

A Collaborative Computational Project, number 4:

Providing Programs for Protein Crystallography

Eleanor J. Dodson

University of York, Heslington, York YO1 5DD, UK.

ccp4@dl.ac.uk

<http://www.dl.ac.uk/CCP/CCP4/main.html>

Abstract

The CCP4 (Collaborative Computational Project, number 4) aims to provide first a state-of-the-art suite consisting of a collection of programs plus associated data and subroutine libraries for the determination of macromolecular structure by X-ray crystallography. The programs are from a wide variety of sources but all use agreed standard data file formats. The suite is designed to be flexible, allowing users a number of methods of achieving their aims and so there may be more than one program to cover each function. The package has been ported to all the major platforms under both Unix and VMS and is freely distributed to academics by anonymous FTP from Daresbury Laboratory. It is widely used throughout the world. Secondly the Project has a responsibility to provide support, both in installing, documenting and maintaining the suite, and in educating budding crystallographers in methodological and computing techniques.

key words: CCP4 / program suite / X-ray / macromolecular crystallography

1 Introduction

CCP4 (Collaborative Computational Project, number 4 - see CCP4 (1994)) was established in 1979 by a group of lonely protein crystallographers. In the 1960s macromolecular X-ray crystallography in Britain was concentrated at the Laboratory of Molecular Biology in Cambridge, and the Biophysics Department in Oxford, and the program developers had been members of large groups where ideas were debated and tested on a variety of problems, and there was good computing support. By 1979 the number of institutions doing macromolecular structure determination had increased, and the size of the new groups was often much smaller. This new community needed some structured approach to help maintain and extend an adequate set of software, to discuss installation problems, new algorithms and bug fixes, and to educate students in the field. The project started in the UK, modestly funded by the Science and Engineering Research Council (SERC). Now its funding, currently from the Biotechnology and Biological Sciences Research Council (BBSRC) with contributions from industrial companies, has grown to allow the employment of three people based at Daresbury, and to cover post doctoral positions and short term contracts for individuals who are interested in tackling specific perceived problems. Collaboration on the development of the suite was previously extended into Europe under the auspices of the European Science Foundation (ESF) Network of the European Association of the Crystallography of Biological Macromolecules (EACBM).

1.1 Support

The initial impetus to set up CCP4 was the panic a group of us felt when we realised how alone one was as the only computing enthusiast in a small group, with the biochemists wanting their structure NOW, and infuriated by the failure of the software to deliver. The initial project provided funds to allow the programmers from all the UK groups to meet every three months, and to run an annual two day meeting to address some specific issue. The quarterly meetings allowed members to discuss future developments, to analyse bugs, and to keep in touch in pre Email days.

The annual meetings have become valuable both for their information content, for the opportunity they give for members from different laboratories to meet, and for the published proceedings, which are often one of the most up-to-date texts available on the chosen subject. Some of the meetings held to-date are:

- (1980) Refinement of protein structures. DL/SCI/R16
- (1985) Molecular replacement. DL/SCI/R23
- (1987) Computational aspects of protein crystal data analysis. DL/SCI/R25

- (1988) Improving protein phases. DL/SCI/R26
- (1989) Molecular simulation and protein crystallography. DL/SCI/R27
- (1990) Accuracy and reliability of macromolecular crystal structures.
- (1991) Isomorphous replacement and anomalous scattering. DL/SCI/R32
- (1992) Molecular replacement. DL/SCI/R33
- (1993) Data Collection and Processing. DL/SCI/R34
- (1994) From First Map to Final Model. DL/SCI/R35
- (1995) Making the Most of your Model. DL-CONF-95-001
- (1996) Macromolecular Refinement.

As funding has increased and the Project has acquired paid staff much more support has been offered. Now there is a comprehensive Manual, a simple installation procedure for both Unix and VMS machines is provided, there are man pages, documentation, and example procedures, and an active Bulletin Board to provide help and advice to users, and to stimulate discussion.

