CCP4 NEWSLETTER ON PROTEIN CRYSTALLOGRAPHY

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

Number 42 **Summer 2005**

Contents

8. **The Phenix refinement framework** doc
   Afonine P.V., Grosse-Kunstleve R.W, Adams P.D.
   Lawrence Berkeley National Laboratory, One Cyclotron Road, BLDG 64R0121, Berkeley, CA 94720 USA

Methodology

9. **On the Fourier series truncation peaks at subatomic resolution** doc
   Anne Bochow, Alexandre Urzhumtsev
   Physics Department, Faculty of Sciences and Technologies, University H. Poincaré Nancy 1, B.P. 239, 54506 Vandoeuvre-lès-Nancy, France
10. **Characterization of X-ray data sets** doc
    Peter H. Zwart, Ralf W. Grosse-Kunsteleve, Paul D. Adams
    Lawrence Berkeley National Laboratory, 1 Cyclotron Road, BLDG 64R0121, Berkeley California 94720-8118, USA

Editor: Francois Remacle

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

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. HTML is preferred but other editable formats are also acceptable.

CCP4 Main Page

CCP4 General news:

*Peter Briggs, Charles Ballard, Martyn Winn, Daniel Rolfe, Graeme Winter, Ronan Keegan, Norman Stein, Francois Remacle, Paul Emsley**

CCP4, Daresbury Laboratory, Warrington WA4 4AD, UK
**Structural Biology department, York University, York,UK*

Table of Content

Future Release 6.0 of CCP4

General information

The Collaborative Computational Project Number 4 in Protein Crystallography was set up in 1979 to support collaboration between researchers working on such software in the UK, and to assemble a comprehensive collection of software to satisfy the computational requirements of the relevant UK groups. CCP4 was originally supported by the UK Science and Engineering Research Council (SERC), and is now supported by the Biotechnology and Biological Sciences Research Council (BBSRC). The project is coordinated at CCLRC Daresbury Laboratory. The results of this effort gave rise to the CCP4 program suite, which is now distributed to academic and commercial users world-wide.

During its history it passed through different releases. Each of these releases adding new programs from the developers community, offering new tools and techniques to make the suite more complete in order to provide a powerful tool to its users.

Now, version 6.0 is being developed and tested. We are going to outline below its new features and improvements from last release.

What's new

In future releases, the CCP4 Suite will be separated into a number of packages, in order to provide the user with an easier way to download and install the programs, and

in order to facilitate subsequent updates. In release 6.0, the following different packages will be available:

CCP4 program suite: Containing usual programs, libraries, tutorials, examples, CCP4i and new tools as you will see below.

CCP4 Molecular Graphics: From authors Liz Potterton and Stuart McNicholas, CCP4MG enables to displays molecules with simple, flexible selection tools and a variety of display styles and colouring schemes through a simple interface. It also provides different structure analysis.

COOT: From author Paul Emsley, Coot is a tool that enables to display maps and models and allows certain model manipulations: idealization, real space refinement, manual rotation/translation, rigid-body fitting, ligand search, solvation, mutations, rotamers, Ramachandran plots, model validation and others...

Phaser 1.3 and CCTBX: Developed at university of Cambridge, Phaser is a program for phasing macromolecular crystal structures with maximum likelihood methods. It currently has methods for brute force and fast likelihood-based rotation and translation functions for molecular replacement. Methods for experimental phasing are under development.
The Computational Crystallography Toolbox (cctbx) is being developed as the open source component of the PHENIX system. It contains different modules for different purpose in macromolecular crystallography.

CHOOCH: From author Gwyndaf Evans, The program CHOOCH determines values of anomalous scattering factors from raw fluorescence spectra.

The basic CCP4 program suite package will provide a series of new tools:

Bp3*: From author Navraj Pannu, Bp3 is a program for obtaining phase information from an S/MIR(AS) and/or S/MAD experiment(s) by multivariate likelihood estimation. Bp3 takes part in the works done by crank.

Crank*: From author Steven Ness, Crank is a new suite of programs for automated macromolecular structure solution. It uses an XML based framework to join many different crystallography programs into a unified whole. CRANK is intimately linked to the CCP4 package, using CCP4i for job setup and control.

Superpose and SSM: From author Eugine Krissinel, (SSM) Secondary Structure Matching is a tool for protein structure comparison in 3D. Superpose is a program making secondary structure superposition using the functions provided by SSM library.

Pirate: From author Kevin Cowtan, Pirate is a program performing statistical phase improvement by classifying the electron density map by sparseness/denseness and order/disorder, with the aim of obtaining superior results to conventional solvent mask based methods without requiring knowledge of the solvent content.

Clipper Utilities: From author Kevin Cowtan, these are some utilities providing useful functionnalities from Clipper libraries.

Chainsaw: From author Norman Stein, Chainsaw is a utility for Molecular Replacement, which mutates a template pdb file using a sequence alignment between the target and template.

* For more information about the work done by crank and Bp3 you can read the article concerning them in this newletter.

Updates from version 5 of CCP4

In addition to the new series of programs and packages, CCP4 program suite will also include the up to date version of the CCIF, Clipper, MMDB and CCP4 Libraries, the latest versions of Pdb-Extract, Molrep, Mosflm, Refmac, Sfcheck and Scala and the updated version of CCP4i (you can read the article concerning the new version of CCP4i).

OnGoing Projects and Pre-releases

In addition to CCP4 v6.0 there are other projects that are ongoing around CCP4, new delivery systems, new programs. Some of these will available as pre-release together with release of CCP4. Currently there are the following items that will be pre-released:

Linux Install Wizard: Based on the installshield technology this project is trying to create an installer as straightforward and robust as its bigger brother available on windows.

Pointless: From author Phil Evans, Pointless is a program that enables to determine the Laue groups using the symmetry functionalities of CCTBX.

The BIOXHIT Project CCP4 and BIOXHIT

BIOXHIT started in January 2004 and is an "integrated project" funded for four years within the 6th Framework Programme of the European Commission. BIOXHIT is coordinating scientists at all European synchrotrons and leading software developers in a joint effort to develop, assemble and provide a highly effective technology platform for Structural Genomics. CCP4 is involved in workpackages which aim to implement data management and project tracking in structure solution, and in work which complements the CCP4 Automation Project. The project currently funds one-full time programmer.

As a key part of this work the CCP4i database is currently being expanded and standardised as a Project database for non-CCP4(i) applications operating in a multi-user computing environment. The scope of the data stored in the database - both the raw data and the history record information - will also be extended as part of the project, and visualisation tools will be developed to help users make sense of the data.

The aim is to provide a system which is useful for both ongoing "work-in-progress" structure determination projects (being performed either manually or through automated systems). We are working with a variety of different partners both within and outside of the BIOXHIT project to ensure that the system will be compatible with and useful to other software projects. Currently prototypes exist for the database and the "broker" application which mediates access to it. Work is also ongoing on visualisation tools.

CCP4 is also contributing to work within the BIOXHIT framework on data models for information exchange between programs for the purposes of automation. In February 2005 CCP4 co-organised a workshop which brought together the developers of a number of automated systems to discuss the issues, and the final report from the meeting along with the supporting documents can be found at http://www.ebi.ac.uk/msd-srv/docs/bioxhit05_1.html.

The main BIOXHIT website is at http://www.bioxhit.org. Information about CCP4 and BIOXHIT can be found at http://www.ccp4.ac.uk/projects/bioxhit.html. Please contact Peter Briggs (*p.j.briggs@ccp4.ac.uk*) for more information about the CCP4 contribution to the BIOXHIT project.

CCP4 and e-HTPX

e-HTPX is a BBSRC-funded e-science pilot project which aims to link the various stages of protein crystallography into one single all-encompassing interface from which users can initiate, plan, direct and document their experiment either locally or remotely from a desktop computer. The e-HTPX project covers the stages from crystallisation, through data collection to structure solution. The latter is of particular interest to CCP4, and complements efforts in the CCP4 Automation project. Here we describe those aspects of e-HTPX relevant to CCP4 - for more information on e-HTPX itself, see www.e-htpx.ac.uk.

Early work looked at running CCP4 programs on clusters, and parallelisation of the underlying code. Parallelised versions of BEAST and SCALA were written using the MPI library for message passing on distributed memory systems, such as Beowulf clusters. In the former case, the aim is to make it feasible to run a slow program in a reasonable timescale. In the latter case, the aim is to turn a relatively quick program into one that is fast enough to provide real-time feedback during data collection.

Later work has looked at using clusters to do parameter space screening. As an example, a python script has been written that will perform molecular replacement using a variety of template structures, trial model generation methods, and choices of molecular replacement program. This is a very general framework within which a number of different approaches can be tested in parallel. A faster, cut-down version suitable for a desktop will be included in a later version of CCP4.

e-HTPX has also contributed effort to the DNA project, which automates data collection and processing at synchrotron beamlines (home sources may be covered later). Finally, e-HTPX is also developing tools for doing protein crystallography in a Grid environment, which will be relevant for new facilities such as Diamond.

CCP4 automations projects

Autoamore

Autoamore is a project on automated molecular replacement methods, base on the program *AMORE* distributed in CCP4 suite.

Solving a structure by molecular replacement using the Amore program involves running the program many times, for example rotation functions and translation functions have to be solved for separately. If there is more than one molecule in the asymmetric unit, an additional Amore run is required to find each extra molecule. Autoamore is a Python script which automates the whole procedure, thus allowing Amore to be run with no more user intervention than would be required to run other

molecular replacement programs such as Molrep and Phaser. One advantage of using Amore is that it is fast, and therefore particularly attractive to users with less powerful machines.

The Autoamore script calls various other CCP4 programs in addition to Amore. Matthews is used to estimate the number of molecules in the asymmetric unit. Wilson and Baverage are used to determine the difference in B factor between the model and target data, the difference then being input to Amore using the BADD variable. The rotations and translations found by Amore are applied to the atom coordinates using Pdbset and a single output pdb file generated, suitable for subsequent input into model building/refinement programs. A check for clashing is made using Distang. Autoamore generates its own summary file, listing important parameters concisely.

Autoamore also uses the Peakmax program to check if pairs of molecules are potentially related by Translational NCS. If this is the case, the translation vector is supplied to Amore, which then positions molecules on a pairwise basis.

To use Autoamore, the user must create a simple input file, listing the names of the model pdb file and the target mtz file, the column name for the structure factor in the mtz file, the resolution limits desired and the number of residues. Once Autoamore is given the name of this file, no further user input is required. Autoamore forms part of CCP4 Automation and will also be incorporated as a module in the BMP molecular replacement pipeline.

HAPPy

We are working on a new automated experimental phasing system called HAPPy (Heavy Atom Phasing in Python). This project (previously known as PyChart) will replace and expand on the capabilities of Paul's Chart package [1]. The goal is to use processed (i.e. post-TRUNCATE) experimental data, determine the heavy atom structure and phase probability distributions, then take these to optimize the map and potentially build structure. The first release will handle SAD data only, with MAD, MIR and MIRAS modes added later.

As with several other automation projects, HAPPy is being written in Python, and will employ existing packages for the various stages of the structure solution. Where possible, CCP4 [2] programs will be used, but non-CCP4 programs will also be used where appropriate. SHELXD [3] is used for heavy atom substructure determination, followed by Phaser [4] for the SAD phasing and Pirate is used for phase improvement. Buccaneer [5] will be used for the model building in future.

HAPPy will be designed to cooperate with other automation packages, for example using the output from automated data processing software DNA/XIA-DP [6]. Well-defined APIs and data formats will be used wherever data exchange is necessary.

References

[1] Chart: www.chem.gla.ac.uk/~paule/chart
[2] CCP4: www.ccp4.ac.uk
[3] SHELXD: shelx.uni-ac.gwdg.de/SHELX/
[4] Phaser: www-structmed.cimr.cam.ac.uk/phaser/
[5] Buccaneer: www.ysbl.york.ac.uk/~cowtan/buccaneer/buccaneer.html
[6] XIA: Graeme Winter, in preparation

Developments in CCP4i: July 2005

Peter Briggs, Francois Remacle, Martyn Winn, Charles Ballard

CCP4, Daresbury Laboratory, Warrington WA4 4AD, UK

Introduction

CCP4i is the CCP4 graphical user interface. The last officially released (and still current) version of the interface is 1.3.19, which is included as part of CCP4 5.0.2. Some of the significant changes and updates in that version included:

- **New interfaces:** for MOSFLM-in-batch, AREAIMOL and ClustalW
- **Data Harvesting Management Tool:** to facilitate reviewing the harvest files during or at the end of a project
- **Viewing and Graphics Utilities module:** grouped together utilities like MAPSLICER, LOGGRAPH, TOPDRAW and others for easy access
- **Bubble help:** also referred to as "balloon help" or "tooltips", these are the little yellow bubbles that appear when the cursor lingers over a particular widget. Bubble help is turned on or off via the "Configure Interface" window (accessed from the "System Administration" button).

There were also many bug fixes and other minor improvements.

Since then further work has been done on the interface, and this article outlines the major changes in the next (currently officially unreleased) version of CCP4i, version 1.4.0. CCP4i 1.4.0 will be available in the next release of CCP4, version 6.0.

New and Updated Interfaces in 1.4.0

There are a number of significant new task interfaces in CCP4i 1.4.0:

- **Crank** is a suite of programs for automated macromolecular structure solution, which has been developed at the University of Leiden in the Netherlands and which makes extensive use of the CCP4i infrastructure. Currently **Crank** supports SAD, SIR and SIRAS experiments (MAD and MIR(AS) are being added) and makes use of various new and existing programs, including BP3, SHELX and various CCP4 programs.

  Crank is a fully functional suite and allows the solution of macromolecular structures up to the point of density modification. At the same time it has also been designed to help teach novice users about the various programs used in crystallography (a so-called "translucent box" design). It can be found as part of the "Experimental Phasing" module in CCP4i.

- **shelx_cde** is a new CCP4i interface to the Goettingen SHELX programs and facilitates running various combinations of SHELX C, D and E. The task takes either SCALEPACK format reflection files (note that the SCALA task can now output SCALEPACK-style files that are suitable for input into SHELX and

SOLVE, amongst others) or MTZ files containing intensities (preferred) or structure factor amplitudes) as input.

The task can be used to run the programs in a "pipeline" fashion from data preparation through heavy atom site location to density modification and hand determination, and generates useful plots from the output of each program. It can also output the phases for each hand in MTZ format.

- There are also interfaces to accompany the new programs PHASER (in the "Molecular Replacement" module), BP3 (in "Experimental Phasing"), PIRATE (in "Density Modification") and Clipper utilities (which have their own new module). There is also a new task which enables the CCP4 molecular graphics package CCP4mg to be launched from within CCP4i (in the "Viewing and Graphics Utilities" module).

In addition there are updates to a number of other tasks, for example the Accessible Surface Area (areaimol), Cell Content Analysis (matthews), Edit PDB (pdbset/pdbcur) and others, in order to accommodate new functionality available in the underlying CCP4 programs.

Another change in CCP4i 1.4.0 is that you may notice that the text on the buttons for certain tasks are "greyed out", and that the tasks themselves cannot be launched (for example, PHASER or SHELX). This is because the underlying software that the task uses is not available (for example, the SHELX programs are not installed on your path). In this case you should check the CCP4i documentation to find out which programs are missing, and how to install them. Once the required software is installed CCP4i will automatically detect it and make the task available again.

New Core Functionality

CCP4i 1.4.0 includes some new tools that will be useful to long-term users of the interface: **Database Search/Sort Utility**, **Job Database Display Colour Customisation** and **CCP4i project shortcuts**. Each of these is described in the following sections:

1. Database Search and Sort Utility

This is a powerful new utility that allows the user to search and sort the contents of the current project database. Searching can be performed using various criteria, including:

- Task name and status
- Job title and date
- Associated input and output files

Different searching and sorting criteria can be combined to easily perform powerful searches on the contents of the job database, and the database utility tools from the main window (e.g. Rerun job, Delete/Archive etc) are also available in the search window, as can be seen in the screenshot in figure 1.

This utility should be very valuable when reviewing the contents of the job database, and is accessed from the **Search/Sort** button in the utilities menu on the right-hand side of the main CCP4i window.

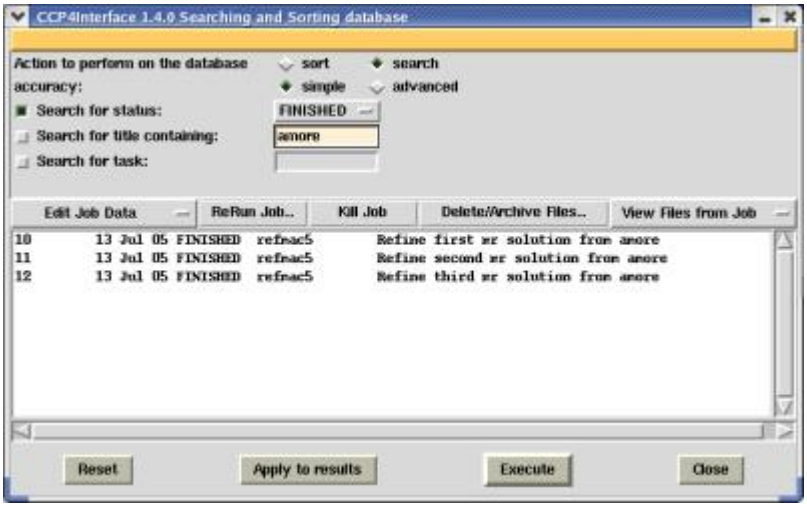


Figure 1: An example of the job database search & sort utility window, showing the results of a query

2. Customisation of the Job Database Display Colours

Traditionally the job database display in CCP4i has consisted of a flat "black-on-grey" colour scheme. This new feature allows users to customise the display by choosing their own background and text colour schemes. A default colour scheme can be set for the entire display, while further customisation allows jobs to be displayed in different colours based on various criteria (for example, job status or task name).

An example of a custom colour scheme is shown in figure 2a (below left), where colouring is by job status. This functionality is useful in helping to distinguish between different jobs in a single database. The customisation options can be accessed from the "Configure interface" window (see example in figure 2b, below right), which is launched from the "System Administration" button on the right-hand side of the main CCP4i window.

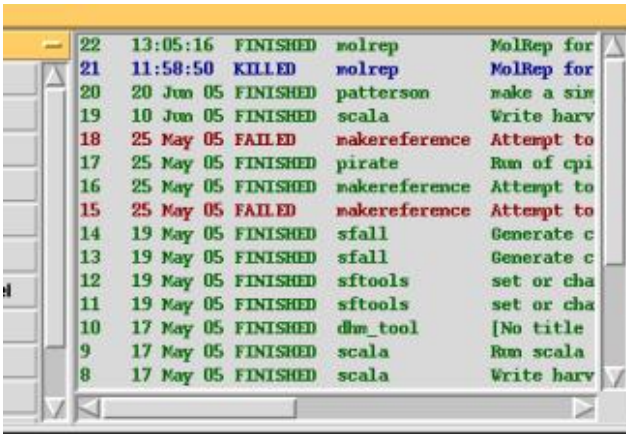


Figure 2a: An example of the job database coloured using custom settings

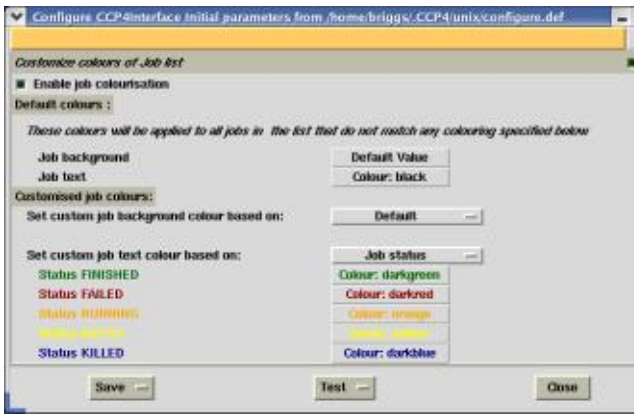


Figure 2b: Setting the custom colours in the "Configure Interface" window

3. Shortcuts Between Projects

This is accessed via a new menu button labelled "Change Project", located in the top right-hand corner of the main CCP4i window, next to the "help" button. Clicking on this button brings up a list of the available projects (with the current project in italics), as shown in figure 3 (right) - selecting a project name closes the current project and opens the new one.

