VOLUME TWO

COMPUTATIONAL CRYSTALLOGRAPHY NEWSLETTER

JULY MMXI

DATA VIEWER, MR-ROSETTA, LYSOZYME, SPOTFINDER

Table of Contents

Editor

Nigel W. Moriarty, NWMoriarty@LBL.Gov

Contributors

P. D. Adams, P. V. Afonine, D. Baker, G. Bunkóczi, F. DiMaio, N. Echols, J. J. Headd, R. W. Grosse-Kunstleve, D. Lucent, N. W. Moriarty, J. Newman, T. S. Peat, R. J. Read, D. C. Richardson, J. S. Richardson, N. K. Sauter, T. C. Terwilliger

PHENIX News

New releases

The default behavior of KiNG has been improved when loading electron density maps. Now when KiNG is launched from the refinement and validation GUIs, maps are automatically loaded and presented using the Coot-default color scheme. This Coot-default color scheme can also be accessed when opening maps with the command-line *phenix.king* by using the `-phenix` flag. Finally, KiNG includes and applies better map presets for 2Fo-Fc, Fo-Fc and anomalous maps.

A new graphical tool for visualization of reciprocal-space data is now available in *PHENIX* and is discussed in the short communications section starting on page 88.

Open source spotfinding code has been released in the cctbx for use at beamlines. A description of this extremely fast program is in the short communications section on page 93.

New features

The recent inclusion of Rosetta in *PHENIX* via the *phenix.mr_rosetta* has had many new features added. Hints on how to using it to its full potential are included in the short communications on page 94.

Refinement in *PHENIX* continues to be improved. An article about improved target

The Computational Crystallography Newsletter (CCN) is a regularly distributed electronically via email and the PHENIX website, www.phenix-online.org/newsletter. Feature articles, meeting announcements and reports, information on research or other items of interest to computational crystallographers or crystallographic software users can be submitted to the editor at any time for consideration. Submission of text by email or word-processing files using the CCN templates is requested.

weight optimization including use of parallel processing begins on page 99.

Crystallographic meetings and workshops

PHENIX User's Workshop, 22 September, 2011

A *PHENIX* user's workshop is being planned in Durham, North Carolina on the 22nd of September for local area students, postdocs and other interested parties. Please contact Jeff Headd at JJHeadd@lbl.gov for further information.

IUCr Commission on Crystallographic Computing, Mieres 2011, Crystallographic Computing School, 16-22 August, 2011

A crystallographic computing school run by the IUCr Commission on Crystallographic Computing will be held in Oviedo, Spain from the 16th to the 22nd of August 2011. *PHENIX* developers will be giving lectures and available for questions.

Expert advice

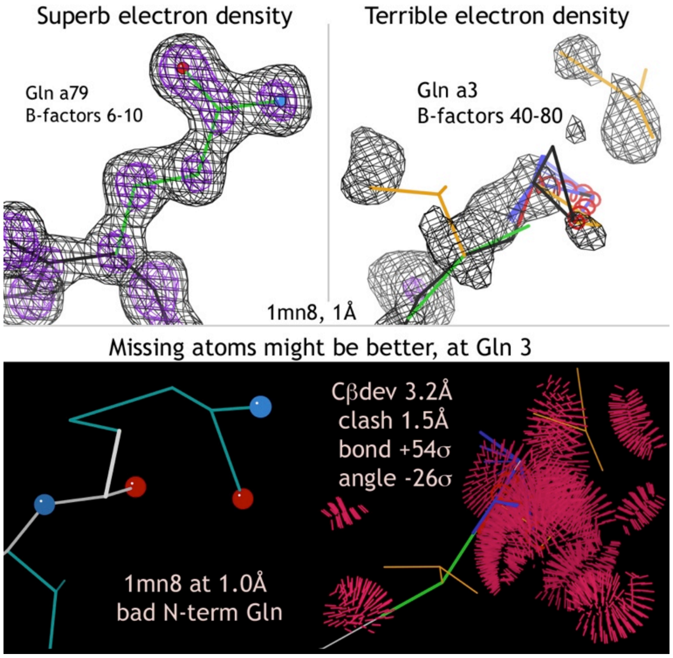
Fitting Tips

Vincent Chen, Christopher Williams and Jane Richardson, Duke University

Even very high-resolution structures are prone to a few types of systematic error that would be better avoided.

When you're lucky enough to be at 1Å resolution, the electron density is gorgeous, unambiguous and delightful in most places - like the upper left figure of Gln 79 in 1mn8 (2Fo-Fc with contours at 1.2 and 3.0 s). It is very tempting, then, to strongly down weight the geometry terms, or even turn them off altogether. That produces a good model in the well-ordered regions with low B-factors. [Note that normal weighting would also produce a good model there.]

However, even at ultra high resolution there are almost always a few disordered places with very poor density and high B-factors, such as at Gln 3 in the upper right figure. A reasonable peptide and sidechain were probably fit here initially. Then refinement tried too hard to move atoms into what little density it could find, resulting in the violently distorted model at lower left. As shown at upper and lower right, there are bond length outliers up to 56σ, bond angle outliers up to 26σ, a 3.2Å Cβ deviation, many steric clashes with all-atom overlap up to 1.5Å, 2 bad rotamers and a Ramachandran outlier. It seems clear that no one looked at this region in the final model, because surely they would have been motivated to do something about it.



This example is an extreme case, but not an unusual problem. The tips here are:

1) Keep a non-negligible weight on the geometry term (except perhaps in a local test that won't be deposited). B-factor dependent weights would be a desirable option.
2) Don't rely on overall rmsd for bond lengths and angles - always look at map and model for the worst individual deviations and check out the chain termini.
3) Perhaps residue 3 should have been omitted as well as 1 and 2. If you do choose to fit into very poor density, enforce acceptable geometry and conformation.

FAQ

How do I make composite omit maps in *PHENIX*?

This can be achieved using the GUI by choosing the "AutoBuild – create omit map" option under the "Maps" tab in the main GUI window. It is also very easy using the command line. To make a simple omit map of the model, the following options can be used with the Autobuild command:

```
phenix.autobuild data=data.mtz model=coords.pdb composite_omit_type=simple_omit
```

Coefficients for the output omit map will be in the file resolve_composite_map.mtz in the subdirectory OMIT/ . A simulated annealing omit map can be generated by changing the type:

```
phenix.autobuild data=data.mtz model=coords.pdb composite_omit_type=simple_omit
```

The region of the omit map can be specified by adding the "omit_box_pdb" option thus:

```
phenix.autobuild data=data.mtz model=coords.pdb composite_omit_type=simple_omit
                  omit_box_pdb=target.pdb
```

Once again, coefficients for the output omit map will be in the file resolve_composite_map.mtz in the subdirectory OMIT/ . An additional map coefficients file omit_region.mtz will show you the region that has been omitted. (Note: be sure to use the weights in both resolve_composite_map.mtz and omit_region.mtz).

More information about maps can be found online.

A lightweight, versatile framework for visualizing reciprocal-space data

Nathaniel Echols[a] and Paul D. Adams[a,b]

[a]Lawrence Berkeley National Laboratory, Berkeley, CA 94720
[b]Department of Bioengineering, University of California at Berkeley, Berkeley, CA 94720

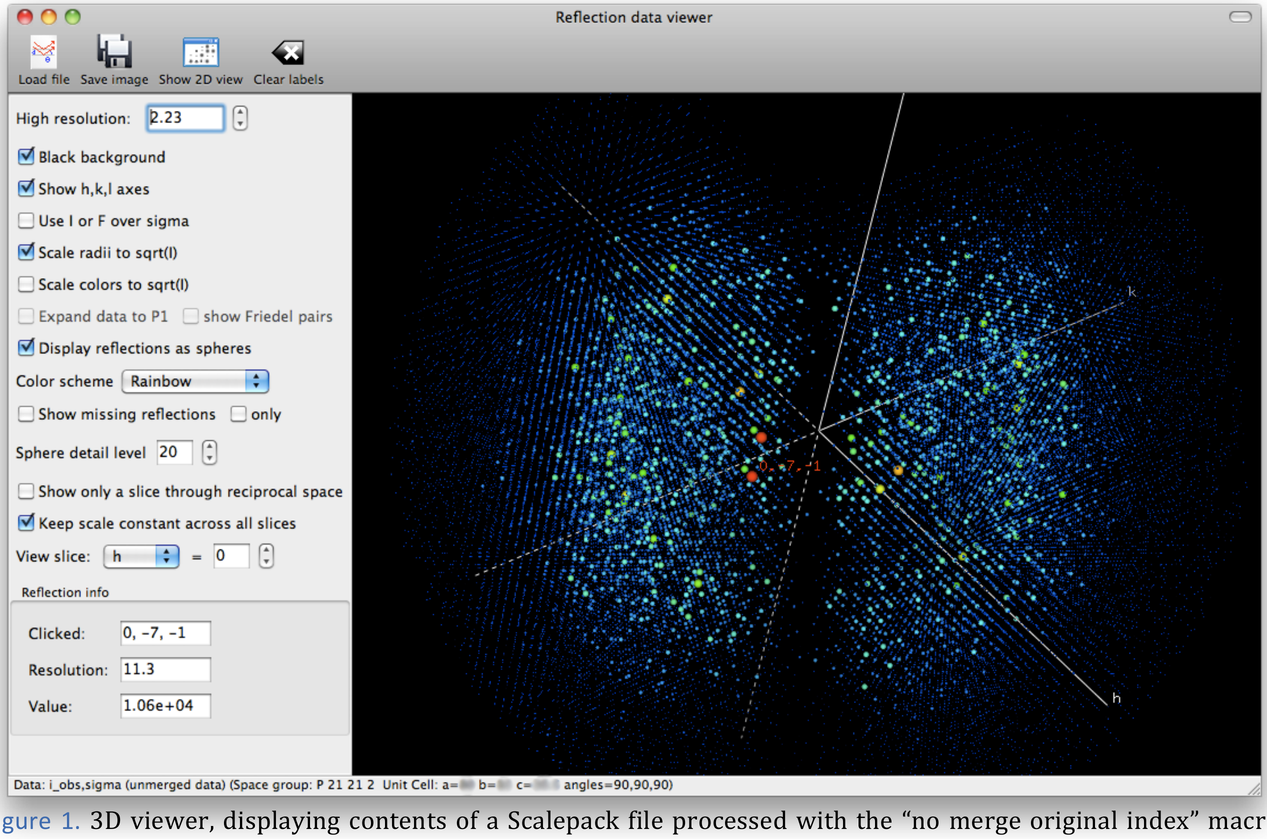
Correspondence email: nathaniel.echols@gmail.com

Introduction

For diagnostic and educational purposes, it is often useful to display data from reflection files in graphical format. In macromolecular crystallography, the CCP4 program hklview [1] has been the primary tool for this, but it is limited to 2D pseudo-precession camera "slices" through reciprocal space. Ongoing work on assessing data pathologies and improving refinement and map quality in PHENIX, especially at low resolution, necessitated the development of a simple program capable of both 2D and 3D views of reflection data.

The complete program, phenix.data_viewer, is written as a standalone wxPython app, but was designed to be easily embedded in other programs and potentially re-used in other contexts. The 3D viewer relies entirely on OpenGL for rendering, using a custom set of Python OpenGL bindings in the gltbx module of CCTBX. The 2D viewer uses low-level wxPython drawing commands on a blank canvas, with the underlying native graphics API performing the actual work. Both views support saving screen captures of the canvas in PNG format. In principle these frontends could be replaced with GUI-independent output formats, for instance using the GD drawing library [2], facilitating use in web servers.