2 Philosophy of the CCP4 suite

Unlike many other packages, the CCP4 suite is designed to be loosely organised, so that it is very easy for different developers to add new programs or to modify existing ones without upsetting other parts of the suite. It consists of a set of separate programs which communicate via standard data files, rather than having all operations integrated into one huge program. This is the approach successfully taken by Unix, and now apparently being embraced by some of the large commercial software houses. It has some disadvantages in that tasks often require a script to chain together several programs, e.g. to calculate a difference map to cover a molecule, it is necessary to generate structure factors (`sfall`), do a fast Fourier transform (`fft`), and extend the asymmetric unit to cover the molecule (`mapmaskOR extend`). This means that in some cases information from one program needs to be transferred to the next by hand, and initially the programs were less consistent with each other. In recent years a lot of work has been done to improve the consistency and to simplify the input both by assigning sensible defaults and by using standard keywords for input.

Converting a program to use the standard CCP4 file formats is generally straightforward, and the philosophy of the collection has been to be inclusive, so that several programs may be available to do the same task. The components of the whole system are thus a collection of programs using a standard subroutine library to access standard format files, a set of examples scripts and documentation available for both the VMS and Unix operating systems. Most of the programs are written in standard Fortran—currently the obsolete FORTRAN77 version, but some are in Ansi C.

Briefly, the suite contains programs covering all aspects of macromolecular crystallography from data processing to analysis of a refined model; for example the reduction and analysis of intensity data, structure solution by isomorphous replacement and molecular replacement, and refinement and analysis of the structure. There are also many utility programs for converting formats, etc.

3 File Formats

Users do not usually require detailed information about the format of reflexion, map and coordinate files since libraries are provided for reading and writing them. Crystallographic and book-keeping information is stored in the headers of reflexion and map files to facilitate their use.

The reflection and map file formats are binary. There are two basic reasons for this:

- Clear text files are substantially larger and consequently take more time to read from disc even if the decoding is relatively cheap and space is not a consideration.
- With clear text accuracy is lost on reading and writing, particularly on repeated i/o.

3.1 Labelled Column Reflection Data Files (MTZ)

The MTZ reflection file format (renamed from LCF for three of its progenitors, Sandra McLaughlin, Howard Terry, and Jan Zelinka) uses fixed length records for each reflection with a minimum of 4 columns (H K L plus at least one data column) and currently a maximum of 200 columns of data per reflection record. The columns of the reflection data records are

identified by alphanumeric labels and column type flags held as part of the file header information. The user relates the item names used by the program to the required data columns, identifying them by their labels, by means of assignment statements in the program control data. The programs check to see the associated column type is valid for the program operation, e.g. that a phase is not being assigned to a standard deviation. (This may bring to mind 'tables' or 'relations' in relational databases - intentionally so.) Definitions of acceptable types, and a list of common program labels are given in Figure 1. Additional crystallographic information (title, cell dimensions, column labels, symmetry information, resolution range, history information and, if necessary, batch titles and orientation data) is contained in header records identified by keywords.

Program Label

Type

Description

H, K, L

H

Miller indices.

M/ISYM

BATCH

Y

B

Partiality flag and symmetry number Batch number.

I SIGI

J Q

Intensity I . sI (standard deviation).

FRACTIONCALC

R

Calculated partial fraction of intensity.

IMEAN SIGIMEAN

J Q

Mean intensity. sI_{mean} .

FP FC FPH n

F F F

Native F value Calculated F . F value for derivative n .

DP

DPH n

D

D

Anomalous difference for native data (F+ - F-). Anomalous difference for derivative *n*.

SIGFP SIGDP SIGFPH_n SIGDPH_n

Q Q Q Q

sFP (standard deviation) sDP sF_n sDPH_n

PHIC PHIB

P P

Calculated phase. Phase.

