CCP4 NEWSLETTER ON PROTEIN CRYSTALLOGRAPHY

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

Number 39 **March 2001**

Contents

Editor: Peter Briggs

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

News from CCP4: March 2001

Peter Briggs, Martyn Winn, Sue Bailey, Alun Ashton, David Brown, Charles Ballard

1. Staff changes

In some ways it is the end of an era for CCP4, with the announcement of two departures from the core CCP4 team based here at Daresbury.

Colin Nave writes: ``Those who attended the CCP4 study weekend will remember that **Sue Bailey** will be taking up a new job at Berkeley, California in April. Sue's drive and organisational abilities have been a key factor in the growth of the CCP4 project over the past 8 years. At the same time she has managed a succesful research program at Daresbury, investigating the structure of DMSO reductase and other proteins. I have found it a pleasure working with her during her time at Daresbury. I am sure the CCP4 community will join me in wishing Sue and her family all the best in California." The DL staff would like to thank Sue for her invaluable contributions to the project and wish her all the best for the future.

Also **David Brown**, the CCP4 administrative assistant, announced his intention to retire this coming summer. David has been with CCP4 for two years now and during that time has made valuable contributions to the overall running of the project - particularly in dealings with our commercial customers and also in the organisation of the annual Study Weekends. We are sorry to see him go but wish him a relaxing and enjoyable retirement.

2. Workshops and Conferences

Since the last CCP4 newsletter almost a year ago there has been a large number of CCP4-related activities.

Last July Alun Ashton, Peter Briggs and Harry Powell made a trip to the **50th Anniversary ACA Meeting** in St. Pauls, Minnesota, where we had a stand in the exhibition, demonstrating CCP4 and MOSFLM, and were treated to some American hospitality! We had a great time meeting American users of the software, many of whom were only known to us previously as names on the bulletin board.

From left to right: Katherine McAuley, Alun Ashton, Peter Briggs and Harry Powell at the ACA 2000

So we would like to thank everyone who came to visit the stand. We would also like to thank Katherine McAuley for helping out with demonstrations of the software, and to the ACA Council, who very kindly offered us the complementary exhibition booth.

If you missed us then we will be back at the ACA again this summer, this time in Los Angeles - see the web page at http://www.hwi.buffalo.edu/ACA/ACA-Annual/LosAngeles/LosAngeles.html.

Also last summer Martyn Winn, Charles Ballard and Harry Powell represented CCP4 at the **ECM 19 meeting in Nancy** last August (thanks to those who came to look at our posters). Martyn also presented recent developments in REFMAC at the **Gordon Conference on Diffraction Methods in Molecular Biology** held last July.

More recently, the **CCP4 Study Weekend 2001** took place in York in January, on the topic of "Molecular Replacement and its relatives". CCP4 would like to thank especially the scientific organisers - Jim Naismith (St-Andrews) and Kevin Cowtan (York) - and of course all the speakers. Alun Ashton, David Brown and Daresbury staff (Pat Broadhurst, Alison Mutch and Sue Waller) also played a vital role in making sure that things ran smoothly on the ground. The proceedings from the study weekend will be published later this year in *Acta Cryst. D*, but in the mean time Maria Turkenburg has compiled a useful set of associated links at http://www.ccp4.ac.uk/stwk01URLs.html

This year the Study Weekend was flanked by a number of additional activities. The **SRS PX Specialist User Group Meeting** took place on the Thursday afternoon prior to the start of the workshop (see the report by Pierre Rizkallah), while on either side of the Study Weekend there was the **CCP4/MAX-INF Refinement Workshop** also in York (see Eleanor Dodson's report for more details).

Finally on the Friday morning a short **"Introduction to CCP4"** session was held just before the start of the official Study Week programme. CCP4 staff and friends (Martyn Winn, Alun Ashton, Maria Turkenburg, Peter Briggs, and Harry Powell) presented material aimed at both new and existing users of the suite. We would like to thank those who attended (the hall looked quite full from the stage!) and hope that you found it useful.

CCP4 also participated in the **PX Data Collection Workshop** held at Daresbury Laboratory at the start of February. Twenty students from labs around the UK attended the

week-long event, which covered various aspects of protein crystallography and featured a number of hands-on practical sessions. (We would like to say particular thanks to Lisa Wright for bravely using the CCP4 graphical interface for her Molecular Replacement session!). We have two reports on the workshop in this newsletter, one by organiser Liz Duke and the other by workshop student Jasveen Chugh.

Bringing us more-or-less up-to-date, Alun Ashton attended the recent **ESRF High Throughput Structural Biology** satellite meeting in Grenoble, presenting a poster on the developments within CCP4 in anticipation of high-throughput structure determination.

CCP4 presence of some description is already planned at a number of forthcoming meetings, including: **BCA Spring Meeting** in Reading (April), the **ECM, Krackaw** (August; see http://www.ch.uj.edu.pl/ECM2001.htm), **BCA Summer School** in St. Andrews (September). Details of relevant courses and events can as always be found on the CCP4 "Courses" page at http://www.ccp4.ac.uk/ccp4course.html.

3. New Release 4.1

The end of January saw the release of version 4.1 of the CCP4 suite, followed shortly after by the patch release 4.1.1. As always the patch release is intended only to fix minor bugs in the release, and if you are already using 4.1 without any problems then we don't recommend you bothering to upgrade.

The major changes from 4.0 to 4.1 are:

- **REFMAC5**
  This is a major new version of the refinement program which can now automatically prepare geometric restraints and identify disulphides, covalent bonds and cis-peptides prior to refinement (dispensing with PROTIN). Includes new libraries for nucleotides, sugars and some common ligands.
  A molecular **SKETCHER** is provided to create and modify monomer library entries. REFMAC5 also offers an improved bulk solvent correction and the option of refining TLS parameters.
- **MOSFLM** **6.10**
  The data processing program MOSFLM is now included as part of the suite and will build automatically under the **--with-x** option of configure. On systems where this option is not supported, manual building of MOSFLM should still be possible. *(For more news about MOSFLM, see Harry Powell's article in this newsletter.)*

The release also includes the following new programs:

- **ANISOANL**: analyses of model anisotropic U values *(see the article on ANISOANL in this newsletter)*
- **CAVENV**: calculate macromolecular cavities & envelopes
- **COMBAT**: prepare data for input to SCALA (replacement for ROTAPREP)
- **DTREK2MTZ**: converts d*trek scalemerge output to MTZ format
- **FFFEAR**: fitting model fragments into electron density
- **MAPSLICER**: interactive viewer for map sections *(see the article on MAPSLICER)*
- **ROTGEN**: simulates X-ray diffraction rotation images

Updates to the **graphical user interface CCP4i**:

- **Installation**: CCP4i no longer requires Tcl/Tk to be built with a non-default flag. It is best run using the bltwish interpreter.
- **New and updated tasks**, including:
  - *Monomer Library Sketcher* (for graphics display of monomers and interface to Libcheck and the monomer geometry libraries).
  - New interfaces to: *BAVERAGE*, *CONTACT*, *FFFEAR*, *DMMULTI*, *DTREK2MTZ*, *SFTOOLS* for SF analysis, *SIGMAA*, and *WATERTIDY*.

Other highlights in 4.1 include:

- New versions of **MOLREP**, **RSPS** (4.2), **DM** (2.1), **SCALA** (2.7.5), **MAPMASK** and **MAPROT**
- The MTZ file format has been expanded to include dataset-specific cell and wavelength information
- Updated **xdl_view** library (4.4)
- Option to include the LAPACK linear algebra package using **--with-lapack** configure switch
  (See the article on LAPACK in CCP4 4.1.)

Finally, we are once again making precompiled binaries of CCP4 4.1.1 available for a limited number of platforms (essentially, only those we have easy access to!): IRIX (o32- and n32-bit versions, prepared on IRIX6.5 R10k) and alpha (prepared on Digital UNIX V4.0F). As before, these must be downloaded **in addition** (not instead of) the normal CCP4 distribution, and you should read the enclosed BINARY.readme file.

As always, details of all the changes can be found in the CHANGES file in the top-level directory ($CCP4), and in $CCP4/html/CHANGESinV4_1.html. We also urge people to check the CCP4 Problems Pages before reporting any bugs (with fixes, if possible!) to ccp4@ccp4.ac.uk.

On a personal note I would like to thank the Daresbury staff for their hard work in making the new release. I would also like to thank the York programmers for their help, and all those at various sites who expended time and effort to test the various beta releases and send me bug reports and fixes - thank you!

4. Other News

Newsletters: This will be my final newsletter as editor, as of issue 40 the mantle passes to Charles Ballard. Please e-mail Charles at c.c.ballard@ccp4.ac.uk if you wish to contribute articles to the next newsletter.

CCP4i: Following the release of CCP4 4.1, maintenance and development of CCP4i (the CCP4 graphical user interface) has passed to the Daresbury staff, allowing Liz to concentrate more on the CCP4 Molecular Graphics project (see her article in this newsletter for more details). CCP4 are still committed to maintaining and developing the interface; please visit the new CCP4i home page at http://www.ccp4.ac.uk/ccp4i_main.php for further information.

New location at DL: We have finally made our move across the Daresbury site to our new palatial offices above the Structural Biology Lab (SBL), so please come and visit us if you are on site. PX users are also welcome to ask for demonstrations of CCP4i or other CCP4 software while they are at the Lab.

S_c: measuring shape complementarity at protein-protein interfaces

Michael C. Lawrence

Biomolecular Research Institute, 343 Royal Parade, Parkville, Victoria 3052, AUSTRALIA

Current address: CSIRO Health Sciences and Nutrition, 343 Royal Parade, Parkville, Victoria 3052, Australia

email: mike.lawrence@hsn.csiro.au

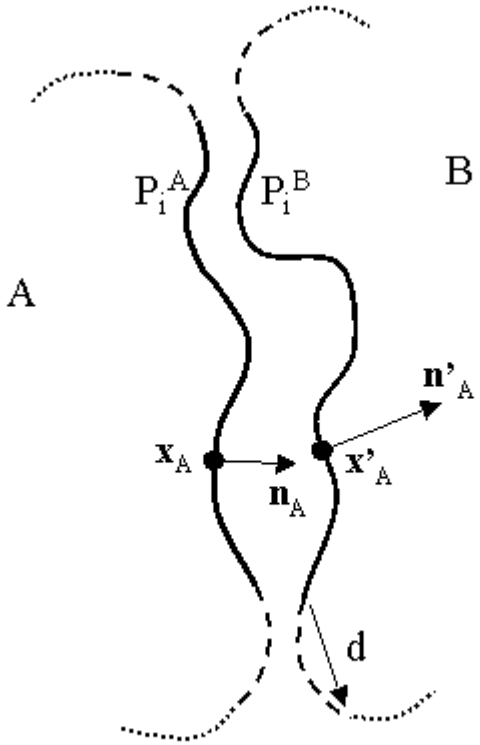
Background

The statistic S_c (Lawrence & Colman, 1993) measures the geometric surface complementarity of protein-protein interfaces. S_c depends both on the relative shape of the surfaces with respect to each other and on the extent to which the interaction brings individual elements of the opposing surfaces into proximity. The first dependence is *via* the use of normal products, and the second dependence is *via* an inverse exponential dependence on the distance of separation. The original SC software was distributed as a developmental version. The author's current version has now been released in CCP4 and is completely revised. It is considerably faster and easier to use than the developmental version.

In the original article the authors computed S_c for a number of different types of protein / protein interfaces and concluded that protein antibody / protein antigen interfaces appeared on average less complementary than both protein subunit / protein subunit interfaces and protein / protein inhibitor interfaces. This result was rationalized on the basis that the evolution of an antibody-antigen interface occurs in a different fashion to that of a protein subunit / protein subunit interface or a protein / protein inhibitor interface. It also appeared consistent with the observation that antigen binding sites on antibodies contain on average a higher percentage of aromatic residues (Padlan, 1990) than other interfaces - these residues having fewer conformational degrees of freedom than smaller hydrophobic or polar residues and their prevalence may well be expected to lead to less intimate packing. S_c has also been used to examine the shape complementarity of a T cell receptor in complex with a self peptide bound to a class I MHC molecule (Garcia *et al.*, 1998; Ysern *et al.*, 1998). The shape complementarity S_c was yet lower here than the average for protein antibody / protein antigen interfaces, presumably consistent with the need for a relatively low affinity interaction between the T-cell receptor and the peptide-MHC complex. More recently, higher S_c values have been reported for a T-cell receptor in complex with a foreign peptide bound to a Class II MHC molecule (Reinherz *et al.*, 1999).

Definition of S_c

Consider two interacting molecules A and B and their molecular surfaces (Figure 1).



For each molecule we compute that portion of the molecular surface that is buried from the solvent *via* the interaction with the other molecule - these portions are termed P^A and P^B respectively. A peripheral band is then removed from each of these buried surfaces P^A and P^B by excluding that area of each that is within a distance d of the solvent-exposed portion of the respective molecular surface. The resultant subset of each buried surface is termed P_i^A and P_i^B respectively, where the subscript i denotes "interior". In the above Figure, accessible surface is shown dotted, peripheral surface is shown dashed and interior surface as unbroken line.

For each point $\mathbf{x}_A$ in P^A we find its nearest neighbour $\mathbf{x}'_A$ on P_i^B. Let $\mathbf{n}_A$ be the outwardly-oriented surface normal at $\mathbf{x}_A$ and $\mathbf{n}'_A$ be the inwardly-oriented surface normal at $\mathbf{x}'_A$. Define the scalar function

$$S^{A->B}(\mathbf{x}_A) = (\mathbf{n}_A . \mathbf{n}'_A) \exp[-w (|\mathbf{x}_A-\mathbf{x}'_A|)^2]$$

on the surface P_i^A, where w is a scalar weight.

Likewise by considering all points $\mathbf{x}_B$ on surface P_i^B we may define

$$S^{B->A}(\mathbf{x}_B) = (\mathbf{n}_B . \mathbf{n}'_B) \exp[-w (|\mathbf{x}_B-\mathbf{x}'_B|)^2]$$

where $\mathbf{x}'_B$ is the nearest point to $\mathbf{x}_B$ on P^A, $\mathbf{n}_B$ the outwardly-oriented normal at $\mathbf{x}_B$ and $\mathbf{n}'_B$ the inwardly-oriented normal at $\mathbf{x}'_B$.

S_c is then defined as

$S_c = (\{S^{A->B}\}+\{S^{B->A}\}) / 2$

where braces denote the median of the distribution of $S^{A->B}(\mathbf{x}_A)$ and $S^{B->A}(\mathbf{x}_B)$ values over P_i^A and P_i^B respectively. Use of the median reduces the dependence of S_c on points that are outliers in the respective distributions.

Numeric calculation of S_c can be achieved *via* approximating the buried surface surfaces as uniformly distributed sets of points ("dots") sampled in the fashion outlined by (Connolly, 1983). In the standard calculation of S_c (Lawrence & Colman, 1993), d = 1.5 Å, w = 0.5 $Å^{-2}$ and the surface sampling density is 15 dots / $Å^{-2}$.

Cross comparison of S_c values

As is clear from the definition above, S_c depends not only upon the atomic coordinates, but also upon a set of parameters, and hence any published value for S_c should state the values used for these parameters. The impact of these various values upon S_c are as follows:

1. *Atomic coordinates*

   The atomic coordinates underlie the definition of the protein interface. These coordinates have an error associated with them and the reliability of S_c will consequently be lower for less well-determined structures. However, (Lawrence & Colman, 1993) argue that coordinate error, given its relatively random nature, may not impact greatly on the value of S_c.

   The inclusion of solvent molecules within the interface needs special consideration. (Lawrence & Colman, 1993) suggest performing two calculations of S_c – first with the solvent associated with one molecule and then with the other, and simply taking the average. In other circumstances it may be more appropriate to omit the solvent altogether.

2. *Probe and atomic radii*

   These radii define the molecular surface and altering them will alter S_c. However, a change in atomic radius at a given site on the interface would not be anticipated to alter substantially the normal product of juxtaposed surface elements, its effect would be through the distance exponential.

3. *Width d of excluded interface periphery*

   The periphery of the buried interface is excluded from consideration in S_c for the reasons outlined above. Decreasing the width of the excluded band will decrease S_c *via* the inclusion of intrinsically non-complementary surface.

4. *Distance weighting factor w*

   The exponential distance weighting factor w acts as a scale factor for the "fit" of the surfaces. Altering this parameter should not affect S_c for highly complementarity surfaces, but will have a significant effect on S_c for more poorly fitting surfaces.

5. *Surface point density*

   Under-sampling the molecular surfaces should be avoided as this will have a marked impact on S_c. Test calculations show that the computation is stable at a dot density of around 15 / $Å^2$.

The CCP4 version SC is distributed with the same set of radii and default parameters that accompanied the original software obtained from the author and which were used in the calculations cited in (Lawrence & Colman, 1993). These values should be used for all calculations if cross-comparison is to be made with already-published literature values of S_c.

Major improvements contained in the CCP4 version of SC

A number of major improvements have been made to the SC software:-

1. Surface calculation is now carried out within the program, rather than piping intermediate results to Connolly's MS program in a stand-alone fashion. The program also now incorporates Connolly's new mds subroutine (obtainable from http://www.biohedron.com), which has a faster surface generation algorithm than the original MS.
2. Selection of interface atoms is now done *via* a distance metric, avoiding the need for the initial low density surface computation.
3. Molecule definition is now handled *via* a chain/residue/atom name parser.
4. The interface to GRASP (Nicholls, 1993) is now incorporated directly in the program.
5. Graphical output of $S^{A->B}$, $S^{B->A}$ and distance histograms is now provided in CCP4 xloggraph-compatible form.

Conclusions

S_c is established as a useful tool quantifying the geometrical packing of protein interfaces and the new CCP4 version SC should greatly facilitate its use.

Given the large increase in the number of structures available since the original work of (Lawrence & Colman, 1993), it would be appropriate to re-assess the original conclusions regarding the complementarity of antibody-antigen interfaces compared to other forms of protein-protein interfaces, and also to consider extending these calculations to assess the broader range protein protein-receptor complexes that are now available. Such work is in progress in our laboratory (see for example (Epa and Colman, 2001)).

Acknowledgments

I thank my colleague Brian Smith for assistance with coding the new version of SC, Michael Connolly for making available the mds subroutine and Peter Briggs for incorporating CCP4 compatibility into SC.

References

Connolly, M. L. (1983). Analytical molecular surface calculation. *J. Appl. Crystallogr.* **16**, 548-558.

Epa, V.C. & Colman, P.M. (2001). Shape and Electrostatic Complementarity at Viral Antigen-Antibody Complexes. *Curr. Topics. Microbiol. Immunol.* (in press).

Garcia, K. C., Degano, M., Pease, L. R., Huang, M., Peterson, P. A., Teyton, L. & Wilson, I. A. (1998). Structural basis of plasticity in T cell receptor recognition of a self peptide-MHC antigen. *Science* **279**, 1166-1172.

Lawrence, M. C. & Colman, P. M. (1993). Shape complementarity at protein/protein interfaces. *J. Mol. Biol.* **234**, 946-950.

Nicholls, A. J. (1993). GRASP: graphical representation and analysis of surface properties. *Biophys. J.* **64**, A116.

Padlan, E. A. (1990). On the nature of antibody combining sites: unusual structural features that may confer on these sites an enhanced capacity for binding ligands. *Proteins* **7**, 112-124.

Reinherz, E. L., Tan, K., Tang, L., Kern, P., Liu, J., Xiong, Y., Hussey, R. E., Smolyar, A., Hare, B., Zhang, R., Joachimiak, A., Chang, H. C., Wagner, G. & Wang, J. (1999). The crystal structure of a T cell receptor in complex with peptide and MHC class II. *Science* **286**, 1913-21.

Ysern, X., Li, H. & Mariuzza, R. A. (1998). Imperfect interfaces. *Nature Struct. Biol.* **5**, 412-414.

ANISOANL - analysing anisotropic displacement parameters

Martyn Winn

Daresbury Laboratory,
Daresbury,
Warrington
WA4 4AD, U.K.
m.d.winn@dl.ac.uk

Introduction

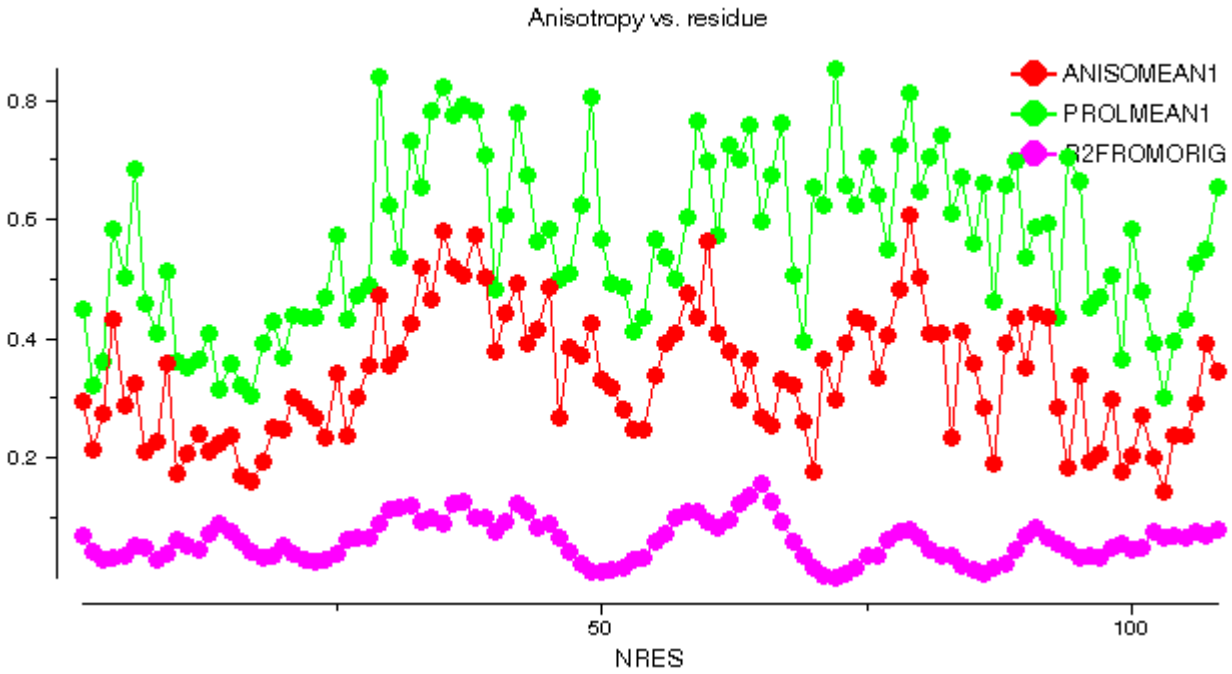
With improving data collection techniques, there are an increasing number of structures being solved to atomic resolution (1.2A or better). At such resolutions, anisotropic displacement parameters (ADPs) can be determined, providing detailed information on the averaged atomic displacements (see e.g. Merritt (1999) for a recent review). However, with six parameters per atom from the symmetric **U** tensors which represent the ADPs, there is perhaps an overabundance of information, and there is a need for simple methods with which to interpret the data.

ANISOANL is a new CCP4 program (available with version 4.1) which provides some simple tools for analysing ADPs. In this article, I will give an overview of these tools. Examples are taken from a 1.5A structure of barnase (PDB id 1a2p) and a 1.15A structure of myoglobin (PDB id 1a6g). For information on running the program, please see the program documentation. For information on obtaining the program, please see the CCP4 web pages.

Plots derived from ADPs

The PLOT option provides a series of plots in a similar style to the program BAVERAGE for the isotropic case. Several plots are given, reflecting the more complex information given by ADPs. The equivalent isotropic displacement parameter (Uiso = trace(**U**)/3), gives a measure of the size of the overall displacement. The anisotropy A, defined as the ratio of the smallest to the largest eigenvalue of **U**, gives a measure of how non-spherical the thermal ellipsoid is (1 implies complete isotropy, while 0 is extreme anisotropy). An anisotropic ADP can be either prolate (cigar-like) or oblate (disc-like). These two cases are discriminated by the value of the ratio of the middle to the largest eigenvalue of **U**, which is 1 for oblate and equal to A for prolate.

barnase: 1a2p



This graph shows some plots for chain A of barnase. ANISOMEAN1 is the anisotropy A, averaged over main chain atoms for each residue, while PROLMEAN1 is the prolate/oblate discriminator, similarly averaged. Most residues have an average anisotropy in the range 0.2 to 0.6 which is fairly typical. The value of PROLMEAN generally lies closer to A than to unity, implying a tendency towards prolate ellipsoids. R2FROMORIG shows the square of the distance from the centre of mass of the molecule.

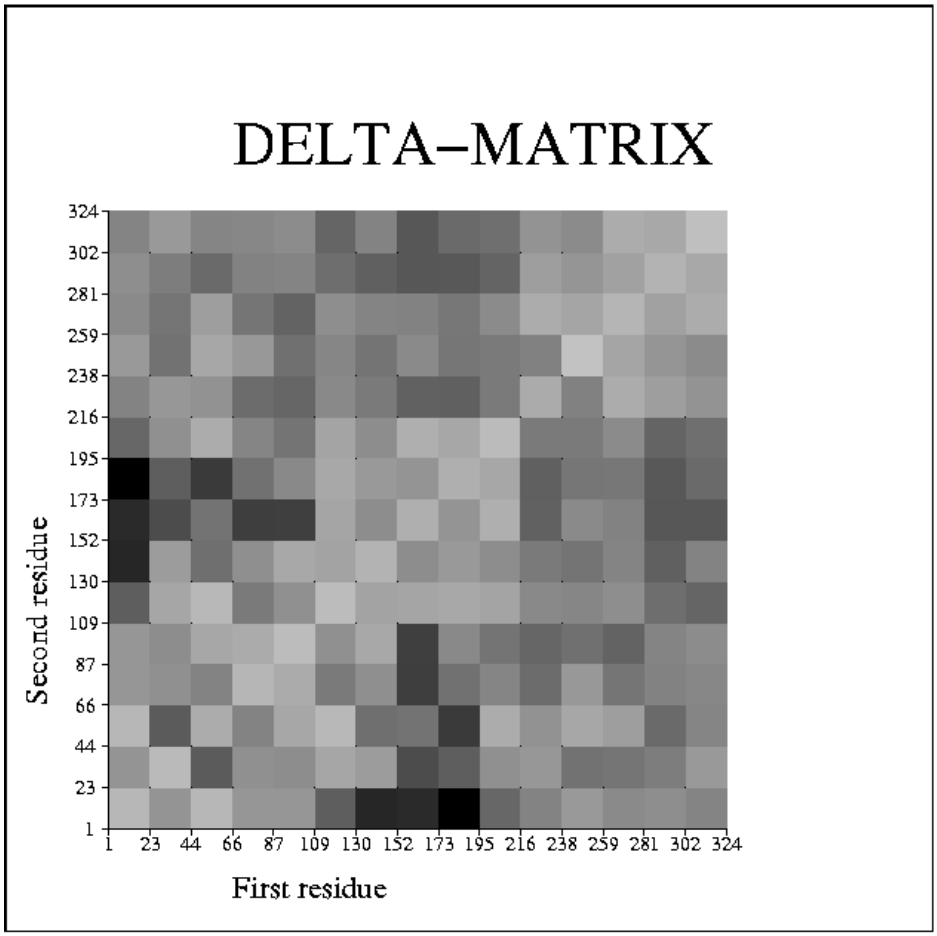
Rosenfield's rigid-body postulate

ADPs are useful for discerning detailed motions at an atomic level, for example at catalytic sites (with the usual provisos that ADPs bundle all kinds of displacements and errors, and there is no information on interatomic correlations). But it is also often useful to identify displacements operating at a larger scale, for example displacements of secondary structure elements. The simplest example of this is the case of a group of atoms moving as a quasi-rigid body.

Rosenfield *et al.* (1978) proposed a `rigid-body postulate' based on refined ADPs. Since interatomic distances within a rigid body are fixed, the difference in the projections of the ADPs of 2 atoms in a rigid body (the `Delta' value) must be zero. Note that this applies to all pairs of atoms in the rigid body, and not just bonded pairs. This is a necessary but not sufficient condition for rigid-body displacements. In any case, since proteins are never completely rigid, we can only identify possible quasi-rigid groups from low values of Delta for a set of atoms. Schneider (1996) has applied this approach to the protein SP445.

ANISOANL will produce a postscript figure giving Delta values between pairs of atoms, averaged over a number of bins. For example, using the main chain atoms of barnase, one gets the following figure:

barnase: 1a2p



Light shading corresponds to low values of Delta, while dark shading corresponds to high values of Delta. Possible rigid-body behaviour is indicated by blocks of light shading. The blocks may or may not be contiguous along the protein chain. These plots are usually very noisy, but it is usually possible to discern some structure (perhaps with the help of the `Geophys' team!). In this example, it is possible to identify the three molecules of barnase that occur in the asymmetric unit (labelled as residues 1-108, 109-216 and 217-324). Thus it appears that the 3 molecules each move as a quasi-rigid body.

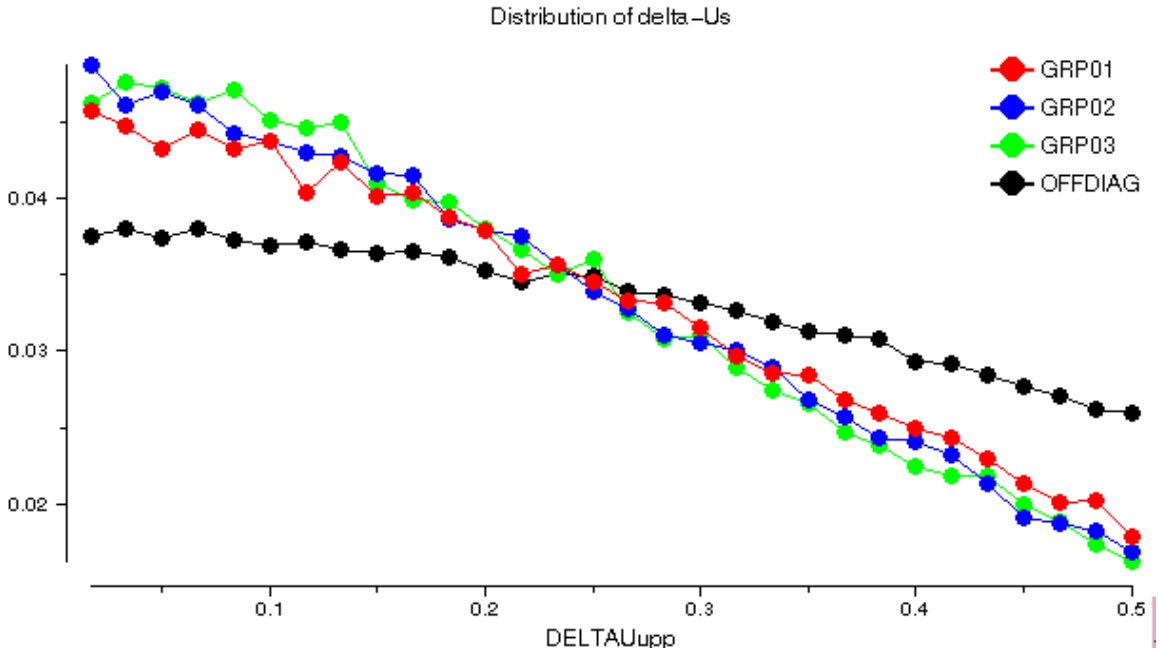
To analyse these results further, it is necessary to explicitly identify the three molecules as three rigid bodies. This is done via the TLSIN file, which in this case is:

```
TLS
RANGE  'A   3.' 'A 110.' MNCH

TLS
RANGE  'B   3.' 'B 110.' MNCH

TLS
RANGE  'C   3.' 'C 110.' MNCH
```

Each "TLS" record begins a new rigid group, consisting of the atoms specified in one or more "RANGE" records.

ANISOANL plots the distribution of Delta values for each rigid group, and for all pairs of atoms belonging to different rigid groups (the OFFDIAG plot):

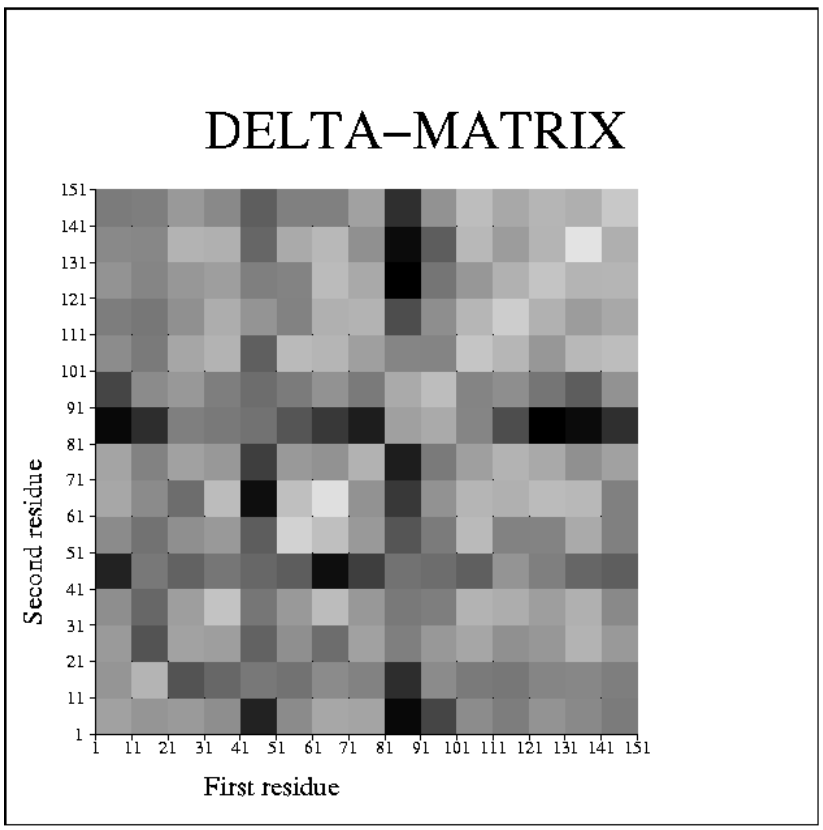
barnase: 1a2p



The distribution is similar for each putative rigid group, while the OFFDIAG plot is skewed towards higher values of Delta. This implies that pairs of atoms within a molecule satisfy the rigid body postulate better than pairs of atoms from different molecules.

A second example, which looks at the internal structure of a single molecule rather than several whole molcules, is given by the 1.15A structure of a myoglobin-CO complex (PDB id 1a6g). The plot of the Delta matrix looks like:

myoglobin-CO: 1a6g



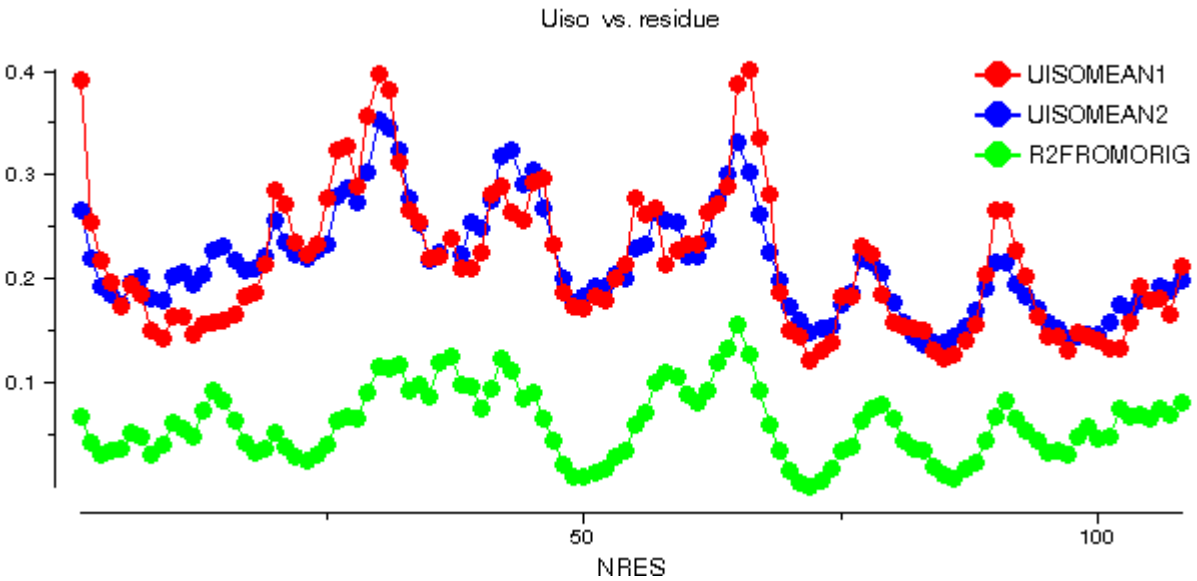
Again there is a lot of noise, but it is possible to pick out possible pseudo-rigid regions at residues 1-21, 21-41, 51-81, 81-101 and 101-151. In fact, these regions correspond closely to helices A (residues 3-18), B and C (20-35,36-42), D and E (51-57,58-77), F (86-95) and G and H (100-118,124-149) respectively. Looking at the inter-helix Delta values, helix F stands out in particular as having large Delta values, and thus not being part of any larger pseudo-rigid group. It appears that helices, or pairs of helices, form the relevant units for describing large scale displacements in myoglobin.