The 2D and 3D displays are nearly identical with respect to input and options, but operate independently of each other. Both have a control panel with all user-adjustable parameters, plus information on the last clicked reflection (Figures 1 and 2). We have attempted to provide the user with



High resolution: 2.23

☑ Black background

☑ Show h,k,l axes

☐ Use I or F over sigma

☑ Scale radii to sqrt(I)

☐ Scale colors to sqrt(I)

☐ Expand data to P1 ☐ show Friedel pairs

☑ Display reflections as spheres

Color scheme: Rainbow

☐ Show missing reflections ☐ only

Sphere detail level: 20

☐ Show only a slice through reciprocal space

☑ Keep scale constant across all slices

View slice: h = 0

Reflection info

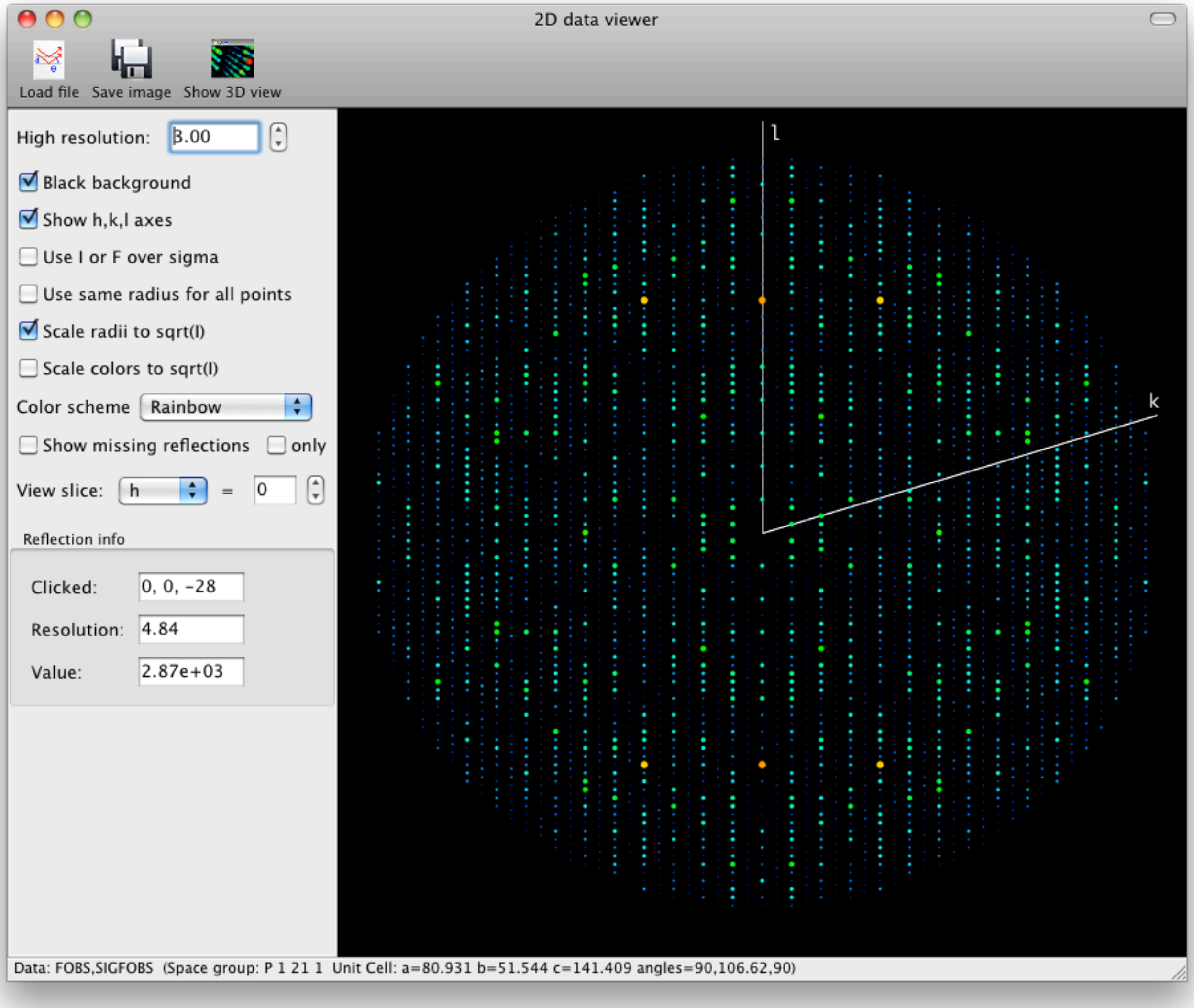
Clicked: 0, -7, -1

Resolution: 11.3

Value: 1.06e+04

Data: i_obs,sigma (unmerged data) (Space group: P 21 21 2 Unit Cell: a=[redacted] b=[redacted] c=[redacted] angles=90,90,90)

Figure 1. 3D viewer, displaying contents of a Scalepack file processed with the "no merge original index" macro. The dataset was collected as a 100-degree wedge with inverse beam geometry.



2D data viewer

High resolution: 3.00

☑ Black background

☑ Show h,k,l axes

□ Use I or F over sigma

□ Use same radius for all points

☑ Scale radii to sqrt(I)

□ Scale colors to sqrt(I)

Color scheme: Rainbow

□ Show missing reflections □ only

View slice: h = 0

Reflection info

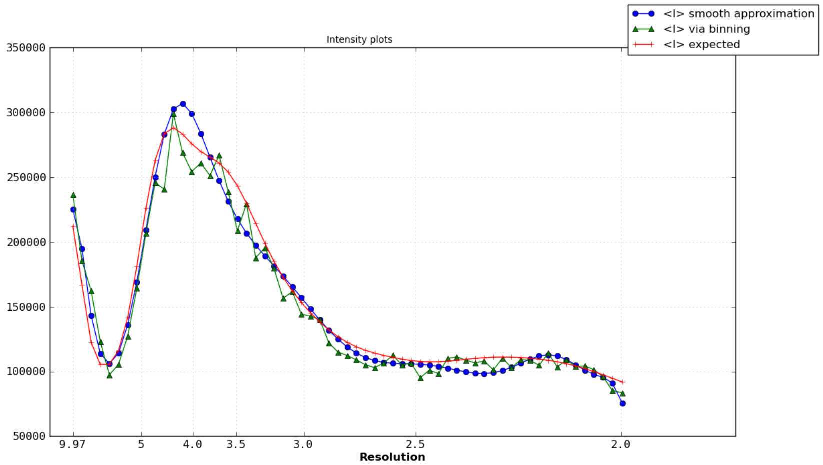
Clicked: 0, 0, -28

Resolution: 4.84

Value: 2.87e+03

Data: FOBS,SIGFOBS (Space group: P 1 21 1 Unit Cell: a=80.931 b=51.544 c=141.409 angles=90,106.62,90)

Figure 2. 2D viewer, showing the 0kl section from a dataset with pseudo-translational symmetry (PDB ID: **3ori**, truncated at 3.0Å resolution), resulting in alternating strong and weak columns of spots. The effect is especially noticeable when viewing successive sections along the *k* axis, as shown in Figure 4. Also clearly visible is the sharp dip in mean intensity around 7Å resolution characteristic of protein crystals; the Wilson plot from *phenix.xtriage* is shown below for comparison.



a large amount of control over how the data are displayed. By default, both point size and color are used to convey the relative magnitude of reflections, using a variety of scaling options. We have found this to be more intuitive than the monochromatic or grayscale rendering, especially in 3D where the number of reflections may be in the tens (or hundreds) of thousands. In addition to the reflections actually present in the input file, missing reflections may also be visualized (Figure 3), either alongside the real data or independently. This may be useful for judging errors and/or pathologies in data collection [3].

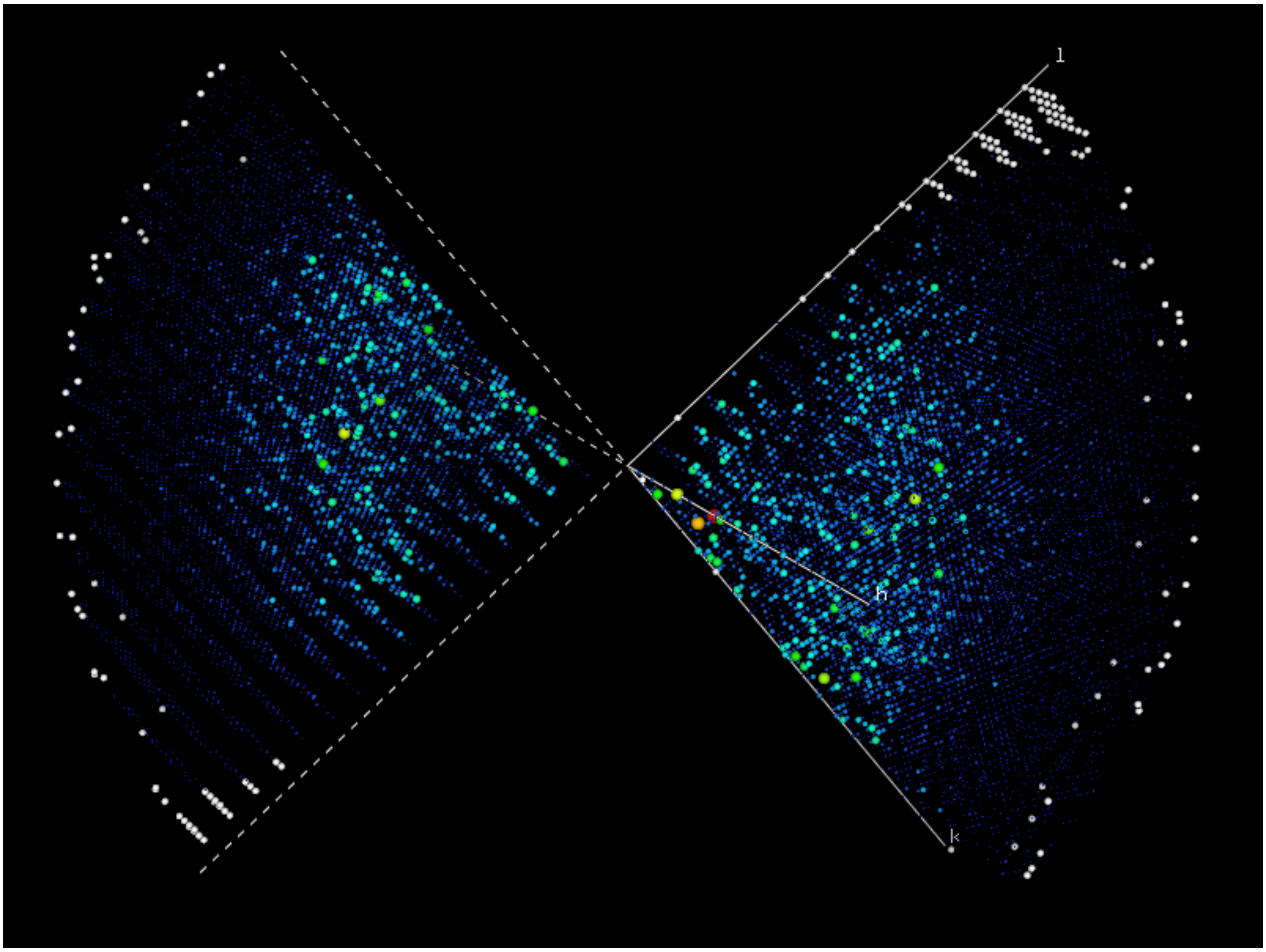


Figure 3. The same dataset shown in Figure 1 merged to contain only symmetry-unique data, with missing reflections displayed as white spheres.

Because CCTBX deals primarily with fully processed (merged and scaled) data, support for unmerged reflection files is currently uneven. By default, the 3D viewer displays only those reflections in the input file, although the controls allow expansion to P1 symmetry and generation of Friedel mates. The 2D view approximates the behavior of hklview: when unmerged data are provided, it will only display the original reflections, potentially leaving some regions empty, but automatically expands merged data to cover all of reciprocal space. Visualization of missing reflections in unmerged data is limited to the reciprocal-space asymmetric unit, which may lead to visual artifacts depending on the oscillation range (Figure 3).

Although visualizations of this sort are useful for interpreting many properties of reciprocal space, especially with regards to missing and/or pathological data (Figure 4), they are not intended to directly represent the data as they appear in the actual diffraction experiment [4]. In particular, the size of the spheres or circles representing individual reflections has no relationship to the apparent "size" of the reflection as captured on an area detector, which is actually determined by factors such as crystal mosaicity, beam divergence, etc. However, a possible future enhancement is the addition of an Ewald sphere and display of its intersection with the reflections as it would appear on an area detector, given user-defined parameters for mosaicity and other experimental properties.

