

Daresbury Laboratory



JOINT CCP4 AND ESF-EACBM NEWSLETTER ON PROTEIN CRYSTALLOGRAPHY

An informal Newsletter associated with the SERC Collaborative Computational Project No.4 on Protein Crystallography and the ESF Network of the European Association of the Crystallography of Biological Macromolecules.

Number 29

November 1993

Contents

CCP4 version 2.2

D Love

Some thoughts on scaling

I Tickle

Chicken egg-white lysozyme grown in microgravity and on earth for a comparison of their respective perfection

S. Weisgerber & J R Helliwell

Some thoughts about processing, accessing and analyzing PDB data

J P M Postma

Biomacromolecular speleology

G J. Kleywegt & T. Alwyn Jones

DL POLY: A macromolecular simulation package

T Forester & W Smith

Measurements and corrections of spatial distortions in imaging plate systems

S O Svensson, A.P. Hammersley, A. Thompson, A Gonzalez and T. Ursby

The peptide-binding protein OppA from *Salmonella typhimurium*

J Tame, G Murshudov, E Dodson and A. Wilkinson

MAD data collection and phase determination of OppA

I. Glover, R. Denny, N. Donatus Nguti, S. McSweeney, A. Thompson, E. Dodson, T. Wilkinson & J. Tame

CCP13: Software development for fibre diffraction

Standard stereochemical dictionaries for protein structure refinement and model building

J.P. Priestle

A data format for the ESRF

P.J. Daly

Network news

D. Love

Diary of future meetings

Editors: Susan Bailey

Keith S. Wilson

SERC Daresbury Laboratory
Daresbury, Warrington WA4 4AD UK

EMBL c/o DESY
Notkestrasse 85, D-2000 Hamburg 52
Federal Republic of Germany

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

CCP4 version 2.2

Dave Love, Daresbury

A new release of the suite is now available after a rather long gestation period—we hope it was worth the wait. A version of the release notes is given here. Thanks to the people who have contributed, especially the brave souls who attended the recent ‘documentation workshop’ in an effort to improve things. *Please read the results and let us know what’s still wrong/unclear.* There really is supposed to be decent documentation for everything—I was amazed how long it took to find out that many .doc files had been left off the ftp area previously!

Summary of significant changes to CCP4 at version 2.2

Binary file transparency

The low-level i/o routines can now read MTZ and map files written on architectures with different number formats *providing they are written with this version of the library.* There is some overhead in this, but no significant overhead reading files written on the same architecture. Note that not all the possible combinations have been tested...

Categorisation of programs

There is a new ‘unsupported’ category of programs. (This does not imply that we guarantee support for the others.) They are supplied as is and we will not necessarily fix them (but still welcome bug reports or your fixes). Many obsolete programs have been thrown out completely. You can install only the major programs and not the unsupported category if you want. There is a new directory: unsupported.

Documentation

- There is a much-expanded, but still incomplete new manual. This includes material formerly in the ‘overview’ man ‘page’.
- new directory man/man3 for library man pages (incomplete)
- There is now a document for every program (we hope) and many have been much revised and improved. Complete formatted copies are supplied, along with man whatis files and the .doc files can be used with the unix man system. Most of the documents have been examined and revised.

Examples

The examples/unix/runnable directory has scripts which will run with the small dataset from examples/toxd (which also contains working scripts). The ctest directory has been removed.

VMS library support

the VMS assembler routines have been dropped—we now only provide the C version of the low-level library routines since we can't maintain the assembler code.

You need a C compiler for the VMS compilation now. See the manual for how to get the free GCC compiler if necessary (although this hasn't been tried for building the suite).

The C routines are slower than the old assembler ones—sorry. If this proves to be a particular problem, let us know.

Changes/fixes to configure and makefiles and VMS installation

The VMS installation is now simpler and cleaner but is not guaranteed in OpenVMS yet since we have had problems with an apparent run time library bug.

Library bug fixes

- MTZLIB: IPRINT=4 didn't print symmetry any more; user label input and output variables reset in lropen and lwopen
- some timing routines were wrong
- the setting of CCP4.OPEN is now obeyed for MTZ and map files
- BINSORT alignment problem on some architectures
- possible inability to redefine logical names in VMS

General library updates

- MODLIB: routines removed: match, ma21ad, mc10ad
- MTZLIB: more batches allowed; interpret absence flags from the Groningen BIOMOL system (just set to zero at present)
- SYMLIB: routines removed: turnip, turn3, decsym, syminv, iminv, symat; added: patsgp; changes to symop.lib; more batches allowed; new library data file symop_all.lib has all Int. Tab. A SGs
- FFTLIB: added routines factrz, fndsmf, setlim, setgrd
- GRAFLIB: no longer included by default. routines now in PLOTSUBS removed
- PLOTSUBS: replaced with routines removed from graflib
- removed modules: graphics, hlplib, mdflib, optfflib, plot9x, lib_kraulis, frodo-maplib, mthlib, hdrlib
- LCFLIB now won't write LCF files
- ccperr now tries to report the last system error and times in VMS
- PARSER: new routines RDRESL, GTTREA, GTTINT; the (mis)feature of allowing arithmetic between values read by the parser has been removed
- MAPLIB: improved precision of RMS and mean calculation
- \$CINCL/(default,enviro).def: tidying and commenting
- protin.dic: updated and an alternative provided (VLdict.idl) from Victor Lamzin
- code for binary file transparency (as above)
- Lots of internal reorganisation

Significant program bug fixes

- ABSURD: problems with 'reindex'
- AGROVATA: xloggraph o/p wrong; default INTENSITIES is PROFILE; changes if SCALE column present
- ALMN: divide-by-0 errors
- FFT: error for spacegroups P65 (and 61 ...); get Patterson SGs correct
- FHSCAL: several corrections
- HKLPLOT: spurious 'Disagreement between symmetry operators'; error in multiplicity
- MLPHARE: fix divide-by-zero etc.
- MTZ2VARIOUS: don't output excluded reflexions, treat formats as similarly as possible; output [FPH-FP] if FPH assigned; END not compulsory; output always to HKLOUT
- OVERLAPMAP: fix problems of rounding errors in calc. of correlation coeff.
- POLARRFN: failed to calculate rms value; put weighted RMS and mean into map header; correct sampling intervals in map header
- PROLSQ: can no longer attempt to access hkl(in,out)(LCF files)
- REINDEX: new version
- ROTAPREP: takes DENZO input and madnes for050 files
- ROTAVATA: confusion about FTWO/NOFTWO rejections between ROTAVATA and AGROVATA; change default on number of batches
- SFALL: various small changes
- VECREF: made consistent with FHSCAL (errors in them cancelled out)
- WILSON: xloggraph plot

Significant program updates

- ABSCALE: new REINDEX option; changes for mosflm summed partials
- ABSURD: storage increase
- AGROVATA: accepts SCALE column etc.; redimensioned; ROGUES file extended; changes for mosflm summed partials; negative values of SDADD
- ALMN: peak search changed, restored weighting of RMS and mean
- CAD: fairly major re-write
- COMPLETE: changes for triclinic data etc.
- COORDCONV: was coord_conv
- ECALC
- EXTEND: new algorithm; MAPSYMMETRY option
- F2MTZ: CTYP obligatory; FORMAT option

- FFT, FTTBIG: EXCLUDE option; changes to default grid
- LCF2MTZ: grok multi-record LCF files
- MLPHARE: various
- MTZDUMP: now as previous dumpreso
- MTZUTILS: wasn't applying resolution cutoffs to output file
- OVERLAPMAP: CHAIN option; change residue packing
- PDBSET: various extra options
- PEAKMAX: keyworded and various other changes
- PROLSQ: new LIMIT keyword; various other changes
- ROTAVATA: default INTENSITIES is PROFILE; changes for mosfilm summed partials
- SFALL: various changes
- SURFACE
- TRUNCATE: changes to interpolation etc.
- WATPEAK: includes contact list; carries through occupancy, *B*-factor info.

[Changes to things in the unsupported directory not given here, although many have had some changes.]

New programs

- ACT: analyse coordinates.
- AMORE: monolithic molecular replacement program.
- GENSYM: symmetry-related sites from atoms and list of vectors within defined volume
- NPO: new version of PLUTO
- PROCHECK system: stereochemistry checking
- RSPS: Patterson searching (previously aggregated)
- UNDUPL: essential for some data from Madnes (following Absurd).
- VOLUME: polyhedral volume around selected atoms

Programs deleted or moved to the aggregated directories (with some suggestions for alternatives)

acsa, area [use areaimol], atconv, atomsf, atpeak [use watpeak], bavsc2, brk2dia, brkmat, brksort [use procheck], burgess, cadpro, chelp, chkplt, cli_help, convmnp, coordswap, crossecc, dabs, deg, diffop, dumpreso [use mtzdump], hlbgen, hlptree, limits [use pdbset or extend], mapbrick [frodo], mapcont [frodo], mapform [use mapexchange], minlim [use extend/pdbset], nadia [use angles/procheck], o2brk [O distribution (replaced by pdbset)], oscgen [with mosfilm], patvec [use vectors], pdbrt [with ribbon], pltout [with ribbon], postplot [with ribbon], peaksearch [use peakmax], resection, reslicer, ribbon [aggregated], ribrot [with ribbon], rwbred, skewbrk, splitd [ribbon], symgen [use pdbset], tidytext, transz1, xplor2pdb [use pdbset], xyzlim, xyz_modify [use coordconv]

Housekeeping

(All?) the distributed files have been examined and only relevant ones should now be distributed. All the programs should now print version (date) information when they run.

Binaries

will normally not be supplied on VMS tapes now. (We can't make AXP binaries anyway.)

Known or potential problems

This is a partial list of known problems with the CCP4 release 2.2 code. They are either problematic to fix or have not been fixed yet for lack of time.

- PLTDEV: scaling wrong for HP output and reported to throw blank pages and hang the plotter; doesn't produce EPSF PostScript.
- PLUTO has many potential problems, including bad control flow (into loops) in routine circo – NPO is the replacement
- In VMS programs report the time used by the session rather than the program when they finish.
- The manual is still in rather draft form.
- Documentation for the libraries is still inadequate in some cases—consult the library source or a program doing something similar if necessary.
- Some man pages assume fixed-pitch fonts and don't format properly with troff.
- AMORE uses enormous amounts of memory and probably won't work on many systems (including VAXen?).
- Some of the programs may use too much memory to link on small VMS systems—no workaround known.
- On AXP-OpenVMS (Alphas) only, there have been problems with the diskio library due to seeks going wrong. The kluge for this may or may not entirely solve the problem. If not, it does appear to be DEC's problem.
- It's not clear whether or not you can successfully use VMS C compilers other than DEC's—specifically GCC—to compile the library code. If you succeed, please let CCP4 know.
- HPUX: there has been a report that the compiler options in the configuration for HPUX 9 don't work, although they do in a supposedly identical system I've tried. Can't change without breaking where it works. Feedback wanted!
- OSF/1 1.3 may or may not have `fsplit` shipped with it. If not, the libraries won't build. There is a free `fsplit` available. Feedback wanted!

Some thoughts on Scaling.

Ian Tickle, Dept. of Crystallography, Birkbeck College, London.

Scaling of structure factor amplitudes is superficially a very simple arithmetic operation, yet in practice it proves to be far from trivial to implement a reliable algorithm. The major difficulty is that the quantities we are attempting to scale have errors whose distribution functions we can only estimate.

I was prompted to write this article partly through something I read in an article (Vonnrhein, 1993) in the last newsletter about calculation of difference Pattersons with X-PLOR, and also partly through some lateral thinking on the part of one of my colleagues at Birkbeck (unfortunately as this took place some time ago, I forget exactly who it was).

The observation in the above-mentioned article that drew my attention was that the *Fobs/Fcalc* scaling option in X-PLOR (Brünger, 1992) when applied as a F_{PH}/F_P isomorphous derivative scale factor is not as accurate, as judged from the appearance of minor site cross-vectors, as that obtained using programs specifically designed for the purpose, such as CMBISO (PHASES suite) and FHSCAL (CCP4).

Contrariwise, my colleague had shown me some results using the FHSCAL program to scale *Fcalc* to *Fobs* in preparation for a difference Fourier. The purpose of FHSCAL is to calculate and apply a F_{PH}/F_P derivative scale factor according to a formula derived from that of Kraut *et al.* (1962); for full details of the derivation see my contribution to the *CCP4 Study Weekend Proceedings* on "Isomorphous Replacement and Anomalous Scattering" (Tickle, 1991).

My initial reaction to the results from FHSCAL was that the program was never designed to perform this operation, and it was therefore not a sensible thing to do (the column label assignment feature of the standard CCP4 reflection file handling routines is sometimes a little too flexible in this respect!). After some discussion and further thought however, I realised that my colleague had not been quite so dumb after all.

It is crucial however when doing this that one assigns the FP program label to the $|F_o|$ data column, SIGFP to $\sigma(|F_o|)$ (provided the σ 's are reliable), and FPH to $|F_c|$; SIGFPH is not relevant in this situation and should be left unassigned. Different (and wrong) results will be obtained if the assignments are made the other way round, for reasons that will become clear.

The parallel between the two situations can be seen in the equations:

$$\begin{aligned}\sum_h (k|F_{PH}|)^2 &= \sum_h |F_P + F_H|^2 = \sum_h |F_P|^2 + \sum_h |F_H|^2 \\ \text{and } \sum_h (k|F_c|)^2 &= \sum_h |F_o + \epsilon|^2 = \sum_h |F_o|^2 + \sum_h |\epsilon|^2\end{aligned}$$

In the first case the assumptions are that F_P and F_H are uncorrelated, so that the cross-term vanishes, and that F_H accurately describes the change on soaking in the heavy atoms, *i.e.* the derivative is perfectly isomorphous. In the second case it is similarly assumed that the true structure factor, whose experimentally measured amplitude is $|F_o|$, and the structure factor ϵ arising from errors in the model are uncorrelated. Actually I've cheated a little here: F_c and ϵ should really be on the same scale, but ϵ is going to be eliminated anyway so it doesn't

matter.

The equations that eliminate the unknowns $\sum_h |F_H|^2$ and $\sum_h |\epsilon|^2$ in the respective situations are also almost analogous:

$$\sum_h |F_H|^2 = \sum_h m((k|F_{PH}| - |F_P|)^2 - (k^2\sigma^2(|F_{PH}|) + \sigma^2(|F_P|)))$$

$$\text{and } \sum_h |\epsilon|^2 = \sum_h m((k|F_C| - |F_O|)^2 - \sigma^2(|F_O|))$$

where $m = 1$ for centric, 2 for acentric. Note that unlike the previous equations, these are approximations (albeit rather good ones) based on the assumption that F_H and ϵ represent relatively small perturbations. This is true even for the centric reflections; in fact, contrary to popular belief, the approximation is actually worse for the centrics.

So solving for the Kraut scale factor k in the *Fobs/Fcalc* case:

$$k = \sum_h ((m+1)|F_O|^2 - m\sigma^2(|F_O|)) /$$

$$(\sum_h m|F_O||F_C| + ((\sum_h m|F_O||F_C|)^2 - \sum_h ((m+1)|F_O|^2 - m\sigma^2(|F_O|)) \sum_h (m-1)|F_C|^2)^{1/2})$$

In practice the scale factors are calculated in shells, smoothed and interpolated, so each reflection may have a different scale factor.

All of the above formulae involving the *Fobs/Fcalc* scale factor are strictly correct only for normally distributed coordinate errors; they do not deal correctly with other errors such as missing atoms, and therefore may not be suitable in the initial stages of refinement. The fundamental difference between the two types of error is that whereas ϵ is in theory uncorrelated with F_O but partially correlated with F_C , the structure factor F_A calculated from atoms absent from the model is clearly uncorrelated with F_C (i.e. the structure factor for the atoms present in the model) but partially correlated with F_O . So including the "absent atom" structure factor (and again cheating with the scale factor!):

$$\sum_h |kF_C + F_A|^2 = \sum_h |F_O + \epsilon|^2$$

$$\text{or } \sum_h (k|F_C|)^2 = \sum_h |F_O|^2 + \sum_h |\epsilon|^2 - \sum_h |F_A|^2$$

The equation used above to eliminate ϵ now becomes:

$$\sum_h |\epsilon|^2 + \sum_h |F_A|^2 = \sum_h m((k|F_C| - |F_O|)^2 - \sigma^2(|F_O|))$$

but generally $|F_A|$ will be unknown, so now it is impossible without additional information on $|F_A|$ to eliminate both the unknowns $\sum_h |\epsilon|^2$ and $\sum_h |F_A|^2$.

Read (1986) noted that the effects of coordinate errors and missing atoms are statistically equivalent and can be rolled up with the *Fobs/Fcalc* scale factor into one parameter: σ_A .

In practice atoms, such as solvent, are usually missing because their thermal parameters are so high that they do not make an appreciable contribution in the resolution range of the $|F_O|$ used (typically for refinement a low resolution cutoff at around 10-8Å is applied). Obviously if one were scaling for an "omit map", the scaling must be based on the structure factor calculation done before any deliberate omissions.

With the assumption then that we can forget about missing atoms, it is apparent that as all the sums of squares are positive, the correctly scaled $\sum_h |F_c|^2$ must always be greater than $\sum_h |F_o|^2$, just as the scaled $\sum_h |F_{PH}|^2$ must always be greater than $\sum_h |F_p|^2$. This is why the column assignments must be the right way round!

I decided to write a little program (Fortran source available on request) to simulate the errors and test various formulae for the *Fobs/Fcalc* scale factor that are in use in various structure factor and refinement programs, assuming Wilson's centric and acentric distribution functions for the true structure factor amplitudes, and making some simplifying assumptions about the error distributions of $|F_o|$ and F_c .

Actually I assumed that $|F_o|^2$ has a normal error distribution with a standard deviation based on Poissonian counting statistics, whereas F_c also has a normal error distribution, but bivariate in the complex plane for acentrics and univariate for centrics, with a fixed standard deviation. The errors were generated using a Gaussian random number generator (Numerical Recipes), and their expectations were set so that the r.m.s. error in $|F_o|^2$ was 6%, and the standard crystallographic *R*-factor between $|F_o|$ and $|F_c|$ for the complete model was 30%. 10% of the reflections were chosen at random to have a centric distribution, the remainder being acentric. I ran two simulations, one for a 100% complete model, and the second with 90% partiality, or fractional scattering power, defined as: $\sum_h |F_c|^2 / (\sum_h |F_c|^2 + \sum_h |F_a|^2)$.

Scale factors and % deviation from true scale.

Partiality	=	100%	90%
$k_1 = \sum_h F_o F_c / \sum_h F_c ^2$	=	0.866 (-13.4%)	0.883 (-11.7%)
$k_2 = \sum_h F_o / \sum_h F_c $	=	0.907 (-9.3%)	0.948 (-5.2%)
$k_3 = (\sum_h F_o ^2 / \sum_h F_c ^2)^{1/2}$	=	0.908 (-9.2%)	0.948 (-5.2%)
$k_4 = \sum_h F_o ^2 / \sum_h F_o F_c $	=	0.951 (-4.9%)	1.019 (+1.9%)
$k_5 =$ Kraut scale factor	=	0.993 (-0.7%)	1.097 (+9.7%)
True scale factor	=	1.000	1.000

The choice of the value 1 for the true scale is completely arbitrary; any other choice would obviously result in all the scales being multiplied by that value. 250,000 reflections were used in each simulation, so that the random sampling error is $250000^{-1/2} = 0.002$, so it is clear that the calculated scale factors are all systematically biased to varying degrees. Because the error in $|F_o|$ is small compared with that in F_c , the choice of values of the $|F_o|$ error distribution parameters actually has virtually no effect on the results; in fact one may as well assume zero error in $|F_o|$. In contrast, the assumed magnitude of the random error in F_c is strongly positively correlated with the systematic errors in the scale factors.

Formula k_1 is the one in most common use, yet it is by far the least accurate! It is based on the following argument: the required scale factor k is obtained by minimising the sum of

squares of differences: $\sum_h (|F_o| - k|F_c|)^2$ with respect to k . One can of course use weights based on $\sigma(|F_o|)$, but there is still something fundamentally wrong with this argument.

