



CCP4 NEWSLETTER ON PROTEIN CRYSTALLOGRAPHY

An informal Newsletter associated with the BBSRC Collaborative Computational Project No. 4 on Protein Crystallography.

Number 43

Winter 2005

Contents

CCP4-Worldwide

1. **Report on the Joint CCP4-RCSB PDB Workshop at the IUCR XX**

Peter Briggs

CCP4, Daresbury Laboratory, Warrington WA4 4AD, UK

Softwares

2. **Movies in minutes with CCP4mg**

Liz Potterton⁽¹⁾, Stuart McNicholas⁽¹⁾, Martin Noble⁽²⁾

¹Department of Chemistry, University of York, York Y10 5YW, ²Laboratory of Molecular Biophysics, University of Oxford, Oxford OZ1 3QU

3. **Java Based Developments for Protein Crystallography**

John W. Campbell

campbell_w_j@bcs.org.uk

4. **PDB Depositions with RCSB PDB Tools**

Christine Zardecki

Nucleic Acid Database Project, Department of Chemistry, Rutgers, The State University of New Jersey

Automation

5. **Application of Automation to Data Processing & Analysis**

Graeme Winter

Daresbury Laboratory, Keckick lane, Warrington WA4 4AD

6. **HAPPY Progress**

Daniel Rolfe⁽¹⁾, Charles Ballard⁽¹⁾, Maria Turkenburg⁽²⁾, Eleanor Dodson⁽²⁾, Paul Emsley⁽²⁾

¹Daresbury Laboratory, Kekwick Lane, Warrington WA4 4AD, ²University of York, York Y010 5YW

Methodology

7. **Xtriage and Fest: automatic assessment of X-ray data and substructure structure factor estimation**

P. H. Zwart, R. W. Grosse-Kunstleve, P. D. Adams

Lawrence Berkeley National Laboratory, 1 Cyclotron Road, BLDG 64R0121, Berkeley, California

Editor: Francois Remacle

Daresbury Laboratory, Daresbury,
Warrington, WA4 4AD, UK

NOTE: The CCP4 Newsletter is not a formal publication and permission to refer to or quote from the articles reproduced here must be referred to the authors.

Contributions are invited for the next issue of the newsletter, and should be sent to Francois Remacle by e-mail at <mailto:fr45@ccp4.ac.uk> by next season. HTML is preferred but other editable formats are also acceptable.

Report on the Joint CCP4-RCSB PDB Workshop at the IUCr XX

Peter Briggs

CCP4, Daresbury Laboratory, Warrington WA4 4AD, UK

In a previous CCP4 newsletter (see [Newsletter 42](#)) we reported on the one-day CCP4 workshop held at the ACA meeting in Orlando Florida at the start of the summer. We were pleased to be able to bookend the summer conference season with a second one-day satellite workshop - this time a joint effort between CCP4 and the RCSB PDB, at the XX IUCr Congress in Florence, Italy. This workshop - entitled "*A Protein Crystallographic Toolbox: CCP4 Software Suite and PDB Deposition Tools*" - continues a tradition of collaboration between the two projects. The workshop attracted around sixty registered participants ranging from graduate students through to experienced crystallographers, bringing with them a broad range of familiarity with the CCP4 and RCSB software. The workshop followed our familiar format, with the speakers (generally a mixture of software developers and expert users) giving detailed user-focused presentations on the software in their area of expertise. The presentations were intended to help novice users get started while at the same time surprising more expert users with new or less-well known aspects of the software.

After a brief joint introduction from **Peter Briggs** of CCP4 and **Helen Berman** of the RCSB-PDB, the first session concentrated on giving a broad overview intended to lay the foundations for the later talks. This included an introduction to the non-crystallographic "technical" aspects of using the CCP4 suite and an excellent overview from **Martin Noble** of the range of crystallographic functions in the suite. His tour of the suite emphasised CCP4's "cradle to grave" coverage of the structure determination process and its close integration with other non-CCP4 software packages.

The remaining presentations focused on practical aspects of the software for key stages of structure determination and deposition. **Gwyndaf Evans** covered data processing, integration and scaling using MOSFLM and SCALA, including tell-tale signs which indicate whether data processing is working (*hint*: look for sudden deviations from otherwise smoothly-varying behaviours) and how to address various problem cases. **Martyn Winn** talked about refinement with REFMAC5, focusing particularly on the use of TLS parameters. He made a good case for trying TLS, as at worst it will make no difference to your refinement while at best it can give a big improvement. Analysis of TLS values may also be a useful tool in suggesting biologically significant protein domain motions.

This was followed by impressive demonstrations by **Liz Potterton** and **Paul Emsley** of two faces of CCP4's molecular graphics project, CCP4MG and Coot respectively: the former currently focuses on providing presentation-quality representations of molecular models, while the latter is a platform for powerful model building tools. Both packages are now becoming widely used and are already key tools in many macromolecular crystallographers' toolkits (they will also both be available as part of the forthcoming CCP4 v6.0 release). **Kyle Burkhardt** of the RCSB-PDB closed appropriately by focusing on the deposition of refined protein structures. She emphasised a number of issues, including the

existence of software tools to help make deposition easier and the importance of validation prior to and during the deposition process, and reminded the audience that the deposition sites want to work with the authors to ensure the best possible representation of their data in the archive.

The workshop closed with a summary of the most interesting "bullet points" from each of the talks. Although the audience response was muted (possibly due to the vastness of the Cimabue Room!) the feedback provided via the questionnaires was very positive, and we were also pleased to see a large percentage of younger crystallographers in attendance. Many people reported that they found the presentations helpful and that they would be trying out the software afterwards. As always there were also requests for a longer workshop with more hands-on demonstrations and more examples of real-life problems - we are of course interested in addressing these issues in future (and in this regard CCP4 would also welcome any offers of help in organising such workshops).

Next year we hope to be back at the [ACA in Hawai'i](#) and at the [ECM in Leuven, Belgium](#). The materials from this IUCr workshop can be found online via <http://www.ccp4.ac.uk/courses/IUCr2005/>. More information about CCP4 can be found at <http://www.ccp4.ac.uk> and about the RCSB PDB at <http://www.rcsb.org/pdb/>.

Acknowledgements

The workshop organisers Maeri Howard, Peter Briggs and Judy Flippen-Anderson would like to thank the IUCr for the opportunity to run the workshop, the speakers for their presentations and the delegates for attending. We also wish to acknowledge the help of Annalisa Guerri in setting up the workshop. Financial support was provided by CCLRC (from CCP4 industrial income), the RCSB-PDB and from workshop registrations by the IUCr.



*From back to front, left to right:
Gwyndaf Evans, Martin Noble,
Peter Briggs, Maeri Howard,
Paul Emsley, Liz Potterton,
Kyle Burkhardt, Martyn Winn*

Making Movies in Minutes with CCP4mg

*Liz Potterton, Stuart McNicholas & Martin Noble**

Department of Chemistry, University of York, York YO10 5YW

**Laboratory of Molecular Biophysics, University of Oxford OX1 3QU*

Introduction

Simple movies are often good in a presentation or on a web page as they can convey the 3-dimensionality or some dynamic aspect of a structure better than a static picture. But creating movies can be time consuming. We have attempted with the CCP4 molecular graphics program, CCP4mg(ref 1), to make movie making as simple as creating static pictures. So creating a simple movie, like those shown below, should only take a few minutes.

Like most molecular graphics packages CCP4mg will output a series of screen snapshots; each snapshot corresponds to one frame of the movie. These frames are then compiled into a movie by another, external program. CCP4mg has a very simple interface for setting up the movie and a 'Compile' button that will create the movie file automatically.

The main question with movie making is: 'what is actually going to move?'. Some possible answers are shown below.

Moving the camera

For this, simplest sort of movie, the displayed structures do not change in the course of the movie but the camera is considered to move. Actually it looks as if the displayed structures are rotating, rocking from side to side or moving in some other simple fashion. It is usual to ensure that at the end of the movie the motion has returned to the starting point so that, when displayed, the movie can loop through multiple times without an obvious break. CCP4mg has simple options to set up 360° rotation, rocking, or motion between two views selected by the user. An example of the later is shown [here](#). For this movie, the user needs only to indicate the required start and end view and program will interpolate between the two views.

Animating the structure

An animation might be a dynamics trajectory or a morph such as is created by the [Yale Morph Server](#). CCP4mg can load the coordinates for each step of the animation from either separate PDB files or from the multiple 'NMR' models in one PDB file. There are a huge number of dynamics trajectory formats; if there are any you would particularly like us to support then please let us know.

It's probably best to retain the same view throughout while recording a movie, but you can also move the camera if you wish. It is possible to use any of the usual model display styles or colour schemes for the animation and, if necessary, properties such as hydrogen bonds, secondary structure assignment, surfaces or electrostatic potential will be updated. But the latter two may take a few more minutes!

Here is a [simple animation movie](#).

Varying the electron density contour level

You can change some properties of the displayed data through the course of a movie. The most useful things to change are probably the electron density contour level or the position of a model but other things that might change are: position of text or images, size of images and transparency levels. Here is an example of [changing electron density levels](#). In this movie the map has been clipped to cover only a small number of residues and is drawn as thin cylinders.

Moving a model

Moving a model or fragment of a model can be useful to show the relationship between different models or perhaps the sort of movement that must occur in domain motion. In the [moving model movie](#) there are two models corresponding to the two chains of the immunoglobulin 4fab. One section of the movie shows the transformation of the heavy chain on to the light chain. This section was created by saving two states of the display: one with the two chains in their usual positions and one after the chains have been superposed by the Secondary Structure Matching tool(ref 2). The program will automatically interpolate between the two positions of the heavy chain through the movie.