Availability

phenix.data_viewer is included with all PHENIX installers starting with build 780, and can be run from

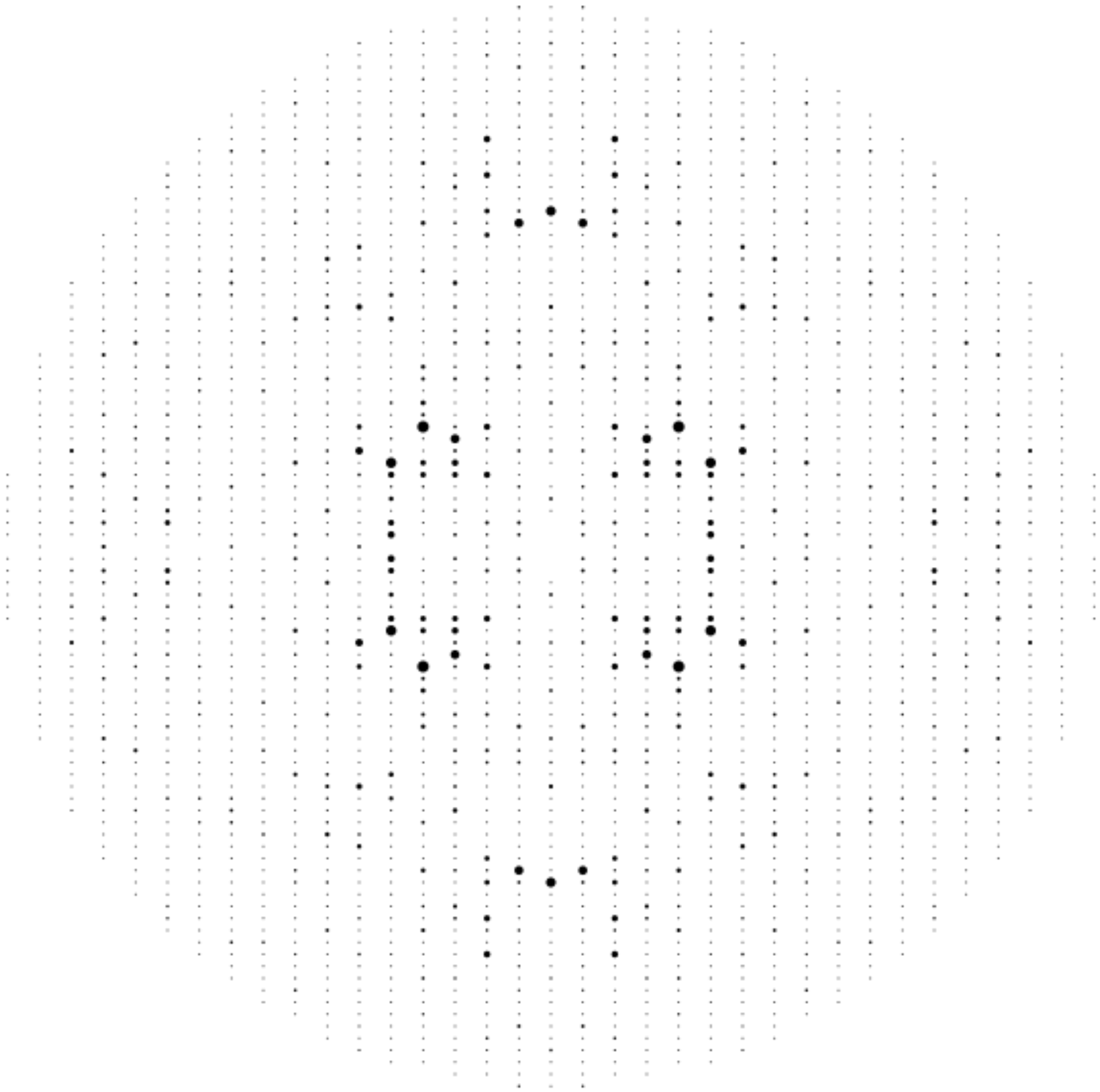


Figure 4. The h0l and h1l sections of the dataset with translational NCS shown in Figure 2. The images are displayed in black-and-white for clarity, but the effect is more striking using the default settings when viewed interactively.

the command line or from the PHENIX GUI (under "Reflection tools"). It is also available in standalone CCTBX builds, but must be compiled from source due to the wxPython and OpenGL dependencies. The code is available as unrestricted open-source under the CCTBX license, in the module `crys3d.hklview`.

Acknowledgments

We thank Jaroslaw Kalinowski for suggesting the 3D viewer concept.

Notes

1. Originally written by Phil Evans, and described at http://www.ccp4.ac.uk/html/hklview.html. Prof. Evans has called our attention to an excellent replacement for hklview called ViewHKL (available as a standalone download at http://www.ccp4.ac.uk/prerelease/) written by him and Eugene Krissinel.

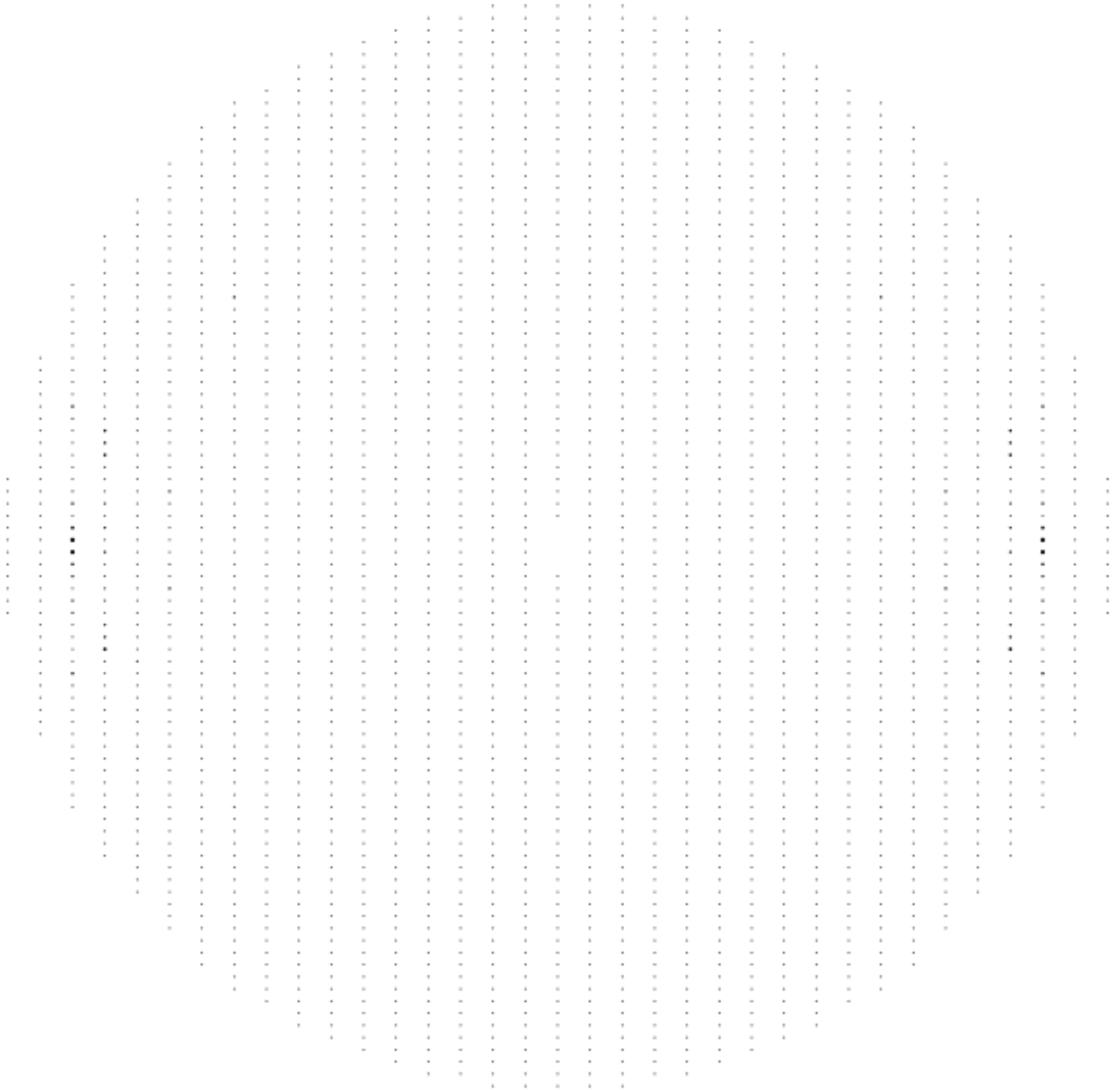


Figure 4. (continued)

Notes (continued)

2. http://newcenturycomputers.net/projects/gdmodule.html

3. See also Urzhumtsev, A. G. (1991). Acta Cryst. A47, 794-80 and Urzhumtseva & Urzhumtsev (2011) J. Appl. Cryst. vol. 44 part 4.

4. The program labelit.precession_photo, described in the previous issue of this newsletter, generates similar 2D slices of reciprocal space using the raw diffraction images directly.

An extremely fast spotfinder for real-time beamline applications

Nicholas K. Sauter

Lawrence Berkeley National Laboratory, Berkeley, CA 94720

Correspondence email: NKSauter@LBL.Gov

The Bragg spot analyzer described last year [*CCN* **1**, 18-23 (2010)] has been enhanced for high-throughput applications such as diffraction mapping and continual monitoring for radiation damage. The software provides raw measurements that can be harnessed by beamline developers for graphical display and instrument control. Recent work improves the program's output and performance.

Most significantly, the spotfinder package is now released under the cctbx open source license (see http://cctbx.sf.net), which now makes it independent of the packages *LABELIT* and *PHENIX*, and accessible to all beamlines worldwide. Succinct instructions for download and installation are posted at http://cci.lbl.gov/labelit, under the link for "Beamline Server". New cctbx code is tested, packaged and released on a near-daily basis; interested users are encouraged to either contact the authors with feature requests or join the cctbx open-source development group.

High-throughput performance is achieved by delegating the analysis of individual diffraction images to separate processors on a multicore CPU. The overall software architecture includes a "client" process (such as the beamline graphical user interface), which contacts the multiprocessing "server" whenever a Bragg spot analysis is required for a new image. The client, which is developed by the beamline group, can be implemented in any language (Java, TCL, Python, etc.) that supports the http: protocol needed to contact the server. In fact, it is straightforward to test the server with a standard Web browser, by requesting a URL that includes the file name of the diffraction image and any desired processing options. A simple mapping is used to convert the Unix command line for the underlying spotfinder program into a URL for the spotfinder server. Two separate implementations of the spotfinder server are now released, one using all-Python tools, and a second that uses the Apache httpd Web server for multiprocess control, within which a Python interpreter is provided by the mod-python package. The two servers give identical data analysis and similar performance, but there are some tradeoffs: the Python server is slightly easier to download and install, but the Apache/mod-python is superior in its ability to tune for peak performance. We observe the following general performance benchmarks under 64-bit Linux:

OS	Fedora 8	Fedora 13
CPU	Intel Xeon	AMD Opteron
Clock speed	2.93 GHz	2.20 GHz
# of processors	16 cores	48 cores
Overall throughput	8.9 frames/s	25 frames/s

These tests involved the processing of 720 Pilatus-6M images, with diffraction spots identified out to the corner of the detector.

Finally, many new features have been added to the spotfinder. Diffraction strength can now be summarized as a function of resolution bin, which should be of particular interest for monitoring Bragg spot quality over time from a given specimen. Additional quality measures have been added, such as background level, and signal-to-noise expressed as I/σ(I). Numerous additional options are available for controlling the algorithm, all of which are documented on the Web page. The spotfinder work was funded under NIH/NIGMS grant numbers R01GM077071 and R01GM095887.

Hints for running *phenix.mr_rosetta*

Thomas C. Terwilliger[1], Frank DiMaio[2], Randy J. Read[3], David Baker[2], Gábor Bunkóczi[3], Paul D. Adams[4], Ralf W. Grosse-Kunstleve[4], Pavel V. Afonine[4], Nathaniel Echols[4]

1 Los Alamos National Laboratory, BioScience Division and Los Alamos Institutes, Los Alamos, NM 87545
2 University of Washington, Department of Biochemistry, Seattle, WA, 98195, USA
3 University of Cambridge, Department of Haematology, Cambridge Institute for Medical Research, Cambridge, CB2 0XY, UK
4 Lawrence Berkeley National Laboratory, One Cyclotron Road, Bldg 64R0121, Berkeley, CA 94720, USA

Introduction

A combination of structure-modeling tools available in *Rosetta* (Qian et al., 2007, DiMaio et al., 2009) and the molecular replacement (Read, 2001) and model-building (Terwilliger et al, 2008) tools available in *Phenix* (Adams et al., 2010) has been very useful in determining structures by molecular replacement (DiMaio et al., 2011). The approach is most appropriate for cases where the best template is somewhat too different from the target structure to be useful in conventional molecular replacement. In a previous *Phenix* newsletter we summarized the *phenix.mr_rosetta* tool and how to use it. The basic idea is that *Rosetta* modeling can be useful at two stages in molecular replacement. First, it can be useful in improving a template before using it as a search model. Second, it can be useful in improving a model that has been placed in the unit cell and where an electron density map is available. By combining *Rosetta* with *Phenix* tools, the range of models useful for molecular replacement can be expanded. Here we give some hints for getting the most out of this approach.

Downloading templates from the PDB based on an alignment file

One useful feature of *phenix.mr_rosetta* is the ability to use an alignment file that lists templates available in the PDB and alignments of those templates to the target structure. These alignment files can be obtained from the *hhpred* server (http://toolkit.tuebingen.mpg.de/hhpred; Soding, 2005). You can supply one or more alignment files to *phenix.mr_rosetta* with the keyword

```
hhr_files=my_hhr_file
```

and you can specify how many of the templates in each file (e.g. 5) are to be used with

```
number_of_models=5
```

It is a good idea to have a close look at the alignment file and choose how many models to download based on the range of sequence identities in the file. If there are a few models with high (>40%) identities, just use those. If there are many templates with similar sequence identities (and over similar parts of the target sequence) then you might want to include many of them, particularly if the sequence identity is low (<25%).