The fundamental tenet of least squares is that when minimising $\sum_i (y_i - ax_i)^2$ to determine the unknown parameter a from a set of data points (x_i, y_i) , it is assumed that x is the independent variable which is known without error, and all the error resides in the dependent variable y . So it is being inherently assumed that $|F_c|$ is error-free and all the error resides in $|F_o|$ - hardly a realistic assumption! In the simulation above the errors in F_c were assumed to be 10 times those in $|F_o|$ and in reality they are probably never less than about 4 times; if this were not the case we would all regularly get refinement R -factors of 2 to 5%.

This suggests that one might do better simply by interchanging the rôles of $|F_o|$ and $|F_c|$ (and inverting the scale factor), thus minimising: $\sum_h (|F_o|/k - |F_c|)^2$. This gives formula k_4 above, which is indeed more accurate. Note that even if zero errors in $|F_o|$ are assumed, k_4 and k_5 are different in the acentric case, though they become identical for centrics.

The small deviation of the Kraut scale factor from the true value for the complete structure case is due to the approximation mentioned previously, and so the centric reflections give a worse estimate of the scale factor than the acentrics do. For example all else being equal, for 100% acentric data the calculated Kraut scale factor k_5 was 0.995, whereas for 100% centric the same scale factor was 0.985, 3 times less accurate. This arises for the same reason that the centrics always give higher R -factors.

It is somewhat paradoxical that the Kraut scale factor which is designed to work in the case of an incomplete structure (F_p after all represents an incomplete structure relative to F_{PH}), fares worse when applied in that very situation. The reason is of course that in order to deal correctly with coordinate errors, the equivalences have to be between $|F_p|$ and $|F_o|$, and between $|F_{PH}|$ and $|F_c|$, whereas to deal with missing atoms they would have to be the other way round.

REFERENCES

- Brünger, A.T. *X-PLOR (Version 3.1) Manual*, Yale Univ. (1992).
- Kraut, J., Sieker, L.C., High, D.F. & Freer, S.T. *Proc. Natl. Acad. Sci. USA* **48** (1962) 1417-24.
- Read, R.J. *Acta Cryst.* **A42** (1986) 140-9.
- Tickle, I.J. in "Isomorphous Replacement and Anomalous Scattering", *Proceedings of the CCP4 Study Weekend*, eds. W.Wolf, P.R.Evans & A.G.W.Leslie, SERC Daresbury (1991) 87-95.
- Vonrhein, C. in *Joint CCP4 & ESF-EACBM Newsletter on Protein Crystallography*, ed. K.S.Wilson, No. **28** (1993) 55.

Chicken egg-white lysozyme grown in microgravity and on earth for a comparison of their respective perfection

Susanne Weisgerber¹ and John R. Helliwell^{1,2}

1. Department of Chemistry, University of Manchester, M13 9PL, England

2. SERC, Daresbury Laboratory, Warrington WA4 4AD, England

Abstract

This report describes the crystallization of chicken egg-white lysozyme in microgravity and an experimental setup for the subsequent measurement of rocking widths. The spot size or rocking width is the first indicator of crystal perfection. Lysozyme was chosen as a test sample because much is known about it already and because its crystallization process is quick enough to be completed by the end of a 7-day flight. Control experiments on earth were carried out in parallel to the microgravity experiment, using identical crystallization chambers and material from the same batch.

Introduction

In macromolecular crystallography where crystal samples are fragile, sensitive to the X-ray beam and do not diffract strongly there is scope for improving data quality and resolution by growing crystals to as high a perfection as possible. In order to achieve that, the crystallization process needs to be understood and improved. In addition to adjusting the nucleation rate it is most important to control crystal growth itself. Ideally its rate should be dominated by diffusion only. In order to satisfy that condition convection, and, connected with it, turbulences, have to be eliminated. Currently there are two approaches: crystallization in gels, characterized by a very high viscosity, and crystallization in microgravity, where the influence of gravitation is eliminated. In both cases the system is not only free of convection but crystals do not sediment either.

Crystallization trials

The European Space Agency (ESA) and, on contract, Dornier GmbH, developed the Advanced Protein Crystallization Facility (APCF) as a standard tool for microgravity crystallization experiments on the Space Shuttle¹. APCF can accommodate 48 crystallization chambers (also called reactors) which are kept at one pre-selected process temperature. It fits into a Middeck Locker and can be activated and deactivated at the push of a button. Three types of crystallization chambers with different sizes of the protein volume have been designed for APCF. They allow application of the vapour diffusion method, the liquid-liquid interface diffusion method and the dialysis method. The latter was used in the experiment described here, and the corresponding reactor type consists of two quartz blocks

which are separated by a dialysis membrane (plus silicone seals and an O-ring for a leak-tight fit). The upper block contains the protein volume (188µl in this case), the lower block the salt volume and a small buffer volume. The salt and buffer volumes are separated by a cylindrical quartz plug which itself contains a salt volume. In order to activate the reactor, this plug is turned by 90° so that all volumes come into contact and the crystallization process can begin.

In preparation for the Space Shuttle flight STS-57 on which APCF was used for the first time the reactors had to be tested, and optimized crystallization conditions found. Eventually the crystallization chambers were prepared as follows: a solution of 7.9g NaCl in 100ml distilled water was filled into the salt volumes, then the small buffer volume filled with 0.04M acetate buffer (pH 4.7) and covered with a dialysis membrane. Then 21mg lysozyme (3x crystallized, dialyzed and lyophilized powder of chicken egg-white lysozyme from Sigma, product no. L-6876) were dissolved in 250µl of the same buffer. The solution was centrifuged for 30 seconds at 13000 rpm and 180µl pipetted into the protein volume. At room temperature (tetragonal) crystals appeared a few hours after activation and grew to ca. 0.5-0.8mm length within two days. They all grew on the walls and at the bottom of the protein volume. This effect had been observed on lysozyme before² and had led to problems in crystal harvesting. In subsequent trial crystallizations the upper glass block was therefore siliconized for 10 minutes prior to reactor assembly and filling. As a consequence, only a few crystals were growing on the walls, but the majority had sedimented. These crystals, which had grown into each other and formed a ca. 1mm thick solid layer, could not be used. The crystals which had grown on the walls, however, could be harvested now without inflicting damage. These conditions seemed well suited to the microgravity crystallization experiment where sedimentation cannot occur. In the planned ground reference experiment these conditions would not yield enough crystals so that the siliconization process had to be adjusted. Although it is not easy to control different lengths of treatment (1min and 5mins in comparison to 10 mins) were tried. 1 minutes' treatment was not long enough to have much effect. 5 minutes' siliconizing was sufficient to make crystal harvesting possible, whilst the majority of the crystals were growing on the walls. The four APCF reactors to be flown on STS-57 (the 'flight reactors') were therefore siliconized for 10 minutes and the four APCF reactors allocated to the ground control experiment (the 'spare reactors') 5 minutes. With this treatment harvesting of crystals was expected to be straightforward.

Crystallization of chicken egg-white lysozyme on Space Shuttle flight STS-57 and in a ground reference experiment

10 days before the scheduled launch all experimenters involved in the project met in two laboratories in Europe for the filling operations. Subsequently the crystallization chambers were stored in temperature-controlled (18°C) transport containers and kept at this temperature for transport to the USA, the space mission itself and until after return to the experimenters. Due to a launch delay that became known only after the filling procedure had already been completed, the crystallization chambers stayed in the pre-process condition, where the salt and buffer / protein volumes are still

separated, for 28 days. Prior to installation in the APCF, the protein chambers of all reactors were inspected and photographed for future reference by Dornier engineers. On this occasion some precipitate was observed in some of the reactors, and especially in the flight reactors rather than the spare ones. Therefore the scientists and engineers from ESA and Dornier GmbH decided to exchange the reactors prepared for the mission with those prepared for the ground control experiment. This decision was considerate enough but could be expected to lead to difficulties in retrieving enough undamaged crystals. Due to the different siliconization the space-grown crystals could be expected to stick to the walls and the earth-grown crystals to sediment.

The mission lasted 10 days overall, including a further delay of 2 days due to adverse weather at Cape Canaveral. The APCF was activated 5 hours after launch and deactivated 7 days and 10 hours later. After retrieval from the APCF, photographs were taken of all protein chambers again to provide an optical reference in case of damage to the crystals during the transport to the individual home laboratories.

Back in Manchester, we assessed the numbers and sizes of lysozyme crystals in each reactor under an optical microscope. In each of them, ca. 100-150 crystals up to a size of 0.8mm were observed; most were 0.6-0.7mm long. In the flight reactors ca. 80% of the crystals were growing on the walls, the rest on the top and the bottom membranes. The ground control experiment had yielded the same number of crystals and of the same size, but the vast majority had sedimented so that only the few crystals that were growing on the walls (ca. 10-20) could be harvested.

Crystal perfection assessment

Rocking width measurements need to be done at a synchrotron to ensure that the angular width of the diffraction profile is dominated by the mosaicity of the crystal rather than the beam divergence or cross section. One possible setup is a diffractometer with a small angular stepwidth (e.g. $\leq 0.0005^\circ$) combined with a monochromatic synchrotron beam of very low divergence (e.g. $\leq 0.001^\circ$). Then a representative number of reflections at a given Bragg angle are scanned and the full width at half maximum (FWHM) determined (e.g. Colapietro et al.³). A similar setup is available on station D23 at LURE. Because of a shutdown in July and August 1993 it was not possible to do any experiments there yet but beamtime is envisaged for early November 1993. Also, beamtime on the undulator beamline at the ESRF is granted for this project in November 1993. Due to the launch delays some beamtime allocated for this project at the SRS Daresbury was lost. It was possible, however, to obtain a half-share of 24 hours' beamtime on station 9.5 in Laue mode.

The Laue geometry offers another possible setup for crystal perfection measurements where the rocking width is derived from measured spot sizes. The mosaicity of the crystal determines the shape of Laue diffraction spots: their radial extension is given by (cf. Andrews et al.⁴)

$$\Delta_{\text{radial}} = 2\eta \frac{D}{\cos^2(2\theta)}$$

where η is the rocking width, D the crystal-to-detector distance and θ the Bragg angle. The transverse extension of a Laue diffraction spot is independent of η . According to this equation it is possible to

determine the crystal rocking width from the radial length of Laue reflections. In order for the effect to be measurable, especially in the case of well-ordered crystals (the limit of perfection theoretically obtainable is described in ref. 5) two requirements have to be met: the beam needs to be trimmed to obtain a very low divergence (in our case: to 0.002° horizontally, 0.0007° vertically) and a very long crystal-to-detector distance has to be chosen. In each exposure one filmcassette was set up at the end of the optical bench (1678mm from the crystal sample) and three others fixed on top of the radiation shielding at the back wall of the hutch so that crystal-to-film distances of nearly 2.5m could be achieved. The direct beam was recorded and its size and shape measured under a microscope. In addition, for each crystal two reference shots at different spindle angles were taken at a distance of 90mm from the crystal (after the long-distance measurements had been done).

Eight crystals (four space-grown, four earth-grown) from two of the eight crystallization chambers had been mounted for this experiment. Out of those, two (1 space-grown, 1 earth-grown) were damaged during retrieval and were therefore used to find the right exposure time. One of the remaining 6 crystals (earth-grown) turned out to have a high mosaicity, probably also due to damage during harvesting.

When comparing the patterns recorded at a distance of 2.5m it turned out that both earth-grown crystals had yielded significantly larger spot sizes than the worst space-grown crystal. The diffraction patterns of the crystals recorded at a crystal-to-film distance of 90mm, however, were not distinguishable with regard to their quality. The possible benefit of such more perfect protein crystals could be both in terms of lifetime in the beam and / or the resolution limit of diffraction. Further experiments will be undertaken at LURE and at the ESRF, as mentioned earlier, to make a complete set of crystal perfection experiments. The initial results, however, look promising.

Acknowledgements

We would like to thank the European Space Agency for providing the opportunity for this experiment. Dr. Bosch, Dr. Potthast, Dr. Lautenschlager and Dr. Stapelmann of Dornier GmbH for the excellent design of the APCF reactors and for their support in the handling of them. Our thanks to Dr. Sean MacSweeney for his help on station 9.5 at the SRS Daresbury and to Dr. William N. Hunter for valuable advice on crystallizations

References

- [1] R. Bosch, P. Lautenschlager, L. Potthast, J. Stapelmann (1992), *J. Cryst. Growth* **122**, 310-316
- [2] W. Littke, Ch. John (1986), *Science* **225**, 203-204
- [3] M. Colapietro, G. Cappuccio, C. Marcianite, A. Pifferi, R. Spagna, J. R. Helliwell (1992), *J. Appl. Cryst.* **25**, 192-194
- [4] J. S. Andrews, J. E. Hails, M. M. Harding, D. W. J. Cruickshank (1987), *Acta Cryst. A* **43**, 70-73
- [5] J. R. Helliwell (1988), *J. Cryst. Growth* **90**, 259-272

Some thoughts about processing, accessing and analyzing PDB data

Johan P.M. Postma

Biological Structures and Biocomputing Programme

EMBL, Meyerhofstrasse 1

Postfach 10.22.09

69012 Heidelberg, Germany

1. Introduction

This is a report on an ongoing project at EMBL, Heidelberg related to processing and accessing PDB data. When initially conceived this project was thought of as being the data manager in an interactive Computer Graphics environment. The goal was to establish a uniform environment to access PDB data. The philosophy behind it is simple. Create a parser that processes all relevant structural data from the ASCII PDB data files and which outputs the data in the form of a binary unformatted representation. This datafile can be regarded as a metafile from which all other data retrievals take place. In addition we design basic data access functions which can return data from this binary datafile. We will show that we have been able to design an environment in which the retrieval of any PDB data item is done in an efficient, easy and above all fast way.

We will discuss the modules that are part of the data generating and data retrieval system (YABP, K?, QUE - [1-2]), and will then describe some typical retrieval and query programs, ranging from simple to more complex.

The current implementation is on a VAX/VMS architecture, however an implementation on a UNIX/OSF1 based machine is planned. The following people have made significant contributions to the project: Heinz E. Bosshard, Rob van Unen, Hans Neervoort and Jean P. Canuti.

2. Objectives

Getting specific data items from the ASCII PDB files is not always a simple and straightforward task. Many of these problems arise from the fact that the PDB format of storing structural data is based on a human readable form. Furthermore much of the information about certain data objects is given in an implicit form. Because the data is stored as ASCII files and thus can be edited many inconsistencies or contradictory information can be encountered. Our initial goal was to create a "general" PDB parser (YABP - data input module/data file generating tool), which should process all relevant structural data from the PDB files. The basic data access functions (K? - data output modules/data retrieval modules) should be able to get any data item, thus making the input module of any application program that needs PDB data independent from the ASCII PDB files (i.e. from parsing these files). Another requirement was that the software implementation of the data access mechanism should be as fast as possible.

A graphical layout of how these tools build up the data retrieval system is given in Figure 1.

It is important to point out that it is not our goal to create a database management system; for example, there are no data update modules. In order to maintain and update PDB data the usage of a commercial database management system is probably the safest choice.

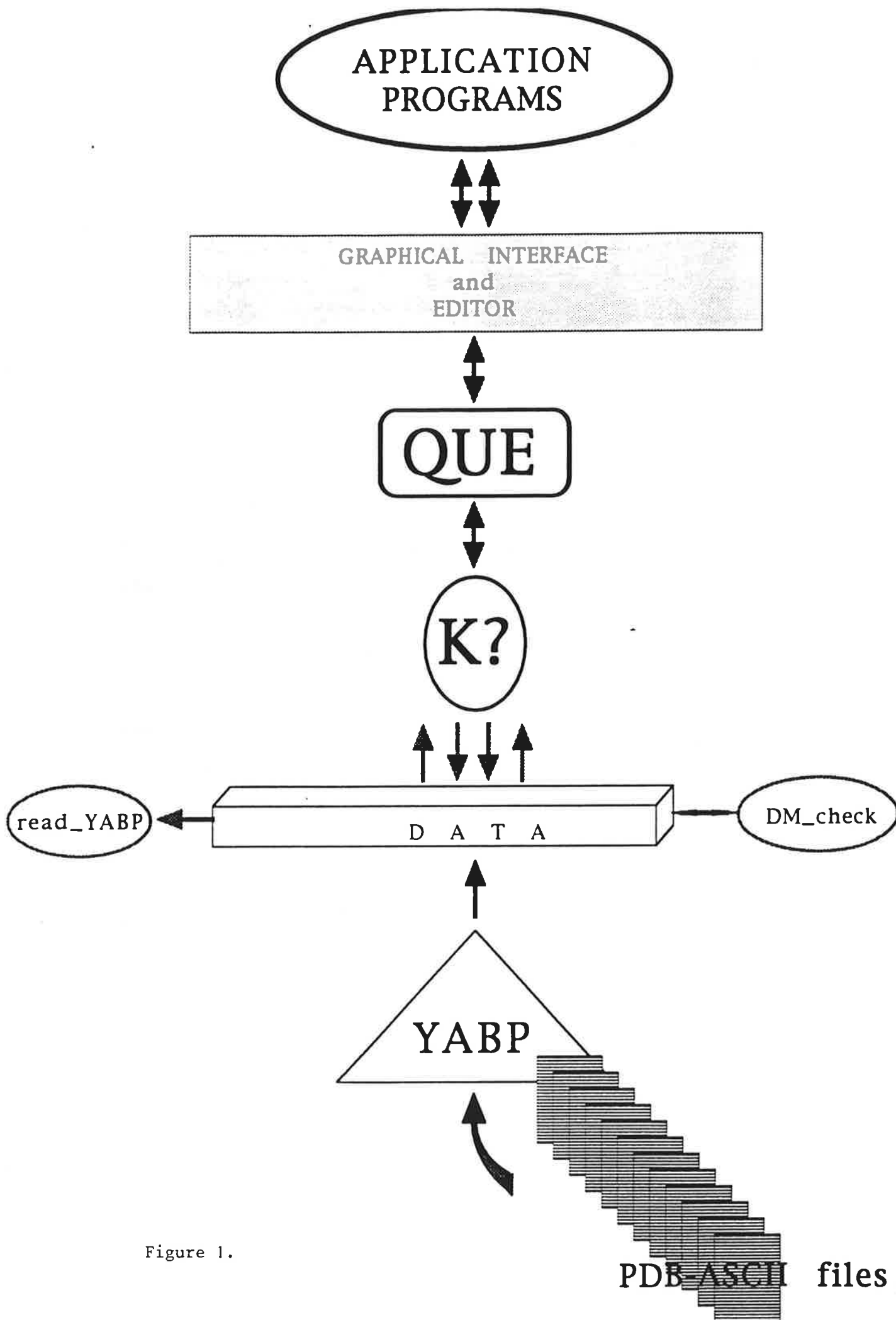


Figure 1.

3. YABP - input module/data generating module

This utility processes all PDB files, checks for inconsistencies and puts the structural data into one unformatted file. Only syntactical errors are checked i.e. no "data validation" of any kind is done. We decided to make this stage as simple as possible in order to anticipate future extensions and major changes to the PDB files. We have therefore tried to separate the input functions from the output functions, thus making a transition to other "data formats" easy. Furthermore it is to be expected that whenever a "new" formatted release of the PDB data will emerge (CIF) it will include the appropriate parsing functions. In this case an adaptation of YABP will be extremely easy. Data validation (or data error-detection) can easier be implemented as an application program based upon K?, instead of making it part of the parsing stage. The latter choice would only make the PDB parser unnecessarily complex and hard to maintain. Presently about 10% of all PDB files (which number 1250 as of May 1993) are rejected. Most of these rejected entries contain structures determined by NMR methods in which multiple models are deposited in one entry. Entries in which the SEQRES information does not match the residue information found in the ATOM records are rejected. Entries that have detailed helix information (same holds for strands/sites), but no corresponding ATOM and/or residue information are also rejected. Parsing all PDB files takes about 45 minutes of CPU time on a VAX 9000-420. We decided to process all PDB files, however there exists a facility to only view a subset (family) of PDB entries. Thus accessing data can be done on all PDB data or on a user-defined subset. This allows for a highly flexible data access mechanism. So, one might choose for example, only all lysozymes and their mutants.

