CCP4 NEWSLETTER ON PROTEIN CRYSTALLOGRAPHY

Number 33. January 1997

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

Contents

Editor: Martyn Winn
Daresbury Laboratory, Daresbury,
Warrington, WA4 4AD, UK

CCP4 - Recent changes

Martyn Winn & Adam Ralph

Daresbury Laboratory,
Daresbury,
Warrington
WA4 4AD, U.K.

Since the last Newsletter, version 3.2 of the CCP4 suite has been released. In addition to a number of bug fixes and enhancements to existing programs, this release includes the following new programs:

1. MAKEDICT: will produce TNT or PROTIN dictionary entries from a PDB or PROTIN file (Phil Evans).
2. MATTHEWS_COEF: written by Misha Isupov to calculate the Matthews coefficient given the cell, symmetry, number of atoms etc..
3. RASMOL: RasMol 2.6 (unix and VMS versions) is now distributed with CCP4 (Roger Sayle).
4. SCALEPACK2MTZ: Convert SCALEPACK output into an MTZ file. It then must go through TRUNCATE to convert to Fs.

A full list of changes for version 3.2 is appended below.

Documentation for most of the main programs is now distributed in html format, in addition to the man page source and formatted versions. Point your browser at $CCP4/html/INDEX.html These html files are at the moment generated automatically, and are perhaps not ideal. They will be gradually improved, and at some point the man page source files (man1/*.1 etc.) will be made obsolete.

An official mirror site for the Suite was set up in September at the Photon Factory, Japan by Atsushi Nakagawa. This supplements the other mirror site in San Diego. Details can be found on the CCP4 home page.

If you are having problems with CCP4 programs either compiling or running then have a look at the Problem Page. This contains various fixes effected since the latest release.

Finally, thanks to all those who visited us at the IUCr Commercial Exhibit Show in Seattle. We hope you were suitable impressed by the poster!

Changes to the Suite

Building etc.

- $CCP4/x-windows/XCCPJIFFY/Imakefile: compiler options for HPUX sent by Morten Kjeldgaard.
- $CCP4/x-windows/xdlgjk/Makefile: added install and empty-targets procedures, in line with $CPROG. Added MAKEFILE.COM
- $CPROG/MAKEFILE.COM: procedure for Solomon was causing a warning but compilation was not affected. This has been fixed to eliminate warning. Solomon compiled with case sensitivity.
- Compilation of C code changed due to library.h.
- configure: because of problems with the optimisation the default is now O1 for OSF1. IRIX6.2 section added. Warning messages have been surpressed for IRIX5*, HPUX A.09.* and OSF1.

Program changes

- AMORE: if the B for an atom is zero or below then it is reset to 20.0. A message is printed if this is the case.
- BONES2PDB: latest version from KDC on 7/10/96.
- DM: lastest version from KDC on 7/10/96.
- ECALC: resolution keyword is now acted upon.
- F2MTZ: calls to fatal were incorrect.
- FFT: keyword FILL didn't do anything - fixed. New keyword GRID SAMPLE to specify the sampling grid as a fraction of the resolution. FFTs in P2 did not work properly and has been removed. P1 should be used instead of P2.
- FHSCAL: labels DANO and SIGDANO changed to DPH and SIGDPH to be consistant with SCALEIT.
- IPDISP: spdfils have been changed for MARs. -Ms is now a bigmar with squashed format. -Mo has been removed but is equivalent to -M. -m small mar in original format. All these are big endian. -V is the same and now -v is small mar from VAX i.e. little endian.
- MAMA2CCP4: latest version from KDC on 7/10/96.
- MAPMASK: latest version from KDC on 7/10/96.
- MAPROT: latest version from KDC on 7/10/96.
- MLPHARE: problems with printing MNFs when monitoring reflections fixed.
- MTZ2VARIOUS: typo in defining external function was causing compilation problems. Now fixed. EXCLUDE FREER will now exclude any freeR subset. Any FreeR sub-set in the MTZ file can be output as the FreeR set to XPLOR, SHELX. Dummy columns IDUM?? have been added, these will output as integers in USER mode. An mmCIF file of the reflection data and associated data can now be generated. XPLOR output changed to output SIGMA instead of SIGM.
- MTZDUMP: FORMAT keyword introduced so that the reflection list can be output in a certain format. Resolution limits are applied to the overall statistics. Missing reflections are flagged with '?'. Ranges are now calculated for the partial statistics tables (STATS keyword).
- MTZUTILS: the UNIQUE option data from file 2 was output instead of file 1. Bug now fixed. Associated columns properly scaled with SCALE option.

- NCSMASK: latest version from KDC on 7/10/96.
- POLYPOSE: Bugs fixed so that default values work if FIX and INCLUDE not specified.
- POSTREF: the column label 'ABSFAC' and 'SIGABS' have been changed to 'SCALE' and 'SIGSCALE' in order to work with output from SCALA.
- PROCHECK: a write statement in PPLOT was over 132 characters. Changed the write to avoid this. Only a problem with VMS.
- PROTIN: +PROLSQ & REFMAC the format of the PROTOUT file has changed so that the files will not be compatible with last version. Change is so that residue numbers up to 9999 are allowed.
- REFMAC: Output FOM as well. Made P1 default symmetry. Now supports cubic spacegroups. Defaults have changed since last version please read the documentation.
- RESTRAIN: updated to version 4.3.5 from Ian Tickle. Isotropic Bs are written out properly when outputing aniso. Us.
- ROTAPREP: the way the batches are handled is now more consistant. NBATCH keyword has been removed, the number of batches is taken from the file where appropriate. MISBATCH keyword added to specify missing batches.
- SCALEIT: interesting stats. on large isomorphous/anomalous differences produced for REFINE and ANALYSE modes. Normal probability analysis now seperates centrics and acentrics. Now outputs Kraut scale instead of reciprocal.
- SIGMAA: problem with missing FC sorted.
- SOLOMON: CCP4 masks were not being read properly, now fixed. O style rotation and translation operators do not have to have the header line. Problems have have been caused when averaging using CCP4 masks, now resolved.
- TRUNCATE: divide by zero problem fixed. The output now contains H K L F SD Dano SD F+ SdF+ F- SdF- Imean SD I+ SdI+ I- SdI-. Also, scaling with the Wilson plot is compulsory.
- XDLMAPMAN and XDLDATAMAN: removed system dependent routines. Altered the menu and io windows so there are sized flexibly. Also, added command line qualifier so font for menu can be changed.

New programs

- MAKEDICT: will produce TNT or PROTIN dictionary entries from a PDB or PROTIN file.
- MATTHEWS_COEF: written by Misha Isupov to calculate the Matthews coefficient given the cell, symmetry, number of atoms etc..
- RASMOL: RasMol 2.6 (unix and VMS versions) are now distributed with CCP4.
- SCALEPACK2MTZ: Convert SCALEPACK output into an MTZ file. It then must got through TRUNCATE to convert to Fs.

Library

- $CLIBD/(no)chiral_pep*.dic: correction to ARG sidechain.
- $CLIBD/atomsf_neutron.lib: Change atomic number of D to 101 to bring into line with RBROOK.

- chelp.c: some irregularities fixed.
- chelp.h: machine dependent stuff now in library.h.
- library.c: integer type was incorrectly defined for Big-endian machines with CONVERT_FROM set.
- library.h: has been created from library.c. This will contain the machine dependences required for all C programs.
- mtzlib.f: bug in LRREFL fixed; affected BIOMOL programs.
- rwbrook.f: s/r SFREAD moved here from AMORE,MLPHARE,REFMAC,SFALL

Documentation

- RSTATS: document brought in line with prog. FREE keyword added.
- Tutorials: a new set of tutorials have been made in five main areas: MIR, MR, refinement, density moification and MAD. These procedural scripts can be found in $CEXAM/tutorial.
- html versions of the documentation for most of the main programs are in $CCP4/html Point your browser at $CCP4/html/INDEX.html
- x-windows/doc: file names have been changed but essentially the *.txt files are similar to the old *.doc* files.

Withdrawal

- ROTAVATA: will be unsupported in release 3.3. SCALA will become the recommended program.
- PROLSQ: will be unsupported in release 3.3. REFMAC will become the recommended program.

AUTO-INDEXING OSCILLATION IMAGES USING A PATTERSON FUNCTION

John W. Campbell, CCLRC Daresbury Laboratory

INTRODUCTION

The possibility of using a three dimensional FFT as the basis of an auto-indexing method was mentioned to me in an informal discussion with Phil Evans. This article describes some tests carried out to examine how useful such a procedure might be as the basis of an auto-indexing or crystal orientation determination procedure for monochromatic protein oscillation images. The trials described focussed on the use of a **single** oscillation image for the orientation determination. The trials were carried out using both simulated and real data.

BASIS OF THE METHOD

The basis of the method is simply that the Fourier Transform can provide a direct transformation between a reciprocal lattice (which may be determined experimentally) and a real space set of vectors from which the cell vectors and the cell orientation may be found. In theory this should provide a more direct route to the required information than the more usual procedures involving the determination and clustering of difference vectors in reciprocal space (see refs. 2, 3 & 4).

To carry out the procedure, a fine three dimensional orthogonal reciprocal lattice grid (parallel to a set of laboratory axes) is set up. A set of reciprocal lattice points is determined from the observed diffraction data. For each of these reciprocal lattice points, an 'intensity' value of 1.0 is assigned for the nearest grid point of the othrogonal lattice. All other points of this orthogonal lattice are assigned intensities of zero. A three dimensional Patterson function is then computed from the orthogonal lattice using an FFT program and the resulting Patterson function is searched for the vectors to determine the cell and its orientation.

POTENTIAL PROBLEMS

A number of problem areas are obvious. In the first instance, it is necessary to carry out the calculations using a 'fine' grid for the 3-dimensional FFT as the finer the grid, the less will be the rounding errors is assigning the observed reciprocal lattice points to the nearest FFT grid points. How fine this grid needs to be in practice is probably one of the main factors determining whether the method has practical value using today's computers.

Assuming a sufficiently fine grid can be used, the following problem areas still remain.

- The very sparse sampling of the reciprocal lattice.
- The asymmetry of the sampling of the reciprocal lattice.
- The errors introduced by the fact that a finite oscillation angle is used when recording the image.
- Errors in the measurement of the primary beam position, the spot positions and the crystal to image distance.

The latter two categories of error, of course, affect any method using oscillation images.

INITIAL TRIALS

Some initial trials were carried out in one and two dimesions using grids from 2048 down to 512 points with a data resolution of 1.0 Angstroms and cell parameters up to about 85 Angstroms. Typically a random sample of 5% of the precicted reciprocal lattice points data was included in the FFT calculation. As the results seemed promising, the trials were extended into three dimensions. The first trials in three dimensions were carried out using a grid of 512x512x512 points. Although again they looked promising, the computation time for such a 3-D FFT was taking around 13 minutes of cpu time with an elapsed time of around 45 minutes on an Indy workstation. This basically suggested that the method was unlikely to be practical unless a coarser grid could be used and a grid of 256x256x256 was then tried. This reduced the cpu requirement to less than 2 minutes and it was this grid that was used in all the subsequent examples unless otherwise stated. Such computaion time could probably be significantly reduced by resticting the number of grid divisions grid to a power of 2 and making more use of the fact that the input data are very sparse.

PROGRAMS USED

The main programs used in the trials were the FFT and PEAKMAX programs from the CCP4 program suite and IMSTILLS from the MOSFLM suite. A series of jiffy programs were also written and used.

Special versions of the routines OUT_SPOTS
: These were used in building temporary versions of the ROTGEN program to create some simulated data for trying out the method.

PREP_MTZOSC
: Prepare an MTZ file for input to the FFT program from reciprocal lattice coordinates written from a version of ROTGEN with a special version of OUT_SPOTS included.

STILLS_MTZ
: Prepare an MTZ file for input to the FFT program from reciprocal lattice points calculated from spot positions measured using IMSTILLS.

PVEC
: Analyse selected peaks as output from PEAKMAX.

PVEC_REFN
: Analyse selected peaks as output from PEAKMAX and carry out some limited refinement.

CELL_VEC

Find vectors within given distance distance and angular criteria from peaks output by PEAKMAX.

FIND_ORIENT

Output cell parameters and missetting angles from a set of three selected cell vectors.