If you want to have more control over your search models, then you can download them yourself from the PDB and edit them with the *phenix.sculptor*. Alternatively you can download and edit them simultaneously with *phenix.mr_model_preparation.* Then you can specify these as search models with

```
search_models="model1.pdb model2.pdb"
```

and *phenix.mr_rosetta* will use each of these in turn as a search model.

Automatic searching for multiple NCS copies

You can control whether *phenix.mr_rosetta* checks for variable numbers of NCS copies in the asymmetric unit with the parameter

```
ncs_copies="1 2 4"
```

which will instruct *phenix.mr_rosetta* to search (in separate runs) for 1, 2, and 4 copies. You can also say,

```
ncs_copies=None
```

which will try all plausible values of ncs_copies. This can be convenient, but you might want to instead run 3 separate runs, specifying one value of ncs_copies in each. The reason this may be a good idea is that *phenix.mr_rosetta* does not stop when a satisfactory solution is found. Instead it will complete all the jobs and then report the best one. So if the job with `ncs_copies=4` takes a really long time (as it might if there are not actually 4 copies) then the whole *phenix.mr_rosetta* job would take a long time to complete.

Improving templates with *Rosetta* to use as search models in molecular replacement

One of the uses of *phenix.mr_rosetta* is to carry out homology-modeling of a template before using it as a search model. You can do this automatically during a *phenix.mr_rosetta* run with the keywords

```
run_prerefine=True
number_of_prerefine_models=1000
```

Typically you would want to generate about 1000-2000 models with *Rosetta.* Then the best model will be used in the following steps. Generating models at this stage with Rosetta does not take too long; a 150-residue protein might take about 5 minutes for each model.

Note that it is best to specify the number of `ncs_copies` if you use `run_prerefine`. If you do not, then you may end up running several parallel jobs, each of which is independently carrying out prerefinement on the same input model (to be used later with different numbers of ncs copies). Once you have run your job with one value of `ncs_copies`, you can just use the best prerefined model from that job as a search model in your other runs.

If you just want to run *Rosetta* rebuilding on a template and you don't want to do anything else, you can use a simple command to do this:

```
phenix.mr_rosetta \
  seq_file=seq.dat \
  search_models=coords1.pdb \
  run_prerefine=True \
  number_of_prerefine_models=1
```

Your prerefined model(s) will be listed in

```
MR_ROSETTA_1/GROUP_OF_PLACE_MODEL_1/RUN_FILE_1.log
```

and you can pick the best of these (most negative score, listed first).

Fragment files for *Rosetta*

If your model has gaps in it, then you will need to provide fragment files for *Rosetta* to use in filling in those gaps. If your chain has 650 residues or fewer, then this is fairly straightforward, and you can paste your sequence into the *Robetta* fragment server (http://robetta.bakerlab.org/fragmentsubmit.jsp; Chivian et al., 2003).

If your chain has more than 650 residues then you will need to break it up into segments and submit separate requests to the fragment server for each segment. Then you will get several 3-mer and 9-mer fragments files, one for each piece that you submit. You can then simply paste these together after editing all but the first to fix the residue numbers. To edit the files just use

```
phenix.offset_robetta_resid \
  <fragment_file_name> \
  <new_fragment_file_name> \
  <offset-for-residue numbers>
```

If you have multiple chain types in your structure then you will want to have a separate set of fragments files for each chain type. You can specify these with the keywords *fragment_files_chain_list*, *fragment_files_3_mer_by_chain*, and *fragment_files_9_mer_by_chain* instead of the keyword *fragment_files*.

Use *fragment_files_chain_list* to define which chain ID each of your *fragment_files_3_mer_by_chain* and *fragment_files_9_mer_by_chain* go with. Note that you only need one set of fragments files for each unique chain. So if chains A and C are the same, you just need to specify fragments for chain A.

Testing your installation of Rosetta and *phenix.mr_rosetta*

As a run of *phenix.mr_rosetta* can take a long time (hours to days or even weeks depending on how many models you search for and how many processors you have available) you may want to make sure everything is working properly before you start. You can test that both *Rosetta* and *phenix.mr_rosetta* work properly with the command

```
phenix_regression.wizards.test_command_line_rosetta_quick_tests
```

This takes about 15 minutes and will end with "OK" if everything is all right.

Running *phenix.mr_rosetta* on a cluster

You probably will want to run *phenix.mr_rosetta* on a cluster as it can take so much computational time to run. You can run on a Sun Grid Engine, Condor, or other cluster. If you run on a Sun Grid Engine (SGE) cluster, you only need to specify two keywords. The first tells *phenix.mr_rosetta* how to submit a job:

```
group_run_command=qsub
```

If your job submission is more complicated you can specify that:

```
group_run_command="/etc/run/qsub -abc"
```

You can then specify how many processers are to be used:

```
nproc=200
```

Note that *phenix.mr_rosetta* will submit individual jobs to the queue, not array jobs. This means that many jobs may be submitted.

On a condor cluster, you can specify

```
group_run_command=condor_submit
```

instead of "qsub".

On other clusters and supercomputers job submission may be more complicated. However you can control how it is done with the group_run_command keyword and with the keyword queue_commands. For example on a PBS system you

```
queue_commands='#PBS -N mr_rosetta'
queue_commands='#PBS -j oe'
queue_commands='#PBS -l walltime=03:00:00'
queue_commands='#PBS -l nodes=1:ppn=1'
```

When *phenix.mr_rosetta* actually submits a job, these commands will appear at the top of the script that is submitted (just after the definition of the shell to use), like this:

```
#!/bin/sh
#PBS -N mr_rosetta
#PBS -j oe
#PBS -l walltime=03:00:00
#PBS -l nodes=1:ppn=1
cd /home/MR_ROSETTA_3/GROUP_OF_PLACE_MODEL_1
sh /home/ MR_ROSETTA_3/GROUP_OF_PLACE_MODEL_1/RUN_FILE_1.sh
```

Finding your results with *phenix.mr_rosetta*

When *phenix.mr_rosetta* has completed you can find the best model and map by looking at the end of the log file that has been written. You should see something like:

```
Results after repeat_mr_rosetta:

ID: 306
R/Rfree:   0.24 /   0.27
MODEL:
/net/omega/raid1/scratch1/terwill/blind_tests/all_cases/mr_rosetta_from_sta
rt/1_ag9603/MR_ROSETTA_6/ONE_REPEAT_1/RUN_1/GROUP_OF_AUTOBUILD_1/RUN_2/Auto
Build_run_1_/cycle_best_2.pdb

MAP COEFFS
/net/omega/raid1/scratch1/terwill/blind_tests/all_cases/mr_rosetta_from_sta
rt/1_ag9603/MR_ROSETTA_6/ONE_REPEAT_1/RUN_1/GROUP_OF_AUTOBUILD_1/RUN_2/Auto
Build_run_1_/cycle_best_2.mtz

Writing solutions as csv to
/net/omega/raid1/scratch1/terwill/blind_tests/all_cases/mr_rosetta_from_sta
rt/1_ag9603/MR_ROSETTA_6/repeat_results.csv
Saved overall mr_rosetta results in
/net/omega/raid1/scratch1/terwill/blind_tests/all_cases/mr_rosetta_from_sta
rt/1_ag9603/MR_ROSETTA_6/repeat_results.pkl

To see details of these results type
    phenix.mr_rosetta
mr_rosetta_solutions=/net/omega/raid1/scratch1/terwill/blind_tests/all_case
s/mr_rosetta_from_start/1_ag9603/MR_ROSETTA_6/repeat_results.pkl
display_solutions=True
```

This will be the model with the lowest R value obtained. If you want to see information about all the models (including intermediate models produced) then you can use the command that is listed at the end of this run:

```
 phenix.mr_rosetta
mr_rosetta_solutions=/net/omega/raid1/scratch1/terwill/blind_tests/all_case
s/mr_rosetta_from_start/1_ag9603/MR_ROSETTA_6/repeat_results.pkl
display_solutions=True
```

If the *phenix.mr_rosetta* run involved more than one cycle it will say " Results after repeat_mr_rosetta: " (as in the case above). In this case the list of solutions obtained with the above command will only include results from the repeat cycle.

To obtain results from earlier stages, look earlier in the log file to the place where the word "RESULTS OF AUTOBUILDING" (in capitals) first appears and then search down to the next `display_solutions` command:

```
=================================================================
RESULTS OF AUTOBUILDING:
=================================================================

ID: 222
R/Rfree:   0.25 /   0.28
MODEL:
/net/omega/raid1/scratch1/terwill/blind_tests/all_cases/mr_rosetta_from_sta
rt/1_ag9603/MR_ROSETTA_6/GROUP_OF_AUTOBUILD_1/RUN_2/AutoBuild_run_1_/cycle_
best_4.pdb
MAP COEFFS
/net/omega/raid1/scratch1/terwill/blind_tests/all_cases/mr_rosetta_from_sta
rt/1_ag9603/MR_ROSETTA_6/GROUP_OF_AUTOBUILD_1/RUN_2/AutoBuild_run_1_/cycle_
best_4.mtz

Writing solutions as csv to
```

```
/net/omega/raid1/scratch1/terwill/blind_tests/all_cases/mr_rosetta_from_sta
rt/1_ag9603/MR_ROSETTA_6/autobuild_results.csv

Saved overall mr_rosetta results in
/net/omega/raid1/scratch1/terwill/blind_tests/all_cases/mr_rosetta_from_sta
rt/1_ag9603/MR_ROSETTA_6/autobuild_results.pkl

To see details of these results type
    phenix.mr_rosetta
mr_rosetta_solutions=/net/omega/raid1/scratch1/terwill/blind_tests/all_case
s/mr_rosetta_from_start/1_ag9603/MR_ROSETTA_6/autobuild_results.pkl
display_solutions=True
```

where the appropriate command will be listed.

When you print out a list of solutions in this way, each solution is listed along with the lineage of that solution (all the solutions obtained on the path to this solution).

Restarting *phenix.mr_rosetta* if something goes wrong

As *phenix.mr_rosetta* completes each stage, it writes out a file that contains all the information needed to go on from that stage. In the examples above where the command `display_solutions=True` is used, this file is read and the information is simply printed. If you want to use this information to carry on, then you need to specify two things. First you need to name the file containing the solutions you want to use. This file is listed in your log file as in the examples above, and a file is written out after each major step. You specify it with:

```
mr_rosetta_solutions=working_solutions.pkl
```

Second you need to specify where to start. You can do this with the keyword "start_point":

```
start_point=rosetta_rebuild
```

This will start with *Rosetta* rebuilding with density (provided you have supplied solutions that include the previous step, `rescore_mr`).

Acknowledgments

We gratefully acknowledge the financial support of NIH/NIGMS under grant number P01GM063210. Our work was supported in part by the US Department of Energy under Contract No. DE-AC02-05CH11231.

References

Adams P.D., Afonine P.V., Bunkoczi G., Chen V.B., Davis I.W., Echols N., Headd J.J., Hung L.W., Kapral G.J., Grosse-Kunstleve R.W., McCoy A.J., Moriarty N.W., Oeffner R., Read R.J., Richardson D.C., Richardson J.S., Terwilliger T.C., and Zwart P.H.. (2010). Acta Cryst. D66, 213-221.

Chivian D, Kim DE, Malmstrom L, Bradley P, Robertson T, Murphy P, Strauss CEM, Bonneau R, Rohl CA, Baker D. (2003) Automated prediction of CASP-5 structures using the Robetta server. Proteins 53 Suppl 6:524-33

DiMaio F., Tyka,M.D., Baker, M.L., Chiu,W., Baker, D. (2009). Refinement of protein structures into low-resolution density maps using rosetta. Journal of Molecular Biology 392: 181-190.

DiMaio, F., Terwilliger, T.C., Read, R.J., Wlodawer, A., Oberdorfer, G., Wagner, U., Valkov, E., Alon, A., Fass, D., Axelrod, H.L., Das, D., Vorobiev, S.M., Iwai, H., Pokkuluri, P.R., Baker, D. (2011). Improving molecular replacement by density and energy guided protein structure optimization Nature 473, 540-543.

Qian, B., Raman, S., Das, R., Bradley, P., McCoy, A.J., Read, R.J., and Baker, D. (2007). High resolution structure prediction and the crystallographic phase problem. Nature 450, 259-264.

Read, R.J. (2001). Pushing the boundaries of molecular replacement with maximum likelihood. Acta Cryst. D 57, 1373-1382.

Söding J. (2005) Protein homology detection by HMM-HMM comparison. Bioinformatics 21, 951-960.