This shortcutting avoids the need to bring up the "Directories&ProjectDir" window each time the user wants to change between projects. It is still necessary to access this window if you want to create a new project.

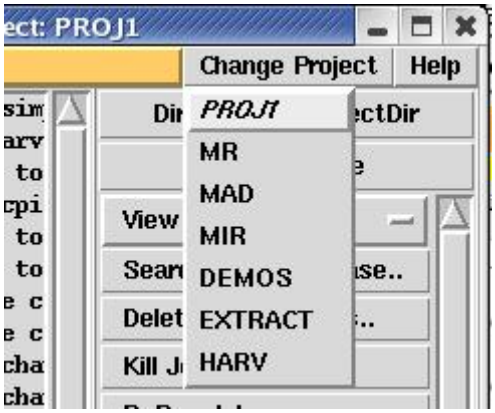


Figure 3: An example of the list of available projects accessed from the "Change Project" menubutton

Other updates

MapSlicer is now able to read in CCP4 format mask files and display them in "mask" mode (figure 4a, below left). It can also render sections from normal maps in a "greyscale" format, viewed with or without contours overlaid (figure 4b, below right).

MapSlicer also nows retains the user settings for contour levels, view orientation and other parameters between runs of the program, making it easier to customise the program for frequent use.

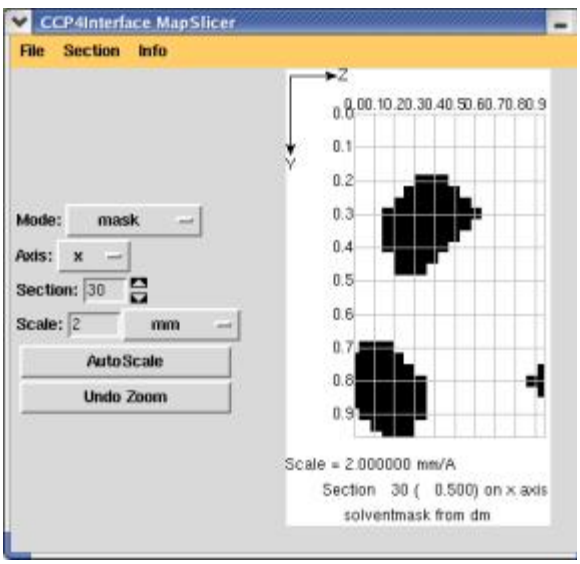


Figure 4a: Mapslicer operating in "mask" mode.

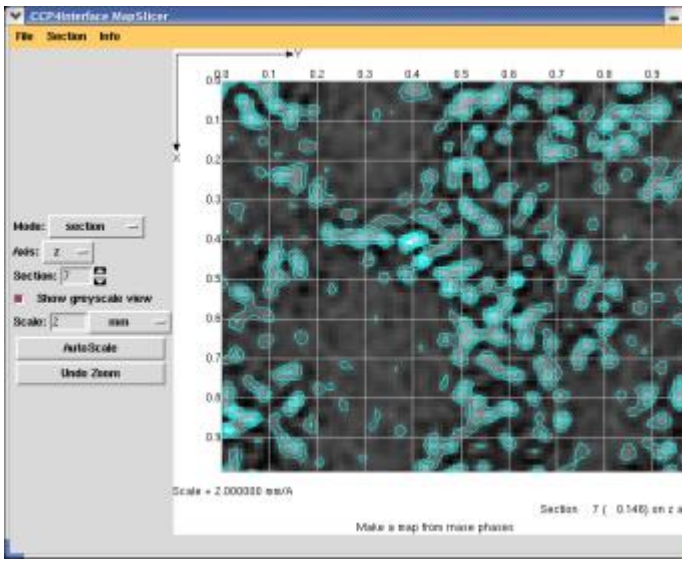


Figure 4b: Greyscale display with contours overlaid in MapSlicer.

The **Run remote job** function can now use either ssh or rsh to run jobs on remote machines - this is significant as the use of rsh is increasingly being deprecated by computer managers.

Other minor improvements

In addition to a large number of bug fixes, we have attempted to address some of the minor problems reported by users of the interface. These include:

- No longer presenting the option to output O and/or QUANTA format maps, if the underlying software is not installed on the user's path - only the available map formats will be listed.
- Giving the option to step back to the previous frame when viewing a logfile that has been split into multiple frames. This makes it easier to navigate long logfiles.
- When browsing for files on Windows platforms, CCP4i will now also allow browsing of the different drives such as d: or e:.
- Harvest files are now correctly associated with the job which generated them. This will make it easier to benefit from the data harvesting functionality when using CCP4i.
- Improvements to accessing the online help from the main CCP4i window - clicking on the help button now brings up a menu with shortcuts to different "topics" (see figure 5, right).

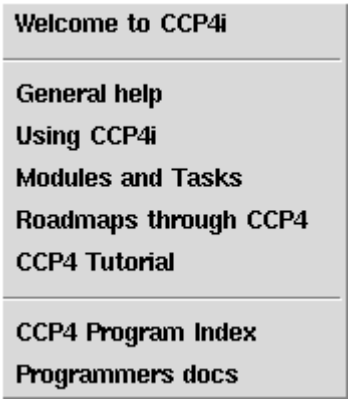


Figure 5: Help topics accessible from main CCP4i help

Improvements for Developers

There have been a number of improvements for developers who are working with CCP4i:

- **FindExecutable**: will return the full path of the specified program executable - essentially this acts like a Unix which command for CCP4i.
- **CreateLabinLine4**: this is an extended version of the CreateLabinLine command, which creates a group of up to four "coupled" MTZ column label selection menus in a single command, for example to deal with H-L coefficients or F(+) and F(-)-style groupings.
- **Export tasks**: this interface has been improved to make it more usable - it is possible to specify a "basename" for a task interface (rather than manually selecting each file for a task), and to save and then restore the parameters used in exporting a task (cutting down on tediously set up each time).
- **GroupMtzCols**, **MtzColSameDataset**, **GetMtzColType**, **GetMtzGroupByType**: a set of new commands to handle the sorting of MTZ columns based on dataset and type information.

Some issues still remain to be addressed, particularly with regard to handling information relating to crystals and datasets in MTZ headers.There are also some ongoing issues with CCP4i operating under Windows, where dealing with file and path names which contain space characters continues to be a problem.

Acknowledgements

CRANK is being developed by Steven Ness at the University of Leiden by Steven Ness. The CRANK website can be found at http://www.bfsc.leidenuniv.nl/software/crank. Steven Ness also developed the interface to BP3.

The shelx_cde interface was written by Peter Briggs but draws heavily on the hkl2map interface of Thomas Schneider and Thomas Pape. The SHELX suite of programs are not part of CCP4 and more information (including details of how to obtain them) can be found at http://shelx.uni-ac.gwdg.de/SHELX/.

The interfaces for PHASER have been developed by Anne Baker and Airlie McCoy, with input from Peter Briggs.

Francois Remacle implemented the Database Search and Sort tool, the job display colour customisation and the project switching functionality, with input from other Daresbury programmers.

Liz Potterton contributed code to ensure compatibility with CCP4mg, including the launcher for CCP4mg.

CCP4i is maintained and developed by the Daresbury CCP4 staff (Peter Briggs, Martyn Winn, Charles Ballard, Francois Remacle, Norman Stein and Dan Rolfe) who contributed other fixes and developments. Please send questions, requests and bug reports to us at ccp4@ccp4.ac.uk.

Peter Briggs (p.j.briggs@ccp4.ac.uk)

Report on the CCP4 Workshop at ACA 2005, Orlando Florida

Peter Briggs

CCP4, Daresbury Laboratory, Warrington WA4 4AD, UK

Over the past three years CCP4 has established a tradition of holding a one-day satellite workshop as part of the annual American Crystallographic Association (ACA) meeting. This year's meeting was held at the Walt Disney Swan Hotel in Orlando, in the "Sunshine State" of Florida, and once again we were also there.

The workshop - entitled "The CCP4 Software Suite: A Protein Crystallographic Toolbox" - was held on the 28th May (the day before the full meeting) and attracted over fifty delegates with a wide range of experience both with CCP4 and with macromolecular crystallography in general. The aims were therefore to get novice users started with the suite and show them how to use some of the key programs, while at the same time trying to surprise more expert users with new or less well-known aspects of the software.

The workshop followed the same format as in previous years: introductory talks gave overviews of the software suite as a whole and prepared the way for presentations on the specific packages. In the first session **Peter Briggs** outlined many of the technical non-crystallographic aspects of the software, focusing on CCP4i and the gory details of MTZ files, followed by **Johan Turkenburg** giving a tour of the crystallographic functionality of the suite from a user perspective. He introduced a number of programs which even some more experienced users might be unfamiliar with, and also stressed the compatibility between CCP4 and other software suites such as SHELX and ARP/wARP. **Maria Turkenburg** then gave an overview of the broad range of help available within the suite and the CCP4 website.

The remaining sessions focused on practical aspects of running of some of the "flagship programs". **Harry Powell** talked in detail about data processing, integration and scaling using MOSFLM and SCALA, in particular focusing on practical aspects such as how to tell if data processing is working, and how to address warnings and problem cases. **Roberto Steiner** covered the background theory of REFMAC5 and gave a live demonstration of key features including the use of TLS parameters and the generation of restraints dictionaries.

Finally live demonstrations were given by **Stuart McNicholas** and **Paul Emsley** of the two aspects 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. The entertaining "double-headed" demonstration of Coot by Paul and Johan was particularly popular, generating

"oohs" and "aahs" from a rapt audience as they watched side chains wriggle their way into the correct density.

The feedback from the workshop was very positive, and many people reported that they would be trying out the things that they had heard about (something that was borne out by the people who visited the CCP4 exhibition stand after the workshop). There were also requests for a longer workshop with more hands-on demonstrations and more examples of real-life problems, which we will aim to address in future.

The workshop organisers **Maeri Howard** and Peter Briggs would like to thank the ACA for the opportunity to run the workshop, the speakers for their presentations and the delegates for attending. We would also like to acknowledge the help of **Ed Collins** and **Marcia Colquhoun** in setting up the workshop. Financial support for the workshop was provided by CCLRC (from CCP4 industrial income) and by the ACA.

The materials from the workshop can be found online at http://www.ccp4.ac.uk/courses/ACA2005/ACAworkshop05.html. A similar workshop will be held as part of the IUCr 2005 meeting in Florence in August, for details see http://www.ccp4.ac.uk/courses/IUCr2005/iucr05.html.

Workshop speakers (from L-to-R, back row) Peter Briggs, Roberto Steiner, Maria Turkenburg, Stuart McNicolas (front row) Johan Turkenburg, Harry Powell, Paul Emsley

Roberto gives an indication of the radius of convergence for Refmac5

Roberto and Paul field questions from a workshop participant at the end of the day

---

Peter Briggs (p.j.briggs@ccp4.ac.uk)

Diamond: Status Report on MX Beamlines and Computing

http://www.diamond.ac.uk

The Synchrotron

Diamond is a new synchrotron radiation source being built at the site of the Rutherford Appleton Laboratory in Oxfordshire. The facility will comprise a 3 GeV electron storage ring, injected from a 100 MeV linac through a full energy booster synchrotron, and an initial complement of seven beamlines. The properties of each beamline were developed following extensive consultation with user communities to deliver a facility optimised for a range of applications, including macromolecular crystallography (MX).

Table 1. Diamond at a Glance	
Electron Beam Energy	3 GeV
Circumference - Storage Ring	561.6 m
Number of cells	24 double-bend achromatic
Insertion devices straights	4 x 8 m
	18 x 5 m
Beam current	300 mA (500 mA)
Beam emittance	2.74 nm rad (horizontal)
	0.0274 nm rad (vertical)
Beam life time	>10 h (20h)

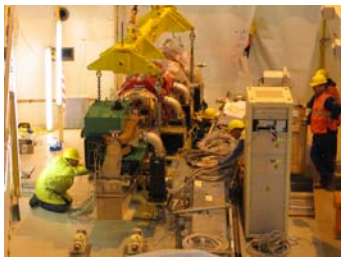


Figure 1: June 2005

Left: Installation of a storage ring girder.

Right: exterior of the diamond experimental area building, connecting bridge and office block housing Diamond staff since February 2005.

The MX Beamlines

The first set of seven beamlines will go into operation in January 2007 and will include three macromolecular crystallography beamlines designated I02, I03 and I04. A microfocus beamline and a fixed wavelength side-station are planned for a later stage. The first three beamlines will receive radiation from in-vacuum undulators. The optical hutches for each beamline will contain a fixed exit Si111 double crystal monochromator, with the first crystal indirectly cryogenically cooled and the second thermally linked to the first to ensure matching of the crystal planes. Two bimorph mirrors will allow independent horizontal and vertical focusing. In addition, prior to each principal component are a set of slits and at least one diagnostic. A schematic of the optical elements is shown in Figure 2. The hutches for each of the beamlines are now complete. The assembly work has started on the optical components, and their installation is due to be completed by the end of 2005.

Table 2. Technical details for beamlines I02, I03 & I04	
Rapidly tuneable	0.5 – 2.5 Å
High flux	10^{12} ph/s in 100 μm x 100 μm at 1 Å
Small beam size	10 – 200 μm (FWHM)
Low beam divergence	< 100 μm (V) x < 50 μm (H) at 1 Å
Detector	ADSC Quantum 315
Robotic sample changer	Rigaku/MSC ACTOR

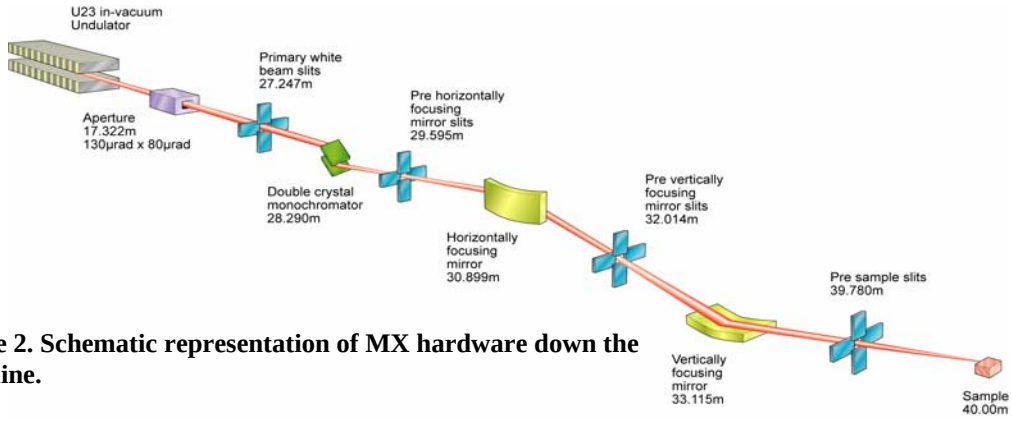


Figure 2. Schematic representation of MX hardware down the beamline.

The beamlines will all be tunable over a wide wavelength range to enable Multiwavelength Anomalous Diffraction (MAD) experiments to be carried out. Each beamline will be optimised for operation around 1 Å as it is anticipated that a significant amount of work will be MAD experiments at the Se K-edge at 0.979 Å. All three beamlines have very similar specifications, but I03 is unique since it will also include Category 3 biological containment in the experimental hutch.

The detailed design work for the experimental hutches is close to completion. Each hutch will contain a single-axis rotation stage, CCD detector, robotic sample changer, fluorescence detector and cryocooler. The beamlines will be capable of fully automated operation by combining robotic sample mounting with software-controlled loop (or crystal) centring and automated data collection software. However, completely manual operation of the beamline will still be possible for the traditionalists!

The robotic sample changers will be supplied by Rigaku/MSC and will operate with both flash-cooled crystals in loops and room-temperature crystals mounted in capillaries. An on-axis viewing system will give the user an "X-ray's view" of the crystal, making it easier to select a specific region of the crystal to be exposed to the X-rays. Three ADSC Quantum 315 detectors, with large active areas of 315mm x 315 mm and very fast readout times of 0.25 - 0.9s, have been ordered for the beamlines.

Facilities will also be available for sample storage and manipulation with the provision of basic equipment such as microscopes, pipettes and crystal freezing facilities.

Table 3. Timescales	
First electrons into storage ring	January 2006
First beam on the beamlines	April 2006
First beamline commissioning	April/May 2006
Commissioning with users	October/November 2006 onwards
User operations	January 2007

Further MX Beamlines

In August 2004, work commenced on the design of the microfocus macromolecular crystallography beamline I24. This beamline will produce a focused X-ray beam of less than 5 μm enabling the measurement of diffraction data from tiny, weakly diffracting crystals. The microfocus beamline is currently scheduled for completion in 2008.

A fixed wavelength (around 1 Å) side-station for "high-throughput" applications (I04.1) is due online in January 2009.

There are also plans for a beamline which will be optimized for data collections at long wavelengths. This beamline would be particularly suited for Sulphur SAD experiments. The timescales for this project are still to be determined.

Computing at Diamond.

The scientific IT needs of Diamond are met by the data acquisition and scientific computing group. The group is part of the Science Division within the structure of the Diamond Light Source. Its members engage to provide computing support for our scientific staff in the collection and analysis of data from all beamlines, and as such, perform a key role in the research emerging from Diamond.

Prior to the user arriving at the synchrotron it is hoped that data management and information capture will have occurred throughout the process undertaken to gain a crystal and stored in a Laboratory Information Management Systems (LIMS). One such LIMS being developed is the PIMS (Protein Information Management System, www.pims-lims.org). As Diamond is a partner in the PIMS project, work is already underway to ensure that relevant information about a sample, which will assist data collection, will be easily transferred and available on the beamlines.

Table 4. Diamond computing at a glance.

Hardware Control	EPICS
Data Acquisition software	GDA
MX Experiment environment	DNA
Standard Operating System	RedHat Enterprise 4
Beamline data storage	20 Tb RAID 50*
I02, I03 & I04 medium term data storage and extra computing	160 Tb and cluster computing*
Long term data storage	Atlas Data Store – perpetual*
MX image data format	imgCIF/CBF
Long term data format	NeXuS*

*Currently being proposed.

For the beamline, development of Diamond's data acquisition environment is benefiting greatly from a close collaboration with the current data acquisition group at the SRS in Daresbury. The Generic Data Acquisition (GDA, http://www.srs.ac.uk/srs/gda/gdaoverview.htm) will form an underlying, common architecture to all beamlines in Diamond and the JAVA based environment will have the ability to offer a customisable GUI or Jython scripting and enable remote viewing or monitoring and even the possibility of control if required.

An alternative interface to run and control the experiments on MX beamlines in Diamond will be provided by the automateD collectioN of datA (DNA) project (www.dna.ac.uk), a collaboration between a number of European facilities and academic sites including Diamond. The DNA software will facilitate the automated screening, ranking and collection of data from macromolecule crystals.