FOM WT

W W

figure of merit weight

HLA, HLB, HLC, HLD

A

ABCD H/L coefficients

FreeR-flag Miscellaneous

I R or I

free *R* flag (as flag label) Any attribute you require

Figure 1: MTZ standard program labels and column types

The model for an MTZ file is thus based on two components, one (the header) keyed on keywords such as SYMMETRY, CELL, etc. and the other (comprising the reflections) keyed on the H, K and L attributes/columns. An example helps to make this clear. A reflection file in the CCP4 examples area contains observations for the dendrotoxin from green mamba (tox_d, Skarzynski (1992)). The file contains the native data plus three derivative data sets, one with anomalous measurements. The derivatives are Hg, I and Au. The labels and column types are:

```
H K L FTOXD3 SIGFTOXD3 ( indices, native F and sd)
H H H F Q ( column type flags )
FMM11 SIGFMM11 ( Hg F and sd)
F Q ( column type flags )
FI100 SIGFI100 ( I F and sd)
F Q ( column type flags )
FAU20 SIGFAU20 ANAU20 SIGANAU20 (Au F, SD, F(+) -F(-), SD)
F Q D Q (column type flags )
```

The header contains the information:
Cell Dimensions : 73.58 38.73 23.19 90 90 90
Resolution Range: 36.761 - 2.300 A
Space group = P212121
and so on.

These are used as input to the phasing program (MLPHARE) like this:
LABIN FP=FTOXD3 SIGFP=SIGFTOXD3 -
FPH1=FAU20 SIGFPH1=SIGFAU20 -

```
DPH1=ANAU20 SIGDPH1=SIGANAU20 -
FPH2=FMM11 SIGFPH2=SIGFMM11 -
FPH3=FI100 SIGFPH3=SIGFI100
```

The output labels required for the MIR phase and its figure of merit could be named like this.
LABOUT PHIB=PHI_Au_Hg_I FOM=FOM_Au_Hg_I

3.1.1 Missing Data Treatment

In a typical series of diffraction experiments, not all Bragg reflections for a given resolution range are in fact recorded. Hence, after truncate some reflection data records may be entirely missing from the MTZ file, although the reflection indices lie within the measured resolution range. It is strongly recommended that index sets are made complete within the desired resolution range - a script to do this is provided in \$CETC/uniqueify. The MTZ file will then contain records where there are indices but no measured data. (These are flagged MNF for **missing number flag** or **measurement not found**) e.g.:

```
0 0 2 MNF MNF
0 0 4 517.0 23.0
0 0 6 1567.0 57.0
... ..
```

This means that it is easy to estimate completeness and programs such as reftmac and sigma can "restore" data estimates where required. Furthermore, a particular reflection may be recorded for the native protein but not for a derivative, and the corresponding combined reflection data record should indicate "missing data" for the derivative.

3.2 Maps

The electron density map is stored in a randomly-accessible binary file as a 3-dimensional array preceded by a header which contains all the information needed to describe it. This includes the extent of the array, and the grid it is calculated on, the axis order, the cell and symmetry, a title and the minimum, maximum and mean density. Maps are structured as a number of sections each containing a (fixed) number of rows and each row contains a (fixed) number of columns. The format is also used for envelope masks and images.

3.3 Coordinates

The standard format adopted for coordinate data is that used in the Brookhaven Protein Data Bank. The programs of the suite will handle either complete files or ones containing only a subset of the allowed record types. In particular the records containing the cell (CRYST1 and SCALEx) and coordinate data (ATOM or HETATM records) are of interest. The Protein Data Bank provides a full description of the complete format.

The standard setting of the orthogonal axes relative to the crystallographic for the Brookhaven format is:

$$x \parallel \mathbf{a}_y \parallel \mathbf{c}^* \quad x \parallel \mathbf{a}_z \parallel \mathbf{c}^*$$

The suite assumes these settings if the SCALEx cards are not present in a coordinate file. It is hoped to replace this soon by the new macromolecular **mmCIF format**, which has many of the features incorporated in the reflection format (Bourne et al (1995)). Peter Keller has been funded by CCP4 to develop library routines to facilitate this.