Terwilliger, T.C., Grosse-Kunstleve, R.W., Afonine, P.V., Moriarty, N.W., Zwart, P.H., Hung, L.W., Read, R.J., and Adams, P.D. (2008). Iterative model building, structure refinement and density modification with the PHENIX AutoBuild wizard. Acta Cryst. D 64, 61-69.

Improved target weight optimization in *phenix.refine*

Pavel V. Afonine[1], Nathaniel Echols[1], Ralf W. Grosse-Kunstleve[1], Nigel W. Moriarty[1] and Paul D. Adams[1,2].
[1]*Lawrence Berkeley National Laboratory, One Cyclotron Road, MS64R0121, Berkeley, CA 94720 USA*
[2]*Department of Bioengineering, University of California Berkeley, Berkeley, CA, 94720, USA.*
Correspondence email: PAfonine@lbl.gov

Abstract

Restrained refinement of individual atomic coordinates and atomic displacement parameters combines experimental observations with prior knowledge. The two contributions need to be properly weighted with respect to each other in order to obtain the best results. This article describes a new target weight determination procedure in *phenix.refine* and presents the results of systematic tests on structures with lower resolution data.

Introduction

In *phenix.refine* (Afonine et al., 2005, Adams et al., 2010) the refinement of individual atomic coordinates or individual atomic displacement parameters (ADP, also known as B-factors) involves the minimization of a refinement target function that includes prior chemical or empirical knowledge. In the case of individual coordinate refinement this target function T is defined as:

$$T = w_{xc} \cdot w_{xc\_scale} \cdot T_{data} + w_c \cdot T_{geo\_restraints} \quad (1)$$

T_{data} is the target function quantifying the fit of experimental observations (X-ray and/or neutron data) and model-based predictions, using, for example, a least-squares or maximum-likelihood function. $T_{geo\_restraints}$ quantifies the fit of current model geometry (such as bonds, angles, dihedrals and nonbonded interactions) to tabulated "ideal" geometry, for example as inferred from high-resolution diffraction experiments. The three weight factors w_{xc}, w_{xc_scale} and w_c are redundant; equation (1) could be reformulated with only one weight factor. However, the formulation with three weight factors is helpful in practice. This is also true for the analogous formulation used in ADP refinement:

$$T = w_{xu} \cdot w_{xu\_scale} \cdot T_{data} + w_u \cdot T_{ADP\_restraints} \quad (2)$$

The weight factors w_c and w_u are usually one, but can be set to zero for unrestrained refinement. The weights w_{xc} and w_{xu} are determined automatically as described by Brünger *et al.* (1989) and Adams *et al.* (1997), using the ratio of the gradient norms after removing outliers:

$$w_{xc} = \sqrt{\frac{\langle \nabla T_{geo\_restraints}{}^2 \rangle}{\langle \nabla T_{data}{}^2 \rangle}} \qquad (3)$$

$$w_{xu} = \sqrt{\frac{\langle \nabla T_{ADP\_restraints}{}^2 \rangle}{\langle \nabla T_{data}{}^2 \rangle}} \qquad (4)$$

w_{xc_scale} and w_{xu_scale} are empirical scale factors, usually with values between 0.5 and 1.0.

An automatic weight determination procedure based on equations (3) and (4) has been used in *phenix.refine* from the beginning of its development. The procedure is usually reliable at typical macromolecular resolutions (around 1.5-2.5 Å) but sometimes problematic at significantly lower (> 3 Å) or higher (< 1.5 Å) resolutions. Typical problems are unexpectedly high R_{free} values, large gaps between R_{free} and R_{work}, unreasonably large geometry deviations from ideality, high Molprobity clash-scores, or large differences between ADPs of bonded atoms.

Brünger (1992) described a procedure that systematically searches for the weight leading to the lowest R_{free}. Until recently, the implementation in *phenix.refine* used an array of 10-20 values for w_{xc_scale} or w_{xu_scale}, with values distributed between 0.05 and 10. A full trial refinement was performed for each weight. In our experience, using R_{free} as the only guide for determining the optimal weight can sometimes discard results that are clearly more preferable if other quality measures are also taken into account. For example, R_{free} may oscillate only slightly while R_{work}, bond and angle deviations, or clash-scores change significantly. In this article we describe an enhanced weight search procedure that makes active use of an ensemble of quality measures.

Methods

In contrast to the previously used procedure, the new procedure in *phenix.refine* examines trial weights on an absolute scale:

$$T = w_{trial} \cdot T_{data} + T_{restraints} \qquad (5)$$

Since *phenix.refine* uses normalized targets for data and restraints, the range of plausible values is predictable. For example, the amplitude-based ML target (Lunin & Skovoroda, 1995; Afonine *et al.*, 2005) typically yields values that fall in the range between 1 and 10 (depending on resolution, data quality and model quality). The weight optimization procedure is parameterized with a spectrum of trial weights that is sufficiently large to offset such variations in the scale of T_{data}.

The new procedure executes the following steps:

1. For each trial weight, perform 25 iterations of LBFGS minimization (Liu & Nocedal, 1989) and save R_{work}, R_{free}, R_{free} - R_{work}. For coordinate refinement, also save bond and angle RMSDs and the clash-score. For ADP refinement, also save the mean difference between B-factors of bonded atoms <ΔB_{ij}>.
2. Select the subset of plausible results corresponding to R_{free} values in the range [R_{free}^{min}, R_{free}^{min} + Δ], where Δ is a resolution-dependent value in the range from 0 (high resolution) to 2% (low resolution) and R_{free}^{min} is the smallest R_{free} value obtained in step 1.
3. Reduce the subset further by applying selection criteria based on the R_{free} - R_{work} difference and bond and angle RMSDs (coordinate refinement) or <ΔB_{ij}> (ADP refinement).
4. In the case of coordinate refinement, reduce the subset further based on the clashscores (c). The first step is to the select results that satisfy the condition $\bar{c}/3 < c < 3\bar{c}$. For the second step recompute the mean $\bar{c}_{new}$ for the new subset and select results in the range from the minimum of the clashscores, c_{new}^{min}, to $c_{new}^{min} + w_{cs} * \bar{c}_{new}$. Currently the default value for w_{cs} is 0.1.
5. For the remaining subset select the result that corresponds to the lowest R_{free}.

The choice of Δ values in step 2 is based on the evaluation of a large number of refinements. We selected a number of data/model pairs covering a range of resolutions. For each pair we ran multiple refinements with identical parameters, except for the random seed used in the target weight determination and the simulated annealing module. In another series of tests, we applied modest random shifts to coordinates and *B*-factors before refinement. An ensemble of similar solutions is obtained for each data/model pair. Identical solutions cannot be expected (for example, see Terwilliger *et al.*, 2007) because the refinement target function is very complex and populated with many local minima; therefore the starting point is important. In addition, the structural deviations can be a consequence of static or dynamic disorder that is difficult to model. The Δ values reflect typical R_{free} fluctuations we observed in our refinement results, ranging from small fractions of a percent for high-resolution refinements and approaching 2 percentage points in a few low-resolution cases.

The optimal value for <ΔB_{ij}> in step 3 is not clearly defined, as discussed in Afonine *et al.* (2010a). Our current working estimate is 0.1<*B*>, where <*B*> is the average *B*-factor.

The weight optimization procedure is easy to parallelize since the refinements with different trial weights are independent. Starting with PHENIX version dev-810, the `refinement.main.nproc` parameter is available to specify the number of CPUs the weight optimization procedure may use in parallel. To give one example, the command

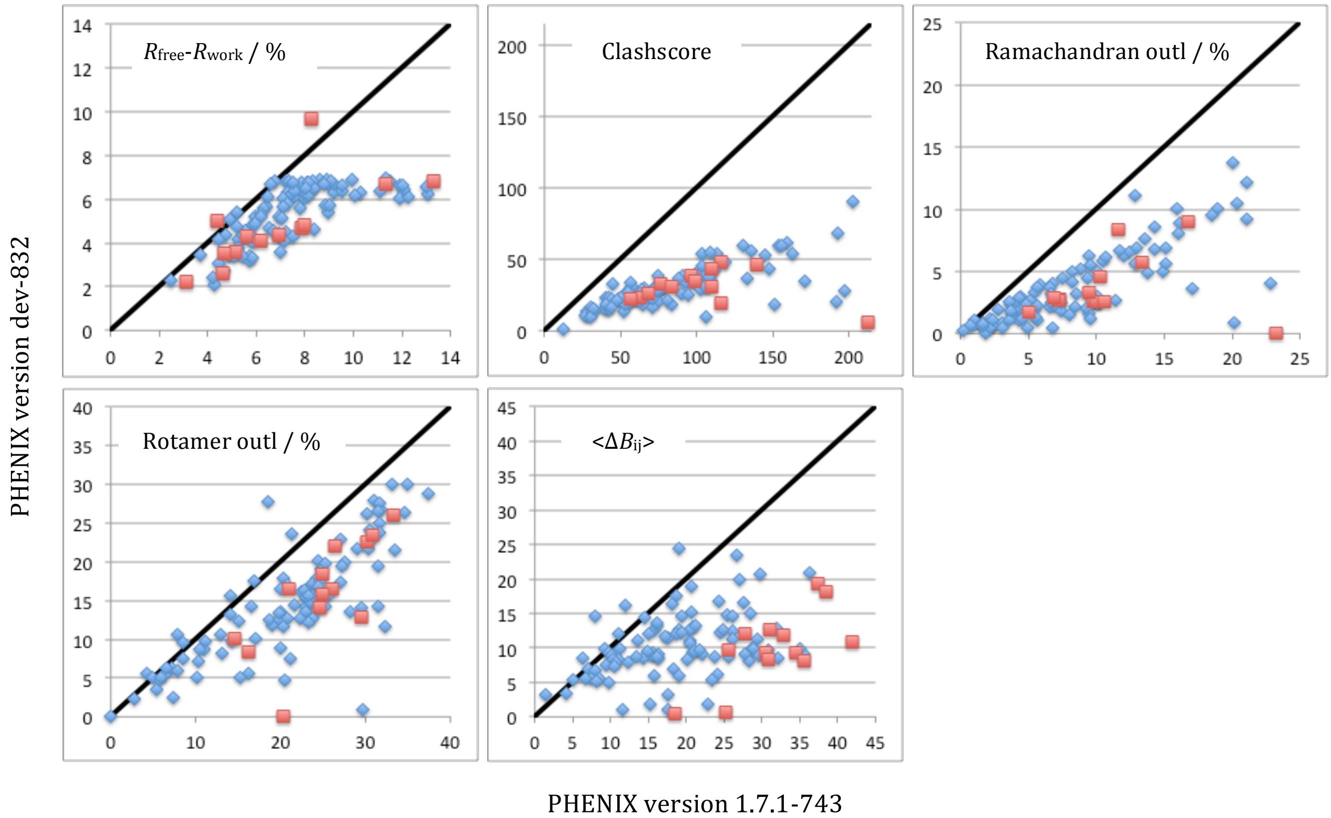
```
phenix.refine 1av1.pdb 1av1.mtz \
  optimize_xyz_weight=True \
  optimize_adp_weight=True nproc=16
```

finishes in approximately 430 seconds on a 48-core 2.2GHz AMD Opteron system. With `nproc=1` the refinement requires more than 2000 seconds on the same machine.

