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)
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
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, 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.

• 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**
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."


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

Click or drag-and-drop files onto a program to launch it. To switch to a project, click the "Choose this project" button.

### Projects

<table>
<thead>
<tr>
<th>ID</th>
<th>Last modified</th>
<th># of jobs</th>
<th>R-free</th>
</tr>
</thead>
<tbody>
<tr>
<td>rnae</td>
<td>Jul 12 2010 12:14 PM</td>
<td>0</td>
<td>None</td>
</tr>
<tr>
<td>industry_MTP</td>
<td>Jul 12 2010 12:17 PM</td>
<td>0</td>
<td>None</td>
</tr>
</tbody>
</table>

### Reflection tools

- Model tools
- Experimental phasing
- Molecular replacement
- Building and refinement
- Maps
- Ligands
- Validation
- Utilities

Output directory: /Users/afonine/Desktop/zz/zz1

PHENIX version 1.6.2-432
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).

drag data arrays to output list
advanced output settings
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.

Data analysis with **phenix.xtriage**

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

- Any reflection file format permitted
- Drag-and-drop supported
- One-click re-use of partial solutions from past runs
- Can use a low-resolution map as a search model
- Most keywords found here
- GUI examples: PHASER
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 \(F(\text{obs})\) with \(F(\text{calc})\): 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.

![Ligand Builder & Optimisation Workbench Software Interface](image-url)
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
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

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

Configuration

You can add input files by either dragging them from the desktop into this window, or clicking the '+' button and selecting a file from the browser. All reflection file formats, PDB files (starting model, heavy atoms, or ligands), CIF (restraint) files, and phenix.refine parameter files are supported. Please see the documentation ("Help" button on the toolbar) for instructions on what files are required to run the program.

File path

<table>
<thead>
<tr>
<th>Format</th>
<th>Data type</th>
</tr>
</thead>
</table>

Modify file data type...

Input file options...

Space group:  
Unit cell:  

High-resolution limit: 0.0  
Rebuild in place: Auto  
Include input model

Max. iterative build cycles: 6  
Max. iterative rebuild cycles: 15  
Quick mode

NCS copies:  
Number of processors: 1  
Build SeMet residues

All parameters...

Output

Run title:  

Idle

Project: ppqqqq
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:

<table>
<thead>
<tr>
<th>Projects</th>
<th>ID</th>
<th>Last modified</th>
<th># of jobs</th>
<th>R-free</th>
</tr>
</thead>
<tbody>
<tr>
<td>lysozime</td>
<td>1rc9</td>
<td>Jul 17 2010 04:34 PM</td>
<td>3</td>
<td>0.2352</td>
</tr>
<tr>
<td>lysozime</td>
<td>AF</td>
<td>Jul 17 2010 02:48 PM</td>
<td>1</td>
<td>0.4791</td>
</tr>
<tr>
<td>lysozime</td>
<td>rama2</td>
<td>Jul 14 2010 04:10 PM</td>
<td>3</td>
<td>0.1894</td>
</tr>
<tr>
<td>lysozime</td>
<td>rama1</td>
<td>Jul 14 2010 12:31 PM</td>
<td>1</td>
<td>0.1780</td>
</tr>
<tr>
<td>lysozime</td>
<td>industry_MTP</td>
<td>Jul 12 2010 12:17 PM</td>
<td>0</td>
<td>None</td>
</tr>
<tr>
<td>lysozime</td>
<td>trnase</td>
<td>Jul 12 2010 12:14 PM</td>
<td>0</td>
<td>None</td>
</tr>
</tbody>
</table>

Reflection tools
- Model tools
- Experimental phasing
- Molecular replacement
- Building and refinement
- Maps
- Ligands

Validation
- Comprehensive validation
  - Model quality assessment, including real-space correlation and geometry inspection using Molprobity tools

Utilities
- POLYGON
  - Graphical comparison of validation statistics and the PDB
- PDB Statistics Overview
  - Histograms of selected statistics for structures in the PDB (same data as POLYGON, in a different format)