Fitting TLS parameters

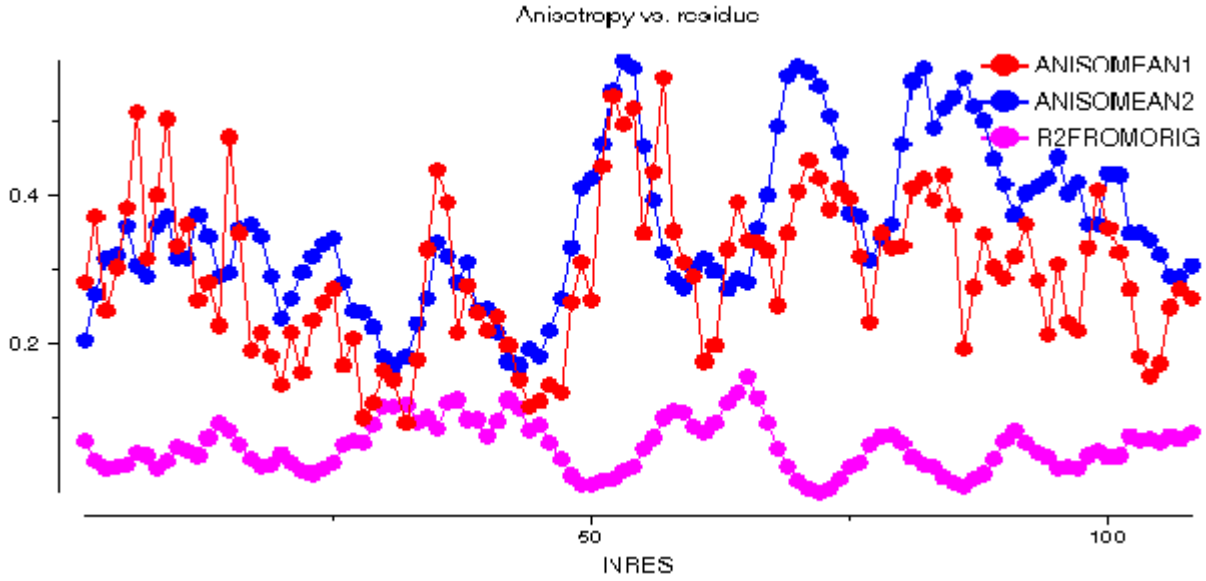
Given refined ADPs for a structure, one can try to fit TLS parameters describing rigid body motion to these ADPs (see Schomaker and Trueblood (1968)). Note that this is distinct from refining TLS parameters directly against Xray data. Also, the fitting will maximise the contribution of TLS, and therefore overestimate rigid-body motion. ANISOANL will fit TLS parameters with the FITTLS option, using TLS groups specified in the TLSIN file.

I have done this for the barnase structure treating each molecule as a separate TLS group (i.e. using the TLSIN file given in the previous section). The success of the fitting can be assessed by comparing the equivalent isotropic displacement parameter, the anisotropy and the prolate/oblate discriminator derived from the TLS parameters (labelled UISOMEAN2, ANISOMEAN2 and PROLMEAN2) to those derived from the refined ADPs. This is shown for chain C of barnase (this shows the best fit, though chains A and B are similar):

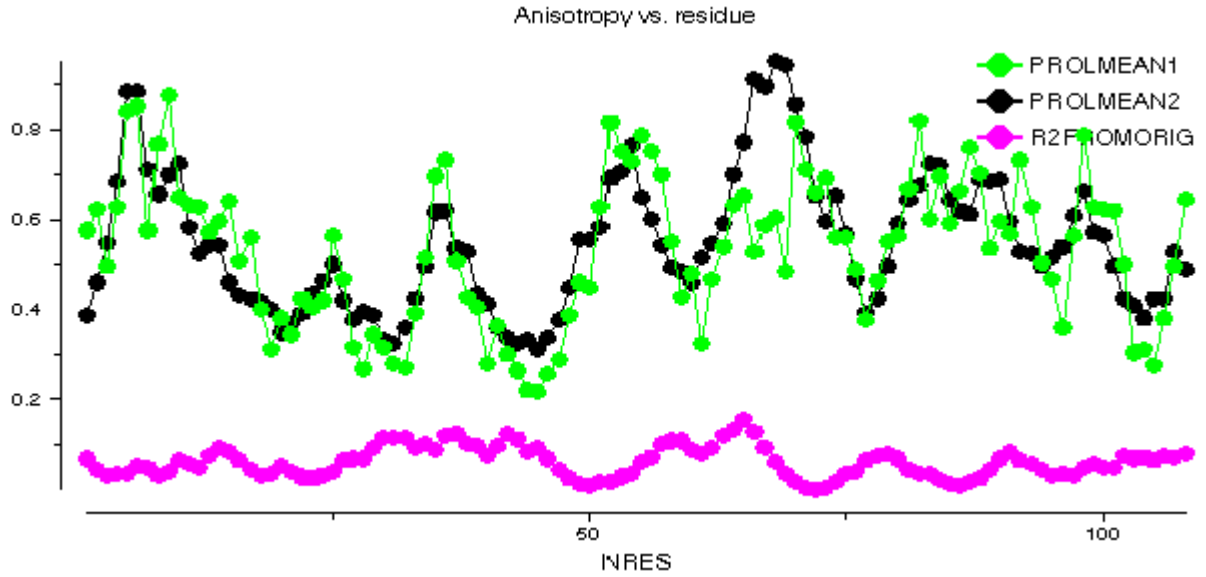
barnase: 1a2p



barnase: 1a2p



barnase: 1a2p



The fit is fairly good, and thus TLS provides a good first-order description of the refined ADPs. The discrepancies show, however, that there is some detail unaccounted for by the pseudo-rigid body description.

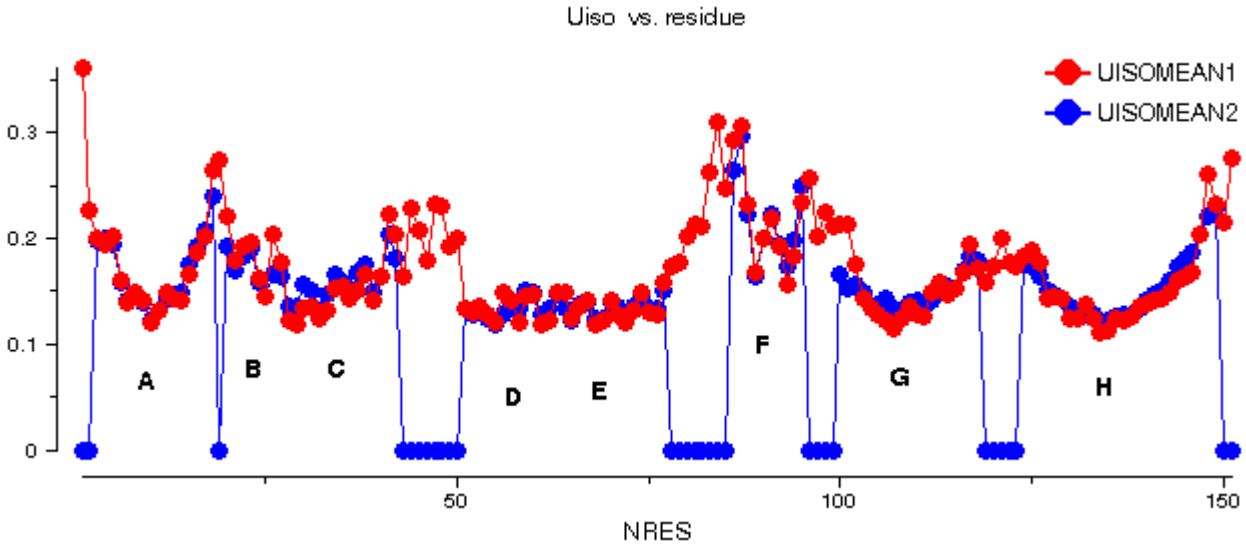
I have also fitted TLS parameters to the myoglobin structure. Following the results of the Delta-matrix analysis given above, I have used 6 TLS groups based on the alpha helices, with helices D and E being treated as a single group, and likewise helices G and H:

```
TLS
RANGE   'A   3.' 'A  18.'  FIT  MNCH

TLS
RANGE   'A  20.' 'A  35.'  FIT  MNCH

TLS
RANGE   'A  36.' 'A  42.'  FIT  MNCH
```

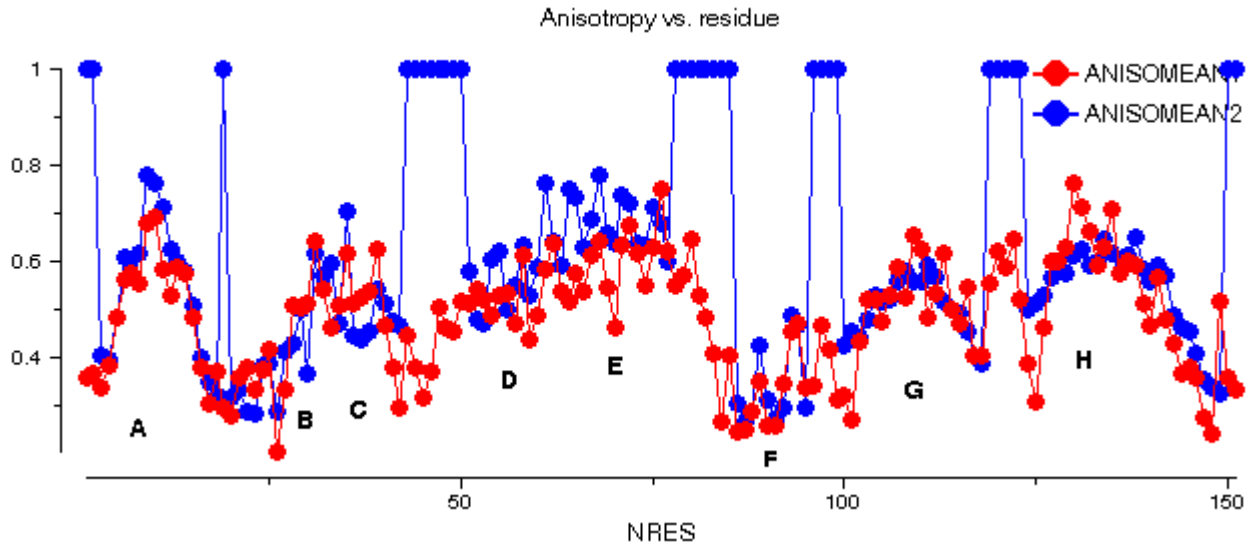
```
TLS
RANGE  'A  51.' 'A  57.'  FIT  MNCH
RANGE  'A  58.' 'A  77.'  FIT  MNCH

TLS
RANGE  'A  86.' 'A  95.'  FIT  MNCH

TLS
RANGE  'A 100.' 'A 118.'  FIT  MNCH
RANGE  'A 124.' 'A 149.'  FIT  MNCH
```

The resultant TLS parameters are used to calculate ADPs which are compared to the refined ADPs as follows:

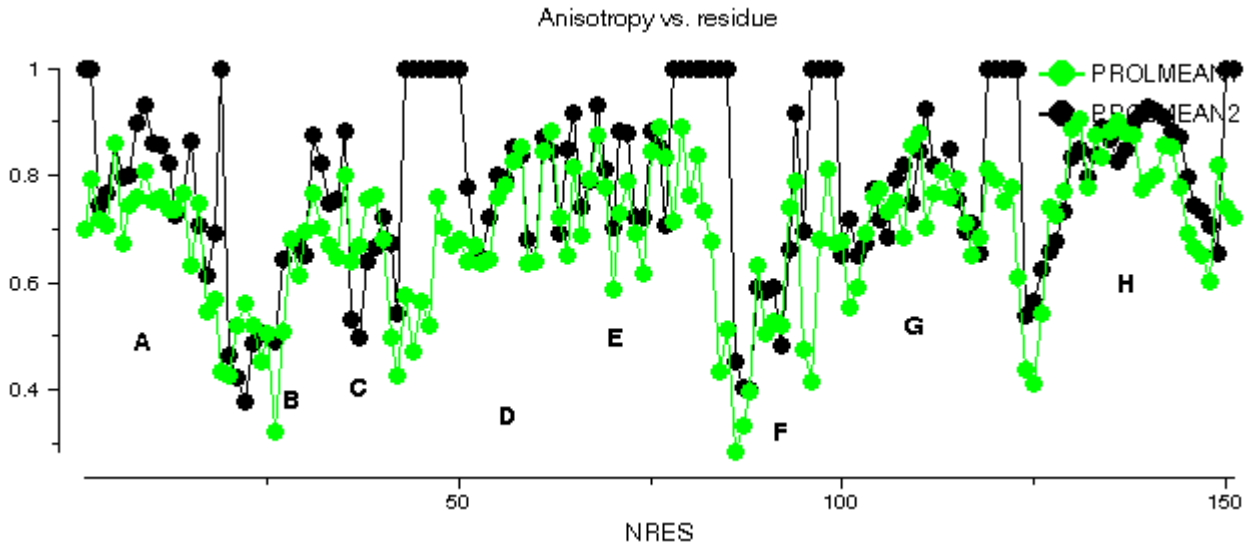
myoglobin-CO: 1a6g



myoglobin-CO: 1a6g



myoglobin-CO: 1a6g



The fit is clearly very good for both Uiso and the anisotropic components (residues with UISOMEAN2 = 0 and ANISOMEAN2 = PROLMEAN2 = 1 are those not included in any TLS group).

A more detailed rigid-body analysis of 1a6g was given by Vojtechovsky *et al.* (1999) who fitted TLS parameters to helices E and F, as well as to the heme group. They concluded on the basis of this fit that helix F and the heme group had ADPs consistent with rigid-body displacements, while helix E is less well described by a rigid body model.

These authors gave TLS parameters for helix F (and the heme group) which can be compared with the present results (analysis of the TLS tensors was done using the CCP4 program TLSANL):

	T axes and eigenvalues (A**2)				L axes and eigenvalues (deg**2)			
Vojtechovsky et al.	[-0.20,	-0.25,	0.95]	0.18	[0.07,	-0.16,	0.98]	27.2
	[0.06,	-0.97,	-0.24]	0.09	[0.19,	-0.97,	-0.17]	4.3
	[0.98,	0.01,	0.21]	0.08	[0.98,	0.20,	-0.04]	0.6
This study:	[-0.26,	-0.30,	0.92]	0.21	[0.15,	-0.11,	0.98]	34.1
	[0.00,	-0.95,	-0.31]	0.09	[0.35,	-0.92,	-0.15]	6.7
	[0.97,	-0.08,	0.25]	0.15	[0.93,	0.37,	-0.10]	-2.0

There are some discrepancies due to differences in the atom selection used for the TLS groups, but the overall picture is the same. (The negative value for the 3rd eigenvalue of **L** is unphysical, but allowed by the fitting procedure, and illustrates that there is some over-fitting.) In particular, there is a dominating libration parallel to the Z axis, and approximately parallel to the helix axis. These axes may be displayed using the AXES option of TLSANL.

Program usage

The program takes two input files with logical names XYZIN and TLSIN. XYZIN is a standard PDB file, with ADPs recorded on ANISOU lines. The TLSIN file is used for atom selection: often it is used to define TLS groups, but it can be used more generally. If the RIGIDBODY option is used, then a postscript file PSOUT is written containing the Delta-matrix plot. If the FITTLS option is used for fitting TLS groups to the input ADPs, then two

more files are written out: TLSOUT contains the fitted TLS parameters, XYZOUT contains residual ADPs (the difference in the input ADPs and those derived from the TLS parameters).

In the version released with 4.1, ANISOANL can only be run from the command line. However, a task interface for ccp4i is in preparation, and will be made available soon. For full details of the program, see the distributed documentation.

References

1. E.A.Merritt *Acta Cryst.*, **D55**, 1109 (1999)
2. R.E.Rosenfield, K.N.Trueblood and J.D.Dunitz, *Acta Cryst*, **A34**, 828 (1978)
3. T.R.Schneider, *Proc. CCP4 Study Weekend*, 133 (1996).
4. V.Schomaker and K.N.Trueblood, *Acta Cryst.*, **B24**, 63 (1968)
5. J.Vojtechovsky, K.Chu, J.Berendzen, R.M.Sweet and I.Schlichting, *Biophysical Journal*, **77**, 2153 (1999).

LAPACK in CCP4 version 4.1

Peter Briggs
CCP4 Daresbury Laboratory
p.j.briggs@ccp4.ac.uk

Introduction

As of version 4.1, CCP4 now includes the option of using the LAPACK linear algebra package. This article is intended to give some background information about LAPACK and the functionality it provides, as well some details of how CCP4 can be configured at installation time to use it. Links to relevant documentation are given, and a simple example using LAPACK in the CCP4 program MLPHARE is provided as a practical demonstration.

What is LAPACK?

LAPACK stands for **L**inear **A**lgebra **Pack**age [1], and is a collection of standardised subroutines for various mathematical operations, for example solving systems of linear equations or eigenvector problems. Some systems or system configurations include libraries which incorporate the LAPACK routines; for those systems which don't, the nessary source code can be obtain from the NetLib archive [2].

LAPACK is based on the BLAS (**B**asic **L**inear **Al**gebra **S**ubprograms) [3], another set of standardised subroutines and functions which deal with vector and matrix operations. Again, many systems provide a vendor version of the BLAS routines; if not then a ``reference'' BLAS can be obtained from NetLib. Since the vendor BLAS have been optimised for a specific system these are expected to be more efficient than the reference version. On systems without a vendor BLAS (or for those people not wishing for whatever reason to use vendor BLAS) freely available projects such as ATLAS [4] can be used to optimise BLAS performance.

LAPACK is a freely-available software package which can be incorporated into other software packages (including commercial ones) provided that proper credit is given to the authors. To reference LAPACK in a scientific publication you should cite the LAPACK Users' Guide [1]. For more details it is recommended that you consult the LAPACK FAQ [5].

LAPACK in CCP4 4.1

As of version 4.1, CCP4 now includes the option of linking in the LAPACK libraries. This has to be done at install time, by running the main configure with an extra option **--with-lapack**.

This latest release includes the source code for both LAPACK 3.0 and the reference BLAS, both obtained from NetLib - but configure should first search for ``native'' LAPACK libraries on your system, and will use these preferentially if found. If it cannot find a LAPACK library then it will search for native/vendor BLAS libraries and use these in preference to the reference BLAS, since they should be more efficient. Only if neither are found will the reference BLAS be built.

This behaviour can be over-ridden using the **--with-lapack=FORCE** option with configure. This forces building of the reference BLAS and LAPACK routines. Using the reference BLAS can be expected to result in poorer performance (in terms of speed) than when using vendor BLAS.

More information can be found in the CCP4 installation document [6] and in the MODLIB documentation [7] (which includes links to vendor BLAS and LAPACK libraries for some systems).

It should be noted that LAPACK provides routines for real, double precision, complex and double complex variables, and while the source code for all four precisions is supplied with CCP4 4.1, only real, double and complex are actually built under the **--with-lapack** option. For non-IEEE compliant machines there is one further issue which is addressed below.

Once CCP4 has been configured to use the LAPACK libraries, the XLAPACK_LIB variable in the Makefiles should show how to reference them for linking purposes - for example:

```
XLAPACK_LIB = -L/ccpdisk/xtal/ccp4-4.1/lib/lapack -llapack -L/usr/lib -lblas
```

None of the existing CCP4 libraries or programs currently use LAPACK - its inclusion at this point is to allow developers to take advantage of the routines in their programs in future. It is expected that future releases of the suite will use LAPACK routines, for example to replace some of the existing MODLIB routines, and at this point LAPACK will become a standard part of the CCP4 installation.

Quick Overview of the LAPACK Functionality

LAPACK is capable of solving systems of linear equations, linear least squares problems, eigenvalue problems and singular value problems. It is also able to handle many associated computations, such as matrix factorizations or estimating condition numbers.

The routines themselves are divided into three sets:

- **Driver routines** for solving standard types of problem
- **Computational routines** for distinct computational tasks
- **Auxiliary routines** to perform subtasks or low-level computations.

Typically driver routines will call a sequence of computational routines, and for standard problems it is likely that there will be a suitable driver routine which can be used directly without needing to deal with the computational routines underneath.

There are various ways of quickly identifying a suitable driver routine for a particular problem. I would recommend the LAPACK User Guide [1] as the best starting point:

- The user guide contents can be used to find lists of suitable routines for each problem quickly and easily; see http://www.netlib.org/lapack/lug/node1.html
- The user guide also contains a combined index of the driver and computational routines at http://www.netlib.org/lapack/lug/node142.html, which can be used as a quick reference.

However the User Guide appears to stop short of giving a full specification of the subroutine arguments, so at this stage it is necessary to refer to the comments in the

source code for the specific routine, for example by looking in the directory *$CCP4/lib/lapack/src/*, or by directly accessing the source code on the web (which can be done using the links from the LAPACK FAQ at http://www.netlib.org/lapack/faq.html#1.15.)

It may be that you have a particular problem which is not addressed by an existing driver routine. It should be noted however that taken together the computational routines can perform a much wider range of tasks than represented by the driver routines alone - so in these cases it should be possible to create your own "custom" driver. The computational routines are documented with the same level of detail as the drivers, and this documentation can be accessed in the same ways as outlined above.

An Example of using LAPACK in CCP4: MLPHARE

In principle many programs in the CCP4 suite contain subroutines which could be replaced by calls to LAPACK routines; for example MLPHARE (heavy atom refinement and phase calculation program) uses the following:

- EIGN1M: find the eigenvalues and vectors of a real symmetric matrix
- MATSOL: uses a modified Cholesky method to solve the matrix system A**x**=**b**
- MATIN1: matrix inversion with accompanying solution of linear equations

In practice replacing these routines means creating wrappers to prepare input for the appropriate LAPACK routine(s) and then performing any necessary conversion of the LAPACK output into the form expected by the calling subprogram. In many cases this is likely to result in a disproportionate amount of additional code.

For demonstration purposes however a version of MLPHARE was prepared in which the subroutine EIGN1M was replaced by a new routine using the LAPACK routine DSYEVR to perform the same task. DSYEVR computes selected eigenvalues and, optionally, eigenvectors of a real symmetric matrix using "Reasonably Robust Representations" (RRR).

The full code of the altered program is not reproduced here, but a patch with the replacement routine EIGN1M_W_LAPACK can be accessed at http://www.ccp4.ac.uk/newsletter39/04_mlphare_patch.f. It should be possible to compile and run a version of MLPHARE with this patch under CCP4 4.1, provided that it has been configured to use LAPACK as described previously (though see the comments below regarding installation on non-IEEE compliant machines).

Eigenvalues and eigenvectors of the B-factor matrix are only calculated in MLPHARE when anisotropic B factors are supplied. Comparison of the output from the two versions in all the test cases gave the same sets of eigenvalues. For the simple case where the anisotropic B factor coefficients are estimated from the isotropic value <B> using:

$B_1 = B_4 = B_6 = \langle B \rangle$
$B_2 = B_3 = B_5 = 0.0.$

the eigenvalues are three-fold degenerate and equal to <B> - so any set of three linearly independent vectors forms a valid set of eigenvectors. In this case the sets of eigenvectors from the two versions differed simply as a consequence of using different algorithms.

Once the anisotropic B estimate is perturbed slightly (for example by setting B_3 = 1.0) to break the degeneracy, then the two versions reassuringly give identical sets of eigenvalues and vectors.

Installation Issues for non-IEEE Compliant Machines

There is at least one installation issue which isn't yet automatically resolved by configure: the routine ILAENV, which tests for (amongst other things) IEEE-754 compliance for Nan and infinity arithmetic. If LAPACK is being installed on a non-IEEE compliant machine then these tests will fail at run-time and cause a program crash.

Once it has been established that you are installing on a non-IEEE compliant machine then the following change needs to be made to the ILAENV source code, *$CCP4/lib/lapack/src/ilaenv.f*:

```
diff ilaenv-dist.f ilaenv-non-ieee.f
527,528c527,528
< C     ILAENV = 0
<       ILAENV = 1
---
>       ILAENV = 0
> C     ILAENV = 1
538,539c538,539
< C     ILAENV = 0
<       ILAENV = 1
---
>       ILAENV = 0
> C     ILAENV = 1
```

This is necessary at least for on a DEC Alpha running OSF1 v4.0 - for instance, on this platform the call to DSYEVR fails in the MLPHARE example above unless the change is made.

Note that this is not strictly a bug: more background information about this issue can be found in section 6.1.4 of the LAPACK "Quick Installation Guide for Unix Systems" [8]; in future the CCP4 install should take care of this problem automatically.

Summary

As of CCP4 4.1 it possible to make use of the LAPACK linear algebra library as part of the suite. To enable LAPACK within CCP4, run the main configure script with the **--with-lapack** option.

LAPACK offers a robust and comprehensive set of routines which can be used to solve a number of standard numerical problems. In future it is envisaged that LAPACK will be a standard part of the CCP4 installation, and it is hoped that developers will take advantage of the routines to develop new code more quickly.

Finally, the LAPACK installation under CCP4 is still relatively untested - so I would be grateful to hear from anyone who can give me feedback (good or bad) about installing and using the routines. Please e-mail me at p.j.briggs@ccp4.dl.ac.uk if you have any comments.

References

[1] Anderson, E. and Bai, Z. and Bischof, C. and Blackford, S. and Demmel, J. and Dongarra, J. and Du Croz, J. and Greenbaum, A. and Hammarling, S. and McKenney, A. and Sorensen, D., *LAPACK Users' Guide, Third Edition, 1999, pub. Society for Industrial and Applied Mathematics, Philadelphia, PA, ISBN 0-89871-447-8 (paperback)*
A HTML version of the LAPACK 3.0 Users Guide can be found at *http://www.netlib.org/lapack/lug/lapack_lug.html*

[2] Netlib web address: http://www.netlib.org

[3] BLAS: http://www.netlib.org/blas/

[4] Automatically Tuned Linear Algebra Software (ATLAS): http://www.netlib.org/atlas/

[5] LAPACK FAQ: http://www.netlib.org/lapack/faq.html

[6] CCP4 installation document: http://www.ccp4.ac.uk/dist/INSTALL.html), section E "Configure Options"
The local version is in $CCP4/INSTALL.html.

[7] CCP4 MODLIB documentation: http://www.ccp4.ac.uk/dist/html/modlib.html), section 2 "Information on BLAS and LAPACK routines".
The local version is in $CCP4/html/modlib.html.

[8] LAPACK Working Note 81: Quick Installation Guide for LAPACK on Unix Systems can be downloaded as a postscript file from http://www.netlib.org/lapack/lawns/lawn81.ps, but it should be noted that the routines distributed with CCP4 4.1 form only a subset of the full LAPACK distribution available from NetLib.
The full set of working notes are available via http://www.netlib.org/lapack/lawns/index.html

A system for storing annotated diffraction data

John Badger

Structural GenomiX, 10505 Roselle St., San Diego, CA 92121, USA

1. Introduction

The emergence of several public consortia (for example, the recent NIH initiative) and privately funded groups engaged in high throughput protein crystallography highlights a long-standing need to develop simple, automated and standardized mechanisms for annotating and storing *all* sets of experimental data that contribute to the solution of a macromolecular crystal structure.

A brief survey of the current situation illustrates the deficiencies of the mechanisms available at the present time.

Diffraction data in the public domain

The maintainers of the Protein Data Bank do not usually receive any diffraction data other than the data set used to refine the deposited structure and, in many cases, do not receive any diffraction data at all. In the PDB snapshot taken at October 1, 2000 there were only 4,129 structure factor files for the 10,900 structures determined by diffraction techniques. This shortfall in the number of diffraction data sets is not solely a problem with legacy structures - for the 1601 structures solved by x-ray and released between January 1, 2000 and October 1, 2000 only 820 sets of experimental data were available.

Most of the structure factor data at the PDB is stored in very simple mmCIF files[1]. Data annotation (i.e. information on merging R-values, redundancy etc) is provided via REMARK fields in the associated PDB *coordinate* files. If completed by the depositor, this information provides a reasonable summary of overall data quality. However, it is unavoidably limited, in both accuracy and scope, by the fact that it is manually entered into a PDB deposition interface or deposition text file.

One can only speculate on what becomes of the multitudes of heavy atom derivative and anomalous scattering data sets collected in individual laboratories but which are not normally deposited with the PDB. It seems more than optimistic to believe that many of these data files remain available for more than a few years after the date that the structure was solved. Furthermore, even when diffraction data sets are securely stored, there may not be sufficient information on their content and accuracy for them to be useful.

Diffraction data in the working laboratory

As *working* formats, the two most widely used at the present time appear to be the CCP4/MTZ format and the reflection file format used by programs in the X-PLOR/CNS/CNX lineage. Obviously, several other more *ad hoc* formats (usually simple ASCII reflection lists) are used by other individual programs. As working formats, both the CCP4/MTZ and X-PLOR/CNS/CNX formats are quite satisfactory and , in particular, the

mapping of data types provides some safeguards against inappropriate use. The major disadvantage of these formats for data archival is that they do not provide space for data annotation. This means that information on the reliability and purpose of data stored in these types of file can easily become lost over time.

Key issues

The discussion above points out the lack of any public domain 'off the shelf' standards or data models for maintaining macromolecular structure factor data. The most significant issues are as follows:

1. Only the data set against which the structure was refined is securely stored at the PDB.
2. The data annotation is disconnected from the file containing the reflection list.
3. The data annotation is limited to key summary information.

In the case of our own organization (Structural GenomiX) many dozens of data sets, predominantly involving anomalous diffraction for MAD phasing, have been obtained in the last few months. There is an urgent need to develop not only a conceptual standard for annotating these data but also to provide a practical implementation, compatible with the output diagnostics of the current data processing programs.

This communication is intended to stimulate interest in this subject and describes our own efforts towards solving the problem of diffraction data storage within our internal data base. Diffraction data files developed along these lines may well be useful for other groups interested in creating local archives of well-annotated diffraction data. It is *not* our intention to encourage structure depositors to provide diffraction data files of this type to the PDB since the development of new formats and mechanisms for the annotation of diffraction data at the PDB would require a considerable public discussion, involving a number of organizations.

2. Design principles

Our set of principles for developing a reporting standard for diffraction data are that:

1. Data annotation should include (but not be limited to) the data quality indices normally reported in journal publications.
2. The set of data files associated with a single structure entry should provide all information (other than amino acid sequence) that is needed to solve the structure.
3. The data annotation should reside within the same file as the reflection data in order to avoid any possibility of information loss.
4. The data annotation should provide sufficient information for an experienced crystallographer to detect problems and limitations in the data set.
5. The diffraction data should be presented in a layout that is easily translatable to other formats for use with current crystallographic sofware.

Items (1), (2) and (3) above require little comment.

The rationale behind item (4) is that we anticipate that the processing of raw (frame) data will increasingly become the domain of technical specialists using semi-automated procedures. The reduced data will then be provided to the crystallographer responsible for

solving the structure. This scenario is likely to become very common for commercial organizations and public consortia engaged in high-throughput crystallography.

Item (5) expresses a practical consideration that most crystallographers use a variety of software, which require data in different formats. This means that the storage format should be easily parsable, with straightforward extraction of relevent data items.

3. Conceptual Design

Basic approach

We have taken the point of view that each set of reflections used in a structure determination should be represented in the same way and should be stored as a separate data set. Specifically, this means that we have not employed any special reflection types to represent data obtained from MAD experiments or from derivative crystals. This viewpoint contrasts with conventional mmCIF view which has special category groups (*_phasing_set_refln*, *phasing_mir_der_refln*, etc) for reflections used in different types of experimental phasing.

We have taken this approach for reasons of conciseness and to avoid some awkwardness with these mmCIF categories for representing experimental phase information derived from *multiple* sources (i.e. phase information derived by both anomalous diffraction and isomorphous replacement). Furthermore, in some very common structure determination scenarios (for example, two-wavelength MAD from Se-Met crystals, where the 'best' data set plays the role of native in structure refinement) there is no 'native' data set in the conventional sense. In this case it seems undesirable to arbitrarily classify one of the data sets as 'native'. In fact, a survey of the PDB holdings to 1 October 2000 reveals that there are already 108 coordinate sets corresponding to Se-Met proteins.

Annotation

The conceptual design of the file for diffraction data storage involves selecting what information should be reported and the form in which it should be expressed. Figure 1 provides an example of the current content of the file that we have developed.

Our data files use mmCIF categories and records (mmCIF) to provide a well-defined annotation. The decision to use the mmCIF dictionary as the basis for our diffraction data files is motivated by the fact that it contains most of required annotation items and expresses them in a way that is usually fairly obvious to human readers.

Where possible we have used the existing mmCIF definitions for these diffraction data files; where no mmCIF record was available we have developed our own categories and records. In fact, there are 15 locally defined records from the 59 records normally used in our diffraction data files. The need to develop a significant number of new record types should not be considered surprising for a 'real-world' application developed some years after most of the the conceptual development of the mmCIF dictionary. For comparison, it is interesting to note that the current RCSB PDB x-ray deposition form contains as many as 30 locally defined items (i.e. items with prefixes '*ndb*' or '*rcsb*') out of a total of 107 fields. We anticipate that additional records will be required in our diffraction data files as new data quality metrics emerge and/or further experience with reporting diffraction data in this format reveals a need to provide additional information.

The addition of new mmCIF records has mostly been necessary to report data statistics and reflection values relating to the use of anomalous scattering information. We have introduced *_refln.sgx_Fplus_meas_au*, *_refln.sgx_Fplus_meas_sigma_au*, *_refln.sgx_Fminus_meas_au* and *_refln.sgx_Fminus_meas_sigma_au* records to capture Friedel pairs as separate data items. Since we frequently want to extract anomalous differences from our data sets, this layout is much more convenient than storing all these data within the single column of *_refln.F_meas_au* data. In this case *_refln.F_meas_au* and *_refln.F_meas_sigma_au* represents the merged reflection and this representation allows the merged value of the structure factor to be contained within the same data loop as the Friedel pairs.

The records in the new *_reflns_info* group are used to provide information on the use of the data set for phase determination; the *_reflns_info.sgx_der_type* tag is used to record whether the sample contains heavy atoms and *_reflns_info.sgx_se* reports whether the crystal contains Se atoms.

A few mandatory tags that have no real use within the context of reflection data at Structural GenomiX are included for formal compliance with the mmCIF dictionary.

Fig 1. An illustrative example of a mmCIF file containing diffraction data

```
data_example.1
_reflns.entry_id                     'example.1'
_cell.entry_id                       'example.1'
_symmetry.entry_id                   'example.1'
_reflns_info.id                      'example.1'
_diffrn.id                           1
_diffrn_radiation_wavelength.id      1
_diffrn_radiation_wavelength.wavelength 1.20
_reflns_info.sgx_der_type            'none'
_reflns_info.sgx_se                  'yes'
_reflns.sgx_merged_anom              'no'
_reflns.ndb_netI_over_av_sigmaI      13.9
_reflns.ndb_Rmerge_I_obs             0.113
_reflns.sgx_Rmerge_anom              0.052
_reflns.sgx_number_measured          228381
_reflns.percent_possible_obs         96.6
_reflns.ndb_redundancy               6.8
_reflns.sgx_anom_completeness        95.4
 loop_
_reflns_shell.d_res_low
_reflns_shell.d_res_high
_reflns_shell.meanI_over_sigI_obs
_reflns_shell.Rmerge_I_obs
_reflns_shell.sgx_Rmerge_anom
_reflns_shell.number_measured_all
_reflns_shell.number_unique_all
_reflns_shell.percent_possible_all
_reflns_shell.ndb_redundancy
_reflns_shell.sgx_anom_completeness
   .     9.49   20.5   0.076   0.058     4973     1039     85.5      4.8     82.7
 9.49    6.71   22.0   0.073   0.052    12969     2066     93.4      6.3     96.9
 6.71    5.48   20.5   0.094   0.058    17289     2601     95.2      6.6     96.8
 5.48    4.74   19.9   0.104   0.047    20646     3018     95.7      6.8     96.2
 4.74    4.24   19.2   0.103   0.041    23571     3419     96.2      6.9     96.2
 4.24    3.87   16.6   0.120   0.045    26387     3759     96.5      7.0     96.6
 3.87    3.59   13.6   0.141   0.052    28558     4060     96.6      7.0     96.2
 3.59    3.35   10.5   0.159   0.058    30344     4340     96.7      7.0     95.5
 3.35    3.16    7.6   0.201   0.072    31541     4599     96.7      6.9     95.1
```

```
 3.16   3.00    5.1   0.278   0.103    32103     4826     96.6      6.7     94.4
_reflns.B_iso_Wilson_estimate        75.366
_cell.length_a   63.131
_cell.length_b   84.626
_cell.length_c  316.239
_cell.angle_alpha   90.000
_cell.angle_beta    90.000
_cell.angle_gamma   90.000
_symmetry.Int_Tables_number     18
_symmetry.space_group_name_H-M 'P 21 21 2'
_reflns.number_all   33714
_reflns.d_resolution_low    20.00
_reflns.d_resolution_high    3.00
 loop_
_refln.wavelength_id
_refln.crystal_id
_refln.scale_group_code
_refln.index_h
_refln.index_k
_refln.index_l
_refln.F_meas_au
_refln.F_meas_sigma_au
_refln.sgx_Fplus_meas_au
_refln.sgx_Fplus_meas_sigma_au
_refln.sgx_Fminus_meas_au
_refln.sgx_Fminus_meas_sigma_au
1 1 1     0    0   16   172.6586     6.8931   172.4213     6.9680   172.8959
11.8956
1 1 1     0    0   17    14.2712     6.6891    11.7388     8.2799    16.8037
10.5080

...plus 33714 more rows of reflection data.....
```

The focus of this communication is to describe a mechanism for reporting the data processing statistics that are available directly following data reduction. However, we are are using the same type of file to store the additional reflection information that becomes available at the completion of the phase determination and refinement processes. We are using the new records *_refln.sgx_HL_A*, *_refln.sgx_HL_B*, *_refln.sgx_HL_C*, *_refln.sgx_HL_D* to denote Hendrickson-Lattman phase probability coefficients regardless of their experimental source since phase probabilities only seem to appear in the *_phasing_MIR_der_refln* records in the mmCIF dictionary. Thus, the set of additional *_refln* records thatwe are employing to report phase determination and refinement information is:

```
_refln.sgx_Fmap
_refln.fom
_refln.phase_meas
_refln.sgx_HL_A
_refln.sgx_HL_B
_refln.sgx_HL_C
_refln.sgx_HL_D
_refln.status
```

The first seven fields in the list above provide a means to maintain complete experimental phase information, allowing reproduction of the initial experimentally phased electron density map. The final field reports whether a reflection was used within the 'working' refinement data set or it was a part of the 'test' cross-validation data set.

4. Mechanism for creating mmCIF diffraction data files

The annotation provided in these files could be entered manually (by hand editing), although the process would be a little tedious.