Results and discussion

To evaluate the new procedure we selected a set of low-resolution structures from the PDB (Bernstein *et al.*, 1977; Berman *et al.*, 2000) using on the following criteria:

- data high resolution limit between 3.5 and 4.5 Å,
- data completeness (overall and 6 Å – inf) better than 85%,
- data collected from untwinned crystals,



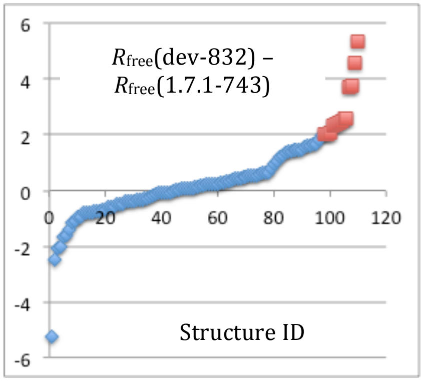
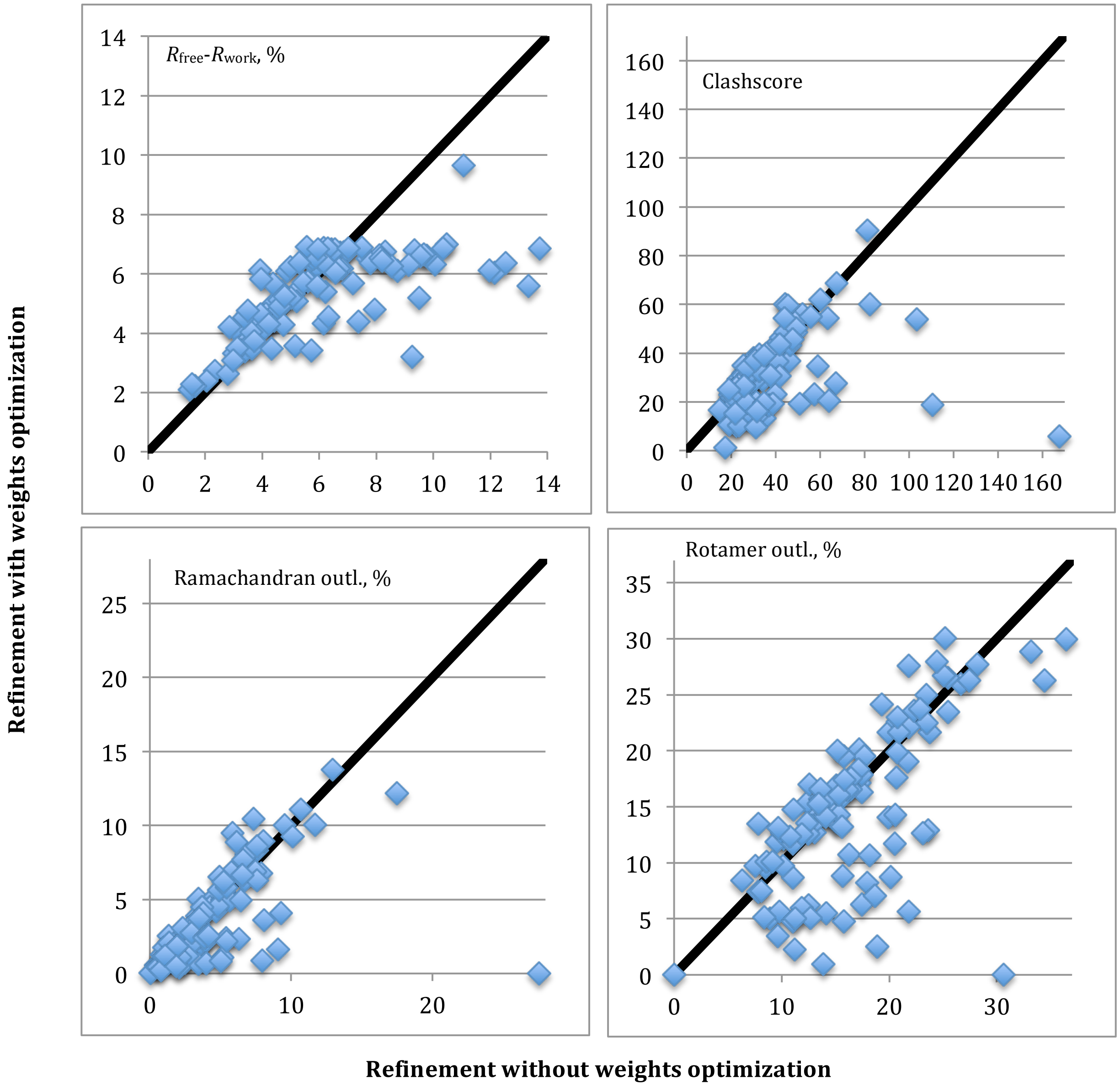


Figure 1. Comparison of refinement results between two PHENIX versions 1.7.1-743 and dev-832 (see text for details). Red squares highlight cases where the difference in R_{free} was larger than 2 percentage points (see the last plot where the results are ordered by R_{free} difference between the runs).

- data extractable from the PDB archive using *phenix.cif_as_mtz* (Afonine *et al.*, 2010b),
- models consisting only of common protein residues, ligands, heavy atoms and water,
- non-zero occupancy for all atoms,
- R_{free} flags available and a minimal gap between R_{free} and R_{work} of more than 2 percentage points.

We found 108 matching structures. Each structure was refined with 5 macro-cycles of restrained refinement of individual coordinates and ADPs (Afonine *et al.*, 2005). Most selected structures contain NCS-related molecules. NCS-related groups were determined automatically by *phenix.refine* and restrained in Cartesian space (atom-pair-wise harmonic restraints to the group average). Refinements were performed using the old weight optimization procedure as available in PHENIX (Adams *et al.*, 2002; Adams *et al.*, 2010) version 1.7.1-743 and the new procedure as included in a current development version (dev-832). The detailed results are presented in Figures 1 and 2.

Figure 1 compares refinement results (R_{free} - R_{work} difference, clash-score, <ΔB_{ij}>, percent of rotamer outliers and Ramachandran plot outliers) between *phenix.refine* runs using PHENIX versions 1.7.1-743 and dev-832. The results show clearly that the new procedure provides much improved geometry statistics and lower <ΔB_{ij}> values in most cases. The reduction of Ramachandran and rotamer outliers is especially noteworthy, since these quality measures are not directly used in the target weight optimization. The R_{free} comparison shows that the deviations are mostly within the Δ parameter range (2 percentage points), as



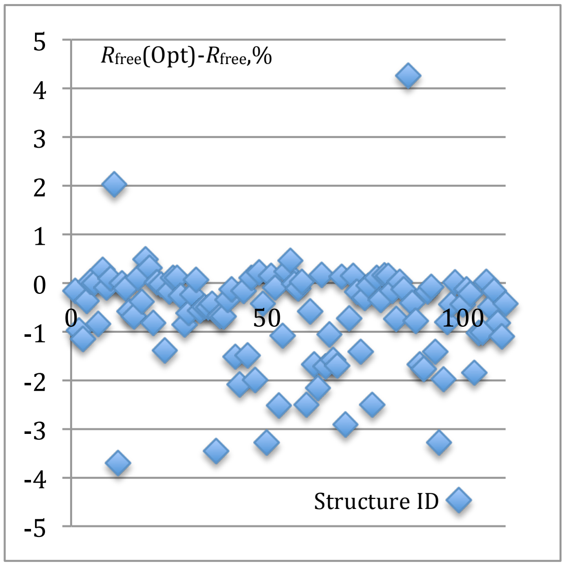


Figure 2. Results of refinement with and without weights optimization using PHENIX version dev-832.

expected. The few outliers with differences larger than 2 percentage points (red squares on Fig. 1) may be due to non-optimal NCS group selections that require further analysis, or <ΔB_{ij}> values that were forced to obey the requested limit. The large number of Ramachandran plot outliers may also indicate problems with the starting models that are beyond the anticipated convergence radius of these refinement procedures. Examining these cases in detail may lead to further improvements.

Figure 2 shows a comparison of statistics similar to Figure 1, after refinement with and without weight optimization using the current PHENIX development version (dev-832 or later).

References

Adams, P. D., Afonine, P. V., Bunkoczi, G., Chen, V. B., Davis, I. W., Echols, N., Headd, J. J., Hung, L. W., Kapral, G. J., Grosse-Kunstleve, R. W., McCoy, A. J., Moriarty, N. W., Oeffner, R., Read, R. J., Richardson, D. C., Richardson, J. S., Terwilliger, T. C. & Zwart, P. H. (2010). *Acta Cryst. D* **66**, 213-221.

Adams, P. D., Grosse-Kunstleve, R. W., Hung, L. W., Ioerger, T. R., McCoy, A. J., Moriarty, N. W., Read, R. J., Sacchettini, J. C., Sauter, N. K. & Terwilliger, T. C. (2002). *Acta Cryst. D* **58**, 1948-1954.

Adams, P. D., Pannu, N. S., Read, R. J. & Brünger, A. T. (1997). *Proc. Natl. Acad. Sci*. **94**, 5018-5023.

Afonine, P. V., Grosse-Kunstleve, R. W. & Adams, P. D. (2005). *Acta Cryst.* D**61**, 850-855.

(a)Afonine, P.V., Urzhumtsev, A., Grosse-Kunstleve, R.W. & Adams, P.D. (2010). *Computational Crystallography Newsletter*. 1, 24-31.

(b)Afonine, P.V., Grosse-Kunstleve, R.W., Chen, V.B., Headd, J.J., Moriarty, N.W., Richardson, J.S., Richardson, D.C., Urzhumtsev, A., Zwart, P.H. & Adams, P.D. (2010). *J. Appl. Cryst*. **43**, 669-676.

Afonine, P. V., Grosse-Kunstleve, R.W. & Adams, P.D. (2005). *CCP4 Newsletter on Protein Crystallography* **42 (8)**.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. & Bourne, P. E. (2000). *Nucleic Acids Res* **28**, 235-242.

Bernstein, F. C., Koetzle, T. F., Williams, G. J. B., Meyer, E. F., Brice, M. D., Rodgers, J. R., Kennard, O., Shimanouchi, T. & Tasumi, M. (1977). *J Mol Biol* **112**, 535-542.

Brünger, A. T. (1992). *Nature* **355**, 472-475.

Brünger, A.T., Karplus, M. & Petsko, G.A. (1989). *Acta Cryst.* A**45**, 50-61.

Liu, D. C. & Nocedal, J. (1989). *Math. Program*. **45**, 503-528.

Lunin, V. Y. & Skovoroda, T. P. (1995). *Acta Cryst.* A**51**, 880-887.

Terwilliger, T. C., Grosse-Kunstleve, R. W., Afonine, P. V., Adams, P. D., Moriarty, N. W., Zwart, P. H., Read, R. J., Turk, D. & Hung, L.-W. (2007). *Acta Cryst*. D**63**, 597-610.

Mite-y Lysozyme Crystals and Structures

Janet Newman*, Del Lucent and Thomas S Peat

Molecular and Health Technologies, CSIRO, 343 Royal Parade, Parkville, VIC, 3052, Australia

*Correspondence email: janet.newman@csiro.au

Synopsis

Three different sandwich spreads were tested for their ability to crystallise lysozyme, the resultant crystals were of uniformly high quality and produced structures that fell within the envelope of the known structures of lysozyme.

Abstract

Marmite, Promite and Vegemite are three variations of yeast extract pastes, which are considered edible foodstuffs by many people around the world. These spreads all report high levels of sodium on their nutritional information labels and we were interested if this would correspond to an ability to support lysozyme crystallisation, which may be easily crystallised from sodium chloride solutions. Counter diffusion crystallisation experiments were set up with hen egg white lysozyme, using Marmite, Promite or Vegemite as the crystallant. The technique of counter diffusion was chosen as this allowed crystal growth to be observed, despite the black, opaque nature of the crystallants. Crystals grew from all three spreads and these crystals were tested for diffraction quality and structures were produced from crystals of each variety of yeast extract paste. The tested crystals grew in the familiar $P4_32_12$ tetragonal space group, with cell dimensions of approximately 79 x 79 x 38 Å^3.

Keywords: counter diffusion, vegemite, lysozyme, multiple structure alignment

1. Introduction

Yeasts have been used in the production of human foods for millennia (Cavalieri *et al.*, 2003, Legras *et al.*, 2007). One of the more recent incarnations of yeast food products are the yeast extract pastes that are popular spreads for toast and sandwiches in some countries. The process for making concentrated yeast extract requires the addition of sodium chloride to a yeast cell pellet in order to induce autolysis after which the resulting lysate is filtered, flavoured and concentrated (Irving, 1992, Cook, 1910), this is a modification of a process developed by Liebig, modelled in turn after his process for the extraction of the essence of meat (Brock, 1997). One of the earliest of these products to be available commercially was Marmite, which was sold by the Marmite Food Company (later Marmite Ltd) of Burton on Trent, UK in 1902 (*The Bumper Book of Marmite*, 2009). Similar products are available in Australia (Vegemite, Promite), Switzerland (Cenovis) and New Zealand (Marmite) - note that New Zealand Marmite is produced under license and has a different formulation than the British product of the same name (wikipedia.org/wiki/Marmite). Vegemite has been touted as being potentially one of the most culturally specific foods – if you eat Vegemite, then you are very likely to be Australian and *vice versa* (Rozin & Siegal, 2003). These products are renown for their salty tang and the nutritional information labels show that these products contain anywhere from 3400 mg to 4844 mg sodium per 100 g of product. Assuming that the counter-ion of the sodium is chloride and given the mass percentage of sodium in NaCl is 39.34%, this would suggest that the products contain from 8.7 g to 12.3 g of NaCl per 100 g.

Hen egg white lysozyme (HEWL) is a readily available protein that is notoriously overused as a crystallisation test protein (see for example (Lu *et al.*, 2010, Newman, 2005, Newman *et al.*, 2007, Vrikkis *et al.*, 2009)). This protein crystallises out of numerous conditions (Newman *et al.*, 2007) but is often crystallised from a sodium acetate / sodium chloride crystallant, where the acetate is buffered to around pH 4.6 and the NaCl concentration is around 5% (or equivalently, around 1 M) (Bergfors, 2009).