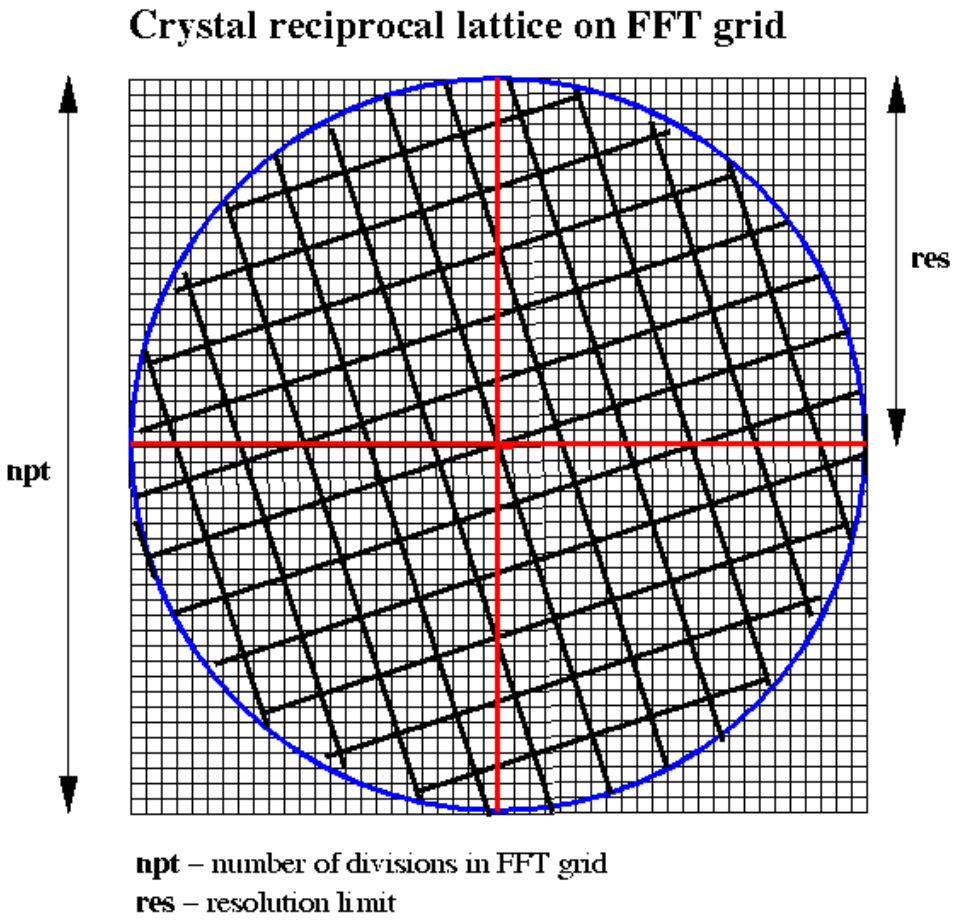
THE FFT GRID

As indicated above, almost all the trials were carried out using a grid of 256x256x256 for the FFT calculations. This meets the following requirements:

- It is about the finest grid which could currently be envisaged as the basis for a practical method.
- The grid is a power of two which would be most beneficial for future optimisation.
- It matches reasonably for the size of cell envisaged provided that reasonably low resolution data are used for the indexing - see further below.

The grid used for the FFT and its relationship to the crystal reciprocal lattice for data of a given resolution are illustrated in two dimensions in the following diagram.

Figure 1 Choice of FFT Grid for Crystal Data of a Given Resolution



Let **npt** be the number of divisions in each dimension of the grid and **res** be the resolution of the data to be used, then the cell size for the 3-D Patterson function to give vectors of the correct dimensions is given by:

```
FFT cell size = npt*res/2.0
```

The grid spacing **rlgrid** is given by:

```
rlgrid = 2.0*alam/(res*npt)
```

where **alam** is the wavelength at which the data are collected.

The FFT grid corresponds to a index range of (-npt/2 + 1) to (npt/2 - 1)

The conversion of a reciprocal lattice point (dimensionless) to the nearest indices (hkl) for the FFT input data is given by:

```
h = NINT (x/rlgrid)
k = NINT (y/rlgrid)
l = NINT (z/rlgrid)
```

where x,y,z are the dimensionless reciprocal lattice coordinates for an observed reflection.

It is probably desirable when it comes to searching for vectors in the 3-D patterson function that the primary cell vectors should be less than half the FFT cell dimension in length. If the FFT grid is to be restricted to 256, then this means that the resolution of the data to be included in the calculation will need to be restricted to give a sufficiently large FFT cell based on the expressions given above. For a cell of 200 Angstroms the data input would need to be restricted to a resolution of say 3.5 Angstroms whilst for a cell of 100 Angstroms, 2.0 Angstrom data would be suitable.

STEPS FOLLOWED

The calculations were carried out in the following stages:

Find spots or simulate data

For real data, find spot positions from the oscillation image using the program IMSTILLS. For simulated data, use a temporary version of the program ROTGEN with a modified spots output routine linked in.

Create an MTZ file

Calculate the nearest grid points to the reciprocal lattice coordinates computed from the spot positions data (or read in for simulated trials) and output this data as an MTZ reflection file for input to the FFT program. For real data, the program STILLS_MTZ is used and for simulated data, the program PREP_MTZOSC is used.

Calculate a 3-D Patterson Function

A three dimensional Patterson function in space group P1bar is calculated using the FFT program. The map is scaled to give an origin peak of height 100.0.

Find peaks in the Patterson

The Patterson map is searched for peaks using the PEAKMAX program. In most cases a threshold of 30.0 (i.e. just under 1/3 of the origin peak) was used for the search.

Analyse the vectors Found

This is currently done with a number of jiffy program.

PVEC and PVEC_REFN select peaks above a requested threshold, find the vector lengths within a requested range and output details of the vectors selected and the angles between them. The PVEC_REFN attempts some preliminary refinement of these vectors using the reciprocal space data using a general least squares routine or a method described by Clegg for refining 't' vectors (ref 1.).

CELL_VEC allows for a search of any solutions for which each of three vectors are within given ranges and are separated by requested minimum angles.

Determine the Orientation

The program FIND_ORIENT can be used to determine the cell parameters and crystal setting from three vectors selected manually from the output of the PVEC_REFN program. This data can be fed into ROTGEN to check whether the correct orientation has basically been found.

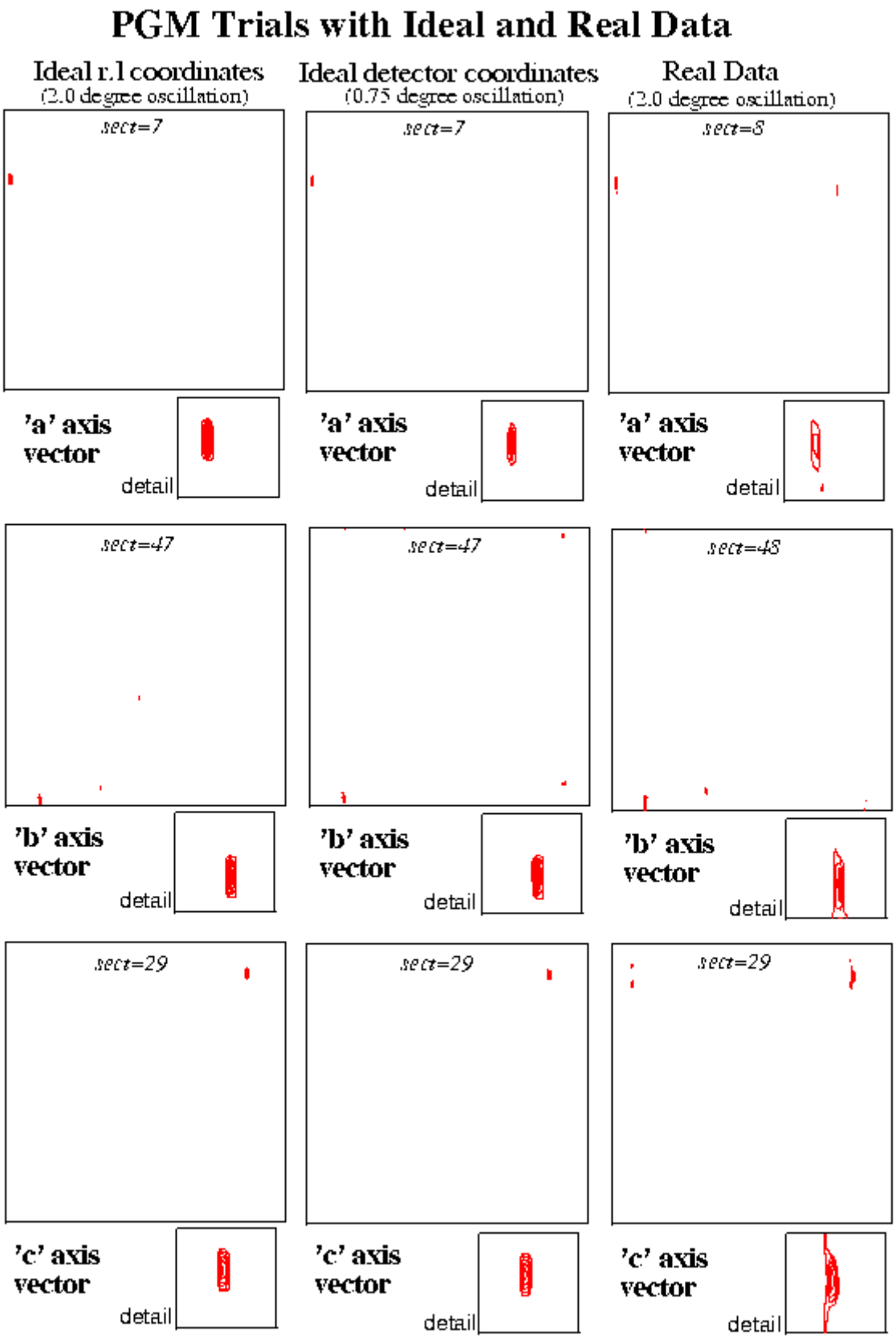
YEAST PGM - SIMULATED AND REAL DATA

Two simulated data trials were carried out for a crystal of the enzyme Yeast Phosphoglycerate Mutase (PGM) (Monoclinic, C2, a=96.2 b=85.8 c=81.9 beta=120.5). In the first of these, the ideal three dimensional reciprocal lattice coordinates were calculated (using a modified version of ROTGEN) for the reflections which would occur on a two degree oscillation image. In the second case data were calculated for a 0.75 degree oscillation image again using a modified version of ROTGEN. In this case the reciprocal lattice points were calculated from the predicted spot positions on the detector thus including the unavoidable source of error due to the use of a finite oscillation angle. The procedure was then applied to the first real set of data, a 2.0 degree oscillation PGM image recorded on image plate (small MAR). All three sets corresponded to the same orientation of the crystal. A crystal to image distance of 250.0 mm was used for the simulated data and the experimental distance of 137.0 mm was used for the real data.

Using the jiffy programs described above, the Patterson peaks were analysed and in each case the unit cell vectors could be clearly seen. Other high peaks obviously resulted from vectors to the 'C' centre, across the cell faces etc. In this example the cell vectors are all of a similar magnitude and the vectors corresponding to multiples of the unit cell were outside the range examined. Such multiple cell length vectors were however clear in some of the other examples described below.

The sections of the Patterson map containing the peaks corresponding to the cell vectors are illustrated for the two sets of simulated data and the real data.

Figure 2 PGM Trials with Ideal and Real Data



For the real data, a series of runs were done making adjustments to the initial starting position for the centre of the primary beam. The best position for the beam, as judged by the closeness of cell dimensions obtained to those of a reference set, was very close to that of the refined centre position from a run of MOSFLM. The results were encouraging in spite of large oscillation range used and in spite of the fact that the 3.0 Angstrom data used occupied only the central section of an image recorded to 1.8 Angstroms resolution.