4 Library routines

One fruit of the collaborative nature of program development has been an extensive and exhaustively tested set of routines, covering most basic crystallographic applications. This is desirable both for speed in developing new software, which can utilise these, and for accuracy; bugs in code are best uncovered by frequent and varied use.

The CCP4 library subroutines perform the basic crystallographic and programming operations. There are routines for handling symmetry, and for reading and writing the standard format files for reflections, atomic coordinates, and maps. The library also contains forward and reverse fast Fourier transform routines (Ten Eyck (1973)). Utility routines parse the keyworded input and generate the metafiles used for 2-D plotting. There are also a small number of clever machine-specific routines which handle dynamic core allocation, file assignment and so on.

The data library contains tables of such things as space group symmetry operators, atomic form factors, the standard groups used in `protin`, and much other useful basic data.

Here is a brief list of the modules in the CCP4 program library. Documentation on them is available, either as man pages or as `.doc` files in the distributed `$CDOC` directory.

- **symlib** useful routines for handling symmetry operations.
- **mtzlib** reflection file handling;
- **maplib** for handling CCP4-style map data;
- **rwbrook** for handling coordinate (PDB/Brookhaven) files;
- **ftlib** crystallographic fft routines;
- **parser** processing free-format input containing 'keywords' -it actually lexes more than parses;
- **keyparse** a higher-level interface to the parser routines;
- **ccplib** contains various utility routines which are potentially machine-dependent. It is built up from either VMS-specific code in `vms.for` and `vmsdiskio.for` or Unix-specific code in `unix.m4` and `library.c`;
- **diskio** `io.for` or Unix-specific code in `unix.m4` and `library.c`; and `vmsdiskio.for` or Unix-specific code in `unix.m4` and `library.c`;
- **diskio** contains routines for random access to stream-mode files, but most of the relevant code is actually in `library.c`. The VMS version is in `vmsdiskio.for`; **plot84lib** low-level graphics with `plot84` metafiles;
- **plotsubs** higher-level interface

5 Program overview

The list of programs distributed by CCP4 is given in Appendix A. Some of these (marked with an asterisk) are not part of the CCP4 suite, but are nevertheless distributed by CCP4 ('aggregated' software). As techniques develop new programs are added, and as these usually are written in response to the requirements of particularly challenging problems, they are frequently innovative and represent genuine advances in the field. The CCP4 infrastructure means these can be distributed to the community of users extremely quickly, and the interchange between programmer and users is a valuable component in the development process, both in sharpening the algorithms and in finding bugs. In some ways the growth of the suite has been almost organic. I would like to highlight the process with some recent examples.

5.1 Data scaling and merging - *scala*

As larger proteins are studied, and multiple anomalous wavelength (MAD) phasing becomes more routine, there is a need for better experimental data. Part of this improvement must come from better scaling and merging algorithms. Also refinement programs require a reliable estimate of the standard uncertainty of each reflection and this has to be determined at this stage. Phil Evans is developing a program, **scala**, to allow scaling against many variables, (rotation angle, detector position, and so on). One extremely useful option is to include a master data set in the minimisation, which gives a more robust variant of local scaling. The estimates of standard uncertainty are obtained by modifying those given by the processing package to take account of agreement between symmetry equivalent reflections.

5.2 Heavy atom phasing: (MIR or MAD using *mlphare* and density modification)

5.2.1 Heavy atom refinement, and initial phase and weight estimates

Zbyszek Otwinowski appreciated that many older heavy atom refinement programs produced biased parameters. The heavy atom sites were used to determine preliminary protein phases, which were then treated as fixed during the subsequent refinement of the sites. By the simple improvement of testing all possible phases for each reflection, and appropriately weighting these, he obtained more reliable parameters, more accurate protein phases, and more realistic probabilities for each phase. This program is now widely used for heavy atom refinement, and for both MIR and MAD phasing in conjunction with density modification. (Otwinowski (1991)).