Instead, we have developed a parser for automatically extracting most of the information from the reflection files and the output of data processing programs. Specifically, log files from CCP4/SCALA, CCP4/TRUNCATE and the MTZ reflection file are parsed to generate most of the content of the annotated diffraction data file. The EBI/CCP4 data harvesting system also provides a mechanism for extracting key information in standardized form from these programs. For our own database we considered it important to provide a mechanism of extracting data items absent from the data harvesting output, albeit at cost of the effort involved in writing and maintaining a parser for the CCP4/SCALA and CCP4/TRUNCATE log files.

Although the content of these diffraction data files appears neutral with respect to the data processing programs used, the CCP4/SCALA, log file provides a particularly convenient and complete report of data quality and the information provided in some fields of the data diffraction file would be difficult to capture from the output of certain other data reduction software.

5. Applications and related developments

This file format is presented as a practical and self-contained format in which all data sets and key information associated with a structure determination can be maintained over long periods of time. All that is required is that the set of data files are stored in secure directories or on permanent media such as CD-ROM.

We have developed a storage system within a relational database (Oracle) to provide access to our entire corpus of annotated crystallographic data and coordinates. This crystallographic database, containing consistently and fully annotated diffraction data, should facilitate a longer term scientific objective of evaluating and improving structure determination methodologies. For example, it is clear that the rate and ease of a structure determination is strongly associated with the quality of the first experimentally phased electron density map. However, to properly determine the data collection regimes that optimize the chances of getting a 'good' initial map from a given crystal in the presence of radiation damage (i.e. decisions relating to the trade-off between the numbers of wavelengths for MAD versus redundancy of data) requires complete and consistent information on the outcomes of a large number of structure determinations.

This general approach of crafting appropriately extended mmCIF files may also be used in other areas of macromolecular crystal structure determination. For example, we have also developed an automated system that validates our structures using the SFCHECK V 5.3.4[2] program and the PROCHECK V 3.5[3] software (i.e. versions of this software incorporated into the CCP4 4.0 release) . This system collects key results from the output of these programs into a single mmCIF coordinate file, which provides a convenient and consistently annotated report of structure quality.

The virtue of these approaches is that they establish a highly consistent set of self-contained structure files without requiring the crystallographer to perform tedious clerical tasks. Time and expertise is better spent studying the structure and carrying out follow-up crystallography, biochemistry and molecular modelling studies.

6. Acknowledgements

I wish to thank Janet Newman, Tom Peat, Eric de la Fortelle, Sarah Dry and Phil Bourne for comments on this manuscript. The diffraction data that is being used with this system is the result of work by the Crystallography group at Structural GenomiX.

7. References and notes

1. These files are non-standard in that the mandatory *_refln.scale_group_code* field is usually absent and the *_refln.status field* (used to denote free and working data sets) is usually represented by a numerical value '0' or '1' rather than the characters 'o' or 'f' defined in the mmCIF dictionary. The CCP4/MTZ2VARIOUS tool creates the *_refln.status field* according to the mmCIF dictionary definition. However it appears that the mmCIF parsers in both the SFCHECK[2] and CNX programs read numeric rather than character codes.

2. A.A. Vaguine, J. Richelle, and S.J. Wodak. (1999). *Acta Cryst.* **D55**, 191-205.

3. R.A. Laskowski, M.W. MacArthur, D.S. Moss and J.M. Thornton. (1993). *J. Appl. Cryst.*, **26**, 283-291.

CCP4 Molecular Graphics

Liz Potterton March 2001

Introduction

CCP4 is planning to develop a molecular graphics system which can be integrated with software to perform the many functions required by crystallographers, such as structure solution, structure analysis and comparison and creating presentation graphics. Initially we will be concentrating on creating a 'core' system which has the basic functionality to display molecules and maps. The core system will also provide a library of routines which any programmer can use to link their software into the molecular graphics. From the programmers perspective the molecular graphics would just be *molgraphlib* to complement mtzlib, symlib etc. Since the molecular graphics system will be freely available like other CCP4 software this means that any scientific developer can use *molgraphlib* to make their programs graphical and we hope, this way, that many people may contribute to the project. But we do expect CCP4 staff and collaborators to do a lot of the work to provide the essential scientific functionality.

Why another molecular graphics program?

Crystallographic model building can be semi-automated (rather than reliably totally automated) and the appropriate methods are very data dependent. So it is useful to tie the model building into a graphics system which can co-opt the crystallographer to help out with the tricky decisions. To date the development of automated methods has been restricted to laboratories and companies with their own molecular graphics system but an open CCP4 system could change this.
It would make things easier for the average crystallographer if one molecular graphics system provided an interface to a full range of functionality - model building, model analysis, interaction with databases, presentation graphics etc.. The design of the proposed system should make it possible to build in all the required functionality. CCP4 is in a better position than most individual groups to provide the resources for development and maintenance and to guarantee long term support for a large scale software package.

Progress So Far

There are several projects underway in CCP4 which will tie in with the molecular graphics. Eugene Krissenel, who works for CCP4 based in the MSD group at the , is developing a library to handle coordinate data . See here for progress on interfacing to the existing RWBROOK libraries, but this project involves much more than reading and writing PDB and mmCIF files - it will perform data validation atom, selection and molecular editing and will provide the basis for all functionality relating to molecule structures. Kevin Cowtan's Clipper project is providing a similar, basic library for handling experimental data and there are plans afoot to provide a similar map handling library.
Martin Noble at Oxford has volunteered to develop the graphics part of the system. Martin has written Aesop, a program for displaying molecular structures. CCP4 is going to hire two new developers to work on the the molecular graphics project. I will be coordinating the project and probably concentrating on the user interface.

When Will It Be Available?

We are more concerned to build a solid basis rather than produce something quickly so it will be probably be at least a year before anything is available. But we are keen to get feedback from users so will make the package available as soon as it has the minimal functionality to be useful. This means that a first release will probably not do significantly more than a program like RasMol.

How to Get Involved

We should be advertising new posts shortly but if you are interested in working on the project please contact me (lizp@ysbl.york.ac.uk).
If you are interested in following the project and contributing to discussion or even contributing code then sign up to the bulletin board - by sending a "subscribe 3dccp4" message to Majordomo@dl.ac.uk.

ARP/wARP goes CCP4i

Anastassis Perrakis, Liz Potterton and Victor Lamzin

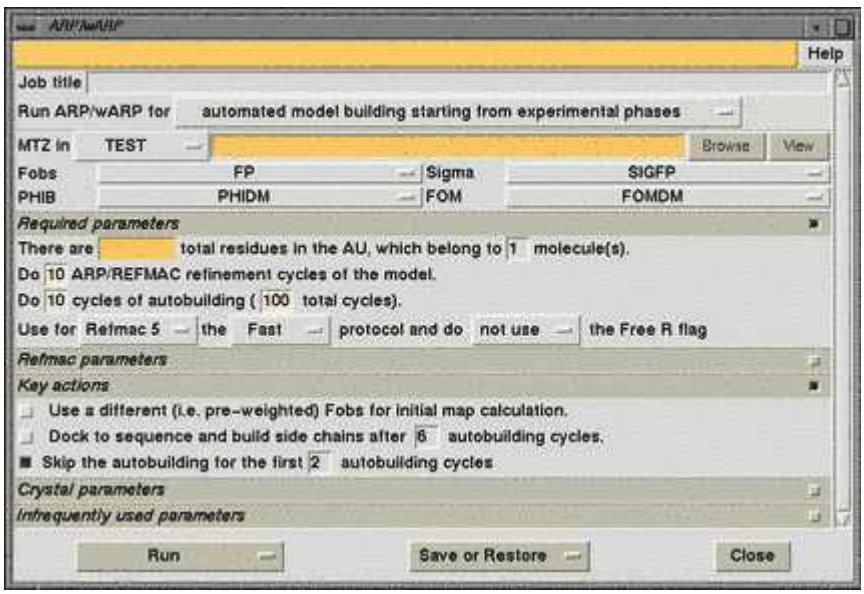
The ARP/wARP package, from version 5.0 onwards, has diverted from the classical way of executing CCP4 programs. Instead of relying on the user to edit command files and run them, arp_warp_setup.sh was extracting the necessary information from the user input and setting up a 'parameter' file (warp.par) which was further used by the scripts within the package. Such an approach was necessary, because of the complexity of the arp_warp.sh file.

Now ARP/wARP is back to the mainstream. The ARP/wARP CCP4i interface was designed based on the principles and libraries developed by Liz Potterton and the CCP4 staff. The ARP/wARP is based on CCP4i and looks like CCP4i, however it is not exactly CCP4i. The underlying architecture of the 'warp.par' file and the 'arp_warp.sh' script is maintained for the GUIphobic folks out there. 'arp_warp_setup.sh' will go on working alright, but the CCP4i 'layer' is there to make running ARP/wARP even easier.

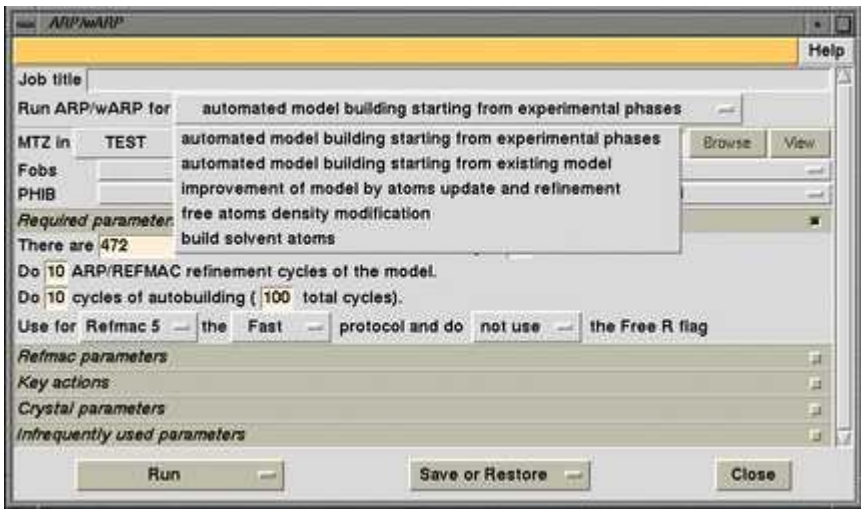
The ARP/wARP CCP4i interface exemplifies how a wide variety of CCP4 programs can be used together with other 'user-modules' (in this case the autobuilding programs) and command scripts under a common user-appearance, the CCP4i interface look-and-feel, in order to achieve a certain crystallographic goal, in our case combination of model building and refinement.

The main features of the ARP/wARP CCP4i interface include:

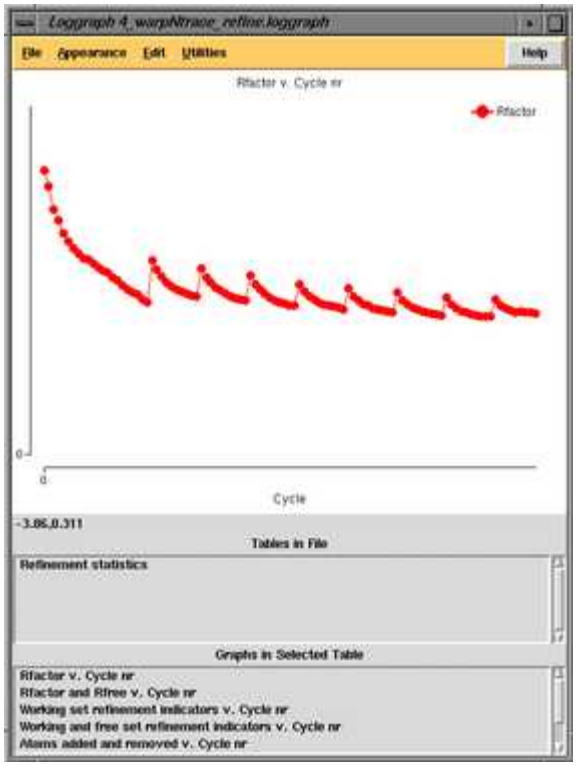
- Single button tracing: All you have to do is to browse for the name of the mtz file which has the amplitudes and phases, press 'Run' and ARP/wARP will deliver a pdb file with the model! There is some controversy regarding the number of residues input: ARP/wARP can suggest it to you, but given that you should at least know how big your protein is, this input will be obligatory for now! A new feature is that you can choose if you want to dock the autotraced parts to the known amino-acid sequence towards the last cycles of autobuilding. This way, you do not have to run the side chain docking as a standalone script and at the same time there is no effort spend on docking the initial, incomplete, models to the sequence.



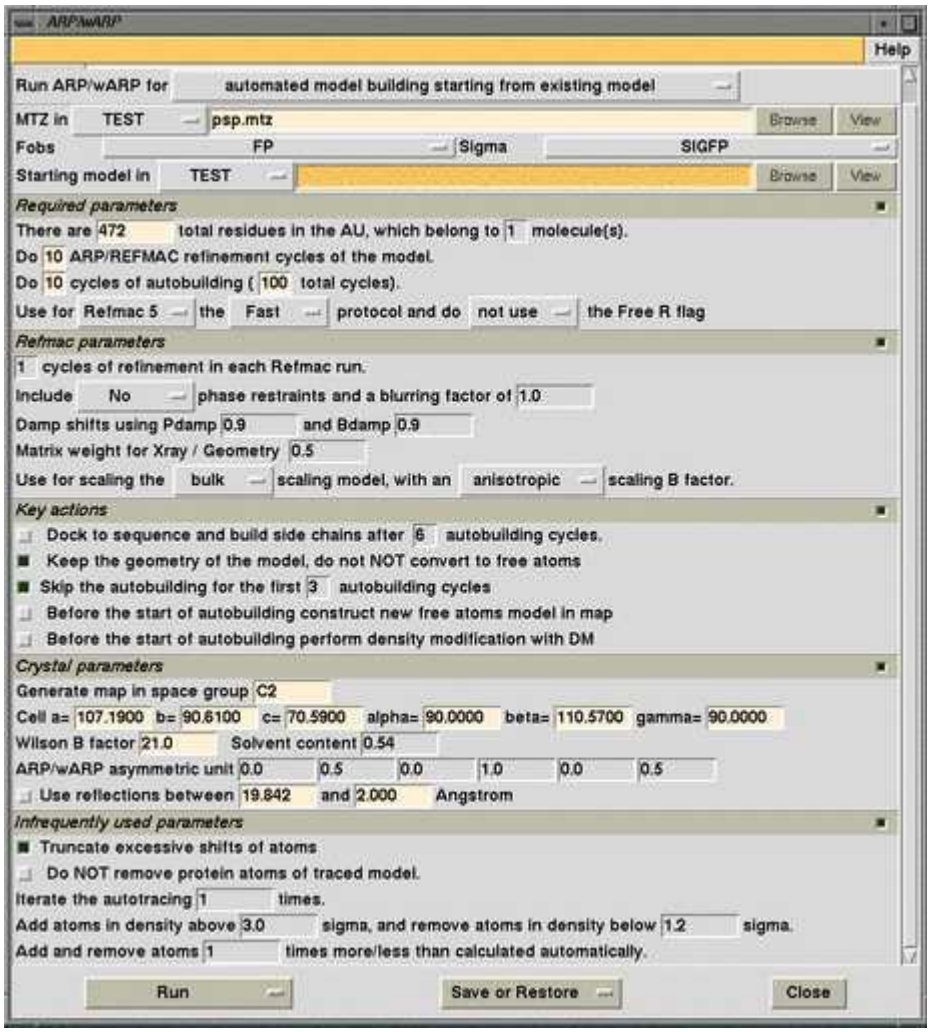
- There is simple choice between diverse protocols: Autobuilding from experimental phases, autobuilding starting from a pre-existing model, model refinement by atoms update, just free atoms refinement to improve a map or solvent building are the choices. Lots of things are made easy this way, especially 'start' files do not have any more cryptic names, but can be chosen from any directory.



- Apart from the usual 'screen output' of the arp_warp.sh script, which is now the 'Log file' for the CCP4i job, all other extensive log files from refinement, custom graph files, output PDB files and map files can be found under 'Output files' from the main CCP4i window. The graph files are now compatible with 'loggraph' of CCP4i and a few defaults presentations are chosen which are handy for quick inspection of the job.



- Although everything has defaults, you can customise almost all the parameters. It is a central point in the philosophy of the interface that nearly user-free operation must be possible, while retaining all the functionality that is desired by the expert user. The user can set preferred parameters for ARP specific things (like the sigma values for adding and deleting atoms), most of the available 'REFMAC' parameters, or modify the way the ARP/wARP protocol works (i.e. include solvent flattening before start, and others). Special care has been given to the treatment of the autobuilding when starting from an existing model. Some of the choices available specifically for this mode are extremely usefull when starting from 'nasty' molecular replacement solutions (more about that will be published in the CCP4 study weekend proceedings).



The ARP/wARP CCP4i interface will be available soon through the ARP/wARP release 5.2 from http://www.arp-warp.org. It will be compatible with CCP4 4.1 and will be included in the main CCP4 distribution from version 4.2 onwards.

Vector-Search Methods in Molecular Replacement

Carmen Álvarez-Rúa, Javier Borge and Santiago García-Granda.
Departamento de Química Física y Analítica
Facultad de Química. Universidad de Oviedo
C/ Julián Clavería, 8. 33006 Oviedo. SPAIN

Introduction

OVIONE is a computer program that uses a vector-search rotation function for macromolecular crystal-structure determination by the molecular-replacement method.

In order to determine the orientation of the search model in the crystal unit cell, a vector set obtained from the model is rotated through the asymmetric unit of the angular space. In each angular position an Image Seeking Function is evaluated. This function acts as a criterion of fit between the vector set from the search model and the observed Patterson map of the target structure, and it is expected to attain a maximum value at the correct orientation of the search model.

The rotation search is followed by a refinement of the highest peaks of the rotation function, that is also carried out in Patterson space.

Methodology

The use of ISFs (Image Seeking Functions) (Buerger, 1959) as rotation functions was proposed for the first time by Nordman (Nordman & Nakatsu, 1963), who used these functions as a criterion of fit between vector sets and Patterson maps. A new ISF proposed by Nordman, the ``weighted minimum-average function'' (Nordman, 1966; Schilling, 1970) was later implemented for the determination of the orientation of a known molecular fragment in the program ORIENT (Beurskens *et al.*, 1987) included in the DIRDIF system (Beurskens *et al.*, 1999), and widely used in crystallography of small molecules.

The same function has now been implemented in OVIONE (Álvarez-Rúa *et al.*, 2000), with some modifications that allow the use of this methodology in macromolecular crystallography (Borge *et al.*, 2000).

The rotation function algorithm in the program consists of the following steps:

- First, a calculated Patterson map of the search model is computed and a set of intramolecular vectors (the self-vector set) is extracted from it.
- An observed Patterson map is calculated from experimental data of the target crystal.
- An statistical analysis is performed to check if the problem presents the adequate conditions for the Image Seeking Function to work properly.
- In the next step, the rotation search is carried out. Currently, two ISFs are implemented in the program: the ``weighted minimum-average function'' (Nordman & Schilling, 1970) and the ``weighted sum function'' (Nordman, 1972).

- Finally, a refinement of the highest peaks of the rotation function is carried out by means of a minimization algorithm known as the ``downhill simplex method'' (Nelder & Mead, 1965).

Availability

Details about the methodology implemented in the program can be found at the OVIONE home page[1] together with some examples of application of the method.

The results of the program are written to an output file which contains the list of the Euler angles that represent the possible orientations of the search model. If required by the user, the program also rotates the atomic coordinates of the search model according to the highest peak from the rotation (and refinement) process and writes the result in a PDB-formatted file.

Since the molecular replacement process does not finish once the orientation of the model is determined (except for crystals with P1 symmetry) some procedures have been developped which act as an interface with other translation function programs, such as the version of AMoRe incorporated in the CCP4 package.

The last release of the program (OVIONE 1.1; March 2001) presents some new technical features, mainly:

- The amount of physical memory required by the program has been substantially reduced, which allows the treatment of proteins of bigger size.
- If required by the user, the observed Patterson map is now written to disk and stored. The program is able to read in again this map. This can be useful and time-saving in case the program is run several times, for instance, with different search models.
- In the same way, the self-vector set can be now calculated and stored. This avoids having to repeat the self-vector set selection process, which is one of the most time-consuming steps of the algorithm, in case the user wants to rerun the program under the same conditions, but using, for example, a finer angular grid around a possible orientation of the search model.

Acknowledgements

The authors wish to thank CCP4 for permission to incorporate in OVIONE some routines from the CCP4/Daresbury Laboratory libraries. Professor M. G. Rossmann is also thanked for allowing us to use some FFT routines from the Purdue Library of Programs.

This work was partially supported by CICYT (BQU2000-0219).

References

Álvarez-Rúa, C., Borge, J. & García-Granda, S. (2000). *J. Appl. Cryst.* **33**, 1436-1444.

Beurskens, P. T., Beurskens, G., Strumpel, M. & Nordman, C. E. (1987). *Patterson and Pattersons. Fifty years of the Patterson function*, edited by J. P. Glusker, B. K. Patterson & M. Rossy, pp. 356-367. New York: Oxford University Press.

Beurskens, P. T., Beurskens, G., de Gelder, R., García-Granda, S., Gould, R. O., Israël, R. & Smits, J. M. M. (1999). *The DIRDIF-99 program system*. Crystallography Laboratory, University of Nijmegen, The Netherlands.

Borge, J., Álvarez-Rúa, C. \& García-Granda, S. (2000). *Acta Cryst.* D**56**, 735-746.

Buerger, M. J. (1959). *Vector space and its application in crystal structure investigation*. New York: John Wiley & Sons, Inc.

Nelder, J. A. & Mead, R. (1965). *Computer Journal* **7**, 308-313.

Nordman, C. E. (1966). *Trans. Am. Crystallogr. Assoc.* **2**, 29-38.

Nordman, C. E. (1972). *Acta Cryst.* A**28**, 134-143.

Nordman, C. E. & Nakatsu, K. (1963). *J. Am. Chem. Soc.* **85**, 353-354.

Nordman, C. E. & Schilling, J. W. (1970). *Crystallographic Computing,* edited by F. R. Ahmed, pp. 110-114. Copenhagen: Munksgaard.

Schilling, J. W. (1970). *Crystallographic Computing,* edited by F. R. Ahmed, pp. 115-123. Copenhagen: Munksgaard.

MAPSLICER: an interactive viewer for contoured map sections

Peter Briggs
CCP4 Daresbury Laboratory
p.j.briggs@ccp4.ac.uk

Introduction

MAPSLICER is a prototype interactive viewer for contoured sections through CCP4 maps, ``inspired'' by the existing CCP4 program NPO. Although it doesn't offer anything like the range of functionality of NPO, it does have some nice features, principally its ease of use for the novice.

Starting Up

The MAPSLICER interface is loosely based on the ccp4i *loggraph* program - on a UNIX/LINUX machine it is started from the command line simply by typing

```
> mapslicer map_file_name
```

which will automatically load in *map_file_name*. You can also use

```
> mapslicer
```

and then use the file browser options to open the desired map. An example of the program displaying a section from a Patterson map is shown below in figure 1.

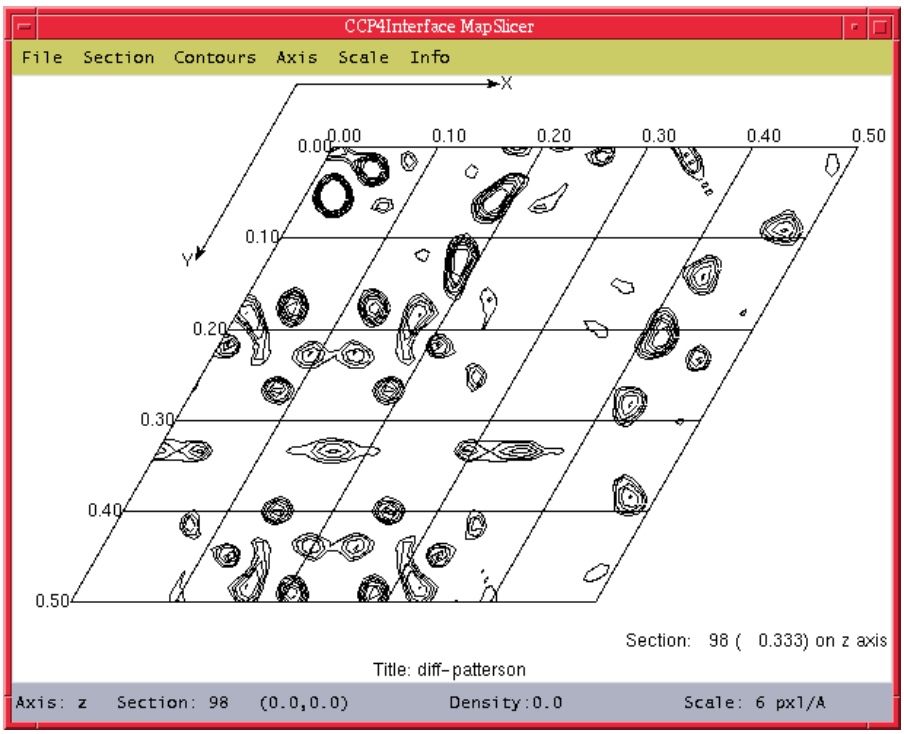


Figure 1: Screen-shot of MAPSLICER in action.

Current Features

MAPSLICER reads in a CCP4-format map file which is sectioned in any order, and holds the whole map in memory. Once the map is inside the program it is possible to view any section along any of the principle axes x, y or z, regardless of the order of the sectioning. It is easy to change between axes and to go directly to any section within the map.

Sections can be specified in either fractional units or grid units, as can the extent viewed. The contour levels can be specified interactively, as can the on-screen scaling, and the program also offers a (limited) print facility, allowing the section to be dumped to a postscript file. All these features are easily accessible from the menu bar at the top of the window.

MAPSLICER was originally conceived to make it easier to look at Harker sections, so there is also the option to "short-cut" to Harker sections. The only drawback here is that the Harker sections are derived from the "true" spacegroup, while the map has the Patterson spacegroup - so the user must enter the true spacegroup manually first.

A bar at the bottom of the window gives basic information about the current section. Cursor feedback tells you the fractional coordinates of a particular position within the section, as well as the density at that position. More detailed information about the map or the current section can be obtained from the "Info" menu option.

Problems and missing features

MAPSLICER is still very much prototypical, so there are bound to be some problems of varying degrees both during installation and at run-time. Some major bugs have already been fixed in time for the patch release of CCP4 (4.1.1).

Other problems are less critical (for example, occassionally strange behaviour of axis labels, and limited print options) and there are some missing features - for example, you can't overlay multiple sections from the same map ("slab" option), and there is no way of colouring contours or of importing coordinates (e.g. heavy atom vectors, atom positions etc). Over time these should be fixed - meanwhile bug reports and suggestions are welcome.

How to get and install it

MAPSLICER is a TCL/Tk script, but it also needs a special version of the TCL/Tk interpreter *wish*, called *ccp4mapwish* (short and sweet!) in order to function. The source code for *ccp4mapwish* has been included as part of the latest CCP4 release (4.1 and 4.1.1), but it is not installed by default yet because the installation procedure has not been widely tested, and so cannot be automatically performed reliably.

You can find the code for *ccp4mapwish* in the $CPROG/ccp4mapwish_ directory, along with a configure script which will try and build the interpreter and then install in the $CBIN directory along with the other CCP4 programs. Instructions can be found via the documentation in $CHTML/ccp4mapwish.

IMPORTANT: there have been a number of changes to both the ccp4mapwish.c source code and to the configure script between 4.1 and 4.1.1. These changes are required to successfully compile with Tcl/Tk 8.3 (the most recent and stable version), and also to fix a

bug with reading maps with non-zero starting section numbers. Please use the most recent version, or contact me if you are not sure.

ccp4mapwish has been successfully built on IRIX 6.5, Dec Alpha OSF1 V4.0, and RedHat Linux. If the build fails then please let me know and I will try to help you - all feedback is extremely useful at this early stage.

Once you have *ccp4mapwish* then all you need to start MAPSLICER is to invoke the *mapslicer* script (as described previously in ``Starting Up''). In the latest release this script is in the $CCP4I_TOP/bin directory - so it should be on your path already, provided that the CCP4I_TOP environment variable is set correctly in your *ccp4.setup* file. This in turn calls the Tcl code for the main MAPSLICER program, which is in the $CCP4I_TOP/mapslicer directory.

Technical Details

MAPSLICER is a TCL/Tk script which uses a set of new TCL commands implemented in the *ccp4mapwish* interpreter to read a CCP4 map into memory and then contour and display a specific section.

ccp4mapwish is written in a combination of C (which is required to use the TCL C library commands) and Fortran (which is required to access the CCP4 library, particularly the map reading routines). The new commands are described in the *ccp4mapwish* documentation, $CHTML/ccp4mapwish.html, along with a link to the document describing how to build the interpreter.

The build is complicated by the compilation requiring TCL, Tk, Fortran and X window libraries to be linked in - all of which may have different names and locations on different systems. Also, version 8.0 or better of the TCL and Tk libraries are required. However, a configure script is supplied which tries to take care of these issues for the user.

I would be happy to hear people's experiences of building and using the program, with the aim of improving future versions.

Future Developments

MAPSLICER is still in its infancy (practically embryonic in fact) and this is evidenced by the limited number of features. The most immediate aim is to fix up some of the existing features (for example printing, setting contour levels, improved drawing of axes) which are rather kronky.

Beyond that: the ability to view several sections overlayed ("slab" mode), colouring contours, read in coordinates to display atoms, peaks and vectors are all on the list. "Under the bonnet" I am also intending to rewrite the contouring routines in C, and generally to remove as much of the Fortran code as possible over time - the user probably won't see much difference, but it should hopefully make the core interpreter more portable and easier to build.

Acknowledgements

ccp4mapwish makes use of the Fortran 2-d contouring routine from NPO. Stylistically the appearance of the sections and some of the input features have also been inspired by NPO.

Liz Potterton's CCP4i libraries were used inside MAPSLICER.

Martyn Winn provided positive feedback and encouragement.

Peter Briggs, March 2001

CCLRC Daresbury Laboratory

Protein Crystallography Specialist Users Group Meeting

York University, Thursday 4 January 2001

Report by Pierre Rizkallah

Agenda changes by the SR Users Forum at the SRS Users Meeting in September 2000 meant that the usual Specialist Users sessions to cover individual scientific areas were not convened. However, users of the Protein Crystallography facilities on the SRS expressed an interest in holding an alternative Specialist Users Group meeting at a convenient time and place. An appropriate opportunity presented itself in the form of a satellite meeting prior to the CCP4 Study Weekend, normally held early in January. The CCP4 organisers were very kind to extend support and assistance for the SUG meeting, and it was held the day before the start of the Study weekend. The delegates arriving early at York appreciated the opportunity for discussing common topics of interest. There were some 35 delegates, representing many of the user groups from the UK.

A scientific presentation from Robert Esnouf, Oxford University, discussed the usage of anomalous diffraction from S atoms, in the structure determination of a salivary protein from ticks. Although there had been high-resolution data available, collected at SRS Station 9.6 to a resolution of 0.9 Å, the structure could not be solved with direct methods. Data were then collected on SRS Station 14.1 at a wavelength of 1.488 Å, where the f'' signal from S atoms was equal to almost 0.5 e•. Such a small signal was countered by a high redundancy data set recorded over 855°. The resolution was curtailed due to the geometry limitations, but I/sig(I) was around 30 in the outermost shell, 1.7 - 1.6 Å. Shake'n'Bake located 16 S atoms in the 8 disulphide bridges of the 2 protein molecules in the asymmetric unit, using the Bijvoet differences. Phase extension to higher resolution was successful with ACORN. Chain tracing with ARP/wARP located all the ordered residues. The final stages of the refinement used SHELX. Further map interpretation will be needed to model the mobile N-termini.

Michele Cianci described the usage of longer wavelength (2.0 Å) in the structure determination of a-crustacyanin, a lipocalin. The similar protein structures available in the PDB could not be used in Molecular Replacement. The softer X-rays were used to collect a native data set with 20-fold redundancy, a data set with Xe derivative, and another native data set under similar conditions to those of the Xe derivative. All these data sets were limited by diffraction geometry and detector dimensions to a resolution of 2.3 Å. Finally, a 1.2 Å data set was collected with high intensity short-wavelength X-rays (0.87 Å). The Xe anomalous data set was used in Shake'n'Bake to solve the Xe structure. The 4 sites found were used to determine the positions of the S atoms. The correct hand was determined by verification against the high redundancy data set. The phases were used in the high-resolution data set and extended to cover all the resolution range. The various density manipulation techniques were used to get a good quality map, which was traced automatically with WARPnTRACE. All the sequence could be traced and minimal intervention was needed. The final model was refined against data to 1.35 Å resolution, with an R-factor of 17.5% and an R-free of 22.2%. Comparison with the other lipocalins is now underway.

Colin Nave then reported on the status of the SRS. After the vacuum leak in Nov 2000, which forced an early shutdown, repairs have been completed successfully, and the RF system upgraded, with a good prognosis for a smooth restart in January 2001. There were longer-term plans for wiggler cryogenics replacement and a further upgrade of the RF system. An extra degree of flexibility has been introduced into the overall scheduling system by incorporating "contingency days" which will either be shutdown or scheduled beam depending on previous events. Colin also reminded those present of the possibility of requesting extra or alternative days by sending an e-mail message to pxsched@dl.ac.uk, after inspecting the current schedule at http://www.dl.ac.uk/SRS/PX/flex_sched.html. New ionising radiation regulation requires users to be instructed in the safe usage of each station, and a form to be filled logging this training. A member of staff would be available at the weekend to provide this training for groups starting on weekend days, and to assist with incidental problems. Planned developments of the PX facilities included robotics and automatic systems for sample changing, centring, data collection and processing. Remote experiment control by operators at their home institution will also be developed. All the main data analysis software was made available at the stations, so users can complete their experiment while still at DL. A new development is also underway to build a new multipole wiggler beamline, in Straight 10, over the next two years. This will be Beamline 10, for MAD applications.

The station managers then reported on their respective stations:

- SRS 9.6 was fully utilised during 2000, producing an estimated 1000 data sets. A new sample visualisation system was installed giving enhanced visibility of very small crystals. The system comprises a colour 3 chip CCD camera with polarizers and a ¼ phase shift filter that convert small refractive index differences, between crystal and solvent, into colours. A large format CCD detector, with novel lens optics, will be loaned for tests, from AXS Bruker in allocation period 37. The detector will be tested on Station 7.2.
- SRS 14.1 commissioning had progressed well, and it was producing high quality data sets routinely. It could operate at two wavelengths, 1.488Å (Ni edge) or 1.214Å (W edge). Optics focusing was due to be optimised after the restart in January 2001. Change between the two wavelengths was estimated to take 2 to 3 hours, but a small band pass around each could be accessible rapidly.
- SRS 14.2 was commissioned successfully and started producing data early in 2000, until the PX210 detector arrived. It needed extensive work to commission it for PX usage, which was ultimately unsuccessful. The station was then released with the MAR345 image plate detector, until the arrival of the latest ADSC Q4 (delivered in Jan 2001).
- SRS 9.5 operated routinely, particularly with its tuneable beam capabilities. A stream of publications was coming out, based on data collected at this facility. MAD experiments now can benefit from the recent upgrade of the fluorescence detector, which made edge detection much more easy and reliable. Other recent upgrades include an alpha for faster data processing and an EMBL rotation axis, providing faster alignment and improved crystal viewing.

The final part of the meeting was a general discussion session where participants addressed many topics:

1. Facility developments should continue apace, in a ‘run up’ to DIAMOND. Methodology should be made immediately transferable when DIAMOND comes on-line

2. Station improvements are continuing, and feedback was called for, to ensure optimal response to the users needs.
3. The Biology Programme Joint Steering Committee have approved funding for automation developments, specifically a sample changer.
4. The grant for Beamline 10 would be administered by the Research Councils, and the beamline was due for completion earl in 2003. The construction work was forecast to impact on Line 9, causing some station closures for short periods.
5. A stream of projects should be built up to capitalise on the new facilities being developed, especially in the wake of automation.
6. Flexible scheduling is under evaluation. Extensive usage was asked for, in order to investigate potential shortcomings.
7. The participants encouraged the development of a longer wavelength data collection facility, either on Station 7.2 or Station 9.5. A rapid CCD system would be necessary for the collection of the required high redundancy data.
8. Automation was envisaged as an important factor for crystal screening, remote monitoring of experiments, and, in due course, remote control of experiments by scientists at their laboratory.
9. A PX service was under consideration, to offer operator data collection for qualifying users.

The participants expressed satisfaction at the alternative arrangements made for the Specialist Users Group session. They also expressed their gratitude to the CCP4 organisers for accommodating this session at short notice, and would hope to make it a regular feature in the coming years.

Maximum-Likelihood Refinement of Atomic Models using Least-Squares Criterion

by

P. Afonine[$,*], V.Y. Lunin[#,*] & A. Urzhumtsev[*]

[$] *Centre Charles Hermite, LORIA, Villers-lès-Nancy, 54602 France*

[#] *IMPB, Russian Academy of Sciences, Pushchino, 142290, Moscow Region, Russia*

[*] *LCM3B, UPRESA 7036 CNRS, Université Henri Poincaré, Nancy 1, B.P. 239, Faculté des Sciences, Vandoeuvre-lès-Nancy, 54506 France*

e-mail: sacha@lcm3b.uhp-nancy.fr

---

1. Notation

F_{obs}(**s**) - observed structure factors magnitudes

F^*(**s**) - modified structure factors magnitudes

F_{mod}(**s**) - magnitudes of structure factors calculated from the model

w(**s**) - weights for least-squares terms

LS - least-squares criterion calculated with F_{obs}

*LS** - least-squares criterion calculated with F^*

ML - maximum likelihood criterion (logarithm of likelihood gain)

a, b - parameters of the join probability distribution of structure factors considered as a function of random atomic models

e - reflection multiplicity

$I_0(x)$, $I_1(x)$, $I_2(x)$ - modified Bessel functions of 0, 1 and 2 order of argument x

cosh(x), tanh(x) - hyperbolic cosine and tangent of argument x

2. Least-squares and maximum likelihood criteria

The basic goal of a crystallographic refinement is to find an atomic model such that it minimises the functional

$$R = R_X + R_O,$$

where the crystallographic criterion R_X describes the quality of fit of structure factor magnitudes, F_{mod}, calculated from the model, to the experimental data, F_{obs}, and the R_O embodies other terms such as stereochemical criteria, a phasing criterion etc. In order to analyse the dependence of the refinement results on the choice of the crystallographic criterion R_X, in the current work the term R_O was excluded from all calculations.

$$LS = \sum_{\mathbf{s}} w(\mathbf{s})\left(F_{mod}(\mathbf{s}) - F_{obs}(\mathbf{s})\right)^2.$$

In practice, the basic statistical hypotheses for this criterion break when the atomic model is incomplete and the errors in experimental data F_{obs} are not independent, and the *LS* criterion becomes inadequate to the situation.