EXAMPLE DATASETS USED

The following table gives a complete list of the protein crystals used in the indexing trials:

Table 1: Protein Crystals used for Auto-Indexing Tests

Code	Protein	Space-group
PGM	Yeast Phosphoglycerate Mutase	C2
LYS	HEWL Tetragonal Lysozyme	P43212
PRI	Prismane protein	P212121
PST	Pig Serum Transferrin	C2
NPL	Narcissus Pseudonarcissus Lectin	C222
INS	Insulin	P212121

All data used were collected on the Daresbury SRS. The PGM images were collected by H.C. Watson and J.W. Campbell. The LYS, PRI and PST images were collected by E.M.H. Duke and the NPL and INS images were collected by P.J. Rizkallah and M.Z. Papiz.

RESULTS FROM REAL EXAMPLES

The parameters used for the recording and processing of the test images (one from each dataset) are shown in the following table:

Table 2: Parameters used with example datasets

Sample type	Detector type	Lambda (A)	ctof (mm)	Osc. angle	Resolution Obs.	Resolution Used	No. spots	FFT cell	Peaks-in-FFT rms	Peaks-in-FFT thresh
PGM	IP	1.00	137.0	2.0	1.8	3.0	278	384.0	4.24	30
LYS	IP	0.92	310.0	1.0	2.1	2.5	311	320.0	4.00	35
PRI	IP	1.70	270.0	1.5	3.4	3.5	232	448.0	4.77	50
PST	IP	0.92	310.0	1.0	2.1	2.5	206	320.0	4.93	50
NPL	CCD	0.87	90.0	1.0	1.4	2.5	162	320.0	5.56	30
INS	CCD	0.87	90.0	0.1	1.4	2.5	141	320.0	5.95	40

Note: Grid for FFT was 256 in all cases

Two figures show the Patterson peaks at the ends of the unit cell vectors. In each case a detail for the section (a 20x20 grid point box) is shown. The elongation of the peaks in the vertical direction on the plots corresponds to a elongation in the direction of the X-ray beam and is presumably due to the lower resolution of the recorded reciprocal lattice data in that direction.

Figure 3 Patterson Peaks from MAR Image-Plate Examples

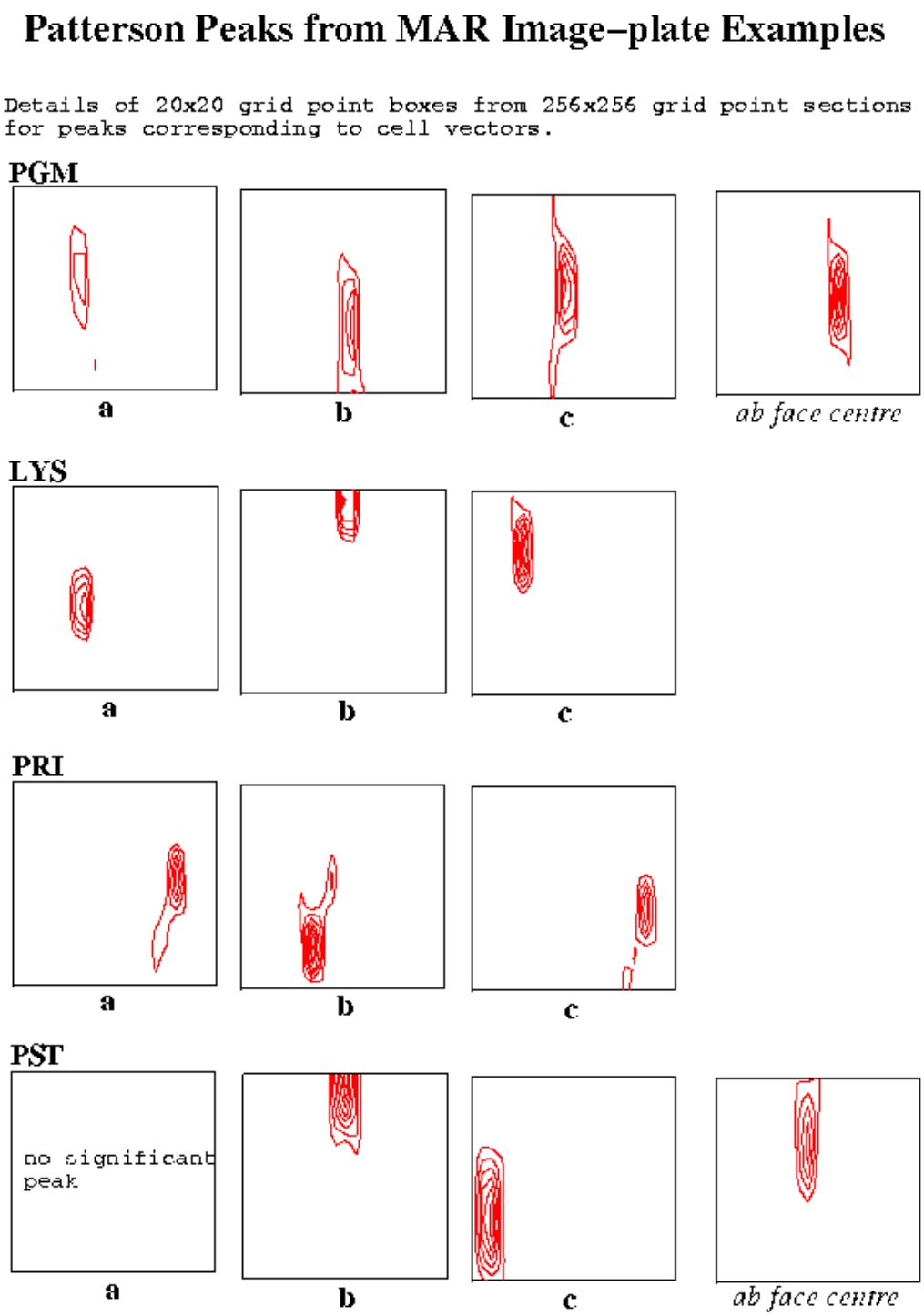
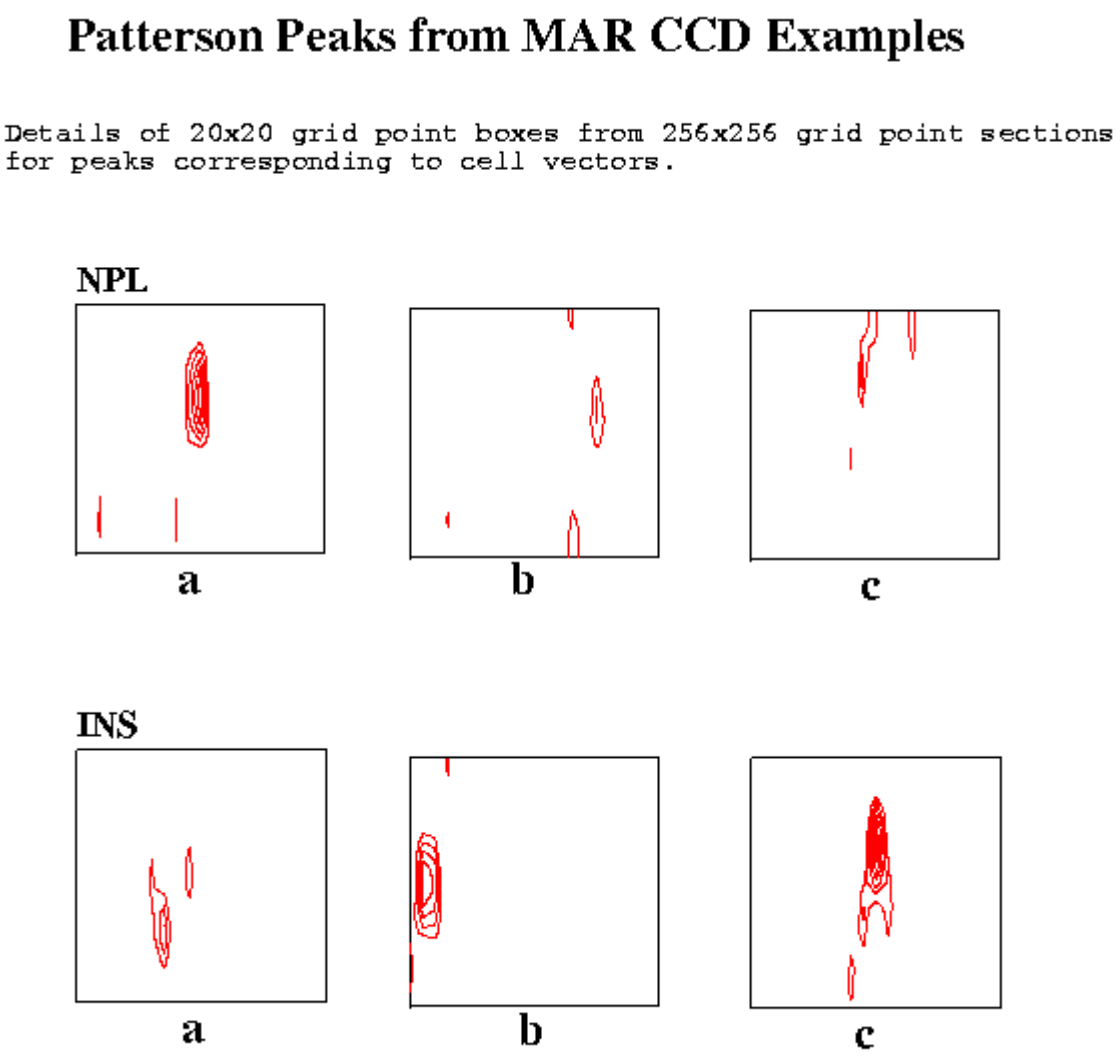


Figure 4 Patterson Peaks from MAR CCD Examples



As might be expected, the weakest of the three unit cell vectors in the PGM example was the one lying approximately in the direction of the X-ray beam. The effect of sparser data in this direction and also the finite size of the oscillation angle are presumably factors affecting such peaks. The two degree oscillation angle is probably larger than desirable in any case but it is encouraging that the required vectors are nevertheless still well defined. The vector to the 'C' face centre is stronger than that of the 'a' vector and is also shown in the first figure.

In both the LYS and PRI cases, all the desired vectors showed up well.

The most disturbing result was in the case of the PST crystal (C face centred, monoclinic cell) where the 'a' axis vector could not be detected. On the other hand, the vector to the face centre gave a strong peak and the choice of this vector with the two other clearly identifiable cell vectors gave a valid primtive cell. The FIND_ORIENT program was used to determine the crystal orientation for this primitive cell and a simulation using ROTGEN gave a good match to the observed

image. The procedure was repeated using 3.5 Angstrom data with a 256x256 grid and also 3.0 Angstrom data with the finer 512x512x512 grid. In both these cases, the 'a' vector could be seen.

The NPL data collected on the MAR CCD system gave reasonable peaks for the cell vectors.

The INS data were collected in narrow phi slices of 0.1 degrees and, as in the other cases, a single image was used. Though, in this case, not given the cell dimensions but given that the cell was orthorhombic, the cell dimensions were readily determined.

The results of the cell determinations are shown in the following table together with reference values for comparison.

Table 3: Determined and reference cell parameters.

Test	---Determined-cell-parameters----						----Reference-cell-parameters-----					
	a	b	c	alph	beta	gamm	a	b	c	alph	beta	gamm
PG	97.1	85.7	82.7	90.5	121.4	89.0	96.2	85.8	81.9		120.5	
LYS	79.5	80.6	38.3	86.4	87.7	89.8	79.6		38.5			
PRI	65.0	65.9	154.8	90.1	90.5	89.7	65.1	65.7	155.2			
PST	--	45.2	79.6	90.0	--	--	224.8	45.2	79.2			
	45.2	79.6	116.6	106.6	101.3	90.0	45.2	79.3	114.7	104.9	101.3	89.9
NPL	71.8	101.1	38.3	89.2	86.3	90.9	73.1	100.9	37.0			
INS	52.8	57.7	36.2	91.4	91.3	90.8	50.9	56.9	37.1			

Table 4: Peak heights in the Patterson of cell vectors

Sample	------Peak- Height------			
	a	b	c	ab-cen
PGM	35.0	55.0	62.5	73.6
LYS	61.3	75.8	57.7	
PRI	62.0	85.9	59.8	
PST	--	96.1	76.7	56.0
NPL	33.7	32.1	67.2	
NPL	40.0	60.5	76.6	

It was generally observed that peaks with heights of 30 or more were found for the expected vectors. With the current refinement procedures for the individual 't' vectors, the examples show errors of up to about 3% in the lengths of the cell vectors found and up to 3.7 degrees in the cell angles.