Output directory: /Users/afonine/Desktop/AUSTRALIA_SCHOOL_JUL2010/lysozime

PHENIX version 1.6.2-432 Project: lysozime
Good morning Paul. Welcome to Coot.
PHENIX tools for model validation

outliers in graphs also recenter Coot
New developments: integration of PHENIX with Rosetta

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 analysis)*:
    ```
    phenix.xtriage porin_fp.mtz
    ```
### Some commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>phenix:</td>
<td>Run Phenix graphical user interface</td>
</tr>
<tr>
<td>phenix.about:</td>
<td>Summarize contributors, packages, and info for phenix</td>
</tr>
<tr>
<td>phenix.acknowledgments:</td>
<td>Summarize third-party components of Phenix</td>
</tr>
<tr>
<td>phenix.autobuild:</td>
<td>Iterative model-building density modification and refinement</td>
</tr>
<tr>
<td>phenix.autosmr:</td>
<td>Automated MR and model-building</td>
</tr>
<tr>
<td>phenix.cif_as_mtz:</td>
<td>Convert CIF to MTZ</td>
</tr>
<tr>
<td>phenix.clashscore:</td>
<td>Evaluate model based on all-atom contacts</td>
</tr>
<tr>
<td>phenix.cns_as_mtz:</td>
<td>Convert CNS to MTZ</td>
</tr>
<tr>
<td>phenix.commands:</td>
<td>List command line Phenix methods</td>
</tr>
<tr>
<td>phenix.elbow:</td>
<td>Ligand builder (CIF from PDB, SMILES etc)</td>
</tr>
<tr>
<td>phenix.ensembler:</td>
<td>Superpose PDB files to create ensemble for MR</td>
</tr>
<tr>
<td>phenix.fetch_pdb:</td>
<td>Download a model from PDB (specify PDB code)</td>
</tr>
<tr>
<td>phenix.find_all_ligands:</td>
<td>Find ligands from a list in a map</td>
</tr>
<tr>
<td>phenix.find_helices_strands:</td>
<td>Build helices and strands into a map</td>
</tr>
<tr>
<td>phenix.find_tls_groups:</td>
<td>Automatic identification of appropriate TLS groups in model</td>
</tr>
<tr>
<td>phenix.fit_loops:</td>
<td>Fit missing loops in a model</td>
</tr>
<tr>
<td>phenix.fmodel:</td>
<td>Calculate structure factors from model</td>
</tr>
<tr>
<td>phenix.form_factor_query:</td>
<td>f' and f&quot; table lookup given element and wavelength</td>
</tr>
<tr>
<td>phenix.ksdssp:</td>
<td>Identify secondary structure in a model</td>
</tr>
<tr>
<td>phenix.map_value_at_point:</td>
<td>Get map value at given coordinates</td>
</tr>
<tr>
<td>phenix.maps:</td>
<td>Create maps from PDB and MTZ files</td>
</tr>
<tr>
<td>phenix.metal_coordination:</td>
<td>Generate restraints for metal coordination sites</td>
</tr>
<tr>
<td>phenix.model_vs_data:</td>
<td>Evaluate model using experimental data</td>
</tr>
<tr>
<td>phenix.mtz.dump:</td>
<td>Dump MTZ file contents</td>
</tr>
<tr>
<td>phenix.mtz2map:</td>
<td>Convert MTZ file to map (superseded by phenix.maps)</td>
</tr>
</tbody>
</table>
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_pdbds**: Superimpose PDB files using aligned sequences
- **phenix.table_one**: Prepare Table 1 for publication
- **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
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).

---

**PHENIX nightly build distribution**

These installers are built automatically using the current source code. Although some automated testing occurs as part of the build system, and we do not release installers that have known major problems, they have not been checked as thoroughly as the official distributions, and individual programs may be broken. **Use at your own risk!** Each installer directory should contain reports about any errors that occurred during building/testing. In most cases, if you do not see any errors, the installer can be considered functional enough for daily use. Official releases and builds deemed to be "stable" are noted as such and highlighted in bold type.

If you do choose to use the nightly builds, we recommend updating frequently; once a build disappears from the list here, it can be assumed to be obsolete.

If you encounter errors when using any of these builds, please update to the most recent version before sending a bug report. (It is generally a good idea to update frequently anyway, since the code changes rapidly.) We recommend keeping the most recent official distribution (version 1.7.1) installed as well, since it is more stable.