YABP produces structured data objects, a graphical layout of these data objects can be found in Figure 2. Most of them are similar to the official PDB description, some 'added' data objects like SET, CHAIN, etc.. contain explicit information which can only be implicitly found in the PDB files.

We decided not to include derived data, i.e. data that can be produced by analysis programs using the raw PDB data, like secondary structure analysis, atomic accessibility, atomic charges, atomic electrostatic potential values etc.....these data can be derived using a specific algorithm and because different algorithms have different advantages and disadvantages we did not want to force any particular algorithm upon users. One derived (or merely abstracted) data item, namely the pseudo residue sphere, has been added in order to allow for quick interresidue distance calculations. We do however anticipate the necessity of using certain derived data. We have two different schemes for dealing with these kind of data. Either it is a data-field in the datafile (atomic accessibility), which can be filled in by the user or it is a separate data file (see the ULLI application program described below) loosely linked with the raw data file. Again calculating derived data can be best implemented using the access functions described in the next section.

4. K? - output modules/data retrieval modules

K? is essentially a set of basic data access functions which can retrieve any single data item from the data file produced by YABP. It is composed of two separate parts. One part is visible to the user ("upwards module"), which can retrieve data of any kind by issuing data requests. This module checks whether the request is valid. If so it breaks down the (possible) aggregate data-request into basic request operations which are flushed to the second part ("downward module"), which is doing the actual data transfer. If the requested data is present, it is sent back to the application program. We decided to hide this "downward" interface with the underlying data, thus making an application program independent of where the data is residing and freeing it from any other detailed data manipulation schemes.

K? has to be regarded as a toolkit for an experienced programmer who wants to make his or her own PDB-query program. Query programs can easily be designed in Fortran, Pascal and C. K? only contains data retrieval modules i.e. it does not contain any data update modules or computational modules.

5. Some example query programs

Over the period we have assembled a set of query programs. Some of them are simple and merely show the functionality of the access system. Others are more complex and do some form of data validation and analysis. We will briefly mention some of them, including typical CPU timings. All timings are done on a VAX/VMS 9000-420.

The total datafile contains 1042 PDB entries and has a size of roughly 66 Mbytes (this version does not contain HETATM information). The output of the query programs is presented as ASCII output. Whenever possible the output is also presented in such a form that it can readily be plotted by a 2D plot facility or when 3D data is involved, as a Midas-script. This script file can be sourced in the Computer Graphics program MidasPlus (UCSF) [17] and the results of the query can be readily displayed. In this way we have created a loosely coupled environment with a 3D graphics application program. Any other 3D graphics program that has a similar facility as MidasPlus could be used instead.

- caca_dist

This query program calculates for each C α atom in the datafile its distance to its nearest neighbour, second nearest neighbour up to its fourth nearest neighbour. The program identifies cis-peptides and checks for wrong C α -C α distances. A similar query program does this specifically for helices and strands. See Table I and Figure 3.

CPU time taken 30 seconds

Table I (Output of example query program caca_dist)

entry	origin	residue/chain	residue/chain	dist	cis-peptide
12CA	XRAY (2.40)	29(SER)	30(PRO)	3.1112	CIS
12CA	XRAY (2.40)	125(THR)	127(LYS)	4.1890	wrong
12CA	XRAY (2.40)	201(PRO)	202(PRO)	3.1572	CIS
(entries deleted)					
9RSA	XRAY (1.80)	92(TYR)A	93(PRO)A	2.9308	CIS
9RSA	XRAY (1.80)	113(ASN)A	114(PRO)A	2.9003	CIS
9RSA	XRAY (1.80)	92(TYR)B	93(PRO)B	2.9379	CIS
9RSA	XRAY (1.80)	113(ASN)B	114(PRO)B	2.9503	CIS
9XIA	XRAY (1.90)	186(GLU)	187(PRO)	2.9479	CIS
Number of PDB-entries		1042			
Number of nearest neighbours		257369			
Number of second nearest neighbours		255888			
Number of third nearest neighbours		254463			
Number of fourth nearest neighbours		253044			
Number of bad CA-CA (< 3.6 > 4.0)		1737			
Number of cis-peptides		670			

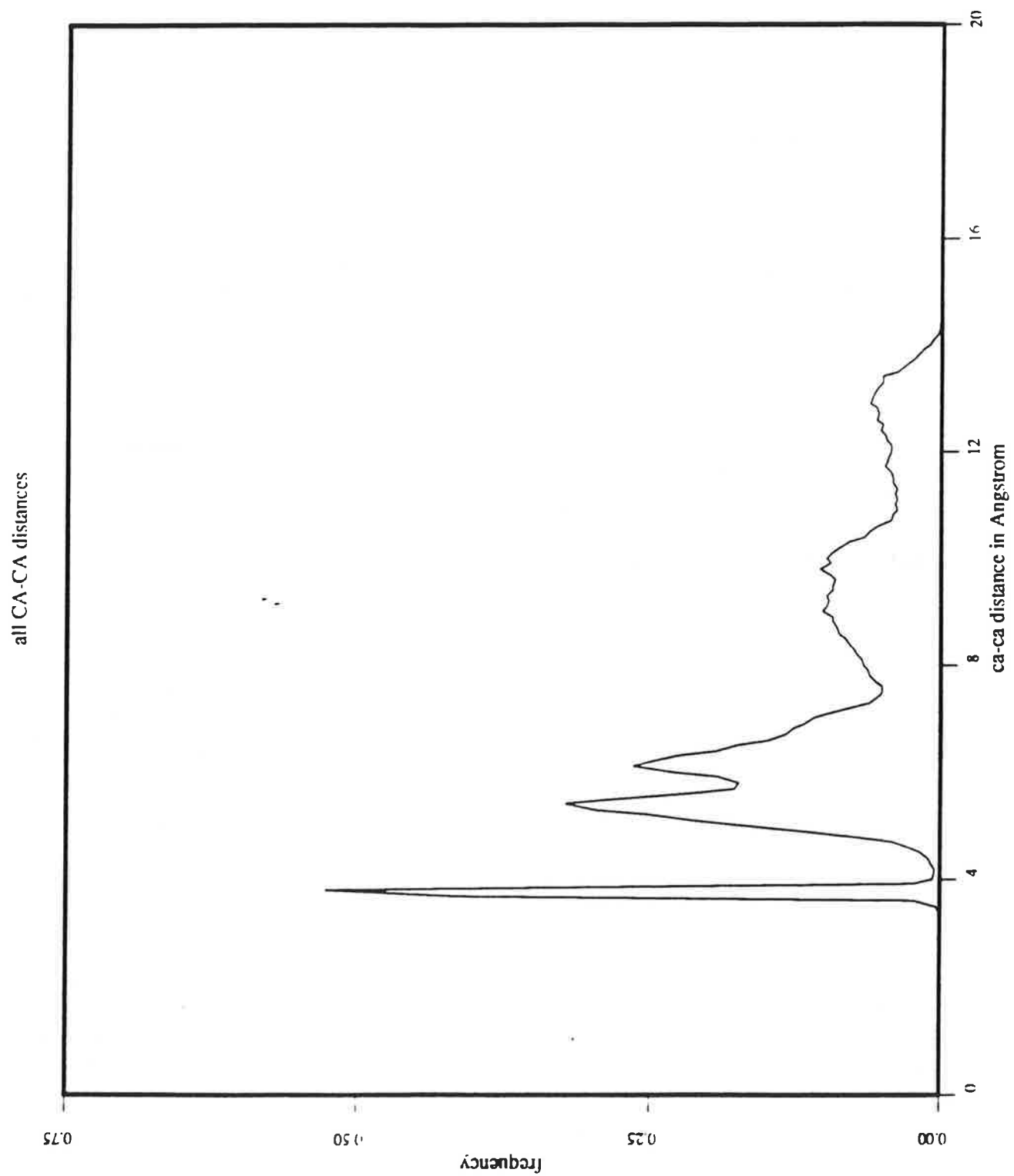


Figure 3.

- proh2

This query program identifies all proline residues which are in the second position in a helix. If such an helix is found calculate the ϕ/ψ dihedral angles of this pro-line.
CPU time taken 13 seconds. (Table II).

Table II (Output of example query program proh2)

entry	helix	sequence	phi / psi	resolution	helix class
1ALC	B	LPE	-60.1413 / -37.6314	XRAY(1.70)	1 Right-handed alpha helix
1ALD	I	TPE	-55.2985 / -34.2967	XRAY(2.00)	1 Right-handed alpha helix
1APM	E	EPH	-70.7281 / -28.9283	XRAY(2.00)	1 Right-handed alpha helix
1ATP	E	EPH	-65.6512 / -39.3159	XRAY(2.20)	1 Right-handed alpha helix

(entries deleted)

9PAP	L3	YPW	-61.2961 / -33.3308	XRAY(1.65)	1 Right-handed alpha helix
9WGA	DA	GPG	-47.9191 / -42.4679	XRAY(1.80)	1 Right-handed alpha helix
9WGA	DB	GPG	-53.7733 / -53.9647	XRAY(1.80)	1 Right-handed alpha helix

Number of PDB-entries 1042
Number of cases found 618

- hssbs

This query program finds a helix joined to a strand by an SS bond. The spacing between the two cysteines should be in the range of 10 to 20 residues. Furthermore, if such a helix/strand motif is found, calculate the S-S bond distance as well as the χ_3 dihedral angle
CPU time taken 1.4 seconds (Table III).

Table III (Output of example query program hssbs)

entry	res1	res2	SS-bond	CHI3	gap	helix-class
1BP2	84	96	2.078	-96.824	12	1 Right-handed alpha helix
1CHG	168	182	2.141	-48.220	14	1 Right-handed alpha helix
1CHO	168E	182E	2.031	-87.046	14	1 Right-handed alpha helix
1CPB	66	79	?????	?????	13	1 Right-handed alpha helix
1EST	168	182	2.031	-83.204	6	1 Right-handed alpha helix
1GCT	168A	182A	1.979	-88.500	14	1 Right-handed alpha helix
1GCT	168A	182A	1.979	-88.500	14	1 Right-handed alpha helix
1NTP	168	182	1.778	91.794	14	1 Right-handed alpha helix

(entries deleted)

5CHA	168B	182B	2.016	-80.784	14	1 Right-handed alpha helix
6CHA	168A	182A	2.086	-54.422	14	1 Right-handed alpha helix
6CHA	168B	182B	2.052	-82.995	14	1 Right-handed alpha helix
6EST	168	182	2.022	-87.376	16	1 Right-handed alpha helix
7EST	168E	182E	2.029	-113.516	16	1 Right-handed alpha helix
8EST	168E	182E	2.027	-84.099	16	1 Right-handed alpha helix
8GCH	168	182	2.032	-81.498	14	1 Right-handed alpha helix

Number of PDB-entries 1042
Number of cases found 43

- aa

This query program identifies aromatic pairs (F/W/Y/H). For each pair the relative orientation of the aromatic rings is being calculated, which results in a perpendicular parallel or random orientation. The distance between the C α 's of each pair is calculated as well as the number of residues in between. Furthermore the pairs are grouped into aromatic clusters. Only Xray structures with resolution less then 3.0 Å are accepted.
CPU time taken 115 seconds. (Table IV).

Table IV (Output of example query program aa)

entry	res	chain	res	chain	gap	CA-CA	orientation	origin/resolution
12CA	5(TRP)		16(TRP)		11	6.52	perpendicular	XRAY (2.40)
12CA	5(TRP)		64(HIS)		59	9.41	random	XRAY (2.40)
12CA	7(TYR)		16(TRP)		9	9.84	perpendicular	XRAY (2.40)
(entries deleted)								
1COL	86(TRP)A		161(TYR)A		75	7.43	parallel	XRAY (2.40)
1COL	125(TYR)A		130(TRP)A		5	8.31	perpendicular	XRAY (2.40)
1COL	86(TRP)B		161(TYR)B		75	7.42	parallel	XRAY (2.40)
1COL	125(TYR)B		130(TRP)B		5	8.23	perpendicular	XRAY (2.40)
(entries deleted)								
9XIA	178(PHE)		212(TYR)		34	4.77	parallel	XRAY (1.90)
9XIA	243(HIS)		285(HIS)		42	7.88	random	XRAY (1.90)
9XIA	269(PHE)		270(TRP)		1	3.82	perpendicular	XRAY (1.90)
Number of PDB-entries processed						1042		
Number of parallel pairs of aromatics						1529		
Number of perpendicular pairs of aromatics						3943		
Number of random pairs of aromatics						2294		
Number of cases found						7899		

- ulli

This is a secondary structure motif searching program. Given a specific arrangement of helices and strands (target motif) it is able to identify similar "motifs" in the datafile. This query program uses derived data which resides in a separate datafile i.e. all relationships between helices and strands are kept in this datafile. The relationships are: the angle between all secondary structure elements and the distances of closest approach respectively. Depending on the number of helices and strands in the target motif and the number of hits found a typical motif-query takes between 30 and 60 seconds of CPU time. A mechanism is provided that allows the query to accept hits with one or more helices and/or strands being 'missed'. This, however, can increase the amount of CPU time involved significantly. This tool can also be used to search for substructures in a larger set of structures.

6. QUE - General QUERy system

While K? has to be regarded as a toolkit for an experienced programmer to design a customized data retrieval / query program, QUE is a general query system which should be able to query a large subset of all possible queries. Certain queries which are doing statistics over the complete database (like caca_dist) are much more easily implemented as a standalone program-tool. QUE was initiated during 1991/1992 and is still a prototype. Below we will discuss some specifics of this query system.

QUE frees the user from designing a typical query program. It is actually nothing but an application program based on K?. It uses a so-called description file which has to be supplied by the user. This file contains a detailed description of the query to be done and it allows for a high degree of flexibility. The objects that can be queried closely follow the natural hierarchy of the data objects in the data file. These objects can be regarded as the nodes in a tree (Figure 4). Four "levels" can be distinguished. Lines or links connecting nodes between two different levels and lines or arcs connecting two nodes of the same type describe the relationship between the queried objects (i.e. the nature of the query). Whereas there can only be one ENTRY or START node, all other levels may have more than one instance of an object. Each object (node) in the tree may have attributes (or functions) associated with it. We can distinguish between three different types of attributes.

1) Each object can have (primary) attributes which only apply to that specific object, i.e. the attribute of the ENTRY level called 'origin' governs the 'view' of that object. An object of the SECTRU level can have attributes like 'length-of-sectruct' etc....

2) A line (link-function) connecting two different levels implies that the object in the lower level is contained in the upper level. This is expressed as an (foreign) attribute in the upper level object

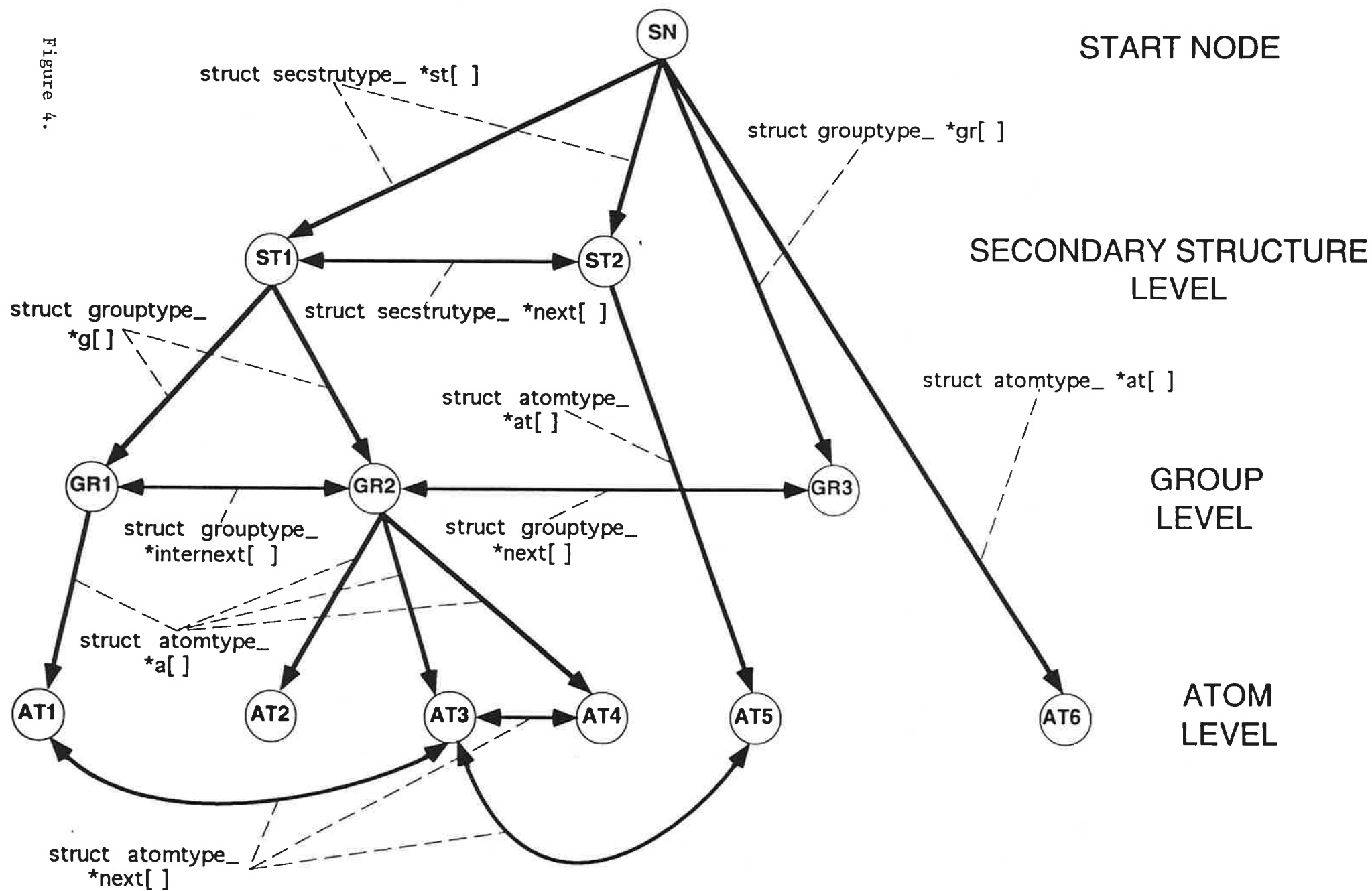
3) A line (link-function) connecting two different nodes in the same level implies a relationship between those nodes i.e. the angle between two secondary structure elements or the number of residues (groups) between two queried groups. This relationship is expressed in both nodes as an (relational) attribute 'pointing' to a relational object, i.e. the relationship between the nodes is regarded as being an object, which on its own can have (primary) attributes.

The following example shows a typical query which can be issued. This example is actually a translation of the example query program proh2. QUE is still under development. Much of the nomenclature could be made more verbose. Secondly the description of a more complicated query can become a practical problem. We plan to investigate whether a Graphical Editor of some kind may be of help in designing such complicated queries.

Example: Find a Proline in the second position in a helix and calculate ϕ/ψ

```
DEFINE ENTRY      /stpointer    = @s1      ! 'Points' to i.e. contains object s1
                  /filename     = "proh2.out" ! defines where the output should go
                  /origin       = XRAY      ! only accept XRAY structures
                  /resolrange   = 0,2.5     ! with resolution less the 2.5 A
DEFINE SECSTRU    /id           = %s1      ! Identification of this object
                  /type         = helix     ! object is of type helix
                  /grpointer    = @g1      ! 'Points' to i.e. contains object g1
DEFINE GROUP      /id           = %g1      ! Identification of this object
                  /type         = pro       ! object is of type pro
                  /N_ter_range  = 1,1      ! relative position of this object in s1
                  /function     = PHIPSI    ! function to be calculated
```

Figure 4.