CONCLUSIONS

The results obtained would seem to confirm that the proposed method could be put to practical use. A customised program would need to be written incorporating the stages described with particular emphasis on speeding up the FFT calculation step. The refinement of the individual potential cell vectors also needs further attention. Having selected suitable cell vectors, the stages following that including refinement of the cell and its orientation, cell reduction etc. would be analogous to that used in already well established procedures.

As with other auto-indexing or orientation determining procedures, it is important to have good values for the centre of primary beam position and the crystal to image distance.

REFERENCES

1. Clegg W., (1984) "Enhancements of the Auto-Indexing Method for Cell Determination in Four-Circle Diffractometry", J. Appl. Cryst. **17** 334-336.
2. Kabsch W., (1988) "Automatic Indexing of Rotation Diffraction Patterns", J. Appl. Cryst. **21** 67-71.
3. Kim S., (1989) "Auto-Indexing Oscillation Photographs", J. Appl. Cryst. **22** 53-60.
4. Higashi T., "Auto-Indexing of Oscillation Images", J. Appl. Cryst. (1990) **23** 253-257.

ACKNOWLEDGEMENTS

My thanks are due to Phil Evans (MRC Cambridge) for drawing my attention to the subject and to my colleagues at the Daresbury laboratory, in particular Liz Duke, Pierre Rizkallah and Miroslav Papiz for making available some test images for the trials.

Interactive Visualization of Macromolecular Crystal Packing with Internet Tools: VRML and Java

Yu Wai Chen and Tai Y. Fu

Information interchange has been totally revolutionized by the recent developments of internet-relating software standards. Among these, two of the hottest areas that are of particular interest for interactive visualization of 3-D models are the Virtual Reality Modeling Language (VRML) and the Java language. VRML is an emerging standard for describing non-static navigatable 3-D environments and is particularly useful for sharing 3-D models over the internet. Java is an object-oriented language that allows generation of machine-independent programs to be executed from WWW browsers. With Java, 3-D models can be described with objects like lines and spheres.

To exploit the use of these tools in visualizing intermolecular interaction of macromolecules, the authors have developed WWW services which transforms Brookhaven PDB coordinate files into 3-D crystal packing models that can be manipulated interactively. VRML 3-D world files can be examined with an external VRML browser (e.g. Webspace from Silicon Graphics) or a VRML viewer plugin (e.g. CosmoPlayer from SGI) for the WWW browser. A 3-D model created with the Java applet can be displayed and manipulated in a Java-enabled WWW browser (e.g. Netscape 3.0) directly. The VRML method has been described in full in Fu & Chen (1996).

VRML service, "*xpack*": http://dta.med.harvard.edu/ubc/banff/xpack.html
Java service, "*Java lattice*": http://dta.med.harvard.edu/ubc/kelowna/latte.html

The user interfaces of both services are very simple; users are just required to fill out a form specifying only two parameters:

- The **radius** of a viewing sphere within which the symmetry-equivalent neighbour molecules will be included.
- The source of the **PDB file** - this source can be specified either as a URL where the PDB files can be picked up off the Internet (FTP or HTTP protocols) or as a full-path filename in the user's local file system (the WWW browser needs to support "file upload"; e.g. Netscape 2.0).

The core of both Web services consists of a "translator" program that converts a PDB file into the respective 3-D models. In the case of the VRML service, this is achieved with a Common Gateway Interface (CGI) script written in the Perl language. The PDB file submitted by a user is translated in the remote server machine into a VRML world file which is then returned to the user's local machine for display. The user can also choose to install the service on their local machine if they plan to use the service frequently. For the Java service, the "applet" contains all programming instructions required for coordinates translation and model display. Coordinate translation is done in the user's local machine.

The algorithm of the two translator programs are similar:

Unit-cell parameters and space group information are first extracted from the 'CRYST1' record of the PDB file. If this record is missing, only the identity molecule will be displayed in the output model. The translators do not read the 'SCALE*n*' lines and do not support non-standard orthogonalization of coordinates. The symmetry operation data used are extracted from the library file, *symop.lib* from the CCP4 suite.

Speed is the major concern with any 3-D rendering. The manipulation of complex models requires extensive computing resources and can be painfully slow. Hence, in designing the Web services, the aim is to make the simplest 3-D model (wireframe) which allows the fastest response to the client and, at the same time, contains adequate molecular information for packing analyses. Consequently, the simpler approach of drawing all atoms and determining their connectivities by summing their covalent radii was abandoned. Instead, only the most interesting features of the macromolecule(s) are presented to achieve maximum computing performance. At present, these features include the protein alpha-carbon (C-alpha) backbone, disulphide bridges, haem groups, and the phosphate backbone of nucleic acids. The resulting 3-D model consists of the crystallographic unit cell and the macromolecules in various symmetry-related positions. In the Java service, the user can also choose to display in the all-atom mode.

Both of these 2 methods are potential tools for further development in the visualization of 3-D scenes. We can envision that machines with ever-increasing CPU speed and memory will one day allow photo-realistic model rendering with surface polygons (rather than lines) to be used in this kind of applications.

Reference

Fu, T.Y. and Chen, Y.W. (1996). Visualization of macromolecular crystal packing using Virtual Reality Modelling Language (VRML). *Journal of Applied Crystallography*, **29**, 594-597.

Taking the fun out of map interpretation ...

Gerard J. Kleywegt & T. Alwyn Jones
Department of Molecular Biology
Biomedical Centre, Uppsala University
Uppsala - Sweden

Interpretation of experimental electron-density maps has traditionally been a difficult, time-consuming, yet also fun and exciting step in the structure-determination process. It is one of the few activities for which (successful) black boxes have not yet been developed.

ESSENS

We have recently described a method to detect rigid structural entities in electron-density maps **[1, 2]**, and implemented this in a program called ESSENS **[3]**, which is distributed as part of the RAVE package **[4]**. Briefly, this program reads a map and a structural fragment, oes a complete rotation of the fragment for every grid point in the map, and for each position and orientation calculates how well the fragment fits the local density (the "score"). When the calculations are finished, the scores are written out as a new map, which can be contoured in O **[5]**. By using a penta-alanine fragment in either ideal alpha-helical or ideal beta-strand conformation as the template, the result is a new map which shows how well a helix or strand fits at each point in the map. Examples of the application of this technique to two structures are discussed in **[1]**, and figures of the "helix and strand maps" are also available on the Web **[2]**. This technique, simple (and CPU-time consuming) as it is, turns out to be very powerful. First, if the score map reveals helices and strands, the model-building process is greatly facilitated. Second, if the score map is featureless, this may well indicate that the phases are not good enough, and that the crystallographer is probably better off spending some more time on collecting more derivative data than on staring at an uninterpretable map.

MORTEN'S MODIFICATION

The previous paragraph basically describes the state of the program at the time when the paper **[1, 2]** was submitted. Since then, a number of improvements have been made. First, Morten Kjeldgaard suggested a modification of the algorithm which essentially imprints and image of the helix or strand in the score map. The effect of this is that the map becomes much clearer and easier to interpret. In the original implementation, the image of a strand or helix looked essentially like a C-alpha or main-chain trace with little detail. In the new version, the image includes all atoms of the poly-alanine fragment, and in particular the visibility of the C-beta atoms is a major improvement (which, in the case of helices, makes it very easy to deduce their directionality). Initially, Morten's modification was applied for every orientation at every grid point, which made the method ~2 times slower than the original one. However, it can easily be applied *a posteriori* , which means that in a single run of the

program one obtains both the original score map and the map which results from Morten's modification. This is how the method is currently implemented in ESSENS. The map resulting from Morten's modification is called the display map, since it is the most appealing for contouring purposes.

EXTRACTING SOLUTIONS

A shortcoming of the original program was that, although the output map showed where the structural fragment fitted the map, the information about the best-fitting orientation was completely lost. This has recently been changed, by storing the best set of rotation angles for every grid point (encoded as a single integer number), and writing these out to a file. A new program, SOLEX **[6]** (for SOLution EXtractor) was written which reads the score map, the rotation file and the structural fragment, and extracts the top solutions, writing them to a new PDB-formatted file. This PDB file can be read into O again, and the fragments of strand or helix can be used in the model-building process. In addition, specifically for helices and strands, an (experimental !) option is available which will attempt to "connect the dots", *i.e.* to merge bits of strand or helix which lie close to one another and are more or less parallel, in order to build up longer secondary structure elements. At present, the algorithm appears to be working reasonably well, except that the directionality is sometimes wrong, so the crystallographer must be alert.

IF THE FOLD IS OLD ...

The secondary structure elements that are found by SOLEX are also written to a file which can used as input to DEJAVU **[4]**. DEJAVU is a program that looks for fold similarities between a structure and a large database of structures derived from the PDB. At the 1994 CCP4 study weekend, we demonstrated the use of DEJAVU in finding proteins with similar folds using only Bones-derived "secondary structure elements" (in effect, these are simply guesses of the beginning and end coordinates of helices and strands), even when the directionality of the helices and strands is unknown (or uncertain). Similarly, the files now produces by SOLEX can be fed into DEJAVU, and the program will attempt to find other proteins with a similar (partial) fold. In our standard test case (P2 myelin protein), this works surprisingly well. SOLEX finds eight beta-strands, and when DEJAVU is asked to look for similar proteins which have at least six strands in common with our "unknown" structure (ignoring directionality), the program comes up with four correct hits (including P2 myelin protein itself), and no false hits, albeit that the orientation of one of the hits is wrong. Nevertheless, if this were a real case, the tracing problem would essentially be solved, in that the coordinates of one of the DEJAVU hits could be "stolen" to jump start the model-building process.

OTHER APPLICATIONS

ESSENS and SOLEX also have other applications, for example in real-space, phased molecular replacement calculations (although in this case reciprocal-space methods are probably to be preferred for reasons of speed). This method was used in the structure determination of an acetylcholinesterase/fasciculin complex **[7]**. The acetylcholinesterase molecule was easily found using standard molecular replacement techniques. However, positioning the fasciculin molecule in a map phased on acetylcholinesterase was difficult. ESSENS was used to verify the manually obtained solution (which was correct), by using a truncated poly-alanine

model of fasciculin as the search fragment (see **[1]**, **[2]**, and **[7]** for details). For this type of calculation, the ability to extract any number of solutions conveniently with SOLEX is obviously important.

Finally, we have experimented with other types of template, such as a di-alanine, a tryptophan ring, *etc.* , with varying results. There is no inherent limitation on the type of structural fragment that is used as a template other than that it must be (assumed to be) rigid, *i.e.* not contain any free conformational torsion angles. Hence, the method should also work very well when searching for nucleic acids, sugar rings, many ligands, structural motifs (*e.g.* , hairpin turns) or even structural domains.

AVAILABILITY

The RAVE package (which includes ESSENS, SOLEX and many other programs) is available to academic users free of charge (from the O ftp server). For more information about RAVE, contact GJK (gerard@xray.bmc.uu.se). For more information about O, contact TAJ (alwyn@xray.bmc.uu.se).

REFERENCES

- **[1]** Kleywegt, G.J. and Jones, T.A. (1997). *Acta Cryst.* **D53**, in the press.
- **[2]** A preprint of reference **[1]** is available from our Web site at URL: http://alpha2.bmc.uu.se/~gerard/essens/essens.html
- **[3]** The manual for this program is available at URL: http://alpha2.bmc.uu.se/~gerard/manuals/essens_man.html
- **[4]** Kleywegt, G.J. and Jones, T.A. (1994). In "*From First Map to Final Model* ", pp. 59-66, CCP4 Proceedings.
- **[5]** Jones, T.A., Zou, J.Y., Cowan, S.W. and Kjeldgaard, M. (1991). *Acta Cryst.* **A47**, 110-119.
- **[6]** The manual for this program is available at URL: http://alpha2.bmc.uu.se/~gerard/manuals/solex_man.html
- **[7]** Harel, M., Kleywegt, G.J., Ravelli, R.B.G., Silman, I. and Sussman, J.L. (1995). *Structure* **3**, 1355-1366.