5.2.2 Phase improvement and molecular averaging

Kevin Cowtan developed algorithms for phase improvement and extension during his PhD. He was then funded on a short

term contract from CCP4 to extend and encode these, and during this period produced the programs **dm** and **dmmulti**(Cowtan (1994)). Jan-Pieter Abrahams approached a similar problem while working on F1 Atpase in a somewhat different way, and his program **solomon** is now also part of the suite (Abrahams (1996)).

5.3 Molecular Replacement using AmoRe

The most exciting developments in molecular replacement has been the successful use of poorer and poorer models which can be positioned in the new crystal form, and which provide sufficient initial phasing information for other phase improvement techniques to be able to bite. This is only possible when the programs can automatically search large numbers of solutions at each stage rapidly, and without excessive user intervention. Jorge Navaza has incorporated this into his program AmoRe, and the version distributed with CCP4 has solved many structures. (Navaza (1994)). The version distributed with CCP4 has solved many structures. (Navaza (1994)).

5.4 Macromolecular Refinement using refmac

It has been appreciated for many years that least squares minimisation is not the optimal way of refining a set of coordinates which are a long way from their target values, and that it can become trapped in false minima. Garib Murshudov has written a program, **refmac**, which has an option to use a maximum likelihood residual, where the appropriate weighting for reflections is based on the fit of F_o and F_c for the free set of reflections, and includes the experimental standard uncertainty (Murshudov (1996)). This converges more quickly than least squares in many cases, and generates properly weighted and less biased maps for model correction.

5.5 Validation using procheck

This program, developed by Roman Laskowski, does a comprehensive check of a protein's stereochemistry, and highlights parts of the structure where conformations are unusual (Laskowski (1993)). These are due either to interesting properties of the structure, or to possible errors of interpretation.

5.6 Tutorials and example scripts

CCP4 will be giving a demonstration of its software at the IUCr Computing School (held at Bellingham, USA August 1996), in the form of tutorials in certain areas. These five areas are MIR, MAD, density modification using **DM**, molecular replacement using **AmoRe**, and macromolecular refinement using **refmac** and **restrain**. For a description of these tutorials, have a look at the Web page http://www.dl.ac.uk/CCP/CCP4/_tutorial.html. Also, Appendix B gives an outline of some of the examples. For those not coming to Bellingham these tutorials will be distributed in the Suite at a later date. CCP4 also distribute a set of example scripts (unix and VMS) illustrating individual programs and common procedures.

6 Conclusions

There are disadvantages to the diverse traditions and dispersed centres where CCP4 is under development, but these have been largely overcome by centralising the distribution and maintenance at the Daresbury Laboratory. The professional expertise provided there is essential to administer the large body of source code now deposited. This service is only possible because of the central BBSRC funding whose recognition of the key value of this group over the years must be acknowledged. This has been augmented by industrial contributions. The CCP4 tradition of organic growth is based on the interests and enthusiasms of the individuals involved. Such a development could never have a commercial basis; there is no equitable mechanism for making payments to contributors. The CCP4 practices are in the best tradition of science, another example of how scientific research is best fuelled by openness in the exchange of ideas, methodology, and solutions on a generous and shared basis, in which the individuals are rewarded by the successful usage of the contributions. I am sure this is the explanation for the successful growth of the CCP4 suite over the past 17 years.

7 Distribution

The program suite is licensed free to academic institutes by Internet FTP or on a variety of media for a small handling/media charge. The programs may be obtained by Internet FTP from anonymous@ccp4a.dl.ac.uk:pub/ccp4. Separate arrangements are made for commercial organisations who should contact CCP4 directly. For further details about CCP4 or to obtain the programs please contact the CCP4 Secretary at Daresbury Laboratory (email: ccp4@dl.ac.uk).

Acknowledgements

A large number of people have contributed to CCP4 over the years and we thank them for their time and effort. The Daresbury staff are pivotal in directing and maintaining standards, and handling the now extensive administration. CCP4 is supported by the BBSRC and the ESF Network of the EACBM.