The detectors that will be on the first three MX beamlines will have the theoretical ability to produce in excess of 5 Tb of images every day, per beamline. But because MX data collection is not continuous it is envisaged that a data collection session will only produce in the region of 1 Tb of data though this still poses some interesting data management problems. The aim on the MX beamlines is to move the imgCIF/CBF

images from the 2Tb RAID array attached to the detectors, immediately to a 20 Tb RAID array local to each beamline. Here the images will be available for experimental steering software such as DNA with its associated information management system ISPyB (www.e-htpx.ac.uk).

In addition the images will be collated at this point with metadata of the sample and other data from the beamline (e.g. a png of the crystal in the loop, or beamline settings) into an evolving file format known as NeXuS (http://www.nexus.anl.gov/). It is intended that the NeXuS file format will again be common to all beamlines and will be the basis for extracting other formats for legacy code.

The preferred route of external access to data collected on diamond will be through a tool known as a Storage Resource Broker (SRB). The SRB has the distinct advantage of making a single, unique and persistent point of reference to the data, whilst the actual location of the data itself may move. For an MX beamline the data are likely to be removed from the beamline's 20 Tb RAID array within a week and only be available on a larger data store shared between all three MX beamlines. This larger data store will also be attached to a computing cluster which will be available to allow fast structure solution before leaving the synchrotron or even the beamline. The current aim is that the eventual home for the data collected at Diamond will be the ATLAS Petabyte data store (http://www.e-science.clrc.ac.uk/web/services/datastore).

However, by using SRB technology the location of the data will appear to be the same. The SRB currently has a number of interfaces including a programmable API and a web interface similar to a file browser. Work is also underway on the SRB to make the data available through a URI e.g. srb://data.diamond.ac.uk/MX/my/data.img.

Further Information

For more information on Diamond please visit the Diamond web site. http://www.diamond.ac.uk.

Alun Ashton[1]*, Katherine McAuley[2]*, Sara Fletcher[3]*, Jose Brandao-Neto[2], Liz Duke[2], Gwyndaf Evans[2], Ralf Flaig[2], Bill Pulford[1], Thomas Sorensen[2] and Richard Woolliscroft[1].

[1] Data acquisition and scientific computing.

[2] Macromolecular crystallography beamlines.

[3] Science Division.

* The main authors of this article. Material also kindly provided by the Diamond communications group.

diamond Diamond Light Source, Diamond House, Chilton, OX11 0DE.

Crank, Crunch2 and Bp3: A platform for rapid automated structure determination

Steven R. Ness, Irakli Sikharulidze, R.A.G de Graaff and Navraj S. Pannu

Biophysical Structural Chemistry, Leiden Institute of Chemistry, P.O. Box 9502, 2300 RA Leiden, The Netherlands

Introduction

In recent years, there has been much progress on many fronts in the field of macromolecular crystallography, some of which include automation of structure solution, novel crystallographic algorithms, and the use of massively parallel computational resources. Crank (Ness et al., 2004) is a new suite in macromolecular crystallography that combines these concepts into a self-consistent whole and attempts to enable crystallographers to solve structures faster and more efficiently and let them share their results with the world in a standardized way amenable for future data-mining. Crank has an interface based on the CCP4i (Potterton et al., 2003) toolkit, and is designed to integrate into a CCP4 (1994) based workflow to provide a truly integrated system for performing many diverse kinds of crystallographic experiments. Two standalone programs within the Crank suite are Crunch2 for substructure detection and Bp3 for substructure phasing.

Automated Structure Solution

Due to the large amount of data generated by projects in structural genomics and the increased power in the algorithms used, automated structure solution has been gaining in popularity. Programs that do this usually take merged diffraction data and attempt to perform all the steps in a crystallographic structure determination. One of the first programs to do this was SOLVE (Terwilliger & Berendzen, 1999). By combining expert knowledge, heuristics and various programs, SOLVE can often determine a full protein structure starting from intensity data.

In recent years, many other packages, such as AutoSHARP (Bricogne et al., 2002), CHART (Emsley, 1999), ELVES (Holton & Alber, 2004), BnP (Weeks et al, 2002), Shelx[C/D/E] (Schneider & Sheldrick, 2002), HKL2MAP (Pape & Schneider, 2004), PHENIX (Adams et al., 2004), ARP/wARP (Perrakis et al., 1999) and Auto-Rickshaw (Panjikar et al., 2005) have been developed that use various strategies and subprograms in the pursuit of automated structure solution.

Given good quality data, any of these automated structure solution packages can produce a fully traced protein model. However, in the case of more difficult structures, it may become necessary for the crystallographer to try a variety of different packages, each implementing a different strategy and employing a different user interface. The

Crank package provides wrappers for many different crystallographic programs, this allows the user to construct their own strategies for solving difficult structures. In addition, Crank can even provide wrappers for entire automated structure solution packages, thus allowing a user to quickly try a variety of different approaches, finding the best programs for their particular dataset.

By integrating a structure solution database within Crank, it is possible for a user to use the different substructure solutions or phase estimates between multiple Crank runs. In addition, Crank databases can also be exported in a variety of formats including mmCIF, for efficient deposition and sharing of results.

Novel Algorithms

Fairly often, new and novel crystallographic algorithms/programs are created. For example, with the use of programs such as MLPHARE (Otwinowski, 1991), SHARP (La Fortelle & Bricogne, 1997) and BP3 (Pannu & Read, 2004), along with direct methods in programs such as SHELXD (Schneider & Sheldrick, 2002), SnB (Weeks & Miller, 1999) and Crunch2 (de Graaff et al., 2001), crystallographers are solving previously insoluble crystal structures. Due to their new and often developmental nature, these programs are sometimes difficult to use effectively. By providing a common user interface for all programs and by including large amounts of on-line help, Crank speeds up this learning process, enabling users to utilize the most recent algorithms in the process of structure solution.

To further this end, Crank is bundled with several new programs in crystallography, including Crunch2 and BP3. Crunch2 is a new direct methods based program that uses the rank-reduction of Karle-Hauptmann matrices to determine phases. Whereas most of the other direct methods programs for macromolecular crystallography use triplet and quartet relationships, the algorithm used in Crunch2 implicitly utilizes higher phase relationships which are inherent in the rank reduction algorithm. Crunch2 been shown to produce high quality solutions in the case of difficult structures. BP3 is a novel program for substructure refinement and phasing. It employs a multivariate approach and has been shown to outperform the leading substructure refinement programs in some test cases (Pannu & Read, 2004), (Ness et al., 2004).

Crystallographic standards

In macromolecular crystallography, there are a wealth of different programs to help with various aspects of the process of structure solution. These programs are sometimes standalone and accept input in either their own special format or in a more standard format. Sometimes these programs are part of a larger suite like CCP4. In these suites there is often a harmonization of program input and output. In the CCP4 suite, the common input formats are an executable shell script based command script and the MTZ reflection file format. The output formats are MTZ files and logfiles.

With common and standardized input and outputs it becomes easier for the user to take the output from one program and convert into into input for a second program. CCP4i has further helped in this process by giving the user a standard GUI interface with which to build command scripts. Similar graphic interfaces also make it easier

for users to run programs. In addition, the Crank CCP4i interface provides tooltips and other hints for the user, helping them determine the best parameters for their particular project. These kind of interfaces have proven very useful and are now common in many projects, such as AutoSHARP, BnP, PHENIX, Chart, and Elves.

Grid Computation

In the last few years there have been attempts to come up with ways to enable novices to do massive parallel and distributed computational work. Cluster computing, where large numbers of commodity computers are joined together by a queuing system to provide large amounts of available computing power have been popular for the last ten years. However, these efforts have often been stymied by issues of security, reliablity, portability and ease of use. Grid computing is a new concept that combines many of the lessons learned in cluster computing into a single solution. The Grid uses XML as a common language and provides interfaces to help solve the many problems inherent in running programs over a worldwide network of computers.

Because Crank is built upon the same framework of XML and SOAP that the Grid is designed around, Crank integrates smoothly with the Grid. In fact, the entire Crank architecture can be envisioned as a subset of the Grid, with the individual subsections of Crank functioning like separate web services. This kind of architecture can allow truly global computation to take place. In fact, in the future, individual program authors could simply deploy their novel crystallographic algorithms as Grid web services, easily accessible by anyone in the world.

Crank takes all these diverse threads and combines them together in a flexible and extensible framework. Crank does not aim to replace existing programs or suites, but rather to combine them into a self-consistent whole, making it possible for users to run any combination of programs or suites in any order.

Crank Architecture

Crank is a loosely associated collection of programs that communicate via XML. In designing Crank, we held to the old UNIX maxim of "small programs, weakly interlinked", where each program does a specific task, and the programs communicate via a simple common language. Whereas most other packages communicate via language specific mechanisms, for instance Python Pickle objects, Crank instead stores all of it's data in XML. This allows programmers to write their applications in the most appropriate language for them. To this end, we have written Crank in a variety of different languages including Tcl, Python, C, C++, FORTRAN, Bourne shell, C shell and Java. By writing various parts of Crank in these different languages, we try to ensure that the external XML based datastructures we design are sufficiently portable to be used by other developers in their own projects and are suitable for future data-mining activities.

This philosophy has served us well, by weakly coupling different aspects of Crank, it is quite simple to add new programs to Crank. Programs have input and output, and sometimes they generate error messages. To add a program to Crank, one first identifies the different kinds of input to a program. There are two broad categories of input, dataset specific and program specific. Dataset specific information includes

sequence data, information about the macromolecule, and reflection data. Program specific information is simply instructions to the programs, for example the number of cycles of refinement to run. Program output can also be thought of in two broad categories, modified reflection data, and logfile information.

In adding a program to Crank, we identify all the program inputs and outputs and write small conversion programs to change the program input and output into a standard format. By putting these small wrappers around all the various programs, we effectively turn all programs into standardized building blocks, out of which any arbitrary crystallographic experiment can be constructed.

Crank XML

Crank uses XML for all inter-program communication and also for long term database storage and archiving of information. In order to provide for the rapid addition of new programs, Crank Input XML data structures are simply program keywords, transliterated into XML. For example, the following Crunch2 run script:

```
#!/bin/sh

crunch2    HKLIN    crank.drear    MODELIN    coordinaten.xyz    HITS
crunch2.out.hits << END
TRY 1 10
NCYC 400
ICOO 1
CELL 17.8280 31.4450 44.0110 90.0000 90.0000 90.0000
SYMM 19
NATOM 10
SCATT 15
END
```

Was generated from the following XML files:

Crank Input XML

```
<crank>
  <soap>
    <run>
      <job id="2">Crunch2
        <name>Crunch2</name>
        <tag>2_CRUNCH2</tag>
        <input>
           <coords></coords>
           <evalue_columns>
              <fa>1_AFRO_FA</fa>
              <sigfa>1_AFRO_SIGFA</sigfa>
              <e>1_AFRO_E</e>
              <sige>1_AFRO_SIGE</sige>
           </evalue_columns>
        </input>
        <output>
           <coords>2_CRUNCH2.coords</coords>
        </output>
        <crunch2>
           <pmf>1</pmf>
```

```
            <scattering_power>15</scattering_power>
            <max_resolution></max_resolution>
            <min_atoms>3</min_atoms>
            <ntrials>10</ntrials>
            <ncycles>400</ncycles>
            <pmf>
              <npatt>1</npatt>
              <max_resolution>4.0</max_resolution>
           </pmf>
         </crunch2>
       </job>
     </run>
  </soap>
</crank>
```

Crank MTZ XML

```
<crank>
  <dataset_info>
    <cell>
      <cell_a>17.8280</cell_a>
      <cell_b>31.4450</cell_b>
      <cell_c>44.0110</cell_c>
      <cell_alpha>90.0000</cell_alpha>
      <cell_beta>90.0000</cell_beta>
      <cell_gamma>90.0000</cell_gamma>
    </cell>
    <spacegroup>
      <number>19</number>
      <lattice>P</lattice>
      <operator id="0">21</operator>
      <operator id="1">21</operator>
      <operator id="2">21</operator>
    </spacegroup>
    <n_symops>4</n_symops>
    <symmetry_operator id="0">
      <aa>1</aa>                  <ab>0</ab>                  <ac>0</ac>
<atrans>0.0000000</atrans>
      <ba>0</ba>                  <bb>1</bb>                  <bc>0</bc>
<btrans>0.0000000</btrans>
      <ca>0</ca>                  <cb>0</cb>                  <cc>1</cc>
<ctrans>0.0000000</ctrans>
    </symmetry_operator>
    <symmetry_operator id="1">
      <aa>-1</aa>           <ab>0</ab>                      <ac>0</ac>
<atrans>0.5000000</atrans>
      <ba>0</ba>                      <bb>-1</bb>           <bc>0</bc>
<btrans>0.0000000</btrans>
      <ca>0</ca>                  <cb>0</cb>                  <cc>1</cc>
<ctrans>0.5000000</ctrans>
    </symmetry_operator>
    <symmetry_operator id="2">
      <aa>1</aa>                  <ab>0</ab>                  <ac>0</ac>
<atrans>0.5000000</atrans>
      <ba>0</ba>                      <bb>-1</bb>           <bc>0</bc>
<btrans>0.5000000</btrans>
      <ca>0</ca>                 <cb>0</cb>                 <cc>-1</cc>
<ctrans>0.0000000</ctrans>
    </symmetry_operator>
    <symmetry_operator id="3">
      <aa>-1</aa>           <ab>0</ab>                      <ac>0</ac>
<atrans>0.0000000</atrans>
```

```
      <ba>0</ba>                  <bb>1</bb>                  <bc>0</bc>
<btrans>0.5000000</btrans>
      <ca>0</ca>                 <cb>0</cb>                 <cc>-1</cc>
<ctrans>0.5000000</ctrans>
    </symmetry_operator>
  </dataset_info>
</crank>
```

Because there are so many different possible types of XML output, Crank does not try to impose it's own standards on the XML output by programs, but rather maintains a database of program output and how to convert program output to standard Crank Output XML. In the case of a program that already has XML output, XSLT (Extensible Stylesheet Language Transformations) documents are used to convert program output to Crank XML. In the case of programs without XML output, program logfiles are converted to XML by small utility programs. After transforming the program output into standard Crank XML, Crank can then access and use this program output in subsequent "decision" steps in the Crank pipeline.

Designing an experiment in Crank

Most of the other automated pipelines in crystallography use a combination of expert knowledge and heuristics in a pipeline. This expert knowledge is often of very high quality, as in the SOLVE, BnP, autoSHARP, Elves and CHART pipelines. However, as the plurality of pipelines in existence suggests, there are different decisions possible at each stage, and each pipeline will often choose a different strategy. In Crank, we instead create a toolbox that allows a scientist to design their own custom pipeline, with different programs run at each step and multiple paths that can be chosen depending on program output from a previous step. In addition, by utilizing the parallel nature of the Grid, multiple programs can be run at once, and subsequent decisions can be made based on the best available knowledge at the time. This allows Crank to be both flexible and fast.

There are many pre-programmed strategies built into Crank for solving structures, and in addition, a user can build their own custom strategies.

To build a strategy, first start the Crank CCP4i interface. You are presented with an empty canvas, with only a section for protein and dataset information. Different programs can be chosen from the drop down menu by selecting a program and then pressing "Add Program or Decision". After pressing this button, the selected program or decision is added to the Crank canvas. The user can then go in and edit the default parameters for the program being run. This process is repeated until a full experiment is built. An example pipeline for a MAD experiment could be:

Step	Program	Description
0	SCALEIT (Evans, P.R., Dodson, E.J. & Dodson, R., unpublished)	Relative scaling of datasets.
1	AFRO (Pannu, in preparation)	Calcuate E-values
2	Crunch2 (de Graaff et al., 2001)	Determine heavy atom positions

3	
BP3 (Pannu et al., 2003)	Refine heavy atom positions and output phases
4	
SOLOMON (Abrahams & Leslie, 1996)	Density modification
5	
DM (Cowtan, 1994)	Density modification
6	
RESOLVE (Terwilliger, 2003)	Model building
7	
REFMAC (Murshudov et al., 1997)	Model refinement

The output of all these programs is stored in an experiment specific Crank Output XML database. For example, the Luzzatti parameters from the BP3 step would be stored as:

```
<job id="3">
   <program>bp3</program>
   <luzzati id="0">0.3000</luzzati>
   <luzzati id="1">0.3000</luzzati>
   <luzzati id="2">0.3000</luzzati>
</job>
```

In addition to program steps, a user can insert a decision step into an experiment. Decisions take the form of a familiar IF..THEN statement, where the variable referred to in the IF part of the statement is a value that has been stored in the Crank experiment database as the output from one of the programs that have been run.

Because of the simplicity of this approach, and its similarity to the way a scientist thinks about a real experiment, this methodology can be quite a powerful way to solve a structure.

In addition, this approach works quite well with the Grid model of computation, in that multiple experimental pipelines can be run in parallel, with decision steps able to take the current best solution. For example, multiple SHELXD jobs can be started with different resolution limits and as these jobs complete, subsequent programs that need substructure information can take the highest scoring solution as their starting point.

Test cases - Fast phasing

As well as improving the the convergence radius of structure solution, developers have also attempted to obtain a solution quickly. Crank allows the ability to test and implement different protocols, one of which we present below. In order to obtain an answer from Bp3 quickly, we can perform a limited number of cycles of refinement of

error and atomic parameters and output the corresponding best phase and phase probability distribution. This procedure in Bp3 is accomplished with the "PHASe" keyword. We report tests of this compared with the normal/default procedure of BP3. Table 1 shows the details of the data sets used.

Table 1

Molecule	**Experiment**	**Substructure**	**f'' (approx)**	**Reference**
C. acidurici ferredoxin	SAD	8 Fe	1.25	(Dauter et al., 1997; 2002)
Carbohydrate binding module	SAD	4 Se	5.4	(Boraston et al., 2003; Dodson, 2003)
DNA oligomer (CGCGCG)2	SAD	10 P	0.434	(Dauter & Adamiak 2001; Dauter et al., 2002)
Human acyl-protein thioesterase	SAD	22 Br	5	(Devedjiev et al., 2000; Dauter et al., 2002)
E. coli thioesterase II	SAD	8 Se	5.4	(Li et al., 2000; Dauter et al., 2002)
Lysozyme (high redundancy)	SAD	10 S 8 Cl	0.56	(Dauter et al., 1999; Dauter et al., 2002)
Pseudomonas serine carboxyl proteinase	SAD	9 Br	5	(Dauter et al., 2001; Dauter et al., 2002)
Calcium subtilisin	SAD	3 Ca	1.28	(Betzel et al., 1988; Dauter et al., 2002)
MutS binding to G-T mismatch	SAD	45 Se	5	(Lamers, Perrakis et al., 2000)
Lysozyme (low redundancy)	SAD	10 S 2 Cl	0.56	(Weiss, 2001)

To perform the phasing in Bp3, the substructure that was input was determined with Crunch2, using DREAR (Blessing & Smith, 1999) for FA value calculation. Results from the fast phasing (PHASe protocol) and default protocol are shown in Table 2.

Table 2 : Statistics for substructure refinement and phasing in BP3 using two different protocols.