You will need proper authentication to download the installer; see the main download page for details.

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Status</th>
<th>Logs</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev-759</td>
<td>2011-05-13</td>
<td>successful</td>
<td>cctbx, cc_apps, phaser, phenix, misc</td>
<td>docs; changelog</td>
</tr>
<tr>
<td>dev-757</td>
<td>2011-05-11</td>
<td>successful</td>
<td>cctbx, cc_apps, phaser, phenix, misc</td>
<td>docs; changelog</td>
</tr>
<tr>
<td>dev-756</td>
<td>2011-05-10</td>
<td>successful</td>
<td>cctbx, cc_apps, phaser, phenix, misc</td>
<td>docs; changelog</td>
</tr>
<tr>
<td>dev-755</td>
<td>2011-05-09</td>
<td>successful</td>
<td>cctbx, cc_apps, phaser, phenix, misc</td>
<td>docs; changelog</td>
</tr>
<tr>
<td>dev-753</td>
<td>2011-05-07</td>
<td>successful</td>
<td>cctbx, cc_apps, phaser, phenix, misc</td>
<td>docs; changelog</td>
</tr>
<tr>
<td>dev-752</td>
<td>2011-05-06</td>
<td>successful</td>
<td>cctbx, cc_apps, phaser, phenix, misc</td>
<td>docs; changelog</td>
</tr>
<tr>
<td>dev-751</td>
<td>2011-05-05</td>
<td>successful</td>
<td>cctbx, cc_apps, phaser, phenix, misc</td>
<td>new program (+GUI): phenix.cut_out_density; docs; changelog</td>
</tr>
<tr>
<td>dev-750</td>
<td>2011-05-04</td>
<td>successful</td>
<td>cctbx, cc_apps, phaser, phenix, misc</td>
<td>docs; changelog</td>
</tr>
<tr>
<td>dev-749</td>
<td>2011-05-03</td>
<td>successful</td>
<td>cctbx, cc_apps, phaser, phenix, misc</td>
<td>docs; changelog</td>
</tr>
<tr>
<td>1.7.1-743</td>
<td>2011-04-27</td>
<td>successful</td>
<td>cctbx, cc_apps, phaser, phenix, misc</td>
<td>1.7.1 official release; docs; changelog</td>
</tr>
</tbody>
</table>
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](http://www.phenix-online.org)
PHENIX use (May 18, 2011)

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

<table>
<thead>
<tr>
<th>Year</th>
<th>2005</th>
<th>2006</th>
<th>2007</th>
<th>2008</th>
<th>2009</th>
<th>2010</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>7</td>
<td>94</td>
<td>433</td>
<td>1194</td>
<td>1515</td>
</tr>
</tbody>
</table>
NEW Development release of PHENIX version 1.4 now available

Python-based Hierarchical ENvironment for Integrated Xtallography

PHENIX is a new software suite for the automated determination of macromolecular structures using X-ray crystallography and other methods.

Citing PHENIX:

Download the latest development release (1.4-3) [First request download password]

Using PHENIX (release 1.4-3):
- Assessing data quality with phanix.xtriage
- Automated structure solution with AutoSol
- Automated molecular replacement with AutoMR
- Automated model building and rebuilding with AutoBuild
- Automated ligand fitting with LigandFit
- Structure refinement with phx.xreline
- Generation of ligand coordinates and restraints with elbow
- The PHENIX Graphical User Interface

The PHENIX system also includes SOLVE/RESOLVE, Phaser, Textal, the CCI Applications (phenix.xtriage, phenix.xreline, elbow and many more), components from Molprobity, and the Computational Crystallography Toolbox in a Python framework.

Funding for PHENIX: Protein Structure Initiative (NIH General Medical Sciences)

The PHENIX Industrial Consortium

For-profit groups can obtain access to PHENIX through a Consortium agreement. This provides a license to use PHENIX and research funds to develop new features in PHENIX tailored to the needs of commercial users.

Groups developing PHENIX:
Paul Adams
Randy Read
Jane & Dave Richardson
Tom Terwillger
Tom Joerger & Jim Sacchettni

www.phenix-online.org