Recently, the maximum likelihood criterion started to be used (Bricogne & Irwin, 1996; Pannu & Read, 1996; Murshudov *et al.*, 1997) as R_X. The maximisation of likelihood is equivalent to minimisation of negative logarithm likelihood gain, which may be calculated as (Lunin & Skovoroda, 1995)

$$ML = \begin{cases} \sum_{\mathbf{s}} \left(\dfrac{\alpha^2 F_{mod}^2}{\varepsilon\beta} - \ln\left(\mathrm{I}_0\left(\dfrac{2\alpha F_{obs} F_{mod}}{\varepsilon\beta} \right) \right) \right), & \text{for acentric reflections} \\ \sum_{\mathbf{s}} \left(\dfrac{\alpha^2 F_{mod}^2}{2\varepsilon\beta} - \ln\left(\cosh\left(\dfrac{\alpha F_{obs} F_{mod}}{\varepsilon\beta} \right) \right) \right), & \text{for centric reflections.} \end{cases}$$

One of its major advantages is that it takes into account the contribution of atoms missed in an available atomic model (Lunin & Urzhumtsev, 1999).

However, an implementation of the *ML* criterion in existing programs needs their essential modification. An alternative solution would be to approximate this criterion near its minimum by a functional quadratic with respect to structure factor magnitudes, calculated from the atomic model (Lunin & Urzhumtsev, 1999). In this case, such approximation can be written again in the form of the usual *LS* criterion:

$$LS^* = \sum_{\mathbf{s}} w^*(\mathbf{s})\left(F_{mod}(\mathbf{s}) - F^*(\mathbf{s})\right)^2.$$

The values F^* can be considered as modified magnitudes F_{obs} and can be obtained as the solution of the following equation with respect to F^*:

$$F^* = \begin{cases} \dfrac{F_{obs}}{\alpha} \dfrac{\mathrm{I}_1(2hF_{obs})}{\mathrm{I}_0(2hF_{obs})}, & \text{for acentric reflections} \\ \dfrac{F_{obs}}{\alpha} \tanh\left(hF_{obs}\right), & \text{for centric reflections} \end{cases}$$

with a and b estimated as in (Lunin & Skovoroda, 1995) and

$$h = \frac{\alpha F^*}{\varepsilon\beta}$$

The weights *w** are calculated as

$$w^* = \begin{cases} \dfrac{2\alpha^2}{\varepsilon\beta} - 4h^2\left(\dfrac{I_2(2hF_{obs})}{I_0(2hF_{obs})} - \left(\dfrac{I_1(2hF_{obs})}{I_0(2hF_{obs})}\right)^2\right), & \text{for acentric reflections} \\ \dfrac{2\alpha^2}{\varepsilon\beta} - \left(\dfrac{h}{\cosh(hF_{obs})}\right)^2, & \text{for centric reflections.} \end{cases}$$

The tests below show the comparison of the refinement with different criteria: *LS*, *ML* and *LS**. Complete tests results and their analysis will be published elsewhere.

3. Numerical tests on comparative analysis of different criteria

The refinement tests were carried out with CNS complex (Brünger *et al*., 1998) using the structure of Fab fragment of monoclonal antibody (Fokine *et al.*, 2000). This model includes 439 amino acid residues and 213 water molecules. The molecule crystallises in the space group $P2_12_12_1$ with the unit cell parameters a = 72.24 Å, b = 72.01 Å, c = 86.99 Å, one molecule per asymmetric unit.

For test purposes the experimental data F_{obs} at 2.2 Å resolution were simulated by the corresponding values calculated from the complete exact model (Fig. 1). In all tests described below the starting atomic parameters were exact. Due to the absence of some atoms, removed randomly, the minimisation of the crystallographic criterion shifted the atomic parameters from their exact values showing that the minimum of all these criteria does not correspond to the correct model any longer. Smaller resulting errors indicate better quality of the criterion.

3.1. Test1: Random deletion of atoms in the crystal

In this test the atoms were removed randomly, the percentage of removed atoms varied from 0 to 20%. For each incomplete models the minimisation procedure was carried out using three different crystallographic criteria: *LS*, *ML* and *LS** (we remind that all stereochemical criteria were excluded from this refinement).

Figure 2a shows, for every criterion, the mean error in atomic positions for the models after refinement as a function of the size of a deletion. The errors grow with the percent of a deleted structure. The errors obtained with the *LS** minimisation are systematically less then those for the *LS* minimisation and are almost equal to the errors obtained with the *ML* minimisation. It can be noted that the weights *w** are crucial in order to obtain such results.

3.2. Test2: Random deletion of water molecules only

This test is similar to the previous one with the difference that only water molecules were allowed to be deleted from the model. The behaviour of errors (see Fig. 2b) is similar to that in the previous case. However, these errors are significantly larger and they grow faster with the percentage of the deleted structure (compare Fig. 2a and Fig. 2b). The reason for this may be the following: the water molecules are situated at the surface of the protein and not in its volume, and when the same amount of atoms is randomly excluded in both tests, in the case of water molecules they are distributed less uniformly in the space making stronger influence on the structure factors.

4. Conclusions

The tests discussed above show that the incompleteness of the model can seriously affect to the refinement. The more atoms are deleted, the larger are the errors in the model which fits best to the experimental data. Removal of water molecules has a stronger effect than a removal of a similar quantity of atoms randomly in the whole unit cell.

The tests show that the *ML* criterion is less sensible to the absence of a part of a model than the traditional *LS* criterion. In the case when an insertion of the *ML* criterion into an existing program is complicated, it can be replaced by its quadratic approximation. This approximation corresponds to the *LS* criterion calculated with F_{obs} substituted by F^* values and weighted by w^* (expression for both is given in the text). In all tests the least-squares minimisation against modified structure factors F^* gave the models of a significantly higher quality than those obtained by the minimisation against simulated F_{obs} and practically coinciding with the models obtained by maximum likelihood minimisation.

This shows that any crystallographic refinement program based on the minimisation of the least-squares criterion can give the results of the same superior quality as using maximum likelihood criterion without modifying the program itself when proper magnitudes and weights are used.

In this article we presented the results of first tests with an incomplete model without errors. An influence of other sources of imperfection of the model and data on refinement with various criteria will be discussed elsewhere.

The authors thank T. Skovoroda for her help with programming and C. Lecomte for his support of the project.

References

Bricogne, G. & Irwin, J. (1996). *Proceedings of the CCP4 Study Weekend*, 85-92.

Brünger, A.T., Adams, P.D., Clore, G.M., DeLabo, W.L., Gros, P., Grosse-Kunstleve, R.W., Jiang, J.-S., Kuszewski, J., Nilges, M., Pannu, N.S., Read, R.J., Rice, L.M., Simonson, T. & Warren, G.L. (1998) *Acta Cryst.,* D**54**, 905-921.

Fokine, A.V., Afonine, P.V., Mikhailova, I.Yu., Tsygannik, I.N., Mareeva, T.Yu., Nesmeyanov, V.A., Pangborn, W., Li, N., Duax, W., Siszak, E., Pletnev, V.Z. (2000). *Rus. J Bioorgan Chem*, **26**, 512-519.

Lunin, V.Y. & Urzhumtsev, A.G. (1999). *CCP4 Newsletter on Protein Crystallography*, **37**, 14-28.

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

Murshudov, G.N., Vagin, A.A. & Dodson, E.J. (1997). *Acta Cryst*., D**53**, 240-255.

Pannu, N.S. & Read, R.J. (1996). *Proceedings of the CCP4 Study Weekend*, 75-84.

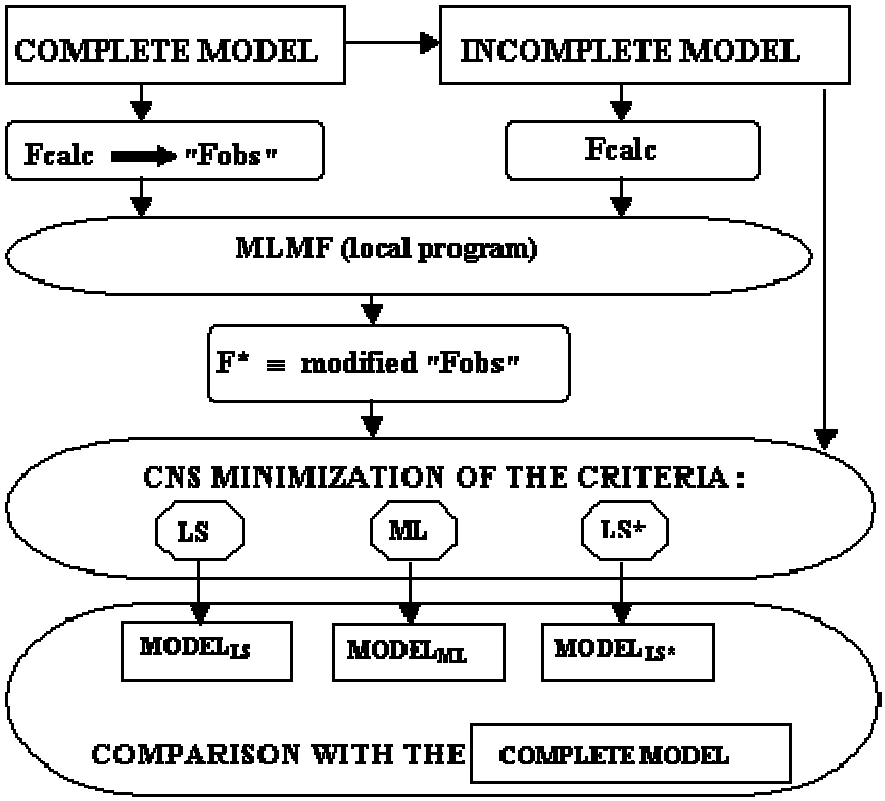


Figure 1. A scheme of the numerical tests.

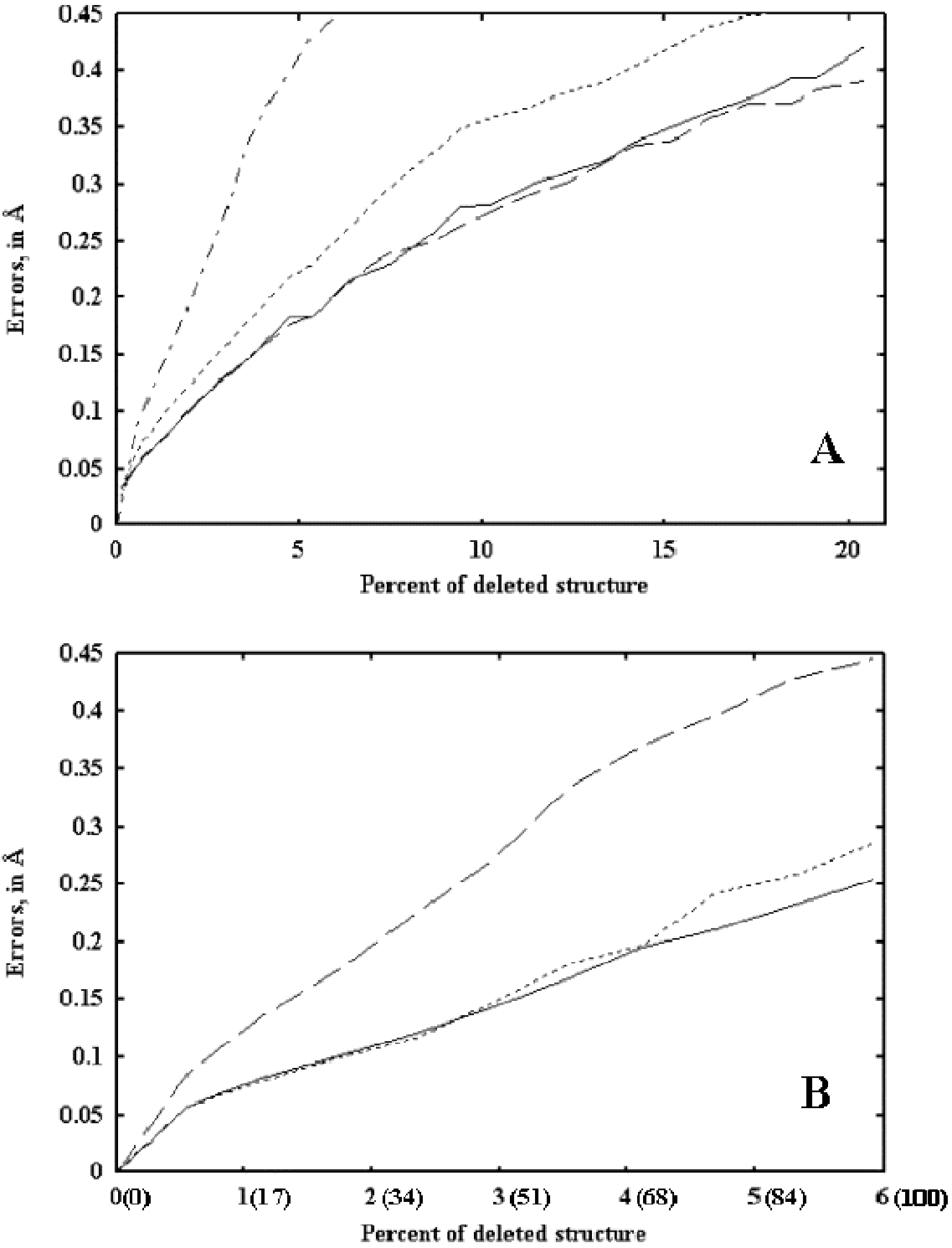


Figure 2. Dependence of mean coordinate error for the models after refinement as a function of the percent of deleted atoms. The dashed, solid and dotted lines show the behaviour of errors after minimisation using *LS*, *ML* and LS^{ac} criteria, respectively.

(A) Deletion of atoms of the whole model. The short line (dashed-and-dotted) is the same as the upper line (dashed) in picture (B) and is shown for comparison.

(B) Deletion of water molecules. Numbers in parentheses specify the percent of deleted water molecules.

CCP4i Chart Interface

Paul Emsley

Chart is a program that uses SHELXS and the CCP4 Program Suite to solve structure with minimum effort. It has recently been given a CCP4i interface. The traditional Chart interface is still part of the Chart distribution, but this new GUI interfaces (more) smoothly with CCP4i.

Chart is part of "Experimental Phasing" and can be found under "Run Mlphare" (Figure 1).

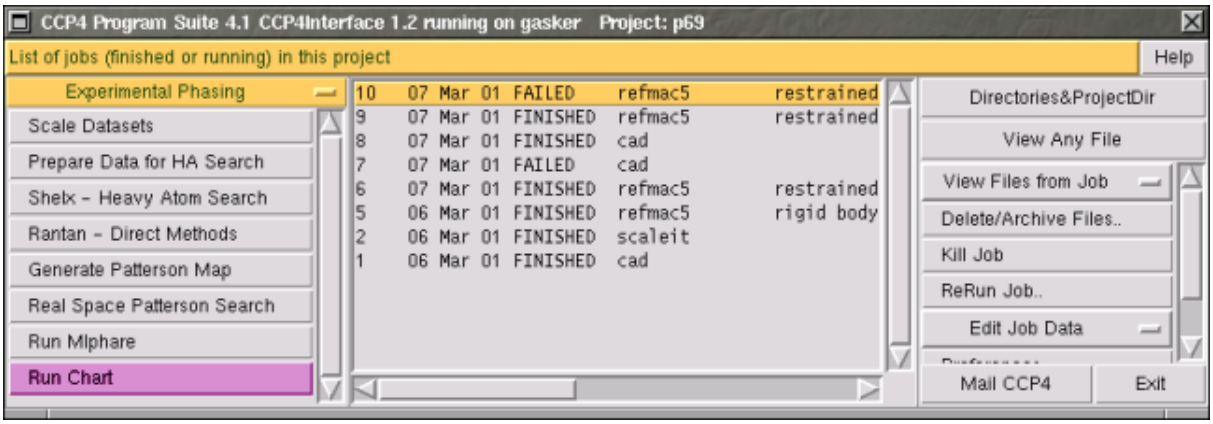


Figure 1. The position of Chart in the scheme of things.

Chart is suitable for MIR(AS), SIR(AS), MAD and SAD structure determination. Shown in Figure 2, is an example usage for MIRAS.

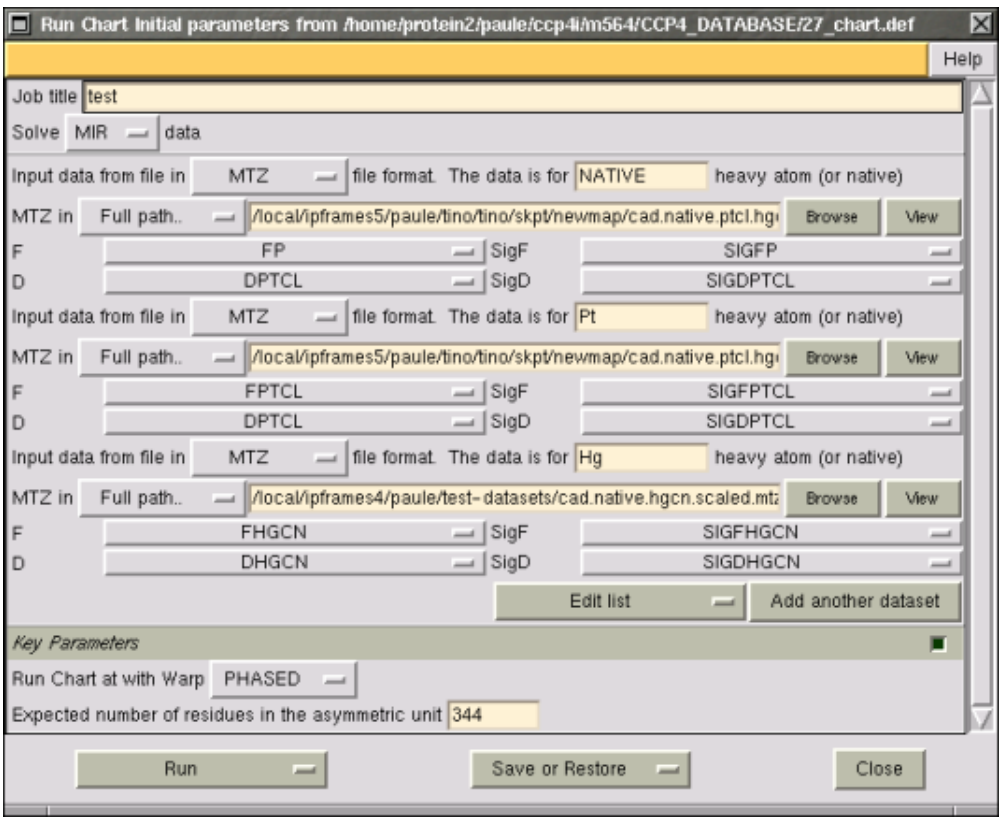


Figure 2: Typical MIR Chart usage.

The starting point for Chart is the data file created by TRUNCATE (truncated.mtz or some such), or you can use post-scalepack merged .sca files. (It is conceivable that Chart can be plugged into the back end of Auto-mosflm - but that is for the future).

Chart proceeds through the steps of structure determiation: scaling, heavy atom site determination, additional difference map sites, MLPhare phasing, hand inversion and density modification. Chart compares the output of both hands and decides if the structure has been solved and then creates plotted maps for both hands so that you can inspect them visually. Additional, during MAD structure determination, Chart will provide maps of the various heavy atom site sets.

Chart can then (optionally) proceed with ARP/wARP (although the parameter choice is limited compared to running ARP/wARP by other methods), this produces a PDB coordinate file.

This interface is new and under development (and it not available in the current stable release). However, we can see no reason why it will not be available in the next CCP4i release (and/or the next Chart release).

To learn more about Chart, visit http://www.chem.gla.ac.uk/~paule/chart.

08-Mar-2001

Efficient calculation of the exact matrix of the second derivatives

by Alexandre G. URZHUMTSEV[a*] and Vladimir Y. LUNIN[ab]

[a] *LCM3B, UPRESA 7036 CNRS, Faculte des Sciences, Universite Henri Poincare, Nancy I, 54506 Vandoeuvre-les-Nancy, France*

[b] *Institute of Mathematical Problems of Biology, Russian Academy of Sciences, Pushchino, Moscow Region, 142290, Russia*

Email:sacha@lcm3b.u-nancy.fr

1.Introduction

Recently, several groups (Murshudov *et al*., 1997; Templton, 1999; Tronrud, 1999) reported some advances in calculation of an approximate matrix of the second derivatives (Hessian or normal matrix) for crystallographic criteria which, in particular, is used in optimisation methods. In fact, the gradient methods need only the product of this matrix by a given vector which can be calculated rapidly and directly, without explicit matrix calculation (Lunin & Urzhumtsev, 1985). The second-order minimisation methods need an inverted Hessian matrix with an inversion being very sensible to approximations; as a consequence, algorithms for the calculation of the *exact* Hessian matrix [H] must be developed in this case. Moreover, it is important to know to calculate such matrix for combined minimisation criteria, with a mixture of different structure-factor depending criteria (crystallographic criterion in what follows), geometric criteria and others. This knowledge can influence the choice of the optimisation methods which vary in computation time per iteration, in the number of iterations necessary to arrive to a similar result, and in radius of convergence.

The direct calculation of the normal matrix of a crystallographic criterion needs of order of MN^2 operations (derivative of the contribution of every reflection with respect to every couple of parameters). Here N is the number of atomic parameters, M is the number of structure factors. In order to gain CPU time, some authors use a sparse matrix (Sussman *et al*., 1977; Hendrickson & Konnert, 1980; Dodson, 1981; Guillot *et al*., 2001). To estimate the full matrix of the second derivatives for the weighted least-squares fit of the magnitudes Templton (1999) suggested a statistical approach. Murshudov *et al*. (1997) and Tronrud (1999) developed the FFT technique suggested for this goal by Agarwal (1978). In particular, in the case of atomic parameters **q** as independent variables, Tronrud (1999) proposed a practical algorithm which needs

$$T_{HT} = C_1N^2 + C_2M \ln M \qquad (1)$$

operations to calculate the principal part of the matrix [H]

$$\sum_{\mathbf{h}} w(\mathbf{h}) \frac{\partial |F(\mathbf{q},\mathbf{h})|}{\partial q_j} \frac{\partial |F(\mathbf{q},\mathbf{h})|}{\partial q_k} \qquad (2)$$

traditionally neglecting the second term

$$\sum_{\mathbf{h}} w(\mathbf{h})[|F(\mathbf{q},\mathbf{h})| - F_o(\mathbf{h})] \frac{\partial^2 |F(\mathbf{q},\mathbf{h})|}{\partial q_j q_k} \qquad (3)$$

Similar formulae were derived by Murshudov *et al*. (1997) for the case of the maximum likelihood criterion. At the same time, the use of Fast Differentiation Algorithm (Baur & Strassen, 1983; Kim *et al*., 1984; for crystallographic applications see Lunin & Urzhumtsev, 1985) allows to calculate rapidly the *exact* matrix of the second derivatives with respect to atomic parameters for both these criteria and to generalise this result for other cases.

2.Fast Differentiation Algorithm and crystallographic refinement

Traditionally, it is considered that the gradient methods are much more time consuming than the methods without derivatives, and that generally every computation of a gradient needs about N times CPU more than a single value of the function, N being the number of variables. However, several groups (*e.g.,* Baur & Strassen, 1983; Kim *et al*., 1984) have shown the following result and its conclusions:

Fast Differentiation Algorithm (FDA)

For any function of N variables, calculation of a single value of which takes the time *T*, an algorithm exists to calculate its *exact* gradient faster than in 4*T independently on the number N* of variables.

FDA : Conclusion 1

For any function of N variables, calculation of a single value of which takes the time *T*, an algorithm exists to calculate its *exact* derivative along a given direction faster than in 4*T independently on the number N* of variables and *without* a gradient calculation.

FDA : Conclusion 2

For any function of N variables, calculation of a single value of which takes the time *T*, an algorithm exists to calculate the *exact* product of the matrix of the second derivatives by a given direction faster than in 20*T independently on the number N* of variables.

In practice, when some common expressions can be saved in memory, the time to calculate all 4 objects, namely the function *f* itself, its gradient ∇f, the function derivative along a given direction **s** and the product [H]**s** of the Hessian matrix [H] by the same direction, can be estimated rather by 4*T* where *T* stands for the time of a single calculation of the function value (Urzhumtsev *et al*., 1989).

FDA shows that the crucial point in the fast optimisation of crystallographic functional is a fast calculation of structure factors. A fast way of their calculations (Sayre, 1951; Ten Eyck, 1977) needs about

$$T_f = C_1 N + C_2 M \ln M \qquad (4)$$

operations (C_1 and C_2 are some constants), instead of CNM operations for a direct calculation.

FDA gives the same amount of operations for the gradient calculation being applied to a least-squares criterion depending on atomic parameters (Lunin, 1978, unpublished; Lifchitz, in Agarwal, 1981) or to other crystallographic criteria (Lunin & Urzhumtsev, 1985; Urzhumtsev *et al*, 1989).

Moreover, because the nth column of the matrix [H] of the second derivatives represents the product of this matrix by the direction (0, 0, … , 0, 1, 0, …, 0) where 1 is in the n-th position, the whole *exact* matrix of the second derivatives composed from N columns can be calculated by

$$T_{HE} = C_1 N^2 + C_2\, NM \ln M \qquad (5)$$

operations. However, a faster and more general way to calculate the exact Hessian matrix can be suggested.

Let a function $R(y_1(x_1, \ldots, x_N), \ldots, y_M(x_1, \ldots, x_N))$ depend on M variables $y_1, \ldots, y_M$ with every y_m depending in turn on N variables $x_1, \ldots, x_N$. The chain rule formula:

$$\frac{\partial^2 R}{\partial x_j x_k} = \sum_m \sum_n \frac{\partial^2 R}{\partial y_m \partial y_n} \frac{\partial y_n}{\partial x_k} \frac{\partial y_m}{\partial x_j} + \sum_m \frac{\partial R}{\partial y_m} \frac{\partial^2 y_m}{\partial x_j \partial x_k} \qquad (6)$$

can be rewritten as

$$\Delta_x R = [dy/dx]\,[\Delta_y R]\,[dy/dx]^T + [\Delta_x y]\,\nabla_y R \qquad (7)$$

Here $[\Delta_x y]$ is a tensor composed of M matrices $[\Delta_x y_1]$, $[\Delta_x y_2]$, ... $[\Delta_x y_M]$. If the gradient $[\nabla_y R]$ is supposed to be known (the calculations are carried out accordingly to the FDA) then the operations needed to get $[\Delta_x R]$ are the operations to calculate the matrix products in (7) plus those to calculate $[\Delta_x y]$

Formula (7) can be simplified for a number of special cases.

Special case 1: If the criterion R is additive, *i.e.* can be presented by a sum of individual contributions, not necessarily quadratic, from the components of the vectors **y**

$$R(\mathbf{y}) = \sum_m f(y_m) \qquad (8)$$

then the matrix $[\Delta_y R]$ becomes diagonal. An example is the least-squares fit of the calculated data to the experimental ones.

Special case 2: When the variables **y** depend linearly on the variables **x** , **y** = [A] **x** , the second term in the formula is absent and (7) becomes

$$\Delta_x R = [A][\Delta_y R][A]^T \qquad (9)$$

An example of such linear dependence is the relation between the electron density and its structure factors.

Special case 3: Quite often variables **x** contribute to variables **y** locally, i.e. each of x_1, x_2, ... x_N contribute only to a small number C_y of variables y_k, $C_y << M$. Complementary, every y_k may depend only on a small number of C_x parameters x_j, $C_x << N$. The matrix [d**y** / d**x**] becomes sparse giving an essential reduction in the number of calculations. An example of such dependence is the calculation of any field (e.g., an electron density) from an atomic model where atoms have a limited radius of contribution and are separated each from others.

It is easy to see that the calculation of normal matrix for crystallographic criteria can profit the features of all particular cases discussed above and it can be shown that the total amount of operations necessary to calculate the exact matrix of the second derivatives of the crystallographic least-squares criterion with respect to the atomic parameters is

$$T_{HC} = C_1M + C_2M \ln M + C_3N^2 \sim C_{12} M \ln M + C_3N^2 \qquad (10)$$

where C_1, C_2, C_3 and C_{12} are some constants which do not depend neither on the number of structure factors M nor on the number of atoms N. This estimate is the same as (1) for an *approximate* matrix where the second-order terms $\partial^2 R / \partial F^2$ are neglected from the beginning and is better than (5) which is obtained for the *exact* matrix by simple N-times calculation of the product of a matrix by a direction following a coordinate axis.

The same amount of operations is enough to calculate the matrix for the intensity least-squares criterion (Sheldrick & Schneider, 1997), for the phase criterion (Lunin & Urzhumtsev, 1985; Pannu *et al*., 1998), for the maximum likelihood criterion (Bricogne & Irwin, 1996; Pannu & Read, 1996; Read, 1997; Murshudov *et al*., 1997; Pannu *et al*.,1998).

In the general situation when the criterion cannot be represented by a sum of contributions from many small subsets of structure factors, the total number of computer operations becomes :

$$T_{HG} = C_6M^2 \ln M + C_3N^2 \qquad (11)$$

If the independent parameters are not atomic ones (*e.g.,* parameters for a rigid groups refinement), the matrix with respect to independent parameters can be calculated in the same way.

3.Direct calculation of the inverted Hessian matrix

The formula (7) gives also an idea that for some special cases the inverted matrix of the second derivatives can be obtained directly without calculation the Hessian matrix itself. Indeed, if the transformation **y**(**x**) is linear, **y** = [A] **x**, then

$$[\Delta_x R]^{-1} = [A^{-1}]^T [\Delta_y R]^{-1} [A^{-1}] \qquad (12)$$

If [A] corresponds to the Fourier transformation, the inverse operation is the inverse Fourier transform for which the matrix $[A^{-1}]$ can be written immediately. The matrix $[\Delta_x R]^{-1}$ can be easily calculated for many crystallographic criteria, in particular for such important criteria as the least-squares or maximum likelihood functionals. Therefore, when the independent parameters are density values at the grid points the inverse Hessian matrix can be easily and directly calculated for these criteria as

$$[\Delta_p R] = \begin{pmatrix} u(\mathbf{r}_1+\mathbf{r}_1) & u(\mathbf{r}_2+\mathbf{r}_1) & \dots & u(\mathbf{r}_K+\mathbf{r}_1) \\ u(\mathbf{r}_1+\mathbf{r}_2) & u(\mathbf{r}_2+\mathbf{r}_2) & \dots & u(\mathbf{r}_K+\mathbf{r}_2) \\ \dots & \dots & \dots & \dots \\ u(\mathbf{r}_1+\mathbf{r}_K) & u(\mathbf{r}_2+\mathbf{r}_K) & \dots & u(\mathbf{r}_K+\mathbf{r}_K) \end{pmatrix} \qquad (13)$$

where all elements of the matrix (13) are presented by the same function calculated as a Fourier series at different points **r** . In order to calculate this function at a grid compatible with the number of Fourier coefficients M, estimating K ~ M , the number of operations needed is about C_2M lnM (Cooley & Tukey, 1965; Ten Eyck, 1973). This results shows that the minimisation methods of simple iteration, usually applied for density modification procedures, can be replaced not only by the gradient methods (Sayre, 1972, and Sayre & Toupin, 1975, for the particular Sayre criterion; Lunin, 1985, for the general case) but even by the methods of the second order. In this case the computational expenses are practically the same as those for the simple iteration methods.

4.Conclusion

In the case of the crystallographic least-squares or maximum likelihood refinement of atomic model, the *exact* matrix of second derivatives can be calculated by

$$T_{HC} = C_{12}\, M\, \ln M + C_3 N^2 \qquad (14)$$

Operations where M is the number of structure factors, N is the number of atomic parameters and C_{12} and C_3 are some constants which do not depend neither on M nor on N. This algorithm suggests step-by-step recalculation of the matrix with respect to variables of different levels of the molecular models (structure factors, density, atomic parameters etc.). The same iterative calculations are basic steps for the fast gradient calculation (Lunin & Urzhumtsev, 1985). It should be noted that such way of calculation allows easily to add the contribution from any other criteria of the same type or of any other type of models, *e.g.,* phase criterion, stereochemical criteria, criteria depending on the electron density etc. Therefore, the formulae which give the expression of the gradient (or of the Hessian matrix) of a crystallographic criterion directly in terms of atomic parameters can be useful for understanding but may be rather misleading algorithmically.

The full material with details of corresponding derivations will be reported elsewhere (paper in preparation for Acta Crystallographica).

The work was supported by RFBR grant 00-04-48175 (VYL). The authors thank A.G. Kushnirenko and K.V. Kim who attracted their attention to the fast differentiation algorithm and C. Lecomte and Centre Charles Hermite (Nancy) for their support of the work

References

Agarwal, R.C. (1978) *Acta Cryst*., **A34**, 791-809.

Agarwal, R.C. (1981) In *Refinement of Protein Stryctures*: Proceeding of the Daresbury Study Weekend 15-16 November, 1980; compiled by P.A.Machin, J.W.Campbell and M.Elder, pp. 24-28. Warrington : Science and Engineering Research Council, Daresbury Laboratory.

Baur, W. & Strassen, V. (1983). *Theoretical Computer Science*, **22**, 317-330.

Bricogne, G., Irwin, J. (1996) In *Macromolecular Refinement*: Proceedingof the CCP4 Study Weekend, E.Dodson, M.Moore, A.Ralph & S.Bailey, eds., pp.85-92. Warrington : Daresbury Laboratory.

Brünger, A.T. (1992) *Nature*, **355**, 472-474.

Cooley, J.W. & Tukey, J.W. (1965) Math.Comput., **19**, 297-301

Cowtan, K. & Ten Eyck, L.F. (1998) *ECM-18 Abstracts, XVIIIth European Cryst.Meeting, 16-20 August 1998, Prague, Republic Czeck*, *E5-P7, Bulletin of the Czeck and Slovak Crystallographic Association,* **5A***,* 136

Dodson, E. (1981) In *Macromolecular Refinement*: Proceeding of the CCP4 Study Weekend, E.Dodson, M.Moore, A.Ralph & S.Bailey, eds., pp. 85-92. Warrington : Daresbury Laboratory.

Guillot, B., Viry, L., Guillot, R., Lecomte, C., Jelsch, C. (2001) *J.Appl. Cryst*., **34**, in press.

Hansen, N.K. & Coppens, P. (1978) *Acta Cryst*., **A34,** 909-921.

Hendrickson, W.A. & Konnert, J.H. (1980) In *Biomolecular Structure, Function, Conformation and Evolution*, edited by R.S.Srinivasan, Vol. 1, 43-57. Oxford : Pergamon.

Hestenes, M.R. & Stiefel, E. (1952) *J.Res.Nat.Bur.Standards*, **49**, 409-436.

Kalinin, D.I. (1981) *Kristallographiya*, **25**, 535-544.

Kim, K.M., Nesterov, Yu.E. & Cherkassky, B.V. (1984) *Dokl.Acad.Nauk SSSR*, **275**, 1306-1309.

Lanczos, C. (1952) *J.Res.Nat.Bur.Standards*, **49**, 33-53.

Lunin, V.Yu. (1985). *Acta Cryst*. A**41**, 551-556.

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

Lunin, V.Yu. & Urzhumtsev, A.G. (1985) *Acta Cryst.*, **A41**, 327-333

Lunin, V.Yu. & Urzhumtsev, A.G. (1999) *CCP4 Newsletter on Protein Crystallography,* **37**, 14-28

Murshudov, G.N., Vagin, A.A. & Dodson, E.J. (1997) *Acta Cryst*. D**53**, 240-255.

Pannu, N.S., Murshudov, G.N., Dodson, E.J. & Read, R.J. (1998) *Acta Cryst*. D**54**, 1285-1294

Pannu, N.S. & Read, R.J. (1996) *Acta Cryst*. A**52**, 659-668.

Read, R.J. (1997) In *Methods in Enzymology*, Academic Press, San Diego., C.W.Carter, Jr., R.M.Sweet, eds., **277B**, 110-128.

Sheldrick, G.M. & Schneider, T.R. (1997). In *Methods in Enzymology*, Academic Press, San Diego., C.W.Carter, Jr., R.M.Sweet, eds., **277B**, 319-343.

Sayre, D. (1951) *Acta Cryst*., **4**, 362-367

Sayre, D. (1972) *Acta Cryst*., A**28**, 210-212

Sayre, D. & Toupin, R.A. (1975) *Acta Cryst*., A**31**, S20

Sussman, J.L., Holbrook, S.R., Church, G.M. & Kim, S.-H. (1977) *Acta Cryst*., **A33**, 800-804.

Templton, D. (1999) *Acta Cryst*., **A55**, 695-699.

Ten Eyck, L.F. (1973) *Acta Cryst*., **A29**, 183-191.

Ten Eyck, L.F. (1977) *Acta Cryst*., **A33**, 486-492.

Ten Eyck, L.F. (1999) *IU Collected Abstracts, XVIIIth IUCr Congress & General Assembly, 4-13 August 1999, Glasgow, Scotland*, 97.

Tronrud, D.L. (1992) *Acta Cryst*., **A48**, 912-916.

Tronrud, D.L. (1999) *Acta Cryst*., **A55**, 700-703.

Urzhumtsev, A.G., Lunin, V.Yu. & Vernoslova, E.A. (1989) *J.Appl.Cryst.*, **22**, 500-506

Reports on the Daresbury Protein Crystallography Data Collection Workshop

We are fortunate to have two reports from the recent Protein Crystrallography Data Collection Workshop held at Daresbury Laboratory - one is an organisers' eye-view of the event by Liz Duke, the other the picture from the ground by workshop student Jasveen Chugh.