Molecule	**PHASe protocol**			**Default protocol**		
	Figure of merit	**cos(Phase error)**	**Time Min:Sec**	**Figure of merit**	**cos(Phase error)**	**Time Min:Sec**
C. acidurici ferredoxin	0.39	0.53	1:30	0.56	0.54	9:52
Carbohydrate	0.24	0.20	1:02	0.25	0.19	6:31

binding module						
DNA oligomer (CGCGCG)2	0.43	0.52	0:10	0.57	0.54	1:14
Human acyl-protein thioesterase	0.34	0.32	1:18	0.36	0.33	22:50
Lysozyme (high redundancy)	0.42	0.43	1:39	0.48	0.44	14:22
E. coli thioesterase II	0.45	0.36	1:44	0.48	0.36	12:12
Pseudomonas serine carboxyl proteinase	0.30	0.12	1:07	0.26	0.12	30:34
Calcium subtilisin	0.22	0.31	0:23	0.33	0.31	3:04
MutS binding to G-T mismatch	0.43	0.36	7:05	0.48	0.39	204:52
Lysozyme (low redundancy)	0.33	0.33	0:39	0.40	0.37	11:06

From Table 2, the general trend is that the PHASe protocol can produce solutions of similar quality, significantly faster than the default protocol. However, the default protocol produces better phase estimates (as judged by a higher cosine of the phase error) and more accurate phase probability statistics (as judged by the agreement between the figure of merit and the cosine of the phase error). These small differences may be important for structure solution with smaller signals and worse phase quality. However, for large signals, the PHASe protocol may be sufficient to generate a completely built model efficiently. A thorough analysis will be performed in the future for obtaining the most complete model automatically and efficiently.

Conclusion

By combining existing concepts into a single suite, Crank allows crystallographers to use the most advanced programs with a common user interface and allows these programs to run efficiently on a world-wide cluster of computers. This enables the crystallographer to try a variety of different hypotheses in a time efficient manner, speeding up the process of structure determination. By exporting the output from various programs into a common format, it helps the user to deposit and share their data more efficiently. In the future, this self-consistent data can be data-mined by the scientific community, adding value to the already valuable information produced by the crystallographic community.

Acknowledgements and Program Availability

We would like to thank the people who contributed datasets and the users who have downloaded the pre-beta version of crank, crunch2 and bp3. Version 0.8 of crank is available from web site http://www.bfsc.leidenuniv.nl/software/crank/ and will be

available via CCP4 and is suitable for SAD or SIRAS experiments. Version 1.0 of crank, with support for MAD phasing, will also be available from our web site and via CCP4 in the future. Funding for this research was provided by the Netherlands Organization for Scientific Research (http://www.nwo.nl).

References

Abrahams, J. P. & Leslie A. G. W. (1996). Acta Cryst. D52, 30-42.

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

Betzel, C., Dauter, Z., Dauter, M., Ingelman, M., Papendorf, G., Wilson, K.S. & Branner, S. (1988). J. Mol. Biol. 204, 803-804.

Blessing, R.H. & Smith, G.D. (1999). J. Appl. Cryst. 32, 664-670.

Boraston, A. B., Revett, T. J, Boraston, C.M, Nurizzo D. & Davies, G.J. (2003). Structure 11, 665-675.

Bricogne, G., Vonrhein C., Paciorek W., Flensburg C., Schiltz M., Blanc E., Roversi P., Morris R. & Evans G. (2002). Acta Cryst. A58, C239.

Collaborative Computational Project, Number 4. (1994). Acta Cryst. D50, 760-763

Cowtan, K. (1994). Joint CCP4 and ESF-EACBM Newsletter on Protein Crystallography, 31, p34-38.

Dauter, Z., Wilson, K.S., Sieker, L. C., Meyer, J. & Moulis, J.M. (1997). Biochemistry, 36, 16065-16073.

Dauter, Z., Dauter, M., de La Fortelle, E., Bricogne, G. & Sheldrick, G. M. (1999). J. Mol. Biol. 289, 83-92.

Dauter, Z., and Adamiak, D. A. (2001). Acta Cryst. D57, 990-995.

Dauter, Z., Li, M. and Wlodawer, A. (2001). Acta Cryst. D57, 239-249.

Dauter, Z., Dauter, M. & Dodson, E.6 (2002). Acta. Cryst. D58, 494-50.

Devedjiev, Y., Dauter, Z., Kuznetsov, S.R., Jones, T. L. Z. & Derewenda, Z. S. (2000). Structure Fold Des., 8(11), 1137-1465.

Dodson, E. (2003). Acta. Cryst. D59, 1958-1965.

Emsley, P. (1999). CCP4 Newsletter on Protein Crystallography. 36.

de Graaff, R. A. G., Hilge, M., van der Plas, J. L. & Abrahams, J. P. (2001). Acta Cryst. D57, 1857-1862.

La Fortelle, E. de & Bricogne G. (1997). Methods Enzymol. 276, 472-494.

Lamers, M. H., Perrakis, A., Enzlin, J. H., Winterwerp H. H. K, de Wind, N. & Sixma, T. (2000). Nature 407, 711-717.

Li, J., Derewenda, U., Dauter, Z., Smith, S. & Derewenda, Z. S. (2000). Nature Struct Biol 7, 555-559.

Holton, J. M. & Alber, T. (2004). Proc. Natl. Acad. Sci. (USA) 101, 1537-1542.

Murshudov, G.N., Vagin, A.A. & Dodson, E.J. (1997). Acta Cryst. D53, 240-255.

Ness S.R., de Graaff, R.A.G., Abrahams, J.P. & Pannu, N.S. (2004) Structure. 12, 1753-1761.

Otwinowski, Z. (1991) Proceedings of the CCP4 Study Weekend. Isomorphous Scattering and Anomalous Replacement. Edited by Wolf, W., Evans, P.R., and Leslie, A.G.W., pp 80-86. Warrington: Daresbury Laboratory.

Panjikar, S., Parthasarathy, V., Lamzin, V. S., Weiss, M. S. & Tucker, P. A. (2005). Acta Cryst. D61, 449-457

Pannu, N.S. & Read, R.J. (2004) Acta Cryst. D60, 22-27.

Pannu, N. S., McCoy A. J. & Read, R. J. (2003). Acta Cryst. D59, 1801-1808.

Pape, T. & Schneider T.R. (2004). J. Appl. Cryst. 37, 843-844.

Potterton, E., Briggs, P., Turkenburg, M. & Dodson, E.J. (2003). Acta Cryst. D59, 1131-1137.

Perrakis, A., Morris, R. J. & Lamzin, V. S. (1999). Nature Struct. Biol. 6, 458-463.

Schneider, T.R. & Sheldrick, G.M. (2002). Acta Cryst. D58, 1772-1779.

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

Terwilliger, T.C. (2003). Acta Cryst. D59, 1174-1182.

Weeks, C.M. & Miller R. (1999). J. Appl. Cryst. 32, 120-124.

Weeks, C.M., Blessing, R.H., Miller, R., Mungee, R., Potter, S.A., Rappleye, J., Smith, G.D., Xu, H. & Furey, W. (2002). Z. Kristallogr. 217, 686-693.

Weiss, M.S. (2001). J. Appl. Cryst. 34, 130-135.

CHOOCH – automatic analysis of fluorescence scans and determination of optimal X-ray wavelengths for MAD and SAD

Gwyndaf Evans

Diamond Light Source
Rutherford Appleton Laboratory
Chilton OX11 ODE
Gwyndaf.Evans@diamond.ac.uk

1 INTRODUCTION

The two dominant approaches to *de novo* structure determination in macromolecular crystallography (MX) are molecular replacement (MR) and heavy atom phasing related methods. Within the latter approach anomalous scattering from heavy atoms plays a key role in generating phase information. This information is sometimes supplementary to isomorphous phasing signal, as in the MIRAS or SIRAS methods, or is the unique source of information in the case of MAD and SAD.

MAD or SAD experiments are usually, although not exclusively, performed at or near absorption edges of the heavy atom bound to the undetermined structure. Typically the form of the absorption edge and the X-ray energy at which it occurs are not well defined due to the effects of the local environment of the heavy atom on the XANES (X-ray Absorption Near Edge Structure) and it is not sufficient to rely on tabulated theoretical values of absorption or anomalous scattering factors near to an absorption edge[1]. Furthermore the X-ray energy of MX beamlines is not always well understood and almost certainly not on an absolute scale[2].

For these reasons it is essential is almost all cases to measure the XANES directly from the heavy atoms in the protein crystal in order to permit determination of values for the heavy atom anomalous scattering factors f• and f• as a function of energy which in turn provide

- the X-ray energies at the f• maximum and f• minimum of the spectra which allow us to perform the optimum MAD or SAD experiment
- values of f• and f• at these positions to use as starting values in heavy atom determination, refinement and phasing.

2 ANOMALOUS SCATTERING AND ABSORPTION

The real (f•) and imaginary (f•) components of an atom's anomalous scattering factor are related to the absorption coefficient of an atom by the optical theorem[3]

$$f''(E) = \frac{mc\varepsilon_0 E\mu_a}{e^2\hbar} \qquad (1)$$

[1] D. T. Cromer and D. Liberman. *J. Chem. Phys.*, 53:1891–1898, 1970.
[2] G. Evans and R. F. Pettifer. *Rev. Sci. Instrum.* **67**(10) 3428 – 3433, 1996.
[3] R. W. James. *The Optical Principles of the Diffraction of X-rays.* G. Bell and sons Ltd, London, 1969.

and the Kramers-Kronig transformation

$$f'(E_0) = \frac{2}{\pi} \int \frac{E f''(E)}{E_0^2 - E^2} dE \qquad (2)$$

where the integral is taken in the upper half plane. Using these expressions it is therefore possible to determine f• and f• directly from knowledge of the absorption coefficient as a function of energy. The practical difficulties in measuring the absorption coefficient of heavy atoms embedded within many other protein atoms forces us to look to measurement of X-ray fluorescence.

When an X-ray photon is absorbed by an atom a bound electron is excited to higher energy levels or ejected from the atom with a given energy. The core-hole left in the atom is subsequently filled by an electron from a higher level. The lost energy is use d to produce a fluorescent photon of characteristic energy. Fluorescence is only one result of this lost energy (Auger electrons being another) and the probability of the generation of a fluorescence photon at an absorption edge is known as the fluorescence yield of that edge for a given element . The absorption coefficient of an atom is thus related to the fluorescence by a constant factor, the fluorescence yield, allowing the determination of a proportionally correct form of an absorption edge by measuring fluorescence as a function of energy.

The standard approach to performing MAD or SAD experiments therefore is to first measure an X-ray fluorescence spectrum from the crystal sample across the heavy atom absorption edge to provide the necessary information for finding f• and f• and in turn the appropriate wavelength for measuring anomalous diffraction data.

3 ANALYSIS OF FLUORESCENCE WITH CHOOCH

Because the fluorescence spectrum is recorded on an arbitrary scale it is necessary to normalise it to some known values of absorption or f•. The deviation of f• away from theoretical values is only observed near the edge and it is therefore possible to use values away from the edge to carry out this normalisation provided, that is, sufficient experimental fluorescence data has been measured away from the edge. By this method an f• spectrum is obtained from the measured fluorescence data.

Determination of f' requires the numerical integration of equation (2). Hoyt, de Fontaine and Warburton[4] derived an approximate expression for equation (2) which is open to numerical evaluation and CHOOCH uses this to obtain f•. The prerequisites for the numerical integration are the 1st, 2nd and 3rd order derivative of the f• spectrum and these are determined by spline analysis after removal of high frequency noise using a Savitsky-Golay filter. The noise filtering is based on knowledge of the beamline energy resolution so that some distinction between real fluctuations in the signal and noise components can be made under the assumption that high frequency fluctuations in the spectrum, which should otherwise be smoothed out by the beamline resolution, must indeed be measurement noise.

[4] J. J. Hoyt, D. de Fontaine, and W. K. Warburton J. Appl. Cryst., 17:344–351, 1984.

4 ORGANISATION OF THE PROGRAM

The steps performed by CHOOCH can be summarised as follows

1. Data input and checking
   - Fluorescence data is read from a file and basic sanity checks are performed on the data. The program attempts to guess which edge has been measured for a specified element by assuming that the middle of the scanned energy range is near the absorption edge of interest.
2. Normalization of input spectrum
   - Normalization is performed as described above. A linear model is used to perform the fitting.
3. Determination of f•
   - Theoretical values of f• are obtained using the mucal.c[5] routine written by Pathikrit Bandyopadhyay which uses the absorption cross-section values as published by McMasters[6].
4. Smoothing and calculation of derivatives.
   - Smoothing is done with a Savitsky-Golay filter using a window width which is determined from the monochromator energy resolution. The resolution may be supplied by the user with the '-r <*resol*>' option
5. Kramers-Kronig transformation to obtain f•
   - The program uses numerical integration routines supplied with the Gnu Scientific Library[7] to perform the K-K transformation.
6. Analysis and output of results
   - The program automatically selects the peak f• energy and the minimum f' energy and outputs them. A PostScript plot of the f• and f• spectrum is generated if requested by the user with the '-p <psfile>' option (see Figure 1).
   -

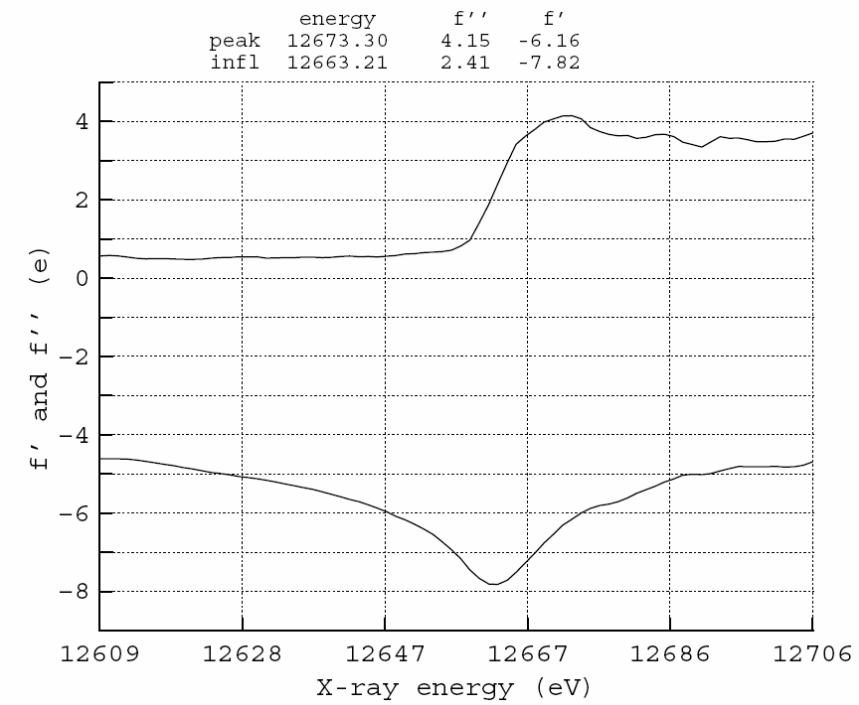


Figure 1 Example of a PostScript output from CHOOCH requested using the -p option

[5] http://ixs.csrri.iit.edu/database/programs/mcmaster.html
[6] W. H. McMasters, D. N. K. Grande, J. H. Mallet, and J. H. Hubbell *Compilation of X-ray cross sections*. Technical Report UCRL-50174, Lawrence Radiation Laboratory (Livermore), 1969.
[7] http://www.gnu.org/software/gsl/

The original versions of CHOOCH (versions 1 to 4) were written in Fortran 77 and required manual intervention from the user at the stages of normalization and fitting[8]. Although this program proved very useful at many MX beamlines worldwide, the growing need for automation placed an emphasis on the requirement for CHOOCH to operate without any user intervention.

The new version 5 of CHOOCH (now to be distributed with CCP4 version 6) has been rewritten incorporating more robust fitting and smoothing algorithm and a carefully selected set of default parameters permitting fully automated execution with minimal input of the element name and the absorption edge being probed (default is the Se K edge). The main improvements to CHOOCH have been

- Better checking of input files for machine and human errors
- Automatic edge detection provided the correct element symbol is input
- Robust fitting algorithms for normalization and better handling of data where no information away from the absorption edge has been recorded.
- Warning messages to highlight potential problems with data
- Verbosity levels for efficient debugging and feedback
- Use of sensible defaults for normalization fitting ranges and smoothing parameters
- User override of default parameters
- Parameter input via command line switches
- Use of robust Savitsky-Golay filtering methods for noise filtering
- Generation of publication quality PostScript output

This version is already addressing the automation needs of beamlines at the APS, ESRF, SSRL and SPRING-8 to name a few and it is hoped that by distributing it via CCP4 many more crystallographers and beamline users will be able to benefit from the software.

4.1 Usage

The command line use of CHOOCH 5 provides the user with several options allowing default parameters and filenames to be overridden. The CHOOCH syntax is

```
chooch -e <element> [options] <input_filename>
```

The following options are available

```
-h                print this message
-s                run silently
-e <element>            element symbol (default Se)
-a <edge>         absorption edge (K, L1, L2, L3, M) (default is auto detect)
-r <resol>        energy resolution (dE/E) (default is Si(111) 1.4x10-4)
-1 <e1>           Below edge fit lower energy limit (eV)
-2 <e2>           Below edge fit upper energy limit (eV)
-3 <e3>           Above edge fit lower energy limit (eV)
-4 <e4>           Above edge fit upper energy limit (eV)
-p <PS_file>            output to PostScript file
-o <efs_file>           filename for efs output (default output.efs
-v <level>        verbosity level (0 -- 3) (default 0)
-w                show warranty information
-c                show redistribution information
-l                show license information
```

This structure permits CHOOCH to be rapidly integrated into beamline control systems providing beamline users and operators with quick feedback and guidance about their heavy atom absorption edges.

[8] G. Evans and R. F. Pettifer *J. Appl. Cryst*. **34**, 82 – 86, 2001.

5 OBTAINING CHOOCH

CHOOCH can be obtained directly from the author by sending a request to gwyndaf@gwyndafevans.co.uk or by downloading the program from the CCP4 distribution sites of the version 6 release. It is distributed under the terms of the Gnu General Public License[9]. CHOOCH makes use of the following external routines

- Gnu Scientific Library[10] version 1.1 or later.
- Cgraph version 2.04 PostScript plotting library[11].
- (optionally) PGPLOT graphics library[12].

The option of using the PGPLOT library gives the user the ability to visualize the intermediate steps performed by CHOOCH but is most useful as a debugging tool.

6 ACKNOWLEDGEMENTS

The author thanks the contributors to the Gnu Scientific Library and R. Freeman for authoring the Cgraph PostScript plotting library used in CHOOCH. Thanks also to Robert Pettifer who contributed to CHOOCH in its infancy.

[9] http://www.gnu.org/copyleft/gpl.html
[10] http://www.gnu.org/software/gsl/gsl.html
[11] http://neurovision.berkeley.edu/software/A Cgraph.html
[12] http://www.astro.caltech.edu/ tjp/pgplot

Coot News

Bernhard Lohkamp, Paul Emsley, Kevin Cowtan

11 July 2005

Coot[1] is a molecular graphics application for electron density-based building, with a particular emphasis on protein model-building. We describe here recent additions to the software.

1 SHELX support

Over recent months[2] Shelx support has been added to Coot. SHELX can output its data in "old style" STAR CIF format. This presented some difficulties because the mmCIF parser built into mmdb[3] (and used by Coot) doesn't parse this type of cif data.

Therefore, it is necessary to convert SHELX cif data [LIST 6] `.fcf` file to an mmCIF-compatible format. Coot detects from the file-name that it needs conversion. This is currently done by using the scripting language to write out an awk file which is excuted using input from the the SHELX `.fcf` file and output to an mmCIF data file. The output file-name is the same as the input file-name with the addition of `".cif"`. This new file can be read directly into Coot and can be used on subsequent occassions obviating the need for the conversion script.

The SHELX coordinates `.ins` file presented more of a problem. Each record is now compared to an extensive (but not officially complete) list of SHELX keywords and only when there is no match is this record considered an atom description. Most of the non-atom keywords are copied without interpretation but some are interpreted, such as the LATT, CELL, SYMM, FVAR cards. In future therefore, it will be possible to modify a SHELX model with operations such as water addition and side-chain re-modelling, modelling of alternate conformations, and creation of an `.ins` file without human intervention from which a new SHELX refinement can be started automatically.