NOTE

The Uppsala Software Factory now has its own home page on the World Wide Web, providing access to many resources and services in Uppsala, including:

- manuals for, and information about Gerard's programs
- previous publications in this Newsletter
- some preprints
- educational material (O, model building, and refinement and rebuilding)
- several resources pertaining to hetero-compounds, and the generation of dictionaries for such compounds for both O and X-PLOR
- our local (non-serious !) Newsletter, DOMBO (to entertain you while you wait for your ESSENS job to finish)

The URL is: http://alpha2.bmc.uu.se/~gerard/manuals/

SCALA

Phil Evans

MRC Laboratory of Molecular Biology, Hills Road, Cambridge, CB2 4RP,UK
pre@mrc-lmb.cam.ac.uk

A new version of Scala with many changes will be included in the next CCP4 release. This article describes some of the main features, and changes from earlier versions.

Scala now includes the averaging of multiple observations of reflections, and supercedes the program Agrovata. Many of the statistical analyses from Agrovata have been retained, and some new ones added. The program typically does four passes through the data, although the steps may be performed separately (steps (3) & (4) always go together):

1. an analysis to get initial scale factors
2. a scaling pass (several cycles), determining the scales according to the chosen scaling model
3. an analysis pass to analyse discrepancies, and by default to correct the standard deviations
4. a final pass to calculate scales, analyse agreement & write the output file. By default, this step now mimics the program Agrovata and writes a file of averaged intensities for input to the program Truncate. There are two other output options: OUTPUT SEPARATE, the scaled observations are written out, as in the earlier versions of Scala; and OUTPUT UNMERGED, the observations are scaled, and partials are summed, but multiple observations are not merged. This last option may be useful for the MADSYS approach to MAD phasing, in which the phasing calculations are done on unmerged data.

Scaling models

Scale & relative Bfactor: for data from synchrotrons, it is usually best to use a separate scale factor for each image (SCALES BATCH), since the incident beam intensity can change discontinuously between images. However, the relative B-factor is essentially a correction for absorption and radiation damage, and depends on the crystal rather than on the beam, so is likely to vary smoothly. The program now has an option to vary the Bfactor smoothly, while giving each image its own scale factor (eg SCALES BATCH BROTATION SPACING 5).

There remains a need for an anisotropic scaling option to cope with crystals whose diffracting power falls off anisotropically: this version contains a new anisotropic scale option, but it is not very satisfactory, as it is usually ill-determined, for the same reasons as the 3-dimensional scaling is ill-determined, unless a reference dataset is used. More work is needed on this.

Partially recorded reflections

Programs such as Mosflm and Denzo which integrate each slice of a reflection separately leave the scaling program with a problem: given a collection of integrated parts, on successive images, how do we know when we have got all of them? This is particularly a problem with Denzo, which normally refines all the prediction parameters for each image, so that different parts of the same reflection may be predicted with different parameters (Mosflm normally refines parameters using data from more than one image, though different parameters are still used for each image). There would be no problem if all parts were predicted with the same parameters, though this would not cope with slipping crystals.

Scala now offers a selection of options for deciding when all parts are present, based on the predicted fraction passed from the integration program (FRACTIONCALC). Partial reflections will be accepted if their total predicted fraction lies between eg 0.95 and 1.05. these limits may be set depending on your confidence in the fractions, as a compromise between completeness and reliability. Mosflm also passes a flag (MPART) which records Mosflm's calculation of which part is which (eg MPART = 43 means this observation is part 3 of 4). Scala will check these flags for consistency as an alternative to checking the total fraction.

Weak reflections in scaling

The default in scaling is to omit weak reflections (EXCLUDE SDMIN 6). This seems to speed convergence considerably, but may cause problems with very weak data. The default value may need to be changed in the light of experience.

TAILS correction

Many protein crystals show marked diffuse scattering, which is seen as long tails on spots in the "phi" direction, so that reflections often appear on the image before they are predicted. If the mosaicity is increased to include these tails, too many reflections may be rejected as overlaps. Fully-recorded reflections are integrated over a smaller phi width than partials, so more of the tails are chopped off for fulls than for partials. This leads to the typical negative partial bias, with partials systematically larger than equivalent fulls.

A correction has been introduced which attempts to correct for the different truncation of diffuse scattering tails, using a simple model of thermal diffuse scattering, expressed as 2 or 3 parameters over the whole data set. This correction reduces the partial bias substantially, and seems to improve the data generally, though sometimes the parameter refinement can be a little unstable.

Normal probability analysis

Normal probability analysis (see for example D.Smith and L.Howell, J.Appl.Cryst (1992) 25, 81-86: D.Smith, CCP4 Study Weekend (1993) 99-106) compares the normalized deviations (Chihl) with a normal distribution

Chihl = (Ihl - <Iothers>)/sqrt(sigma(Ihl)**2 + sigma(others)**2)

where Ihl is a measured intensity, and <Iothers> is the mean of the other observations of the same or equivalent reflections. If the measured intensities Ihl do indeed follow a normal distribution, and the estimated errors shl are correct, then the Chihl will follow a normal distribution with mean 0 and standard deviation 1. From a sorted list of Chihl, we can predict the expected Chi corresponding to that rank in the list, to give a set of Chiobs, Chicalc pairs. Plotting Chiobs against Chicalc should then give a straight line of slope = 1.0. This plot has a number of useful properties. It shows clearly if the errors do not follow a normal distribution, as is commonly the case. The program splits the Ihl data into classes by "run" and for fully recorded and partial reflections, comparing each to the <Iothers> for all observations, so the if the normal probability plots are different for different classes of reflection, this indicates a systematic difference between the classes.

Correction of standard deviations

Because sd(I) estimates from integration programs are often poor, Scala, like Agrovata compares the observed scatter of multiple observations with their estimated sd, and applies a simple correction model

sd(I)' = Sdfac * sqrt[sd(I)**2 + (Sdadd * I)**2]

Scala now estimates the multiplier factor Sdfac automatically, by making the slope of the central part of the normal probability analysis of the scatter equal to 1.0. Automatic calculation of Sdadd is more difficult, and not done at present.

Future plans

Planned additions to the program include improved handling and analysis of anomalous data, spherical harmonic parameterization of the scaling, and multiplicity-weighted statistics as suggested by Diederichs and Karplus (personal communication).

Availability

This version should be in the next CCP4 distribution, 3.3. In the mean time, it is available in a beta test form from ftp://ftp.mrc-lmb.cam.ac.uk/pub/scala_2.2.2.tar.gz It will probably be number version 2.2.3 by the time of the release.

The BLANC program suite for Protein Crystallography

Alexei A. Vagin[1] [2], Garib N. Murshudov[1] [3], Boris V. Strokopytov[1]

1. Institute of Crystallography, Leninsky pr.59, Moscow 117333, Russia
2. UCMB-ULB, Free University of Brussels, avenue Paul Heger cp160/16 - P2 1050 Brussels, Belgium
3. Chemistry Department, University of York, Heslington, York, U.K.

Dedicated to the memory of academician B.K.Vainshtein

Abstract

The BLANC program suite is a set of programs which can be used for macromolecular structure determination by X-ray crystallography. The suite is designed to provide experienced crystallographers and students with a number of simple tools and at the same time allows to build and test new algorithms. Beside a set of small programs, the BLANC system introduces so-called superprograms which represent larger programs composed of several smaller ones. They utilise so-called black-box principle requiring minimum preparations or intervention from a user. The programs are written in standard Fortran77. They are connected by standard BLANC data files. The package has been ported to all the major platforms such as Unix, VMS and DOS. At the moment a current version of the suite is distributed by anonymous ftp.

Introduction

The BLANC program suite project was started in 1979 in the laboratory headed by B.K.Vainshtein in the Institute of Crystallography, Moscow. The goal of the project was to develop an independent flexible set of programs which could communicate with each other through standard data file formats. The programs can be combined in a many different ways allowing user to perform any particular task. All computer code is written in standard Fortran77. The suite contain programs for analysis and merging of intensity data, structure solution programs utilising SIR, MIR, SIRAS, MIRAS, molecular replacement and density modification methods. The complex also contains programs for crystallographic refinement and the programs for analysis of the structures. The programs for displaying electron-density, rotation function, etc. are also available. The suite has been used for the determination of a number of protein structures. Some examples are listed below (Table 1).

Table 1: Examples of protein structures solved using the BLANC suite

Protein	Reference
Tyrosine phenol-lyase	Antson et al. 1992
Catalase	Vainshtein et al. 1981

Thermitase	Teplyakov et al. 1986
Ribonuclease C2	Polyakov et al. 1988
Ribonuclease Pb1	Pavlovsky et al. 1988
Aspartataminotransferase	Malashkevitch et al. 1995
Pyrophosphatase	Chirgadze et al. 1989
Dehydrogenase	Lamzin et al. 1992

The program suite

Basic conception

The main idea behind the BLANC suite is simplicity. Special attention during development of the program system was paid to make it as user-friendly as possible:

- All programs demand minimal necessary parameters. Most of them have default values.
- It is possible to use programs in dialogue mode or in batch mode. Modern computing technology allows to carry out most of the calculations for small and medium sized proteins in real time, therefore, dialogue is a preferable way of running programs in the BLANC program system. However, each program automatically produces a batch command file during dialogue. This feature might be useful for repeated calculations.
- Program parameter requests are self-explanatory because there are short prompts with explanations.
- If necessary it is possible to use keywords to change certain parameters. Keywords are printed by the programs at the beginning of execution.
- Each program has a short description. You can find it in the program texts or in the BLANC manual.

Most of BLANC programs do not require large memory. Most of them can be run on IBM PC with 640 K memory. All BLANC programs are written in standard FORTRAN codes and can be running at least by MS-DOS, VMS VAX, UNIX. Some of the BLANC programs and superprograms are listed in Table 2 and 3.

Table 2: Main BLANC programs

Program	Function
A. Entrance and exit.	
readPDB	converts coordinates file formats to CIF
TOBLANC	converts structure factors to BLANC format
FROMBL	converts structure factor file into CIF
writePDB	converts coordinates to PDB file
B. Fourier transformation.	
COEF	calculates various kinds of Fourier coefficients
FFT	calculates maps using FFT
RFT	calculates structure factors

C. Look up.	
ISOLINE	draws maps in isolines (Postscript format)
D. Statistics.	
FLSTAT	gives various statistics for structure factors files, etc.
MODCHECK	gives statistics about restraints
E. Scaling.	
SCALE	calculates Wilson plot scale
PSCALE	Patterson origin peak scaling
ANISOSCL	calculates anisothermal scaling of two files
F. Modification, copy and merge.	
MODDEN	density modification program
COPYFL	changes file titles, scale, etc.
CONCRD	modifies coordinate files
JOINFL	merges the files of structure factors or phases
SORTMRG	reads, sorts, averages the files of structure factors or phases
G. Molecular replacement.	
RFCOEF	calculates coefficients of spherical harmonics
RFRES	calculates Rotation Function (Euler angles)
RFROT	calculates rotated spheric coefficients
RFADD	adds spheric coefficients
TRPACK	3D translation/packing/phased translation function
RTRANS	transforms Rotation Function map to polar angles
H. Isomorphous replacement.	
PHASE	calculates Henrickson-Lattman coefficients for a derivative
REFINE	heavy atom's full matrix refinement
J. Refinement.	
ROTLSQ	rigid body refinement
K. Others.	
GENDEN	generates electron density
PEAKSRCH	map peak search
WATPEAKS	water peak search and water replacing
FIT	superimposes two sets of coordinates
ABCDPH	phases from Hendrickson-Lattman coefficients
PHABCD	Hendrickson-Lattman coefficients
HISTOGRM	histogram matching

SURFACE	solvent accessible surface area
FRAGSRCH	builds full atomic model of a protein using C_alpha atom coordinates
CONTACT	computes inter or/and intra molecular contacts
L. Not converted to current version yet.	
SEQSRCH	searches aminoacid sequence in the local Sequence Data Bank.
ALIGN	aligns aminoacid sequences
BBONE	inserts side chains of a protein into electron density map
PATLSQ	refines orientation of a model before translation function search
GROUP	converts scattering from protein atoms to group scattering factors
LOCSCL	anisothermal local scaling
DPLOT	draws PostScript stereo picture of the model with electron density
SKELETON	density skeletonisation procedure

Table 3: BLANC superprograms

Program	Function
A. Isomorphous replacement method:	
MIR	automated heavy atom search and phasing
SIR	automated one derivative heavy atom search and phasing
PATTSRCH	automated reciprocal heavy atom structure solution
B. Molecular replacement method:	
MOLREP	performs automated molecular replacement search
SELFROT	calculates self-rotation function
CROSSROT	calculates cross-rotation function
TRFUN	calculates translation and packing function
C. Refinement:	
MMM	Macro Molecular Minimisation/Crystallographic refinement
MAKECIF	creates list of geometric and energetic parameters
LIBCHECK	reads library of monomers, performs various checks

EMIN	performs energy minimisation
DENMOD	phase refinement by density modification
D. Others:	
OMIT	calculates omit synthesis phases
OMIT_MAP	creates global omit map
SFCHECK	checks quality of X-ray structures

Libraries

BLANC maintains a library of subroutines for performing the basic crystallographic and programming operations. Common subroutines, e.g., to open and close data files, read and write data, FFT, matrix operations, etc. are gathered in a special library (LIBUTILS). This shortens markedly program code and makes it easy to read and modify the programs. Each program has a subroutine version gathered in another library (LIBSUBR). This allows a programmer to develop larger programs composed of smaller ones.