7. Data validation - detection - correction

During the last couple of years there has been much discussion about data validation. This topic definitely deserves more attention. We can discriminate between two sources of errors. Obviously conflicts between spacegroup information, cell dimensions and scale cards can more easily be corrected than nonbonded interactions which show unrealistic features. Correcting the latter is of a more fundamental nature than the former. One of the application programs that we have build upon K² is actually meant as a data error-detector (see DM-check in Figure 1). Errors of the first class are very likely caused by the fact that the PDB datafiles are ASCII files i.e. editable and thus more sensitive to typing errors. Referring to the original literature usually overcomes these problems. The situation becomes more complicated if one equates data validation with data quality. What should one do with data that shows unrealistic bonded or nonbonded information i.e. data that contains errors. These "errors" may have different origins. Data, in particular the Cartesian coordinates deposited in the PDB files, are based on experimental data which at least contain systematic errors. In this context we ignore theoretically modelled structures. Errors may also come from the fact that the protein structure has been refined using a particular protocol (procedural errors). But the most serious source is that of atomic models which are incorrectly built into the electron density (model error - incorrect fold). Schemes [4-10] have been proposed during the years to have more detailed information about the quality of a modelled molecular structure or local molecular substructures. In principle it is possible to detect "problem" areas in structural data. In order to detect errors in bonded information one may apply a set of ideal bond lengths, bond angles etc.. and calculate deviations from it. For non-bonded interactions one can apply a set of interatomic potential functions and calculate an energy profile. We have designed such a tool which is based on the GROMOS forcefield and which has to be incorporated in our data access environment.

Applying such a tool to a molecular structure may highlight certain problem regions. However, one must decide on the allowed error-margin. Unfortunately the relationship between the quality of the experimental data (structure factors for Xray and NOE's for NMR) and the quality of the corresponding set of cartesian coordinates is not clear. If we use a certain set of bonded and non-bonded parameters and their application to a set of coordinates detects a region which shows some unlikely features, then this does not necessarily mean that this region has been built and refined incorrectly. The only usage of error-detection tools is to point at possible problem areas in the modelled structure. Data correction can only be achieved by further refinement using the underlying experimental data.

8. Conclusions

Several labs have come up with solutions on how to manage PDB data and derived data ranging from in house written software [4-5,11-12] to using commercially available database management systems [3-5,13-15]. The datafile we have presented in this letter can be regarded as a specific user view i.e. a specific extraction of structural data which could come from a more complete database management system. The scheme we have discussed is, however, sufficiently powerful to perform some fundamental data manipulations, like data statistics and data analyses. Our emphasis is on data-access speed. In that respect a representation of the data in the form of a flat file plus a set of basic data-access function is valid as it does not create any unnecessary overhead. Furthermore, as most of the application programs which are part of our system perform their tasks very efficiently it is possible to quickly change some query input parameters and redo the query. In principle every query based on our data file circumvents the parsing stage of the ASCII PDB files and thus saves a fair amount of CPU time. Another advantage we see is that our system is flexible. If a typical query is not part of QUE one could either design one's own query or extend the functionality of QUE. The former is the best solution if one wants to do specific statistics on all of the PDB data. Still much work can be done in order to complete the access system.

The European initiative in which several labs participate (BRIDGE [16]) is certainly a good way to go. It is hoped that more groups can give contributions to this project. We believe that the only way to treat data correctly is through a spectrum of different tools (both computational and visualization tools), which is as wide as possible and in which the end user is not restricted by the way the data has been arranged by the database designers/managers.

9. References

- [1] EMBL research reports (1991/1992)
- [2] K? user manual; EMBL technical document (1992)
- [3] Sillince, J.A.A. and Sillince, M., *Molecular Databases for Protein Sequence and Structure Studies*. Springer Verlag. 1991. ISBN 3-540-54332-5
- [4] Thornton, J.M. in *Current Opinion in Structural Biology* (1992), 2:888-894 and references therein
- [5] Thornton, J.M. and Gardner, S.P., in *Chemical Design Automation News* (April 1993):18-22 and references therein
- [6] Morris, A.L. et. al., *Proteins* (1992), 12:345-364
- [7] Luthy, R. et.al., *Nature* (1992) 356:83-85
- [8] Vriend, G. et. al., *J. Appl. Cryst.* (1993), 26:47-60
- [9] OOPS. This is part of the "O" set of programs by Alwyn Jones
- [10] PROCHECK. Laskowski, R.A. et. al., in *CCP4 and ESF-EACBM Newsletter*, (1992), 27:34
- [11] IDITIS (TM) Oxford Molecular, Sandford-on-Thames, Oxford OX4 4GA
- [12] Gray, P.M.D. et. al., *Protein Engineering* 1990, 3:235-243
- [13] Islam, S.A. et. al., *Protein Engineering* 1989, 2:431-442
- [14] Huysmans, M. et. al., *Proteins* 1991, 11:59-76
- [15] Bryant, H.S., *Proteins* 1989, 5:233-247
- [16] Wodak, S., in *CCP4 and ESF-EACBM Newsletter*, 1993, 28:24-25
- [17] Ferrin, T.E. et. al., *J. Mol. Graphics* 1988, 6:13-27

10. Figures

Figure 1.

Graphical representation how the several modules constitute the data access system. The Graphical Editor is a possible future extension of our system. An application program can either get it's data from QUE or it can use the K?'s basic data-retrieval functions directly.

Figure 2.

A graphical layout of the objects in the data file.

Figure 3.

2D output of example query caca_dist. A plot showing the C α -Ca distribution. Wrong C α -C α nearest neighbours and cis-peptide C α -C α distances are not shown

Figure 4.

A tree-like representation on how the data objects are related in the QUE system. One can distinguish four different "levels". Lines connecting nodes of different levels imply that the object in the lower level is contained in the object belonging to the higher level. Lines connecting nodes that belong to the same level implies a certain relationship between these nodes

BIOMACROMOLECULAR SPELEOLOGY

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

Most substrate-enzyme and ligand-receptor interactions in proteins take place inside clefts or cavities. Studying cavities ("biomacromolecular speleology", for short) may give insight into the mechanism of such interactions and might thereby help in the design of new ligands, substrates or inhibitors.

Many programs exist with which one may detect and/or delineate and/or measure and/or display cavities (see ref. 1 and papers cited therein). However, since we needed a program which does all of this, does it quickly and interfaces to O (2), we were forced to write our own: **VOIDOO** (1). Since its conception, about a year ago, VOIDOO has evolved into a multi-functional tool which is in use on a routine basis in our lab and elsewhere.

DESCRIPTION

In brief, VOIDOO has the following capabilities:

- detection of a specific void or all voids inside a biomacromolecular complex
- detection of certain cavities which are connected to the "outside world"
- delineation of cavities (*i.e.*, finding their extent in space)
- measurement of cavity volumes
- generation of plot files for O which enable visualisation of cavities
- generation of molecular surface plot files for O
- measurement of molecular volumes

Detecting cavities which are connected to the "outside world" is non-trivial. We have implemented a method (1) which we call atomic fattening: if a certain cavity is "open" while the atoms have their normal Vanderwaals radii, we gradually increase all radii, until the cavity either vanishes or becomes a real void. In order to discern these different types of cavity, we use the following operational definitions:

- a **void** is a cavity which is completely surrounded by the protein and which is therefore closed off from the outside world;
- an **invagination** is a cavity which is connected to the outside world, but which would be closed off if the atomic radii were increased;
- a **pocket** is a cavity which is connected to the outside and which cannot be closed off by increasing the atomic radii.

VOIDOO will detect voids and invaginations, but it cannot pick pockets. Note

that the definition of the **extent** of an invagination is subjective and therefore more or less arbitrary (where does the cavity end and the outside world begin?).

VOIDOO can be set to measure and display cavities in three different modes:

- (1) *Vanderwaals cavity*: the cavity comprises the complement of the Vanderwaals surface of the surrounding atoms;
- (2) *Probe-accessible cavity*: the cavity comprises all of space that can be accessed by the centre of a probe sphere (default);
- (3) *Probe-occupied cavity*: the cavity comprises all of space that can be occupied by a probe sphere (slow).

The input to VOIDOO consists of three parts:

- (1) a library which defines which residue types are considered to be part of the protein (or other molecule) as well the Vanderwaals radii of various atom types;
- (2) an ordinary PDB file;
- (3) parameters which determine what the program should do and how.

An example of part of a library file is given below:

```
REMA *** AMBER van der Waals radii ***
ELEM ' N' 1.75
ELEM ' C' 1.85
...
REMA CH2 = 1.925
SPAT 'SER* CB ' 1.925
...
REMA *** allowed residue types ***
RESI 'ALA'
...
RESI 'VAL'
RESI 'CPR'
END
```

We distribute two protein-oriented libraries, one containing Vanderwaals radii taken from the Amber force-field and one containing MS-like radii. Libraries for other types of molecules (DNA, oligosaccharides etc.) are easy to make.

The output of VOIDOO, in a cavity-detection run, also consists of three parts:

- (1) normal screen output regarding the program's operation;
- (2) a log file containing a summary of the results (extent of each cavity, its volume, its centre of gravity and a list of non-protein atoms that lie inside it as well as protein atoms that border on it);
- (3) files for use with O, namely for each cavity: a map file which can be displayed and a macro which draws all residues inside the cavity as well as those that line the surface of the cavity. The map files are written in EZD-format (EZD ~ "easy density"; this is a formatted ASCII file which can be converted into DSN6-format or read into O directly).

Figure 1 is an example of an O display, showing the cavity as a chicken-wire

contour (on SGI workstations, cavities may also be rendered as semi-transparent surfaces), the ligand inside the cavity (fat lines) and the residues that surround it. This example pertains to cellular retinol-binding protein, whose structure was recently solved in our lab (3).

AVAILABILITY

VOIDOO is one in a series of "O-dalisques", *i.e.* programs that work in conjunction with O. VOIDOO runs on SGI, ESV and DEC ALPHA/OSF1 workstations. For more information, contact TAJ (preferably via E-mail: "alwyn@xray.bmc.uu.se").

REFERENCES

- (1) G.J. Kleywegt & T.A. Jones, "Detection, delineation, measurement and display of cavities in macromolecular structures", submitted for publication.
- (2) T.A. Jones, J.Y. Zou, S.W. Cowan & M. Kjeldgaard, "Improved methods for building protein models in electron density maps and the location of errors in these models", *Acta Cryst. A* **47** (1991), 110-119.
- (3) S.W. Cowan, M.E. Newcomer & T.A. Jones, "Crystallographic studies on a family of cellular lipophilic transport proteins. Refinement of P2 myelin protein and the structure determination and refinement of cellular retinol-binding protein in complex with all-trans-retinol", *J. Mol. Biol.* **230** (1993), 1225-1246.

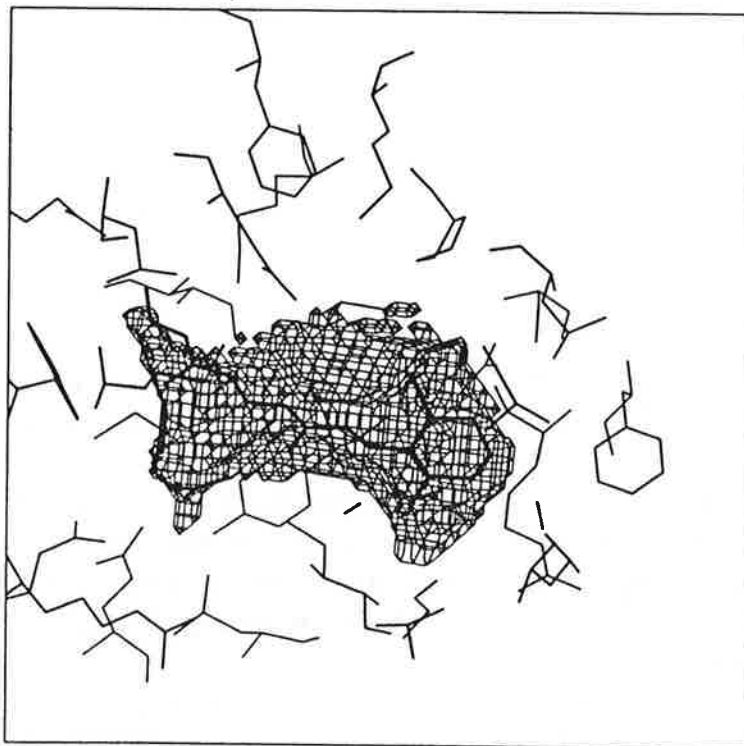


Figure 1) Cavity inside cellular retinol-binding protein, crbp (3). The all-trans-retinol ligand inside the cavity has been drawn with fat lines. The residues that line the cavity were drawn by an O macro that also was produced by VOIDOO. The volume of the protein is $(14,440 \pm 20) \text{ \AA}^3$, that of the retinol ligand is $(337 \pm 1) \text{ \AA}^3$ and that of the probe-accessible cavity is $(174.5 \pm 0.5) \text{ \AA}^3$.

DL_POLY: A macromolecular simulation package

T. Forester and W. Smith,
S.E.R.C. Daresbury Laboratory,
Daresbury,
Warrington WA4 4AD,
England

October 26, 1993

The following is a summary of the DL_POLY presentation given at the recent CCP5 AGM at Keele University, 17 September 1993 . . .

The DL_POLY package is funded by the SERC through the SBCC (Science Board Computing Committee). The motivation of the package is to provide a parallel macromolecular simulation package free to the academic community. The reasons for this are many. For example, while there are a number of commercially available (parallel) macromolecular simulation packages, these often have the drawback that they are not free. Moreover, while these packages tend to have very good "front ends" getting to the source code that drives the simulation dynamics can be difficult. Thus in these "black boxes" it can be difficult to know what the package is actually doing to the simulation and modification or extension of the code is not easily achieved. The DL_POLY package is designed to avoid these problems. The source code is freely accessible, extensively documented and highly modular in nature. We wish to avoid the "black box" approach to simulation by providing code that is readily accessible to both verification and modification by users. The package is thus aimed at users who desire intelligent and informed control over their simulations. The package comes as a series of modules that the user "bolts" together for their own particular application. A series of "makefiles" is provided to facilitate this. The package is able to handle a broad range of macro systems - from material science applications (e.g. crystalline and amorphous solids) to simulation of molecules of biological interest (e.g. DNA, enzymes etc) plus a good deal in between. The target machines are Multiple Instruction Multiple Data (MIMD) parallel machines (eg. the Intel iPSC/860 at Daresbury) and single processor workstations. The idea being that users have a code that both works efficiently on a workstation and that will be quite painless to transfer onto parallel supercomputers should they gain access to such facilities. In developing this package beyond a basic capability we are dependent upon the support of the CCP5 community, for example, to supply additional modules for analysis or system generation.

We have adhered to a fairly stringent programming style. All modules are in FORTRAN 77 (modules written in C are acceptable provided they are FORTRAN callable). Variables and arrays are passed as arguments through subroutine calls to facilitate the interfacing of modules. Consequently there are no common blocks present in the source code. The modules are also extensively documented, the release I documentation totals about 220 pages.

Features currently available:

Release I is built around two basic strategies. The first is that of “Replicated Data” meaning that all processors have a complete copy of the atomic co-ordinates, velocities and forces but that evaluation of pair-wise interactions, integration of the equations of motion, application of constraints etc are split equally among the available processors. This approach works well for systems of up to at least 20000 particles and on machines with up to 100 or so processors. It has the advantage that the algorithm will run just as readily on a single processor as on a parallel machine, it is relatively straightforward to program, and results in excellent parallelisation efficiencies. When the number of processors become very large the algorithm suffers from global communication costs (the requirement that arrays etc are summed globally across all processors), but communication overheads also affect other parallelisation strategies. Replicated data is also more memory intensive than other strategies (e.g. domain decomposition) but we do not consider this to be a serious restriction at present. The second intrinsic feature of the code is that it is atomistic in nature - each site is assigned a mass, charge, coordinated etc and evaluation of the Verlet neighbourhood list is done on the basis of site-site separations. At present we do not have the capacity to handle rigid body equations of motion, massless sites and so forth nor to base real space cutoffs on a “molecular group” strategy.

Other features in the package are given below. In each case the required feature is selected by setting an integer variable in the job CONTROL file. As the package is modular in nature additional features are readily added simply by defining new properties for additional values of the integer keys. For example, all evaluations of periodic images take place within one subroutine (“images”) and additional periodic boundary conditions can be introduced simply by modifying this single subroutine. Note that all the features that follow are couched in the Replicated data / Atomistic code framework.

- A selection of periodic boundaries (free space, cubic, orthorhombic, parallelepiped, truncated octahedral, rhombodal dodecahedral, two dimension periodicity)
- A selection of electrostatic methods (three dimensional Ewald sum, Coulombic, truncated and shifted Coulombic, distant dependant dielectric constant)
- All common atom-atom potentials.
- Flexible bonds and rigid bonds (SHAKE).
- Valence angle and dihedral angle potentials
- A selection of canonical thermostats (Nosé-Hoover and Gaussian constraints) and the non-canonical Berendsen thermostat.
- Single and multiple timestep algorithms
- On-line radial distribution functions and mean squared displacements. The general policy is that analysis routines are intended to be used off-line. This is because the analysis required is usually quite problem specific and so not incorporated into the release package. RDFs and MSDs are however almost universally required and it makes sense to calculate them “on-line”. They can be turned “on” or “off” by keys in the job CONTROL file.

- Utility modules to aid in generation of initial configurations. These include setting up lattice structures, creating structures from database files (such as PDB files), and adding thermalised water to a structure.
- A series of utilities to aid in generation of force field files - particularly for polymer and bio-simulations. Utilities for helping generate GROMOS and AMBER force fields are included.
- Analysis routines: Correlation functions and statistical analysis. These are designed for off-line analysis.

The figure shows the performance of DL-POLY as a function of the number of processors in use. The test case is an antifreeze protein emmersed in 1234 SPC waters. Cubic periodic boundary conditions are in use and electrostatic interactions are calculated using the Ewald summation. The plot shows $\log(\text{average time per step})$ vs the dimension, d , of the hypercube. The number of processors in use is 2^d and ranges from 2 through to 64 processors. On this plot perfect parallelization would correspond to lines with slope -1. This is what is seen for all subroutines except the reciprocal space summation in the Ewald sum (Ewald1) and for SHAKE. The “perfect parallelisation” is a consequence of the Replicated data strategy as no (or very little) inter-processor communication is required during these parts of the calculation and the work is equally shared among the processors. This is true for tasks such as construction of the Verlet neighbourhood list (“parlst”), and evaluation of real space pair interactions (“Ewald2” and “Srfrce”). However for the subroutines “Ewald1” and “SHAKE” some inter-processor communication is unavoidable and the timing of the program reflects this especially as the number of processors increases. Eventually these algorithms become communication bound, that is, there is no reduction in execution time by using a larger number of processors.

Features in development:

Display modules for use with AVS.

Features for future development:

We hope to begin working soon on algorithms that will facilitate the simulation of “mega-systems” - ie. multi-million particle simulations. In particular this will mean exploiting the “domain decomposition” parallelization strategy and exploring the use of “heirachical multipole” methods for efficient handling of long range (e.g. electrostatic) potentials.