This movie was made in four simple stages which either moved the camera or the model or, for a short section, nothing moved. The accumulated frames were then compiled into one movie.

Movie formats and compiling the movie

The movies in this article are in animated GIF format as this is well supported by web browsers. But it is not a well compressed file format. There are a bewildering array of movie file formats and programs to handle them. These have a strong operating system dependency and some tedious associated licencing issues.

CCP4mg outputs each frame in **PNG** format which can be imported to any movie making program. Alternatively, if the appropriate conversion program is installed, CCP4mg will run it automatically as a background job. Currently the [ImageMagick](#) program *convert* and (on Linux only) [mencoder](#) are interfaced.

Future developments

These movies were made with CCP4mg v0.13.

We are working on providing a simple interface for creating more complex movies. It should be possible to create a movie with several distinct scenes and to preview and edit the movie via a visual interface. It should also be possible to create the impression of 'flying through' the structure.

References

- 1). *Developments in the CCP4 molecular-graphics project*. Liz Potterton, Stuart McNicholas, Eugene Krissinel, Jan Gruber, Kevin Cowtan, Paul Emsley, Garib N. Murshudov, Serge Cohen, Anastassis Perrakis and Martin Noble. Acta. Cryst. D12 2288.
2. *Secondary-structure matching (SSM), a new tool for fast protein structure alignment in 3D*. E.B. Krissinel and K. Hendrick. Acta. Cryst. D12 2256

Java Based Developments for Protein Crystallography



John W. Campbell, November 2005

Introduction

This article briefly describes a library of PX related Java classes and two Java programs based on these which have similar functionality to the CCP4 programs ROTGEN and IPDISP. An associated documentation preparation program is also available.

The Java Development Library



The Java Development Library (JDL) is a set of Java classes written for developing Java based software for Protein Crystallography. It contains four main sections of classes as follows:

1. *JdlLib*

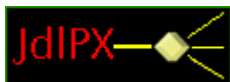
This is a set of general purpose classes which are used in the implementation of the other sections of the library. These include classes and methods for handling strings, files, data conversions and manipulating bits. In addition, there are classes and methods for handling 3 element vectors, 3x3 matrices and 4x4 matrices and a class which carries out general non-linear least squares refinements using the Levenberg-Marquardt algorithm.

2. *JdlView*



JdlView is a package for use in developing windows based programs based on an older X-windows based toolkit [XDL VIEW](#). The JdlView classes basically fall into two categories. These are the general purpose JdlView classes which are written without any particular application in mind (e.g the menu area class or the I/O window class) and those which are written with a particular application or group of applications in mind, particularly for Protein Crystallography. They are primarily intended to be used in programs which use the JdlView management class which forms part of the package but they can also be used independently of that if required to fit within some alternative programming context.

3. *JdIPX*



This is a set of classes specifically written for use in protein crystallography. It includes classes for holding various sets of crystallographic data, symmetry handling, reading and manipulating diffraction images. There are classes and methods for predicting Rotation, Laue and Unique reflection data and classes for storing and manipulating the predicted reflection lists. There is also a class which implements a [Diffraction Data Module \(DDM\)](#)



4. *JdIKDM*

This is a set of classes for defining and handling Keyworded parameter files and their associated data.

[Detailed documentation](#) of the JDL classes may be found on the CCP4 website.

The PXSim Program



PXSim is a program for carrying out simulations of Protein Crystallographic diffraction images and analysing the unique data coverage for one or more crystal settings and series of diffraction images. It is intended primarily as a teaching tool to enable the user to investigate, in some depth, the nature of X-ray diffraction patterns and the amount of data which may be collected using various settings or methods. The program may be used for simulating diffraction patterns for the Rotation, Weissenberg and Laue methods of data collection and analysing the extent of data which may be measured using those methods. The predictions are based on the set of parameters defined in a [Diffraction Data Module \(DDM\)](#).

The use of Java was chosen so that the program could be made available for use across the internet as well as being in a form suitable for porting to any Java supporting platform.

The basic screen layout is shown below:

Figure 1: Main Screen of PXSim

The screenshot displays the PXSim main screen, which is divided into several sections. On the left, there is a 'Display Menu' with buttons for 'Display', 'Remove', and 'Return to Main Menu'. Below this is a 'Current Option' section with dropdown menus for 'Orient' (set to 'default') and 'Multifac' (set to '2.00'). The main area is titled 'Main Parameter Items' and contains a grid of input fields for various simulation parameters. These include 'Set no.', 'Img no.', 'title' (set to 'Yeast P_b'), 'type' (set to 'Rotation'), 'numsets', 'system' (set to 'Monocli_b'), 'lattice' (set to 'C'), 'symm' (set to '5'), 'a', 'b', 'c', 'alpha', 'beta', 'gamma', 'crystal' (set to '1'), 'wavelen' (set to '1.0'), 'resolution' (set to '1.75'), 'mosaicity' (set to '0.3'), 'distance' (set to '138.70'), 'rmax' (set to '90.0'), 'phi_x' (set to '39.1721'), 'phi_y' (set to '-30.2525'), 'phi_z' (set to '7.4183'), 'rotstart' (set to '340.0'), 'rotend' (set to '390.0'), 'osc/inc' (set to '2.0'), 'nwmax' (set to '3'), 'spot_size' (set to '0.9'), 'lam_min' (set to '0.5'), 'lam_max' (set to '1.5'), 'synch' (set to 'Yes'), 'disp' (set to '0.001'), 'divv' (set to '0.025'), 'divh' (set to '0.15'), 'delcor' (set to '0.00'), 'pol' (set to '0.0'), 'det_geom' (set to 'flat'), and 'weiss_c' (set to '0.00'). The interface is designed for configuring X-ray diffraction simulations.

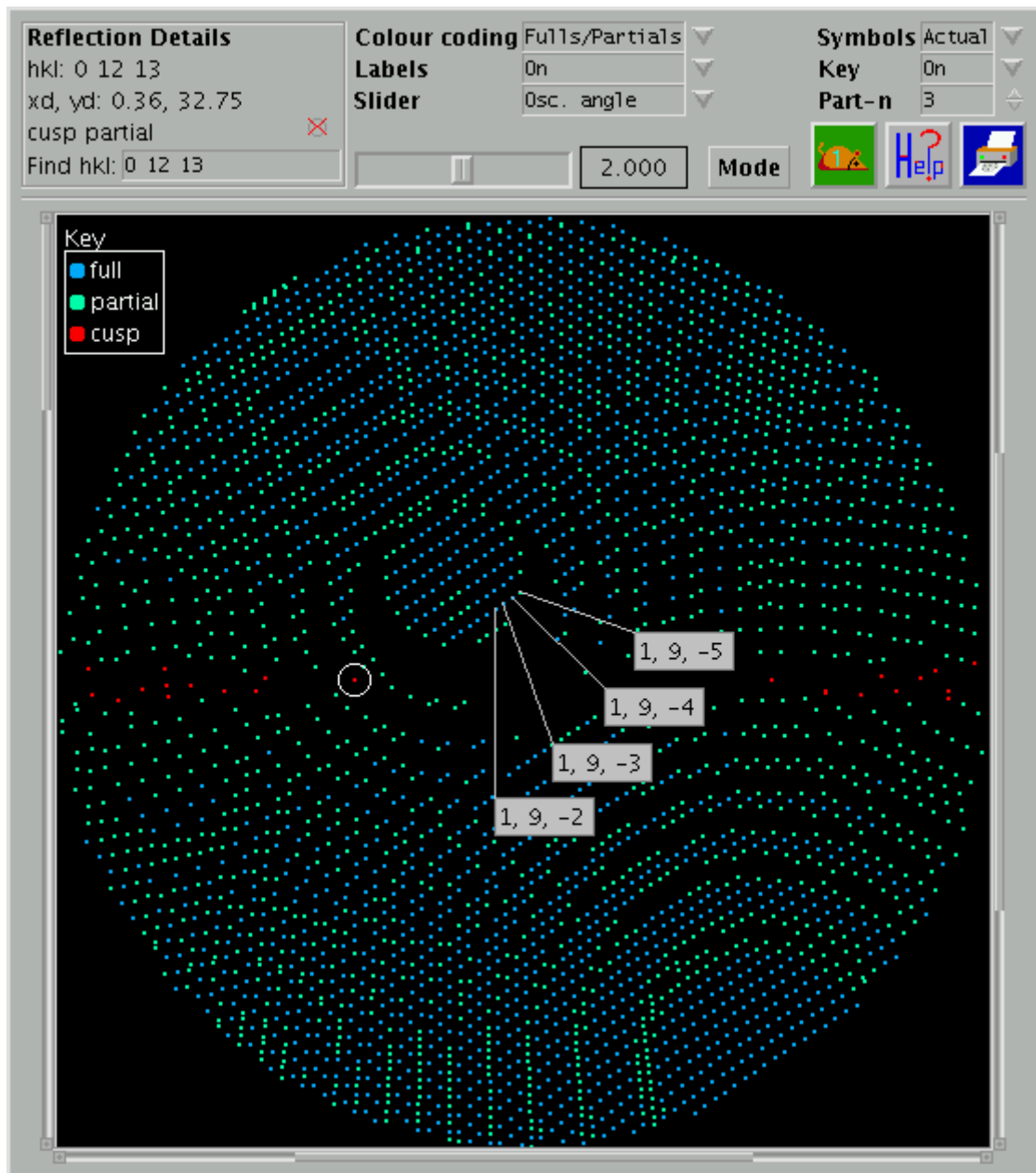
The main options of the programs are as follows:

1. Rotation/Weissenberg/Laue Simulations

Two basic types of simulations are available, standard colour simulations and interactive colour simulations. In each case a window is created with a display area for the simulation, an area to list details of a selected reflection, a control panel and a button used for requesting printer or file (JPEG or PNG) output of the simulation. The standard colour simulations show the patterns with the spots colour coded in a number of different ways. The interactive simulations have a slider which allow the user to investigate the effects of changing various parameters such as the oscillation range and mosaicity (or Lambda-min, dmin for Laue); spot labelling is also available and it is possible to search for reflections by their indices.