Appendix A

Data processing

- **mosflm package*** A widely used data processing program for film and image plate oscillation data. (Aggregated.)
- **LAUE*** For processing data taken with the Laue method. (Aggregated.)
- **ipdisp** For viewing and measuring images under X-Windows.
- **hklview** Displays zones of reciprocal space as pseudo-precession images under X-Windows.

Data scaling and reduction

- **scala** Scales batches of data from processed images with many options, including scaling against a predetermined data set. This is similar to local scaling. It merges data, adds together partially recorded reflections, monitors and rejects bad agreements between repeated measurements or symmetry equivalents and averages them for output. Various statistics on the averaging are produced. The merging statistics are used to improve estimates of standard uncertainties for reflections.
- **truncate** Converts from intensities to F_s by the method of (French and Wilson (1978)), checks the intensity distribution for weak data against expected statistics, and does a Wilson plot to estimate B factor.
- **rotaprep** A jiffy for converting various foreign formats of un-merged intensity data to multi-record **mtz** format for input to **scala**. It can easily be extended to accept additional formats.
- **absurd** Reads data from **madnes**, applies various corrections, reduces to the asymmetric unit and writes **mtz** file for **scala**.
- **postref** Postrefinement of film data.
- **unique** Generates a unique list of reflections.
- **freerflag** Assigns FreeR flags to a percentage of reflections.
- **wilson** Makes a Wilson plot.
- **reindex** Reindexes data when required. (Remember if two cell edges are equal you will not be able to distinguish h k l from k h $-l$.)
- **ecalc** Calculate normalized structure amplitudes E_s from F_s .

Data combination and scaling different sets

- **cad** Combines Assorted Data from several **mtz** files and resorts, changes asymmetric unit etc. See also **mtzutils**.
- **scaleit** Simple scaling of derivative to native data with option for anisotropic temperature factors Useful analyses of outliers in isomorphous and anomalous differences.
- **fhscal** Scales native to derivative data using Kraut scaling method.
- **rstats** Least squares scaling between F_o and F_c .

Obtaining ab initio phases Heavy atom phasing (MIR or MAD)

It is necessary first to collect then scale the different data sets together, (See above.) The next step is to find the heavy atom positions either from Pattersons or by direct method programs which use estimates of the Fh based on the observed differences. Before these methods can work it is essential that outliers have been detected and excluded.

- **fft** Patterson map calculation using fast Fourier algorithm. Coefficients may be isomorphous or anomalous differences.
- **rsps** Determines heavy atom positions from derivative difference Patterson maps. Can be used interactively to examine the fit of potential sites to the map.
- **vecsum** Patterson peak search.

- **vectors** Generates all Patterson vectors from a list of input atoms and produces a list of all vectors which fall within the volume of the Patterson calculated.
- **mlphare** Calculates phases and refines heavy atom parameters.
- **vecref** Refines heavy atom parameters in vector space.
- **crossec** Tabulates anomalous scattering factors f'' and f''' .

Phase improvement and molecular averaging

- **dm** Density modification using solvent flattening, Sayre's equation, histogram matching, NCS averaging and iterative skeletonisation.
- **dmmulti** A multi-crystal version,
- **solomon** Modifies the electron density maps by averaging, solvent flipping and protein truncation.
- **symfit** Fits best molecular transformations to sets of crystallographic coordinates e.g., heavy atom coordinates, related by non-crystallographic symmetry. (Not for polar point-groups.)

Molecular replacement

- **amore** is a complete molecular replacement system in one program, incorporating rotation functions, translation searches, and rigid body refinement. It can generate model structure factors from a coordinate file, or read them as input. This means that the model structure factors can be converted to Es using **ecalc**, or that they could be generated from a piece of electron density.
- **almn** Crowther fast rotation function. Algorithm not as powerful as amore but better output.
- **polarrfn** Kabsch's fast polar rotation function with stereographic plots.
- **ttfc** Space group general translation function.
- **mapsig** peak search and statistics on signal/noise for translation function map. Also sum, product, ratio of two maps.
- **rsearch** Rfactor search.
- **rfcorr** Analyzes correlations between cross- and self-rotation functions
- **rotmat** Converts **X-PLOR/MERLOT/amore** equivalent rotation angles.