In addition there are several developments that can be added within the Replicated Data strategy, these include:

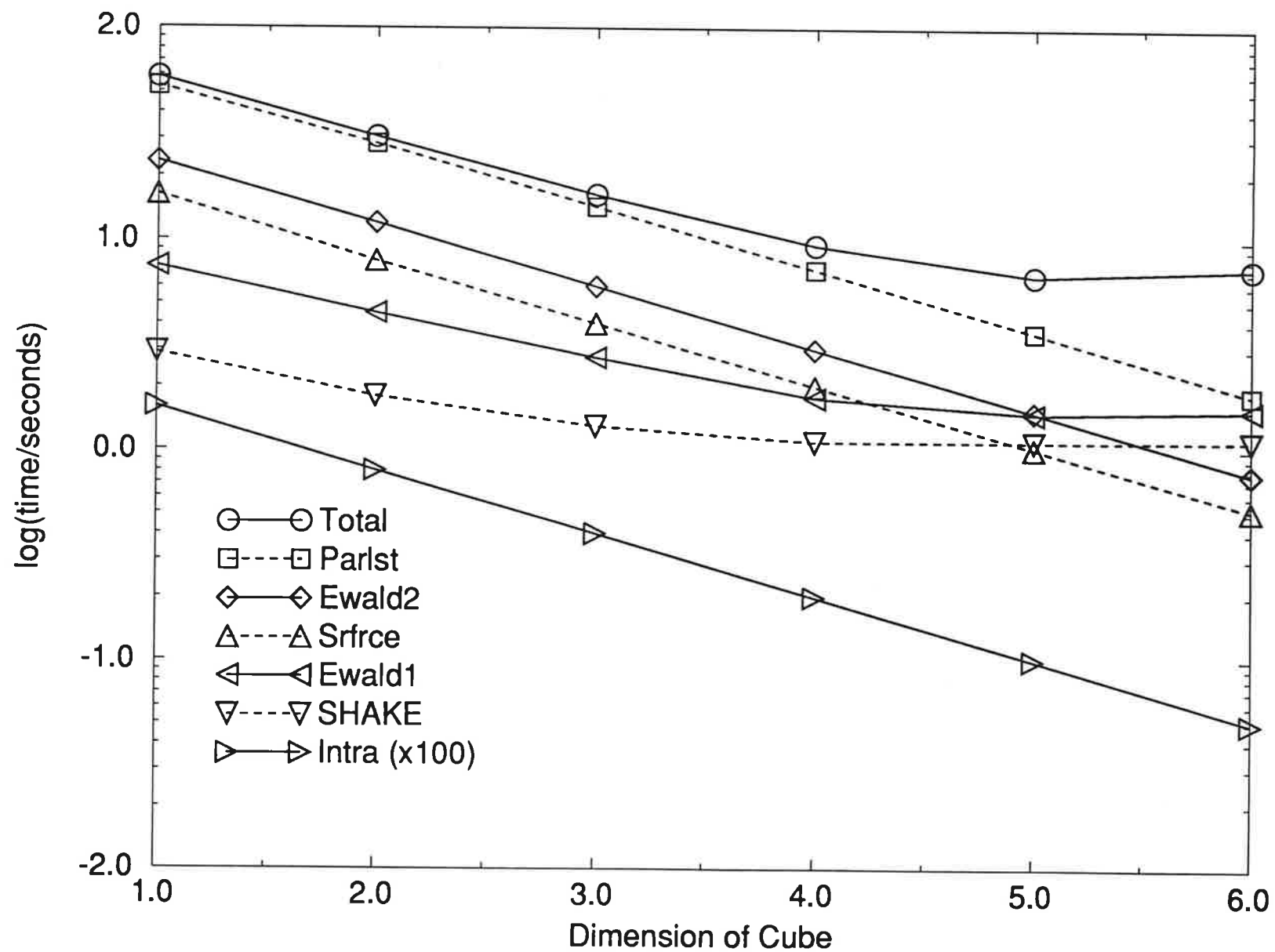
- Algorithms for rigid bodies (and massless sites).
- Algorithms for constant pressure simulations
- Inclusion of anisotropic site potentials (e.g Distributed multipoles).

The first two on this list are straightforward to implement and their implementation is dependent on us finding the time to include them in the package. The last item on the list includes some unresolved problems with handling torques on sites contained in a flexible molecular unit. Note that none of these features will be included in DL-POLY release I.

Availability of the code:

Ultimately the package will be available by anonymous ftp from CCP5. However in the intervening 18 months or so from now the package will be available on restricted release within the U.K. only. Those interested in this limited release should initially contact us at Daresbury for more information.

Algorithm Performance



Measurements and Corrections of Spatial Distortions in Imaging Plate Systems

S.O. Svensson*, A.P. Hammersley*,
A. Thompson[†], A. Gonzalez[‡], T. Ursby*

1 Introduction

An important consideration for X-ray area detectors used in crystallography is spatial distortion. The Imaging Plate (IP) system has been referred to as a system which does not suffer from spatial distortion [1, 2], however we show that this phenomenon can be important.

Several data sets of both protein and small unit cell crystals were collected using 20×25 cm Kodak and Fuji IPs scanned by a Molecular Dynamics (MD) scanner model 400 E. These data could not be properly indexed due to spatial distortion of this scanner. We carefully measured the spatial distortion and developed software to correct for it.

Other characteristics that are of importance for X-ray area detectors for use in detection of diffraction peaks, are the point spread function (PSF), intensity linearity, dynamic range and flat field response. We have measured some of these characteristics along with the spatial distortion, see Table 1 and 2.

2 Measurements

In order to measure the spatial distortion, two nickel grid masks¹ (35×43 cm) were used:

1. A fine grid with $\varnothing$ 200 μm holes separated 2 mm for pixel sizes around 100 μm .
2. A coarse grid with $\varnothing$ 600 μm holes separated 4 mm for pixel sizes around 200 μm .

The holes in the grid were made by electro-forming. The accuracy of the grids was verified by optical measurements (grid step errors < 20 μm).

* ESRF, BP 220, 38043 Grenoble Cedex, France

[†] EMBL, BP 156X, 38042 Grenoble Cedex, France

[‡] SERC, Daresbury Laboratory, Warrington WA4 4AD, UK

¹ Supplied by Attewell Ltd, 4 Southbridge Way, Southall, Middlesex UB2 4BY, UK.

Table 1: Imaging characteristics of the Fuji BAS 2000 scanner and the Molecular Dynamics Phosphor Imager model 400 E.

Detector	Fuji	Molecular Dynamics
Dimensions of IPs (cm)	20 × 40 or 20 × 25	43 × 35 or 20 × 25
Nominal pixel size (μm) ^a	100 × 100 or 200 × 200	88 × 88 or 176 × 176
Measured average pixel size (μm)	99.4 × 102.8 (3.4%) ^b 198.8 × 205.6 (3.4%)	87.0 × 88.2 (1.1%) 176.3 × 175.9 (0.23%)
Intrinsic flat field response	(not measured)	±10%
Corrected flat field response	(not measured)	±0.5%
Dynamic range	10000	close to 100000
RMS spatial distortion x (μm) ^c	133	579
RMS spatial distortion y (μm)	64	256
RMS distortion after corrections x (μm) ^d	105	70
RMS distortion after corrections y (μm)	79	246

^aFor the Fuji, the 200 μm pixel scan is mechanically exactly the same as the 100 μm pixel scan.

^bPixel non-squareness.

^cOnly measured for 200 μm pixel size (Fuji) and 176 μm pixel size (MD).

^dNon-repeatability, measured as differences between two different calibrations. Some of the non-repeatability is due to the grid being placed in a slightly different position.

Table 2: PSF as function of energy for Kodak and Fuji IP [3].

Energy [keV]	Fuji plate		Kodak plate	
	FWHM [μm]	FW @ 0.1 % [μm]	FWHM [μm]	FW @ 0.1 % [μm]
10	140 ± 15	1100 ± 40	160 ± 15	1200 ± 50
20	130 ± 15	1000 ± 40	180 ± 15	1400 ± 50
40	130 ± 15	1100 ± 100	190 ± 15	1600 ± 130

An X-ray tube with a tungsten target running at 30 kV 10 mA was used for exposing the grid pattern onto the IP. The IP was placed 3 m from the tube where the beam measured 20 × 20 cm. The grid was held tightly against the IP by a set of magnets, which together with the large distance from the source, minimised parallax errors. The IP was exposed for 30 s which gave ~ 50000 MD units in the peaks and ~ 1000 MD units in the background. The rest of the IP was weakly exposed by the air scattering but still sufficiently strongly to be used for fitting the peaks.

3 Spatial Distortion Calibration and Correction

The calibration and correction process [4] can be sub-divided into the following operations:

1. Peak Searching: Find the centres of the grid peaks
2. Distortion Definition: Calculate values of the distortion at each measured grid peak
3. Distortion Fitting with Interpolation Function: Find an interpolating function which fits closely (or exactly) to the distortion as measured at the grid peaks
4. Distortion Correction: Estimation of distortion and re-binning

Each operation may be performed using a variety of techniques; the most appropriate technique may vary for different detector systems e.g. the type of interpolating function.

3.1 Peak Search

As a calibration grid may contain thousands of grid holes, the method used to perform the peak search should be reasonably efficient. The peak search may be further sub-divided into the method used to find the best centre of each peak and the method used to search the image for further peaks. We propose cross-correlation of the data with the known peak shape to find the centre of an individual peak. This makes a small number of constraints on the data and provides a method which is optimal both in accuracy and in computational efficiency (for a given accuracy). The assumptions of this method are that the peaks are not overlapped and that the individual peaks are on a plane background.

For a known peak profile, cross-correlation of the measured data with the known profile is optimum in finding the peak centre [5]. (The centre is at the position where the cross-correlation value is a maximum). If the profile is wrong, but both the true profile and the assumed profile are circularly symmetric, then on average, the centre will be correct, but the variance in position will be greater than for the ideal profile. An arbitrary sub-pixel accuracy may be obtained by sub-dividing each data pixel into an arbitrary number of sub-pixels, and calculating the profile function at the sub-pixel resolution.

To measure accurately and define the spatial distortion, the Nyquist criterion says that it is necessary to sample (measure a peak position) the distortion function at twice the highest spatial frequency in the function. If this criterion is fulfilled, or nearly fulfilled, the distortion function will be “smooth” and the position of each peak may be estimated with reasonable accuracy from the positions of previously measured peaks. By estimating the positions of all the peaks from previous peaks, starting from an initial three peaks, much computation may be avoided by only calculating the cross-correlation at a few positions.

3.2 Distortion Definition

Once the centres of all recorded peaks have been found, the values of the distortion at these positions may be calculated. However, this process is complicated by rotation of the calibration grid with respect to the detector raster. Even with careful alignment the calibration grid is likely to be rotated with respect to the detector. Unfortunately, this rotation will also be distorted. The rotation angle is estimated from the averaged rotation angle calculated from pairs of peaks at the ends of all rows and columns to minimise the effect of distortion. From the calculated rotation angle, the known grid peak to peak spacing, and the required ideal pixel dimensions, the values of the distortion function for the X and Y-directions may be calculated.

3.3 Distortion Fitting with Interpolating Function

To be able to estimate the distortion function at all pixel positions for correction of spatially distorted images, it is necessary to use an interpolating function. We chose bi-cubic spline functions defined on an irregular grid. By using spline functions, the distortion may be fitted with an arbitrary shape and efficiently calculated since only a small number of terms are necessary. Defining the spline knots on a grid allows appropriate knot coefficients to be found efficiently. An arbitrary RMS fit value may be specified. If the errors in estimated distortion values are independent and normally distributed, then using the minimum RMS discrepancy is optimum. The routine finds the smallest number of knot points necessary to provide the desired fit. By choosing splines with far fewer knot points than measured peak positions, the spline can be kept constrained by the data, and confidence may be given to the interpolated values between peak positions.

3.4 Distortion Correction

Once the spatial distortion interpolating function has been defined the correction process is relatively simple. The interpolation function allows the value of the distortion for all points on the distorted images to be estimated. Thus, the position of distorted pixels on a corrected pixel image may be calculated. The intensity in each input pixel may be re-distributed to the corrected pixels in proportion to covered area. Whilst the input pixels may be translated to arbitrary shapes on the corrected image, they may be approximated by rectangles. Other techniques exist to improve this approximation. This aspect of the calibration and correction process is adequately discussed by Stanton [6].

4 Results

4.1 Calibration measurements

The distortion is categorised in two types: pixel non-squareness, and local distortion. Table 1 shows pixel dimensions calculated from averaged distances measured between the peaks at the extremes of the calibration grid. The local distortions for the MD scanner are shown in Figures 1 and 2 for the X (laser scanning) and Y (mechanical) directions, and the non-repeatability is given in Table 1.

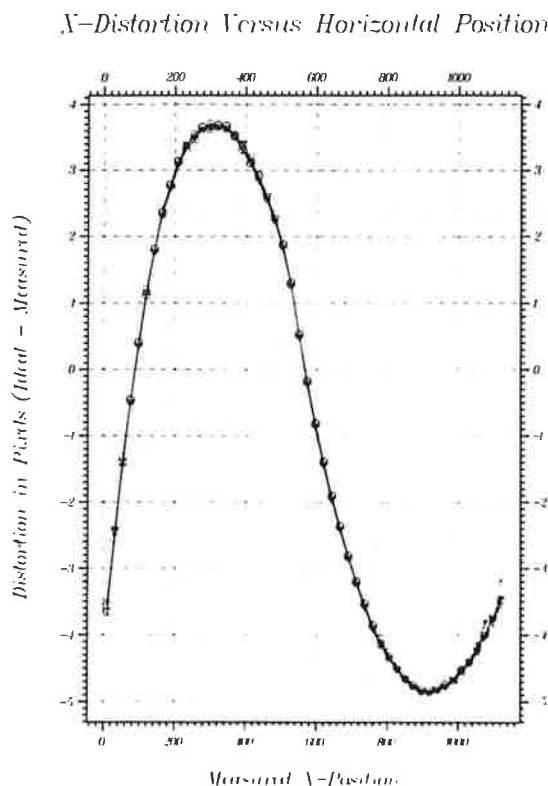


Fig 1: X-Distortion Plotted for 5 Consecutive Columns

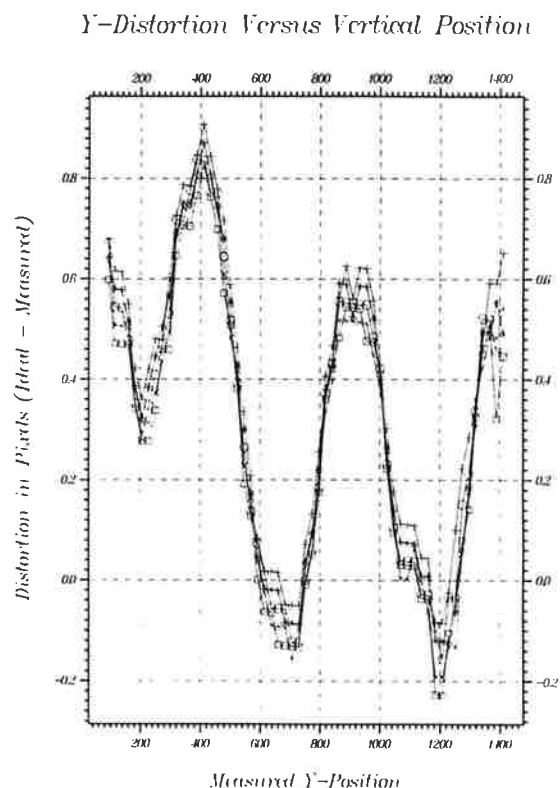


Fig 2: Y-Distortion Plotted for 5 Consecutive Columns

4.2 Application to data processing

The measurement with the grid pattern only reveals the spatial distortion at some discrete number of points, i.e. where the holes in the grid are located. The distortion is not known between two successive holes in the grid pattern. The true test of the method is to analyse real diffraction patterns and see how well the predicted peaks fit to the recorded peaks.

The distortion corrections were applied to data collected on IPs at both the ESRF and Daresbury synchrotrons. The IPs were scanned with MD scanners and the corrections

Table 3: Comparison of reduced χ^2 and RMS values for uncorrected and corrected data.

		Uncorrected	Corrected
Lysozyme	χ^2 x	27.9	2.3
	χ^2 y	4.9	1.6
Magnesium	χ^2 x	16.2	1.0
Formate	χ^2 y	4.7	3.3
Concanavalin A	RMS	95 μm	45 μm

were applied using **FIT2D** [7]. Data were collected as follows:

1. SRS Station 7.2, monochromatic ($\lambda = 1.488$ Å), from tetragonal lysozyme (P4₃2₁2, a = b = 79.1 Å, c = 38.0 Å)
2. ESRF Beamline 2, monochromatic ($\lambda = 0.2$ Å), from a small molecule (Magnesium Formate, P2₁/c, a = 8.64 Å, b = 7.15 Å, c = 9.38 Å, $\beta = 98.05$)²
3. ESRF Beamline 2, Laue data from concanavalin - A, $\lambda_{max} = 1.8$ Å, $\lambda_{min} = 0.3$ Å, (I222, a=89.9 Å, b= 87.2 Å, c = 63.1 Å)

A summary of the positional errors and their effect on the R_{sym} (intensity) is given in Tables 3 and 4. The improved fit of the data is clear from the improved χ^2 values in the case of the monochromatic data sets processed using **DENZO**, and from the improved RMS fit from **INTLAUE**, in the case of the Laue data. The ‘mis-positioning’ of the integration box is clearly illustrated in the **AGROVATA** output for the lysozyme data (corrected and uncorrected) at high resolution. (Note that at the edge of the image plate the deviation of the ‘real’ from the ‘measured’ spot position is largest.) The number of suitable symmetry related reflections integrated has dramatically improved, and the agreement between these data has changed from ‘no agreement’ to ‘reasonable agreement’. The average intensity drops more rapidly and the sigma rises dramatically as the integration box misses the spot position in the uncorrected data.

The more reproducible correction in the laser scan direction of the ESRF data is demonstrated by the large improvement of fit in this direction as compared to the weak improvement in the other direction.

²No merging statistics are presented for this data, since corrections for uniformity and linearity of response of the image plate are currently in progress.

Table 4: Processing statistics (**AGROVATA**) from Lysozyme Data.

Corrected						
Dmin (Å)	R fac	Ave. I	Sigma I	I/Sigma I	Nmeas	Nref
6.11	3.5%	18977	1076	17.6	151	62
4.36	4.2	15638	1079	14.5	293	128
3.58	5.3	20958	1748	12	272	124
3.1	5.5	9905	918	10.8	168	84
2.78	7.5	5370	611	8.8	123	61
2.54	4.5	4138	261	15.9	80	40
2.35	9.9	2306	362	6.4	90	45
2.2	9.2	2560	313	8.2	84	42
Uncorrected						
6.11	3.3	16554	1126	14.7	153	63
4.36	4.5	15180	1310	11.6	294	128
3.58	5.9	18294	1831	10	299	135
3.1	8.2	9079	2148	4.2	196	97
2.78	10.0	4752	913	5.2	155	77
2.54	10.1	3772	823	4.6	98	49
2.35	111.2	427	1545	0.3	26	13
2.2	193.	48	149	0.3	18	9

5 Conclusion

The Molecular Dynamic model 400 E scanner suffers from important spatial distortion, leading to differences up to 1 mm between recorded and predicted spots. This can make indexing very difficult, or impossible, and also the accuracy of the integrated intensities deteriorates, especially at high resolution.

The spatial distortion in the X direction is largely reproducible and can be corrected. In the Y direction the distortion is smaller but not reproducible. The calibration and correction procedure reduces spatial distortion allowing further data processing (**MOS-FLM**, **DENZO** etc.).

The distortion in the Y direction is not reproducible due to the wire mechanism pulling the carriage on which the IP is held. With minor modification to this mechanism it should be possible to make the Y distortion reproducible and smaller.

The Fuji system BAS 2000 showed smaller spatial distortion but it is not reproducible. However this is of such a nature that we believe the functions built in to correct for bulge and twist of X-ray films in **MOSFLM** or **DENZO** could partially correct for it.

6 Acknowledgements

The authors would like to thank for their contribution and support: the ESRF diffraction group and especially Å. Kvick, D. Bourgeois, H. Graafsma and D. Häusermann; C. Nave, C. Hall and the staff of Station 7.2 at SERC, Daresbury; J. Helliwell and E. Snell of the University of Manchester (concanavalin crystals); the ESRF detector group; Z. Otwinowski (**DENZO**), A. Leslie (**MOSFLM**) and the contributors to the Daresbury Laboratory Laue Software Suite and the CCP4 Program Suite.