As has been mentioned, Coot has been designed particularly with proteins in mind. As such, Coot currently does not draw bonds between the main molecule and symmetry related copies. However, this may be an inpropriate restriction for some (particularly inorganic) SHELX models with inversion centres.

2 Flexible Ligands

Ligands in protein binding sites are often not in their minimum energy configuration. Thus, when searching for ligands in electron density Coot is more likely to find the correct position and orientation if a selection of conformations is searched. To modify the conformation, Coot uses the ligand geometry torsion description in the mmCIF-based restraints file that one would use for REFMAC refinement.

In the restraints file each rotatable bond is independently described by an initial torsion angle, a standard deviation and a period, and from these a probability distribution can be

[1] Emsley & Cowtan (2004) *Acta Cryst. D* **60**: 2126-2132 Part 12 Sp. Iss. 1 DEC 2004.

[2] and with the advice of George Sheldrick, Gábor Bunkóczi and Judit Debreczeni.

[3] Krissinel E.B. *et al.* (2004) Acta Cryst. D **60**: 2250-2255 Part 12 Sp. Iss. 1 DEC 2004.

constructed. However this is a naïve description - the torsion angles are generally not independent.

For each rotatable bond Coot makes a random selection of a torsion angle from the probability distribution. A model constructed in this manner can therefore lead to high-energy (and therefore unlikely) conformations. Therefore, an extra step has been added to the generation of ligand conformers which does "energy" minimization using non-bonded distance restraints in the target function[4]. However, although this makes the model more realistic it considerably reduces the speed at which new conformers are generated.

3 Difference Map Variance Analysis

The difference map variance analysis serves several purposes

- checks for sphericity of the electron density at water positions
- finds waters with the wrong temperature factor or occupancy
- (if the difference map is an anomalous difference map) finds "anomalously diffracting" water molecules. Such waters are of course candidates to be ions since water molecules are not generally known for their anomalously diffracting properties.

The difference map is sampled around each water postion at 3 different radii at 14 points on the surface of cannonical spheres of radius 0.4, 0.8 and 1.2Å.

The variance for each radius are totalled for each atom and these totals analysed for outstandingly high variance sum. A clickable list of buttons is created for each deviant water.

4 Coot for Windows

Coot has been compiled for Windows platforms. Initially the program was ported to Cygwin and later to a native port using MinGW [5]. The native build for Windows (WinCoot) has almost the full functionality as the Unix-based original. WinCoot uses the Python scripting option of Coot. Most of the Coot Guile scripts have been translated to Python to be used in WinCoot, however these can also be used within unix-based Python-enabled Coot.

Windows-based computers are used extensively and crystallographic programs are increasingly being made available for Windows platforms. Furthermore the majority of mobile computers seem to run a version of the Windows OS. Therefore there is some pressure from the community to have Coot run on Windows computers. Windows-based crystallographic model-building programs give the opportunity to the crystallographer to build structures whilst travelling, *e.g.* from collecting data at the synchrotron. Coot for Windows (WinCoot) complements the CCP4 Suite[6], which has been available for Windows for some time. This enables crystallographers to use programs from data processing to model-building on their Windows PCs.

The initial version of WinCoot was compiled with the Cygwin Unix emulation under Windows. However the program turned out to perform slowly, due to the emulation and use of `cygwin.dll`[7]. Therefore Coot was compiled natively on Windows using MinGW, this process is be described in more detail below. The Cygwin Coot port is not supported and updates are no longer available.

[4] torsion angles are not included in the target function.

[5] `http://www.mingw.org`

[6] "The CCP4 Suite: Programs for Protein Crystallography" Acta Cryst. D (1994) **50**, 760-763.

[7] rather than directly using the Windows system `msvcrt.dll`.

4.1 Compilation of Dependency Libraries

Coot requires a number of different libraries to be compiled and installed prior to compilation of Coot. All these libraries and WinCoot have been compiled on a Windows 2000 computer running MinGW 3.1.0 with Minimal SYStem (msys) 1.0.10. The GNU compiler suite version 3.2.3 was deployed for compiling and linking. The crystallographic libraries, mmdb (1.0.6), SSMlib (0.0-pre1), mccp4 (0.7.1) and clipper[8] as well as the general scientific libraries, FFTW (2.1.3) and the GNU Scientific Library (1.5) were compiled with no or very little adjustments for the WIN32 system.

Coot supports two scripting languages, guile and Python, of which the guile support is more extensive. Guile 1.6 or higher is required for Coot, however we have been unable to compile this on our Windows system. Therefore Python was chosen as the scripting language for WinCoot. Currently Python 2.3.4 is used and binaries including the developer files were obtained by download from the main Python web site, `www.python.org`. Various graphic libraries are used by Coot, namely Glib, GTk+, GTkglarea, GTk-Canvas and its dependency, imlib. Glib, GTk+ and related libraries for Windows are available for download from the developers of the GIMP package, including developer files. However the (available) libraries used are based on the Glib/GTk+-2 package and not 1.2 as required for Coot on Unix based systems. Further details and implications are described further in the next section. Currently Glib version 2.4.7 and GTk+/GDK 2.4.14 are used. (and lots of others, like pango, freetype, gmodule, gdk-pixbuf, libart, libpng). Using these libraries GTkglarea 1.2.3 compiled without any problems on MinGW. GTk-Canvas (and imlib) requires an X-Windows system, which is not available on Windows without using an emulation like cygwin. Therefore GNOME-Canvas, which readily compiles under MinGW, was employed instead of GTk-Canvas. This and the use of GTk+/GDK-2.4, which includes GDK-pixbuf, made the requirement of imlib redundant.

4.2 Compilation of Coot on Windows

Various adjustments of the Coot source code had to be made to compile it successfully on MinGW. The majority of changes are related to the different structure of the OS, Unix and Windows, as well as using updated graphics libraries and the exclusion of an X-Windows system. Some additional changes were made to implement Python scripting. Most of these are of general Coot interest and are described in futher detail.

The majority of changes were due to the different file and directory structure as well as the recognition of storage devices in the two OSs Windows and Unix. Additionally some changes were made to the use of environment variables and related issues. First `$HOME` was changed to `$COOT_HOME`, since Windows either does not use the environment variable `$HOME`, or it interferes with Cygwin. Second WinCoot was adjusted to read the correct paths for possible CCP4 installations and projects. This is necessary since the handling of users is different in Unix and Windows[9].

The use of GTk+-2 required some syntax changes to the source code. Since Coot uses some GTk+ structures which are deprecated in GTk+-2 and known to be buggy (*e.g.* GtkCList), adjustments were made to maintain functionality of the widgets. Experimentally some of the deprecated structures were translated into new functionality, but implementation is not yet fully complete. Furthermore code for the scrolling functions for changing the map contour level had to be re-written due to the different signal of the mouse scroll wheel in Windows. The change from GTk-Canvas to GNOME-Canvas was straight-forward since GNOME-Canvas is based on GTk-Canvas. All calls for a `gtk_canvas_function()` are changed to the corresponding `gnome_canvas_function()`. WinCoot was compiled with the gcc compiler option `-mms-bitfields`, which is necessary for GTk+ to function.

[8] `http://www.ysbl.york.ac.uk/~cowtan/clipper/clipper.html`

[9] designed in the days before XML, perhaps?

4.3 Windows installer and availability

WinCoot is not straight-forward to compile and for ease of use, is therefore available as compiled binaries. WinCoot can be downloaded as a self extracting exe file (see Section 5).

The WinCoot installer was built with HM NIS Edit, then compiled and compressed with NSIS (Nullsoft Scriptable Install System). It contains all necessary libraries (DLLs) including all required Python files. Therefore no additional installations or programs are required. The installer will create shortcuts to WinCoot for every user, ready to run the program. WinCoot is launched with a Windows batch file (`runwincoot.bat`) which means that no setting or changes of environment variables are necessary, thus avoiding possible conflicts with other programs and editing of Windows system files.

The current version of WinCoot corresponds to Coot version 0.0.25. Availability and integration of WinCoot into the CCP4 package for Windows is planned.

4.4 Running WinCoot

WinCoot was successfully tested on Windows XP, Windows 2000, Windows NT and Windows 98. The `runwincoot.bat` batch file sets all necessary environment variables and then executes the `coot.exe` file. It is possible to run Coot from the Windows Command Shell, Cygwin or MinGW shells, but some manual adjustments to the environment variables or location of the batch file might have to be made in that case.

On multi-user Windows PCs the batch file can be customised, so that each user or project has its own. This will allow individually associated coot setup files (`.coot.py` in `$COOT_HOME`) and backup directories (`$COOT_BACKUP_DIR`). If CCP4 for Windows is installed on the computer REFMAC5 can be run *via* WinCoot (version 0.0.31 or better) and CCP4 project directories are recognised.

Overall WinCoot has nearly the full functionality of the corresponding Coot version. However currently some minor restrictions apply due to deprecated GTk+ functions and/or missing functions in MinGW, *e.g.* filtering of files in file selection widgets. These problems have been recognised and addressed in ongoing development of WinCoot.

4.5 Python Scripting

On initiation Coot reads the `coot.py` module created by SWIG[10]. Then a personal coot setup file, `.coot.py`, is loaded if it exists in the home directory (or for WinCoot in `$COOT_HOME`). This file can (as can the corresponding guile file) contain global parameters and settings, including parameters to run REFMAC5. Then a variety of python modules are loaded from `$COOT_PYTHON_DIR`. If a python state file, `0-coot.state.py` exists in the directory from which `coot` was started, it can be loaded via a GTk dialog (*Calculate* → *Run Script...*).

Currently not all the scripting functions that have been written in guile scheme are available as Python functions. The available modules are `hello.py` (welcome notes), `gap.py` (for loop fitting), `mutate.py` (for mutation of residue range), `fitting.py` (animated protein fit of whole protein), `refmac.py` (execute REFMAC5 from within Coot) and some Coot utility functions in `coot-utils.py`. Further modules are under construction, although some use GTk functions and therefore will require an additional library in form of PyGTk.

[10] `http://www.swig.org`

5 Getting Coot

Coot is Free Software and has a variety of distribution points:

Coot Main Page (with mailing list info, FAQ, links, *etc*):
`http://www.chem.gla.ac.uk/~emsley/coot`

WinCoot:
`http://www.chem.gla.ac.uk/~bernhard/coot/wincoot.html`

Mac Coot by William Scott:
`http://www.chemistry.ucsc.edu/~wgscott/xtal/coot/`

The Phenix refinement framework

Afonine[#], P.V., Grosse-Kunstleve, R.W. & Adams, P.D.

Lawrence Berkeley National Laboratory, One Cyclotron Road, BLDG 64R0121, Berkeley, CA 94720 USA

[#]e-mail: PAfonine@lbl.gov

1: Introduction

Many questions of biological significance require highly accurate knowledge of structural parameters such as atomic positions, atomic displacement parameters (ADP, also known as "B-factors") and occupancies. The refinement of these structural parameters is therefore an essential step of macromolecular structure determination. As part of the Phenix collaboration (Adams et al., 2004) we have developed new refinement tools to increase the automation of refinement.

Macromolecular structure refinement combines a large number of very diverse steps. The current implementation of the Phenix refinement protocol is shown in Figure 1. Making use of modern software development technology, each of the major building blocks is implemented as a reusable set of modules. Most of the modules are available through the open-source cctbx libraries (Grosse-Kunstleve *et al.*, 2002; http://cctbx.sourceforge.net/) which will be included in future CCP4 releases. Some of the cctbx modules make use of CCP4 developments: the Monomer library (Vagin & Murshudov, 2004; Vagin *et al.*, 2004) and the CMTZ library.

The following sections are a brief description of the practical implementation of the Phenix refinement framework, with pointers to open-source modules that are available to the developer community. An overview of the open source libraries can be found in the series of recent IUCr Computing Commission Newsletter articles, issues 1-5 (http://www.iucr.org/iucr-top/comm/ccom/newsletters/). The pointers are given as the names of Python modules, e.g. `iotbx.pdb`.

2: Refinement framework

2.1: Input processing

To initiate refinement, four major sources of information have to be processed:

- Structural model: coordinates, displacement parameters, occupancies;
- Reflection data: pre-processed observed intensities and optionally experimental phases;
- Parameters determining the refinement protocol;
- Empirical geometry restraints (sometimes referred to a "force field"): bond lengths, bond angles, dihedral angles, chiralities, planarities (Vagin & Murshudov, 2004; Vagin *et al.*, 2004; Grosse-Kunstleve *et al.*, 2004).

The structural model and the reflection data are provided by the user. Default parameters and a library of empirical geometry restraints are provided by the refinement framework but can be customized by the user.

The PDB format (Bernstein *et al.*, 1977; Berman *et al*., 2000) is the most commonly used format for exchanging macromolecular model data and is therefore available as the input format for refinement in Phenix. The iotbx.pdb library module performs the first stage of the PDB interpretation. It is designed to construct a five-deep structural hierarchy of **models** (PDB MODEL keyword), **conformers** (PDB altLoc identifier), **chains**, **residues** and **atoms** in the most robust way. Common simple formatting problems are corrected on the fly.

The second stage of the PDB interpretation is to match the structural data against the CCP4 Monomer library in order to derive geometry restraints, scattering types and nonbonded energy types. This function is performed by the `mmtbx.monomer_library.pdb_interpretation` module. Many common simple formatting and naming problems are considered in this interpretation. The PDB interpretation has been tested with all files found in the PDB database (http://www.pdb.org/). The vast majority of files can be processed without any user intervention. Carefully designed diagnostic messages help the user to quickly identify problems that cannot be automatically corrected.

The experimental data can be given in many commonly used formats, including the MTZ format. Multiple input files can be given simultaneously, e.g. a SCALEPACK file with observed intensities, a CNS (Brünger *et al.*, 1998) file with R-free flags, and a MTZ file with phase information. A complex procedure aims to extract the data most suitable for refinement without user intervention. The underlying core functionality is implemented in the `iotbx.reflection_file_server` module.

The large set of refinement parameters is presented to the user in a novel hierarchical organization specifically designed to be extremely user friendly (Grosse-Kunstleve *et al.*, 2005). This is achieved via a very simple syntax, the option to easily override selected parameters from the command line, and automatic adjustments based on the inputs. This parameter handling framework is completely general and can be reused for other purposes unrelated to refinement.

2.2: Core refinement tools

The core refinement procedure involves four major objects: the experimental data, the model (atomic model, ordered solvent model, bulk solvent model, coordinate error model, completeness of the atomic model, scale factors), parameterization of prior knowledge (e.g. geometry restraints), and a target function combining all model parameters. Refinement is the process of optimizing the model parameters in order to obtain a model that is most consistent with the experimental data and the prior information. The measure of consistency is the value of the target function. It is designed to decrease as the model parameters improve. For a number of reasons the optimization of the target function cannot be performed in a single step. The most important problems are:

- The target function has many local minima. Therefore sophisticated search algorithms like simulated annealing may need to be applied (Brün ger *et al.*, 1987; Adams *et al.*, 1997; Brünger & Adams, 2002).
- Some groups of model parameters are highly correlated, e.g. isotropic displacement parameters and the exponential component of the overall scale factor correction, or displacement parameters and occupancies.
- Different model parameters such as coordinates and ADPs have different behavior (Agarwal, 1978).

Therefore it is common practice to perform refinement iteratively, and to split each iteration into several stages. The Phenix refinement protoc ol includes the following stages:

Bulk-solvent correction, scaling and error model estimation

Bulk solvent correction and scaling are among the most crucial steps in macromolecular structure refinement (Jiang & Brünger, 1994; Kostrewa, 1997; Badger, 1997 ; Urzhumtsev, 2000). Experience shows that best results are obtained with the Flat Bulk Solvent model (Phillips, 1980) and anisotropic scaling (Sheriff & Hendrickson, 1987; Murshudov, 1998).

Maximum likelihood target calculations require estimates of mode l errors and completeness, which in turn depend on the current atomic parameters and bulk solvent model (Lunin & Skovoroda, 1995; Afonine *et al.*, 2005). During refinement the atomic parameters and the bulk solvent model are continuously updated. Therefore it is necessary to also update the maximum likelihood error model. This requires special care since the error model is highly correlated with the bulk solvent model and the anisotropic scaling parameters. Recently we described a robust bulk solvent correct ion and anisotropic scaling procedure that combines a grid search and LBFGS minimization (Liu & Nocedal, 1989) using either Least-Squares or Maximum-Likelihood scale target functions (Afonine *et al.*, 2005). These algorithms are implemented in the `mmtbx.f_model` library module (Grosse-Kunstleve *et al.*, 2005).

Ordered solvent (water) modeling

We have implemented a completely automated protocol for updating the ordered solvent model during the refinement process (mmtbx.solvent.ordered_solvent module). If requ ested by the user, waters are updated (added and removed; Badger, 1997; Sheldrick & Schneider, 1997; Lamzin, V.S. & Wilson, K.S., 1997) in each macro cycle as indicated in Figure 1. In the same macro cycle, the complete structure including the waters is su bject to coordinate and ADP refinement. Updating the ordered solvent model involves the following steps:

1) Elimination of waters present in the initial model based on user-defined cutoff criteria on ADP, occupancy and inter-atomic distances (water-water, ma cromolecule-water).
2) Location of peaks in a mFobs-• Fcalc maximum likelihood difference map (equivalent to cross-validated $\bullet_A$-weighted map; Read, 1986; Urzhumtsev *et al.*, 1996).
3) Confirmation of peaks found in the previous step using a $2m$Fobs-• Fcalc differenc e map.
4) Elimination of peaks in regions occupied by the macromolecule. The bulk-solvent mask is reused for this purpose.

5) Elimination peaks too close to each other (the default cutoff distance is 2.0 Å; the strongest peak is retained).
6) Elimination of peaks too close to macromolecular atoms (the default cutoff distance is 1.8 Å).
7) Elimination of peaks too far away from macromolecular atoms (the default cutoff distance is 6.0 Å).
8) Elimination of peaks based on the evaluation of tabulated empirical distance distributions derived from the analysis of high-resolution models in the PDB (Fig. 2). Distance distributions between water oxygen and macromolecular C, N and O atoms are tabulated. Only peaks with a good fit to at least one distance distribution are retained.

The table of distance distributions used in the last step is located in `mmtbx.max_lik` module.

Determination of target weights

As mentioned before, crystallographic refinement is the process of model improvement through the optimization of a target function. Depending on the input parameters, the target function in Phenix is defined as $T_{xyz} = w_{x\_chem} * E_{xray} + E_{chem}$ for coordinate refinement, or $T_{adp} = w_{x\_adp} * E_{xray} + E_{adp}$ for ADP refinement. E_{chem} is a sum over six types of empirical geometry restraints as described by (Grosse-Kunstleve *et al.*, 2004). The weights w_{x_chem} or w_{x_adp} are introduced to balance the contributions from the experimental observations (E_{xray}) and the empirical a priori information (E_{chem} or E_{adp}). The automatic weight estimation procedure is implemented as described in (Brünger *et al.*, 1989; Adams *et al.*, 1997) and used by default since experience shows that it is very robust. However in a few cases it was found to produce poor results. For such cases, the more time-consuming automatic weight optimization procedure as described by Brünger (1992) is also available. The underlying core algorithms for the weight determination are implemented in the `mmtbx.dynamics.cartesian_dynamics` module.

Simulated Annealing refinement

Simulated annealing is a powerful tool for escaping local minima in crystallographic refinement (Brünger *et al.*, 1987; Adams *et al.*, 1997; Brünger & Adams, 2002). Depending on the model and data quality, simulated annealing can be performed during Phenix refinement. This is supported by the `mmtbx.dynamics.simulated_annealing` module.