Report on the Protein Crystallography Data Collection Workshop

Liz Duke

CLRC Daresbury Laboratory, Warrington, UK

A Protein Crystallography Data Collection Workshop, funded by CCP4 and the EU was held at Daresbury Laboratory in February 2001.

The workshop started with an excellent overview of diffraction given by Andy Freer. Andy compared a lab source and the synchrotron and described what happens when a protein crystal is placed in each type of beam. Jim Clarke from Daresbury Laboratory followed up this talk with a superb overview of synchrotron radiation. Such was the standard of the talk that even those members of the audience based at a synchrotron wanted copies of his overheads! Colin Nave spoke about the protein crystallography beamlines at Daresbury and outlined the developments planned for the next few years – enough to keep all the Daresbury staff busy for a good while to come! Anna Lawless then outlined the facilities available for users in the Structural Biology Laboratory at Daresbury. Gordon Leonard from the ESRF finished of the more theoretical aspects of data collection with an overview of the facilities at the ESRF.

Having got what could be perceived as the more boring stuff out of the way; David Owen started the next session off with an excellent talk on Molecular Biology. Lisa Wright followed this up by describing the next step of protein crystallization. Elspeth Garman, as ever, gave a superb talk on how to freeze crystals. I wish it were always as easy as she makes it look! The first evening of the course was taken up with each of the participants giving a 3-minute presentation of their work using a maximum of 2 overheads. Wine and beer were provided to ensure that the evening passed in a relaxed and informal manner.

The second day started with 2 talks on data collection – the first by Dave Lawson on data collection practicalities and the second by Gordon Leonard on MAD data collection. Elspeth then gave a second talk on high-resolution data collection – some people only see such resolution in their dreams – but at least we now know what to do should it ever happen to us!

Having heard about how to collect data, the next step on the road to success was to learn what to do next. Alun Ashton and Pete Briggs did a double act outlining CCP4 and then Harry Powell gave an excellent talk on Mosflm and its capabilities. Paul Taylor then

followed with a talk on his experiences using denzo. Gwyndaf Evans, Steve Prince and Lisa Wright then aired Scala, MIR and MR respectively. I will forever think of molecular replacement as placing the original Mickey Mouse on top of Millennium Mickey. Claire Naylor then added a touch of variety with a talk on how the use of Circular Dichroism can complement information obtained via protein crystallography.

As no data collection trip could be complete without spending hours waiting for the beam, we were able to plan a day waiting for the beam. Fortunately the SRS did deliver the goods in the late afternoon and some of the participants were able to collect data on the stations. Others learnt about the frustrations of having crystals that don't diffract!

The final day and a half were spent at the computer putting into practice what had been learnt in the talks. The participants had the opportunity to try out mosflm, denzo and scala and look at examples of MIR and MR.

At the end of the workshop I took the bull by the horns and asked the participants to fill in a feedback form telling me the best and worst bits of the workshop. As expected, 8 out of 10 cats did prefer whiskers and no one liked the Daresbury Canteen. On a more serious note everyone was very positive about the usefulness of the workshop with most people feeling that the best part was the opportunity to collect data on the stations.

Finally my thanks go to my colleagues here at Daresbury – principally Steve Kinder and Dave Love on computers, James Nicholson on setting up the web pages and sending out emails and Pat Broadhurst who did all of the administration without appearing to bat an eyelid when I changed my mind numerous times in an afternoon. Also I owe a huge thank you to all the speakers who put so much effort into the workshop and made it what it was. Thank you very much, I owe you one.

My First Daresbury Experience

Jasveen Chugh

Birkbeck College, University of London

It was at the beginning of the second year of my PhD when I decided to try some Crystallography. After some time I obtained crystals of a small molecule polypeptide, which diffracted to 1.6Å resolution in-house at Birkbeck College. Looking to collect higher resolution data, acquire knowledge about the field along with practical skills with respect to structure solution I was extremely delighted to be accepted on the PX Course in Daresbury (Feb 2001).

With the possibility of beamtime at the course I arrived with crystals of my peptide on the night of Sunday 4th Feb noticing first the extremely cold weather, I left my crystals in cold storage at Daresbury Labs. Accomodation for the delegates was 25 minutes away from the labs at Padgate Campus. On entering my room my first impressions were of a warm and comfortable room made even nicer with an ensuite!

On Monday morning a coach took I and 19 other students to the tower at Daresbury Labs where the course was being held. On our arrival we were greeted by course organisers Dr. Liz Duke and Dr. James Nicholson. It was exciting to meet new people and very soon I

made new friends, many were from within the UK and some had travelled to the UK for the course. It was nice to meet people from such a wide array of backgrounds.

The first day of talks described the theory behind data collection and the set ups of various synchrotrons along with the designs of PX beamlines. After lunch at the Daresbury Laboratory restaurant where we dined throughout the week, we returned to lectures which described in detail data collection practice, protein crystallisation and cryo-cooling techniques. By dinnertime having made new friends I was starting to enjoy my first day and feel comfortable. After dinner there was an informal session where we all gave a 5-minute presentation about our research work, it was interesting to see how different everyone's work was, and this helped to acquaint us better with one another.

On the second day of the course we heard about different data collection methods such as MAD and high-resolution data collection. Along with the major features of CCP4 supported programs and in particular the advantages of using the CCP4 Graphical User Interface, the use of which as a beginner in the field I found particularly appealing. Later after lunch the rest of the talks consisted of specific methods and programs for data processing and structure solution. The packages MOSFLM and DENZO/SCALEPACK were discussed in great detail and as these are the two common packages used for structure solution these were extremely useful lectures.

After two long days of lectures of all the theory I along with everyone else was excited to have beamtime the next day to be able to try some practical work.

When Wednesday finally arrived we were put into groups and off we went to the stations. We had just about got accustomed with the layout of the station and got to the point of trying data collection when there was no beam! The morning then went by with a lot of coffee, waiting and hoping that there would be some beam. Thus described as the typical Daresbury experience finally at around 3pm I was lucky that there was beam and that I went first to try my crystals at diffraction. With the kind help of Dr. Simon Teat (Station 9.8 manager) and Dr. Pierre Rizkallah (Station 9.6 scientist), who helped put my crystal on the beamline (9.6) and start data collection, I was overjoyed to say the least when my first diffraction pattern showed the crystal had diffracted to 0.9Å resolution!

On Thursday we had practical sessions on how to use different data processing packages followed by a practical on structure solution by molecular replacement. In the evening after practicals we returned to Padgate College where we attended a specially organised dinner in the Buckley Suite. It was nice to spend the last evening with the new friends that I had made in the past few days. Friday, all of us eager to return home we attended closing talks, had lunch and caught trains to come home. Little did some of us know the delays that were ahead!

By the end of an intensive week I was exhausted but extremely pleased at the opportunity I had gained to learn about data collection, processing and structure solution methods. I would definitely recommend this course to those who have just started out in the field, know a little but would like to know more! I would very much like to thank Dr. Liz Duke and Dr. James Nicholson for their hospitality and guidance and mostly for the opportunity to attend such a well organised course.

Improvement of noisy maps by bulk solvent correction

By A. Fokine & A. Urzhumtsev

Laboratory of Crystallography and Modelling of Mineral and Biological Materials, UPRESA 7036 CNRS, University Henri Poincaré, Nancy I, 54506 Vandoeuvre-les-Nancy, France

e-mail : sacha@lcm3b.u-nancy.fr

Introduction

The main goal of the structural crystallography is to build a model of the crystal under study. For macromolecules, to do so an electron density map calculated at a finite resolution, as high as possible, is necessary. A distribution of the electron density in a crystal can be represented by a Fourier series with the coefficients $\mathbf{F}_{obs}$ the main contribution to which comes from ordered atoms, $\mathbf{F}_{atom}$, and from bulk solvent, $\mathbf{F}_{solv}$:

$$\mathbf{F}_{obs} = \mathbf{F}_{atom} + \mathbf{F}_{solv} . \qquad (1)$$

The complex values $\mathbf{F}_{obs}$ are not available from a single X-ray diffraction experiment and usually they are approximated by

$$\mathbf{F}_{obs} \sim |\mathbf{F}_{obs}|\exp(i\varphi_{est}) \qquad (2)$$

where φ_{est} are some estimations for structure-factor phases.

However, since the main goal of the map interpretation is to build an atomic model, it seems that the map calculated with the coefficients $\mathbf{F}_{atom}$ should be more suitable for this purpose (Urzhumtsev, 2000) than the traditional map calculated with the coefficients (2). If we assume that the bulk solvent contribution to the diffraction pattern can be estimated before an atomic model is known, these structure factors can be approximated as

$$\mathbf{F}_{atom} \sim (|\mathbf{F}_{obs}|\exp(i\varphi_{est}) - \mathbf{F}_{solv}). \qquad (3)$$

It should be noted that the substitution $(|\mathbf{F}_{obs}|\exp(i\varphi_{est}) - \mathbf{F}_{solv})$ for $|\mathbf{F}_{obs}|\exp(i\varphi_{est})$ changes not only the phases of the Fourier coefficients but also their moduli, differently from standard phase improvement methods (taking apart weighting of moduli by figure of merit of corresponding phase).

The numerical tests discussed below show that such map correction by subtraction of the bulk solvent contribution can significantly improve the quality of density maps. For such study, we simulated the problem arising in the single isomorphous replacement (SIR) method and analysed the quality of SIR maps, calculated before and after bulk solvent contribution, as a function of resolution and error in the position of the heavy atom.

Tests description

Atomic model

The crystal structure of Fab fragment of monoclonal antibody LNKB-2 solved at 2.2 Å resolution (Fokin *et al*., 2000) was chosen as a test model. Crystals of Fab-LNKB-2 belong to the space group $P2_12_12_1$ (a = 72.24 Å, b = 72.01 Å, c = 86.99 Å) and contain one molecule per asymmetric unit. Solvent region occupies 42% of the unit cell volume. Fab fragment molecule consists of two polypeptide chains (with 219 and 220 amino acid residues, respectively). 213 water molecules are included in the atomic model.

Solvent structure

In order to simulate the contribution of the bulk solvent we used the approach by Jiang & Brünger (1994). A binary solvent mask (1 in solvent region 0 in protein region) was calculated by program CNS (Brunger *et al*., 1998) with the parameters SOLRAD and SHRINK equal to 1.0 Å. The corresponding structure factors $\mathbf{F_{mask}}$ were exponentially scaled

$$\mathbf{F_{solv}} = k_{sol} \exp(-B_{sol}\sin^2(\theta)/\lambda^2)\, \mathbf{F_{mask}} \qquad (4)$$

with scaling parameters k_{sol} and B_{sol} chosen as 0.40 $e^-/Å^3$ and 70 $Å^2$, respectively.

The structure factors $\mathbf{F_{solv}}$ of the bulk solvent were then added to structure factors $\mathbf{F_{atom}}$ calculated from the atomic model giving the total structure factors of the crystal

$$\mathbf{F_{tot}} = \mathbf{F_{atom}} + \mathbf{F_{solv}}. \qquad (5)$$

Each set of structure factors included 23673 reflections and corresponded to the full data set up to 2.2 Å resolution.

Numerical SIR simulation

A SIR-like situation was simulated as it was described previously in (Urzhumtsev, 1991). An artificial gaussian heavy atom was placed in the cavity between two variable and two constant domains of Fab molecule. For this artificial heavy-atom derivative structure factors $\mathbf{F_{der}}$ were calculated as

$$\mathbf{F_{der}} = \mathbf{F_{tot}} + \mathbf{F_H} \qquad (6)$$

where $\mathbf{F_H}$ are structure factors for the heavy atom. The number of electrons in the heavy atom was chosen such that its contribution to the structure factors was

$$R_H = \Sigma|\mathbf{F_H}| / \Sigma|\mathbf{F_{tot}}| = 0.1 \, . \qquad (7)$$

In the tests, the moduli of structure factors $\mathbf{F_{tot}}$ and $\mathbf{F_{der}}$ were supposed to be known and simulated the experimental magnitudes for the native crystal and its derivative. The position of heavy atom was supposed to be determined either absolutely accurately or with the coordinate error of 1.0 or 2.0 Å, depending on the test. The $|\mathbf{F_{tot}}|$ and $|\mathbf{F_{der}}|$ values and heavy atom parameters were used to calculate phases φ_{sir} and their figures of merit m_{sir} with the standard SIR technique (Blow & Rossman, 1961).

Electron density maps

SIR-phased Fourier syntheses

$$\rho_{sir}(\mathbf{r}) = V^{-1}\{\Sigma_{\mathbf{h}}\, m_{sir}|\mathbf{F_{tot}}(\mathbf{h})|\exp(i\varphi_{sir}(\mathbf{h}))\exp(-2\pi i\mathbf{hr})\} \quad (8)$$

and the syntheses with subtracted bulk solvent contribution

$$\rho_{corr}(r) = V^{-1}\{\Sigma_{\mathbf{h}}\,[\, m_{sir}|\mathbf{F_{tot}(h)}|\exp(i\varphi_{sir}(\mathbf{h})) - \mathbf{F_{solv}}\,]\exp(-2\pi i\mathbf{hr})\} \quad (9)$$

were calculated using reflections with the resolution $d > d_{min}$, where d_{min} was equal to 2.2, 3.0, and 6.0 Å depending on the test (here V is the unit cell volume). For comparison we calculated also the maps with the coefficients $\mathbf{F_{atom}}$

$$\rho_{atom}(r) = V^{-1}[\Sigma_{\mathbf{h}}\,\mathbf{F_{atom}}(\mathbf{h})\exp(-2\pi i\mathbf{hr})]. \quad (10)$$

All syntheses were normalised with respect to their mean value and standard deviation σ.

Numerical comparison of electron density maps

When Fourier syntheses ρ(r) are analysed, the objects of interest are the shape and localisation of equipotential surfaces rather than the absolute values of the density. The lower is the cut-off level ρ*, the larger is the region bounded by the surface ρ(r) = ρ*, the more atoms of the model are inside (trapped by) this region. On another hand, when comparing two synthesis taken at an equivalent level, the better one traps more atoms than the worse one. Therefore, the percentage of the missed atoms D(r) as a function of cut-off level can be used to estimate the quality of the syntheses as it is realised in the program MTRAP (Lunina & Lunin, personal communication). While originally in MTRAP the cut-off level is estimated through the relative volume of the selected region, for our tests we used more traditional measure in sigmas.

Results and discussion

Figure 1 illustrates the quality of the maps calculated at 6.0, 3.0, and 2.2 Å resolution with exactly assigned position of the heavy atom. The syntheses with subtracted bulk solvent contribution are significantly better than standard SIR syntheses (compare solid and dotted lines, Fig. 1). The effect of the map improvement due to subtraction of the bulk solvent contribution depends on the resolution and on the level of the equipotential surfaces. For the map calculated at 6.0 Å resolution (Fig.1a) the improvement is significant for levels in the interval 0-1 σ, and for maps calculated at 3.0 Å and 2.2 Å resolution (Figs. 1b, 1c) the improvement is significant for levels in the interval 0-1.5 σ. At higher levels we will see only a small number of strong peaks corresponding to clearly observed atoms. The improvement is maximal for the SIR synthesis calculated at 6.0 Å resolution (compare solid and dotted lines, Fig. 1a). For example, at 0.5 σ the percentage of missed atoms for SIR synthesis is 50 but for the synthesis with subtracted bulk solvent contribution it is less than 20. This is not surprising because the bulk solvent contribution is very significant at low resolution.

The maps calculated with 1.0 Å error in the heavy atom position (Fig. 2) are of lower quality than the corresponding SIR maps calculated with the exact position of the heavy atom (compare dotted lines at Figs. 1 and 2). Higher-resolution SIR syntheses are quite

sensitive to this error; for example, at 1.0 σ the share of missed atoms in the SIR synthesis calculated at 2.2 Å resolution increases from 20% to 50% (dotted lines, Figs. 1c and 2c). On the other hand the quality of the SIR map calculated at 6.0 Å resolution does not change significantly (dotted lines, Figs. 1a and 2a). As previously for the exact heavy atom position, the subtraction of the bulk solvent contribution significantly improves the maps quality (compare dotted and solid lines Fig. 2a, 2b, 2c).

Because low resolution phases are less sensitive to the error in the heavy-atom position, when this error reaches 2.0 Å (Fig. 3) the quality of the SIR map calculated at 6.0 Å resolution becomes even better then the quality of SIR maps calculated at 3.0 Å and 2.2 Å. The most significant improvement after subtraction of the bulk solvent contribution corresponds also to the 6.0 Å resolution synthesis.

Fig. 4a shows the SIR electron density map and the atomic model superimposed. The map contains many noisy peaks in the solvent region. After subtraction of the bulk solvent contribution practically all these peaks in the solvent region disappear (Fig. 4b) which is not surprising. More important is that the map becomes better in the regions of the macromolecule as illustrated in more details in Figs. 5 and 6. A region of the light chain of Fab fragment is shown in the SIR map (Fig. 5a) and in the map with the subtracted bulk solvent contribution (Fig. 5b). After the correction many noisy peaks disappear from the map and the correct electron density appears for residues Gly1 and Val2 where it was absent previously. The same maps show also a significant improvement of the image quality in the region of the heavy chain, for example for the residues Gln77 and Phe79 (Fig. 6).

It can be also noted that the subtraction of the solvent contribution does not change the sharpness of the map; in all cases the maximum value of the map expressed in sigmas is practically conserved. Therefore, the improvement of the model trapping is indeed due to an improvement of molecular contours in the map.

Conclusions

Our tests show that if the bulk solvent contribution to the diffraction pattern is known at early stages of phasing, the quality of the electron density maps can be significantly improved by subtraction of this contribution. The most significant improvement is observed for syntheses calculated at low resolution with large errors in heavy atom position, because the bulk solvent contribution is more significant at low resolution and low resolution synthesis are less sensitive to errors in the heavy atom position.

Currently, the bulk solvent contribution is estimated only when a (preliminary) atomic model is available and used mostly to refine this model. The use of the bulk solvent contribution for map correction needs to develop more sophisticated methods for its estimation without use of atomic models (work in progress).

The authors thank Prof. C. Lecomte for his interest to the project and L. Torlay for technical help. The work was supported through CPER, Pole ``Intellegence Logicielle''.

References

Blow, D.M. & Rossmann M.G. (1961) *Acta Cryst.,* **14**, 1195-1202.

Brünger, A.T., Adams, P.D., Clore, G.M., DeLabo, W.L., Gros, P., Grosse-Kunstleve, R.W., Jiang, J.-S., Kuszewski, J., Nilges, M., Pannu, N.S., Read, R.J., Rice, L.M., Simonson, T. & Warren, G.L. (1998) *Acta Cryst.,* D**54**, 905-921.

Fokine, A.V., Afonine, P.V., Mikhailova, I.Yu., Tsygannik, I.N., Mareeva, T.Yu., Nesmeyanov, V.A., Pangborn, W., Li, N., Duax, W., Siszak, E., Pletnev, V.Z. (2000). *Rus. J. Bioorgan. Chem.*, **26**, 512-519.

Guex, N. & Peitsch, M.C. (1997) *Electrophoresis*, **18**, 2714-2723.

Jiang, J.S. & Brünger A.T.(1994) *J. Mol. Biol.,* **243**, 100-115.
Jones, T.A. (1978) *J. Appl. Crystallogr.,* **11**, 268-272.
Urzhumtsev, A.G. (1991) *Acta Cryst.*, **A47**, 794-801

Urzhumtsev, A.G. (2000) *CCP4 Newsletter on Protein Crystallography,* **38**, 38-49.

Figures

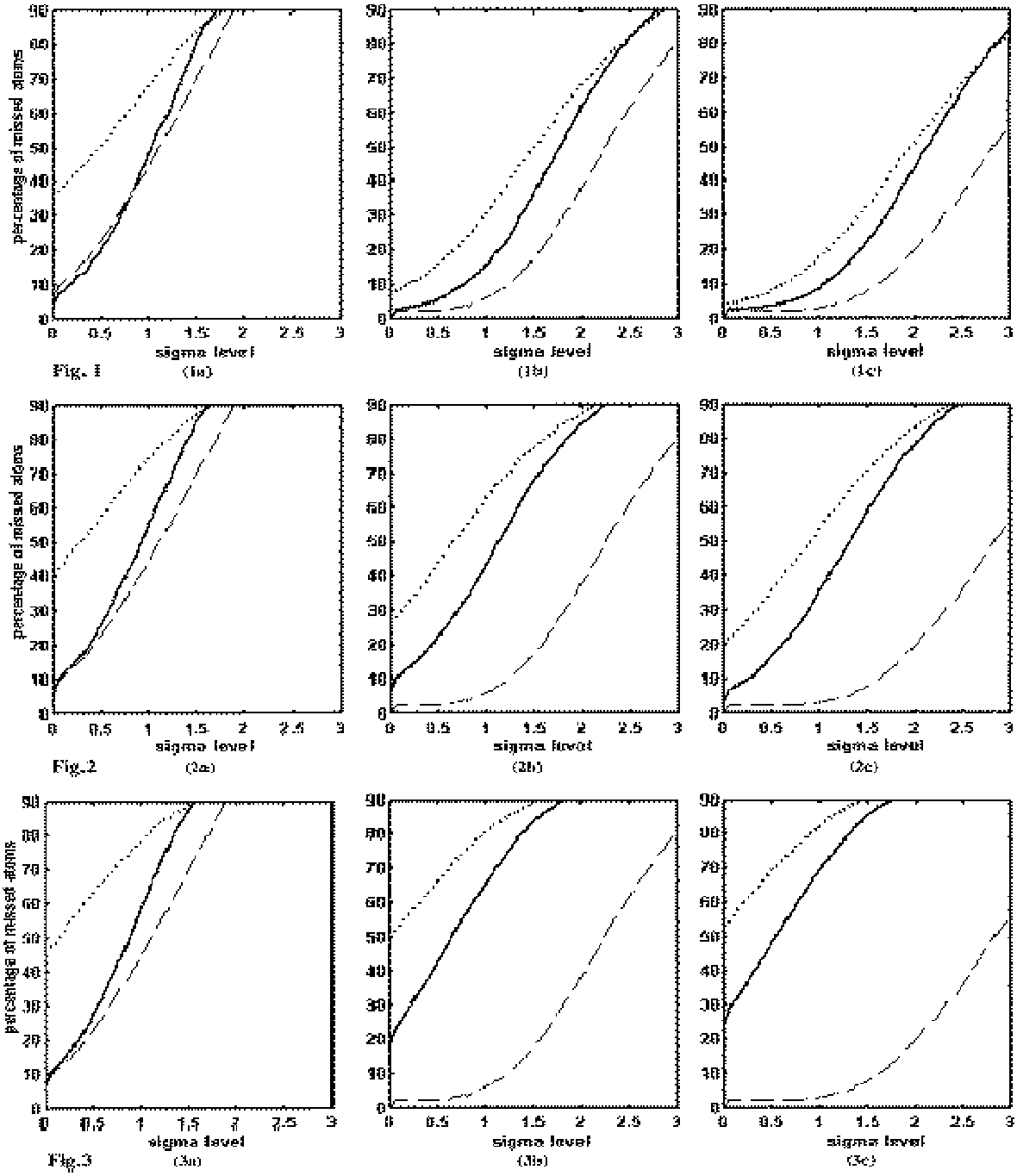
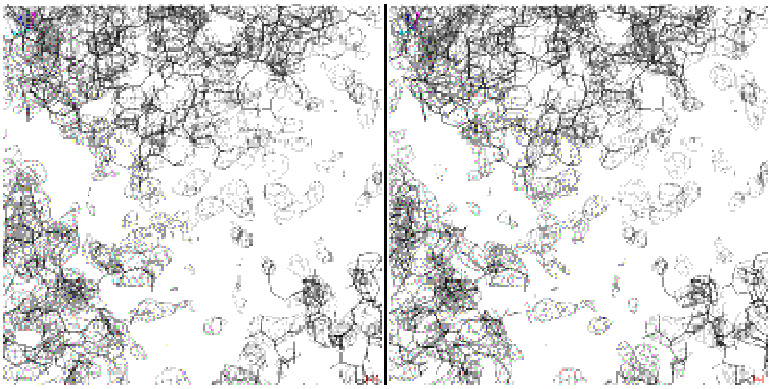


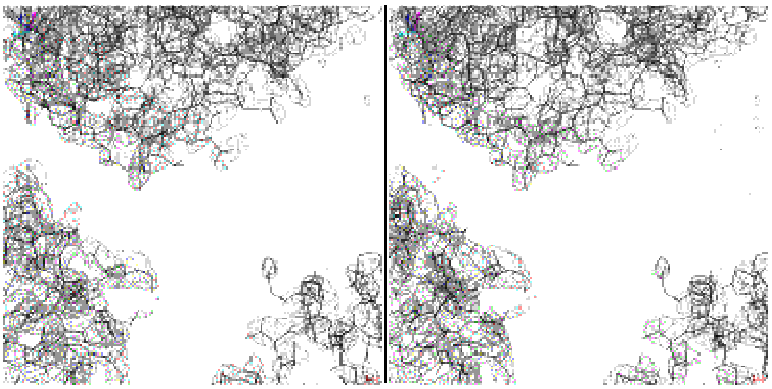
Fig. 1, 2, 3. Percentage of missed atoms depending on sigma level of equipotential surface.
Dotted lines: Maps calculated with the coefficients $m_{sin}|\mathbf{F}_{tot}|\exp(i\varphi_{sin})$.
Solid lines: Maps calculated with the coefficients $m_{sin}|\mathbf{F}_{tot}|\exp(i\varphi_{sin})-\mathbf{F}_{solv}$.
Dashed lines: Maps calculated with the coefficients $\mathbf{F}_{atom}$.

(1a), (1b), (1c) - position of heavy atom is exactly determined.
(2a), (2b), (2c) - position of heavy atom is determined with the error of 1.0 Å.
(3a), (3b), (3c) - position of heavy atom is determined with the error of 2.0 Å.

(1a), (2a), (3a) - maps calculated at 6.0 Å resolution.
(1b), (2b), (3b) - maps calculated at 3.0 Å resolution.
(1c), (2c), (3c) - maps calculated at 2.2 Å resolution.



(4a)

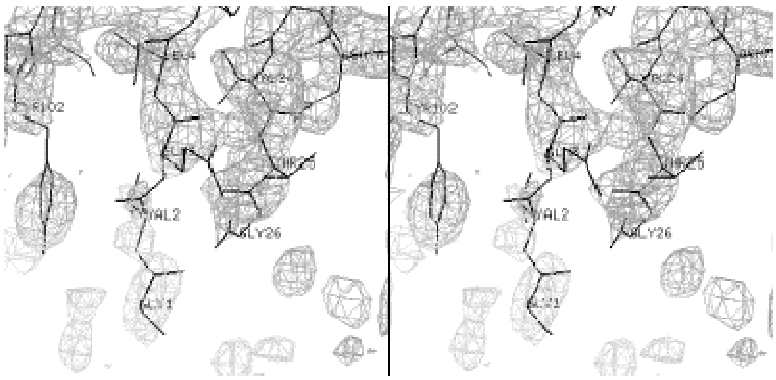


(4b)

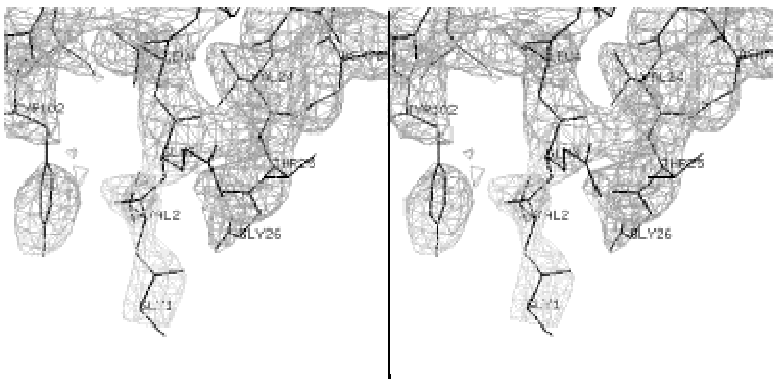
Fig. 4 Stereo-view of the atomic model (Fab LNKB-2) superimposed with electron density maps calculated at 3.0Å resolution with exact heavy atom position. Equipotential surfaces at the level 1.5 sigma are shown. The figure was done using the program CHAIN (Jones, 1978).

(4a) - map calculated with the coefficients $m_{sir}|\mathbf{F_{tot}}|\exp(i\varphi_{sir})$.

(4b) - map calculated with the coefficients $m_{sir}|\mathbf{F_{tot}(h)}|\exp(i\varphi_{sir}) - \mathbf{F_{solv}}$.



(5a)

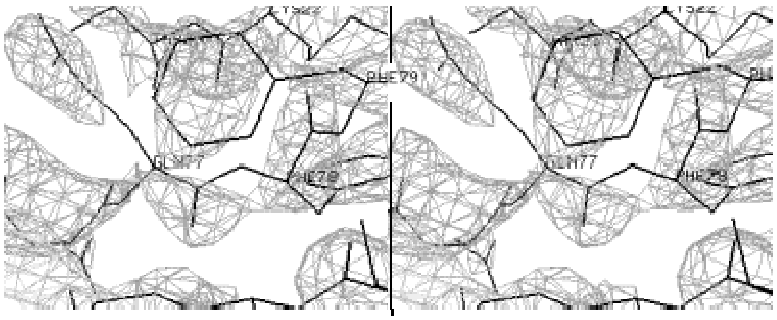


(5b)

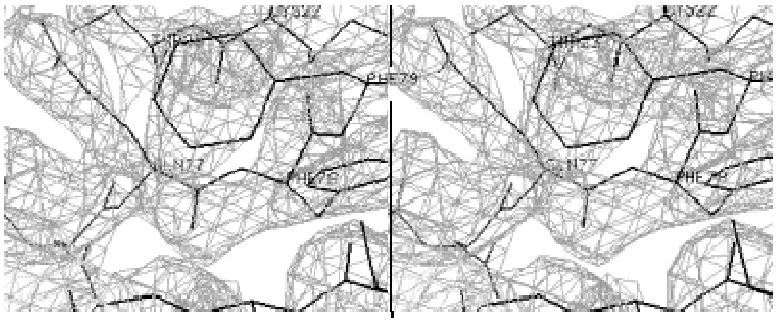
Fig. 5 Stereo-view of a fragment of the light chain of Fab LNKB-2 superimposed with electron density maps calculated at 3.0 Å resolution with exact heavy atom position. Equipotential surfaces at the level 1.5 sigma are shown. The figure was done using the program Swiss-Pdb-Viewer (Guex & Peitsch, 1997).

(5a) - map calculated with the coefficients $m_{sir}|\mathbf{F_{tot}}|\exp(i\varphi_{sir})$.

(5b) - map calculated with the coefficients $m_{sir}|\mathbf{F_{tot}(h)}|\exp(i\varphi_{sir})-\mathbf{F_{solv}}$.



(6a)



(6b)

Fig. 6 Stereo-view of a fragment of the heavy chain of Fab LNKB-2 superimposed with electron density maps calculated at 3.0 Å resolution with exact heavy atom position. Equipotential surfaces at the level 1.5 sigma are shown. The figure was done using the program Swiss-Pdb-Viewer (Guex & Peitsch,1997).

(6a) - map calculated with the coefficients $m_{cin}|\mathbf{F_{tot}}|\exp(i\varphi_{cin})$.

(6b) - map calculated with the coefficients $m_{cin}|\mathbf{F_{tot}(h)}|\exp(i\varphi_{cin})-\mathbf{F_{solv}}$.

Multiple rotation function

By L. Urzhumtseva & A. Urzhumtsev

Laboratory of Crystallography and Modelling of Mineral and Biological Materials, UPRESA 7036 CNRS, University Henri Poincaré, Nancy I, 54506 Vandoeuvre-les-Nancy, France

e-mail : sacha@lcm3b.uhp-nancy.fr

Introduction

One of principal tools of macromolecular crystallography, the molecular replacement procedure (Rossmann, 1972, 1990), is based on several assumptions the main of which is the following :

- the search model is sufficiently close to the model of the crystal under study (or to its large enough part) so that it fits best to the experimental structure factor magnitudes being placed at the correct position.

While the search directly in the 6-dimensional space is eventually possible, specially with modern computers and efficient algorithms (Chang & Lewis, 1997; Kissinger *et al*., 1999), it does not solve the problem when the model is imperfect and the Main Assumption is not verified. In such difficult cases, two separate consecutive searches in three-dimensional spaces, rotation and translation, can have an advantage.

The rotation search is traditionally done by comparison of Patterson maps, when even a partial model eventually can be recognised. More difficulties arise for search models composed from several blocks whose relative orientation is different from that in the molecule under study. For a small number of such rigid groups, a so-called PC refinement (Brünger, 1990; DeLano & Brünger, 1994) can help weaking therefore the Main Assumption.

At the step of the translation function, traditionally the structure factor magnitudes calculated from the search model are compared with the corresponding experimental values. If the search model is quite incomplete or contains significant errors, there is no many reasons why this fit will be best when the model is placed correctly (a long history of Molecular Replacement shows many examples *pro* and *contra*). Often, *a posteriori* analysis shows that the solution was in the list but it was difficult to recognise it among a very large number of possible positions. In this case, the knowledge of the model orientation could remove many spurious peaks for wrong orientations and thus solve or simplify the problem.

Alternatively, a search with incomplete models can be improved when maximum likelihood (ML) approach (Read, 1999, communication at the IUCr Meeting, Glasgow) is used. This technique allows to take into account a missing part of the model (note that here the Main Assumption changes its original form; the calculated structure factors are not fitted directly to the experimental values; some discussion can been found, for example, in Lunin & Urzhumtsev, 1999). However, the ML criterion is essentially more time consuming, there is no evident way to calculate it rapidly as it is done for the least-squares criteria (Navaza, 1994; Navaza & Vernoslova, 1995). In this case, the reduction of the number of possible orientations from several tens to a few possible variants is also crucial.

Multiple rotation function analysis

When the molecular replacement search does not give an evident solution, an old idea is to repeat the search varying the models (whole model, main chain model, Ca model, a model with deleted loops, etc), the set of structure factors (for example, selected by its resolution) or the parameters (for example, the integration radius). If no evident solution appears, many rotation functions are analysed together with the hope that the signal is consisted and can be identified in many of these functions. Such comparative analysis is not at all transparent because it is complicated to estimate visually the closeness of rotation angles, specially when the space group has symmetry operations and when the programs like AMoRe (Navaza, 1994) make some pre-rotation of the model before the search.

Such a comparison of several rotation functions is important also when the search is done with NMR models. Usually, they are several tens, neither of them is quite close to the correct model, rotation functions are quite noisy with the correct answer hidden in the middle of the list of peaks.

The goal of our current approach is to find unambiguously the molecular orientation in the crystal. With this, the Main Assumption can be replaced by a weaker one :

- a model taken in its correct orientation fits well enough to the experimental data in comparison with all its other positions in the same orientation

Computationally, the knowledge of the orientation (or a few orientations) allows to test possible positions with more sophisticated, powerful but time-consuming criteria, take into account all structure factor corrections like the bulk solvent correction etc. This article does not concern the study of this improved translation searches which is the object of our independent work and deals only with the rotation analysis of many rotation functions considered simultaneously.

In order to compare several rotation functions, the following procedure has been proposed :

1) Rotation functions are calculated varying the models and/or parameters of the rotation function including the resolution of the data set; if several search models are tested, they must be superimposed before to calculate the rotation functions ;

2) For each pair of the rotation angle triplets (α_m, β_m, γ_m) and (α_n, β_n, γ_n) coming from all lists of the peaks, the distance between them is calculated taking symmetry operations into account ;

3) A clustering procedure is applied for the calculated matrix of distances; the clustering results are represented in the form of a cluster tree and the clusters are defined varying the minimal interangular distance; for a chosen cut-off level of the interangular distance, the peaks inside the cluster are considered to be coincided, the size of all clusters is calculated and used as the information to choose the solution.

We believed that such procedure will give a signal because noisy peaks are distributed relatively randomly in the space and therefore are associated to different clusters while the correct peaks should be close enough each to others and will belong to the same cluster. Moreover, there is the second reason. Usual variations in the arrangement of secondary

structure elements will lead to several optimal orientations of the same model relatively close each to other – in one orientation one group of the secondary structure elements is superimposed better, in another orientation – another group.

Several comments can be done.

First, while in our work all molecular replacement searches where done by AMoRe (Navaza, 1994), the analysis of the rotation function is general and can be applied to lists of rotation function peaks obtained by any means but expressed in Eulerian angles α, β, γ (see Urzhumtseva & Urzhumtsev, 1997, for different rotation systems). The peak comparison is done for the *final* values of the rotation angles; this means that for the programs like AMoRe that preliminary puts the model to some special orientation, the lists of peaks *or1.s* are compared not directly but using corresponding files *tabl1.s* for the pre-rotations. The program gives the answer in both terms.

Second, the distance between a pair of rotation angles is expressed through the effective rotation angle k between two corresponding model orientations. If M_m and M_n are corresponding rotation matrices then the matrix of the relative rotation is calculated as the product $M_m M_n^{-1}$ and the corresponding efficient rotation angle is calculated as

$$k = \arccos\{ [\mathrm{trace}(M_m M_n^{-1}) - 1] /2 \}.$$

If the space group contains several symmetry operations, the distance is chosen as the minimal value of distance calculated for all symmetry related pairs. Distance between two clusters is defined as the minimal distance between all pairs of rotation angles, one from each cluster. When a noncrystallographic rotation presents in the crystal and its order and the axis direction are known from the self-rotation function, this operation can be also considered at the step of the distance calculation allowing to identify the pairs of angle triplets linked by this symmetry and to enforce the signal. Various distances can be defined for a given pair of angles. However, the architecture of the cluster tree will be the same for any of these definitions as soon as the distance increases with the effective rotation angle which seems to be logical.

Third, when the size of a cluster is calculated, the coincidence (or closeness) of higher peaks could cost more that the coincidence of lower peaks; therefore, the contribution of every rotation function peak can be weighted, for example, by its height. This can be interpreted as an integral measure of the peaks coincidence. The level at which rotation angles are considered to be coincided and the cluster size is calculated cannot be defined once forever. It is an important parameter of an interactive search of the answer.