An example of an 'interactive' simulation is shown below:

Figure 2: Example of an Interactive Rotation Simulation

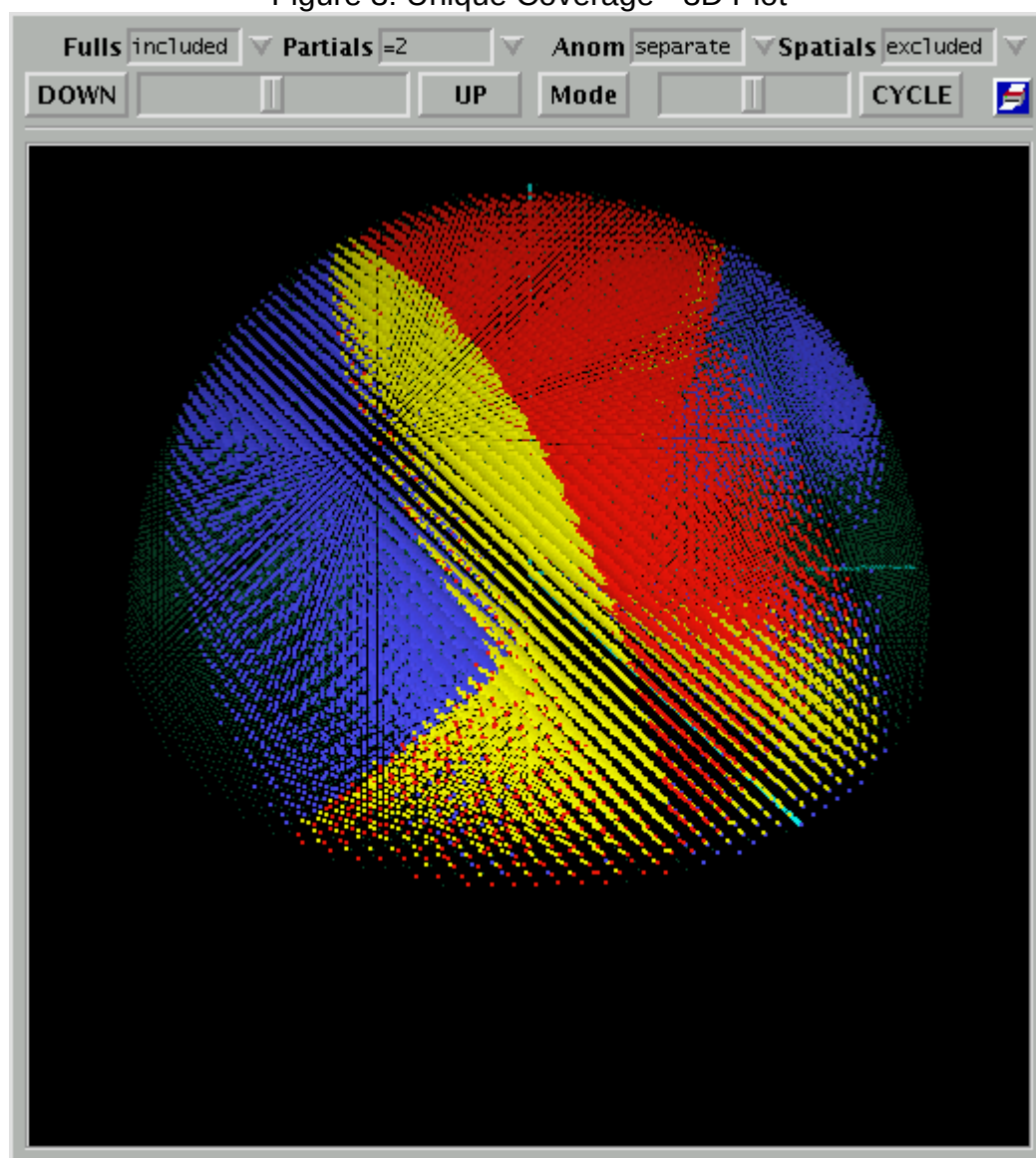


2. Unique Data Coverage

This option enables the prediction of the reflections which would be recorded for the defined crystal sets and to analyse the data coverage in terms of the unique data for the space group, cell and resolution. The analysis may be done for either the current crystal set or for all crystal sets within the data set. The results may be presented in the form of histograms describing the data coverage, in terms of a pictorial representation of the reciprocal lattice sections or as a three dimensional (rotatable) view of the reciprocal lattice.

As an example, I have chosen one of the three dimensional views as this is an option which has been newly developed for this program.

Figure 3: Unique Coverage - 3D Plot



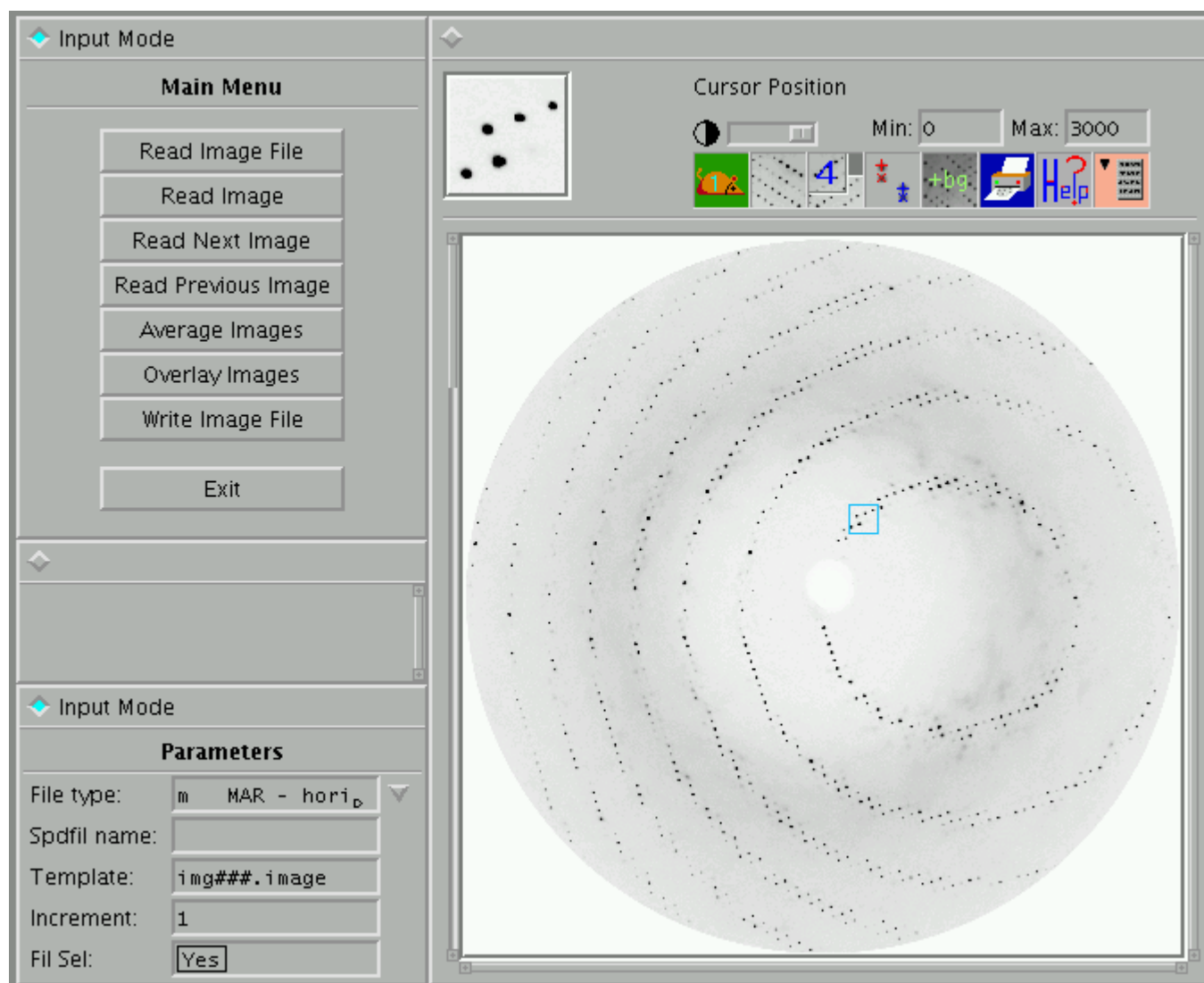
The program is available in [applet form](#) on the CCP4 web site. [Detailed documentation](#) is also available there.