References

- [1] M. Ito and Y. Amemiya, *Nucl. Inst. Methods*, **A310**, 369-372 (1991)
- [2] Y. Amemiya, *Synchrotron Radiation News*, Vol. **3**, No. **2**, 21-26 (1990)
- [3] D. Bourgeois, J.P. Moy, S.O. Svensson and Å. Kvick, (to be published)
- [4] A.P. Hammersley et al., (to be published)
- [5] D.J. Thomas, *J. de Physique*, Supplément au No **8**, Tome **47** (1986)
- [6] M. Stanton, W.C. Philips, Y. Li, and K. Kalata, *J. Appl. Cryst.*, **25**, 549-558 (1992)
- [7] A.P. Hammersley, *ESRF Internal Report*, FIT2D Reference Manual (1993)

The peptide-binding protein OppA from *Salmonella typhimurium*

Jeremy Tame, Garib Murshudov, Eleanor Dodson and Anthony Wilkinson

Department of Chemistry

York University

York YO1 5DD

Bacterial cells can utilise a wide variety of nutrients with which they may come into contact, and have evolved a large number of transport systems to facilitate their uptake. In the case of Gram negative bacteria such as *E. coli* the cell surface consists of three layers, two membranes and the peptidoglycan cell wall which lies between these in the periplasmic space. The outer membrane contains porins which allow small solutes to cross it freely, and the cell wall is also permeable to such molecules. The inner membrane however is impermeable to solutes unless a specific transport system is present. These systems fall into two categories depending on their sensitivity to osmotic shock. This procedure, in which the cells are rapidly transferred from a concentrated sucrose solution to water very rapidly, disrupts the outer membrane and releases proteins normally found in the periplasmic space. Transport systems which depend on such protein are "shock-sensitive" or "periplasmic" transport systems. The soluble, periplasmic component is a binding protein with a high affinity for the molecule to be transported, usually with a K_D of 0.1 - 1.0 μM . Several binding proteins have been solved by Quioco's group in Texas, and their structures are reviewed by Quioco in (1). All are bilobate molecules, hinged at the middle with the ligand binding site between the two domains which close over the bound molecule. There are usually 3 but occasionally 4 membrane-associated proteins in these transport systems which take ligand from the binding protein and transfer it to the cytoplasm with concomitant hydrolysis of ATP. The tight binding between the ligand and the binding protein allows these systems to work at very low substrate concentrations and against a large concentration gradient. The majority of binding proteins are therefore quite specific, although not all of them have unique substrates. Thus leucine is transported both by leucine binding protein and leucine-isoleucine-valine binding protein. Peptides are important nutrients for bacteria and three transport systems are dedicated to them. The oligopeptide and dipeptide permeases (opp and dpp) are shock-sensitive transport systems which carry peptides of widely varying sequence into the cell. Dipeptide binding protein has a marked preference for dipeptides, but OppA, the oligopeptide binding protein, will bind peptides from 2 to 5 residues in length. Both of the binding proteins are unusual in their lack of specificity. Relatively little work has been done to characterise the proteins biochemically,

but it has been demonstrated that acetylation of the N-terminus prevents peptide binding to OppA. OppA will also bind peptides containing D amino acids, though with lower affinity. This may reflect its role in recycling cell wall protein which is constantly lost to the medium. OppA and DppA are both unusually large binding proteins, having molecular weights of about 58 kD, compared to the 30 - 40 kD of most other binding proteins.

OppA from *Salmonella typhimurium* was first cloned in the lab of Chris Higgins, and was crystallised in York in 1988 (2). These $P2_12_12_1$ crystals grew very unreliably and were very fragile, so despite reasonable native data no progress was made with the structure for several years. The breakthrough came when it was discovered that in the presence of uranyl acetate the protein gave large, robust crystals in space group $P2_12_12$ which diffracted to 2.1 Å. The anomalous Patterson map showed a very strong peak at a general position, indicating that a uranium atom had bound to the protein. Attempts to remove the uranium (with EDTA for example) were unsuccessful as the crystals (called "oppau") proved sensitive to any soaking procedure with or without uranium. The presence of an ordered uranium in the crystals made them ideal for multi-wavelength phasing however since uranium shows very strong anomalous effects which change markedly around the L_{III} edge at about 0.72 Å. Four incomplete datasets were collected in limited time at PX 9.5 in August 1991 in collaboration with Ian Glover and Richard Denny of Keele University. These proved to be insufficient for solving the structure due to the very large missing region, and it was nearly two years before time became available for another experiment.

The oppau crystals are grown from protein containing a co-purified peptide. Periplasmic binding proteins generally bind ligands so tightly that drastic conditions such as 1-2 M guanidine hydrochloride are required to prepare the ligand-free form. In the case of OppA it was discovered that a mild acid wash can remove the ligand, and pure peptides added to the protein to form a homogeneous protein-peptide complex. Trilysine was chosen as the first peptide to work with as it causes a large increase in fluorescence of the protein on binding. The K3 complex crystallised under similar conditions to the oppau crystals, but in space group $P2_12_12_1$. Since all the cell dimensions are larger these crystals are called "oppal". These crystals are extremely well-ordered and also freeze very well with little increase in mosaicity. 1.8 Å data have been collected on the Xentronics at York, and the crystals diffract to at least 1.4 Å at PX 9.6.

Rather than one uranium site (as in oppau crystals) the oppal crystals appeared from the anomalous Patterson to have six. More usefully, several of these could be removed to some extent by soaking in lead chloride or potassium platinum VI chloride. Five uranium sites refined satisfactorily and the phases derived allowed a reasonable map to be calculated, which was then averaged with the oppau map. The structure was built into the averaged map using O, and is currently being refined in both oppau and oppal forms. The protein shows

considerable differences from the previous binding protein structures. It is a three domain protein, each domain containing a β sheet and in contact with the ligand. In oppau the ligand density will reflect both any side-chain preferences of the protein and the relative abundance of different peptides in the growth medium when the cells were harvested. In K3-oppal however the ligand is clearly defined. The peptide is completely buried and well-ordered in an extended conformation, forming sheet-like interactions with the edge strand of a β sheet. Pockets lined with a variety of residues accept the ligand side-chains, but the precise interactions will only be known once refinement is complete.

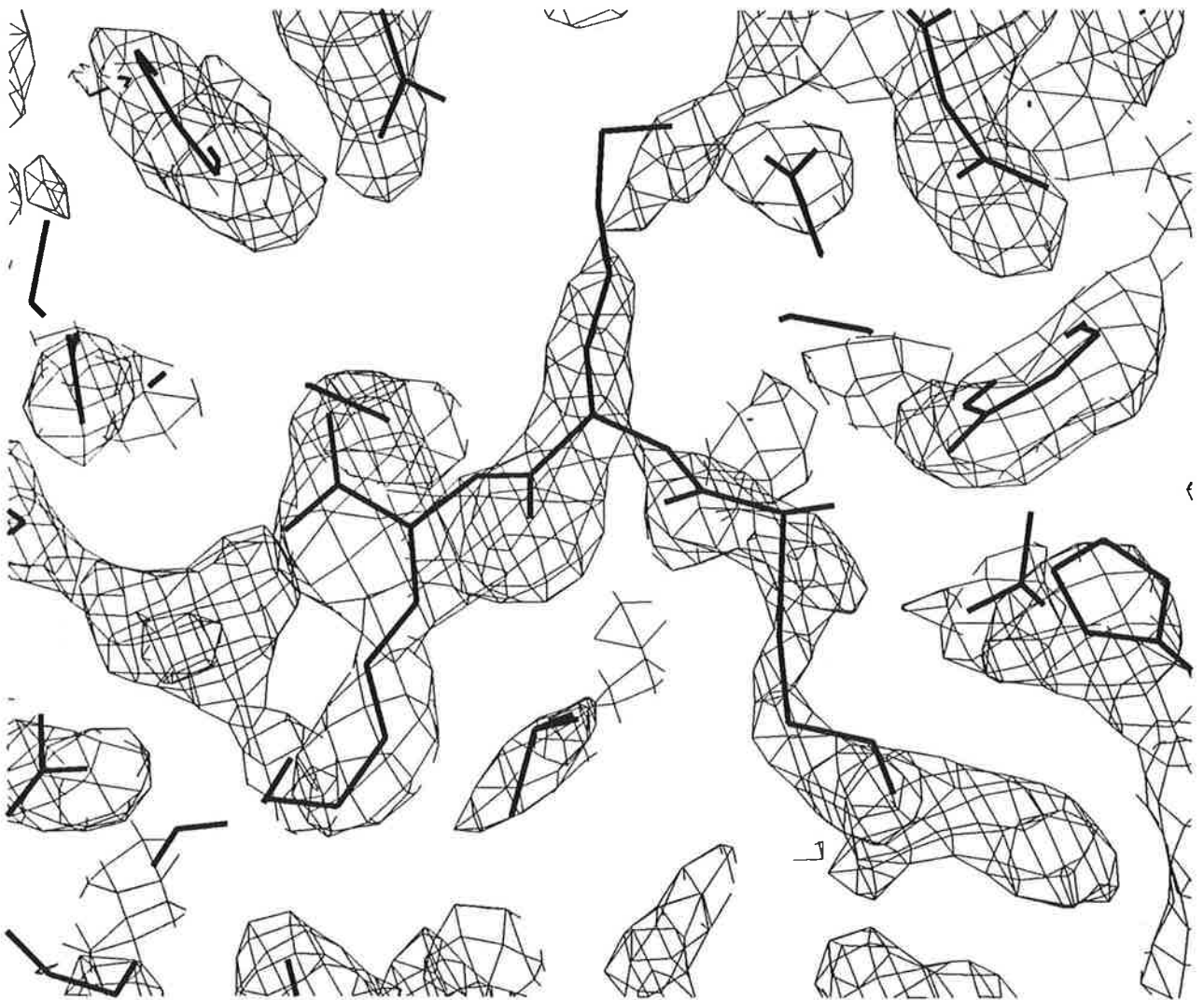
In July 1993 multiwavelength data were collected on the oppal crystals at PX 9.5, again in collaboration with the Keele group. Phases have been calculated from the four datasets using MLPHARE in York and MADLSQ in Keele. Preliminary results suggest that the MLPHARE map is good enough to solve the structure independently of other phase information, but a complete analysis awaits the (imminent!) final refinement of the structure. The experimental details are discussed in the accompanying article.

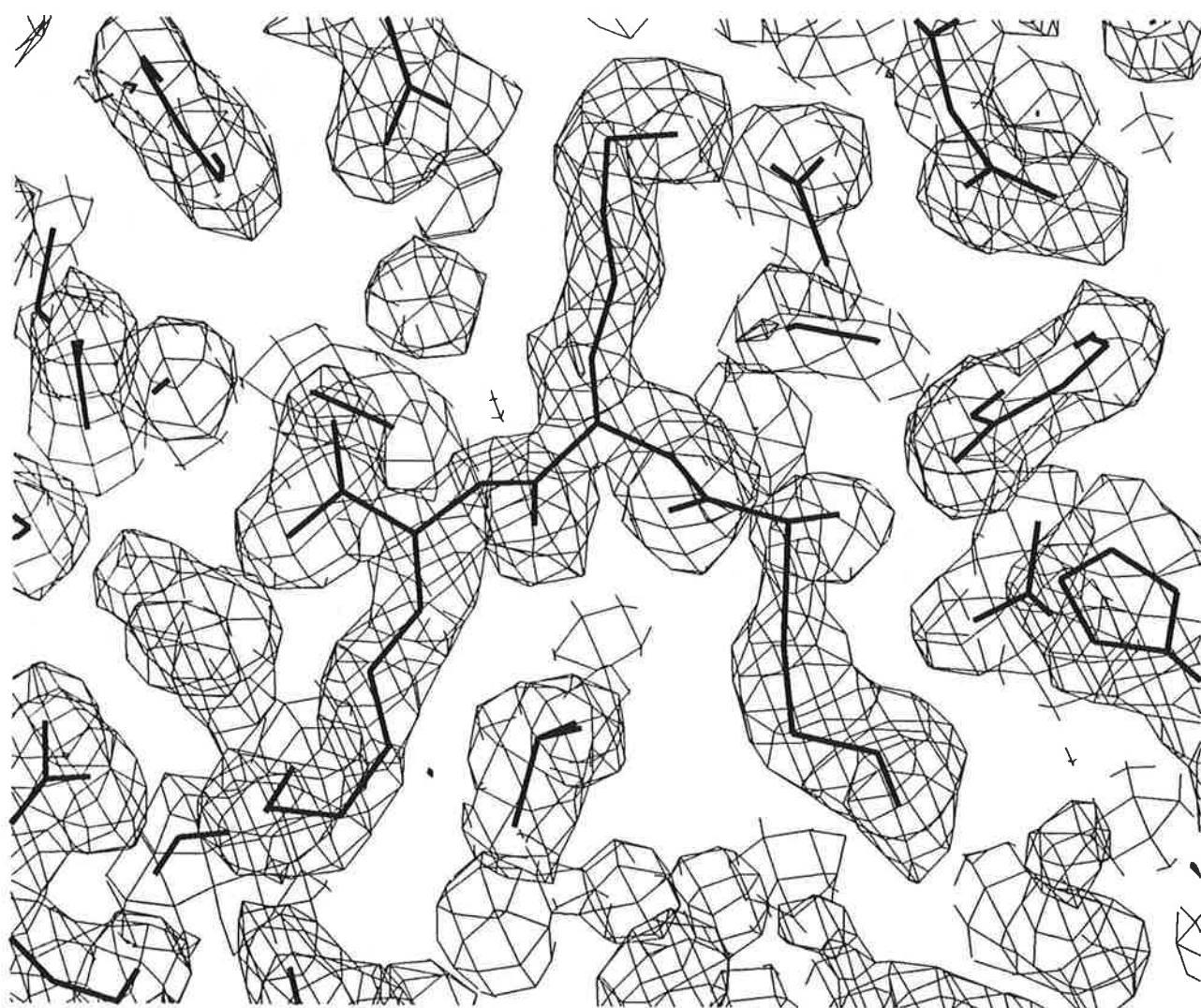
References

1. Atomic structures and function of periplasmic receptors for active transport and chemotaxis. Quiocho, F.A. *Current Opinion in Structural Biology* 1: 922-933 (1991)
2. Crystallisation of the periplasmic oligo-peptide binding protein of *Salmonella typhimurium*. Tolley, S.P., Derewenda, Z., Hyde, S.C., Higgins, C.H. and Wilkinson, A.J. *J. Mol. Biol.* 204: 493-494 (1988)

Figure Legends

- 1) The current coordinates showing the trilycine ligand in the oppal crystal form with the MAD map (MLPHARE phases) superimposed at a level of 1 sigma.
- 2) The same view of the ligand in oppal with the current 2Fo-Fc map superimposed, showing the ligand side-chains are all well ordered.





MAD data collection and phase determination of OppA.

Ian Glover, Richard Denny and N. Donatus Nguti

Department of Physics, Keele University,

Sean McSweeney, Andrew Thompson

SERC Daresbury Laboratory,

Eleanor Dodson, Tony Wilkinson and Jeremy Tame

Department of Chemistry, York University.

OppA is an oligopeptide binding protein isolated from the periplasmic space in bacteria. It forms part of the oligopeptide permease system. It exhibits fairly high binding for its substrate but low specificity. The molecular weight is 58808, indicating that the molecule is approximately twice the size of other periplasmic transport proteins.

OppA from *Salmonella typhimurium* has been cloned and crystallised. Initial crystallisation yielded fragile crystals with poor reproducibility hampering structure determination attempts. In the presence of uranium, however, large, robust crystals were produced giving diffraction to at least 2.1 Å. Anomalous Patterson maps using data collected using CuK α radiation showed strong U peaks but the crystals proved very sensitive to any heavy atom soaking experiments with or without uranium. The presence of well ordered U in the crystal and difficulties in producing other heavy atom derivatives lead to the attempts at multiwavelength anomalous dispersion phasing methods, collecting data about the U LIII edge (0.720 Å), using synchrotron radiation.

Two attempts at the MAD experiment were made, separated by about 2 years. The first experiment yielded only partial datasets but confirmed the potential of the method. The second, succesful attempt was carried out in July 1993.

All data collection was carried out on station 9.5 (Thompson *et al*, 1992), at the Daresbury SRS using radiation from a superconducting wiggler in the storage ring operated at 2 GeV with circulating currents of approximately 250 - 150 mA. Monochromatic X-rays were produced by a channel cut Si(111) monochromator

with cooling on the first crystal downstream from a Pt coated toroidal mirror providing point focussed radiation at the sample. One of the most important steps in MAD is the selection of the incident X-ray wavelengths or energies for data collection. Initial calibration of X-ray energy was carried out with reference to the points of inflection in absorption spectra of Zn foils and uranium acetate powder samples. The wavelengths at which data collection was undertaken were determined from the recorded XANES spectra from a single crystal of OppA recorded using a single wire proportional counter mounted on the goniostat (fig. 1). All settings were determined from monochromator positions yielding nominal wavelengths of 0.7207Å (f' minimum), 0.7199Å (white line) and remote wavelengths of 0.688 and 0.90Å, chosen to give large changes in f' values. The XANES spectra were recorded at the start of each machine fill and monochromator positions updated.

Diffraction data were recorded using a MAR Research Image Plate detector system with a crystal to detector distance of 175mm giving a nominal resolution limit of 2.3Å and data collected using protocols designed to minimise systematic errors, recording Bijvoet mates on the same image and collecting all data contributing to a phase determination as close together in time as possible. The crystals, space group $P2_12_12_1$ $a=110.42\text{\AA}$, $b=77.04\text{\AA}$, $c=71.38\text{\AA}$, were mounted with b^* parallel to the rotation axis and data collected in single ϕ intervals recorded at each of three wavelengths (0.720, 0.719 and 0.68Å) before proceeding to the next ϕ interval. Two crystals were used to record a overlapping total of 97° followed by collection of a dataset at 0.9Å wavelength. All data were collected at room temperature over a period of 5 days.

The collected data were processed using the MOSFLM suite and corrected for L_p using the program ABSCALE and merging statistics calculated using ROTAVATA/AGROVATA (all programs from CCP4 suite). Identification of heavy atom sites was carried out initially using Patterson syntheses. Further sites were identified using direct methods. A total of 6 sites were identified and phase determination carried out using maximum likelihood methods.

Dataset	Wavelength	Rmerge	f'	f'' (calculated)
f' minimum	0.7207Å	7.8%	-21.7	9.3
f'' maximum	0.7199Å	8.1%	-15.2	12.2
remote	0.688Å	8.0%	-7.8	9.2
remote	0.90Å	7.8%	-5.5	5.8

(f', f'' calculated from Cromer (1983) using a bandpass of 5eV)

This approach treats the data collected at different wavelengths as separate sources of information in a conventional multiple isomorphous replacement method. One data set is treated as a native dataset, with anomalous scattering, and data at other wavelengths are treated as derivatives. Dispersive differences between data collected at different wavelengths give rise to the 'isomorphous' differences and f'' the anomalous differences.

Estimates of the positions of the anomalously scattering atoms from Patterson or direct methods are used in iterative cycles of maximum likelihood phase refinement using data recorded at each wavelength as a heavy atom derivative, as implemented in the CCP4 program MLPHARE. In this procedure parameters refined include the positional and thermal parameters, x, y, z and B , along with occupancies split into real (dispersive) and anomalous (absorptive) occupancies for each of the sites of the anomalous scatterer. Phase probability is then integrated at set intervals, using the refined heavy atom contributions.

The program MLPHARE yielded satisfactory refinement of the heavy atom parameters. The refinement was carried out in two stages, refining first the real occupancy factors against the centric data and hence dispersive differences only. This was followed by cycles of refinement of anomalous and real occupancies against all data. In this procedure one site was rejected due to high thermal parameters and excessive coordinate shifts and sd 's. The overall figure of merit was 58%. An electron density map was calculated using the phases direct from MLPHARE and also after solvent flattening and density modification using the program suite SQUASH (from the York program suite).