Three levels of programming in BLANC. Introduction of superprograms

There are three main levels of programming modules in the BLANC suite of programs. The first level is superprograms. The superprograms normally implement some method (e.g., molecular replacement using known model). Some programs may act like subroutines inside superprogram. On the second level we have usual crystallographic programs which perform basic operations like calculation of structure factors, electron density etc. They use subroutines from the library. The subroutines themselves constitute a third level. The main goal of this programming level is to solve local tasks only: matrix operation, FFT, opening and closing files, etc. These special arrangement of BLANC programs simplifies significantly development of new programs.

Original features of BLANC

BLANC contains a set of new original algorithms and programs developed independently by us. Among them algorithms for calculation of translation and packing function (Vagin, A.A., 1983; Vagin, A.A. 1989), new program for data scaling using Patterson origin peak (to be be published elsewhere), program for black-box molecular replacement, black-box heavy-atom search and phasing, global omit map program (Vagin, A.A., unpublished results) and others.

File formats

There are four main types of file format for reflection data, map data, coordinate data and graphics meta-files. The coordinate data files are in ASCII but reflection and map files are binary. The BLANC reflection files in most cases uses 12 bytes of disk memory per reflection. Three reflection indices are packed into one integer*4. Two real numbers are used for storing information about amplitude and error estimate (sigma). The header records contain information such as cell dimensions and symmetry operators. The reflection data are stored notionally as columns of real numbers. There is no need to mark columns by special labels since native, derivative and calculated data are always kept in separate structure factor files. Maps are

stored in a binary sequential access files as a three dimensional array preceded by a suitable header which contains information about map dimensions, cell, symmetry information, maximum and minimum, mean and root-mean-square deviation density values, etc. Each density grid point is packed into two byte integer. There is a possibility to convert BLANC map format to other map file formats for use on graphical devices. The standard coordinate file format is close to mmCIF format (Bourne et al., 1996). The program suite allows conversion from BLANC/mmCIF format to the PDB (Bernstein et al., 1977) format and vice versa. Graphical programs produce output in PostScript format.

Documentation, Installation and Distribution

The BLANC manual gives the details of installation procedures. In order to run the programs certain environment variables need to be set to appropriate values. Output document files are produced which contain necessary information about the progress of each particular run of the program.

The program suite has been implemented on a large number of hardware platforms including Unix. Installation is straightforward and full instructions are given in the BLANC manual.

The BLANC program suite is licensed free to academic institutes. The programs may be obtained by Internet ftp from anonymous@ftp.ucmb.ulb.ac.be. (First read file: pub/alexei/blanc/README). Several programs and superprograms independent from the BLANC suite (SFCHECK, MOLREP, CONTACT, MAKECIF, EMIN, LIBCHECK etc.) are kept in separate directories at the anonymous ftp site. Separate arrangements can be made for commercial organisations. For further details contact Dr.A.Vagin (email: alexei@ucmbcx1.ulb.ac.be).

Acknowledgements

We are very grateful to all our former colleagues who made significant contributions to this project helping us to eliminate bugs in the programs. We thank them for numerous scientific discussions as well.

References

- Antson, A.A., Strokopytov, B.V., Murshudov, G.N., Demidkina, T.V., Fogelman, H.K., Paskhina, O.G., Hennig, M., Nekrasov, Yu.V., Popov, A.N., Rubinsky, S.V., Harutyunyan, E.H. (1992) Three-dimensional structure of tyrosine phenollyase at 4.5 A resolution *Kristallografiya* **37**, 82-89.
- Bernstein, F.C., Koetzle, T.F., Williams, G.J.B., Mayer, E.F., Brice, M.D., Rogers, J.R., Kennard, O., Shimanouchi, T. and Tasumi, M. (1977). The Protein Data Bank: a computer-based archival file for molecular structures. *J.Mol.Biol.* **112**, 535-542.
- Bourne, P.E., Berman, H.M., McMahon, B., Watenpaugh, K., Westbrook, J., No.1, Fitzgerald, P.M.D. "The Macromolecular CIF Dictionary (mmCIF)".(1996) In: *Methods in Enzymology* in press
- Chirgadze, N.Yu., Kuranova, I.P., Strokopytov, B.V., Harutyunyan, E.G., Hohne, W. (1989). Crystal structure determination of MnP-complex of

neorganic pyrophosphotase from yeast using the molecular replacement method at 2.7 A resolution. *Krystallografiya* **34**, 1446-1450.

- Lamzin, V.S., Aleshin, A.E., Strokopytov, B.V., Yukhnevich, M.G., Popov, V.O., Harytyunyan, E.H., Wilson, K.S. (1992). Crystal structure of NAD-dependent formate dehydrogenase. *Eur.J.Biochem.* **206**, 441-452.
- Malashkevich, V.N., Strokopytov, B.V., Borisov, V.V., Dauter, Z., Wilson, K.S. and Torchinsky, Yu.M. (1995). Crystal structure of the closed form of chicken cytosolic aspatate aminotransferase at 1.9 A resolution. *J.Mol.Biol.* **247**, 111-124.
- Pavlovsky, A.G., Polyakov, K.M., Borisova, S.N., Strokopytov, B.V., Vagin, A.A., Vainshtein, B.K. (1987). Structural bases for nucleotide recognition by guanil-specific ribonucleases. In: *Proceedings of the 6th International symposium on Metabolism and Enzymology of Nucleic Acids*, (Eds. Zelinka, J., and Balan, J.). Bratislava, 323-330.
- Polyakov, K.M., Strokopytov,B.V., Vagin, A.A., Bezborodova, S.I., Orna, L. (1988). Three-dimensional structure of RNase C2 from Aspergillus clavatus at 1.35 A resolution. In: *Proceedings of the 6th International Symposium on Metabolism and Enzymology of Nucleic Acids* (Eds. Zelinka, J. and Balan, J.). 227-231. Plenum Press, New York - London.
- Teplyakov,A.V., Strokopytov,B.V., Kuranova,I.P., Popov, A.N., Harutyunyan, E.H., Vainshtein, B.K., Froemmel, C., Hoehne,W. (1986). X-ray study of thermitase at 2.5 A resolution. *Kristallografiya* **31**, 931-936.
- Vagin, A.A. Ph.D. Thesis, Institute of Crystallography, Moscow, (1983)
- Vagin, A.A. New translation and packing functions., Newsletter on protein crystallography., Daresbury Laboratory, (1989) **24**,
- Vainshtein, B.K., Melik-Adamyan, W.R., Barynin,V.V., Vagin, A.A., Grebenko, A.I. Three-dimensional structure of the enzyme catalase., (1981), *Nature* **293**, p.411-412.

Simplified error estimation *a la* Cruickshank in macromolecular crystallography

Garib N. Murshudov and Eleanor J. Dodson

Chemistry Department, University of York, Heslington, York, U.K.

---

1. Introduction

An important part of protein crystallography is refining the fit of model parameters to experimental data. It is intuitively obvious that a model will be more reliable if there are more observations to fit it to, and that some parts of a macromolecular model are more accurately described than others. It is not always easy to parameterise this, but without giving some estimate of the reliability of the model parameters the refinement procedure cannot be complete.

When parameters are estimated by least-squares or maximum likelihood methods their reliability can be estimated from the inverse of the matrix of the second derivatives (see for example Stuart & Ord, 1991). Figure 1. However it is extremely time consuming to both generate and to invert the matrix of second derivatives for many parameters. To our knowledge the only refinement program which has an option to do this, and thus give standard uncertainties of parameters is SHELXL (Sheldrick 1995).

As a community we have been extremely lucky to have interested Durward Cruickshank in this problem. He was instrumental in developing much of the solid theoretical basis for the refinement of small molecules during the 50s and 60s, and has recently addressed the special problems of macromolecules, where there is less reliable data, the range of precision within any given structure is much greater, and the computing problems are still formidable. He points out that protein crystallographers often use somewhat misleading methods to estimate reliability (Cruickshank 1996). One of them is to use the Luzzati plot to assign an overall average error for atomic coordinates. But Luzzati's classic paper (1952) describes the probability distribution of structure factors and does not claim to indicate the reliability of parameters. It is dependent on weights used in refinement. The sigmaA plot described by Read (1986) is also based on a similar distribution. A second method is to use B-values as an indicator of the reliability of atomic positions. As expected, it is easy to demonstrate that there is a relationship between B-value and estimated standard uncertainties (e.s.u.) of atoms but it is important to remember that the B-value is an estimation of atomic mobility but not its reliability.

However approximate standard uncertainties can be obtained from the diagonal terms alone of the second derivative matrix. These can be estimated during the

course of refinement, and it is trivial to carry out the matrix inversion of a diagonal matrix. (Murshudov, Vagin & Dodson, 1997):

$$H_1(x_{ni}, x_{nj}) = 2\pi^2 \sum \left(\frac{\partial^2 f}{\partial A_h^2} + \frac{\partial^2 f}{\partial B_h^2}\right) h_i h_j f_n^2 \qquad (1)$$

where f is residual used for refinement (it could be least-squares or maximum likelihood residuals), A_h and B_h are real and imaginary parts of structure factor, f_n is atomic form factor, x_n is positional parameter.

For B-values:

$$H_1(B_n, B_n) = \frac{1}{32} \sum \left(\frac{\partial^2 f}{\partial A_h^2} + \frac{\partial^2 f}{\partial B_h^2}\right) |s|^4 f_n^2 \qquad (2)$$

B_n is atomic B-value.

In the following sections we will give Cruickshank's equation for a dispersion precision indicator (DPI) and its modification to utilise R_{free}, and extend them to give some simple equations for DPIs corresponding to approximate e.s.u.-s of the individual atomic coordinates and B-values. The equations for B-value dependent e.s.u. are similar to the equations given by Cruickshank (1949a, 1949b)

2. Overall standard uncertainties based on R-value and free R-value

Using equation (1) for orthogonal coordinates and making several simplifications Cruickshank (1960, 1996) gives following equation for an overall dispersion precision indicator (DPI):

$$\sigma^2(x) = 0.65 \frac{N_a}{N_o - N_p} R_{conv}^2 d_{min}^2 C^{-\frac{1}{3}} \qquad (3)$$

where C is completeness, R_{conv} is conventional R-value, d_{min} maximum resolution, N_a number of atoms included in refinement, N_o is number of observations, N_p is number of parameters refined.

He suggests replacing the factor 0.65 by 1.0 as a matter of caution since in the derivation of above equation only diagonal terms of the second derivative matrix are used. This equation does not take into account the effect of geometric restraints and cannot be used at low resolution when N_o - N_p is negative.

If we will assume that R_{free} is the expected value of R and use the relation between them suggested by Cruickshank during the Refinement Workshop reported in Dodson, Kleywegt, Wilson (1996).

$$< R_{expected} > = [N_o/(N_o - N_p)]^{1/2} R_{conv} = R_{free} \qquad (4)$$

then we can base DPI on R_{free}

$$\sigma^2_{free}(x) = 0.65 \frac{N_a}{N_o} R^2_{free} d^2_{min} C^{-\frac{1}{3}} \qquad (5)$$

Since R_{free} is dependent to some extent on the information about restraints and on the parameterisation used for refinement the equations should be meaningful in all cases (it does not matter if you have refined with or without NCS, isotropic anisotropic or overall B-value). But the equations can only give overall DPI, and cannot indicate the relative precision of different parts of a structure.