The PXImage Program



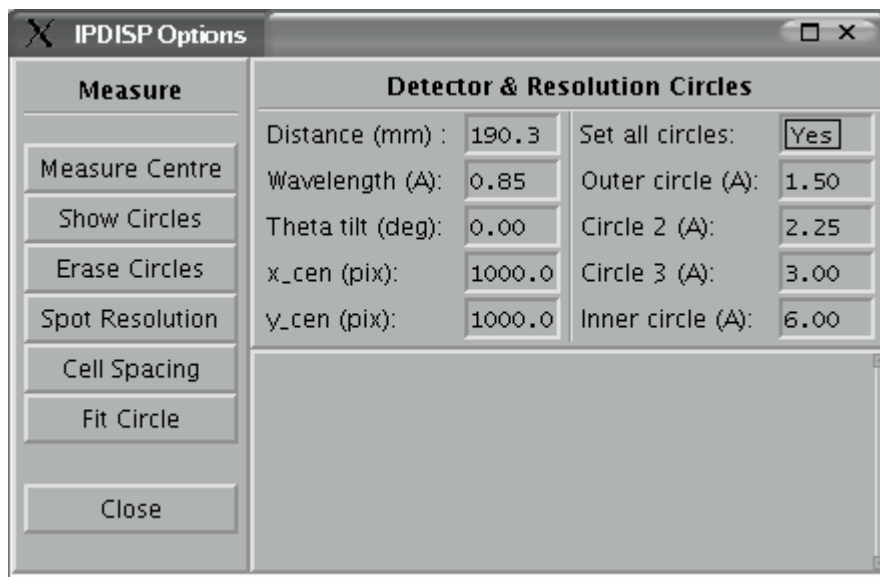
This program incorporates most of the functionality of the program IPDISP. An example is shown below:

Figure 4: PXImage program - main screen



Some of the options for measuring the image are available via a drop down menu and a dialogue window as shown below:

Figure 5: PXImage program - measurement options dialogue



This program is still under development and is not yet available via the CCP4 web site though there is an [image display applet](#) based on the image JdIPXImage class which contains many of the available functions.

The DocExtractor Program



This program is used to prepare documentation files from in a 'pre-xhtml' format or by extracting documentation sections from Java code source files. The output files will normally be in XHTML format. Documentation may be prepared from a single file or a set of linked document files. All internal links (and an optional contents page) are generated automatically by the program. This program has been used to prepare the documentation for the Java Development Library, the Diffraction Data Module and the PXSim program.

PDB Depositions with RCSB PDB Tools

Zardecki Christine



The RCSB PDB (Berman *et al.*, 2000) has developed tools to make depositing structure data easier. There are a few steps a depositor can take to make the process of depositing a structure to the PDB quick, easy, and accurate! It is an iterative process – if you encounter problems at a particular step, please make the correction(s) and go through the steps again.

1. Use the **pdb_extract Program Suite** (Yang *et al.*, 2004)



This software tool extracts information about data collection, phasing, density modification, and the final structure refinement from the output files produced by many applications used for structure determination. The collected information is organized into an mmCIF file that is ready for deposition. Since **pdb_extract** fills in many data fields in this mmCIF file, fewer data items have to be manually entered -- saving time and minimizing errors.

The online version of **pdb-extract** (<http://pdb-extract.rutgers.edu/>) is used to upload output files and create mmCIF files. The files can be used as input to ADIT for a complete deposition.

pdb_extract can be downloaded in source and binary versions for Linux, SGI, SUN, OSF and Mac OSX from <http://sw-tools.pdb.org/>. Source and Linux binary versions of ADIT are also available.

pdb_extract is also part of the CCP4i interface.

2. Check your structure with the Validation Suite and Server



(Westbrook *et al.*, 2003)

to ensure that the data being deposited is accurate and reflects what you intend to submit (<http://deposit.pdb.org/validate/>).

This tool is used by the RCSB PDB for checking and processing structure data. The format consistency of the coordinates can be reviewed, along with validation reports

from MOLPROBITY (Davis *et al.*, 2004), NUCheck (Feng *et al.*, 1998), PROCHECK (Laskowski *et al.*, 1992), and SFCHECK (Vaguine *et al.*, 1999). Sequence/coordinate alignment, missing and extra atoms or residues, and data inconsistencies are also reported here.

The Validation Server is also available from PDB Japan (PDBj) at <http://pdbdep.protein.osaka-u.ac.jp/validate/>.

3. Run BLAST (<http://www.ncbi.nlm.nih.gov/BLAST/>) to compare your sequence to sequence database references (Wheeler *et al.*, 2005). Any necessary corrections can then be made to your sequence and coordinates.

4. Use Ligand Depot



at <http://ligand-depot.rutgers.edu/> to find the proper codes for existing ligands, to link to other entries with a particular ligand, and to search for substructures (Feng *et al.*, 2004). If the ligand is not present in the dictionary, please email detailed information (complete chemical name, 2D figure showing connectivity, bond order and stereochemistry) along with the RCSB and PDB IDs of the entries they are associated to expedite the processing of these files.

5. Deposit your structure using ADIT (<http://deposit.pdb.org/adit/>).



Using its editor to add any missing information to the deposition. A desktop version of ADIT is also available.

ADIT is also available from PDBj at <http://pdbdep.protein.osaka-u.ac.jp/adit/>.

For First Time Depositors

For a detailed packet of information about these tools, including reprints about validation and Ligand Depot, please send your postal address to info@rcsb.org with the subject line "first time depositor packet".

Acknowledgements

The Protein Data Bank (PDB) is operated by Rutgers, The State University of New Jersey and the San Diego Supercomputer Center at the University of California, San Diego -- two members of the Research Collaboratory for Structural Bioinformatics (RCSB). The RCSB PDB is supported by funds from the National Science Foundation (NSF), the National Institute of General Medical Sciences (NIGMS), the Office of Science, Department of Energy (DOE), the National Library of Medicine (NLM), the National Cancer Institute (NCI), the National Center for Research Resources (NCRR), the National Institute of Biomedical Imaging and Bioengineering (NIBIB), and the National Institute of Neurological Disorders and Stroke (NINDS).

The RCSB PDB is a member of the wwPDB.

- 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.
- Davis, I. W., Murray, L. W., Richardson, J. S. & Richardson, D. C. (2004). *Nucleic Acids Res* **32**, W615-619.
- Feng, Z., Chen, L., Maddula, H., Akcan, O., Oughtred, R., Berman, H. M. & Westbrook, J. (2004). *Bioinformatics* **20**, 2153-2155.
- Feng, Z., Westbrook, J. & Berman, H. M. (1998). Report NDB -407. Rutgers University, New Brunswick, NJ.
- Laskowski, R. A., MacArthur, M. W., Moss, D. S. & Thornton, J. M. (1992). *PROCHECK v.2.0 - programs to check the stereochemical quality of protein structures*.
- Vaguine, A. A., Richelle, J. & Wodak, S. J. (1999). *Acta Crystallogr D Biol Crystallogr* **55**, 191-205.
- Westbrook, J., Feng, Z., Burkhardt, K. & Berman, H. M. (2003). *Meth Enz* **374**, 370-385.
- Wheeler, D. L., Barrett, T., Benson, D. A., Bryant, S. H., Canese, K., Church, D. M., DiCuccio, M., Edgar, R., Federhen, S., Helmberg, W., Kenton, D. L., Khovayko, O., Lipman, D. J., Madden, T. L., Maglott, D. R., Ostell, J., Pontius, J. U., Pruitt, K. D., Schuler, G. D., Schriml, L. M., Sequeira, E., Sherry, S. T., Sirotkin, K., Starchenko, G., Suzek, T. O., Tatusov, R., Tatusova, T. A., Wagner, L. & Yaschenko, E. (2005). *Nucleic Acids Res* **33**, D39-45.
- Yang, H., Guranovic, V., Dutta, S., Feng, Z., Berman, H. M. & Westbrook, J. (2004). *Acta Crystallogr D Biol Crystallogr* **60**, 1833-1839.

Application of Automation to Data Processing & Analysis

Graeme Winter (g.winter@dl.ac.uk)

Daresbury Laboratory, Kecwick lane Warrington WA4 4AD

Abstract

Automation and data processing are discussed, followed by a description of a software package "XIA-DPA" for automated data processing & analysis.

Introduction

Processing X-Ray diffraction data can be a pain. It's fiddly, time consuming and often requires careful decision making to get the best from the data. However, careful processing is often critical to the structure determination process. To some extent, "automation" can help.

Often the easiest to use tool is not the best, and many programs have interfaces which are best accessed through inherited command scripts. Again, "automation" can help.

But what is automation?

To some people it is an "intelligent GUI" (e.g. CCP4i) which can help you to compose command scripts to programs and can chain together small programs to produce useful compound tasks. To others it may be a complete system for processing your data & solving the structure (e.g. Elves, Holton & Alber 2004). To some extent there is no right answer here - the appropriate level of automation will depend on the problem in hand. If your data are good and the problem relatively straightforward, then the complete automation approach may be appropriate. If, on the other hand, you are trying to push the boundaries of macromolecular crystallography then you might find yourself doing most of the work by hand.

Automation

Automating the process of protein crystallography is not a new idea. Operation of the beam lines, processing data and solving structures are already almost automatic procedures for simple cases. The differences are now in the scope of automation, and the domain of applicability. The scope is getting broader, with more steps in the process becoming integrated, for example in PHENIX, CCP4 and DNA. The emphasis is also changing, moving away from simple cases to become more generally applicable.

The strongest area for applying automatic techniques has historically been in the structure solution post data processing (e.g. CHART (Emsley, CCP4 Newsletter No. 36), SHELX C/D/E (Sheldrick & Schneider, 1997), SOLVE/RESOLVE (Terwilliger & Berendzen, 1999) and Arp/wArp (Lamzin, Perrakis & Wilson, 2002). At this stage one can

generally assume that the data will be of at least a reasonable quality, and no steps are irreversible. There has traditionally been less emphasis on the automation of data processing, although Elves (Holton & Alber, 2004) includes ``Wedgers" to perform this task.

Applying automation techniques to data processing offers a couple of key advantages. Firstly the results will be reproducible, since exactly the same procedures will be followed every time. Secondly, the decision about the best program to use will not be limited by the user's experience, for instance preferring one program over another due to familiarity alone. Finally, for novice users and those who care more about the biology than the technique, the processing and analysis can be performed with little or no effort.