Map and structure factor calculation

- **sfall** Structure factor calculation using inverse **fft**.
- **fft** Map calculation using fast Fourier algorithm.
- **extend** or **mapmask** Extend asymmetric unit of map to cover any grid volume.
- **refmac** and **sigmaa** both prepare coefficients for calculating weighted $2Fo - Fc$ and $Fo - Fc$ maps.

Map manipulation

- **mapmask** Map/mask extend program
- **maprot** Map skewing, interpolating, rotating and averaging program
- **ncsmask** Performs operations on non-crystallographic symmetry masks, e.g. before **dm**.
- **bones2pdb** Converts a **bones** output file to PDB file for **ncsmask**.
- **mama2ccp4** Converts **RAVE/MAMA**-format masks to ccp4 format.
- **xdlmapman** (associated with **RAVE/O**) Manipulates maps, exchanges formats etc.
- **mapsig** Can do arithmetic on two maps.
- **overlapmap** Map summation (averaging) and subtraction, real-space correlation coefficients and *R* factors.
- **peakmax** Picks peaks on map (e.g., for searching for water).

Refinement of protein models

- **protin** Prepares restraints for Hendrickson-Konnert refinement.

- **refmac** Refines or idealize structures, using intensity or amplitude based least squares or -loglikelihood residuals.
- **sfall** Prepares the X-ray contribution for **prolsq** or do unrestrained refinement.
- **profin** Prepares restraints for Hendrickson-Konnert refinement.
- **prolsq** Hendrickson-Konnert refinement with X-ray contribution read from a file. based least squares or -loglikelihood residuals (Murshudov (1996)).
- **restrain, tlsanl** Restrained geometry, rigid body, use of amplitude and phase observations, group anisotropic displacement parameters, disordered solvent corrections (Driessen et.al. (1989)).

Coordinate analysis

- **act** Coordinate checking.
- **angles** Calculates angles and bond lengths, Ramachandran plot. Alternatively look at your **prolsq** output.
- **areaimol** Finds solvent extended accessible area of atoms in a Brookhaven coordinate file and write an extra column to the file.
- **baverage** Averages *B* values for main and side chain atoms. Very useful program which gives average r.m.s. *B*s for main and side chain atoms. Much simpler alternative to Branden real space *RB*s to exclude wildly too small or too high values. See also **overlapmap**.
- **contact** Calculates various types of contacts and analyses water hydrogen bonding. (See also **act**.)
- **distang** Calculates intra- and inter-molecular distances.
- **gensym** Generates all symmetry-related sites from a list of input atoms, and produces a list of all sites which fall within the volume of the defined volume.
- **geomcalc** Does various geometry calculations on a molecule.
- **hgen** Generates hydrogen atoms for a protein coordinate file with standard geometry.
- **hbond** Calculates possible main chain hydrogen bonds.
- **lsqkab** Least squares fit of two sets of coordinates.
- **pdbset** Various useful manipulations of **pdb** files: e.g., add **CRYST** and **SCALE** lines, generate symmetry-related subunits, rename chains, renumber residues, transform coordinates.
- **polypose** Superposition of many multi-domained structures.
- **procheck** Comprehensive stereochemistry checking, Ramachandran plot, secondary structure features. Assorted pretty plots including Ramachandran plot.
- **profin** Can be run alone to check for bad contacts to symmetry equivalents.
- **resarea** Prints solvent accessible areas for each residue, chain and for the whole molecule.
- **sortwater** Sorts waters by the protein chain to which they “belong” in the case of a protein with several equivalent subunits.
- **surface** Determines accessible surface area.
- **volume** Determines polyhedral volume around selected atoms.
- **waterarea** Analyses solvent accessible areas for water molecules.
- **watertidy** Assigns waters to nearest subunit and residue.
- **watpeak** Lists peaks found by **peakmax** near to atoms.