The electron density map calculated from the MAD(MLPHARE) phases was readily interpretable in terms of the known amino acid sequence of OppA. The protein main and side chain positions could be readily traced and the liganded tri-lysine identified from the MAD electron density. Further structural details may be found in the accompanying paper. The calculated correlation of the MAD electron density with the model at the current stage of refinement is 0.502 (0.58 for the modified phases).

The multiple wavelength method combined with maximum likelihood phasing methods has successfully produced a clearly interpretable electron density map for a large multidomain protein. The use of dedicated SR sources with high brilliance in the hard X-ray region has great promise in the determination, by MAD, of structures of large molecules using a single isomorphous derivative.

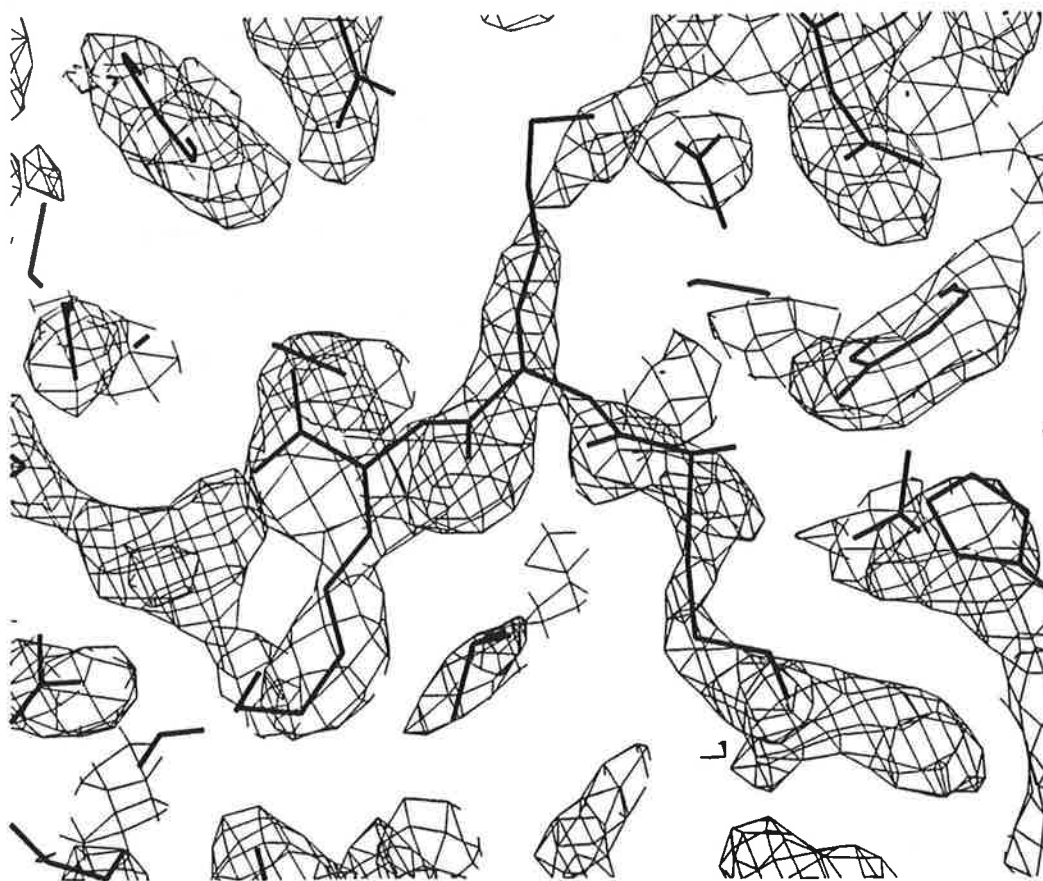
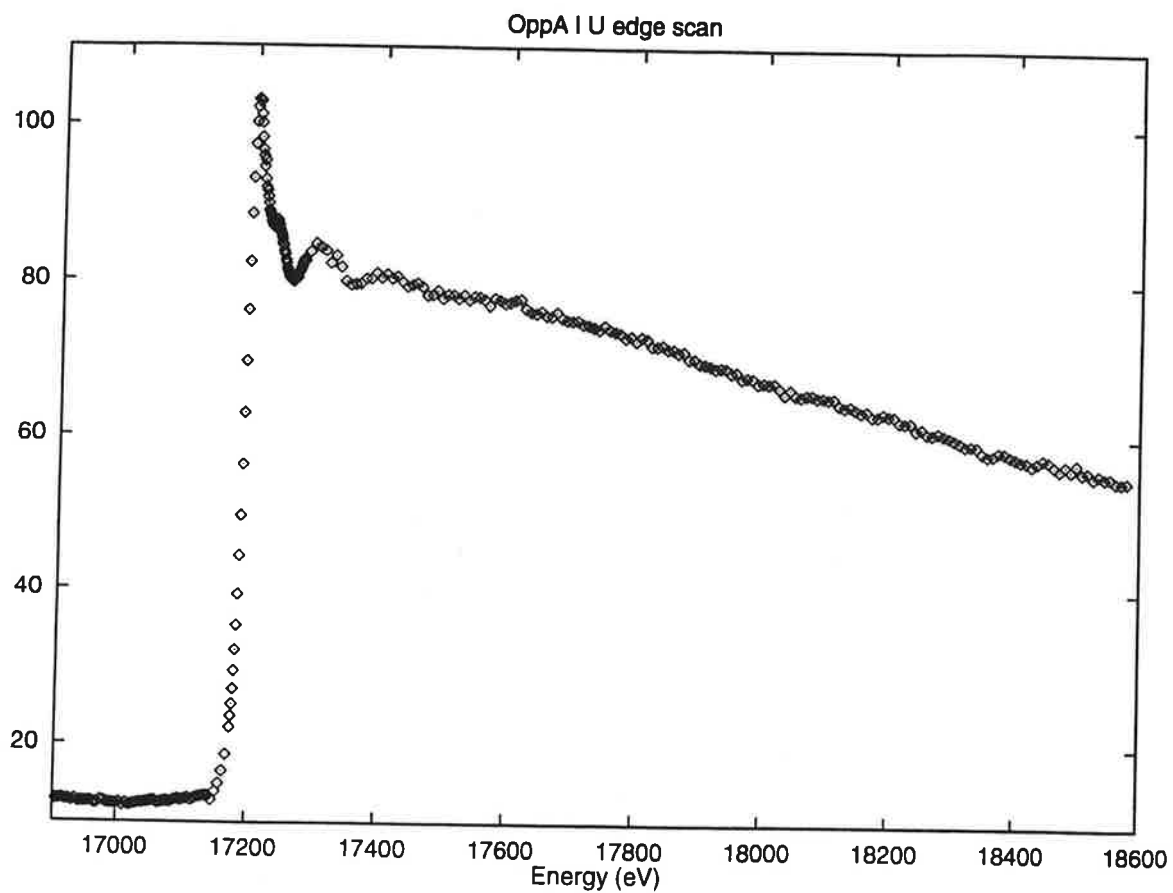
References.

A.W. Thompson *et al.* (1992) Rev.Sci.Instrum. **63** 1062 - 1064

Cromer, D.T. (1983) J.Appl.Cryst. **16** 437-441

Figure Legends.

- 1) The XANES spectrum recorded from a single crystal of Oppa, used for wavelength selection
- 2) The electron density, contoured at 1 σ , calculated from the MAD(MLPHARE) phases in the region of the bound tri-lysine ligand. The coordinates from the current model are superimposed.



CCP13: Software development for fibre diffraction

CCP13 began in 1992, through the efforts of Dr. John Squire (Imperial College), in order to coordinate and extend the development and use of software applicable to the common ground of those using fibre diffraction to investigate various systems. Two successful workshops have been held, organized jointly by the chairman, John Squire and the secretary, Geoff Mant (Daresbury Laboratory), which have attracted distinguished international representation. At this early stage, the need to gauge the requirements of the community as well as to examine existing programs has been an important feature of the workshops. There has also been a need to highlight convergence of interest with groups studying synthetic polymers using both wide and small-angle scattering. The first newsletter was published in December, 1992 with the intention of publishing subsequent editions annually.

As well as the need to cut down on duplication of effort, another motivation for CCP13 is to make the analysis of fibre diffraction patterns more rigorously quantitative. This will allow more meaningful comparisons to be made between datasets and model structures. Currently, the structure of the CCP13 suite is envisaged as containing firstly, programs capable of reading a flexible input image file format for initial processing and analysis of raw data. This requires the development of a convenient and portable graphical interface. Secondly, programs for background subtraction and peak fitting which are flexible enough to cope with the wide variety of diffraction patterns which can occur from fibrous specimens will undergo continuing development. In particular, there are difficulties in dealing with patterns which exhibit both Bragg-sampled and continuous intensity. Up until now, this is where the majority of the work of the CCP13 PDRA, Richard Denny, has been focussed and a paper is in preparation for publication in J. Appl. Cryst. Thirdly, either modelling of structural parameters against intensity data is required or a route made into CCP4 in order to utilize the existing Fourier transform routines and other programs for crystallographic analysis.

The existing CCP13 programs are available to Non-Crystalline Diffraction users at Daresbury and to others on request. The routines are being refined and tested using data from specimens such as DNA, various types of muscle and synthetic polymers.

Clearly, there is much work for CCP13 to do and with the obvious overlap in the techniques used to analyze reduced diffraction data, the project can only gain

by a strong liaison with CCP4. There has already been a successful interaction in which a CCP13 routine has been used to determine the background in diffraction images from single crystals of lysozyme collected at Daresbury. There may even be protein crystallographers who, in previous incarnations, dabbled in fibre diffraction. If so, any methods or expertise they have to offer would be greatly appreciated by those involved in CCP13.

Richard Denny

Standard Stereochemical Dictionaries for Protein Structure Refinement and Model Building

John P. Priestle

Dept. of Biotechnology, Ciba-Geigy, Ltd., CH-4002 Basel, Switzerland

The purpose of crystallographic least-squares refinement is to get the best possible structure from the data that have been collected. This is done by minimizing the sum of the squares of the differences between the observed and calculated structure factors

$$\sum_{hkl} wt(F_{o_{hkl}} - F_{c_{hkl}})^2$$

Because not all observations are equally good (accurate), one should weight them according to the confidence one has in them (although in practice this is almost never done in macromolecular crystallography). As we are all painfully aware, a macromolecular crystal seldom produces enough diffraction data to refine a structure in a proper and robust fashion. To compensate for this lack of data, or more accurately, to improve the poor observations to parameters refined ratio, stereochemical restraints are included as extra "observations" so that the term actually minimized is

$$\sum_{hkl} wt(F_{o_{hkl}} - F_{c_{hkl}})^2 + \sum wt(Q_m - Q_i)^2$$

where Q_m and Q_i are the model and ideal values of some stereochemical quantity, summed over all such quantities, again weighted according to how well we know (trust?) the ideal value.

If one were to make a nearly infinite number of measurements of something, and if those measurements were totally free of systematic error and bias, then one would obtain a normal distribution of values. The best estimate of the "true" value would be the mean and the proper weight associated with that value would be the reciprocal of the variance ($1/\sigma^2$). The types of stereochemical restraints used in protein refinement range from "hard" (well-characterized) terms such as bond distances and angles, through "soft" ones, such as torsion angle preferences, to the poorly described (and therefore often excluded), such as charge interactions. To properly include a stereochemical restraint, then, one needs to know two parameters: its ideal value and its variance. Stereochemical idealization is similar, but excludes the X-ray term and is most often used during model building to repair gross distortions introduced by manual corrections where atoms are moved by methods other than torsion angle rotations, e.g. fragment rotation/translation and free atom translation.

There are many different refinement programs available to the protein crystallographer. Among the currently more popular programs are EREF (Jack and Levitt, 1978), PROTIN/PROLSQ (Hendrickson and Konnert, 1980), TNT (Tronrud et al., 1987), RESTRAIN (Dreissen et al., 1989), and the molecular dynamics programs XPLOR (Brünger et al., 1986) and GROMOS (van Gunsteren and Berendsen, 1987). The most popular model building programs are FRODO (Jones, 1978) and its many derivatives and O (Jones et al., 1991). All these programs contain a dictionary of ideal stereochemical parameters for the amino acids and often a few other molecules found in protein crystal investigations (water molecules, sulfate ions, heme groups, etc.), although the degree to which one can understand how these dictionaries function and the ease of including new molecular entities varies considerably from program to program. Weights for the various stereochemical restraints are either part of

the dictionary itself or part of the input for running the program and they can be given as the standard deviation associated with a measurement or as a force constant.

Since these parameters describe the same systems (amino acids), one might naively assume that their values are quite similar. In fact, they are not. Fig. 1 shows an analysis of the bond lengths associated with the side chain of tryptophan from the programs we have in our laboratory:

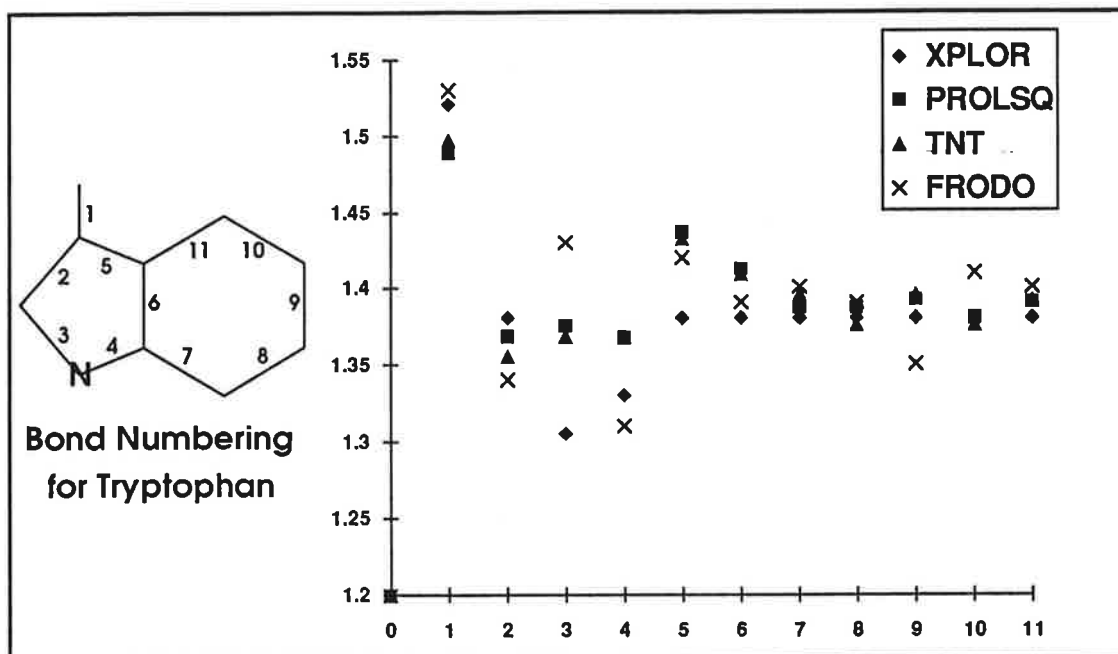


Fig.1 Analysis of ideal bond lengths for the side chain of tryptophan from various protein refinement programs. The bond number runs along the x-axis, its "ideal" length in Ångströms along the y-axis.

The variance between the various dictionaries is much larger than the supposed precision of the values ($\sigma \sim 0.02\text{Å}$). Laskowski et al. (1993) examined this phenomena and asked whether these differences left an imprint on the final, fully refined structure. By a rather crude analysis of bond lengths alone, they were able to correctly identify the refinement package used on a particular protein 95% of the time(!), implying that the program used, or more precisely the standard stereochemical dictionary used, does indeed leave a strong bias on the final structure. What this also suggests is that if you go from one program to another, e.g. for model building or correction, one program's idealization is another program's distortion. Clearly, a set of standard stereochemical parameters that is consistent across all programs is needed.

This need was recognized by Engh and Huber (1991), who systematically investigated some 80'000 small molecule crystal structures in the Cambridge Structural Database to get a set of best bond distance and angle parameters, and almost equally important, an accurate measure of their variance. This they distributed as parameters for XPLOR (topology file "tophcsdx.pro" and parameter file "parhcsdx.pro"). Two test cases using these new parameters demonstrated a small decrease in the R-factor (0.1-0.3%) and a decrease in the energy of the system, in spite of the fact that the new parameters have much larger force constants ($\sim 3\times$ larger for bond lengths and $\sim 7\times$ larger for bond angles).

I have recently rewritten the stereochemical dictionaries for the other refinement and model building programs in our laboratory to be consistent with the Engh &

Huber XPLOR parameters. The programs for which updated dictionaries exist are PROTEIN/PROLSQ (as implemented in the CCP4 program package suite), TNT, XPLOR, FRODO and O. The root mean square deviations between the various "ideal" amino acid bond lengths have been reduced from 0.031 Å to 0.003 Å for bond distances and from about 1.9° to 0.3° for bond angles. Perfect agreement is not possible because of the different ways some groups are handled by the different programs. For example, some programs treat the main chain of all amino acids as being identical, while others allow for the subtle differences in e.g., proline and glycine. Likewise, how the amino and carboxyl terminal groups are handled varies from program to program. In any case, a significant improvement in the agreement between the various programs has been effected.

The updated stereochemical dictionaries have been sent to the CCP4 secretary to be made publicly available and they may be obtained directly from the author (E-mail address: PRIESTLE@FMI.CH).

REFERENCES

- Brünger, A.T., Kuriyan, J. and Karplus, M. (1987) *Science* **235**, 458-460.
- Dreissen, H., Haneef, M.I.J., Harris, G.W., Howlin, B. Kahn, G and Moss, D.S. (1989) *J. Appl. Cryst.* **22**, 510-516.
- Engh, R.A. and Huber, R. (1991) *Acta Cryst.* **A47**, 392-400.
- Hendrickson, W.A. and Konnert, J.H. (1980) in "Computing in Crystallography" (Diamond, R., Ramaseshan, S. and Venkatesan, K., eds.) *Indian Acad. Sci., Bangalore, India*, pp. 13.01-13.23.
- Laskowski, R.A., Moss, D.S. and Thornton, J.M. (1993) *J. Mol. Biol.* **231**, 1049-1067.
- Jack, A. and Levitt, M. (1978) *Acta Cryst.* **A34**, 931-935.
- Jones, T.A. (1978) *J. Appl. Cryst.* **11**, 268-272.
- Jones, T.A., Zou, J.-Y., Cowan, S.W. and Kjeldgaard, M. (1991) *Acta Cryst.* **A47**, 110-119.
- Tronrud, D.E., Ten Eyck, L.F. and Matthews, B.W. (1987) *Acta Cryst.* **A43**, 489-501.
- van Gunsteren, W.F. and Berendsen, H.J.C. (1987) *BIOMOS, b.v., Groningen, The Netherlands*.

A Data Format for the ESRF

Peter J. Daly
European Synchrotron Radiation Facility,
BP 220,38043 Grenoble cedex, France
e-mail: daly@esrf.fr

1 Introduction

This article describes a data format for storage and retrieval of data at the ESRF. The format has the properties described below. It is envisaged that newer versions of the software will be backwardly compatible with older versions.

- it should be flexible enough to cover all types of data collected at the ESRF
- it should allow multiple sets of disparate data
- it should provide as much information about the data itself and any information the user should require with it
- all the calls to read/write the data should be callable from both C and Fortran
- it should be extensible

The basic principle behind the format is that a user who can define his data in a suitable way should be able to obtain that data regardless of the machine or format it is written in. Given the fact that there is no definition of what data actually *is* the format must be flexible enough to cope with different user requirements. The data should be describable in a standard way, if this is possible. For example, a 16-bit image would need the X and Y dimensions, the byte order of the image and it's type to describe the image itself.

2 Data Format Definitions

2.1 Logical File Structure

The data file has a *logical* structure, which is a view of the data file irrespective of it's physical disk layout, as follows: a **Global Header Section** that describes the properties belonging to *all* data within the file and a number of **Data Blocks**, each with it's own **Header Section** that describes properties local to this particular **Data Section**, these local values will displace any global value in the Global Header Section. That is to say, if a global header section specifies an image size of 512x512 and a particular image in the file specifies 1024x1024 then the 1024x1024 size is used; if an image has no information about it's dimensions then the 512x512 value will be used.