One area where automation of data processing is critical is in remote and automated data collection, in the DNA ¹ and e-HTPX ² projects. In e-HTPX, a key aim is to allow the remote user to process and analyse the data collected on their behalf. Traditionally this would involve providing a remote graphical interface to the data processing packages. Automating data processing can allow the same functionality but with far fewer transactions, making remote access more reliable. To this end, a small suite of programs for automated data processing and analysis, XIA-DPA, has been developed and will be described here. Finally, a key aim of the automation of data processing in e-HTPX is to ensure compatibility with other ongoing automation efforts, in particular DNA and CCP4, with the HAPPy and BMP projects.

Data Processing

The processing of X-Ray diffraction data is already reasonably automatic. With a relatively small number of keystrokes and mouse clicks an experienced user can go from a sweep of diffraction images through to intensities or structure factors. There are, however, a small number of ``gotchas" which can upset the process, including:

- Incorrect assignment of lattice parameters.
- Transforming between coordinate frames.
- Transforming between file formats.

These are frustrating in that they require concentration, which may be in short supply during a long synchrotron visit.

There are a large number of tools available for data processing, with different strengths and weaknesses. Tools such as Mosflm/CCP4 & HKL2000 provide the user with a complete suite and complete graphical user interfaces. This, combined with modest documentation, can help the most inexperienced user to to a reasonable job. However, the provision of a user interface implies an important component: *the user*. The features they should be looking for (e.g. spot profiles, lattice penalties) can be explained in vague terms, then experience gained. This is harder for an automated data processing system. A solution is to combine tools which are most amenable to automation with expertise to handle the decision making & judgement - between them the challenges may be overcome.

XIA-DPA

XIA, a crystallographic infrastructure for automation, is derived from software developed for the DNA project, and essentially provides a small "core" of tools for running software in an automated environment. On top of this the "DPA" toolkit has been developed, to provide a set of automated programs for data processing & analysis of X-Ray diffraction data, building on a number of commonly available tools, including Mosflm (Leslie, 1992), Labelit (Sauter, Grosse-Kunstleve & Adams, 2004), XDS/XSCALE (Kabsch, 1988a/b), Pointless (Evans, unpublished) and the CCP4 suite. The objective of the suite is to provide some small "macro" tasks which make use of the (in the author's opinion) currently best programs for each given task, although it is straightforward to incorporate new and improved programs. The core of the suite consists of two programs, `xia-autoprocess-2d` and `xia-autoprocess-3d`, which perform the operations shown in Figure 1.

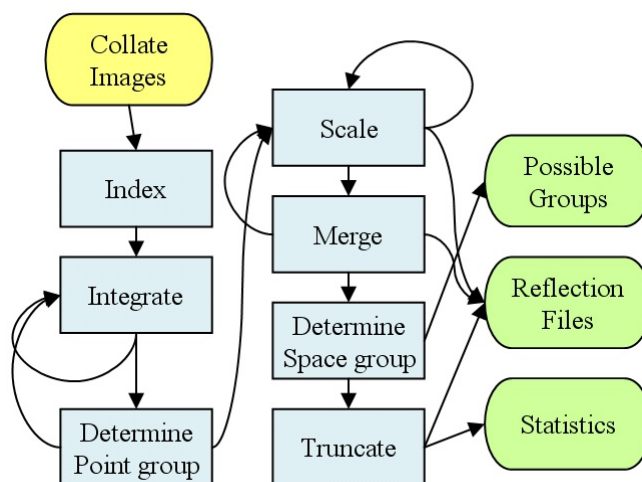


Figure 1: The "workflow" for processing X-Ray diffraction data.

This flow will not be news to anyone familiar with processing & analysing diffraction data, but it does include a couple of gremlins - like performing the scaling correctly and getting ideas of the possible spacegroups. Such things are straightforward, and indeed have "been done", but need to be considered when automating.

The Bottom Line

Essentially, the ambition of XIA-DPA is to

- Process your data sufficiently well that you can use it.
- Tell you something useful about the data.
- Fill the gap between data collection and phasing, either *via* MR or experimental phasing.
- Do fiddly things so you don't have to!

... from only the frames.

At the moment the software (version 0.1.4) is available at the SRS Daresbury and ESRF - please contact the author if you would like access.

The Manual

Properly automated software (indeed, ideally all software) shouldn't need a manual. However, some guidance for the user is always useful. The programs in XIA-DPA all have the same general user input style:

```
xia-\${tool} [options] /path/to/image.img.
```

It is worth making a new directory for working in since this will make a lot of files!

Examples

Simple: 1VPJ

Much of the test data used in developing DPA has been provided by the Joint Structural Genomics Consortium [3](#). The example data sets which led to the structure 1VPJ were of particularly good quality, and hence represent a simple example set to show the process.

Starting the Processing

To get the ball rolling the following commands were issued:

```
2d> mkdir infl lrm peak
2d> cd infl/
2d/infl> xia-autoprocess-2d \
/media/data1/12287/12287_1_E1_001.img > autoprocess.log 2>&1 &
[1] 19206
2d/infl> cd ../lrm/
2d/lrm> xia-autoprocess-2d \
/media/data1/12287/12287_1_E2_001.img > autoprocess.log 2>&1 &
[2] 19215
2d/lrm> cd ../peak/
2d/peak> xia-autoprocess-2d \
/media/data1/12287/12287_2_001.img > autoprocess.log 2>&1 &
[3] 19224
```

Following which, the author went to find a cup of coffee. On return the processing was complete. There were a large number of files in each directory, but the key ones were (in infl above):

```
2d/infl> ls *mtz *log *sca
12287_1_E1_001.mtz      3_ipmosflm.log
12287_1_E1_090.mtz      3_pointless.log
12287_1_E1_.mtz         4_ipmosflm.log
12287_1_E1_scaled.mtz   4_reindex.log
12287_1_E1_scaled_reindex.mtz 5_ipmosflm.log
12287_1_E1_scaled.sca   autoprocess.log
12287_1_E1_scaled_sorted.mtz messenger-1133866582.log
12287_1_E1_truncated.mtz messenger-1133866604.log
2_ipmosflm.log          messenger-1133866851.log
```

The files 12287_1_E1_scaled.mtz and 12287_1_E1_truncated.mtz are the merged intensities and structure factors in CCP4 MTZ format, and 12287_1_E1_scaled.sca contained the unmerged but scaled reflections in unmerged scalepack format. The standard output of the processing script going to autoprocess.log contains much useful information, some of which is highlighted below.

Looking at the Output

The first section of the output describes the data set:

```
-----
Data set: 12287_1_E1_###.img
Directory: /media/data1/12287
Number of frames: 90 (1 to 90)
-----
First frame: /media/data1/12287/12287_1_E1_001.img
First frame header information:
Direct beam:      105.100 101.050
Distance:         170.00
Wavelength:       0.97966
Energy (eV):      12655.86052
Resolution Max (A): 1.22525
Dimensions:       2048 2048
Pixel dimensions: 0.10240 0.10240
Oscillation range: 290.000 291.000 (1.000)
-----
Data collection from: Sun Sep 26 14:01:35 2004
                   to:   Sun Sep 26 14:22:03 2004
-----
```

The information included here is derived entirely from the image headers, and is very helpful in the above example, since the order in which the data were collected could be derived - useful in the later analysis.

After the description of the data set, the results of autoindexing were displayed, including the references for the programs used:

```
Autoindexing using images: 1 90
Autoindexing results:
Spacegroup: P4 (75)
Cell: 51.850000 51.850000 158.130000 90.000000 90.000000 90.000000
Mosaic: 0.075000
Refined beam: 108.930000 105.040000
RMS Deviation (mm): 0.048000
References for programs used:
Program: labelit.screen
Robust indexing for automatic data collection.
N.K. Sauter, R.W. Grosse-Kunstleve, and P.D. Adams.
J. Appl. Cryst. 37, 399-409 (2004)

Program: ipmosflm
Leslie, A.G.W., (1992), Joint CCP4 + ESF-EAMCB Newsletter
on Protein Crystallography, No. 26.
```

... always helpful when writing up. The results of cell refinement and integration then follow, which are rather verbose and won't be included here, since they are best viewed through CCP4i's loggraph. Once the integration is complete pointless (Evans, unpublished) is used to determine the most likely point group:

Reindexed (pointless) from P 4 to P 4 2 2

A small cad script is included, in the cases where the correct lattice is higher than that used in integration, to appropriately constrain the cell parameters for the higher symmetry. Once the point group has been determined the scaling proceeds, with some parameter refinement. Again, there is some output best observed through loggraph, followed by:

```
Data set summary statistics:
Resolution range: 52.56 - 1.44 (1.52 - 1.44)
Rmerge:          0.095 (3.386)
Completeness:    84.21 (39.95)
Multiplicity:    5.65 (2.18)
I/sigma:         9.21 (0.42)
N obs:           189003 (4858)
N unique:        33450 (2232)
Resolution estimates: I/sig Resolution
                   1.00 1.52
                   2.00 1.61
                   3.00 1.66
```

A summary of the merging statistics. Finally, the systematic absences are analysed and compared with theoretical values computed from the CCP4 symmetry libraries to give:

```
Likely spacegroups:
p 41 21 2
p 43 21 2
```

Like all output from the program, this should be taken as advice rather than fact. However, this may greatly reduce the amount of time taken to test all of the possible spacegroup options.