Coordinate refinement

Coordinate refinement is performed by LBFGS minimization of the target T_{xyz} w.r.t. atomic coordinates, while keeping all other parameters fixed. T_{xyz} can be the Least-Squares target (LS, as defined in Afonine *et al.*, 2005), the amplitude-based Maximum-Likelihood target (ML, as defined in Afonine *et al.*, 2005) or the Phased Maximum-Likelihood target (MLHL, Pannu *et al.*, 1998). Some other target functions are available for research purposes, for example the quadratic approximation of ML (LS*, Lunin & Urzhumtsev, 1999) or LS with different types of weighting and scaling schemes. The underlying core algorithms can be found in the `cctbx.xray.target_functors` module.

ADP refinement

In the refinement of Atomic Displacement Parameters (ADP) the target T_{adp} is minimized w.r.t. isotropic *ADP*s while all other model parameters are fixed. E_{adp} is defined as:

$$E_{adp} = \sum_{i=1}^{N_{atoms}} \left[\sum_{j=1}^{M_{atoms}} \frac{1}{r_{ij}^{k}} \frac{(B_i - B_j)^2}{B_i + B_j} \right]$$

Here N_{atoms} is the total number of atoms in the model, the inner sum is extended over all M_{atoms} in the sphere of radius R around atom i, r_{ij} is a distance between two atoms i and j, B_i and B_j are the corresponding isotropic ADPs and k is user-defined constant. By default, R and k are fixed at 5.0 Å and 1.0, respectively, but they can also be refined. This "3 in 1" target function makes use of the following ideas:

- A bond is almost rigid, therefore the ADPs of bonded atoms are similar (Hirshfeld, 1976);
- ADPs of spatially close (non-bonded) atoms are similar (Schneider, 1996);
- The bond rigidity, and therefore the difference between the ADPs of bonded atoms, is related to the absolute values of the ADPs. Atoms with higher ADPs can have larger differences (Ian Tickle, CCP4 Bulletin Board, letter from March 14, 2003).

3: Conclusion

The Phenix refinement framework is a rapidly growing set of modular, reusable refinement tools, designed for future development of ever more integrated, highly automated structure determination methods. To enable collaboration among all developers, the core libraries are made available to the community as open source.

4: Acknowledgments

We gratefully acknowledge the financial support of NIH/NIGMS through grants 5P01GM063210, 5P50GM062412 and 1R01GM071939. Our work was supported in part by the US department of Energy under Contracts No. DE-AC03-76SF00098 and DE-AC02-05CH11231.

5: References

- Adams, P. D., Pannu, N. S., Read, R. J. & Brünger, A. T. (1997). *Proc. Natl. Acad. Sci*. **94**, 5018-5023.
- 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. and Terwilliger T.C. (2004). *J. Synchrotron Rad.* **11**, 53-55.
- Afonine, P.V., Grosse-Kunstleve, R.W. & Adams, P.D. (2005) . *Acta Cryst.* D**61**, 850-855.
- Agarwal, R.C. (1978). *Acta Cryst.* A34, 791-809.
- Badger, J. (1997). *Methods Enzymol*. **277**, 344-352.

- Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N. & Bourne, P.E. (2000). *Nucleic Acids Research*, **28**, 235-242.
- Bernstein, F. C., Koetzle, T. F., Williams, G. J., Meyer, E. F. Jr, Brice, M. D., Rodgers, J. R., Kennard, O., Shimanouchi, T. & Tasumi, M. (1977). *J. Mol. Biol.* **112**, 535-542.
- Brünger, A. T., Kuriyan, J., Karplus, M. (1987). *Science*. **235**, 458- 460.
- Brünger, A.T., Karplus, M. & Petsko, G.A. (1989). *Acta Cryst.* A**45**, 50-61.
- Brünger, A.T. (1992). *Nature (London),* **355**, 472-474.
- Brünger, A.T., Adams, P.D., Clore, G.M., DeLabo, W.L., Gros, P., Grosse -Kunstleve, R.W., Jiang, J.-S., Kuszewski, J., Nilges, M., Pannu, N.S., Read, R.J., Rice, L.M., Simonson, T. & Warren, G.L. (1998). *Acta Cryst.* **D54**, 905-921.
- Brünger, A. T & Adams, P. D. (2002). *Acc. Chem. Res*. **35**, 404-412.
- Collaborative Computational Project, Number 4 (1994). *Acta Cryst.* D**50**, 760-763.
- Grosse-Kunstleve, R.W., Sauter, N.K., Moriarty, N.W. & Adams, P.D. (2002). *J. Appl. Cryst.* **35**, 126-136.
- Grosse-Kunstleve, R.W., Afonine, P.V., Adams, P.D. (2004). *Newsletter of the IUCr Commission on Crystallographic Computing* , **4**, 19-36.
- Grosse-Kunstleve, R.W., Afonine, P.V., Sauter, N.K., Adams, P.D. (2005). *Newsletter of the IUCr Commission on Crystallographic Computing* , **5**, 69-91.
- Hirshfeld, F.L. (1976). *Acta Cryst.* A**32**, 239-244.
- Jiang, J.-S. & Brünger, A. T. (1994). *J. Mol. Biol*. **243**, 100-115.
- Kostrewa, D. (1997). *CCP4 Newsl*. **34**, 9-22.
- Lamzin, V.S. & Wilson, K.S. (1997). *In Methods in Enzymology*. (Carter, C. & Sweet, B. eds.) **277**, 269-305
- Liu, D.C. & Nocedal, J. (1989). *Mathematical Programming*, **45**, 503-528.
- Lunin, V.Y. & Skovoroda, T.P. (1995) . *Acta Cryst*., A**51**, 880-887.
- Lunin, V.Y., Urzhumtsev, A.G. (1999). *CCP4 Newsletter on Protein Crystallography* , **37**, 14-28.
- Murshudov, G.N., Davies, G.J., Isupov, M., Krzywda, S., Dodson, E.J. (1998). *CCP4 Newsletter on Protein Crystallography* , **35**, 37-43.
- Pannu, N. S., Murshudov, G. N., Dodson, E. J. & Read, R. J. (1998). *Acta Cryst*. **D54**, 1285-1294.
- Phillips, S. E. V. (1980). *J. Mol. Biol*. **142**, 531-554.
- Read, R.J. (1986). *Acta Cryst*. A**42**, 140-149.
- Schneider, T. (1996). *Proceedings of the CCP4 Study Weekend*. SERC Daresbury Laboratory, Daresbury, U.K., 133 -144.
- Sheldrick, G. M. & Schneider, T. R. (1997). *Methods Enzymol*. **277**, 319-343.
- Sheriff, S. & Hendrickson, W. A. (1987). Acta Cryst. A **43**, 118-121.
- Urzhumtsev, A.G. (2000). *CCP4 Newsl*. **38**, 38-49.
- Urzhumtsev, A., Skovoroda, T.P. & Lunin, V.Y. (1996). *J.Appl.Cryst*., **29**, 741-744.
- Vagin, A.A. & Murshudov, G.N. (2004). *Newsletter of the IUCr Commission on Crystallographic Computing* , **4**, 59-72.
- Vagin, A.A., Steiner, R.A., Lebedev, A.A, Potterton, L., McNicholas, S., L ong, F. & Murshudov, G.N. (2004). *Acta Cryst*. D**60**, 2184-2195.

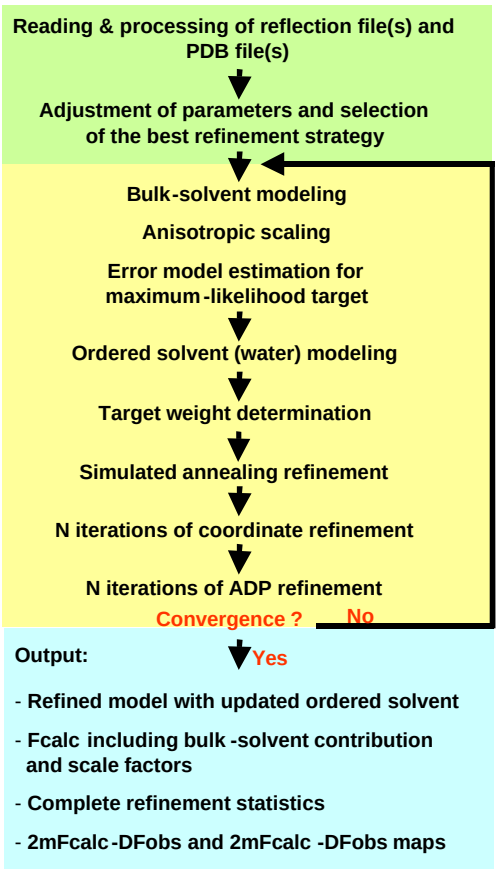


Figure 1. Phenix refinement protocol.

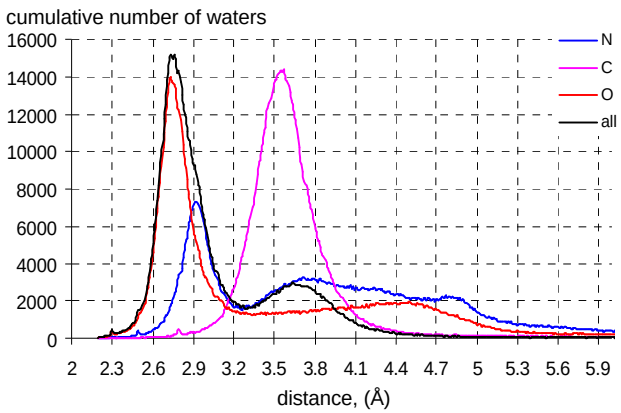


Figure 2. Statistics over high-resolution PDB models: distance distribution for water molecules; blue: water-protein N; magenta: water-protein C; red: water-protein O; black: sum of the three distributions.

On the Fourier series truncation peaks at subatomic resolution

Anne Bochow and Alexandre Urzhumtsev

Physics Department, Faculty of Sciences and Technologies,
University H. Poincaré Nancy 1,
B.P. 239, 54506 Vandoeuvre-lès-Nancy, France

1. Introduction

Recently, several macromolecular structures have been resolved at a subatomic resolution as can be found in PDB (Berman *et al*., 2000; Berstein *et al*., 1977). At such a resolution, new structural details become visible at the corresponding Fourier maps. A small size of these details requires more careful analysis of the images of the electron density. In particular it is important to avoid confusion between signal and noise.

The noise in Fourier maps can be attributed to several factors: errors in the experimental magnitudes, phase errors, Fourier series truncation. Some analysis of the first two sources of errors on macromolecular images at subatomic resolution has been reported previously (Afonine *et al*., 2004). This article addresses the role of errors in images caused by the resolution cut-off when this latter is unusually high for macromolecules, above 1 Å.

The problem of Fourier series truncation is well known in macromolecular crystallography. At a very low resolution the ripples have a large scale and may systematically increase or decrease the values of the Fourier map in large regions. They complicate the definition of the correct molecular envelope. However, there are too few structural studies at such a resolution to bring much attention to this difficulty. At a conventional resolution of 2-3 Å, in general these errors do not pose particular difficulties due to a relatively weak effect. In most of cases, they are mentioned in relation to analysis of isolated ions or solvent molecules.

It is known from many decades of studies on small molecules that at a subatomic resolution the noise caused by the series truncation is very significant. One of the main reasons for such increasing noise are very low values of the atomic displacement factor *B* which may reach 1-2 $Å^2$ for such extremely well ordered structures. To decrease the noise in density deformation studies, this density is analysed at difference maps.

The goal of the study presented here is to numerically compare the size of the density peaks due to density deformation with those caused by the series truncation. Not only the size of the noise peaks but also their shape is important to make a distinction between the noise and the signal. Both conventional and difference maps are studied.

2. Test data

The tests were conducted with the data previously described by Afonine *et al*. (2004). For a peptide model, placed in an orthogonal unit cell with parameters a=b=18 Å, c =

15 Å, two sets of structure factors have been calculated. First, the density was calculated without taking atomic interactions into consideration. For each atom, its electron density distribution $\rho^{spher}(\mathbf{r};B)$ was generated as a spherical function with conventional 5-gaussians atomic factor (Brown *et al*., 1999). Initially all atoms were considered as immobile with $B = 0$ Å^2. From this electron density, a set of Fourier coefficients $\{F^{spher}(\mathbf{s})\exp[i\varphi^{spher}(\mathbf{s})]\}$ was calculated at the resolution of 0.3 Å (the highest resolution at which a X-ray structure was determined is 0.25 Å, Takahashi *et al*., 1998). Secondly, an electron density distribution $\rho^{cryst}(\mathbf{r};B)$ was calculated by the quantum-chemical *DFT* method (program *SIESTA*, Sanchez-Portal *et al*., 1997) taking into account atomic interactions. Another set of Fourier coefficients $\{F^{cryst}(\mathbf{s})\exp[i\varphi^{cryst}(\mathbf{s})]\}$ obtained from this new density corresponded to structure factors of a crystal. Their magnitudes simulated the structure factor magnitudes obtained from the diffraction experiment. Comparative tests, conducted with these simulated data and with experimental data available for several other molecular crystals have shown that this method reproduces very realistically the practical situation (Afonine *et al*., 2004).

The difference between these two density distributions

$$\rho^{diff}(\mathbf{r};B) = \rho^{cryst}(\mathbf{r};B) - \rho^{spher}(\mathbf{r};B)$$

allowed for the estimation of the height of the density deformation peak at peptide bonds as roughly 0.4-0.5 e/Å^3. These peaks are reproduced quite exactly in the maps of the resolution of d = 0.5 Å and higher. In the maps at a resolution of about d = 0.9 Å, the peak value decreases roughly to 0.3-0.4 e/Å^3. With B increasing, the value of the peaks decreases. For this study, the centre of attention was the analysis of the deformation density and the noise around the C-C bond where the deformation density peak is roughly in the middle of the bond.

To analyse the perturbations influenced by the Fourier series truncation a series of maps with the Fourier coefficients $\{F^{spher}(\mathbf{s})\exp[i\varphi^{spher}(\mathbf{s})]\}$ at various resolution d was calculated. Since these maps show the image of the electron density corresponding to the model of spherical atoms, all peaks except those at the atomic centres are noise.

3. Main features of the noise

Fig. 1 shows a typical image of electron density, calculated from a spherical-atoms model at the resolution of 0.5 Å. One may note the blob at the C-C bond, the disk at the C-N bond and the ring at the C=O bond.

This shape of the noise can be understood given the image of an isolated atom as a central spherical peak surrounded by a series of ripples. For two more or less identical neighbouring atoms the superposition of such distributions gives an image with a cylindrical symmetry around the interatomic vector. The superposed ripples from the two atoms may form rings or blobs, depending on the distance to atomic centres and on the resolution. It is easy to estimate from one-dimensional analysis of images of the δ-function at the resolution d that the closest noise peak is at the distance roughly 5d/4 to the centre of the main peak. This allows a fast estimation for 2 bonded atoms to be done. The first noise peaks are superposed in the middle of the bond (that is the position of the deformation density peak for C-C) at the resolution 0.5-0.6 Å. At lower resolutions, the superposition should happen outside of the bond. In this case, the noise peak forms a ring with radius decreasing with the resolution.

A more precise analysis was conducted for the series of maps calculated with the coefficients $\{ F^{spher}(\mathbf{s})\exp[i\,j^{spher}(\mathbf{s})]\}$ at the resolution d. For each map the mean density values $<\rho(R;d)>$ were calculated at the distance R to the centre of the C-C bond in the plane perpendicular to this bond at its middle. The maps were calculated for different B values. Corresponding curves are shown in Fig. 2.

At the resolution d = 0.9 Å (with the atomic factor B = 0) the first peak is at the distance R • 0.9-1.0 Å from the bond (there is a noise ring of this radius), and its value is larger than the value of the deformation density peak. However, at such a resolution the noise in the middle of the bond is negligible. At a resolution of about d = 0.7 Å, the noise ring is closer to the bond, and the peak is higher. The noise in the middle again is negligible. When the resolution approaches d = 0.5 Å, the ring "collapses" at the bond between the two atoms. The value of the corresponding peak is one order of magnitude larger than the deformation density. When the resolution increases further, there is no more ripple superposition in the middle of the bond and this noise peak decreases.

The characteristics of the image are similar when B is different from 0. The curve smoothens, the ripples become less and less pronounced and the density in the middle of the bond becomes larger. However, it is much less for d = 0.7 Å than for d = 0.5 Å, as previously for B = 0. At the same time, the deformation density peaks decreases (Afonine *et al*., 2004) but is still significant at least for $B < 5$ Å^2.

4. Density images

Fig. 3 shows a series of three-dimensional images of the Fourier maps calculated at different resolution from 0.9 Å to 0.3 Å. In all cases the displacement factor B is equal to 0. As expected, the maps confirm the analysis of the curves shown in Fig. 2. Several complementary details can be observed. Since the height of the peaks and the noise increase significantly with the resolution, we failed to present well-illustrative images at the same cut-off density level.

At the resolution of 0.9 Å each of rings (rather, their arcs) belongs to several neighbouring bonds at the same time. At 0.7 Å these merged arcs are separated into individual full rings, one per bond. At 0.6 Å the radius of these rings is significantly smaller and the density is larger. At 0.5 Å the rings for all bonds but C=O collapse into a blob. These blobs should not be confused with the deformation density. The noise around the C=O bond behaves slightly differently due to a particular asymmetric character of this bond. This is also reflected in a particular behaviour of the corresponding deformation density. At resolution 0.4 Å these blobs are decomposed into separated small rings, which are shifted from the middle of the bond towards the atoms. Noteworthy, the ring shown for the C=O is a new ring, corresponding to the superposition of second ripples, while the first ring is attached to the central peak of the O atom (Fig. 4a). This phenomenon continues at 0.3 Å. At the resolution of 0.3-0.4 Å ripples can be seen inside the central "atomic" peak.

When similar images are analysed for larger values of B, both rings and blobs start to merge with the major peaks but continue to be seen until B reaches a critical limit (see a discussion in Afonine *et al*., 2004). Fig. 4b shows a superposition of two 0.5 Å-resolution maps calculated at B = 0 and at B = 2 Å^2. In the second map, the blob at the C-C bond is seen equally well as in the first map while of coarse at a lower cut-off level.

5. Images of difference density

The results given above show the importance of the computational noise when working at subatomic resolution. To avoid ripples in high-resolution images, crystallographers that work with small-molecule crystals use difference maps. There are two main ideas behind this approach. First, in conventional maps it is easy to see heavy (non hydrogen) atoms and many hydrogens; when looking for details at the next level, it is advisable to remove the main contribution already known. Second, it is known that there are ripples caused by heavy atoms showing large and sharp density peaks. Removing these peaks will stop ripples.

As illustrated by Afonine *et al*. (2004) the model refinement at a subatomic resolution often leads to the models where the values of the atomic factor B are larger than the real values. In this case, there is a risk of residual peaks and, as a consequence, a risk of residual ripples. To simulate such a situation the difference density has been calculated at the resolution of 0.5 Å with the coefficients

$$\{F^{\mathrm{spher}}(\mathbf{s})\exp[i\,\mathrm{j}^{\mathrm{spher}}(\mathbf{s})]\}_{\mathrm{B=1}} - \{F^{\mathrm{spher}}(\mathbf{s})\exp[i\,\mathrm{j}^{\mathrm{spher}}(\mathbf{s})]\}_{\mathrm{B=2}}$$

This image corresponds to the situation when the true value of B for all atoms is equal to 1 Å^2 but during refinement it has been estimated as 2 Å^2, similar to values found by Afonine *et al*., 2004).