The suggested procedure was realised in a FORTRAN program with an interactive interface in Tcl/tl (Ousterhaut, 1993). This program allows to read a list of rotation function files (*or1.s* in AMoRe format) and corresponding pre-orientation protocols (*tabl1.s* in AMoRe format), to define a list of symmetry operations including noncrystallographic symmetries if available, to obtain a cluster tree with references to the initial rotation functions, to define the cluster size with a variable cut-off level of the interatomic distance (Fig. 1). A selection of a cluster in the histogram indicates it in the cluster tree, gives the corresponding angle values and can provide with the atomic models rotated respectively.

First tests

This procedure has been tested first with a synthetic case and then was successfully applied to several experimental cases where the structure could not be solved previously by conventional molecular replacement procedures.

In this first series of tests, a simple but usual situation was simulated when the model is quite poor to give a strong signal in the rotation function. The N-terminal end (first 100 residues from 689 in the complete model) of a large protein, the elongation factor G (Aeverson *et al*., 1994) was used as the search model. Corresponding crystals have the symmetry $P2_12_12_1$, unit cell parameters a = 75.6, b = 106.0, c = 116.6 Å. The rotation function was calculated for the same model but in different resolution ranges : 4 – 15 Å, 4 – 10 Å, 4 – 8 Å, 5 – 10 Å. While individual rotation functions do not allow to identify the solution (Table 1), the merging of the rotation peaks in the cluster tree and cluster selection with the distance of 5 degrees, showed the correct orientation unambiguously (Fig. 1). This peak is stable in a large range of the distance cut-off. It can be noted also that, being presented as they are in the rotation function files, not all angles of this cluster are close between themselves from the first look (Table 1). When the distance cut-off decreases to 3 degrees the cluster is reduced to three closest peaks (first three lines of the Table 1) very close to the exact answer.

The second series of tests was done with experimental data of ER-1 protein (Anderson *et al*., 1996) called by the authors “A challenging case for protein crystal structure determination”. This small 40 amino-acids protein crystallises very densely in the space group C2 with the unit cell parameters a = 53.91, b = 23.08, c = 23.11 Å, β = 110.4°. The authors failed to identify the correct rotation using available 20 NMR models.

In this case of a small protein the data of the resolution of at least 8Å and lower should be excluded from the calculation due to a very strong influence of the bulk solvent on structure factors; Anderson *et al*. found that the best resolution cut-off is even 7 Å. Two sets of rotation functions were calculated varying the model, one at the resolution of 3-8 Å, and the second at the resolution of 4-8 Å. Similarly to the previous report (Anderson *et al*., 1996), AMoRe did not find the solution in any of these runs. In fact, the lists of the rotation peaks contain orientations close to the correct one; translation functions calculated with them also contain the correct position; however, it is not possible to recognise the answer among many tens of variants with a better correlation, sometimes even essentially better.

Multiple rotation function analysis with the functions calculated at 3-8 Å shows an extremely strong peak when the angular distance is equal to 9 degrees (Fig. 2, peak contribution to the cluster size was weighted by their height). When the angular distance is decreased to about 5 degrees, the cluster is split into 2 subclusters where the larger one is closer to the correct solution. If the orientation of the first model is chosen from this cluster, the translation function and the intermolecular distance allow immediately to identify the solution (Table 2) even by traditional translation search.

For the rotation functions calculated at the resolution 4-8 Å, the peaks are weaker and further from the correct orientation and their cluster analysis shows the answer unambiguously only at a quite high interangular distance, of order of 10°. A common analysis of all 40 rotation functions (20 at every resolution shell, 4-8 Å and 3-8 Å) showed again the correct orientation clearly.

In general, from our experience is seems to be efficient to start the clustering analysis from relatively high interangular distances, of about 10 degrees, to find the principal cluster or clusters and then decrease the distance level to select the solution (or few possible solutions, in general case) inside them. Very high interangular distance, of 20 degrees and higher, starts to put together the peaks which have nothing in common and can lead to misleading results.

>The third series of tests has been done with experimental data of thioredoxin h from *Chlamydomonas reinhardtii* (A. Aubry, personal communication) where it was not possible to solve the structure by the conventional molecular replacement using available 23 NMR models (the structure has been solved in a different way, the paper is in preparation). In this case, when the standard AMoRe protocol does not give the answer, the clustering with the distance level of 3° and higher shows immediately the correct orientation corresponding to the cluster of the size 3 times larger that the size of the next cluster. Use of an existing noncrystallographic symmetry doubled the signal. Details of this test and some others will be discussed elsewhere.

Conclusions

Cluster analysis of multiple rotation functions can be useful in many practical situations when searching for the model orientation with imperfect models. A relatively random distribution of noisy peaks allows to identify the signal which appears systematically (but, maybe, weakly) in the rotation functions. Naturally, the cluster analysis gives an information which is definitely more reach than a single orientation for such or such model. The use of this information for further steps of molecular replacement, specially for the translation function, will be discussed elsewhere.

Acknowledgment

The authors thank C. Lecomte for his interest to the project, A. Aubry for the thioredoxin data available before their publication, and L. Torlay for the technical help.

References

Ævarsson, A., Braznihnikov, E., Garber, M., Zhelnotsova, J., Chirgadze, Yu., Al-Karadaghi, S., Svensson, L.A. & Liljas, A.(1994). *EMBO Journal*, 13, 3669-3677.

Anderson, D.H., Weiss, M.S. & Eisenberg, D. (1996) *Acta Cryst.,* D**52**, 469-480.

Brünger, A.T. (1990) *Acta Cryst.,* A**46**, 46-57.

Chang, G. & Lewis, M. (1997) *Acta Cryst.,* D**53**, 279-289.

DeLano, W.L. & Brünger, A. (1995). *Acta Cryst.,* D**51**, 740-748.

Kissinger, C.R., Gehlhaar, D.K. & Fogel, D.B. (1999) *Acta Cryst.,* D**55**, 484-491.

Lunin, V.Y. & Urzhumtsev, A.G. (1999). *CCP4 Newsletter on Protein Crystallography*, **37**, 14-28.

Navaza, J. (1994) *Acta Cryst.,* A**50**, 157-163.

Navaza, J. & Vernoslova, E. (1995) *Acta Cryst.,* A**51**, 445-449.

Ousterhout, J.K. (1993) *"Tcl and the Tk Toolkit".* Addison-Wesley Publishing Company.

Rossmann, M.G. (1972) *The Molecular Replacement Method.,* Gordon & Breach; New York, London, Paris.

Rossmann, M.G. (1990) The Molecular Replacement Method. *Acta Cryst*., A**46**, 73-82.

Urzhumtseva, L.M., Urzhumtsev, A.G. (1997) *J.Appl. Cryst.,* **30**, 402-410.

Table 1. Rotation functions analysis for the N-terminal end of the EFG. The correct solution is (27.6, 21.9, 148.3).

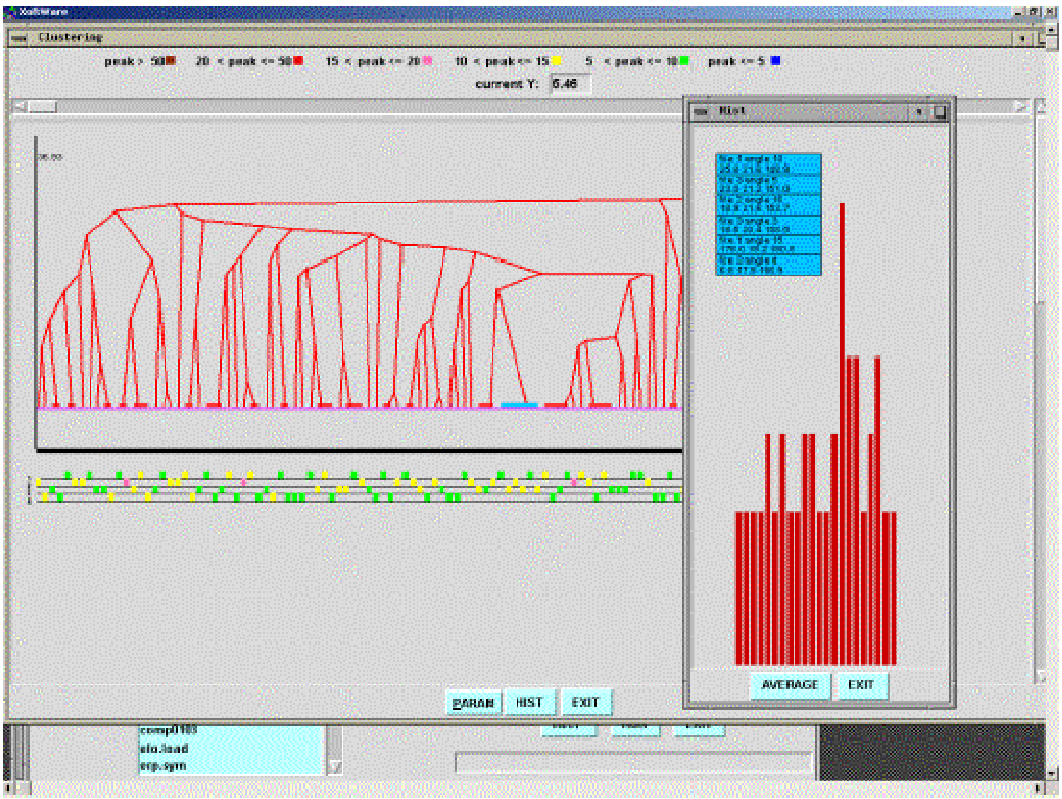
Resolut. limits	Sequen. N of the peak	α,β,γ	Height of the peak	Height of the1st peak	Height of the 2nd peak
4-10	10	25.8, 21.6, 148.9	10.0	13.2	12.4
5-10	5	23.0, 21.2, 151.0	11.3	14.1	13.1
4-15	16	18.9, 21.6, 153.7	13.4	18.5	15.7
5-10	3	18.5, 20.4, 158.5	11.3	14.1	13.1
4-10	15	176.0, 18.2, 180.8	9.8	13.2	12.4
5-10	4	6.8, 17.9, 166.9	11.3	14.1	13.1

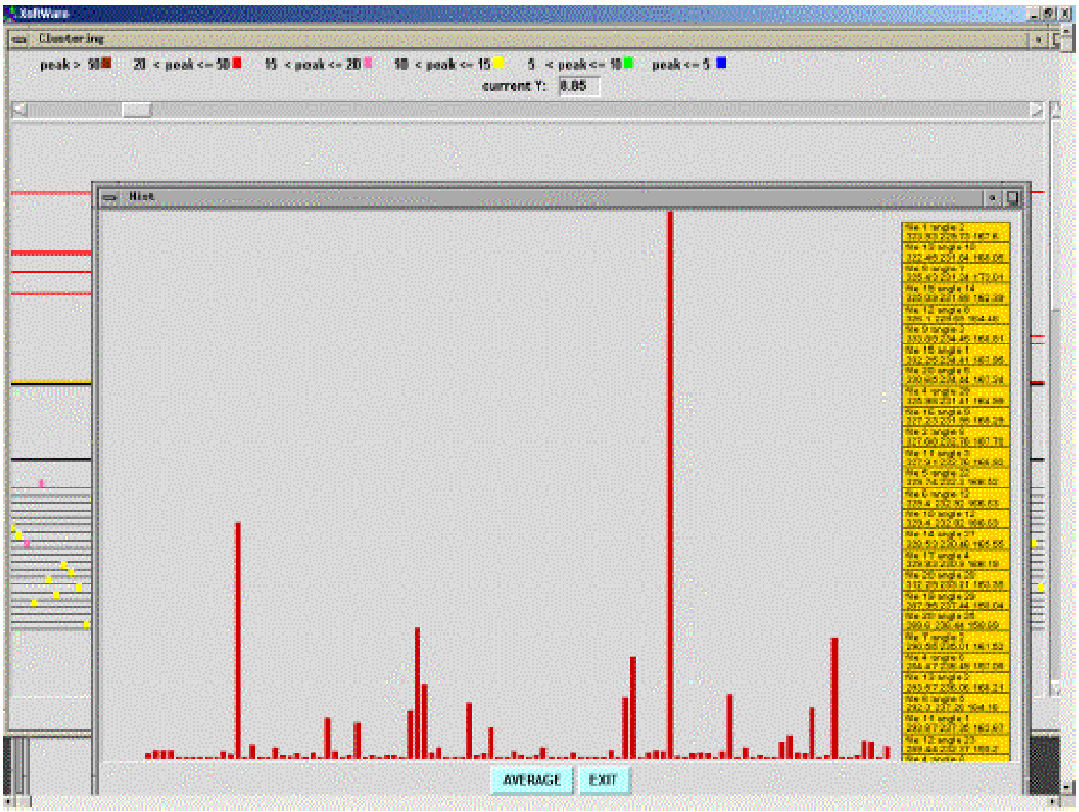
Table 2. Translation search for the ER1 (first NMR model) for the rotation angles defined by the multiple function analysis as (116.2, 73.3, 209.9). The correct orientation found from the optimal model superposition is (113.3, 77.2, 200.3) and the position is (0.3151, 0.0, 0.4892). Appropriate solutions are indicated by *.

Peak N	α,β,γ	Molecular position	Correlat.	Intermolec. distance
1	113.1 77.9 202.0	0.4260 0.0 0.4493	49.2	7.0
2**	110.8 74.6 207.7	0.3209 0.0 0.4936	37.5	14.6
3*	114.2 76.6 203.6	0.3823 0.0 0.4902	35.4	12.9
4	113.7 77.9 204.8	0.4714 0.0 0.3263	30.7	7.1
5	112.8 77.7 207.6	0.0837 0.0 0.3635	27.7	13.2
6	113.0 72.9 210.2	0.2043 0.0 0.4097	26.8	12.4

Fig. 1. Copy of the screen during the program session when comparing several rotation functions for the EFG N-terminal model (see Section 'First Tests'). The correct orientation corresponds to the cluster (shown in light bleu in the cluster tree) with the largest cluster size (shown in the inserted window). Initial rotation angles (as they are done in the *or1.s* files) are shown in bleu frame. Several parallel lines with squares below the cluster tree show the rotation peaks in different rotation functions with their height indicated by colour. A variable cut-off interangular distance is indicated by a pink line above the zero level (black line)

Fig. 2. Cluster size analysis for the ER-1 protein (see Section 'First Tests'). The correct orientation corresponds to the cluster with the largest cluster size. Note the contrast of the signal. Final rotation angles (corresponding to sequential rotation defined in *tabl1.s* and *or1.s* files) are shown in yellow frame.


Clustering
peak > 50
20 < peak <= 50
15 < peak <= 20
10 < peak <= 15
5 < peak <= 10
peak <= 5
current Y:
0.46
Hist
PARAM
HIST
EXIT
AVERAGE
EXIT


Clustering
peak > 50
20 < peak <= 50
15 < peak <= 20
10 < peak <= 15
5 < peak <= 10
peak <= 5
current Y:
0.85
Hist
AVERAGE
EXIT

An Open Source Multi-purpose Programming Environment for Macromolecular Crystallography

Thomas Hamelryck[*] and Morten Kjeldgaard[+]

[*] *ULTR department, Free University of Brussels (VUB), Paardenstraat 65, B-1640 St-Genesius-Rode, Belgium.*
[+]*IMSB, Aarhus University, Gustav Wieds Vej 10C, 8000 Aarhus C, Denmark.*

Abstract

A computational framework for macromolecular crystallography is presented, consisting of a collection of freely available classes that perform various computational tasks, currently mainly including building and analyzing macromolecular crystal structures. These classes can be readily used or extended to quickly create simple applications, but they can also be used to build more complicated and elaborate programs. As an example of this, we present a program to interactively build, visualize and analyze macromolecular crystal structures. The project is currently in a preliminary state, and this short overview also serves as an invitation to join in the development of this Open Source project.

1 Introduction

The rapid growth of the amount of crystal structures of macromolecules puts high demands on the computational tools that scientists use to build, refine and analyze crystal structures. Traditionally, the crystallographic community has used a myriad of different, mostly FORTRAN, stand alone utilities, tied together in various improvised ways. In the future, it will be more and more difficult to use this approach, since the sheer number of structures will make a more integrated approach necessary.

Another, less benign evolution is the increasing spread of proprietary (and often expensive) scientific software, which cannot be modified, extended and/or properly understood. In addition, the procedures implemented in such programs are commonly not well documented, let alone published. This development is particularly surprising since the Open Source software development model is gaining more and more popularity in areas outside science.

Another impediment to future software development in the field of macromolecular crystallography is the declining number of students knowledgeable about programming, and the FORTRAN programming language in particular. It can therefore be foreseen that the maintenance and development of the many existing FORTRAN programs will be increasingly difficult. Add the fact that the basic algorithms in these programs tend to be obscured by the constructs required by the limitations of the language itself.

In this article we present a comprehensive software toolkit (called "Birdwash"), consisting of a set of classes that deal with various aspects of structure building, refinement and analysis. These classes can be used to rapidly create programs that deal with various

computational problems. A very important aspect of the toolkit is that it is easy to extend with new classes, add novel features to existing classes or even integrate external software components, including external databases.

The toolkit encourages code reuse and distribution, and is subject to the GNU Public License (GPL).

2 Implementation

The toolkit is implemented in Python, an interpreted, object oriented programming language. Python is easy to learn, freely available and has a rapidly growing user base. The main disadvantage of Python is that it is considerably slower than a compiled language like FORTRAN, C or C++. However, this problem is easily overcome by migrating speed critical parts of the program to extension modules written in a compiled language (typically C, C++ or even FORTRAN). For this reason, most of the time-demanding calculations will take place in binary program segments, and the interpreted Python language will merely control the flow of the procedure. The use of Python in scientific computing is rapidly catching on, and is used in related fields such as molecular modeling and visualization (MMTK by Konrad Hinsen, PyMol by Warren DeLano and MSV by Michel Sanner), and crystal structure refinement (the PHENIX project by Paul Adams, Ralf Grosse-Kunstleve *et al.* of the Computational Crystallography Initiative). The Python programming language is also at the heart of the Nonius Collect program suite by R.W.W. Hooft, for the control data collection, integration of images, and image analysis.

One immediate advantage of the approach is the existence of a great number of available modules available for the Python language. The most important example is the Numerical Python (``Numpy'') module, which very efficiently implements vector and matrix operations, along with a number of associated procedures from the LAPACK library, such as matrix diagonalization. Lengthy numerical calculations are slow when performed extensively in Python itself. It is therefore desirable to perform as many calculations as possible in Numpy. Fortunately, it is often possible to recast existing algorithms in a set of array operations. Fig. 1 shows how a simple peak search algorithm can be formulated as a series of shift-compare operations on the entire electron-density array, using the rich set of array manipulating functions available in Numpy.

Another example of the use of standard Python extension modules, is the very efficient XML[1] parser, which is used to parse the content of configuration and data files. This enables the storage of information about connectivity, bond angles, bond lengths, etc. in XML format, and allows for easy parsing, browsing and validation.

Figure 1: A simple peaksearch algorithm implemented in Python, using the Numerical Python extension module. From the electron density Numpy array, different copies are made, each with a different shift of the map. These copies in fact share the same physical memory, but represent different modes of access. The ``greater'' function returns an array of elements with the value 1 if the condition is true, else 0. The ``nonzero'' function returns a list of indices where the array elements are different from zero. At the end of the computation, the ``peaklist'' array contains peak coordinates in grid space. The calculations takes around 5 seconds on a 400 MHz Pentium II processor.

```
shift_zero = ed[1:-1, 1:-1, 1:-1]
shift_left = ed[:-2, 1:-1, 1:-1]
shift_right = ed[2:, 1:-1, 1:-1]
shift_up = ed[1:-1, :-2, 1:-1]
shift_down = ed[1:-1, 2:, 1:-1]
```

```
shift_hither = ed[1:-1, 1:-1, :-2]
shift_yon = ed[1:-1, 1:-1, 2:]

flag = greater (shift_zero, shift_left)
flag = flag * greater (shift_zero, shift_right)
flag = flag * greater (shift_zero, shift_up)
flag = flag * greater (shift_zero, shift_down)
flag = flag * greater (shift_zero, shift_hither)
flag = flag * greater (shift_zero, shift_yon)

peaklist = nonzero(flag)
```

3 Structure representation

Birdwash uses a structure/model/chain/residue/atom (SMCRA[2]) hierarchy to represent a crystal structure. Biopolymers like proteins or nucleic acids are grafted on this basic representation. New polymer types can easily be added by providing a simple description file in XML format that described bonds, angles and torsion angles in the polymer.

The SMCRA hierarchy is general enough to accommodate any type of biological polymer, as well as smaller entities such as waters, cofactors, etc. The classes used to represent a structure provide various operations like copy, delete, replace and add operations. However, while the SMCRA hierarchy contains all relevant information that can be perused from a PDB entry, it is not suited for computational tasks. The Structure class is typically handed to a computational client who extracts the information it needs. Symmetry expansion, for example, is handled by extracting all atomic positions into a Numpy array, which is multiplied with the symmetry operator and the atomic positions are reinserted into the SMCRA structure. While such an approach may seem cumbersome, it is extremely fast and strictly maintains the modular and cooperative approach of the toolkit.

Support for alternate atomic positions are fully supported by the toolkit, and from the point-of-view of the toolkit, there is no difference between an atom in one position, or one split between alternate sites. For example, it is possible to apply a rotamer transformation to the alternate configuration of a side chain. Birdwash also deals with anisotropic B factors if they are present.

To demonstrate a simple use of this Fig. 2 shows a Python code snippet downloads entry 1ABC from an FTP server and prints the average B factor of all C-alpha atoms. Afterwards the structure is deleted.

Figure 2: Simple program using the toolkit. First, the program opens a parser with input from an FTP site. The (structure, model, chain, residue) hierarchy is traversed to extract the temperature factor of all C-alpha atoms, and the average temperature factor of these atoms is computed.

```
from Birdwash.Parsers import PDBFTPParser
parser = PDBFTPParser("ftp.ebi.ac.uk", "/pub/pdb/")
structure = parser.get("pdb1abc.ent")

avb = 0
nr_atoms = 0

for model in structure.get_list():
    for chain in model.get_list():
        for residue in chain.get_list():
         if residue.has_key["CA"]:
```

```
                avb = av_b + residue["CA"].get_bfactor()
                nr_atoms = nr_atoms+1

print "Average B factor of CA atoms", av_b/nr_atoms
```

3.1 Polymer classes

The SMCRA hierarchy is a convenient way to store the information in a coordinate entry. However, this representation is in most cases too simple. For example, it contains no information about which atoms are bonded to which, and which residues are attached to which. This is the role of the Polymer classes. For example, from the XML residue dictionary, the Polymer classes knows about intra- and inter-residue restraints, and can therefore participate in structure regularization. In practice, a specific class exists to deal with a special polymer type. Most biological polymers are linear, so protein and RNA polymers are handled by classes which are subclasses of the LinearPolymer class, which in turn is a special subclass of the Polymer class.

Another higher level organization of molecular structures are that of biological assemblies of multimers. In crystal structures, such multimers may exist in the asymmetric unit of the crystal, or may be formed by the application of crystal symmetry to the asymmetric unit. A class hierarchy capable of handling biological entities in this manner will be developed in the future.

3.2 Parsers

Birdwash contains parser classes for the PDB and the mmCIF format. Other file formats can be added easily since the code that deals with the actual building of the structure is stored in a separate builder class, and can thus be easily reused. Parsers are available that directly download structures from FTP repositories (see Fig. 2). The PDB parser is completely implemented in Python, whereas the mmCIF parser on the lowest level makes use of a lexical token analyzer written in C and lex. A parser capable of loading coordinate data directly from an SQL relational database is currently under development.

3.3 Persistence

The Zope Object DataBase (ZODB) is a high-performance, transaction based database written in Python, that provides objects with Persistence storage and management. Persistence of the living objects in Birdwash is achieved using a ZODB database . This is especially important for the graphics program Birdbuilder.

4 Something about basic crystallographic data structures

4.1 Symmetry

Classes to deal with crystallographic symmetry are fully implemented in Birdwash. For example, methods are present to lookup symmetry operations based on space group name, to find crystal contacts and to apply arbitrary transformations to atomic co-ordinates. Internally, spacegroup lookup is implemented using Ralph Grosse-Kunstleve's SGInfo library.

4.2 Structure factors

Currently, only parts of the low level infrastructure dealing with structure factors have been developed. Structure factor file parsers exist for MTZ, CNS and mmCIF format. The mmCIF parser makes use of the same general mmCIF parser module being used for parsing coordinates, while the MTZ format is handled entirely in Python. On the lowest level, a Python module emulating the low level file i/o of the CCP4 suite is reading the binary data from disk. The actual data the parsers is being delivered in Numpy arrays. We are currently tracking the development of the Clipper library by Kevin Cowtan. Such an approach is very easily integrated into Birdwash, and fits well with the overall scheme.

The object oriented approach is attractive within crystallographic computing because it is straightforward and logical to deal with data as objects. For example, when thinking about a Fourier transformation of structure factors it is evident that applying an FFT operation on the structure factor *object* would return a map *object* (Fig. 3). The actual FFT is based on the Python bindings for the FFTW package which is under the GPL, but it may be advantageous to wrap Ten Eycks FFTLIB library, since it is fast and deals with crystallographic symmetry.

Figure 3: Pseudo-code snippet demonstrating how crystallographic data items can conveniently be treated in an Object Oriented Programming approach. A reflection file is read and stored in the object ``data". The appropriate data is extracted in columns and passed to the FFT procedure which generates a map object.

```
data = hklFile("data.mtz")
fo = data["fobs"]
fc = data["fcalc"]
phi = data["phi"]
map = fft(2*fo-fc, phi)
map.write("map.cdf")
```

4.2.1 Persistence of structure factor data

In Birdwash, structure factors are stored in netCDF (network Common Data Form) files. The netCDF format is actively being developed by the Unidata Program Center in Boulder, Colorado, and defines a machine-independent, self-describing format for representing scientific data. Multidimensional data may be accessed one point at a time, in cross-sections, or all at once. Data is directly accessible, permitting efficient access to small subsets of large data sets. Bindings for various programming languages such as C, C++, FORTRAN, Perl or Java exist, and Python bindings that deliver the data in form of Numpy arrays have been developed by Konrad Hinsen. Utilities for converting structure factor information to and from netCDF data format exist as a part of Birdwash and are easy to program. We encourage the crystallographic community to consider the netCDF format as a truly portable, cross-platform general purpose format for binary data.

4.3 Density maps

The low level infrastructure for dealing with electron density maps is available in the toolkit. Currently, data from CCP4 and CNS format density maps can be imported. The three-dimensional contouring in Birdbuilder (see section 5) is carried out by a Python module which in turn calls a two-dimensional contouring routine implemented in C. Remarkably, there is no noticeable speed penalty in this implementation compared to the equivalent functionality in O [1].

The Birdwash generic persistence format for electron density maps (and masks) is again based on the netCDF standard, offering platform independence and random access.

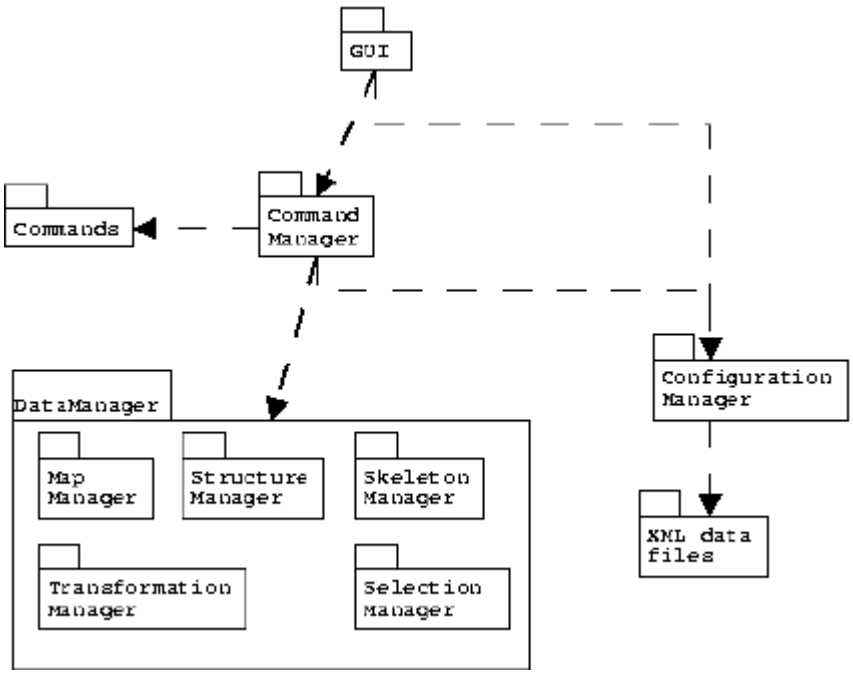
5 The Birdbuilder application

Birdbuilder is an application framework that deals with the interactive/automated building of a crystal structure in the electron density. In addition, it provides tools to analyze structures and will in the future interact with various databases. Birdbuilder uses much of the above described functionality. The three-dimensional graphics interface of Birdbuilder is based on OpenGL by way of the OpenGL Python bindings. The GUI toolkit used is wxPython, which is based on wxWindows, a general purpose GUI toolkit running on top of several OS'es.

5.1 Overall architecture

Birdbuilder consists of three large parts: the structure manager, the command manager and the GUI (Fig. 4).

Figure 4: UML (Universal Modeling Language) figure of the overall architecture of Birdbuilder.



Every action performed on a loaded structure is performed by calling the execute method of a specific Command class. These Command classes are loaded on program startup by the command manager. The command manager also takes care of undo operations by calling the undo methods of the Command classes involved. This architecture allow the moderately advanced user to add novel commands to the program, simply by writing a new Command class and adhering to a couple of conventions. If the user supplies an ``Undo'' method in a Command class, the actions are reversible as well.

Note that Birdbuilder does not simply provide a macro facility: it provides a full blown programming environment that can be used to extend the program. It also allows the user to install only the commands that are relevant to the intended use of Birdbuilder (eg. it is not necessary to install commands that deal with nucleic acids if the program will only work with proteins).

5.2 Features

Birdbuilder (see Fig. 5 for a screenshot) currently incorporates a number of the features expected in an interactive molecular graphics program. The current version although still in early development contains (unordered list):

- Fully dynamic data structure for molecules (add/delete/replace/move/etc.)
- Visualizing Maps/Molecules in various ways
- Manipulating molecules (side chain rotamers, also of residues with atoms in alternate positions; moving/rotating residues or zones of residues, ...
- A conjugate gradient minimization module
- Automatic superposition of molecules (using a graph theory based method)
- Symmetry
- Skeletonized maps/Bones
- Measuring angles, dihedrals, distances, drawing unit cells, etc.
- Disordered residues fully supported
- Deals with many structural features like RNA, DNA, carbohydrates, glycosylation, disulphide bridges
- Input of coordinate files in PDB or mmCIF format, from file or directly via FTP
- XML configuration files XML is a standard format developed to store data in an easy accessible and flexible way. All data (eg. rotamer libraries, polymer definitions, etc.) are stored in XML files. This means that the data can easily be altered or viewed using one of the numerous XML software tools available. It also means that you can use these tools to work with your own XML configuration files when you are implementing new functionality.
- The program is fully extensible by the user with new commands. The user can drop a Python module in a specific directory, and the new command will appear on the menubar. The command module needs to adhere to a few standard rules, and can easily be customized from the available template. The command has full access to all tools in Birdbuilder/Birdwash.
- Multiple drawing canvasses
- Support for insertion codes, multiple models (for NMR structures), etc
- Unlimited Undo/Redo Extensible with new commands
- Persistent objects in ZODB database
- GNU Public Licence

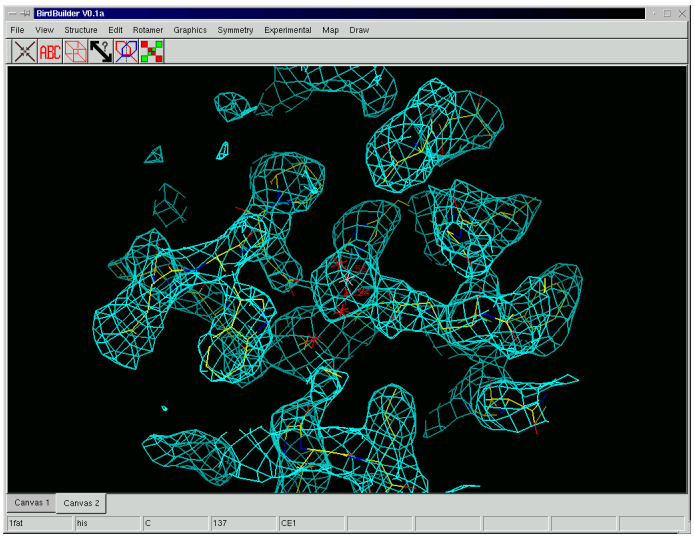


Figure 5: A screenshot of Birdbuilder with a contoured electron-density map.

An important aspect of Birdbuilder will be interaction with various databases. Currently, the interaction with many databases typically occurs through a web based interface. This is fine for casual users, but more experienced users would like to see a more sophisticated interface to these databases. We are currently trying to put our efforts in tune with external database efforts. Birdbuilder has the necessary tools to efficiently interact with external databases over the Internet, either using simple SQL queries or via a more advanced CORBA interface.

As a first start, we have implemented the automatic retrieval of a structure from the PDB database (ie. you provide a PDB identifier and the structure immediately pops up on the screen). More advanced database connectivity will need the co-operation of the various maintainers of these databases.

6 Conclusions

The current project started in late 1999, and has been under development for a little over a year. We were initially curious about the idea of programming a major program system like a molecular graphics program in an interpreted language like Python. Indeed, it appears to be a crazy idea! It was not at all clear at the time that the speed of such a program be acceptable. Time and experience has shown that the choice was right. From working with the system, and developing it, we have experienced that the speed of development is probably a factor of 3-4 faster than using a compiled language. This is also partly due to the advantages of using an Object Oriented programming method, partly to due to the extremely rich selection of ``add-on'' Python modules on the Internet, and partly due to the rich set of build-in types in Python, such as lists and dictionaries. With these tools, one can very easily test algorithms and implementations without constantly having to hand-craft these basic data structures.

The Numerical Python module is an essential prerequisite of any serious computation using Python. It is extremely efficient, versatile, and fun to work with. And, last but not least, is actively developed and maintained.

There was never a doubt that the three-dimensional graphics API for Birdbuilder was going to be OpenGL. Finding the right GUI took a bit of testing, until we finally settled on wxPython, which we now realize was the best choice. It is very complete, robust, and flexible, and again, actively and enthusiastically maintained.

A main design decision was to make use of existing libraries where available on the Internet. Why spend time programming a conjugate gradient or simulated annealing refinement procedure, when it has already been done by specialists?

Birdbuilder and the Birdwash toolkit are still in an early experimental stage. At this point, we have constructed the basic data structures and a basic functionality, and we are hoping to gradually involve other daring computer-nerds and programmers in the project. It is still a long way from being a tool for the Casual User. However, it is our experience that many students of crystallography in their projects meet problems that require some programming to be done. Rather than starting from scratch, and having to work out a bunch of basic problems that have already been solved, it is better to stand on the shoulder of others, and spend more time on the problem itself.

All software components can in principle run on many different platforms, including such exotic ones as MS Windows. However, the hardware mainly targeted is an mid-upper

range PC running Linux. We supply a set of RPM packages for the extra software needed to run the package. The Birdbuilder/Birdwash directory tree is currently only available as a gzipped tar archive.

Additional information is available on the Birdwash home page.

Acknowledgements

TWH has a grant from the Fonds voor Wetenschappelijk Onderzoek (FWO). MK is the recipient of a Hallas-Møller fellowship from the Novo-Nordisk Foundation.

References

[1] T. A. Jones and J-Y. Zou and S. W. Cowan and M. Kjeldgaard (1991) ``Improved Methods for Building Protein Models in Electron Density Maps and the Location of Errors in these Models'', Acta Cryst. A47, 110-119.

Footnotes

[1]eXtensible Markup Language

[2]pronounced: *simcra*

Recent improvements to *Mosflm* - version 6.11

Harry Powell, MRC-LMB, Hills Road, Cambridge, CB2 2QH
harry@mrc-lmb.cam.ac.uk

Mosflm version 6.11 was released recently. It includes many small bug fixes and improvements over previous releases. This article describes the improvements introduced since the release of version 6.01 in July 1999 (see CCP4 Newsletter 37, October 1999).

Further information about the program can be found at http://www.mrc-lmb.cam.ac.uk/harry/mosflm, including an on-line keyword search and brief command synopsis to complement the help document and user guide.

New and improved features

1. The spot-finding routines have been improved and the DPS autoindexing code has been modified to search for more likely combinations of basis vectors. This results in more robust autoindexing and allows more rapid generation of the set of reciprocal lattice projections. In addition, the DPS autoindexing estimates the longest cell edge likely by analysing the spot separation and the detector parameters rather than assuming a default value.

   In order to aid discrimination of correct solutions, the user can now choose to pre-refine the autoindexing solutions before making a choice.

   Autoindexing using the DPS code can now be performed without the GUI running, using the **AUTOINDEX DPS** keyword pair. A number of options are available to the user and described in the user guide, viz:

   - unknown cell - default mode, no extra keywords required
   - known space group, unknown cell - needs **SYMM** keyword
   - known space group and cell - needs **SYMM** and **CELL** keywords
   - prerefine solutions to aid discrimination - needs **REFINE** subkeyword on the **AUTOINDEX DPS** line
   - choosing a particular solution from the list - needs **SOLUTION** subkeyword on the **AUTOINDEX DPS** line
2. In releases of *Mosflm* prior to version 6.10, post-refinement of cell and detector parameters could only be performed when a reasonable number of reflections were partial over only two adjacent images; this limitation has now been removed so that reflections spread over many images can now be used. This allows correct processing of fine-phi sliced data or images with high mosaicity.
3. Estimation of mosaicity routine included. Once an image has been indexed, it is now possible for the program to calculate an initial estimate of the mosaicity of that image.
4. The main log file produced by the program can now be written with a version number from 1 - 99 by setting the environment variable **MOSFLM_VERSION_NUMBERS** to be **TRUE**. This can avoid the sometimes annoying problem caused when the log file is overwritten when the program is run.
5. Several new image formats have been introduced from version 6.10; the most important is the CBF file. This is a binary representation of an IUCr agreed standard, the imgCIF. The principle advantages of this over other images are that it is portable between detxector types and comprehensive experimental information is

included in the image file in a uniform way. This is good news for users for two principle reasons;

- image files written as CBFs are processable by a single version of any program which supports the format; CBF files produced by a novel detector can be processed by any CBF compatible program and neither the program nor file should need modification.
- CBF files contain all information relevant to an experiment, e.g. wavelength of radiation, polarization, beam centre, oscillation angle etc., so users are no longer dependent on their notebooks for this information.