Counter diffusion is a crystallisation technique in which a concentrated protein solution is introduced into a capillary and the crystallant solution is allowed to migrate into the capillary. If the capillary is of sufficiently narrow bore, the

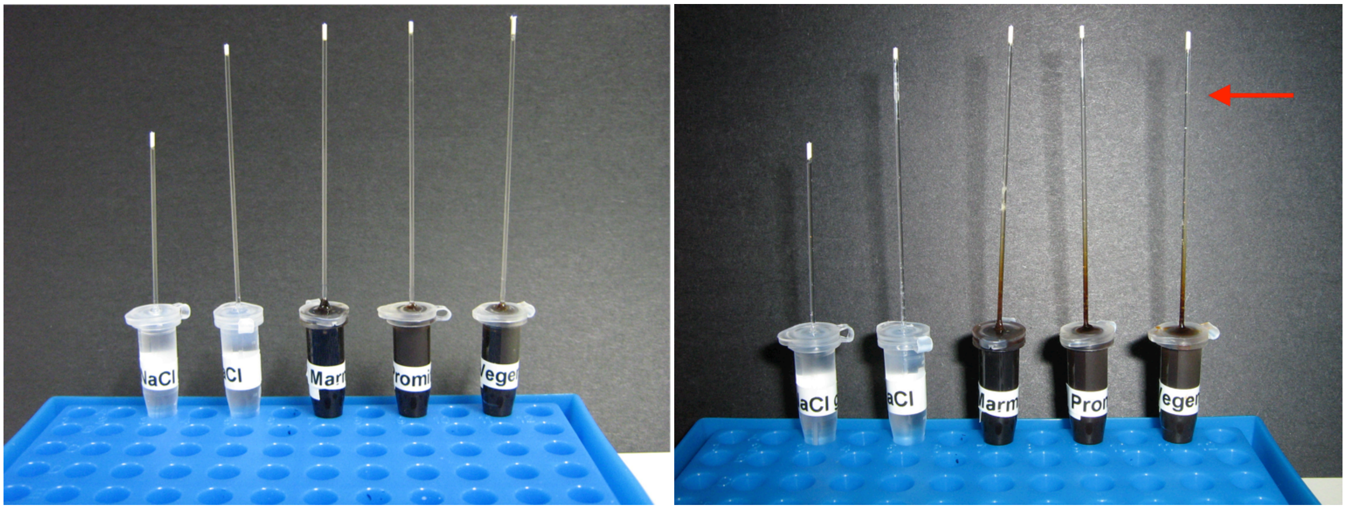


Figure 1. (a) The capillary counter diffusion experiment on setup at left, (b) shows the same experiment 16 days laterat right. The red arrow shows the position of a crystal in the Vegemite experiment. These experiments were set up at room temperature and incubated at 4C.

crystallant moves into and along the tube only by diffusion and sets up a concentration gradient along the length of the tube (Garcia, 2003, Ng *et al.*, 2003, Ng *et al.*, 2008). Because of this transient gradient, large, well formed crystals can grow even if the crystallant is of much higher concentration than would normally be used in a more standard vapour diffusion experiment

We set up counter diffusion experiments with commercial HEWL and yeast pastes for a number of reasons - primarily to build expertise with the counter diffusion technique, but also as we were rather curious whether crystallogenesis could be achieved with these salty foodstuffs. We were also interested in determining whether the structures of HEWL determined from crystals grown in the spreads would be significantly different from the large number of structures already available for this protein.

2. Materials and methods

Counter diffusion experiments were set up in two slightly different formats: in both cases all three pastes were set up along with two sodium chloride control experiments. In one variation, three vials were prepared by adding approximately one millilitre of Vegemite (Kraft Australia), Marmite (Sanitarium NZ) or Promite (Mars Food Australia) to the bottom of a push-cap vial. The pastes are hard to work with neatly - eventually a technique was developed where a spatula was used to scoop paste into a 5 ml syringe, this syringe was used to fill a 1 ml syringe, which was used to deposit the spread cleanly at the bottom of the vial. One millilitre of melted 1% agarose gel (Applichem) was layered over the pastes and allowed to solidify. Two controls were set up: the first control vial was set up by adding 1 ml of 5 M NaCl (Sigma S7653) and layering over 1 ml of the 1% agarose solution, the second by allowing 0.5 ml agarose to harden in the bottom of a vial, then adding 1 ml of the 5 M NaCl solution on top. Five 64 mm long, 0.63 mm internal diameter glass capillaries (Drummond MicroCap 1-000-0200) were filled with a solution of 40 mg/ml lysozyme (Sigma L6876) in 50 mM sodium acetate pH 4.5 and one end sealed with Haematocrit sealing compound (Brand). The open end of the capillary was inserted through the agarose gel into the paste (or salt solution). The setups were stored at room temperature. The capillaries were examined for crystal formation under a light microscope without removing them from the vials (Figure 1a)

In a second variation on the counter diffusion technique, capillaries were filled with the protein solution and one end sealed as above. The crystallant was contained in 0.65 ml Eppendorf tubes, which had lids pierced with an 18-gauge needle. The vials were filled with crystallant: for the spreads, the tubes were filled with the paste. The NaCl controls were set up with either an agarose layer in the bottom of the Eppendorf tube with 5 M NaCl solution filling the remaining volume, or with 0.5 ml of the 5 M NaCl solution

Table 1. Sample information

Macromolecule details			
Database code(*s*)	PDB code: 3N9A (Vegemite), 3N9C (Marmite) and 3N9E (Promite)		
Component molecules	Hen Egg White Lysozyme (EC number: 3.2.1.17)		
Mass (Da)	14,700		
Source organism	Gallus gallus (details: Purchased from Sigma L6876)		
Crystallization and crystal data			
	Crystal 1 – Vegemite	Crystal 2 – Marmite	Crystal 3 – Promite
Crystallization method	Free interface diffusion/counterdiffusion	Free interface diffusion/counterdiffusion	Free interface diffusion/counterdiffusion
Temperature (K)	293	293	293
Apparatus	Drummond microcaps	Drummond microcaps	Drummond microcaps
Atmosphere	1	1	1
Seeding	None	None	None
Crystallization solutions			
Macromolecule	20 ml, lysozyme (40 mg ml^{-1}), sodium acetate (pH 4.5, 50 m*M*)	20 ml, Lysozyme (40 mg ml^{-1}), sodium acetate (pH 4.5, 50 m*M*)	20 ml, lysozyme (40 mg ml^{-1}), sodium acetate (pH 4.5, 50 m*M*)
Unit-cell data			
Crystal system, space group	Tetragonal, $P4_32_12$	Tetragonal, $P4_32_12$	Tetragonal, $P4_32_12$
***a*, *b*, *c* (Å)**	79.29, 79.29, 37.89	79.41, 79.41, 38.02	79.29, 79.29, 38.00
a, b, g (°)	90, 90, 90	90, 90, 90	90, 90, 90

placed into the empty tube and the agarose gel layered over. The unsealed end of a prepared capillary was inserted through the lid into the filled (and closed) Eppendorf tube and the gap between the capillary and lid was sealed with a dab of clear nail polish (Figure 1b). These setups were stored at 4 C. In all 10 experiments, care was taken to ensure that there was protein solution all the way to the ends of the capillaries and that the unsealed end of the capillaries did not touch the vials/tubes, to ensure that the crystallant could diffuse freely into the protein solution.

Data were collected at the MX1 beamline of the Australian Synchrotron from crystals grown from the three yeast extracts from the room temperature experiments. Data were collected on the crystals *in-situ*, at room temperature (see table 1 for details). The crystals were prepared for data collection by wicking away most of the mother liquor and re-sealing the capillaries with wax. The capillaries were mounted on a magnetic cap with modelling clay and we translated the crystals several times during data collection to introduce a fresh part of the crystal to the beam. 180 frames of data were collected, with each frame being a 1 degree oscillation exposed for 1 second. The crystal to detector distance was 140 mm, leading

Table 2. Data collection and structure solution statistics. Values for the outer shell are given in parentheses.

	Diffraction set 1 (crystal 1 -Vegemite)	Diffraction set 2 (crystal 2 - Marmite)	Diffraction set 3 (crystal 3 - Promite)
Diffraction source	AS, MX1	AS, MX1	AS, MX1
X-ray beam size	0.1 mm x 0.1 mm	0.1 mm x 0.1 mm	0.1 mm x 0.1 mm
Sampling protocol	1º oscillation, 1 sec/º	1º oscillation, 1 sec/º	1º oscillation, 1 sec/º
Wavelength (Å)	0.98	0.98	0.98
Detector	ADSC Quantum 215	ADSC Quantum 215	ADSC Quantum 215
Temperature (K)	293	293	293
Resolution range (Å)	39.7–1.40 (1.48–1.40)	40.0 – 1.50 (1.58-1.50)	39.7 – 1.38 (1.45-1.38)
No. of unique reflections	23901 (2999)	19946 (2867)	25326 (3480)
No. of observed reflections	443919	208981	495066
Completeness (%)	98.0 (87.0)	100 (100)	99.4 (96.0)
Redundancy	18.6 (13.3)	10.5 (10.6)	19.5 (11.6)
< *I*/s(*I*)>	35.7 (6.1)	19.9 (4.6)	35.1 (4.8)
R_{merge}	5.5 (45.4)	6.7 (49.5)	5.4 (47.4)
$R_{p.i.m.}$	1.3	2.2	1.2
Data-processing software	*SCALA*	SCALA	SCALA
Phasing method	MR	MR	MR
Starting model data set	2BLX	2BLX	2BLX
Alterations to search model	none	none	none
Solution software	PHASER	PHASER	PHASER

to a maximum resolution of around 1.4 Å (see table 2 for details).

Data were indexed with MOSFLM (Leslie, 1992), merged and scaled with SCALA, molecular replacement (using the structure 2BLX from the Protein Data Bank (PDB, http://www.pdb.org) (Deshpande *et al.*, 2005)) was performed using PHASER and the models refined with REFMAC 5.6 (Collaborative Computational Project, 1994) (see table 3 for refinement details).

The refined structures of HEWL generated in this work were compared to other HEWL structures available in the PDB. A sequence search was run using the "Sequence (Blast/Fasta)" option in the PDB website where the sequence of HEWL (pdb 2BLX) was used, along with a BLAST (Altschul *et al.*, 1990) cutoff E-value of 10^{-50}. This returned 332 structures, which had identical or 1 amino acid difference to the search sequence. A python script was created which used PyMol version 1.2r3 (Delano, 2003) to align the 332 sequences to the sequence from the PDB entry 2BLX and the aligned sequences were used to generate an ensemble of aligned structures. This was represented by drawing a "sausage" with a radius corresponding to the difference seen at each main chain atom position around the structure 2BLX. Two envelopes of the ensemble were calculated: one using the root mean square deviation (rmsd) of the main chain atoms from the reference structure and a second where the maximal distance from the main chain atom was used to set the radius of the "sausage" at that atom. We superposed the structures from the

Table 3. Structure refinement and model validation. Values for the outer shell are given in parentheses.

	Diffraction set 1 (crystal 1 -Vegemite)	Diffraction set 2 (crystal 2 - Marmite)	Diffraction set 3 (crystal 3 - Promite)
Refinement software	*REFMAC* 5.6	*REFMAC* 5.6	*REFMAC* 5.6 Both the
Refinement on	*F*	*F*	*F* salt
Resolution range (Å)	56.1–1.4	56.2 – 1.50	56.1 - 1.38
No. of reflections used in refinement	22633	18886	23991
Final overall *R* factor	16.3 (23.1)	16.0 (26.6)	16.2 (26.1)
Atomic displacement model	isotropic	isotropic	isotropic
Overall average *B* factor (Å^2)	19.8	19.9	20.7
No. of protein atoms	1001	1001	1001
No. of nucleic acid atoms	0	0	0
No. of ligand atoms	0	0	0
No. of solvent atoms	78	77	82
Total No. of atoms	1204	1215	1231
No. of refined parameters			
Non-crystallographic symmetry restraints	None	None	None
Final R_{work}	16.3 (23.1)	16.0 (26.6)	16.2 (26.1)
No. of reflections for R_{free}	1224 (78)	1016 (62)	1288 (71)
Final R_{free}	18.9 (27.5)	19.0 (25.4)	18.3 (30.8)
Ramachandran plot analysis			
Most favoured regions (%)	97.8	95.7	96.6
Additionally allowed regions (%)	2.2	4.3	3.4

Vegemite, Marmite and Promite crystallisations on this ensemble, to obtain a visual gauge of how similar these structures were to the myriad of other structures available for HEWL.