Fig. 4c shows the corresponding Fourier map. The image is similar to the conventional map at 0.5 Å resolution, though the corresponding distribution is much lower. There is a significant peak in the middle of the C-C bond due to the Fourier series truncation. Its height is about 0.5 e/Å^3, the same as the height of the peak for the deformation density. At such a “resonance” resolution, when the superposition of the noise ripples creates a peak directly on the bond, there is still a high risk of confusion between the deformation density and the computational noise. Fig. 4c confirms that this problem may not be resolved even with difference maps traditionally used for studies of deformation density

As shown previously, it is very important to include all available data into refinement, that helps to estimate B values as precise as possible (Afonine *et al*., 2004). With correct B values, the deformation density peaks can also be seen in lower resolution maps. Our study suggests that even with the availability of 0.5 Å high-resolution data, it is worthy to calculate density maps at lower resolution. This is important to verify if the peaks indicate deformation density or computational noise. The computational noise would not be conserved but deformed or it would disappear.

6. Conclusions

In this study the behaviour of noise in Fourier maps at subatomic resolutions is shown. The shape of noise peaks is very specific. In most cases the bonds are surrounded by noise rings with a radius decreasing with the resolution.

A critical resolution is at about 0.5 Å when the rings transform into blobs at the interactomic bonds. These noise peaks could easily be confused with deformation density.

At this critical resolution the noise blobs may be present even at difference maps, traditionally used for deformation density studies. The value of the noise is comparable to the size of peaks of deformation density. This may be a source of an important confusion.

As a practical tool, this study suggests that when working at subatomic resolution, it is worthwhile to analyse a series of maps at a resolution below the limit. With this procedure the consistence of the peak which is supposed to indicate a deformation density can be verified. Especially strong inconsistencies may be observed when switching from 0.5 Å resolution maps to 0.7 Å resolution maps. In this case, the peaks can most likely be contributed to the Fourier series truncation effects and therefore do not indicate deformation density.

The authors thank V.Yu.Lunin for his suggestion to analyse the characteristics of noise peaks and P.Afonine for providing the data. This study has been conducted as part of a project towards a Master in Physics by A. D. Bochow.

References

Afonine, P.A., Lunin, V.Yu., Muzet, N., & Urzhumtsev, A. (2004). *Acta Cryst*., D**60**, 260-274

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

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

Brown, P.J., Fox, A.G., Maslen, E.N., O'Keefe, M.A. & Willis, B.T.M. (1999). *International Tables for Crystallography, Vol. C.*, Wilson, A.J.C. & Prince, E., eds., Kluwer Academic Publishers, Dordrecht, Boston, London, 548-589.

Sanchez-Portal, D., Ordejon, P., Artacho, E. & Soler, J.M. (1997). *Int. J. Quant. Chem*., **65**, 453-461.

Takahashi, Y., Ohshima, K.-i., Yamamoto, K., Yukino, K. & Okamura, F.P. (1998) *J.Appl. Cryst.*, **31**, 917-921

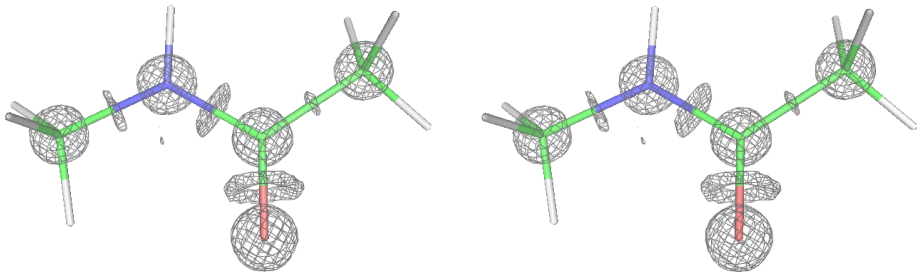


Fig. 1. Map calculated with the Fourier coefficients $\{ F^{spher}(\mathbf{s})\exp[i\,j^{spher}(\mathbf{s})]\}$ at the resolution of 0.5 Å for $B = 0$; cut-off 3.5 e/Å^3 .

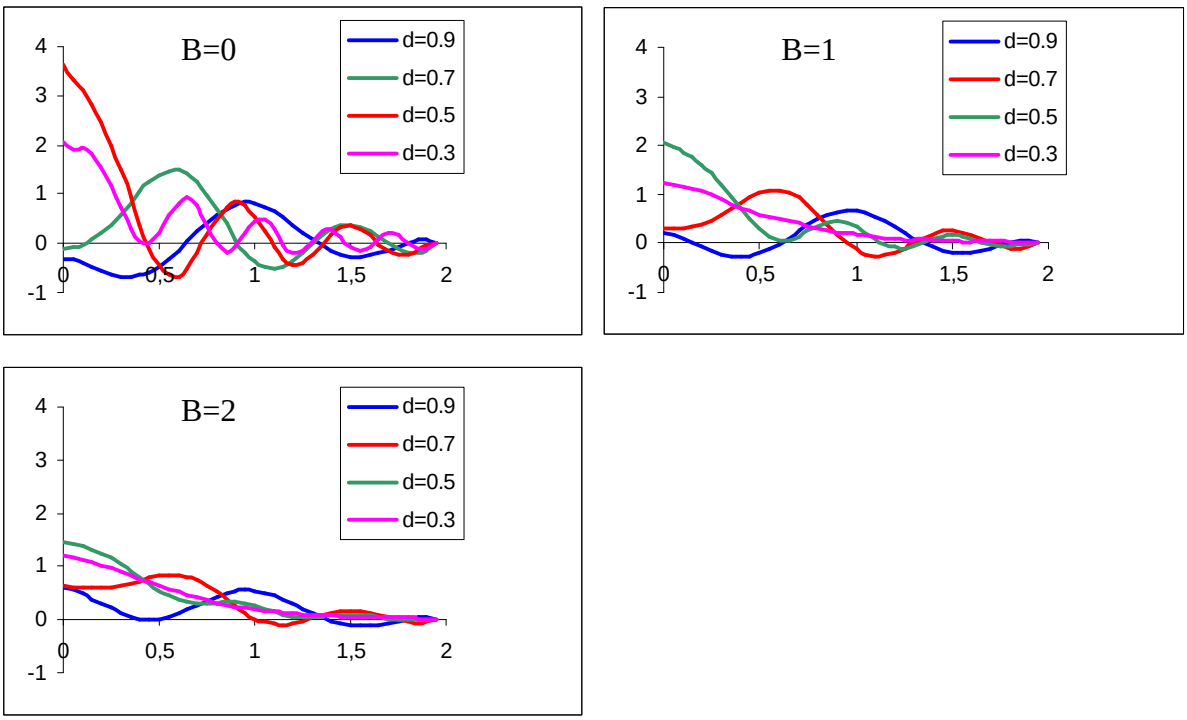
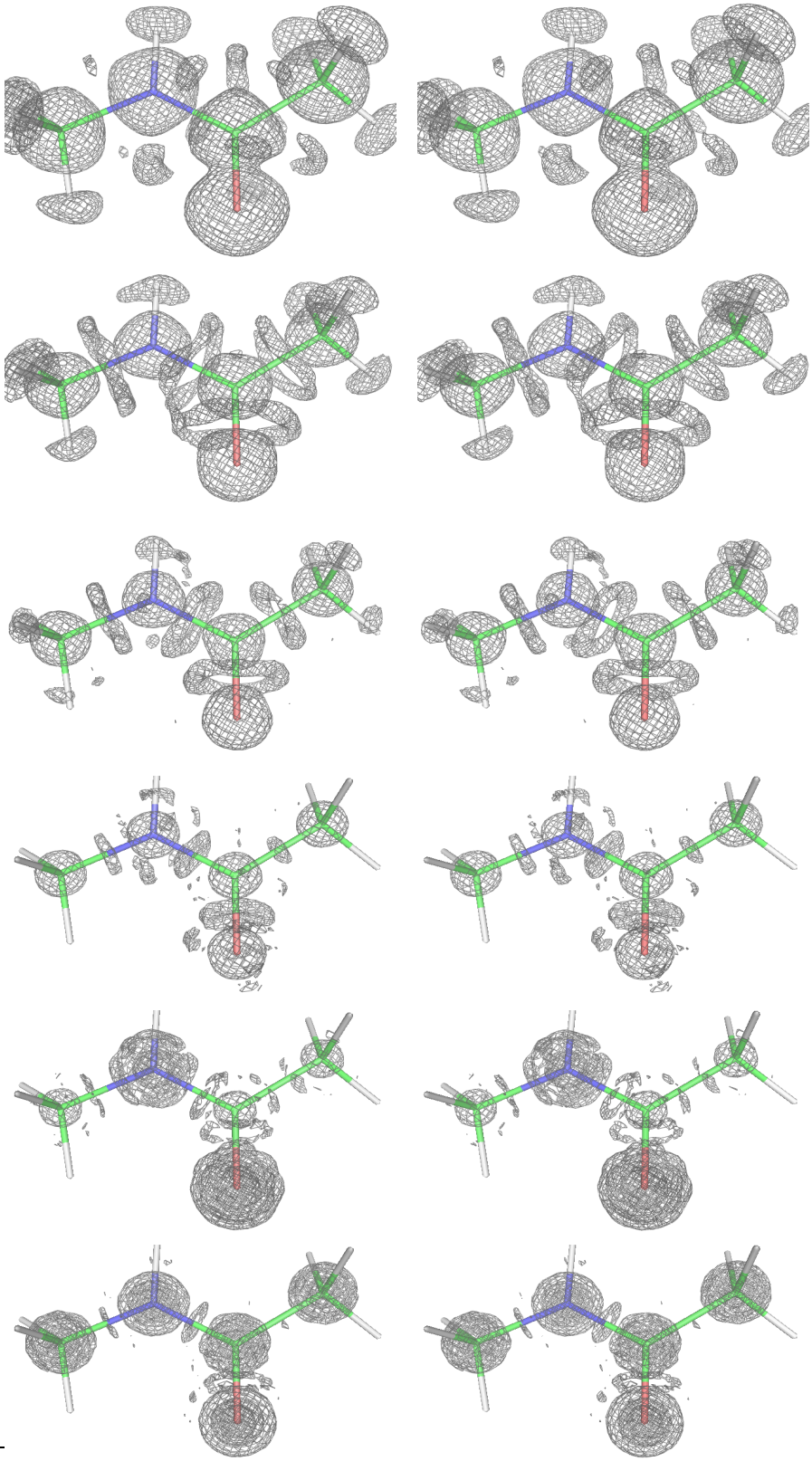


Fig. 2. Mean density in the plane perpendicular to the C-C peptide bond, taken at the middle of the bond, as a function of distance to the bond



a) 0.9 Å

b) 0.7 Å

c) 0.6 Å

d) 0.5 Å

e) 0.4 Å

f) 0.3 Å

1

Fig. 3. Map calculated with the Fourier coefficients $\{F_{\text{obs}}(\mathbf{s})\exp[i\,\varphi_{\text{calc}}(\mathbf{s})]\}$ for $B = 0$. a) resolution 0.9 Å ; cut-off 0.9 e/Å^3 ; b) resolution 0.7 Å ; cut-off 1.4 e/Å^3 ; c) resolution 0.6 Å ; cut-off 2.0 e/Å^3 ; d) resolution 0.5 Å ; cut-off 2.8 e/Å^3 ; e) resolution 0.4 Å ; cut-off 3.2 e/Å^3 ; f) resolution 0.3 Å ; cut-off 2.9 e/Å^3 ;

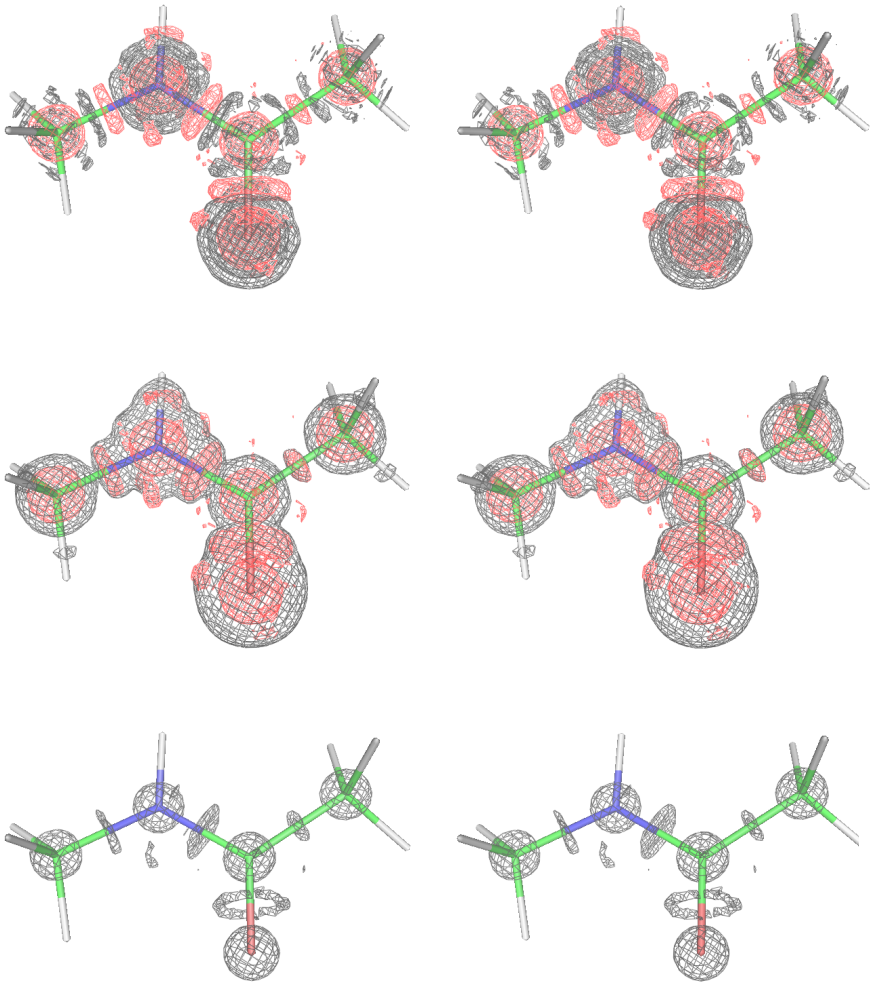


Fig. 4. a) Superposition of maps calculated with the Fourier coefficients $\{F^{spher}(\mathbf{s})\exp[i\,j^{spher}(\mathbf{s})]\}$ for $B = 0$ at the resolution of 0.5 Å (red) and at the resolution of 0.4 Å (black);

b) Superposition of maps calculated at the resolution of 0.5Å with the Fourier coefficients $\{F^{spher}(\mathbf{s})\exp[i\,j^{spher}(\mathbf{s})]\}$ for $B = 0$ (red; cut-off 2.8 e/Å^3) and $B = 2$ (black; cut-off 1.3 e/Å^3);

c) difference map at the resolution of 0.5 Å between the densities for the models with $B = 1$ and B=2 (see Section 5 for a detail description); cut-off 0.5 e/Å^3 .

Characterization of X-ray data sets

Peter H. Zwart, Ralf W. Grosse-Kunsteleve & Paul 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

1. Introduction

With the emergence of structural genomics, more effort is being invested into developing methods that incorporate basic crystallographic knowledge to enhance decision making procedures (e.g. Panjikar, 2005).
A key area where some crystallographic knowledge is often vital for the smooth progress of structure solution is that of judging the quality or characteristics of an X-ray dataset. For instance, detecting the presence of anisotropic diffraction or twinning while a crystal is on the beam line, may allow the user to change the data collection strategy in order to obtain a better or a more complete data set. In post-collection analyses, the presence of (for instance) non-crystallographic translational symmetry might help the user (or program!) to solve the structure more easily.
Of course, the identification of problems is by no means a guarantee that the problems can be overcome, but knowledge of the idiosyncrasies of a given X-ray data set permits the user or software pipeline to tailor the structure solution and refinement procedures to increase the chances of success.

In this report, a number of routines are presented that assist the user in detecting specific problems or features within a given dataset. The routines are made available via the open source *CCTBX* libraries (http://cctbx.sourceforge.net) and will also be included in the next available *PHENIX* (Adams, *et al.*, 2004) release.

2. Methods

2.1. Likelihood-based scaling

Absolute scaling is performed using a maximum likelihood method as proposed by Popov & Bourenkov (2003). The X-ray amplitudes are assumed to follow a Wilson distribution, with a resolution dependent variance that takes into account the effects of geometric regularities on the average intensity (Zwart & Lamzin 2004; Morris *et al*., 2004):

$$f(F_{obs} \mid k) = \frac{2(kF_{obs})}{\varepsilon\sigma^2(d^*)[1+\gamma(d^*)]} \exp\left[-\frac{(kF_{obs})^2}{\varepsilon\sigma^2(d^*)[1+\gamma(d^*)]}\right] \qquad \mathbf{1}$$

In the latter probability density function, $\sigma^2(d^*)$ is equal to the sum of squared atomic form factors and the term $\gamma(d^*)$ is a correction term accounting for resolution dependent behavior of the mean intensity due to geometric regularities. The term $\gamma(d^*)$ has been obtained from 20 high quality experimental datasets in a manner similar as described by Zwart & Lamzin (2004). $\sigma^2(d^*)$ is determined from the cell contents as provided by the user.
The factor ε accounts for the statistical effect of symmetry on the expected intensity (Stewart & Karle, 1976). F_{obs} is an observed structure factor amplitude and k is a scale factor that brings the observation to an absolute scale with atomic displacement parameters equal to 0:

$$k = \exp[-k_s]\exp\left[-\mathbf{h}^{\mathrm{T}}\mathbf{U}^*\mathbf{h}\right] \qquad \mathbf{2}$$

The tensor $\mathbf{U}^*$ is an anisotropic atomic displacement parameter (Grosse-Kunstleve & Adams, 2002), the vector **h** is a Miller index. Note that the scalar part of the scale factor is an exponent, $\exp[-k_s]$, rather than the simple constant that is more frequently used (Giacovazzo (1992), expression 5.12). The use of an exponent has the benefit that no special precautions need to be taken during minimization procedures to ensure the positivity of k.

The scale factor and elements of $\mathbf{U}^*$ are determined via the minimization of the negative of a log-likelihood function:

$$\Lambda[\{F_{obs}\} \mid k_s, \mathbf{U}^*] = -\sum_{j=1}^{N_{obs}} Ln\left[f\left(F_{obs,j} \mid k(k_s, \mathbf{U}^*)\right)\right] \qquad \mathbf{3}$$

The negative log likelihood is optimized using a gradient driven L-BFGS minimizer (Liu & Nocedal, 1989). During optimization, symmetry constraints on the elements of $\mathbf{U}^*$ and its effect on the partial derivatives are taken into account (Grosse-Kunstleve *et al.*, unpublished results).

A related (and independent) implementation of the likelihood-based scaling routine is available in *PHASER* (McCoy *et al.*, 2005). An isotropic, moment based method has been implemented in ARP/wARP (Morris *et al.*, 2004).

2.2 Detection of pseudo translational symmetry

The presence of pseudo translational symmetry can often be detected by computing a native Patterson at truncated resolution. A significant off-origin peak indicates the presence of a large number of parallel inter-atomic vectors, due to translational NCS or due to an n-fold NCS axis parallel to an n-fold crystallographic axis. In order to determine whether an off-origin peak is significant, a frame of reference is needed. For this purpose, the largest off-origin peaks for roughly 500 high quality data sets from the PDB with 1 molecule in the asymmetric unit were computed and stored. In the latter calculations, only peaks further then 15 Å away from an origin peak were considered and the Patterson function was calculated using data between 10 and 5 Å resolution. The peak height of the largest peak in a Patterson map was expressed as a fraction of the height of the Patterson origin peak.