The following specific image types can now be processed; DIP 2040, R-Axis V, Oxford PX210 CCD and Brandeis 2x2 CCD.

6. Programmers working for the CCP4 in Daresbury have identified and corrected several long-standing errors in the XDL_VIEW code which have caused a variety of problems in porting *Mosflm* to new platforms. These robust bug fixes mean that *Mosflm* can now run on Linux PCs using 24- and 32-bpp colour graphics, and separate **XDL_VIEW** code is no longer needed for Compaq Alpha workstations with the 4D60T (and similar) graphics processors.
7. Data harvesting code (written by Kim Henrick, EBI) has been included to help with tracking of experimental method and results.

The building and installation of *Mosflm* has been rationalized. A **build** shell command file sources **include.$HOSTTYPE** files to set compilation and linking flags for different platforms. These files can be modified easily with a text editor for new platforms.

Since CCP4 version 4.1, *Mosflm* has been distributed with the CCP4 suite and is available from ftp://ccp4a.dl.ac.uk/pub/ccp4 as well as via the MRC-LMB website.

March 2001

Recent CCP4BB Discussions

Maria Turkenburg (mgwt@ysbl.york.ac.uk)
March 2001

To make things much easier for both the users of the bulletin board and us writing this newsletter, *members who ask questions or instigate discussions on the board are now asked (urged!) to post a summary of all the reactions received*, whether on or off the board.

The introduction to Martyn Winn's summary article in the October 1999 newsletter also goes for this article:

For each subject below, the original question is given in italics, followed by a summary of the responses sent to CCP4BB (together with some additional material). For the sake of clarity and brevity, I have paraphrased the responses, and all inaccuracies are therefore mine. To avoid misrepresenting people's opinions or causing embarrassment, I have not identified anyone involved: those that are interested in the full discussion can view the original messages (see the CCP4BB web pages on how to do this).

These summaries are not complete, since many responses go directly to the person asking the question. While we understand the reasons for this, we would encourage people to share their knowledge on CCP4BB, and also would be happy to see summaries produced by the original questioner. While CCP4BB is obviously alive and well, we think there is still some way to go before the level of traffic becomes inconvenient.

Thanks to all the users who are now dutifully posting summaries. Also I would like to thank Eleanor Dodson for helpful discussions.

Data harvesting for CCP4

(February 2000)

How can I automatically generate a PDB header? I found the program "harvesting" in CCP4, it writes mmCIF files. Is there a way to produce directly a PDB deposition file?

From our EBI correspondent:

The latest version of CCP4 will output mmCIF files for several of the programs (MOSFLM/SCALA/TRUNCATE/MLPHARE/RESTRAIN/REFMAC). At the end of a refinement one can make a xx.tar.gz file of these files and submit these for deposition.

At this stage the atom coordinates are not output in mmCIF format and would have to be submitted as a separate file, within a set of files (see end of this note about deposition to the EBI).

It is not possible to convert an existing CCP4 logfile into harvest format. Harvest files are only generated by using the latest version of CCP4.

The semi-automatic tracking of data through CCP4 will become easier when the next version of MOSFLM is released - this will impose a responsibilty on the user to start labelling data sets and the labels will be included into the first MTZ file. Subsequent use of

these MTZ files will transfer labels to both 'harvest results files for each step' and to subsequent MTZ file headers. It is however the user's responsibilty to track each stage of a structure solution and carry out ultimate book-keeping. The harvest files from each stage can then be accumulated into a compressed tar file and submitted for simplified deposition. In addition - it hasn't yet been used in the real world and we don't know how it will mesh with different in-house practical use of different software; as it is common to say start with DENZO, use CCP4 then CNS then O then REFMAC - only experience will allow resolution of any practical difficulties.

The work done at ESRF/BNL and other synchrotron sites to add data labels to image headers will further smooth the flow of information from data collection to refinement by having MOSFLM and HKL2000 read the headers and transfer this to derived files. This is not yet in place.

One can also use CNS and use the mmcif_deposition macro which was in

.../cns_solve_0.9a/inputs/xtal_mmcif/deposit_mmcif.inp
this only does refinement:
.../cns_solve_0.9a/modules/xtal/exportmmcifrefine
.../cns_solve_0.9a/modules/xtal/exportmmcifstruct

there is also in pdb format (read by autodep):

.../cns_solve_0.9a/inputs/xtal_pdbsubmission/xtal_pdbsubmission.inp
.../cns_solve_0.9a/modules/xtal/pdbsubmission

The CNS macros (and the equivalent XPLOR PDB macros) deal only with information known for refinement.

Both CCP4 PDBSET and the CNS/XPLOR PDB macros can generate PDB SEQRES records for sequence, although there are problems in some cases with different conformations and the enthusiam that the macros add all, including water, to SEQRES.

There is also info to be written out from HKL2000 (DENZO/SCALEPACK) although this isn't currently the same as CCP4 (MOSFLM/SCALA).

As far as deposition - the EBI is still using PDB(BNL) legacy code in the form of autodep (autodep.ebi.ac.uk) as we in are the final stages of developing a completely new submission system that can handle these files. We have not as yet modified autodep to read mmCIF files - a function this software could never do.

However, until the new system is ready, if you want to use CCP4 or CNS mmCIF output in your deposition, then please do the following:

1. Start an autodep submission, upload the files and email pdbhelp@ebi.ac.uk to say you have done this.
2. We will convert the files to autodep and allow you to continue the submission.

How to deal with refinement parameters

REFMAC

Resolution in REFMAC

(February 2000)

Is there a simple possibility to improve the resolution limits in small steps from one refinement cycle to the next using REFMAC (something like the STIR instruction in SHELX)?

Such an approach should not be required in a maximum likelihood refinement package.

In least squares refinement, the high resolution terms are a handicap with a poor starting model becuase they vary much faster with small shifts to the model.

However, in ML refinement, the level of error in itself is a parameter which is determined and refined, and acts to weight down the high resolution terms in proportion to the poorness of the model. No resolution extension scheme is required, because the likelihood already provides a better scheme than any which could be determined by the user.

As far as I know no STIR-like command is available and I don't think it's necessary. Just go on in steps with the command files and that's just fine.

And is there a high resolution limit for the refinement with REFMAC or should one use other programs - like SHELX or CNS - for data better than let's say 1.1Å?

I don't see any reson why there should be a high resolution limit: REFMAC uses the full 5-Gaussian model for scattering factors.

I have refined a couple of atomic resolution structures with REFMAC (as I write this I am refining with 0.98Å data!). The biggest advantage of using REFMAC is that is runs *fast*. I also believe that at that resolution the maps with the maximum likelihood coefficients from REFMAC contain practically no bias to the model. So, it is very useful to do corrections in the initial stages of the refinement, try out double conformations and find quickly all/most of the waters in your structure in conjunction with ARP. Once that is done I usually apply the "finishing touches" (refining occupancies, etc) with SHELXL.

I've refined with REFMAC a 0.97Å structure and it works fine and FAST!!! You can calculate esu from inversion of LSQ matrix with SHELX afterwards.

Refining occupancies in REFMAC

(February 2000)

Is it possible to refine occupancies for alternate conformations using REFMAC and how does it work? Or is there another CCP4 program which can do this?

Refining occupancies directly, is not possible in REFMAC. The work-around, is to refine the B-values and judge/adapt the occupancies accordingly. There is a debate on whether

it is likely that the data can support occupancy refinement unless you have very high resolution:

- Even with data going out to 0.7Å there's a strong correlation between B-factor and occupancy, so I wouldn't refine occupancies and B-factors in the same refinement run without the use of something like similarity restraints for the B's and/or constraining the sum of the occupancies to unity (for small molecule structures this is exactly what I do); SHELXL should allow you to do this easily.
- Individual occupancy refinement for all atoms indeed does not make sense as a value of 1.00 is a very good estimate for the bulk of our models. However, you normally only want to refine occupancies for small groups of atoms for which you have reason to believe that they have non-zero occupancies. That is for sidechains built in multiple conformations and bound ligands. There remains the problem that B-value and occupancy are highly correlated in refinement, but if B-values increase to unreasonable values it may be better to fix the B-value to a reasonable value and refine the occupancy.
- I think it should be possible if one maintains a few reasonable constraints:
    a. the occupancies for all alternates should sum to 1.0
    b. the B factors for all alternates should be the same

    Perhaps you could allow manual override for the adventurous.

High resolution refinement

(September 2000)

I would like to pose one question concerning refinement of high-resolution structures. This topic has been discussed to some extent at the beginning of this year, but I would like to go for sure and have (an) additional question(s):

The situation:

I have refined a quite large structure (6 monomers of 50kDa each in the asu) at 1.7Å to quite reasonable R-values (Rwork: 16%, Rfree: 19%). Now I have measured a 1.295Å data set of the same crystal. The unit cell dimensions differ less than 1%. Because of the huge amount of data (close to 700.000 independent hkls) refmac5 and ARP/wARP with refmac4 appear to be THE refinement-programs to use (also in terms of tolerable time for refinement).

The questions:

Is it the reasonable way to start the refinement of the 1.295Å structure right from the beginning with all the data I have and let maximum likelihood find its way? Or is it better to use the "old-fashioned" way to extend the high resolution limit in small steps and search water in each step? Is the following refinement strategy reasonable (each step until Rfree converges)?:

- *refinement at 1.295Å without water starting from 1.7Å structure with refmac5*
- *adding water with ARP at 1.295Å*
- *refinement with anisotropic B-factors*
- *introduction of multiple conformations*
- *refinement of occupancies*

- *refinement with hydrogen contributions*

Taking into account resolution and the amount of data: Where should the R-values converge after proper refinement?

The answers:

- Various comments:
    1. I would always start with a bit of rigid body to accommodate the small changes in cell dimensions.
    2. Can't see any reason to throw away hard earned water positions - the wrong ones should disappear soon enough.
    3. Biggest challenge and most time is needed to find alternate conformations, and ARPing can make this harder.
    4. My order is different: Automatic refinement of complete model against all data (probably with hydrogens - can't see why not use something which is chemically sensible); rigid body first, then isotropic. Sort refined model on B-values, and look at highest 10-15% to see what has happened. You usually pin-point some residues which are in loops - maybe all copies have the same conformation, and can be rebuilt into an averaged map; others which are obviously in multiple conformations - usually with H2Os sitting where the 2nd conformer should be. Once you have made the obvious corrections then maybe start adding more waters, anisotropic parameters, etc.
       Protocols can differ to accommodate NCS.
- A few points:
    1. The structure is very likely substantially correct as is. Do some rigid body refinement to position it correctly in the new cell, then refine to convergence at the old resolution of 1.7Å with the new data. Include in this process an overall B-factor refinement, as the B's have likely gotten much smaller. Check maps for alternative conformations, clear new waters, etc., and rebuild as needed. I would use the 1.7Å water set as a starting point.
    2. Push resolution to 1.3 in 2 jumps... you can try aniso B's at around 1.5Å, but definitely at 1.3. Also add in the hydrogens. Your R's should drop substantially at this point (2-5%).
- I suggest to use all data straight away. Since your cell parameters slightly differ and the position of the molecule may be affected, the quickest would be either: Run molecular replacement or Run PDBSET/COORCONV to convert coordinates to fractionals and then back to orthogonals with the new cell parameters. In our experience it is quicker to start with addition of solvent sites at 1.3Å (refinement of both bulk solvent and overall anisotropy within REFMAC), followed by introduction of riding hydrogens, then anisotropic refinement and finishing up with multiple conformations and refinement or estimation of their occupancies. Very much depends on the Wilson plot B-factor and overall anisotropy. A typical value for R-factor would be 11-13 %.
- If you are using REFMAC5 you don't need to run COORDCONV. REFMAC5 itself will convert cell dimensions of coordinate file according to MTZ. My experience with higher resolution data refinement agrees with what others are saying. Use all data straight away. Especially in the cases where you have refined data to a little bit lower resolution there should not be a problem.
- A few small comments: I would put in multiple conformations before individual anisotropic Bs, as the latter can sometimes mask multiple conformations. Otherwise

the refinement protocol looks fine to me. I don't think there is any point in doing steps of higher and higher resolution. The maximum likelihood combined with some checking on the graphics should sort things out fine. The target R and Rfree are difficult to estimate - it depends a lot also on the quality of the data; it will be important to check the REFMAC output carefully for geometry statistics and to do validation for example by Whatcheck. Then you can vary the geometrical restraints so as to minimise Whatcheck complaints as well as Rfree...

Refining occupancies of alternate configurations

(July 2000)

I'm trying to refine alternate configurations of substrates in an active site. This was once possible in x-plor, so I'm using CNS. But the cns-bug-reporter told me that they haven't yet implimented the option to refine alternate configuration occupancies in CNS. As you can imagine, since it's my active site, my chemical argument sort of relies on it. So, I need to find a program that will do it.

I looked at the documentation for REFMAC, but don't see anything obvious in the docs or the examples that flags to refine occ's of alternate configurations. Can anyone tell me if REFMAC is the way to do this? Or is there some other program people would recommend that I didn't look at yet?

Additional info: The alternate configurations come from a reaction in the active site of the form: A + B -> C + D, where there is a mixture of products and unreacted reactants seen.

There are 3 basic answers:

1. Proceed with caution because OCC's and B's are absolutely coupled and what you choose for one will affect the other.

   This is true, of course. A method that someone suggested to exercise this caution is...

   Depending on your resolution, I would opt to only refine B's and adjust the occupancies manually.

   With this, you can make an initial guess at the relative OCC's based on the density, then refine B's. Presuming the B's for config_1 and config_2 are similar, you can manually adjust OCC's until the B's for the 2 configurations are similar to each other.

2. Use SHELX

   I have not tried SHELX, but this seems the best option for high resolution data.

3. Use CNS

   The authors have told me that they have not implimented the option to refine OCC's for alternate configurations. Many people insisted that you _can_ use CNS, but I think I believe the authors on this one. Someone suggested this manual adjust to the CNS limitation:

I used the alternate conformations feature in CNS to refine my substrate with two different puckers. Then used the q-group protocol to define the occupancy for each conformation. Then manually adjust the occupancy to be sure it adds up to one.

But, since my substrate is in the form

```
A + B -> C + D
```

and I know that if A is 80% occupied, then neither C nor D can be greater than 20%... I can say with firm certainty that this manual adjust is not accurate.

Correlation between B-factor and resolution

(January 2001)

Does any one know if there is any correlation between the overall B-factor of a structure in relation to its resolution? Are there any publications on this topic?
Also is there any correlation between the extent of disorder in a structure and the R-factor/Rfree?

To summarize, many of you believe that there is a (good) correlation between the overall B and the resolution cutoff. But then Gerard's statistics showed otherwise. Some of you attributed this observation to the correlation being masked by effects of experimental limitations.

These discussions are very helpful. However, it would be even more helpful if someone can cite (or tell me the lack of) literature references on this topic.

Ramachandran idealisation

(February 2000)

Quick question regarding a homology model I've built. One could argue this is redundant since it's only a predicted structure - but... Is there a program available which can introduce appropriate phi/psi angles to a given structure so that Ramachandran competent regions are not violated? In other words, can I get my hands on some software which can clean up a bad structure and make the Ramachandran plot look nice? I have quite a few unpleasant residues which would take some time to fix manually.

Here's the quick-and-dirty answer:

A nice set of web-based programs can be found on:
WHAT IF
These web-based programs can check, validate and repair your model... You probably can obtain the program for in house use too...

Then came:

I think the question should not be "how?" but "why on Earth?". Without any experimental data, the best any program can do is to pull residues into the nearest favourable area, but there is no guarantee that this will be the *correct* area (see our 'databases' paper in Acta

Cryst D, 1998, for an example of this -even with x-ray data- and a discussion). All this would amount to is (at best) cosmetic nonsense. Moreover, it could give a false impression of the quality of the model.

This was strengthened by:

The answer to your question below is found in a paper by Sali (guru of homology modeling) et al - Proteins, Str. Func. Genetics 23:318-326 (1995) - Evaluation of Comparative Protein Modeling... where it was shown that: "most ways of relaxing the template coordinates to improve the stereo chemistry of the model increase the rms differences from the correct target structure".

Then the following was added to the original question:

I feel I ought to explain my last e-mail regarding Ramachandran idealisation. As I mentioned, yes, it's a predicted structure, and so there is no way of knowing whether the correct angles have been introduced. However, why build a model at all if it violates standard geometry of proteins? That's like making a model of a house which doesn't have a front door or any windows (bad analogy - but you get the point!). Why not introduce beta phi/psi angles to a region which is predicted to be a beta-strand? Just wanted to get that off my chest.

This spawned a final reaction:

Your metaphor is beside the point. Standard geometry includes bond lengths and angles, planar groups, getting your chirality right and not letting two atoms occupy the same part of space - things we "know" and which we don't really need xtallographic or nmr data for to confirm. However, the conformational torsion angles in proteins (phi, psi, chi1, ...) contain the essential information about the structure of the protein - determining their values is what experimental structural biology is all about; predicting their values is what homology modelling is about. Since these torsion angles do not have a unimodal distribution (*i.e.*, more than one "ideal" value) you cannot impose "standard values". Moreover, the Ramachandran plot in particular is often used as a hallmark of structural quality in x-ray/nmr work, so "fudging" its appearance might convey a false message - which I'm sure you wouldn't want to happen. A better analogy would be the following: you build a model of *my* house! Now, it's a safe bet that my house has a front door and windows. But what you are trying to do is decide what colour my wall paper is, without any experimental observations of said wall paper ... Sure, you modelling my wall paper as "pink with elephants on it" makes your prediction look very precise and detailed - but it is in fact wholly inaccurate! (honest!)

Pseudo-symmetry

NCS analysis for 222 symmetry

(February 2000)

I am working on the structure of a tetramer with non-crystallographic 222 symmetry. It is not difficult to determine the three two-fold axes separately by superimposing dimers, but

a. *they are not necessarily perpendicular to each other and/or might not intersect in one point and*

b. *from the direction cosines of the axes you cannot conclude where exactly to put the two-folds with respect to the tetramer (i.e. where the origin of the 222 system is located).*

I would need this information e.g. to compare two of my structures by superimposing their 222 axes.
So, basically, my question is, does anybody know about a program which extracts the best 222 axes set from my coordinates?

Obviously, there is no easy answer to this problem. Nobody could point me to a program that reads in a pdb-file containing four monomers with non-crystallographic 222 symmetry and writes out three mutually perpendicular 2-folds which best describe the approximate symmetry of the tetramer in the pdb-file.

BUT, there are some 'approximations' which work as well, at least for the time being and the goal in mind.

1. Guogang Lu referred me to his website, which, besides some other useful stuff, hosts a program called FIT.
   FIT superimposes molecules (like lots of other programs do, too) AND writes the rotation axis out as a vector (described by two points with orthogonal coordinates) of arbitrary length, defined by the user. Then the whole problem is reduced to lsq-fitting the three two-folds (= six points) to another six points 'simulating' an ideal set of three two-folds with the same length as was used in FIT.
   In this context it was pointed out that one could include the origin as a seventh point for fitting if the point closest to the three axes could be determined. Which sounds reasonable to me but I haven't had the time yet to check whether such a routine is available from somewhere or whether one would have to write it from scratch.
2. I was referred to Kim Henrick who recommended and helped me with a no longer supported program in CCP4 - SYMFIT. Here it is possible to specify the complete symmetry you're looking for in terms of symmetry operators and then put in sets of 4 corresponding C-alpha atoms (*e.g.*) from the four subunits. Which are then fit/optimised by the program with respect to the specified symmetry operators. The calculated/optimised coordinates for the CA atoms can then be taken into FIT again which gives three beautifully perpendicular axes. Disadvantage is the limitation to 120 sites (the program was originally written for heavy atom sites), so I can only put in ca. 1/10 of my CA atoms. Maybe that problem can be solved by 're-dimensioning' the source code, though.
3. Fred Vellieux also provided me with a program for superposition of structures (SUPPOS). The operators obtained can be put into a subsequent refinement/optimisation based on a Bhat OMIT map (*e.g.*). I have to admit, though, that I have not pursued this strategy yet since I wanted to avoid working with maps - as long as possible.

Pseudo-symmetry R3(2)?

(May 2000)

I wonder if I could get some help on a problem that is driving us mad here. We have a 1.65Å resolution data set collected from a rhombohedral crystal (there was also a low resolution pass at 2.8Å). All data were processed and merged with DENZO/SCALEPACK. If we don't try too hard in the autoindexing we get the following cell:

```
rhombohedral setting:   56.646   56.646   56.646   92.678   92.678   92.678
hexagonal setting:      81.960   81.960   93.417   90.000   90.000  120.000
```

The data set then processes nicely as R32 with a completeness of 100% and an overall Rmerge of 4.4% (11.0% for outer shell). If we do it as R3 it gives virtually identical statistics. In R32 we get one molecule per AU with AMoRe. This refines reasonably well (Rfree around 24%) but a few bond distances persistently misbehave. If we do it as R3 with 2 molecules per AU, we can get the Rfree down to around 22%. However the operation relating one molecule to the other appears to be a near perfect crystallographic one. When the 2 independently refined molecules are superposed the differences are minimal and I would say not significant. I think the improvement in Rfree is simply due to allowing the refinement more freedom.

At this point we went back to the raw data and took a closer look. It turns out that there are alternate layers of strong and very weak reflections. If we drop the peak picking threshold in DENZO until it finds a significant number of these weaker reflections we then get the following cell:

```
rhombohedral setting:   78.313   78.313   78.313   63.129   63.129   63.129
hexagonal setting:      81.987   81.987  187.165   90.000   90.000  120.000
```

i.e. the c axis is twice as long in the hex setting. This also processes nicely as R32 with a completeness of 100% and an overall Rmerge of 5.0% (17.3% for outer shell). When we tried to run AMoRe on this it gave some very strange results - some problem with defining the Cheshire cell for non primitive space groups (although it does say in the manual that you can get problems with rhombohedral cells). We assumed the weak reflections mean that we have a repeating unit consisting of two of the "small" cells with the molecules in virtually the same orientation. So we generated a second copy of the molecule from our small R32 cell by applying half a unit cell translation along c. This model was then put into rigid body refinement in REFMAC, but Rfree was very high around 60% even though the packing looked fine.

If we look at the outputs from TRUNCATE (http://www.ccp4.ac.uk/newsletters/newsletter39/19_truncate_small_r32.log and http://www.ccp4.ac.uk/newsletters/newsletter39/19_truncate_big_r32.log) they are a bit strange, in particular for the big cell. Look at cumulative intensity distributions and moment2 values for acentrics. There is also a bit of a bump in the Wilson plot at about 2Å resolution.

As an added complication there is a very strong non-crystallographic 2-fold axis relating one half of the molecule to the other, so that it looks virtually identical if you invert it. This means you have to be very careful which way up your MR solution is.

We have tried submitting the data to the "Crystal twinning server" but the twin fraction is less than 2%.

So does anyone have any suggestions as to how we proceed? Obviously the easy way out is to ignore the weak data and go with the small cell - after all an Rfree of below 25% is definitely publishable!!

On the basis of several comments we tried a few things and I posted a follow-up message:

We are still struggling with this one, which is why we haven't posted a summary yet. We are now working in the big R3 cell with 4 molecules in the AU. AMoRe didn't work with a single molecule as the search model, but we were successful when we used the dimer from the small R3 cell - confused?

Anyway, the maps look quite nice, but the refinement is a bit disappointing: Rfac=25%, Rfree=28% (they were 19 and 22 in the small R3 cell). Also the overall FOM in REFMAC is 0.51 (it was 0.84). Could this be because the alternate layers of strong and weak reflections make it difficult to scale Fobs to Fcalc in REFMAC? The data have been processed with DENZO/SCALEPACK, maybe the situation could be improved using MOSFLM/SCALA. Unfortunately we can't get MOSFLM to correctly autoindex in the large cell as the spot picking routine struggles to find the weaker spots - it just gives us the small cell. Is there a way to convert a DENZO autoindexing solution into MOSFLM format so that we can proceed with this approach??

Here is a summary of the comments received:

There were several responses suggesting that our problem had many similarities with a case described in Carredano et al., Acta Cryst D56, 313-321 (2000).

Our problem is indeed very similar, but we don't see alternate layers of molecules with high and low B-factors. Also in this paper there weren't alternate layers of strong and weak reflections. They stated that the pseudo-equivalent reflections have the same amplitude within experimental error, thus Rmerge does not discriminate between R3 and R32 which agrees with our observations.

A similar problem was reported in P21 which was so close to P212121 that it could actually be solved in the latter.

The following (good) point was made by some: the MR didn't work well in the long cell because the stronger reflections will dominate the result whilst the weak ones will contribute very little.

Another point: there is a form of disorder giving well-defined spots at non-integral lattice positions which is discussed in Giacovazo's book. This does not, however, seem to apply to this R3(2) case.

Similar experiences in a number of systems including R3 are reported, and suggested that we may have a superlattice. This is under investigation.

The cumulative intensity distributions look fine. With every other layer weak you have many more weak reflections than you would theoretically expect for a structure with random atoms (which is where the theoretical plots originate from). Or in other words, having two molecules in almost the same orientation is indeed far from random!

A similar problem is reported where there were two slightly mis-oriented molecules in the big cell (perfectly oriented in the small cell). They eventually published in the small cell however. The conversion from small to large cell (presumably in R3) however is not simply adding another pair of molecules related to the first by a half cell translation along z. It was suggested taking the latter (m1 + m1 shifted in 0.5 z) and rotating it by 60 degrees, which should give nearly identical Rfactors back in the big cell. In fact, the molecule does not rotate, it is just the axis system rotated when moving to the big cell.

On a similar note: you need to reindex somehow - for R3(2) the requirement is that -h+k+l = 3n. When you double l and change all l1 to 2l2 this no longer holds unless you change the direction of the l axis.. *i.e.* you need to reindex as -k,-h,-2l. If -h1 + k1 +l1 = 3n, then -h1 + k1 +l1 -3l1 (=-h1+k1 -2l1) = 3n'. So reindexing as -k,-h,-2l gives +k -h -2l which is OK.

I'm having trouble getting my head round this but if you rotate the (correct) solution in big the cell onto that in the small cell using LSQKAB we get the following:

```
CROWTHER ALPHA BETA GAMMA     56.59279   179.79269   176.43974
SPHERICAL POLARS OMEGA PHI CHI     89.89634    30.07633   179.90738
DIRECTION COSINES OF ROTATION AXIS      0.86536     0.50115     0.00181
```

So there is not just a simple translation relating one cell to the other. I should also point out that the missetting angles from autoindexing in DENZO are also different.

There do seem to be many cases of such pseudo-symmetry with Fabs. The present case is not an Fab however, but a binding protein.

A program called DENZO2MOSFLM, written by Phil Evans, was recommended, which will do the transformation, but it should be noted that the overall Rfactor and Rfree will ALWAYS be much worse in the real cell SIMPLY because half of the data is systematically very weak. If you were to calculate R and Rfree for the l=even data (try it !!), you should find that they are very much better, if you do it for the l=odd data, you will find they are dreadful.

It was suggested converting .x files from Denzo into a format to be read into SCALA.

It was suggested that to get MOSFLM to deal with your larger cell, just let autoindexing do what it wants, then change the c-cell dimension in the menu top left. Predict after that and confirm that it is looking for the weak spots. (However, some of the observations mentioned above suggest that this wouldn't work for R3 although I haven't tried it). Also, if the two pairs of molecules are perfectly aligned in the model, the structure factor calculations will give exact zeros for the intermediate layers. You will have to use the rigid body refinement in REFMAC to break out of this. Incidentally, when you TRUNCATE the data in the large cell, check the N(z) plot, and make sure that the observed curves are to the left of the theoretical ones. This is an indication of the data being weaker than the cell symmetry suggests, but that is exactly what you have: an accidental alignment in the cell which renders the intermediate layers weak.

Felix Vajdos had a similar situation (see Vajdos et al. Protein Science, 6: 2297-2307 (1997)) that he never quite satisfactorily refined. They had a small (P43) cell which turned out to be a sublattice of a larger P41 cell (P43 is a subgroup of P41 and vice versa provided the c-axis is increased by a factor of 3). We raised some important observations:

1. Translational pseudo-symmetry is particulary insidious because the weak reflections arise due to "breaks" in the crystallographic symmetry. So these reflections, which are important for correctly modelling the structure, are also among the most poorly measured and therefore difficult to refine against.
2. The presence of systematically weaker reflections results in a very non-normal distribution of structure factor amplitudes, which means that it becomes much more difficult to interpret the R-value. The presence of weaker reflections has the effect of systematically decreasing the denominator in the R-value, thus raising its value.
3. They found that the correlation coefficient proved to be more reliable indicator of the progress of refinement. Thus for a structure (1.58Å), we saw correlations of around 0.9 ('free' around 0.87) even though the Rfactor was 39.6% and Rfree was 46.1%!

He suggested if you can get the information you want from refinement in the pseudo-lattice, then do so, and in the paper simply mention the difficulties encountered in the refinement in the true-lattice.

Another comment on the reindexing:
Of course Rfactors are always higher for weak reflections; if you run old RSTATS you can get the Rfactors as a function of |F| as well as resolution. Or split the file into two - hk 2l, and hk 2l+1 using MTZUTILS:
RZONE 0 0 1 2 0 gives l=2n and RZONE 0 0 1 2 1 gives l+2n+1 and run RSTATS on the two subsets.. I suspect you have really good Rfactors for the stronger data and it is OK in fact (you were presumably careful to make sure the pseudo-R32 equivalents are both either Free or non_free..).

Yes we adopted a similar procedure to that used in Carredano et al., Acta Cryst D56, 313-321 (2000). Basically apply Rfree to data set processed in R32 and then expand to R3 using SFTOOLS. Then we CADed these Rfree flags onto our data set processed in R3. Whereas in the paper they just used the expanded R32 dataset.

Another similar problem: P21 crystal, with a 2 fold NCS axis almost parallel to b. The maps looked very good (these were also high resolution data), but the Rfactor got stuck just below 30% (Rfree around 32%). Later on a native data set was collected, which happened to crystallize in the smaller cell (no NCS) and the previous model refined easily to about 15% without doing anything to the protein chain. I always had the strong feeling that it was the weak "in between" reflections which were to blame for the high Rfactor in the case of the former data set.

It was pointed out that what we describe is not pseudo-symmetry but a superlattice. Refining such a thing is known to be a pain in the ass even in small molecule crystallography (I agree!!!!). The fact that your R and Rfree go up when refining in the larger cell is absolutely normal since you are adding a whole load of relatively weak (but perfectly valid) reflections. Remember that R-factors are unweighted statistics. Therefore the correct description for your structure is when using the larger cell which results in higher R and Rfree. It is definitely not a scaling problem. A superlattice means that the internal symmetry in the smaller cell is broken, but that it still holds approximately. There are, however, some small differences introduced and all information relating to these differences is present in the layers with weak spots (so, DON'T use sigma cut-offs to get lower R-factors! It will just hide the correct structure). Using only the small cell will provide you with an average structure. In fact you should not try to compare R/Rfree between the large and small cells since they are not calculated using the same sets of data. The fact that your density in the large unit cell is really clear means that what you are doing is probably correct despite the higher Rfree.

As a result of several suggestions we split the R3 big cell data set into the strong and weak components using MTZUTILS and carried out the following analysis:

	Relative <I>	<I>/<sigI> out. shell(1.65A)	Rcryst (Rfree)(%)
All l	50	6.2	25.5 (28.1)
l = 2n	100	9.3	21.0 (24.0)
l = 2n+1	1	0.5	50.8 (52.1)

So the weak reflections are REALLY weak!

It's good to see that we are not the only ones with this kind of problem and that there is no clear-cut solution - other than to crystallize in another space group! I think we will use the small R3 cell after all.

NCS averaging - translational NCS

(October 2000)

What is the easiest (or best) way to get a NCS-matrix out of a phased map? I got below average MAD-phased map, and I know 2 Se positions. There are 2 monomers per ASU. I tried FINDNCS (I also have 4 very very weak Pt site positons), and the result seems not very convincing. From the non-averaged DM map, I can see the solvent boundary, and I can barely see the two molecules. I need to get the correct NCS-mask, or correct NCS-matrix. When I put the NCS-matrix I got from FINDNCS in DM, it didn't give me anything better than I got as a non-averaged map. I have also played with MAPROT, MAPMASK, NCSMASK, tried to get a NCS-mask but without success.

Addendum:
Before this MAD (2.2Å) data set, we had a 2.4Å native dataset and I have tried MOLREP with a poor MR search model. The self-rotation function did not show a proper two-fold. After many trials in AMORE and MOLREP, all the solutions suggest that two molecules have the same rotation but with a translation shift by half in Z (and less than 1/10 in y). This is in P21. But the MR-solutions are not refinable, so when I collected the MAD data, I tried GETAX after phasing by SHARP, and it didn't work due to lack of proper 2-fold. So that is why I want to find a better way to do NCS-averaging. I will try to use the MR-solution to get a NCS-mask and doube-check if the MR-solution is at least in the right place compared with the MAD map.

This is a brief summary: As it turns out, it is a translation NCS which is not as useful as rotational NCS. We also obtained two crystals from Hg soaking which produces a unit cell half the size of the original. Now I am trying to refine the MR solution using this new dataset. Maybe I should get the Maximum Likelihood MR program from Randy Read for this rather poor MR model.

- Several people suggested using GETAX.
- Try automatic NCS in DM or use any protein with similar size to get a mask, and use O to find the matrix graphically.
- Use bones from the map to get the mask.
- Here is http://www.ccp4.ac.uk/newsletters/newsletter39/19_ncsscript.html using AMORE to find out the NCS-matrix.
- Use FFFEAR.
- Use SHARP and SOLOMON or SQUASH to get the best non-averaged map and look for the secondary structure element in the map, and then figure out the NCS-matrix.
- Use EPMR to get the MR-solution.
- Look at the self-rotaion function and use the solvent flattened map as input to POLARRFN.

PS: one last question, just a little doubt: will that translation NCS prevent me finding the Se sites? I don't think so, but we should have 9 sites per protein instead of one. No matter how hard I tried, I can only find one Se per protein (so two sites per ASU), the two Se sites are related by (0, 0.04,0.5).

Addendum to summary posting: This is bad news; translational symmetry might make it harder - more complicated Pattersons etc., but if you have only found one Se there is something seriously wrong - possibly no Se in the crystal; possibly not enough signal. How did you position the Se? If the MR-solution is correct, phases based on the solution should be able to show you the Se sites; I usually do a Dano Fourier, but a dispersive difference Fourier should show you the same sites.

More translational NCS

(October 2000)

I have a pseudo translational NCS which relates the 4 mols in the asymmetric unit by ~(0.5, 0, 0) and ~(0.25, 0.25, 0). The 4 mols differ slightly on the domain angle as a result of lattice packing. The resolution is 2.4Å, and the space group is C2. MAD maps etc. were not that great, but I managed to build the model manually, and have refined a couple of cycles so far. The 2fo-fc and fo-fc maps have showed many new and nice features, which is encouraging, but R/Rf is about 41/45.

Before going further with the traditional procedure, I think I should understand a few things:

- *Instead of selecting R-free set randomly, do I need to select it specially to avoid any "correlation" of reflections? Some papers suggest to use thin shell selection in the presence of NCS. Is it mainly for rotational NCS? But I have alternative layers of strong (25%), weak (50%), and very weak (25%) data. To make it simpler to think, if I had just (0.5, 0, 0) translation, can I consider (2n, k, l) and (2n+1, k, l) as "correlated"? On the other hand, this translation can also be treated as a pseudo 2-fold rotation if the crystallographic symmetry is considered, and this pseudo 2-fold is parallel with the crystallographic 2-fold. So, is it ok to use thin shells then? Or for every (2n, k, l) reflection with free flag, make the corresponding (2n+1, k, l) also free? What if I have a combination of rotational and translational NCS, or a translation that cannot be turned into a rotation?*
- *What are people's general experiences in the effects of the test set selection on the outcomes of the refinement?*
- *At 2.4Å, is it now possible to do automatic rebuilding with wARP/REFMAC? How about if many reflections are weak?*
- *When a large fraction of reflections is weak across all resolution ranges, would the MAD map quality be largely reduced because the weak Friedel pairs may not be accurately measured?*

Here is the most comprehensive answer to the computational/theoretical questions:

Indeed, thin shell selection is only relevant for rotational NCS. The alternate layers of reflections are not correlated at all (at least not more so than normal neighbouring reflections). Their main difference is that for the even reflections the diffraction of the molecules related by the (0.5, 0, 0) translation add up nearly in phase and for the odd ones they nearly cancel out.

Since the NCS 2-fold is nearly parallel to the crystallographic 2-fold, the "NCS" correlations introduced in reciprocal space are between reflections related by 2-fold crystallography or nearly so. Since we only refine against the "unique data", these 2-fold related reflections will not be in the unique asymmetric unit and will therefore not be in your working set of

reflections. So selecting your test set is unfortunately not your biggest problem. Just use a random selection.

NCS correlations lead to a slight model bias in the test set. The extent of the effect depends on the level of NCS. E.g. I wouldn't worry about 2 or perhaps even 3-fold NCS, above that it may become significant. The general feeling is that even with NCS correlations, the R-free will still indicate if your refinement is heading in the right direction. However, Sigmaa estimation is also based on your Rfree set. The NCS-induced bias will suggest that the model is better than it really is leading to improper weighting. So in theory, yes, you may have to worry about high NCS. The problem is that selecting reflections in thin shells isn't a great solution either. To get a reasonable number of shells they have to be very thin. As a result a reflection in a thin shell often does not have its NCS-related reflections in the same shell, defeating the purpose. Selecting small NCS-related volumes for Rfree may be better and I have started implementing that in SFTOOLS but never got to properly testing it, so it is not in the CCP4 version of the program. In short, I don't think there is an ideal answer. The positive side is, the higher the NCS, the less the risk of overfitting (but remember that in your case of translational NCS this whole issue is not relevant).