To test the agreement between equations (3) and (4) we used catalase from *Micrococcus lysodeikticus* refined at three different resolutions 1.5, 1.83 and 1.96A (Murshudov *et al* 1997). For the structure refined at 1.5A R_{conv} and R_{free} are 11.7 and 14.0%, suggesting DPI-s of 0.045A and 0.048A respectively. For the structure refined at 1.83A R_{conv} and R_{free} are 11.8 and 15.0 % giving DPI-s of 0.082A and 0.086A, while for the 1.96A structure R_{conv} and R_{free} are 16.7 % and 22.7 % with DPI-s of 0.143A and 0.147A respectively. This close agreement show that at medium and high resolution DPI-s can be derived from the R_{free} values quite accurately.

Both these equation can only be used sensibly at the end of refinement, when the parameter is near its minimum value, (see Figure 1) and with the assumption that the model is complete. The DPIs are nonsense initially. To demonstrate this: take the extreme case. If the data is complete to 1.5A resolution, but the model consists of random atoms N_a / N_o approx 0.05, R_{free} approx 0.58, and hence DPI approx 0.13 which clearly is not a measure of the precision of the positional parameters.

3. Approximation to standard uncertainties of individual atomic parameters

Using equation (1) and the approximation:

$$f^2_n(s) \approx \frac{\Sigma_e}{N_e} \frac{z^2_n}{< z^2 >} e^{-\Delta B |s|^2/2} \qquad (6)$$

then for the B-value dependent e.s.u we can write:

$$\sigma^2_B(x_u) = \frac{8}{2\pi^2} \frac{< z^2 >}{z^2_n} \frac{N_a}{N_o - N_p} \frac{\sum_u (|F_o| - |F_c|)^2}{\sum_u \Sigma_c s^2 e^{-\Delta B s^2/2}} \qquad (7)$$

where <z²> is average of square of number of electrons, z_n² is square of number of electrons for given atom, delta-B is difference between the Wilson and this atom's B-value, N_o, N_p are defined above, s is reciprocal space vector, F_o and F_c observed and calculated amplitudes of structure factors, Sigma_c the normalisation factor for calculated structure factors, sum_u the summation over the reflections included in refinement.

To avoid negative differences between N_o-N_p we can replace sum_u(|F_o|-|F_c|)²/(N_o-N_p) by sum_f(F_o-F_c)²/N_{free}:

$$\sigma^2_B(x_n) = \frac{3}{2\pi^2} \frac{< z^2 >}{z_n^2} \frac{N_a}{N_{free}} \frac{\sum_f (|F_o| - |F_c|)^2}{\sum_n \Sigma_c s^2 e^{-\Delta B s^2/2}} \qquad (8)$$

where N_{free} is number of `free' reflections and sum_f is the summation over these.

The same approach could be used for approximate e.s.u. of B-values

$$\sigma^2_B(B_n) = 2\frac{< z^2 >}{z_n^2} \frac{N_a}{N_o - N_p} \frac{\sum_n (|F_o| - |F_c|)^2}{\sum_n \Sigma_c s^4 e^{-\Delta B s^2/2}} \qquad (9)$$

or:

$$\sigma^2_B(B_n) = 2\frac{< z^2 >}{z_n^2} \frac{N_a}{N_{free}} \frac{\sum_f (|F_o| - |F_c|)^2}{\sum_n \Sigma_c s^4 e^{-\Delta B s^2/2}} \qquad (10)$$

Again these equations should be used only at the end stages of refinement, and then Sigma_c could be replaced by Sigma_o and even by |F_o|.

These equations show that the e.s.u. of both positional and thermal parameters will depend on completeness of data, which is expressed through the summation, on the B-value of the atom, and on the agreement between observed and calculated structure factors. More reliable values may be obtained by using a weighted sum over the reflections. Equation (7) and (9) can only be used at high resolution, but equation (8) and (10) could be used at any resolution since they do not involve N_o-N_p. Moreover since equations (8) and (10) use only the agreement of the `free' reflections, the effect of restraints will be incorporated in the estimate.

Note that these approximations are very rough. They could be improved but the effect of the unconsidered non-diagonal terms is expected to be much larger than the effect of approximations and these equations can be used for qualitive reliability assessment.

Again we used the catalase structures for testing. Figure 2-4 shows B-value dependence of the e.s.u. for the positional and thermal parameters. At 1.5A resolution the e.s.u. based on `used' and `free' reflections are very close to each other. At lower resolution this is not so, probably because the `free' reflections contain information about restraints whereas `used' reflection do not know about them.

4. Likelihood based DPI

Using maximum likelihood equations (Murshudov, Vagin, Dodson 1997) instead of least-squares then we can write:

$$\sigma^2_B(x) \approx \frac{3}{8\pi^2} \frac{< z^2 >}{z_n^2} \frac{N_a}{\sum_n (\frac{1}{\Sigma} - \frac{E_c^2}{\Sigma^2}(1 - m^2)) \sigma_A^2 s^2 e^{-\Delta B s^2/2}} \qquad (11)$$

where <z²>, z_i², sum_u, delta-B, s are defined in equation (5), Sigma = sigma_{E;exp}² + epsilon (1-sigma_A²), sigma_{E;exp} is the experimental uncertainty of the normalised structure factor, E_o is the normalised observed amplitude of structure factor, m is figure of merit, sigma_A = sqrt{Sigma_c / Sigma_o} D, D=<cos(2pi s delta-x)> , delta-x is error in positional parameters, Sigma_o and Sigma_c are normalisation factors for the observed and calculated structure factors.

And:

$$\sigma_B^2(B_n) \approx 8 \frac{< z^4 >}{z_n^2} \frac{N_a}{\sum_u (\frac{1}{\Sigma} - \frac{m^2}{\Sigma^2}(1 - m^2)) \sigma_A^2 s^4 e^{-\Delta B s^2/2}} \qquad (12)$$

These equations show that e.s.u. of atomic parameters depend on completeness, resolution and quality of the data, the completeness and quality of the model and the remaining phase error.

If we replace in equation (11 - 12) B_n with B_{Wilson} and z_n² with <z²> we can get the e.s.u. for an `average' atom in the structure. In principle equations (11 - 12) could be used at any stage of refinement but the derivation used only the diagonal terms of second derivative matrix. Especially in the early stages of refinement off diagonal terms which reflect the interaction between different parameters may also be large.

5. Conclusions

1. There is dependence of e.s.u. on B-values as expected, but it is not as simple as substituting the e.s.u. as sqrt{B/8pi²}}
2. There is dependence of e.s.u. on resolution, as expected
3. There is dependence of e.s.u. on completeness of data, as expected
4. There is dependence of e.s.u. on completeness of model, as expected
5. There is dependence of e.s.u. on the quality of data but inclusion of weak data could still improve quality of model. (Sometimes weak data are better than no data)

All equations given here use only diagonal terms of second derivative matrix therefore will give better approximation at high resolution and at the end stages of refinement. These equations do not use restraints.

References

1. Dodson, E.J., Kleywegt, G.J. & Wilson, K. (1996) *Acta Crsyt.* **D52** 228-234
2. Cruickshank, D.W.J. (1949a) *Acta Cryst.* **2** 65-82
3. Cruickshank, D.W.J. (1949b) *Acta Cryst.* **2** 154-157
4. Cruickshank, D.W.J. (1960) *Acta Cryst.* **13** 774-777
5. Cruickshank, D.W.J. (1996) in the *Refinement of Macromolecular structures* Proceedings of CCP4 Study weekend. pp 11-22
6. Luzzati, V. (1952) *Acta Cryst.* **5**, 802-810
7. Murshudov, G.N., Vagin A.A. & Dodson, E.J. (1997) *Acta Cryst.* **D53** in press

8. Murshudov, G.N., Grebenko, A.I., Brannigan, J.A., Antson, A.A., Barynin, V.V., Dauter, Z., Wilson, K.S. & Melik-Adamyan, W.R. (1997) *J.Mol.Biol.* in press
9. Read, R.J. (1986) *Acta Cryst.* **A42**, 140-149
10. Sheldrick, G.M. (1995) *SHELXL-93, a Program for the Refinement of Crystal Structures from Diffraction Data.* Institut fuer Anorg.Chemie, Goettingenm Germany.
11. Stuart, A & Ord, K.J. (1991) *Kendall's Advanced Theory of Statistics.* Vol. **2** 5th ed. London, Melbourne, Auckland: Edward Arnold

---

Figure 1: The parameter for both these distributions has its minimum at 0. The solution for the distribution with second derivative of 2, is more sharply defined than the that with second derivative of 1.

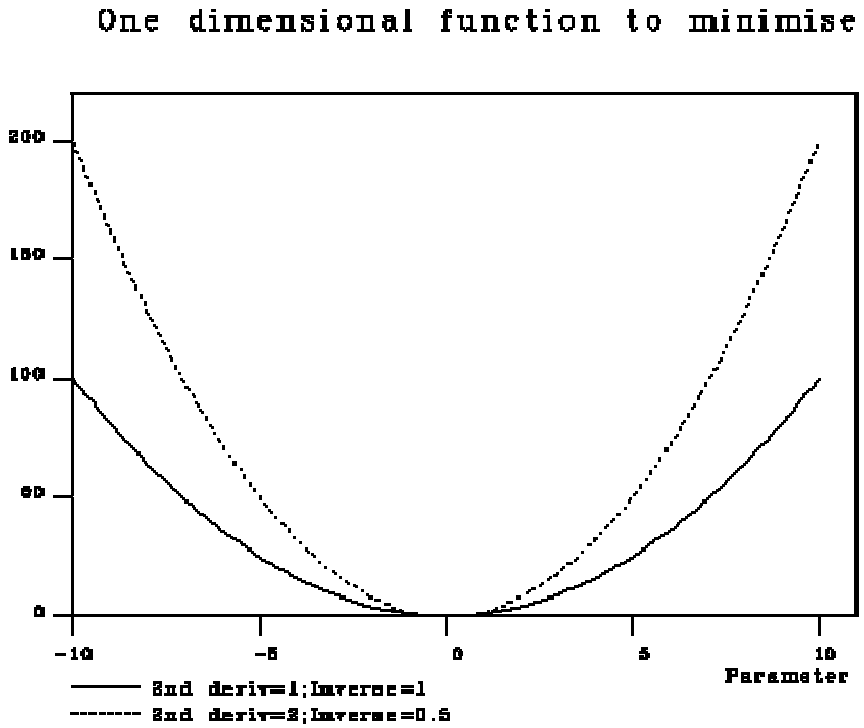


Figure 2: B-value dependence of e.s.u. at 1.5A resolution. Dashed lines correspond e.s.u. derived using agreement of `free' reflections, solid lines show e.s.u. derived using agreement of reflections included in refinement. a) e.s.u. for positional parameters. b) e.s.u. for B-values.