3. Results and discussion

A number of crystals appeared in the capillaries in all experiments within 48 hours of setup. Fewer and smaller crystals were observed in all of the room temperature experiments than in the corresponding 4 C experiments. Sixteen days after setup there were crystals along the entire length of all the capillaries at 4 C, with considerable diffusion of the paste (as seen by a brown colouration) along the sample capillaries (Figures controls and the Promite paste showed poor crystal morphology close to the open end of the capillary and better crystal morphology towards the closed end of the capillary. The cold Vegemite and Marmite experiments had fewer, larger and better formed crystals than the equivalent Promite experiment. The experiments set up at 20 C gave fewer crystals in the paste experiments than the corresponding experiments at 4 C, with no crystals observed at the “distant” – or sealed end of the capillaries. The warm salt controls showed crystal growth along the entire length of the capillaries - mostly the high-salt "sea urchin" crystal habit. The agarose layer over the pastes in the vials turned quite black and there was

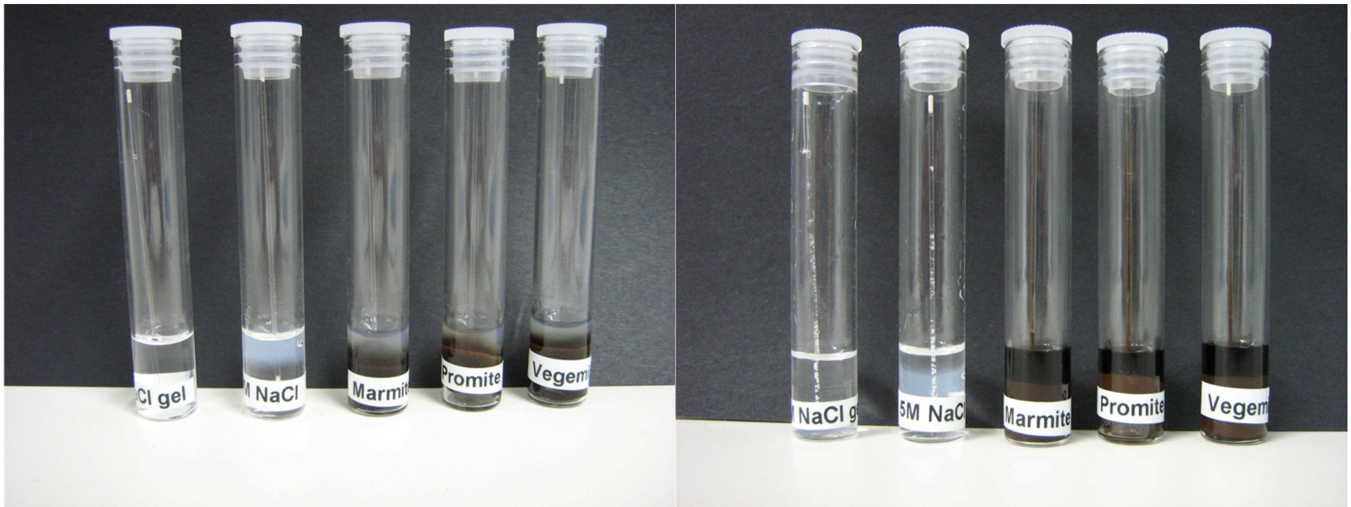


Figure 2. (a) Capillary counter diffusion experiments set up in vials immediately after setup. (b) shows the same experiments 16 days later. These experiments were incubated at 20C.

colouration from the pastes over half the length of the capillaries in the paste setups at room temperature (Figure 2b).

As stated above, both the 5 M salt controls and the Promite setups showed poor crystal morphology in the "near" end of the capillaries, compared to the experiments set up with Vegemite and Marmite. Promite has significantly more sodium (4844 mg 100 g^{-1}) than either Vegemite or Marmite (3489 mg 100 g^{-1} or 3400 mg 100 g^{-1} respectively), potentially leading to this effect. Counter diffusion is an intriguing technique, as a vast range of concentrations can be tested in the one experiment. However, our 20 C 5 M NaCl controls showed poor crystal habit, suggesting that this aspect of the technique can be overwhelmed if the crystallant concentration is too high, or if the capillary is too wide in internal diameter, or not long enough. We wanted to use glass X-ray capillaries for this experiment, but found that they were too delicate for the process – they crushed when the wide end was removed and broke when pushed into the Haematocrit sealant. The experiments where the capillary was enclosed in a glass vial were harder to visualise than the experiments were only the end of the vial was pushed into an Eppendorf tube. A refinement that we now use is to use 0.2 ml PCR tubes to contain the crystallant, as we find that these smaller tubes, when filled completely with liquid, give a very good view of the entire capillary. We saw a large amount of background scatter during data collection, presumably from the Drummond MicroCap capillary and would recommend moving to a thinner walled tube for *in-situ* diffraction studies, although the robustness of the thick MicroCaps has a lot to recommend it. Two crystals produced from the Marmite spread were somewhat less durable in the beam than the crystals produced by the other spreads, but this could be general variation of crystal quality rather than being specific to the Marmite experiment.

The three resulting structures are of very high quality overall, with strong, unambiguous density for residues 1-129. A loop region (residues 70-72), as well as the C-terminus (residues 126-129) were less well defined than the rest of the structure, suggesting that these regions are somewhat flexible. The three structures, one from each spread, where refined in a very similar manner and, unsurprisingly, the results were very similar. There are some differences between the three for side-chains where the density suggested multiple conformers were present (such as Arg73 or Asn19).

There is very little unexplained density in the maps – small blobs, but nothing large enough to be modelled by a sugar or some other cellular component. We do see some radiation damage, (as indicated by negative density around the sulfur atoms) near the disulfide bridge Cys6-Cys127 and a smaller amount associated with the disulfide bridge Cys76-Cys94.

The variation seen in the large number of lysozyme structures (332) obtained from the PDB

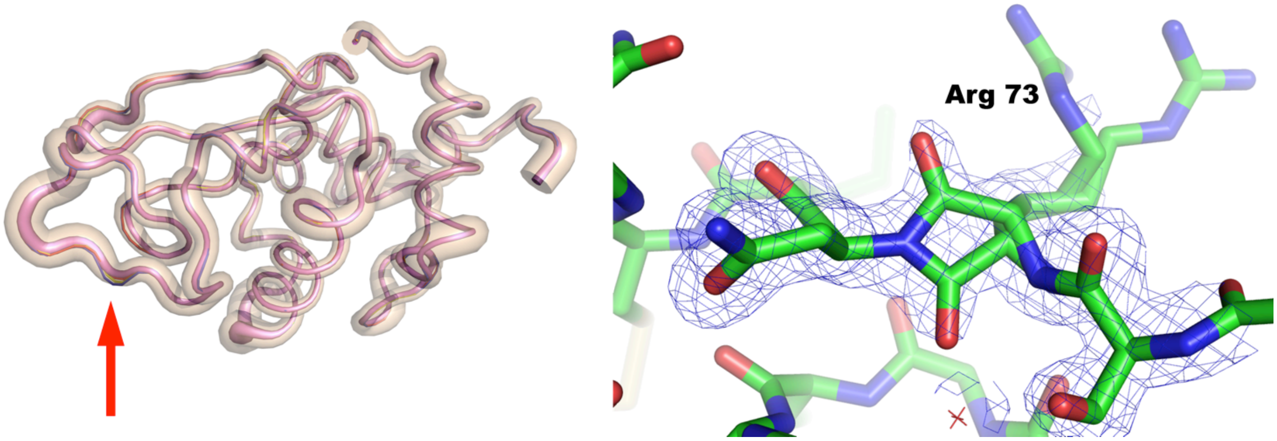


Figure 3. Backbone variation among known lysozyme structures. (a) Sausage diagram showing the maximum backbone distance (transparent tan envelope) and backbone RMSD (opaque violet envelope) for all structures in the PDB with sequence similarity E-values less than or equal to10^{-50} of 2BLX. Shown in ribbons are the backbones of the lysozyme structures crystallized in Vegemite (red), Marmite (blue), and Promite (yellow). The red arrow indicates the position of Arg73, where the three current structures deviate most from the consensus backbone trace. (b) Electron density of the main chain around Arg73 in the Vegemite structure. The density is well modelled by having two conformers for this residue, which include two very different positions for the main chain oxygen.

is remarkably small – the overall rmsd in the backbone positions was 0.55 Å. The envelope of deviation from the chosen structure (Figure 3a) shows some variation along the chain, with an unsurprisingly greater spread being seen in the loop regions, which mirrors what we observed in the three structures presented here. We find notable the lack of difference between over three hundred structures, solved and refined using different technologies by a host of different people. The three structures solved in this present work fall mostly within the smaller, rmsd envelope of all the structures, with some points of difference. The differences tend to occur where we have modelled in alternative conformers of residues; we used COOT (Emsley & Cowtan, 2004) for model building and used the option whereby a complete residue was included as an alternative conformer: older technologies tended to only include side-chain atoms in alternate conformer definitions. Figure 3b shows the region of the electron density for Arg73 in the Vegemite structure that we modelled as two conformers, with quite different positions of the main chain carbonyl oxygen. The other two structures showed similar electron density in this region and were modelled in a similar way. We believe that the deviations of the structures from the envelope of PDB structures can all be explained in this way. The structures have been deposited in the PDB with accession codes 3N9A (Vegemite), 3N9C (Marmite) and 3N9E (Promite).

4. Conclusions

Commercial hen egg white lysozyme may be readily crystallised by the counter diffusion method from common sandwich spreads – the non sodium chloride components of the spreads appear not to have major negative effect on the resulting crystals, as there was no obvious excess density seen in the refined structures and the structures themselves aligned well with a large number of previously solved HEWL structures. We have developed a script that uses PyMol to align and visualise a large number of protein structures, this is available by emailing del.lucent@csiro.au.

Acknowledgements

We thank the beamline scientists of the Australian Synchrotron for their help in data collection. Dr. Del Lucent gets special thanks for agreeing to take and maybe even eat the excess spreads.

References

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (1990). *J Mol Biol* **215**, 403-410.

Bergfors, T. M. (2009). Editor. *Protein*

Crystallization Second Edition La Jolla: International University Line.

Brock, W. H. (1997). *Justus von Liebig: The Chemical Gatekeeper*. Cambridge: Cambridge University Press.

The Bumper Book of Marmite, 2009). Bath: Absolute Press.

Cavalieri, D., McGovern, P., Hartl, D., Mortimer, R. & Polsinelli, M. (2003). *Journal of Molecular Evolution* **57**, S226-S232.

Collaborative Computational Project, N. (1994). *Acta Crystallographica Section D* **50**, 760-763.

Cook, F. C. (1910). *A comparison of Beef and Yeast Extracts of Known Origin*. US Department of Agriculture.

Delano, W. (2003). *The PyMol Molecular Graphics System.* Version 0.99.

Deshpande, N., Addess, K. J., Bluhm, W. F., Merino-Ott, J. C., Townsend-Merino, W., Zhang, Q., Knezevich, C., Xie, L., Chen, L., Feng, Z., Green, R. K., Flippen-Anderson, J. L., Westbrook, J., Berman, H. M. & Bourne, P. E. (2005). *Nucleic Acids Res* **33**, D233-237.

Emsley, P. & Cowtan, K. (2004). *Acta Crystallographica Section D* **60**, 2126-2132.

Garcia, J. M. (2003). *Methods in Enzymology*, Vol. 368. *Macromolecular Crystallography, Part C,* edited by C. Carter, pp. 130-154: Elsevier.

Irving, J. (1992). *Vegemite Cook Book*. Melbourne: Ark Publishing Pty Ltd.

Legras, J.-L., Merdinoglu, D., Cornuet, J.-M. & Karst, F. (2007). *Molecular Ecology* **16**, 2091-2102.

Leslie, A. G. W. (1992). *Joint CCP4 + ESF-EAMCB Newsletter on Protein Crystallography* **26**.

Lu, Q.-Q., Yin, D.-C., Liu, Y.-M., Wang, X.-K., Yang, P.-F., Liu, Z.-T. & Shang, P. (2010). *Journal of Applied Crystallography* **43**, 473-482.

Newman, J. (2005). *Acta Crystallographica* **D61**, 490-493.

Newman, J., Xu, J. & Willis, M. C. (2007). *Acta Crystallographica* **D63**, 826-832.

Ng, J. D., Gavira, J. A. & Garcia-Ruiz, J. M. (2003). *Journal of Structural Biology* **142**, 218-231.

Ng, J. D., Stevens, R. C. & Kuhn, P. (2008). Vol. 426. *Structural Proteomics: High-Throughput Methods,* edited by B. Kobe, M. Guss & T. Huber, pp. 363-376. New York: Springer Science+Business Media LLC.

Rozin, P. & Siegal, M. (2003). *Gastronomica* **3**, 63-67.

Vrikkis, R. M., Fraser, K. J., Fujita, K., MacFarlane, D. R. & Elliott, G. D. (2009). *Journal of Biomechanical Engineering* **131**, 074514.