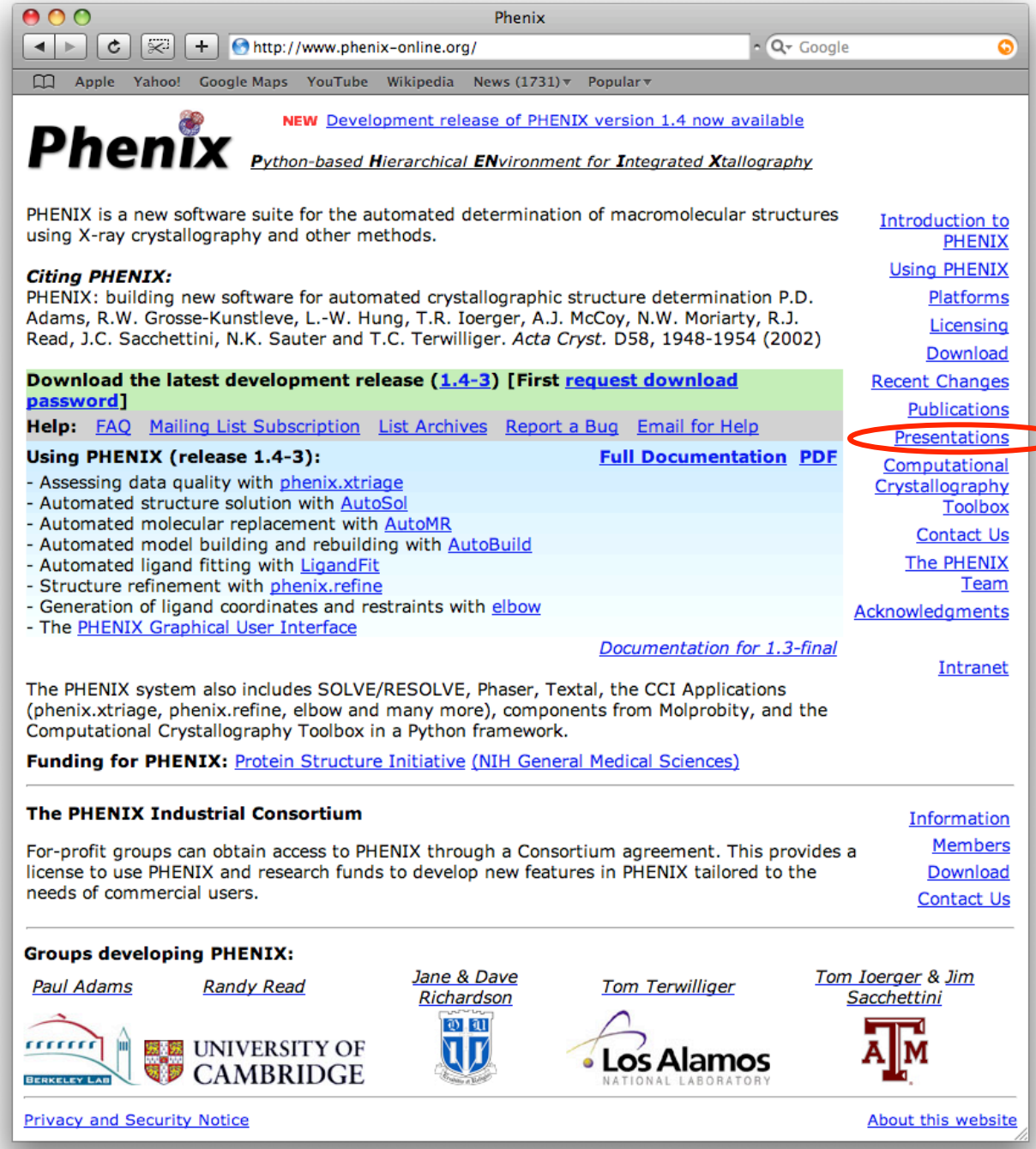
  Good: "Hi! PHENIX crashed. Here are:
  1) PHENIX version;
  2) Command and parameters I used;
  3) Input and output files (at least logs)."

Subscribe to PHENIX bulletin board: www.phenix-online.org

PHENIX use (May 18, 2011)

Number of structures in PDB with “REMARK 3 PROGRAM PHENIX”





This presentation (PDF file) and much more

www.phenix-online.org