With regards to the effect of weak reflections on the map: well it is not going to help you. However, the contribution of a reflection to the map is proportional to the reflection amplitude. So the quality of the map will be dominated by the 50% of strong reflections.

Then came a number of suggestions on working around the problem by going back to the lab bench, for which the summary is:

Others suggested to tweak crystals into a smaller cell by:

- using cryoprotectant:

"... regularize into crystallographic symmetry when cryoprotectant was soaked into the mother liquor and the crystals were frozen",

"adding 5% glycerol (I was using 20% as cryoprotectant) to my crystallization trials not only gave a four times smaller cell with all translation-only NCS removed - it also increased resolution from about 2.5 to 1.5Å.
The same reduction in unit cell occured when going from 0 to 20% cryoprotectant using the original crystallization conditions. But the crystals did show some cracks and were not as good."

- using heavy atoms:

"We had a situation where heavy atom soaking changed to smaller cell with no loss in the diffraction quality and resolution.
Similar was the experience with cocrystallizatin with heavy metal compounds in at least one case I read. In that case, there was even a great increase in the resolution (Ref: Ronning et al. (2000) Nat. Struct. Biol. Vol. 7, 141-146.
Another useful reference regarding pseudo-translation problem is Chook et al. (1998). Acta Cryst. D54, 822-827."

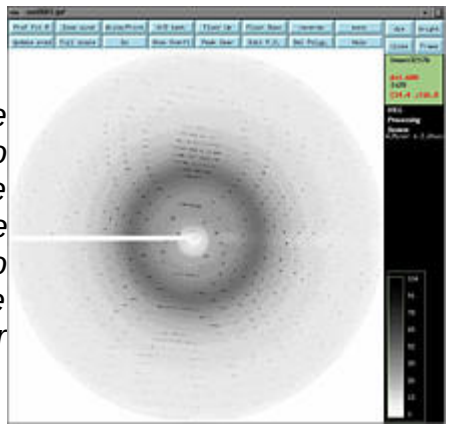
- or reducing humidity

As a super bonus, the new crystal form often diffracted to much higher resolution. It was suggested to use correlation coefficent rather than R-factor and to study into the contributions of the weak reflections on phasing.

Indexing problems

Indexing images with straight streaks

(April 2000)



We have collected some data on station F1 at CHESS. The crystals appeared to be hexagonal thin plates and diffracted to about 2.2A. We collected 75 degrees of data for each of the three crystals. The spot shapes looked OK but we could not index the spots using DPS, DENZO or MOSFILM. The spots seemed to align along straight lines on all images and one could see the hexagonal features. Has anyone come across this before? Your help is greatly appreciated.

The suggestions/conclusions are as follows:

1. Always make sure these parameters are correct: Direct beam position, Wavelength, Crystal to detector distance.
2. Some people had a look at the images. They concluded: "the images look much more like a precession photograph than an oscillation photograph (*i.e.*, there are no clear lunes), which indicates either an extremely high mosaic spread (in excess of five degrees) or the presence of multiple crystals whose orientations are related in such a way that you get an apparent single lattice. Some of the spots are also clearly split."
3. Processing as P1 has been suggested.

Scalepack failure

(May 2000)

*I got hexagonal crystals of a protein. My previous data show the space group is P6122. Recently I put one of my best crystals for data collection. The frame looks good (see picture; click to get better resolution). And I use DENZO program to index the data. All previous steps go on smoothly, until I scale the data. The scale result is very bad (see http://www.ccp4.ac.uk/newsletters/newsletter39/19_yuanpingHkl.txt). The linear R factor is about 0.516 and square R factor is 0.922 and the chi**2 is obviously wrong. But I notice that the chi**2 of each frame during process is near 1 which is OK. Could anyone help me to figure out the possible cause?*

Suggestions:

1. It might be wrong index. Reindex!
2. Try lower symmetry P6.
3. Check the orientations of each frame.
4. Try P3, P3(1)22, P3(2)22, P322.
5. Check whether the spindle axis is wrong.

I tried these suggestions recently. It should be wrong index! But I checked each parameter of index carefully, no one is wrong. I tried the lower symmetry P6 as well as other symmetry suggested. No improvement. In fact, I never doubtle the symmetry of P6122 myself. I tried some smaller crystals as well as heavy atom derivatives before. This symmetry gives successful result. If it is wrong symmetry, all my previous data would have the same problem. The problem seems to be solved after the beam is alined and the detector is pulled back to 200mm. As some of you noticed, c is high, about 196Å, from my previous index result. The error will be disastrous with even a little beam line deviation. Another cause might be short distance (150 mm) diffraction with a very big cystal which give too much crowded and strong spots, although I can not see overlap of spots.

Strange diffraction pattern

(June 2000)

We have now observed for the second time within a few months a strange diffraction behaviour of two totally unrelated proteins which crystallize in space group R32. Whereas for the first protein (sex hormone-binding globulin) we did get nice diffracting R32 crystals with one steroid bound, we now keep getting these misordered crystals with a different steroid. For the second protein, we only get these misordered crystals.

The major characteristics of these images are, that planes with nicely shaped spots do alter with planes which have smeared reflections. Basically planes with l=3n (hexagonal setting) do alter with planes (l=3n+1, l=3n+2) with smeared reflections.

I thought about reticular merohedral twinning, but then I would expect twice the number of reflections on the planes (l=3n+1, l=3n+2) but still nice spots. Maybe not ??

One other characteristics is that the smeared spots on these planes do not fall on the R32 lattice but are slightly displaced. If you draw a line on the 1° oscillation images between the nice spots, the smeared spots do not fit onto the line (see image 3). I therefore tried to index with a P3 lattice instead of the R3 lattice. In that case the reflections are fitted better but of course there is an enormously high number of predicted spot positions with no spots to be seen.

What intrigues us is that we observed this now for two different projects. Probably we will just have to look for better crystallisation conditions, however it would help us if we could understand the problem.

Several replies pointed me to the following papers which deal with statistical layer disorder.

1. Howells and Perutz (1954) the structure of heamoglobin. V. Imidazole-methaemoglobin: a further check of the signs. Proc. Royal Soc. London Ser. A, 255, 308-314.
2. Luo, Laver and Air (1992) Unusual diffraction of type B influenza virus neuramidase crystals. Acta Cryst. A48, 742-744.
3. Bragg and Howells (1954) Acta Cryst 7, p409.
4. Cochran and Howells (1954) Acta Cryst 7, p412.

It took me a while to organise these papers and indeed it seems that such statistical shifts (in both directions of layers) might be at the origin of the smearing of distinct reflection layers while other reflection layers remain sharp. Still intriguing to me is the fact that the

spots in our case don't fit exactly onto the lattice anymore. The same has however also been observed in case of neuramidase (Luo et al). Further suggestions dealt with crystal handling. We actually also mounted crystals in capillaries, however still the same smearing. There seems to be some light on the horizon now by increasing the salt concentration during the crystallisation...

Reindexing tables

(March 2001)

I'm currently looking for a table that lists all possible indexing relationships between two different data sets of the same crystal form if the true space group symmetry is lower than the lattice symmetry (i.e. true space group P3, lattice point group 3barm). I don't need this only for my special case (where I think I've got all possibilities), but I believe this should be of general interest to all crystallographers who have to get consistent data sets from the same crystal form (i.e. all searches by trying different soaking conditions). Of course, the first thing I did was to look into the International Tables A,B,C, but surprisingly, I didn't find such a table (or I have eggs on my eyes). Do you know about such a table and could tell me and the CCP4BB the reference?

Thanks a lot! I've received several pointers to tables with possible reindexing relationships. Many of them were lying directly in front of me! Thanks to all of you!

Here are the pointers:

- $CHTML/reindexing.html
- XDS indexing routine lists reindexing possibilities
- the HKL manual deals with them in its scalepack scenarios
- it's in the special Acta D issue on data collection and processing, Dauter (1999), Acta Cryst. D55, 1703-1717

Patterson for P3121

(August 2000)

I am working on MAD data set at 2.0Å resolution. When I run Patterson function using fft, I do get peaks for heavy atom. Since my space group is P3121, I have confusion how to interpret the map. So how to get the coordinates for these atoms to use in MLPHARE? Is there any software available to interpret (except RSPS) this Patterson and give coordinates of heavy atom, so that I can use it for difference Fourier or in MLPHARE? Can anyone give some suggestion how to work on this space group.

It was a hand problem so I have to change the space group to P3221. I am giving the summary of the responses which could be helpful to others.

- As far as the space group goes, there are no special tricks, all Pattersons obey the same rules; peaks at vectors between atoms. The Harkers follow from the symmetry positions:

(-Y,X-Y,Z+1/3) - (X,Y,Z)	->	(-Y-X,X-2Y,1/3)
(Y-X,-X,Z+2/3) - (X,Y,Z)	->	(Y-2X,-X-Y,2/3)
(Y,X,-Z) - (X,Y,Z)	->	(Y-X,X-Y,-2Z)

and because of the centre of symmetry there will also be a vector at (2X-Y,X+Y,1/3)

- So you can look at the Harker section at z/w=1/3 and see if you can find peaks which satisfy the required algebra, and scan for peaks u,-u,w and then solve for Y-X.
- If the algebra gets too heavy (and I think it does in P3121!) you can use SHELX Patterson search (it is set up as part of CCP4i) but ALWAYS first look at the peak list and Harker sections of your Patterson; if there are no clear peaks then you won't be able to find a solution.
- But remember that you can get a set of solutions (x,y,z) in P3121 or the related set (-x,-y,-sz) in P3221 so when you test the other hand you need to also change the symmetry. This is all set up very nicely in CCP4i.
- I guess the easiest thing is to throw everything into SOLVE, and most likely the program will come up with the right solution. Unfortunately, this will take away all the fun with solving the Patterson by hand. Whatever you choose to do, the correct choice of spacegroup is critical, and I'm not sure what SOLVE does to handle this problem. If you find your sites by hand, and then refine them with MLPHARE, you will need to pay attention to the signs of the anomalous accupancies. There are four possible solutions, corresponding to a left- and a right-handed configuration in each of the two spacegroups. You will need to select for strong and positive anomalous occupancies (there should also be strong and negative, weak positive, and weak negative solutions in the two spacegroups). Throughout, you can initially choose one of the two spacegroups, because even if you picked the wrong one, you will at least get the high anomalous occupancy right (whether negative or positive), if your heavy atom positions are correct.
- You have at least three choices for automatically finding your heavy-atom sites:

  http://cns.csb.yale.edu/
  Go to the tutorial section and look for Phasing by Multiple Anomalous Diffraction (MAD)
  Heavy atom search
  http://www.solve.lanl.gov/
  http://www.hwi.buffalo.edu/SnB/

- If the Patterson has such clear peaks, just run it through SHELX - it will probably do it for you in a minute (literally). Or else SnB, which is very powerful and also very easy to use, although it doesn't use the Patterson as such.

  Web pages are (in case they're not installed already):

  http://shelx.uni-ac.gwdg.de/SHELX/index.html
  http://www.hwi.buffalo.edu/SnB/

  Or you could try the CCP4 program VECSUM, but it's unsupported, apparently. Haven't used it myself, so wouldn't know how effective it is. RANTAN in CCP4 could also do the job.

- As with any Patterson map you can start with the symmetry opperators. These should be pair-wise substracted from each other. Yielding 6 times 6 combinations of (u,v,w)'s (which are combinations of x,y,z). It will become obvious that there are certain Harker sections. The two obvious ones in this space group are z=1/3 and z=2/3. But all others will also be pretty nicely arranged on planes (I don't remember it correctly but these are something like u,v,w = x,2x,any z; anyway these are of less importance). Now you check the positions of the peaks in the Harker plane z=1/3. These have a certain (u,v,1/3); now go back to the equations you have calculated by subtracting all the symmetry operators which gave the Harker section and start calculating. Remember that the (u,v,1/3) can also be (-u,-v,1/3) or (1-u,1-v,1/3). So it gives you a lot of equations that will give you certain x,y for the heavy atom site. Now try and see if these sites give all the other peaks. If so you have got yourself a solution. For more than one heavy atom this becomes pretty complicated, but if you have one heavy atom bound it is do-able and very illustrative.
- Also several people suggested the program SOLVE. It worked nicely with one single solution, which I refined using MLPHARE for both hands.

Data Sharpening

(April 2000)

*I am interested in "sharpening" my reflection data by applying an artificial temperature factor (e.g. -70 Å**2). This seems non-trivial, as I am not sure what do with the sigmas. Before you all think I'm nuts, others have done this in low resolution structures with favorable results, such as: Borhani, et al. (Apolipoprotein A-I) PNAS 94:12291-12296. Stehle, et al. (Simian virus 40) Structure 4:165-182. Unfortunately, these papers were not so informative as to HOW they actually did this. Does anyone have an idea of a program to use that will do this sort of modification?*

David Borhani wrote with advice based on his experience with Apolipoprotein AI. They wrote a modified version of TRUNCATE that would apply the artificial B-factor to the data, but this was never incorporated into the general release of CCP4. He had good suggestions on low resolution refinement as well.

Rod Mackinnon (potassium channel structure) used Xplor to calculate an anisotropic B-factor array, applied this to the F's and converted his Sigmas manually by using Sigmanew=(Fnew/Fold)sigmaold. This is equivalent to directly applying the scalar calculated from exp(-Bsin**2theta/lambda**2) to the F's and Sigmas.

It was suggested to use Ecalc, but I didn't want to normalize the F's at the same time.

Another suggestion was to use the BIOMOL suite of programs, which we don't have installed here.

I subsequently figured out that you can use SFTOOLS to apply functions (real or complex) to various columns of data, but didn't get around to using it yet because:

Eleanor Dodson saves the day with a modified version of CAD that will apply an overall B and Scale factor to the data. This version is now available publicly: see the SCALE keyword in the documentation for CAD if interested.

Various

Molecular Replacement with NMR models

(January 2000)

I am doing some general studies concerning using NMR models in MR. If you happen to know of any such cases, published and unpublished, can you give me some references?

1999:
JMB 292:763
Cell 97:791
Acta Cryst. D55:25
JMB 288:403
JMB 286:1533
NSB 6:72
1998:
Nature 395:244
Biochemistry 37:15277
Biochem. J. 333:183
Structure 6:911
Structure 6:147
Acta Cryst. D54:86
1997:
NSB 4:64
1996:
FEBS Lett. 399:166
Acta Cryst. D52:469
Acta Cryst. D52:973
1995:
PNAS 92:10172
EMBO J. 14:4676
JMB 247:360
1994:
Structure 2:1241
NSB 1:311
1993 and before:
Cell 68:1145
PNAS 88:502
JMB 206:669
Science 235:1049

and finally, an unsuccessful attempt: Structure 5:1219

Websites and courses for Synchrotron Users

(January 2000)

A central resource for macromolecular crystallographers seeking information on synchrotron beamlines in the USA is available at:

http://biosync.sdsc.edu

This website, which has been developed on behalf of BioSync, the Structural Biology Synchrotron Users Organization, combines and organises information about beamlines at five different synchrotrons in the United States. It provides technical information about

beamlines in a standardised format which is easy to compare. The site also functions as a portal for investigators planning visits to synchrotron facilities. Researchers can obtain schedules, contact, logistical and training information, and applications. The information is contributed and maintained by representatives from each of the synchrotrons.

(February 2000)

http://www.x12c.nsls.bnl.gov/rr_course_2k/ - Learn about crystal cryogenics, MAD data-collection techniques, and use of modern phasing software from the experts.

http://www.x12c.nsls.bnl.gov/rr_course/.

Homology model into poor MIR map

(January 2000)

I have a poor MIR map and a homology model. The molecular replacement failed to give a clear translation solution. Does anyone know some real space search programs which may help fitting the model into the Fouirier density for map interpretation?

Kevin Cowtan's 'fffear' (Cowtan K. D., Acta Cryst. D54, 750-756) runs much faster than ESSENS (Kleywegt G. J., Jones T. A., Acta Cryst. D53, 179-185), by which it was inspired. It is much easier to use, and fits with the rest of CCP4.

Genetic Algorithms for Molecular Replacement

(February 2000)

Here is the summary for my question about Genetic/Evolutionary Algorithms applied to Molecular Replacement:

1. EPMR - easy to download, install and use. Acta D55:484-491. Contact author at crk@agouron.com. Mr. Kissinger was very accessible and answer my questions on time.
2. No other options (really, I did not receive responses from Mr. M. Lewis of Acta D53,279-289).

Disulphide reduction in protein structures

(February 2000)

Can buried (i.e. small or no solvent accessible surface area) disulphide bonds be reduced to cysteines in general? I have a structure in which 2 cysteines are in the right position to form a S-S bond but the electron density shows otherwise. My protein was purified in reducing conditions. However, from other evidences (structure homology), these should form S-S. Does anyone know of any published structure that can shed light on my puzzle?

Reduced disulphide bonds can be an artefact of data collection. This happens due to radiation damage at high energy synchrotron sources, even at 100K, and was described in Weik et al., (2000) PNAS 97, 623-628.

Did you measure the distance between the sulphurs? It should be around 2.05Å for a disulphide bond and more than 3Å for a nonbonded contact. If it truly is reduced, look at the CB-SG-SG-CB torsion angle, this likes to be near +/- 90 degrees. If disulphide bond formation would have to indroduce large distortions, then that may stabilise a reduced state. Are both cysteines conserved among related sequences, do they sometimes have only one cysteine or do they always have either two or none? The first and last cases would suggest a special interaction, whereas the middle one suggests that the cysteine is "just a hydrophobic buried residue". No guarantees either way though.

Questions on MOSFLM

(April 2000)

A plea from Harry:

Can I just repeat something that I've posted before on this BB? If you have problems with MOSFLM (especially with crashes and hangups), the people to contact in the first instance are Andrew and/or me; except when we're both out of town, we have the fixes to hand! Oh - BTW, _before_ asking anyone, it's worthwhile having a look at URL www.mrc-lmb.cam.ac.uk/harry/mosflm, which has fixes for numerous problems and hints for installation. It's not comprehensive, but I do have answers to most questions there.

Domain movements

(May 2000)

I would have liked to characterise a domain movement in 2 closely related proteins with DYNDOM.
Following the scripts provided and fiddling with the parameters I always end up with the following message:

```
WARNING NUMBER OF RESIDUES IN CHAINS IS DIFFERENT 303 300

determining backbone atoms of first conformer

determining backbone atoms of second conformer

number of residues used for analysis:  300

rmsd of whole protein best fit:    6.969 A

number of clusters:   1

number of clusters:   2
found cluster for which all domains are less than minimum domain size
so we stop

NO DYNAMIC DOMAINS FOUND
TRY ALTERING PARAMETER VALUES
```

It was indicated that the proteins should have:

- the same No. of residues
- maybe not contain modified residues (*e.g.* phosphoTyrosines).

Editing the pdb-files in this respect gave results only for very small domain sizes ("domain" keyword). Those however did not reflect the actual domain movement but rather some flexible residues at the N-terminus. Cutting the N-terminus from the pdb-files gave more sensible results.

Guoguang Lu advertized his DOMOV server bioinfo1.mbfys.lu.se/cgi-bin/Domov/domov.cgi, which was very convenient to use.

Validation of protein crystal structures at high resolution

(June 2000)

I have refined a high resolution crystal structure using various program packages (CNS, REFMAC, SHELX). For the comparison and validation of the models I would like to get some statistics such as rmsd of chiral volume, bond length, bond angles, torsion angles, contacts between non bonded atoms etc. (the REFMAC output includes this information). I am not searching for a program which produces plots (PROCHECK) or individual rmsd. I would like to have overall values. Is there a program able to do this and is such an analysis reasonable to compare the models?

A warning first: this analysis only makes sense if the used restraints for all programs would be comparable. Example: XPLOR/CNS defines chiral volumes as improper dihydrals.

Annotation: CNS has no implementation to perform anisotropic temperature factor refinement. Jiffies have been written by R. Steiner to calculate characteristic parameters from SHELXL outputs.

The easiest way to compare results is take all the models and run each of them through REFMAC for 1 cycle only - the initial statistics will give you a direct comparison. Whether or not any of these statistics tell you that one model is better or worse than another is of course another matter!

Some useful web sites are

- http://pdb.rutgers.edu/validate/
- http://biotech.embl-heidelberg.de:8400/
  or
  http://www.cmbi.kun.nl/swift/whatcheck/ (the program is free for academic users) (better to have also DSSP to utilize all options !)

Reducing the size of the Rfree set

(July 2000)

I refined a structure against data to 2.3Å, where the Rfree set included 2600 reflections (5% of data). I now have data to a much higher resolution, and putting 5% of data in the Rfree means 11000 reflections, which is quite a lot. I would like to use a smaller fraction for the Rfree set (say 1%). For this, I need only to keep 1 every 5 reflections of the "low resolution" free set. Any hints on how I can do this?

It was suggested to get a new Rfree set with the desired size, and to remove bias with annealing or throuh model coordinates randomisation (PDBSET). Also, to use

ARP_WARP, either to remove bias in the new Rfree set, or to build the structure from scratch.

As a direct answer to the question, this is a way to reduce Rfree set size, using SFTOOLS: If you do want to keep 1 in 5 of your old test data, you can do it fairly easily in SFTOOLS. I'll assume the working data have a 1 in column RFREE and the test data have a 0. I'll also assume you have a column called F_Old which contains the original amplitudes. Then do the following in SFTOOLS:

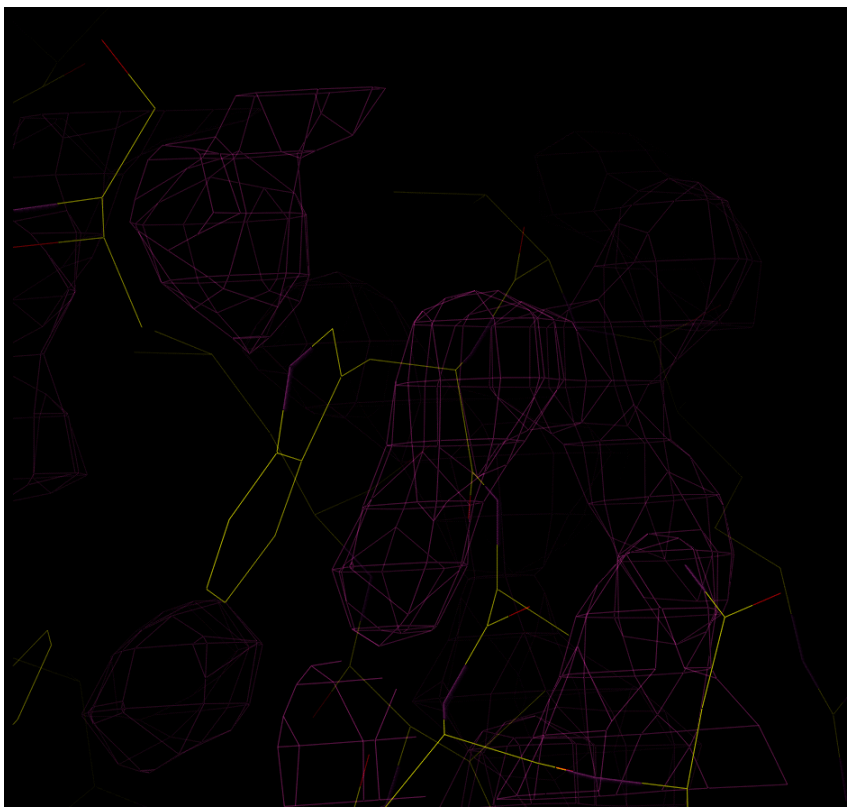
```
read my.mtz
select col rfree = 0          ! only old test data
calc col rfree = rfree(0.2)   ! keeps 1 in 5 of old test data
select all
select col F_old absent       ! only new data
calc col rfree = rfree(0.01)  ! flag 1% of new data
select all
write my_new.mtz
```

Map displacement

(July 2000)

I'm attempting to generate electron density maps using ccp4_v4.0.1. The programs used all report normal termination.I can display these maps correctly when using the O fast map commands however when I use fm_rsr_group I get a segmentation error resulting in a coredump. I run these maps through es_mappage to generate an O style map, however when I display these using the map_file command etc the maps are displaced away from the model. I've also had problems generating xtalview maps.

Here is an image of the problem I had.



My solution was to use SFTOOLS to import the ccp4 map and then ouput the map in ...dn6 format. This has worked.

Other suggestions included checking the indexing of the data, confirming that the unit cell parameters are identical in the pdb and mtz, using FFT.big to generate the maps, and using mapmask to put the map around the the molecule.

Calculating rmsd's for loops

(September 2000)

I'm trying to calculate rmsd's for loops of interest in different proteins and have tried a few different methods. I found that...

1. *CNS calculated rmsd for each residue, but both structures have to be described by the same .mtf file, so I can't evaluate same protein from different species easily.*
2. *DALI gives a single overal rmsd for the whole protein, but I am interested in certain loops.*
3. *TOPP breaks the rmsd's down into 2nd structure, calculating rmsd's for each alpha-helix and beta-strand, but not for loops.*

So, I did what I think has tricked TOPP into doing what I want.... almost....

I made an artificial set of HELIX cards listing all my loop residues as helices. This way, I bypass TOPP's eveluation of secondary structure and it just does the rmsd's on what I've already 'evaluated' as helices. Problem is when a loop contains only 2 residues or when the rms is larger than 2.0. In those cases, the rms is not calculated for that section and I get errors that look like this....

helix: 11 have rms 4.117280 rejected helix 12 not more than 2 atoms, rejected helix 13 rms is 2.6 more than 2.0 rejected

I've tried changing these:

```
RESIDUE 1
DISTANCE 10.0
```
But it doesn't change the errors at all.

Summary of the suggestions with my comments from trying them...

1. Edit .pdb's to only have the loops of interest, send to DALI:

```
      Argh!  I have 9 complexes of dimers with a TIM barrel
             and 8 loops of interest in each active site!
             That's 144 edited .pdb files pairwise to DALI.  ;-)
      Also: if I send edited loops to DALI, the overall superposition
             of the protein is not possible and rmsd's are isolated
           numbers with little meaning.
```

7. LSQMAN: http://xray.bmc.uu.se/usf/welcome2usf.html:

```
      This is a nice program, but with 144 loops to compare, an
     interactive program can't be left to run overnight.
```

9. CCP4's LSQKAB or COMPAR:

```
      These seem convenient, but I didn't
```

```
11.      try them, because I found another one I liked before I got
       to this suggestion.... sorry!
```

12. ALIGN (Cohen, NIH; J. Appl. Cryst, 1997, 30:1160-1161) or HOMOLOGY (Rossmann's):

```
13.     I tried to find these to download from the web, but the names
14.     are not unique enough ...too many hits... so I gave up after
      half hour or so.
```

15. ProFit:

```
16.                    This one I liked the best!!!
17.     -It's very flexible!
18.     -It's script-run... so I wrote the script once and just
19.      changed the filenames to compare different protein pairs.
20.     -You can specify exactly which residues to use for the
21.      LSq-superposition, and exactly which ones to calculate
22.      the rmsd, and they can be different.
23.     -You can specify 'C-alpha' or 'all atom' rmsd, so in the CA
       mode, it overlooks mutations (not like CNS).
```

Note from the ProFit author: There is no official publication on ProFit, apart from the above URL. There will be a new version of ProFit (v2.0) in the very near future.

Calculating interaction surface area

(September 2000)

Can someone give me some hints on how to calculate the interaction surface area between two protein molecules?

- Use CCP4 program AREAIMOL to calculate the accessible surface area for each of the individual proteins and then calculate the surface of the complex. The buried surface area is S(1) + S(2) - S(1-2) and the interaction surface area is half of this value.
- Create the surface for each monomer in GRASP (it gives you the area for the monomer in the little textbox) and then create the surface over the dimer (which gives you the area for the dimer)
  The difference is the area (A) covered: A(interact per molecule) = (A(monomer)+A(monomer)-A(dimer))/2 gives you the area of interaction per monomer (or molecule).
- The way you do your calculation is correct. You do not specify if you calculated the accessible surface area or the molecular surface area, which are two different things. Although the results for most cases are very similar. To divide by two or not, is your choice. Just make clear which way you do it. See Jones and Thornton (1996) PNAS 93:13--20, Lo Conte et al. (1999) J. Mol. Biol 285:2177--2198 and Stanfield and Wilson (1995) Curr. Opinion Struct. Biol 5:103--113, for a few different ways of analysing protein-protein interfaces. Jones and Thornton use the half-values, Lo Conte et al. the total area and Stanfield and Wilson calculate molecular surfaces instead of solvent accessible surfaces.
  You can also calculate the interface area directly in GRASP by following menus Calculate - Area of a Surface/Molecule - Molecule - Excluded area. The program will then list the area of the subunits in complex and free and the difference of the two. You can then use this information to project the data to the surface of a molecule. See manual for more info.

- Try the protein-protein interaction server at UCL:

  http://www.biochem.ucl.ac.uk/bsm/PP/server/

- You can use the CCP4 module AREAIMOL in DIFFMODE COMPARE. Create two individual XYZIN (pdb) files, one with all components and a second with one or more of the components removed. Output files will be generated showing regions with altered surface accessibility for the common coordinates. An aside -- this program was useful to us because it can also give quantitative information about crystal packing interactions. See the documentation for all details: $CHTML/areaimol.html. Also, look at Newsletter 38 - article on surface areas.

Rejecting reflections after processing

(January 2001)

I wanted to exclude a few reflections from my data-file using the REJECT flag in SCALEPACK2MTZ. However, the reflections are kept in the output file. What can I do?

Here a summary of useful hints to the REJECT problem in SCALEPACK2MTZ.

1. This seems to be indeed a bug: The source code has been fixed and is included in the recent new CCP4 release (4.1.1.).
2. There is no other CCP4 program to exclude selected reflections after processing (for some good reasons!).
3. Use SFTOOLS with the following input:

```
4.      SELECT index h = 1
5.      SELECT index k = 10
6.      SELECT index l = 10
7.      SELECT INVERT
8.      PURGE
      YES
```

Using the following awk-script then gives the expected result which can easily be included into an input command file for SFTOOLS:

```
awk '$7=="30.0000" {printf"SELECT index h = %3s\nSELECT index k =
%3s\nSELECT index l= %3s\nSELECT INVERT\nPURGE\nYES\n",$1,$2,$3}' fft.log
```

Various databases

Structure/Sequence Database

(February 2000)

I was having a quick look round RCSB and PDBSum to find a site where I can run a primary structure through a blast/fasta/etc to find sequence homologs of known structure. The old Brookhaven site had one - I am sure there's one still out there somewhere. Could someone please send me the URL.

http://www.ncbi.nlm.nih.gov/blast/psiblast.cgi

http://www.toulouse.inra.fr/prodom.html - something similar but a little less direct

http:/xray.bmc.uu.se/sbnet/prosal.html, where you can get through to most of the following:

- RCSB (with the extensive search facility)
- SAS (also at UCL, like PDBSUM; runs BLAST against PDB and annotates the alignment by a variety of structural properties)
- EBI (FASTA against PDB)
- NCBI (BLAST against PDB)

EBI, then "other search services" and "list of PDB structures in FASTA format".

Compare active sites

(February 2000)

I am looking for a program to compare active sites of enzymes that bind the same ligand but do not have the same fold. The program should propose alignments of the two substructures with amino acids of similar chemical properties in equivalent positions.

Try

- PROCAT - a database of 3D enzyme active site templates
- SPASM by Gerard Kleywegt - to find motifs of main and/or side chains
- ProFit, part of Andrew Martin's software - protein least squares fitting program

Met as Zn ligand? Metal-site databases

(May 2000)

Is there any example in which a protein binds Zn++ via methionine SD? or, Is there any way to search the PDB with unix commands like

*grep LINK *.pdb |grep MET |grep ZN*

if you don't have a local copy of the PDB? I've tried searching the HAD at http://www.bmm.icnet.uk/had/, but it doesn't seem to be fully functional yet.

Here are some URLs:

- http://metallo.scripps.edu/index.html
- http://relibase.ebi.ac.uk/
- http://www.imb-jena.de/ImgLibPDB/pages/siteDir/IMAGE_SITE.shtml

Of these the Jena database was the easiest to use, just enter Zn and Met in the residues textbox and submit. It found no example, but many hits with Cu + Met or Zn + Cys, verifying the search procedure.

The Scripps database found 1 trivial example in which the Zn is bound to mainchain atoms of Met: 1b0n.

Miriam Hirshberg wrote a script to search their local PDB for occurrence of ZN and MET together in LINK records. There were no hits found (1b0n has no LINK records!).

Herbert Nar referred me to Azurin structures in which the metal has been switched. Native has Cu, with a 3.0Å bond to MET SD. When replaced with Zn, Met 121 is no longer coordinated to the Zn (this from abstract of Nar et al. 1992, Eur. J. Biochem. 205, 1123).

So it looks as if Met doesn't bind Zn, which seems reasonable enough. Zn is in the same column of the periodic chart as Hg, which Petsko says doesn't react with methionines. All the zincs I've seen so far bind to cys or his (or in 1b0n to the mainchain N and O of a methionine).

I have this strong anomalous peak at the Zn edge in a sample soaked with ZnCl2, and it's next to a highly conserved MET (so I don't think it's a sequencing error). The distance is a little too long for a S-Zn bond (around 2.0Å in SOD) though, so maybe there's another atom between.

How to get CCP4 going

A lot is written on the bulletin board about the compilation of CCP4 on the various available platforms. Because there are so many details and intricacies, it would be best to check the CCP4 Problems Pages and the bulletin board archives for this (see the CCP4BB web pages on how to do this).

CCP4/Max-INF Workshop on Refinement and Validation of Macromolecular Structure

Wednesday 3rd January to Tuesday 9th January, University of York

Organisers: Garib Murshudov and Eleanor Dodson (York)

Background

Short courses have proved to be a valuable way to teach crystallographic techniques. This workshop aimed to provide a sound training in refinement techniques as part of the avowed CCP4 and EU funded MAX-INF initiatives to promote good practice. Such workshops also give valuable feed-back for method developers, and helps them test their programs against a wide variety of data sets and models.

Format of the course

The course was divided into morning lectures and afternoon tutorials where students were introduced to the software, and encouraged to apply it to their own data. (The full timetable is available at http://www.ysbl.york.ac.uk/~ccp4/workprog2001.html).

The University of York was able to provide two computer classrooms each able to accommodate twenty students, so with some apprehension we accepted 40 participants, who were divided into two tutorial groups. This meant each tutor had to repeat the class twice, and that they had less time with individual students, but it did allow members of more laboratories to have access to this very intensive program.

Course material can be accessed from the above URLs listed in the program.

Details of the course

The tutors were :

Paul Adams	CCI Lawrence Berkeley Laboratory
Ralf Grosse-Kunstleve	CCI Lawrence Berkeley Laboratory
Gerard Bricogne	Global Phasing
Pietro Roversi	Global Phasing
Eric Blanc	Global Phasing
Richard Morris	Global Phasing
Garib Murshudov	University of York & CCP4, CLRC
Eleanor Dodson	University of York
Liz Potterton	University of York
Martyn Winn	CCP4, CLRC
Victor Lamzin	EMBL Hamburg
Anastassis Perrakis	NCI Amsterdam

Randy Read	Cambridge
George Sheldrick	University of Gottingen
Dale Tronrud	Eugene, Oregon

The software packages used were: **CNS**, **BUSTER**, **REFMAC** and associated **CCP4 software**, **ARP/wARP**, **SHELX**, and **TNT**.

The standard of the lectures was excellent; as a community we are very privileged that the developers are willing to devote so much time and effort to teaching. In particular Paul Adams and Dale Tronrud both came from the West Coast of USA, and endured jet lag and British weather for the dubious satisfaction of working 12 hour days, for the reward of free institutional meals. George Sheldrick presented the fundamentals of refinement as well as providing excellent tutorials in the use of SHELX. It was very helpful to have such a clear exposition of this material early in the course and all other lecturers benefited from and built on his presentations. It was pointed out that he was one of the few lecturers who is a full-time academic and although this must limit his time for research it is a great boon for his teaching style.

As well as the listed lectures there were several valuable discussions on special topics. One of these was scaling the calculated and observed structure factors when the model is incomplete (always true to some extent). How best to parameterise solvent, and the unmodelled parts of the unit cell is still a matter for research and it was valuable for the developers to exchange experience and I hope interesting for the students to listen to the discussion and realise that there is often not a "correct" answer to these problems.

Another centred on how to describe and deposit the geometric and stereochemical restraints used during refinement. Traditionally many of these criteria were used as the basis of validation, but obviously this is not appropriate when they are also restrained. Kim Henrick from the European Bioinformatics Centre described the way they plan to both store the target values and report on individual structures.

As usual the end-of-course party was a great occasion, as illustrated by some candid camera shots! Pietro and Eric are welcome ANY TIME they wish to come to York; they kept up a stream of clean crockery and cutlery from our overcrowded kitchen.

Problems of the course

The students were too polite to complain about the overcrowding, and I did not hand out a questionnaire to get such feedback. The classrooms were overcrowded, and the computing resources stretched to their limit. Probably conditions would have been more comfortable with fewer students, but choosing between people from different Universities was invidious and some people made late but compelling cases for inclusion. In the end we felt it better to have 40 people 80% satisfied rather than 20 people 90% satisfied.

An additional problem with this course was travel to and from York. The course was planned before the disastrous floods which washed away part of the York-London railway track. However everyone arrived in the end, despite our anxieties.

Acknowledgements

Garib and Eleanor are enormously grateful to all those who spoke and demonstrated, without them there would have been no course. Kevin Cowtan with the CCP4 staff, in particular Liz Potterton and Alun Ashton, worked extremely hard to guarantee that the network functioned, and that the software was properly installed and accessible, a formidable task. The computing staff at York were very helpful and made sure things went smoother than they should have done. The students worked extremely hard and were very tolerant of defects in the organisation. No student/post-doc had to pay a penny to come thanks to CCP4 and the EU initiative.