Some Analysis

From the descriptions of the data set, the author discovers that the inflection and low remote passes of the data set were collected first (and at the same time) followed by the peak. It is therefore possible that the peak wavelength is affected by radiation damage. As an initial assessment, the command `xia-multi-radiation` was used to get the relative **B** and **R** factors between the wavelengths:

```
2d/infl> cd ..
2d> mkdir compare
2d> cp */*truncated.mtz compare/
2d> cd compare/
2d/compare> xia-multi-radiation 12287_1_E1_truncated.mtz \
12287_1_E2_truncated.mtz 12287_2_truncated.mtz
B factor = 0.000
B factor = -0.249
```

B factor = -1.972
R factor = 0.000
R factor = 0.082
R factor = 0.142
References for programs used:
Uncited program: cad

Uncited program: scaleit

indicating that the peak had a large relative **B** factor and a poor **R** factor. When phasing later, the results may be better excluding the peak. As a test the structure was solved using Solve/Resolve and Arp/wArp using as both two and three wavelength MAD, resulting in

set	R/R_{free}	Res/Chains start	Res/Chains final
3	26.7/31.7	158/3	164/3
2	21.0/23.4	170/5	173/4

indicating both that the data were good enough to automatically solve with no ``clever'' user input, and also that the results were improved (slightly) by the exclusion of the peak. The initial number of residues docked by Arp/wArp is an interesting metric, since it essentially represents the tracability of the first map.

More Challenging: OPPF1314

This was a much more challenging case for the automated data processing, encountered during a SPINE WP7 workshop in York (publication pending). The data were collected from a triclinic crystal in two runs, low (2.3 Å) and high (1.5 Å) resolution, at the ESRF ID14 EH1. The data were manually processed with Denzo and Scalepack, and a solution *via* molecular replacement was found. However, the refinement and rebuilding of this solution failed, and the processing of the high resolution data was believed to be responsible.

Automatically processing with the 2D pipeline shown above resulted in similar data. However, processing with the 3D pipeline gave datasets which appeared to have more reasonable second moment statistics. This illustrates the flexibility of being able to use multiple programs - the high mosaic spread gave rise to reflections spread over many images in the high resolution pass (0.5° oscillations, $\mu=1.5^\circ$), requiring the use of a 3D integration program to get the best measurements.

Discussion

In the above simple example there was nothing which could not have been done, possibly better, by hand. However, the process would have been time consuming and possibly error prone. As it stands the structure was solved with essentially no user effort, only the results of data collection (the frames and f' and f'' values), a small amount of information about the sample (sequence & number of Se atoms) and processing results were used.

In the second example the automation did not completely solve the problem, but it did substantially help. Being able to process the data automatically with two separate pipelines and compare the results ensured that the best processing results were found quickly.

References

The CCP4 Suite: Programs for Protein Crystallography, Acta Cryst. D50, 760-763.

Emsley, CCP4 Newsletter No. 36:
http://www.ccp4.ac.uk/newsletters/newsletter36/08_chart.html

Holton, J. and Alber, T. (2004). PNAS 101, 1537-1542.

Lamzin, V.S., Perrakis, A. & Wilson, K. (2001) IUCR International Tables Volume F. 720-722.

Leslie, A.G.W., (1992), Joint CCP4 + ESF-EAMCB Newsletter on Protein Crystallography, No. 26.

Kabsch, W. (1988a). J. Appl. Cryst. 21, 67-71.

Kabsch, W. (1988b). J. Appl. Cryst. 21, 916-924.

Sauter, N., Grosse-Kunstleve, R., and Adams, P. (2004). J. Appl. Cryst. 37, 399-409.

Sheldrick, G. & Schneider, T. (1997) Methods in Enzymology 277, 319-343.

Terwilliger, T. & Berendzen, J. (1999) Acta Cryst. D55, 849-861.

Footnotes

... DNA¹
<http://www.dna.ac.uk>
... e-HTPX²
<http://www.e-htpx.ac.uk>
... Consortium³
<http://www.jcsg.org>

HAPPy progress

Dan Rolfe, Charles Ballard, Maria Turkenburg, Eleanor Dodson*, Paul Emsley**

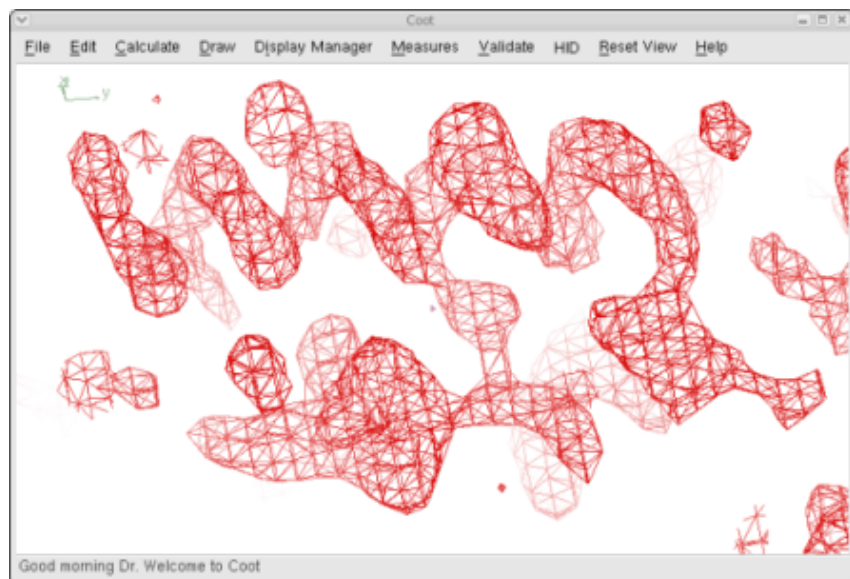
Daresbury Laboratory, Kekwick Lane, Warrington WA4 4AD

** University of York , York YO10 5YW*

Work on the new experimental phasing system, HAPPy (Heavy Atom Phasing in Python), discussed in the previous newsletter, is progressing well. We are testing several packages for inclusion as phasing modules including MLPHARE and PHASER. HAPPy can now take a SAD dataset from the post data processing stage to a refined map with no interaction from the user after providing the initial problem description. The current priority is determining the optimal parameters for the major components and systematic testing and analysis of results to improve the quality of solutions obtained.

There has also been work on the history and data tracking for the next generation CCP4i database and the development of an experimental phasing datamodel for the database being developed by CCP4 in the BioXHIT framework.

The diagram on the right, generated automatically by the HAPPy tracking database, shows the flow of a simple example run. Note that this shows what happened for this particular run; the procedure can become more complicated depending on the dataset and options selected. In this example, HAPPy first ran SHELXD with various sets of resolution and anomalous difference limits, then chose the best resulting heavy atom solution for phasing with MLPHARE. After determining the correct hand for the heavy atom substructure by looking at variance maps of the MLPHARE output, the final map and phases were refined using PIRATE. The figure below shows a section of the final map in COOT.



Xtrriage and Fest: automatic assessment of X-ray data and substructure structure factor estimation

P.H. Zwart, R.W. Grosse-Kunstleve & P.D. Adams

Lawrence Berkeley National Laboratory, 1 Cyclotron Road, BLDG 64R0121, Berkeley California 94720-8118, USA – Email: PHZwart@lbl.gov; www: <http://cci.lbl.gov>

Xtrriage

A command line utility that allows the user to rapidly assess the quality and specific idiosyncrasies of an X-ray dataset has been developed. The program, called *Xtrriage*, combines the twin analyses tools as described in a previous CCP4 newsletter (Zwart, *et al.*, 2005) with other data quality indicators.

In the following sections, the various steps in the characterization of an X-ray data set as carried out by *Xtrriage*, are given.

Unit cell content analyses

The number of residues or nucleotides in the asymmetric unit is estimated on the basis a prior distribution of the solvent content, similar to those used by Kantardjieff & Rupp (2003). In contrast to the distributions obtained by Kantardjieff & Rupp, the distribution used in *Xtrriage* is not conditioned on resolution and describes the probability of a given solvent content rather than a Matthews coefficient. Ideally, the estimate of the number of monomers in the asymmetric unit on the basis of the solvent content should be combined with an analysis of a self rotation function. The estimate provided by the current implementation, is however sufficient in most cases.

Anisotropic Wilson scaling

The anisotropic Wilson B tensor is estimated as described by Popov & Bourenkov (2004) and Zwart *et al.*, (2005). The resulting estimate of the overall B tensor is used to correct for anisotropy in the data. The likelihood based Wilson scaling routines have furthermore been shown to be less sensitive to resolution truncation than classic Wilson analyses, as can be seen from Figure 1.

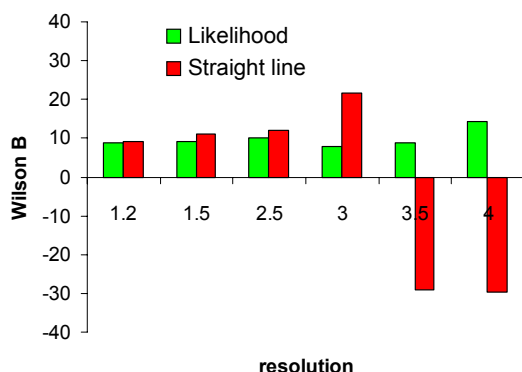


Figure 1: Comparing likelihood based Wilson scaling with classic (straight line) Wilson scaling by progressive truncation of the high resolution limit.