The distribution of the selected peaks heights can be described by an extreme value distribution (Weisstein, 1999). The collected set of Patterson peaks denoted by $\{Q_{\max}\}$ are limited between 0 and 1. The following standard transformation (Zwart, A.P., personal communication) scales the set of Patterson peak heights to the domain $[0,\infty)$:

$$Q'_{\max} = \frac{Q_{\max}}{1-Q_{\max}} \qquad \textbf{4}$$

A theorem similar to the central limit theorem, suggests that the values of $Q'_{\max}$ follow a Frechet distribution (Weisstein, 1999). Applying the transformation specified in equation 4 and assuming a Frechet distribution for $Q'_{\max}$ results in the following cumulative distribution function of the height of the largest off-origin peak in a Patterson map:

$$F(Q_{\max}) = \exp\left[-\left(\frac{Q_{\max}}{a(1-Q_{\max})}\right)^{-b}\right] \qquad \textbf{5}$$

The constants a and b of this distribution function, were fitted using likelihood methods given the observed set of Patterson peak heights. The fitted constants a and b and are equal to $6.79*10^{-2}$ and 3.56, respectively. The observed and modeled cumulative distributions are shown in Fig 1.

The significance of an observed off-origin Patterson peak can be assessed by computing a so-called p-value: the probability that a Patterson peak of that height or larger occurs by chance. This value is equal to $1-F(Q_{\max})$. If a threshold of 1% is chosen, all off-origin peaks with a height larger than 20% of the origin peak are considered to be significant.

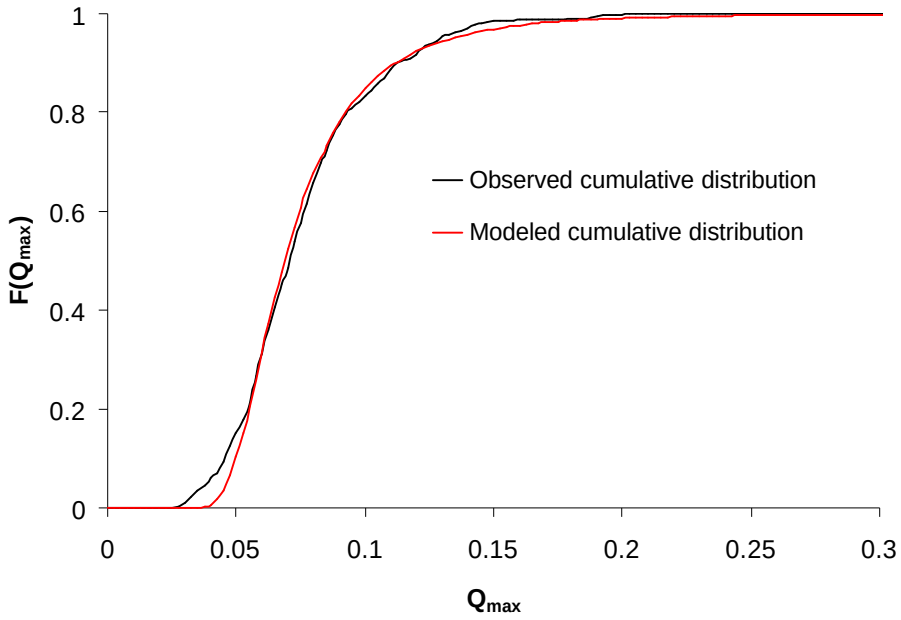


Figure 1: Observed and modeled cumulative distribution of largest off -origin Patterson peak height Q_{max}.

2.3. Twin detection

The presence of twinning can usually be identified on the basis of the Wilson or intensity ratio (e.g. Dauter, 2003). In some cases however, the presence of pseudo translational symmetry or anisotropic diffraction influences the intensity statistics in such a way that twinning cannot readily be detected, even though it is present. Therefore, the |L| statistic developed by Padilla & Yeates (2003) is designed to be a more robust statistic for the detection of twinning, as it is relatively insensitive to anisotropy in the data and the presence of pseudo centering. The |L| statistic is defined as follows:

$$|L| = \frac{|I_1 - I_2|}{I_1 + I_2} \qquad 6$$

The intensities I_1 and I_2 have associated Miller indices that are close in reciprocal space, and are not necessarily related by a twin law:

$$\mathbf{h}_1 - \mathbf{h}_2 = (d_h n_h, d_k n_k, d_l n_l) \qquad 7$$

d_h, d_k and d_l are random signed integers and the constant n_h,n_k,n_l are chosen on the basis of the location of significant off -origin Patterson peaks.

The first and second non-central moments of |L| are equal to 1/2 and 1/3 for untwined, acentric data, respectively. If twinning is present, the moments are lowered and reach a value of 3/8 and 1/5 for perfectly twinned data. In order to detect twinning, the same data sets as used to obtain a distribution of Patterson

peak heights, was used to compute <|L|> and <|L|2> values for data between 10 and 3.5 Å resolution. The resulting set (<|L|>,<|L|2>) was used in the construction of a multivariate Z-score, known as the Mahalanobis distance (Mardia, 1980). For a given observed (<|L|>,<|L|2>) pair, the Mahalanobis distance is equivalent to the distance of the given pair to the multivariate mean in units of standard deviation. Values of the Mahalanobis distance larger then 3 indicate that the (<|L|>,<|L|2>) pair is outside the range expected for experimental data sets and could thus indicate twinning.
The dependence of the Mahalanobis distance on the twin fraction is shown in Fig. 2, and indicates that X-ray data sets with a twin fraction larger then 6% have an expected Mahalanobis distance larger than 3.

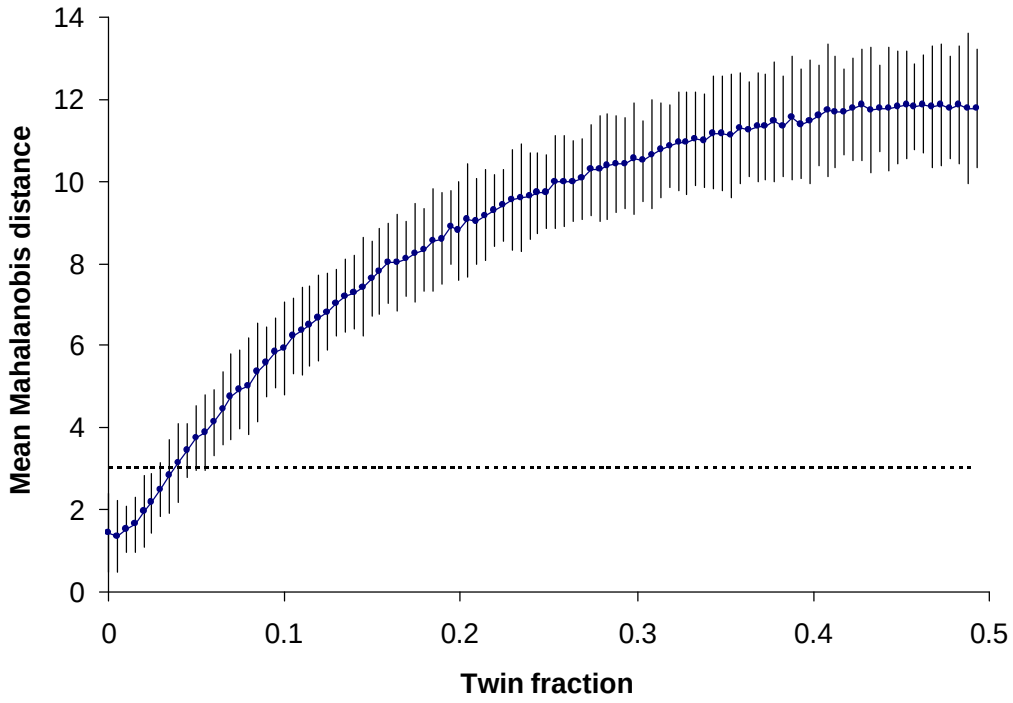


Figure 2: The expected Mahalanobis distance for the fist and second moment of |L| of an X-ray data set (blue dots). The vertical error bars span three times the estimated standard deviation of the expected Mahalanobis distance. The black dotted horizontal line is drawn for the Mahalanobis distance being equal to three. The values shown in this figure were obtained via numerical simulations.

2.4. Estimation of the twin fraction

Although twin detection and the estimation of a twin fraction are related problems, it is useful to leave these topics separated, as will become clear in section 3.3.
Estimating the twin fraction can be carried out in a number of ways. First of all the |H| test (Yeates, 1988; 1997) gives a numerically easily accessible estimate of

the twin fraction. A Britton analysis (Fisher & Sweet, 1980), although less straightforward than the H-test, is another common way of estimating the twin fraction.

Another way would be to estimate the twin fraction using the $|L|$-statistic. As the distribution of L for a given twin fraction is known for acentric reflections, a maximum likelihood approach can be used to estimate a twin fraction. A comparison of the 3 implemented twin fraction estimation procedures is shown in Fig. 3, where the mean values of estimated twin fraction are plotted given the true twin fraction. Although results of these analyses show that the estimation of the twin fraction via the L-statistic is sub-optimal in comparison to the two other methods, especially for large twin fractions, it could be potentially be useful in cases when a two-fold non crystallographic symmetry axis is parallel to a potential twin operator. In that case, the independence between intensities required by the Britton and H-test, is violated, resulting in an overestimation of the true twin fraction. The determination of the twin fraction *via* the L-test is most likely less sensitive to these types of problems. The biggest limitation of twin fraction estimation using the L statistic is its large associated standard deviation (results not shown).

It should be noted that the distribution of the normalized intensity can also be used to estimate the twin fraction within a maximum likelihood framework (Zwart *et al.*, unpublished results). However, the drawback of this method is its extreme sensitivity to translational symmetry.

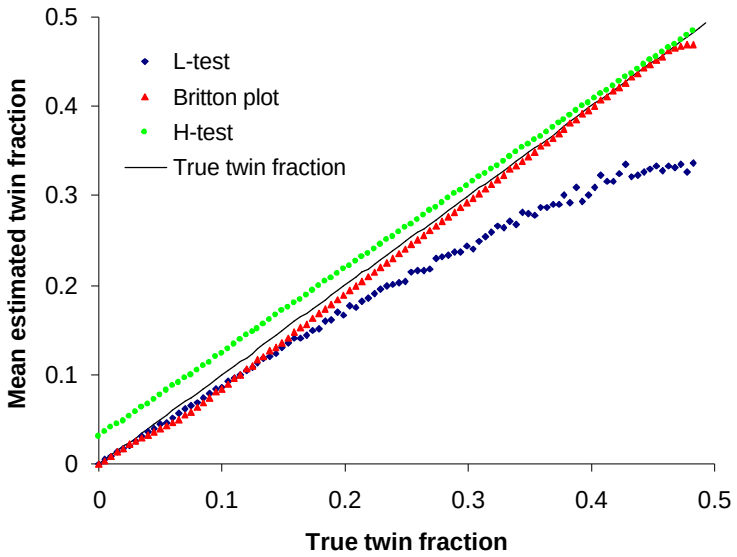


Figure 3: The estimation of the twin fraction on simulated, twinned data. Mean values over 100 trials per true twin fraction are shown. Both the H test and the Britton plot methods behave reasonably over the full range of twin fractions. The estimate of the twin fraction *via* the L statistic shows considerable bias, especially at large twin fractions.

3. Examples

3.1. The effect of anisotropy correction on the cumulative intensity distribution

The X-ray data from PDB entry 1awu is known to be anisotropic (see for instance Padilla and Yeates, Fig. 1) and the resulting cumulative normalized intensity distributions differ significantly from the theoretically expected distributions. However, as a result of the likelihood-based anisotropic scaling procedure outline above, the estimated anisotropic overall B-value can be used to correct for the observed anisotropy. The effect of the anisotropy correction on the cumulative intensity distribution is shown in Fig. 4.

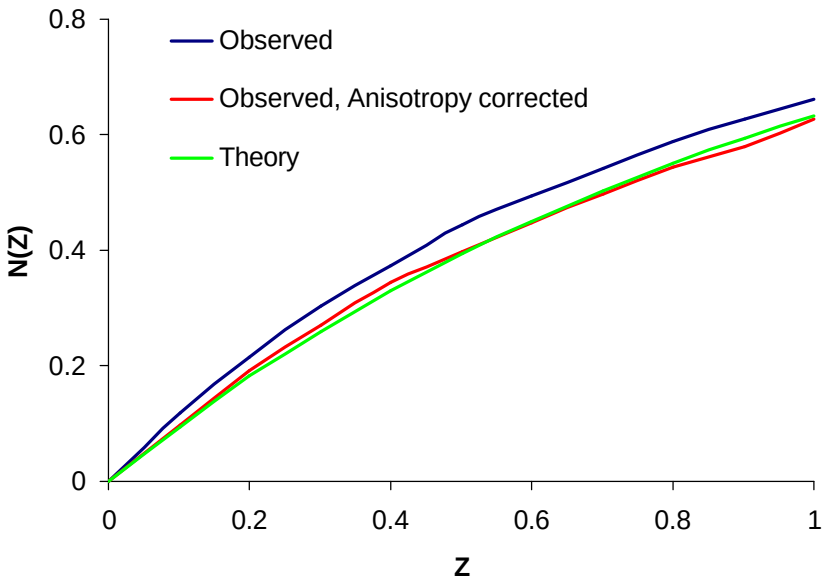


Figure 4: The effect of anisotropy correction on the cumulative intensity distribution.

3.2. Detection of non-crystallographic translational symmetry

The detection of non-crystallographic translational symmetry is illustrated using 4 example data sets obtained from the PDB. The datasets used are 1sct, 1ihr, 1c8u and 1ee2. 1sct is a classic example of pseudo centering, whereas 1ihr and 1c8u are both structures with a two-fold NCS axis (almost) parallel to a two-fold (screw) axis. 1ee2 does not possess any non-crystallographic translational symmetry.

The results for detection of translational symmetry via the presence of significant peaks in the native Patterson function are illustrated in Table 1.

Table 1: The detection of pseud o translational symmetry.

PDBID	Peak Height	p-value (%)	$\langle I^2 \rangle / \langle I \rangle^2$	$\langle \lvert L \rvert \rangle$
1sct	77%	0.0000094	2.81	0.490
1ihr	45%	0.0014	2.51	0.539
1c8u	20%	1	2.22	0.493
1ee2	10%	15	2.09	0.497

Note that the peak height (and thus the p -value) is correlated with the in tensity ratio. The local intensity statistic $\langle |L| \rangle$ is however less sensitive to the presence of pseudo centering.

3.3. Detection of twinning and estimation of the twin fraction

The detection of twinning is illustrated using 5 examples obtained from the P DB. For each data set, the relevant statistics are given, as well as the reported twin fraction, if available. The twin laws for each test case were derived automatically from first principles (Flack, 1987; Grosse-Kunstleve *et al.*, 2005).

Table 2: Detect ion of twinning. The p -value is the p -value corresponding to the height of the largest off origin Patterson peak height. Maha(L) denotes the Mahalanobis distance of the observed $(\langle |L| \rangle, \langle |L|^2 \rangle)$ pair.

PDBID	Space group	Twin operator	p-value (%)	$\langle I^2 \rangle / \langle I \rangle^2$	Maha(L)	Estimated twin fraction			Reported twin fraction *
						L-test	*Britton*	*H-test*	
1hfo	C2	h,-k,-h-l	52	2.00	0.58	0.00	0.01	0.02	None
1o0i	C2	h,-k,-h-l	28	2.09	1.08	0.00	0.44	0.46	N.A.
1hh8	C2	h,-k,-h-l	83	1.89	5.62	0.08	0.02	0.09	0
1xed	$P2_1$	h,-k,-h-l	38	1.79	6.34	0.10	0.35	0.38	0.37
1ap9	$P6_3$	h,-h-k,-l	58	1.84	7.48	0.12	0.28	0.33	None

*: None: no twinning was mentioned in the publication ; N.A.: No publication available.

Although most of the test cases are easy to interpret (1hh8, 1xed and 1ap9 a re all most likely twinned and 1 hfo is not twinned), the X-ray data of 1o0i behaves as if it is untwin ned, but intensities related by the putative twin operator are highly correlated, resulting in a estimated twin fraction of larger then 0.4. This can be rationalized by postulating that the twin operator is in fact a crystallographic symmetry element and that the reported space group is too low.

Note that if the decision about whether or not the data are twinned were made solely on the basis of the estimated twin fraction, 1o0i would be flagged as a potential perfect twin, even though the intensity statistics indicate that the structure is not twinned.

4. Conclusions

The routines presented here are aimed to provide the crystallographer with a set of statistics characterizing a given data set. The likelihood-based scaling routine provides an easy, non-graphical way of detecting anisotropy of the data by inspecting the elements of the estimated anisotropic tensor.
For the detection of pseudo translational symmetry and twinning, a similar philosophy is adopted: the summary statistics of the given data set are listed within the context of a reference set of known structures. The non-graphical nature of these analyses allows a straightforward way of incorporating general crystallographic experience into automated structure solution pipelines and allows expert and non-expert users to quickly place the results in context.

The algorithms are available as part of the open source CCTBX libraries (http://cctbx.sourceforge.net) and will also be available via future *CCP4* releases that incorporate the CCTBX.

5. Acknowledgements

We gratefully acknowledge the financial support of NIH/NIGMS through grants 5P01GM063210, 5P50GM062412 and 1R01GM071939. Our work was supported in part by the US department of Energy under Contracts No. DE-AC03-76SF00098 and DE-AC02-05CH11231.

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.

Dauter, Z. (2003). *Acta Cryst.* **D59**, 2004-2016.

Fisher, R. G. & Sweet, R. M. (1980). *Acta Cryst.* **A36**, 755-760.

Flack, H.D. (1987). *Acta Cryst.* **A43**, 564-568.

Giacovazzo, C. (1992). *Fundementals of Crystallography*, Oxford University Press.

Grosse-Kunstleve, R.W. & Adams, P.D. (2003). *J. Appl. Cryst.* **35**, 477-480.

Grosse-Kunstleve, R.W., Afonine, P.A., Sauter, N.K. & P.D. Adams. (2005). *IUCr Computing Commission Newsletter* **5**.

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

Mardia, K.V., Kent, J.T. & Bibby, J.M. (1980). Academic Press, London, UK.

McCoy, A.J., Grosse-Kunstleve, R.W., Storoni, L.C. & Read, R.J. (2005). *Acta Cryst.* **D61**, 458-464.

Morris, R.J., Blanc, E. & Bricogne, G. (2003). *Acta Cryst.* **D60**, 227-240.

Morris R.J., Zwart P.H., Cohen S., Fernandez F.J., Kakaris M., Kirillova O., Vonrhein C.,
Perrakis A. & Lamzin V.S. (2004). *J. Synchrotron Rad.* **11**, 56-59.

Padilla, J.E. & Yeates, T.O. (2003). *Acta Cryst.* **D59**, 1124-1130.

Panjikar, S., Parthasarathy, V., Lamzin, V.S., Weiss, M.S. & Tucker, P.A. (2005). *Acta Cryst.* **D61**, 449-457.

Popov, A. N. & Bourenkov, G. P. (2003). *Acta Cryst.* **D59,** 1145-1153.

Stewart, J.W. & Karle, J. (1976). *Acta Cryst.* **A32**, 1005-1007.

Weisstein, E.W. (1999) "Extreme Value Distribution.", Mathworld: http://mathworld.wolfram.com/ExtremeValueDistribution.html

Yeates, T. O. (1988). *Acta Cryst.* **A44**, 142-144.

Yeates, T. O. (1997). *Methods Enzymol.* **276**, 344-358.

Zwart, P.H. & Lamzin, V.S. (2004). *Acta Cryst.* **D60**, 220-226.