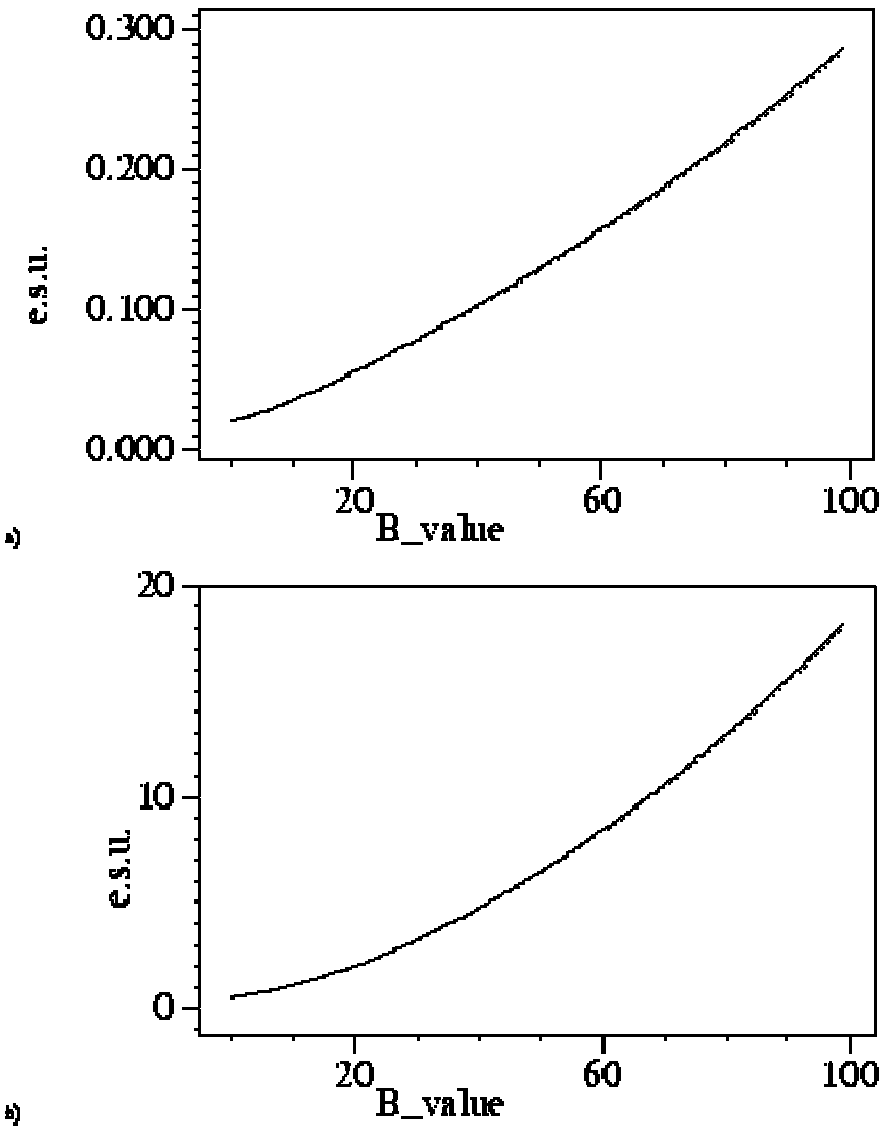


Figure 3: B-value dependence of e.s.u. at 1.83A resolution. Dashed lines correspond e.s.u. derived using agreement of `free' reflections, solid lines show e.s.u. derived using agreement of reflections included in refinement. a) e.s.u. for positional parameters. b) e.s.u. for B-values.

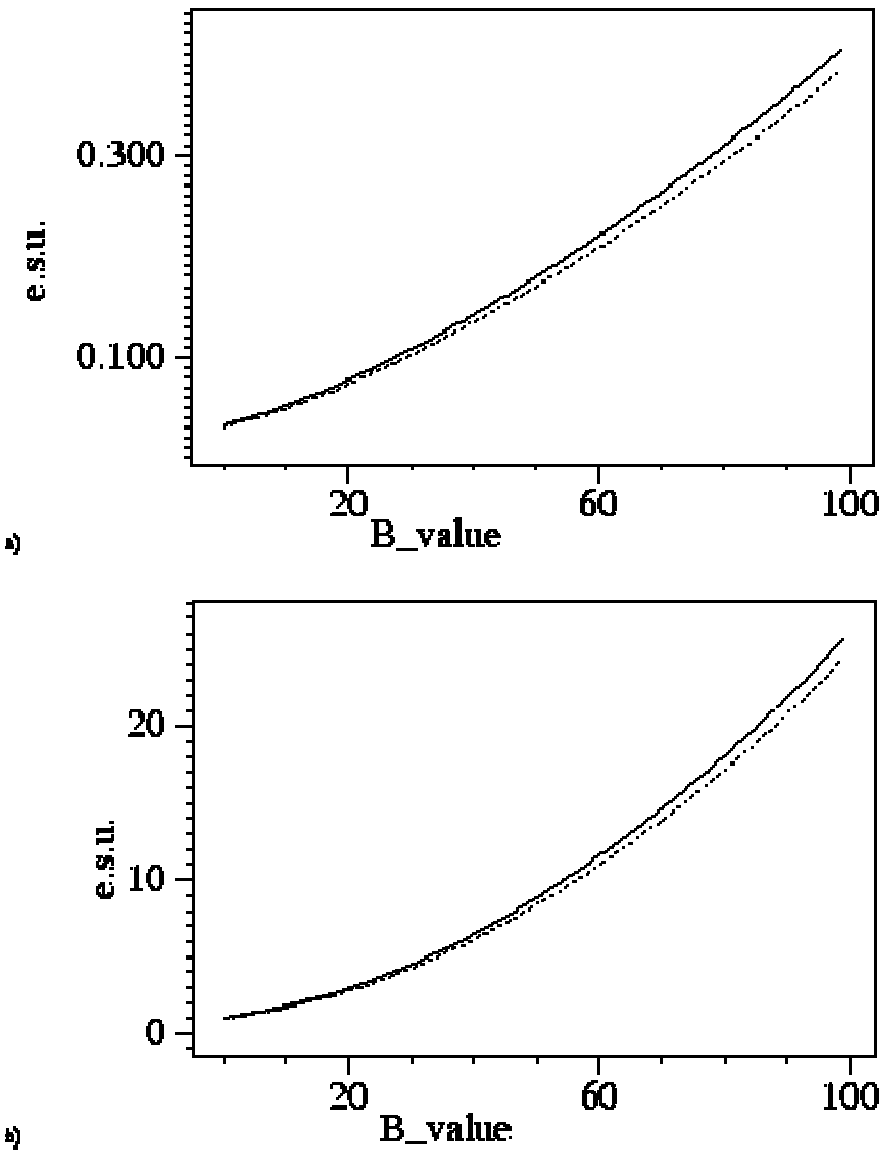
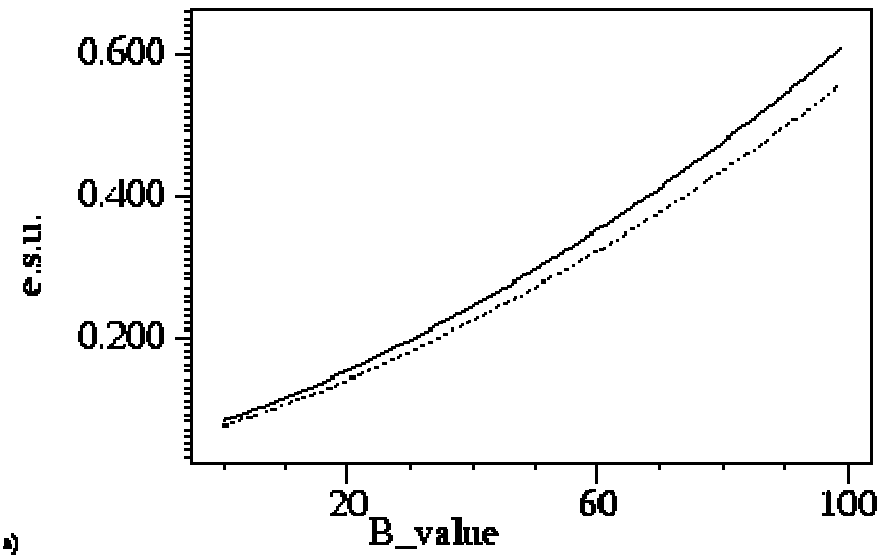
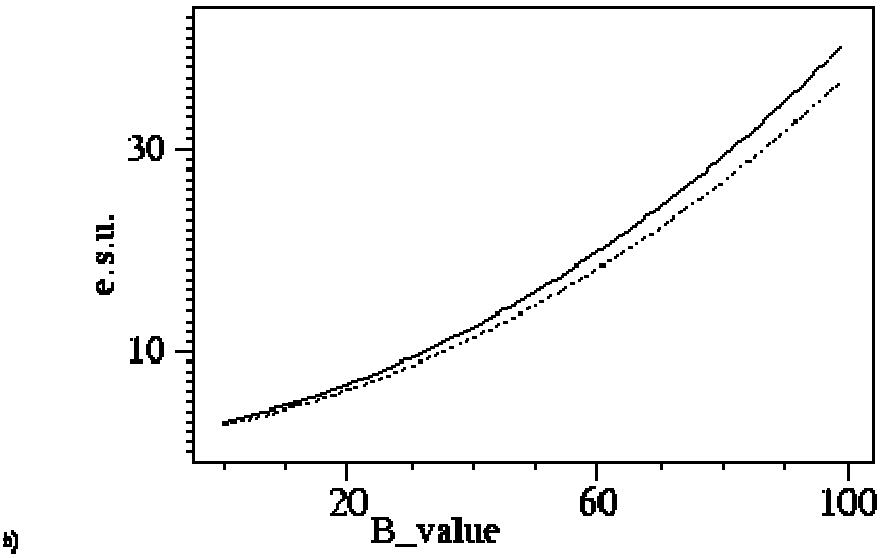


Figure 4: B-value dependence of e.s.u. at 1.96A resolution. Dashed lines correspond e.s.u. derived using agreement of `free' reflections, solid lines show e.s.u. derived using agreement of reflections included in refinement. a) e.s.u. for positional parameters. b) e.s.u. for B-values.



a)



b)

Use of mini-rotation frames for image plate data collection

P. A. Tucker

Structural Biology (including Biocomputing) Programme, EMBL,
Meyerhofstrasse 1, D69012 Heidelberg

The standard method of using an image plate scanner is to record diffraction patterns over rotation ranges greatly in excess of the reflection rocking curve width. In this way one maximizes the density of information on a single image and reduces data collection and processing overheads. The disadvantage of this procedure is clear, namely that the signal to noise ratio must be worse (extra background is accumulating under the reflections when they are no longer in the reflecting condition). There is nothing to prevent one collecting mini-rotation frames on image plate scanners and perhaps one should consider more carefully whether or not to do this, especially as instrument duty cycles become shorter. That the use of mini-rotation frames with single-photon counters is advantageous is well established experimentally (W.Kabsch, Proceedings of the CCP4 Study Weekend on Data Collection and Processing, 1993, 63-70). For a MAR image plate, if one takes 0.25° frames and sums them to give 1° frames the data is better when the narrower frames are used for processing (P.A.Tucker, Joint CCP4 and ESF-EACMB Newsletter on Protein Crystallography. Number 28 (May 1993), 74-76). This experiment ignored the fact that for any analog detector (like an image plate) there must be some readout noise per image which, because this is summed as well, gives a poorer signal to noise ratio for the wider images than is realistic. To do a better experiment is simple, using a small (0.25 x 0.25 x 0.1 mm) crystal of tetragonal hen egg white lysozyme, three equivalent data sets were collected on a MAR image plate scanner using CuKa radiation. All data sets were collected over the same 45° under identical conditions except for the frame width and time. The first and last sets had frame widths of 0.2° measured over 40s. The second set had a frame width of 1.5° measured for 300s. All data sets were processed with XDS using parameterisation for the beam crossfire and crystal mosaicity empirically determined from the first data set. The results are summarized in the Table below and show clearly that narrow frame widths yield better data. Note that the smaller number of processed reflections for the second data set results from excluding reflections that were partially overlapped.

	Set 1	**Set 2**	**Set 3**
Frame width	0.2°	1.5°	0.2°
Integrated reflections	22841	18075	22847
Outliers rejected	225	1025	279
Unique reflections	6370	6048	6371
Rsym as a function of resolution			
15.0-6.32	3.3	2.9	2.9

6.32-4.47	2.9	3.4	3.0
4.47-3.65	2.8	3.1	2.8
3.65-3.16	3.0	3.9	2.9
3.16-2.83	3.3	4.4	3.2
2.83-2.58	3.8	5.3	3.9
2.58-2.39	4.7	6.6	4.7
2.39-2.24	4.9	7.1	5.0
2.24-2.10	6.0	9.5	6.0
% of reflections with I/s(I) > 12			
15.0-6.32	91.3	89.6	91.6
6.32-4.47	94.3	91.3	94.4
4.47-3.65	95.3	92.6	95.4
3.65-3.16	91.4	86.7	91.0
3.16-2.83	87.2	78.8	86.9
2.83-2.58	79.4	71.8	79.1
2.58-2.39	68.9	55.7	68.7
2.39-2.24	61.3	46.9	60.7
2.24-2.10	52.7	36.5	51.1