Outlier rejection

Possible outliers are identified using a procedure similar to that as described by Read (1999). The method is based on the distribution of the largest order statistic (Dudewicz & Mishra, 1988) of the intensities. The cumulative distribution of the intensities, assuming an acentric Wilson distribution of the amplitudes, is equal to

$$F(z) = 1 - \text{Exp}[-z] \quad (1)$$

For a set of N_{refl} normalized intensities, $\{z\}$, the largest value, denoted by z_{max} , out of the set $\{z\}$, is then distributed as

$$F(z_{\text{max}}) = (1 - \text{Exp}[-z_{\text{max}}])^{N_{\text{refl}}} \quad (2)$$

The p -value of the associated largest normalized intensity is then equal to

$$p(z_{\text{max}}) = 1 - (1 - \text{Exp}[-z_{\text{max}}])^{N_{\text{refl}}} \quad (3)$$

and quantifies the probability that the largest normalized intensity in a data set of size N_{refl} , is equal or larger than z_{max} . In this manner, the outlier analysis takes into account the size of the dataset. Given for example a data set with 10000 reflections, one can be 99% sure that the largest E-value has a magnitude not greater than 3.72. However, for a data set containing 1 million observations, the latter limit is equal to 4.29. In *Xtriage*, reflections with a p -value lower or equal than 1% are considered potential outliers.

Wilson plot analyses

A Wilson plot analysis as performed by *ARP/wARP* (Morris *et al.*, 2004) has shown to be a powerful sanity check for the quality of macro-molecular data. Within *Xtriage*, an empirical Wilson plot is used that has been obtained using over 2500 high resolution data sets obtained from the *EDS* (Kleywegt *et al.*, 2004). Empirical Wilson plots for both protein and DNA/RNA structures are present, and can be used to construct expected intensity profiles for protein-DNA or protein-RNA complexes as well. Figure 2 illustrates the difference between the mean intensity of a protein structure and a DNA/RNA structure.

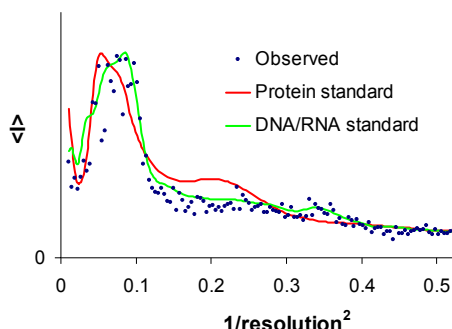


Figure 2: Mean intensity as a function of resolution. In red, the 'standard' protein mean intensity is shown. The 'standard' DNA/RNA curve is shown in green. In blue, the observed mean intensity of a sample DNA structure is shown.

Ice ring detection

If for some reason there was a significant build up of ice on the crystal during data collection, or if no suitable cryo-conditions were available, the diffraction from ice ("ice rings") can have a detrimental effect on the data. If the ice rings are sufficiently diffuse, data reduction programs are capable of subtracting the ice intensities by evaluating the background around the diffraction spots of the crystal. However, if the ice rings are too sharp, the variation in the background may be too large to determine a meaningful intensity to subtract. These difficulties in the diffraction images can result in artifacts in

the merged data, or, when suspicious observations are properly rejected, result in a low completeness in certain resolution ranges. The ice ring detection procedure in *Xtriage* monitors ice ring sensitive resolution areas (Garman & Schneider, 1997) and reports any signs (a local low completeness or a local spike in mean intensity) of ice rings.

Completeness analyses

In order to quickly assess the completeness of the data at low resolution, a completeness analyses is carried out using data up to 5 Å. This analyses is not actively used in further processing of the data, but is provided as a basis for manually judging the low resolution completeness to a greater detail than possible given output of most data reduction or scaling programs.

Analyses of anomalous signal

The presence of anomalous signal is quantified by reporting the fraction of Bijvoet pairs for which the absolute intensity difference is significantly larger than zero, a quantity denoted as the measurability (Zwart, 2005). The resolution range where the measurability lies between 6 and 3% is reported as the resolution range to where the anomalous signal is expected to be useful in substructure solution.

Detection of pseudo translational symmetry

Pseudo translations are located and identified by inspection of the Patterson function using data truncated to 5 Å. Prior information of the distribution of Patterson peak heights in data without pseudo translational symmetry (Zwart *et al.*, 2005) is used to judge on the presence of translational symmetry.

Twinning tests independent of twin laws

Classic indicators of twinning such as the intensity ratio $\langle I^2 \rangle / \langle I \rangle^2$ and the Wilson ratio $\langle F \rangle^2 / \langle F^2 \rangle$, as well as the cumulative normalized intensity distribution are provided. Besides these tests, the L statistic is computed (Padilla & Yeates, 2003), using a Miller index partitioning scheme based on the possible location of pseudo translational symmetry (Zwart *et al.*, 2005).

Derivation of twin laws

Twin laws are determined from first principles as described by Grosse-Kunstleve *et al.* (2005). First, the highest lattice symmetry is determined given the unit cell parameters (Grosse-Kunstleve *et al.*, 2004). Next, the coset-decomposition algorithm as proposed by Flack (1987) is used to enumerate the twin laws, given the space group of the merged data. The derived twin laws are further classified as *merohedral* if the twin operator is a member of the Euclidian normalizer (Koch. & Fischer, 1983) of the given space group of the data, and as *pseudo-merohedral* otherwise. For example, pseudo-merohedral twin laws arise if the space group of the data is monoclinic, but the lattice symmetry is orthorhombic, as is the case for PDBID 1r3h (Rudolph *et al.*, 2004).

Twinning test dependent on twin laws

Using the twin laws as derived above, twin law dependent tests such as the H-test and Britton test are performed. The resulting estimates of the twin fractions, together with the

results from the L-test are used to provide clues as to whether or not the data is likely to be twinned, has a potential non crystallographic symmetry axes parallel to a putative twin law, or if it has been indexed in the wrong space group.

Most results of *Xtriage* can be viewed using the ccp4i program loggraph.

An *Xtriage* tutorial is available online at <http://www.phenix-online.org/tutorials>.

Fest: ΔF and F_A estimation.

The substructure solution engine *HySS* (Grosse-Kunstleve & Adams, 2003), has shown to be a powerful, easy to use and robust tool that allows the user to quickly solve substructure when SAD data is available. Until recently, external tools had to be used to generate ΔF or F_A values given data from substructure solution scenarios such as SIR(AS), RIP(AS) or MAD.

With the goal of furthering automation and integrating basic crystallographic tasks for solving substructures under various experimental scenarios, a number of relative scaling routines have been implemented. The relative scaling and outlier rejection tools needed for ΔF or F_A calculation will be highlighted in the remainder of this report.

Relative anisotropic scaling

An anisotropic scale factor is determined by minimizing the target function:

$$Q = \sum \frac{(I_j^{x1} - k^2 I_j^{x2})^2}{(\sigma_j^{x1})^2 + k^4 (\sigma_j^{x2})^2} \quad (4)$$

where k is the anisotropic scale factor (see Zwart et al, 2005 eq.) to be determined. The super scripts $x1$ and $x2$ are dataset identifiers. Amplitude based target functions, as well as target functions without the use of standard deviations, are available. The optimization is carried out using second derivatives optimization methods currently under development in the *CCTBX* (<http://cctbx.sourceforge.net>).

Local scaling

Local scaling is an effective tool to reduce differences between data sets due to absorption or other sources of systematic errors (Matthews & Czerwinski, 1975) and can play an essential role in obtaining suitable isomorphous, dispersive or anomalous differences needed for a successful substructure solution. Three local scaling functions have been implemented:

1. local least squares target using either intensities or amplitudes (Matthews & Czerwinski, 1975, expression 5):

$$k_{local} = \sum_h X_h^{x2} X_h^{x1} / \sum_h (X_h^{x2})^2 \quad (5)$$

where X denote ϵ corrected intensities or amplitudes.

2. The ratio of local intensity or amplitude moments (Terwilliger, personal communication) :

$$k_{local} = \sum_h X_h^{x1} / \sum_h X_h^{x2} \quad (6)$$

3. The scale factor estimation procedure developed by Nikonov (1983), implemented in a local scaling framework:

$$k_{acentric}^{-1} = \frac{2}{N_{acentric}} \sum_{\mathbf{h}} F_{\mathbf{h}}^{x2} / F_{\mathbf{h}}^{x1} - \left(\frac{1}{N_{acentric}} \sum_{\mathbf{h}} (F_{\mathbf{h}}^{x2})^2 / (F_{\mathbf{h}}^{x1})^2 \right)^{1/2} \quad (7)$$

$$k_{centric}^{-1} = \frac{1}{N_{acentric}} \sum_{\mathbf{h}} F_{\mathbf{h}}^{x2} / F_{\mathbf{h}}^{x1} \quad (8)$$

and

$$k_{local} = (N_{acentric} k_{acentric} + N_{centric} k_{centric}) / (N_{acentric} + N_{centric}) \quad (9)$$

This scaling procedure is designed to take into account the effect that the expected intensity for a derivative is not exactly equal to the expected intensity of the native.

The summation in the three expressions above is carried out around a local sphere around each reflection. Preliminary tests indicate that for reasonable to good data, no specific local scaling method is preferred. However, further analyses are required to make any definite conclusions.

Outlier detection

The presence of outliers in a difference dataset can have a detrimental effect on the success of substructure solution. A few large, but incorrect F_A or ΔF values can dominate the Patterson function, E-maps or triplet relations used to determine the substructure. Outlier detection can be performed using various protocols and criteria. Two following outlier detection schemes are available to the user, both giving similar results:

1. Reject reflections for which $|\Delta F| < C_{\sigma} \sigma_{combined}$ where

$$\sigma_{combined}^2 = \langle \Delta F^2 \rangle_{d^{*2}} - \langle \sigma_{\Delta F}^2 \rangle_{d^{*2}} + \sigma_{\Delta F}^2 \quad (10)$$

The subscript d^{*2} denotes that an averaging over a resolution shell has been carried out. C_{σ} is a constant set to 3, but can be changed by the user.