Pictorial presentation of results

- **npo** Plots maps and draws structure onto them. Various graphical representations.
- **pltdev, xplot84driver, xccpjiffy2idraw** Convert **plot84** metafiles to X-Windows or PostScript format. **xccpjiffy2idraw** converts the result to PostScript which can be edited with **idraw**.
- **procheck** See above
- **xloggraph** Plots graphs of tables from (many) **ccp4** programs’ log files under X-Windows.

Utility programs

- **axissearch** Changes axis and cell. (See also **tracer**.)

- **cad** Combine assorted data (and sort) a number of reflection files with various possible operations on the data items. Apart from manipulating the values, data may be converted from one area of reciprocal space to another. Other special functions allow for the generation of input data, for expansion of the data to a lower symmetry if required, and for the generation of data for input to **rsearch**.
- **coordconv** Interconverts various coordinate formats.
- **f2mtz** Converts (free-)formatted reflection files to MTZ format.
- **hklplot** Plots "precession" pictures from reflection files.
- **hklview** is better than **hklplot** if you have X-Windows.
- **mtzdmp** List header and reflections to terminal or printer (Unix script **mtzdmp** runs it more simply.)
- **mtzmnf** Identify missing data entries in an MTZ file and replace with missing number flag (e.g. NaN).
- **mtztona4** Converts MTZ files to portable **na4** ASCII format. (For exchange with another machine.)
- **mtzutils** Edits columns, title or labels, combines two reflection files. See also **cad**.
- **mtz2various** Produce a file in suitable form for **MULTAN**, **SHELX** or **X-PLOR**. Easy to extend for other formats.
- **na4tomtz** Inverse of **mtztona4**.
- **reindex** Reindexes MTZ files when you realise something is wrong. Also changes reflection to asymmetric unit.
- **sortmtz** Sort and/or merge MTZ files.
- **stereo** Reconstruct 3D coordinates from measurements of stereo diagrams.
- **tracer** Lattice TRAnsformation/CELL Reduction.
- **xldataman** (associated with **RAVE/O**) manipulates reflection files, exchanges formats etc.

References

- Abrahams J. P. and Leslie A. G. W., (1996) *Acta Cryst.* **D52**, 30-42.
- P. E. Bourne, H. M. Berman, B. McMahon, K. D. Watenpaugh, J. Westbrook, and P. M. D. Fitzgerald., The Macromolecular Crystallographic Information File (mmCIF) (1995), *Methods in Enzymology*, submitted.
- Collaborative Computational Project, Number 4, (1994) *Acta Cryst.*, **D50**, 760-763.
- Cowtan, K. (1994), Joint CCP4 and ESF-EACBM Newsletter on Protein Crystallography, **31**, 34-38.
- French, G.S. and Wilson, K.S. (1978) *Acta Cryst.* **A34**, 517.
- Driessen, H., Haneef, M. I. J., Harris, G. W., Howlin, B., Khan, G., & Moss, D. S. (1989) *J. Appl. Cryst.*, **22**, 510-516.
- Laskowski, R.A, MacArthur, M.W., Moss, D.S. and Thornton, J.M. (1993) *J. Appl. Cryst.* **26**, 283-291.
- Murshudov, G., Vagin, A. and Dodson, E.,(1996) in the Refinement of Protein structures, Proceedings of Daresbury Study Weekend.
- Navaza, J. (1994) *Acta Cryst.* **A50** 157-163.
- Otwinowski, W. (1991) in *Isomorphous Replacement and Anomalous Scattering, Proceedings of the CCP4 Study Weekend*(Eds. W. Wolf, P.R. Evans and A.G.W. Leslie), 80-86.
- Skarzynski, T., (1992) *J. Mol. Bio.* **224**, 671-683.
- Ten Eyck, L. (1973) *Acta Cryst.* **A29**, 183-191.