2. Reject reflections for which

$$F_{\mathbf{h}}^{x1} < C_{\sigma} \sigma_{\mathbf{h},obs}^{x1} \text{ or } F_{\mathbf{h}}^{x2} < C_{\sigma} \sigma_{\mathbf{h},obs}^{x2} \quad (11)$$

and

$$|\Delta F| > C_{rms} \left(\langle \Delta F^2 \rangle_{d^{*2}} \right)^{1/2} \quad (12)$$

C_{σ} is a constant set to 3, C_{rms} is set to 4. These values can be changed by the user, if desired.

After potential outliers have been identified, they are removed from the reflection list, and the data is rescaled. This scheme can be iterated until no more outliers are detected, or can be carried out a fixed number of times. Optionally, the value of C_{rms} can be increased for subsequent outlier rejection cycles.

The first outlier rejection scheme is reminiscent to the once used in Solve (Terwilliger, personal communication). The second set of criteria is similar to those present in CNS (Grosse-Kunstleve & Brunger, 1999) and HySS (Grosse-Kunstleve & Adams, 2003).

***F_A* estimation**

The scaling and outlier routines are combined in a program that allows the calculation of ΔF values or the estimation of F_A values. The program, called *Fest*, is a command line

```
scaling.input {  
  basic {  
    n_residues = None  
    n_bases = None  
    n_copies_per_asu = None  
  }  
  xray_data {  
    unit_cell = None  
    space_group = None  
    reference {  
      file_name = None  
      labels = None  
    }  
  }  
  output {  
    log = fest.log'  
    hklout = 'fest.mtz'  
    outlabel = '_A'  
  }  
}
```

Figure 3: basic input parameters for *Fest*.

driven utility that currently supports ΔF or F_A value estimation for phasing scenarios such as SAD, SIR(AS), 2 wavelength MAD as well as RIP(AS).

The F_A value estimation algorithm for MAD data is currently based on the work of Singh & Ramasheshan (1968) and Kingston (2001) and is limited to 2 wavelength MAD. The CNS based method of averaging isomorphous and anomalous difference Pattersons for obtaining F_A estimates, is also available (Grosse-Kunstleve & Brunger, 1999). Other methods are currently under development.

A large number of options in *Fest* are accessible to the user via a comprehensive parameter file, using the *PHIL* module of the CCTBX (Grosse-Kunstleve *et al.*, 2004; see also figure 3). The control over the behavior of *Fest* is very similar to the macromolecular structure refinement module in *PHENIX* (Adams *et al.*, 2004).

The user has full control over which relative and local scaling targets will be used, as well as over outlier detection parameters. Most of these choices are however hidden by default, unless the *expert_level* is set so as to make all customizable variables visible. The basic input parameters for a SAD experiment can be found in Figure 3. Although default settings should be sensible for all scenarios, the full control the scaling procedure and subsequent F_A calculations could be crucial in unusual cases.

Automatic re-indexing

For each data set supplied (other than the first data set given), all possible re-indexing matrices are derived as described by Grosse-Kunstleve *et al.* (2005). Each data set is automatically re-indexed using the matrix that maximizes the correlation of amplitudes.

Examples: RIP

The data sets 2blu and 2blr, the 'before' and 'after' data sets from a RIP experiment (Nanao *et al.*, 2005), were used to assess the behavior of *Fest* and *HySS* when dealing with RIP data.

The RIP scaling strategy is equal to the scaling strategy for SIR, albeit with an option to downscale the 'after' dataset by a user given constant k (controlled by the keyword *nsr_bias*), in line with the work of Nanao, Sheldrick & Ravelli (2005).

pairs of datasets. The correlation coefficient of the best solution is shown in Table 1 for SAD and in table 2, for the 2 wavelength combinations, using only dispersive differences as well as F_A estimates. Although when using SAD, one is able to locate the substructure without any problem, substructure solution with F_A values gave significantly larger correlation coefficients.

Example: SIR

The results of *HySS* using ΔF 's obtained from *Fest* using native data and a mercury derivative of GroEl are shown in table 3. Note that the mercury derivative data is

Table 3: GroEl SIR *Fest* and *HySS* details.

Spacegroup	C2221	
Nominal resolution	30-3.8 Å	
	Native	Derivative
Completeness	Full: 94% 30-5: 98%	Full: 35% 30-5: 53%
Absolute scaling ln(scale)	-0.67	-0.68
Trace of absolute scaling B_{cart}	51, 47, 66	62, 55, 70
Relative scaling ln(scale)	-0.004	
Trace of relative scaling B_{cart}	-13, -6, 25	
ΔF completeness	34%	
$CC_{\text{consensus}}$	46%	
Sites found (correct)	21 (out of 21)	

incomplete, resulting in a rather incomplete difference data set.

Although the completeness is low, *HySS* is readily able to locate the full substructure, and finds 21 out of 21 sites with an *rmsd* of 0.66 Å to the true heavy atom model.

Discussion

The command line utility *Xtriage* has shown to be a practical tool that can be used quickly after data processing to detect possible twinning, as well as point out incorrect assignment of the space group.

The relative scaling and ΔF estimation techniques implemented in *Fest* seem to be able to handle a variety of data originating from various sources. The F_A estimation for the 2 wavelength MAD case did improve substructure solution as compared to SAD, although the SAD data alone was sufficient to locate the substructure. Currently the value of f' and f'' are required on input for the 2 wavelength MAD case. However, approaches are being developed that do not require manual specification of these parameters.

Fest and *HySS* have shown to be able to deal with RIP and SIR data as well. The full customizability of *Fest* might be of use in unusual cases where the default scaling protocol are suboptimal or when downscaling of a native(-like) data set turns out to be critical.

Acknowledgements

PHZ thanks Tom Terwilliger for stimulating discussion regarding local scaling issues and outlier rejection details as implemented in *SOLVE*. The authors thank Dr. Z. Dauter for making the Acyl-CoA Thioesterase II MAD data set available via the Autostruct (<http://www.ccp4.ac.uk/autostruct/>) web pages.

References

- Adams, P. D., Gopal, K., Grosse-Kunstleve, R. W., Hung, L.-W., Ioerger, T. R., McCoy, A. J., Moriarty, N. W., Pai, R. K., Read, R. J., Romo, T. D., Sacchettini, J. C., Sauter, N. K., Storoni, L. C. & Terwilliger, T. C. (2004). *J. Synchrotron Rad.* **11**, 53-55.
- Dudewicz, E. J. & Mishra, S. N. (1988). *Modern Mathematical Statistics*. New York: Wiley.
- Flack, H.D. (1987). *Acta Cryst.* **A43**, 564-568.
- Garman, E.F. & Schneider, T.R. (1997). *J. App. Cryst.* **30**, 211-237.
- Grosse-Kunstleve, R.W., Sauter, N.K. & P.D. Adams. (2004). *IUCr Computing Commission Newsletter* **3**.
- Grosse-Kunstleve, R.W., Afonine, P.A., Sauter, N.K. & P.D. Adams. (2005). *IUCr Computing Commission Newsletter* **5**.
- Grosse-Kunstleve, R.W. & Adams, P.D. (2003). *Acta Cryst.* **D59**, 1966-1973.
- Grosse-Kunstleve, R.W. & Brunger, A.T. (1999). *Acta Cryst.* **D55**, 1568-1577.
- Kantardjieff, K. & Rupp, B. (2003). *Prot. Sci.* **12**, 1865-1871.
- Kingston, R.L. (2001). *Acta Cryst.* **D57**, 101-107.
- Kleywegt, G.J., Harris, M.R., Zou, J.Y., Taylor, T.C., Wahlby, A. & Jone, T.A. (2004). *Acta Cryst.* **D60**, 2240-2249.
- Koch, E. & Fischer, W. (1983). *International Tables for Crystallography*, Vol. *A*, edited by T. Hahn, ch. 15. Dordrecht: Kluwer Academic Publishers.
- Li, J., Derewenda, U., Dauter, Z., Smith, S., Derewenda, Z. S. (2000). *Nat.Struct.Biol.* **7** 555-559.
- Matthews, B. W. & Czerwinski, E. W. (1975). *Acta Cryst.* **A31**, 480-487.
- Morris, R. J., Zwart, P. H., Cohen, S., Fernandez, F. J., Kakaris, M., Kirillova, O., Vonnrhein, C., Perrakis, A. & Lamzin, V. S. (2004). *J. Synchrotron Rad.* **11**, 56-59.
- Nanao, H.M., Sheldrick, G.M. & Ravelli, R.B.G. (2005). *Acta Cryst.* **D61**, 1227-1237.
- Nikonov, S. (1983). *Acta Cryst.* **A39**, 693-697.
- Padilla, J.E. & Yeates, T.O. (2003). *Acta Cryst.* **D59**, 1124-1130.
- Popov, A. N. & Bourenkov, G. P. (2003). *Acta Cryst.* **D59**, 1145-1153.
- Read, R.J. (1999). *Acta Cryst.* **D55**, 1759-1764.
- Rudolph, M. G., Wingren, C., Crowley, M. P., Chien, Y. H. & Wilson, I. A. (2004). *Acta Cryst.* **D60**, 656-664.
- Singh, A.K. & Ramaseshan, S. (1968). *Acta Cryst.* **B24**, 35-39.
- Zwart, P.H., Grosse-Kunstleve, R.W. & Adams, P.D. (2005). *CCP4 Newsletter* **42**.
- Zwart, P.H. (2005). *Acta Cryst* **D61**, 1437-1448.