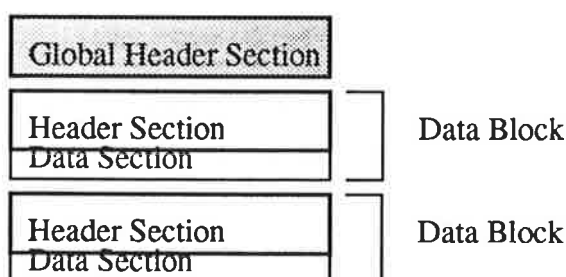


Figure 1: An overview of the data format

In fact there can be more than one global header section providing that the user can identify for himself which global header belongs to which header sections. Files are parsed when opened and the *initial* Global Header Section is the one that contains the **VersionNumber** keyword (if this does not exist no Global Header Section is defined). All the headers are written in ASCII for readability, and each property in a header is described by a statement of the form

KEYWORD = value ;

the KEYWORD identifies the property, the '=' acts as a separator between the keyword and its value(s), the semi-colon terminates the statement, thus a statement can extend over several lines, the rest of the line (up to the newline) is ignored and can act as a comment; whitespace is allowed between any of the elements. Blank lines are ignored. Values can consist of single or multiple items (arrays). The significant length of a keyword is currently 64 characters, although the actual length can be longer if the user wishes.

For each logical header it is the user's responsibility to ensure that some keyword (or group of keywords) can be used to identify the header, failure to do this will mean that a particular header may be unavailable for use. For example, images may be stored with Image = 1, Image = 2, etc.

In the data blocks, the Data Section contains *binary*¹ data described in its header. The Data Section may be empty (i.e. no binary data) but if it does exist it *must* immediately follow the Header Section. Within each Data Section the data may be compressed with a choice of available techniques, random access to any Data Section is allowed. Whitespace is allowed between data blocks to align the start of a block on a convenient boundary (if appropriate).

It should be noted that the format does not require any particular order to the data blocks within the file and implies no particular size of storage element. The Data Sections are stored in whatever size storage element is required by the user, typical values will be Short Integers (16 bit data), Integers (32 bit data), Real Values (IEEE) and some can also be defined as signed or unsigned as required.

Furthermore, there is no requirement that a Data Section read from a file and returned as, for example, REAL is actually stored in the file in the same format that is required. This implies that conversion between the different storage elements takes place.

When reading or writing a piece of data the user may choose to identify the correct place to put it either by performing a search of the headers or by specifying a header number. For example, if we wish to read the title of image 3 then we must specify what we want (i.e. the title) and how to find that information - we can either specify the search condition is 'Image = 3' or (if we know which header this information is to be found) by specifying the header number for this item. However, user's may not always know this value and cannot assume a particular value for a header since the system itself assigns integers as necessary. In general, the search condition 'Image = 3' would be used.

2.2 Physical File Structure

Although the logical structure is all one needs to know at the user level, this section describes the technique of writing information to the file and the reasons for the choice of header format. Furthermore, users can write data files in this format, simply with print statements, provided they remain faithful to the header identifier. At the disk level the structure of the file follows a tagged list; each header has an identity number of the form:

HeaderID = EH:000002:000003:000001;

the numbers are, respectively, the current header number; the header number of the next portion of the header and the header number of the previous portion of the header (the EH stands for ESRF Header and is purely arbitrary). A value of *zero* in the next header field shows that this header is the last in this group and a *zero* in the previous field shows that this is the first portion of the group.

This technique is used rather than byte offsets from the start of the file so that the headers themselves are *editable* by an editor such as *emacs*. Fixed length fields are used (for this statement *only*) since each header *must* have an identifier and variable length identifiers would create problems with overflow. Each header may contain the next strip of data in the binary section. Also in each header is a *Size* keyword to describe the physical size of the data stored. Both the *HeaderID* and the *Size* keyword are managed by the software and should *not be changed by the user*.

1. binary here means that the space between headers is treated as a block of data that the user can interpret at will. For some common storage requirements (e.g. unsigned short integer images) the data will be written in a standard way so the user need not bother about the details, but in some cases the user will simply receive a block of data with no particular emphasis placed on it.

This also implies that a file may have X physical headers but Y logical headers (where $Y \leq X$) since there may be several physical headers to one logical header. As an example, imagine we have the following HeaderID's in a file:

We can see that although there are 5 physical headers in the file, logically there are only 3, since headers 2, 4 and 5 are part of the same logical header. In this case header 2 is (in sequence) made up of headers 2, 5 and 4. The management of this logical versus physical view is provided by the software.

3 Header Section Structure

The example shows the following points:

- this header is number 2, the next logical portion of the header section is number 6 and the previous portion of the header was number 3, thus there are many physical headers to this logical one. The other portions of this logical header would have other suitable keywords and descriptions required by the user.
- comment lines are allowed by prefacing with a semi-colon. The comment extends up to the newline blank lines in the header are ignored,
- some symbolic names are pre-defined e.g. `LowByteFirst` and `HighByteFirst` describe the way that integers are stored (the 'endedness' of the machine),
- multiple values (arrays) are allowed (e.g. the `Cell` statement below),
- some keywords can be initialised but have no value defined until a later time (e.g. the `ProposalNumber` keyword),

the image in this file is 1187x1187 pixels and is compressed - the `Size` keyword records the size of this section of the compressed image

4 Pre-defined Names for common keywords

Keyword Name	Properties
HeaderID	provides a header number and other information to link headers together COMPULSORY
VersionNumber	provides a check on which version of the software wrote the file,
ByteOrder	symbolic values <code>HighByteFirst</code> or <code>LowByteFirst</code> ,
DataType	symbolic values <code>UnsignedByte</code> , <code>SignedByte</code> (both 8 bit), <code>UnsignedShort</code> , <code>SignedShort</code> (both 16 bit), <code>UnsignedInteger</code> , <code>SignedInteger</code> (both 32 bit), <code>Real</code> (32 bit IEEE), <code>DoubleValue</code> (64 bit IEEE). See the section on pre-defined types later,
RealFormat	specify the format of floating point numbers. Can be set to <code>VMS</code> , <code>IEEE</code> (default if omitted).
Size	Size of the Data Section following this header - may be omitted for <code>Size = 0</code> ,
Dim_1,Dim_2, Dim_3, etc.	the size of each dimension of the data actually stored. The maximum number of dimensions is set in the header file define <code>MaxDimensions</code> (currently 512) and follow the C language ordering,
Compression	symbolic names <code>NoValue</code> (no compression), <code>JPEG</code> , <code>JPEG-Lossy</code> (for JPEG lossy compression), <code>RunLengthEncoded</code> , <code>Statistical</code> and other techniques as they are implemented.,
Date	the date the file was written,
Time	the time the file was written,
DetectorName	a string to identify the detector type used,
Title	a title for the header,
SampleName	a string to identify the sample used,
ProposalNumber	value assigned by ESRF from original beam time application

The table above describes the pre-defined keywords for some common statements. Some keywords are required to describe a particular Data Section whilst others are highly recommended. For example, the `HeaderID` is *required*, but the `ByteOrder` keyword is not required (since it will default to the machine type you are using) but is *highly recommended* so that Data Sections can be swapped between different machines with no loss of generality. The `Size` keyword is compulsory if a header has a Data Section.

5 Access to Data Sections

Since the Data Sections may contain any data the user must define how he wishes to identify the Data Sections - access is object-oriented. The user must define the keywords, their values and their types in order to choose a particular Data Section. A Data Section *may* need more than one keyword to access it (e.g. imagine a user who wants an image described by `Image = 1` and `Date = some date`).

In fact, the Data Sections may contain disparate types of data and the user can choose which keys to access those sections he wants. For example, a data file may contain Data Sections that contain calibration information, which the user will identify by the keyword **CalibrationScan** (with some integer value) and image data that he wishes to identify by the keyword **Image** (and a corresponding integer to specify it).

5.4 Sub-sections of Data

It can be seen from the physical file layout that there is no requirement for all the data of, for example, an image be recorded in one Data Section. The data can be divided up into strips (of different sizes) and that each strip can be compressed with a different compression technique. This is managed by the software, however, the user must specify the strip number so that data is not overwritten accidentally.

Furthermore, the interfaces below provide for the user to specify a sub-portion of the data rather than obtain the whole block - again, imagine an image where we only want a small portion of it.

5.5 Defined data types for data stored

The keyword **DataType** is used to describe the data in the data section. It can take any of the symbolic shown in the table below. In the case of **ComplexByte**, **ComplexShortInteger**, **ComplexInteger** and **ComplexLongInteger** the values are treated as signed.

DataType descriptors

SignedByte	UnsignedByte	ComplexByte
SignedShort	UnsignedShort	ComplexShortInteger
SignedInteger	UnsignedInteger	ComplexInteger
SignedLong	UnsignedLong	ComplexLongInteger
FloatValue		ComplexFloat
DoubleValue		ComplexDouble

5.6 Defined types for keywords

When a user wishes to read a header item or search for a header he must specify 3 items of information: the keyword, it's value and it's type. For example, to find a data section described by the statement

```
Image = 1;
```

the user would need to specify the keyword "image", the value (in this case 1) and it's type for a suitable comparison to be made - currently, the types can be **IntegerValue**, **FloatValue** or **StringValue**. We call these keyword descriptors. In the example above the type would, of course, be **IntegerValue**.

6 Interfaces to Manipulate the Data file

The interface routines to manipulate the data file return the (integer) symbolic value **RoutineSucceeded** for success or **RoutineFailed** for failure. Most routines to access the data have three arguments:

- a channel number to specify which file to use, this is an integer to a (private) array - each element of this array is a structure that records various properties of the file opened,
- a control block structure to set values and return results,
- a returned integer error value to describe success or failure of the routine. This value gives a more explicit description of the reason the symbolic name **RoutineFailed** was returned. A routine is provided so that the user can be given a description of the error.

Various flag values have been defined to make the use of the library easier. They are explicitly listed below:

Flag	Used for
ReadData	Specifies Data Section is to be read
ReadHeader	Specifies a Header Section is to be read
InquireData	Specify that we want to know the size and type of the data without the overhead of reading it from the disk
WriteData	Specifies Data Section is to be written
WriteHeader	Specifies a Header Section is to be written
SearchCondition	Tells the interface then argument 5 is a block of conditions rather than a simple header number. This allows searches for different matches and composite keys
FlushTables	Explicitly flush the buffers rather than <i>cache</i> some headers until the files are closed. Can cause fragmentation of the headers.
AllocateSpace	will cause the library to use <i>malloc</i> to create the necessary space. If this is omitted the routines will take the start address as the target for the data and the sizes as inputs.

These flags (and examples of their use) can be viewed in the example programs.

7 Future Plans

In the next release of the software we are planning further compression techniques; an improved parser for the header sections (for example, currently a header statement like `Cell = 105.77 105.77 153.37 90 90 120`; would return the first number in the group (105.77) or a string "105.77 105.77 153.37 90 90 120", which the user would have to parse himself) and a Record Description keyword to provide tabular data. For example, a user may write (something like):

```
Record = UnsignedInteger[3] StringValue[20] ;
```

or

```
Record = UnsignedInteger UnsignedInteger UnsignedInteger StringValue[20] ;
```

meaning the data is recorded as 3 Unsigned Integers and a string of length 20 characters.

Furthermore, the author is currently working on read/write routines for input into *xv* - a program for displaying images on X-terminals.

8 Availability

The library source files, header files and some example programs are available via ftp from the machine `expga.esrf.fr` (Internet number 160.103.2.141). The full documentation is also provided in PostScript form within the tar file.

`~ftp/dist/esrf_data_format.tar.Z` - source code, header files and Makefiles

`~ftp/dist/esrf_examples.tar.Z` - example test files; these are rather large so have been split from the main source

If you require the Fortran interface, you will also need a package called `cfortran.h` obtainable from the ftp site is `zebra.desy.de` (131.169.2.244)

Network News

Dave Love

The following resources which have recently popped up in cyberspace and may be of interest:

- There is a new Usenet newsgroup `sci.techniques.xtallography` in addition to `bionet.xtallography`. It isn't restricted to the macromolecular. (If you don't know about Bionet, you can e-mail `biosci@dl.ac.uk` for information.);

- CONCISE—The European Information Server:

"There is an abundance of up to date information concerning crystallography in Europe and worldwide on CONCISE, the European Information Server.

You can connect to CONCISE by the command:

```
telnet concise.level-7.co.uk
```

You log in with the userid `concise` and password `concise`. You configure the terminal. Usually the best is `d vt100`. You go to the crystallography section which is a sub-section of the special interest groups with:

```
g /sigs/crystal
```

The first item explains how items are to be submitted. Have fun."

(From H. D. Flack (`flack@scsun.unige.ch`).)

If this means nothing to you, perhaps you need the "Big Dummy's Guide to the Internet":

"This posting tells you everything you ought to know about an exciting project that helps so-called Newbies to survive their first year on The Net. It gets you softly into most of the often severe details of mindboggling complexity that popup when you're hooked into The Net. It is also intended to help you get around the trap doors, avoid pitfalls and other non-funny things (e.g. local gurus that laugh at you), you'll encounter during life as a Netizen in Cyberspace."

It's available in many places, e.g. anonymous ftp:

`ftp.eff.org:pub/EFF/papers`, Gopher: `gopher.eff.org` and

Word Wide Web:

`http://www.germany.eu.net/books/bdgtti/bdgtti-intro.html`
or e-mail server:

To: `archive-server@germany.eu.net`

Subject:

BEGIN

REPLY <your e-mail address>

SEND `books/big-dummys-guide/README`

END

DIARY OF FUTURE MEETINGS

1993

- December 9-14 Crystallography in Molecular Biology, Madras, India.
[Prof S Parthasarathy, Dept Crystallography and Biophysics,
University of Madras, Guindy Campus, Madras 600 025, India].
- December 20-21 Oligosaccharides. London. [Dr S Homans, Medical Sciences
Institute, University of Dundee, Dundee, DD1 4HN].

1994

- January 5 BBS Meeting "Biological Mechanisms from Crystal Structures".
NIMR, London. [Dr P Turner, NIMR, Mill Hill, London NW7
1AA].
- January 6-7 CCP4/ESF January meeting "From First Map to Final Model".
Chester [SAS Conference Office, Daresbury Laboratory,
Warrington, Cheshire WA4 4AD].
- January 8-9 ESF workshop "Phase improvement and map interpretation". York.
[Mrs E Dodson, University of York, Chemistry Department,
Heslington, York YO1 5DD].
- January 21 "Antibacterial Peptides". London. [Sheila Pusinelli, The Ciba
Foundation, 41 Portland Place, London W1N 4BN].
- January 31-Feb 12 FEBS Advanced Course. "Biomembranes and Signal Transduction".
Utrecht, The Netherlands. [Dr A J Verkleij, Institute for
Biomembranes, Utrecht University, Padualaan 8, 3584 CH Utrecht,
The Netherlands].
- March 6-9 3rd European Symposium on Calcium-binding Proteins in Normal
and Transformed Cells. Zurich. [Dr C W Heizmann, U Children's
Hospital, Div of Clinical Chemistry, Steinwiestrasse 75, CH-8032
Zurich, Switzerland].
- March 28-31 BCA Spring Meeting. Newcastle upon Tyne. [Prof W Clegg,
Department Chemistry, University of Newcastle, Newcastle upon
Tyne NE1 7RU].

1994

- March 29-30 Ion Channels: Structure and function. London. [Dr D Ogden, National
Institute for Medical Research, Mill Hill, London NW7 1AA].
- April 5-8 European Symposium on Frontiers in Science and Technology with
Synchrotron Radiation. Aix en Provence, France. [Dr J Gastaldi,
CRMC2-CNRS, Campus de Luminy - case 913, 13288 Marseille Cedex
09, France. Tel: +33-91826028; Fax: +33-91826029].

1994

- April 12-14 British Biochemical Society Meeting. Cardiff meeting.
[Julie Staerck, The Meetings Office, The Biochemical Society,
59 Portland Place, London W1N 3AJ].
- April 20 SERC Interdisciplinary Research Centre for Biochemical Engineering,
Annual Research Symposium. London. [Prof P Dunnill, Director, IRC,
Chemical and Biochemical Engineering Department, University College
London, Torrington Place, London WC1E 7JE; Tel: 071-380-7031;
Fax: 071-388-0808].
- May 13-23 FEBS Advanced Course. "Chemistry of Metals in Biological Systems".
Wavre, Belgium. [Prof R R Crichton, Unité de Biochimie, Université
Catholique de Louvain, Place Louis Pasteur 1, B-1348 Louvain-la-
Neuve, Belgium].
- May 15-21 FEBS Advanced Course. "Computational Methods in Chemical Design-
Molecular Modelling, Theory and Experiment". Kloster Isree,
Germany. [Prof G Müller, Universität Konstanz, Fakultät Chemie,
Universitätsstraße 10, D-750 Konstanz 1, Germany].
- May 20 "Cell Adhesion and Human Disease". London. [Sheila Pusinelli, The
Ciba Foundation, 41 Portland Place, London W1N 4BN].
- May 27-June 5 21st Crystallographic Course on Crystallography of Molecular Biology.
Erice, Italy. [L. Riva di Sanseverino, Dipto Scienze Mineralogiche,
Pze Porta S Donato 1, 4126 Bologna, Italy].
- June 13-22 FEBS Advanced Course. "Basic and Specialised Techniques in Cell
Biology". Aarhus, Denmark. [Prof J E Celis, Institute of Medical
Biochemistry and Danish Centre for Human Genome Research,
University of Aarhus, Ole Worms Allé, Building 170, DK-8000 Aarhus,
Denmark].
- June 26-July 1 "Biological Membranes". FEBS Special Meeting. Espoo, Helsinki,
Finland. [FEBS '94, Congress Management Systems, PO Box 151,
SF-00141 Helsinki, Finland. Fax: (358) 0 17012].
- June 26-July 1 American Crystallographic Association. Atlanta, USA. [C Carter,
Program Chair, Dept Biochemistry & Biophysics, 302 Flob CB 7260,
University of North Carolina, Chapel Hill, NC 27599-7260 USA].
- July 11-15 IUPAC 25th International Symposium on Macromolecules.
Akron, USA. [Dr J P Kennedy, University of Akron, Institute of
- July 13-14 British Biochemical Society Meetings. Lancaster meeting.
[Julie Staerck, The Meetings Office, The Biochemical Society,
59 Portland Place, London W1N 3AJ].
- July 17-22 17th International Carbohydrate Symposium. Ottawa. [Doris Ruest,
Executive Secretary, VX11th Int'l Carbohydrate Symposium.,
National Research Council, Ottawa, Canada K1A 0R6].
- July 18-22 5th International Conference on Synchrotron Radiation
Instrumentation. New York. [L Lever, NSLS at Brookhaven
National Laboratory, Building 725D, Upton NY 11973, USA].

- July 24-30 XIIth International Congress of Pharmacology. Montreal.
[Congress Secretariat, IUPHAR '94 Conference Services, National
Research Council of Canada, Ottawa, Canada K1A 0R6].
- August 14-26 FEBS Advanced Course. "Molecular Mechanisms of Transcellular
Signalling: from the Membrane to the Gene". Spetsai, Greece.
[Prof K W A Wirtz, Centre for Biomembranes and Lipid
Enzymology, PO Box 80.054, 3508 TB Utrecht, The Netherlands].
- August 28-Sept 2 ECM-15, European Crystallographic Meeting. Dresden.
[Prof P Paufler, Fachbereich Physik, Technische Universitaet Dresden
Mommensenstrasse 13, D-O-8027 Dresden, Germany, Tel: 3378;
Fax: {[37]51} 463 7109].
- August 28-Sept 10 FEBS Advanced Course. "A World of RNA: Structure and
Function". Spetsai, Greece. [Prof M Grunberg-Manago, Institut de
Biologie Physico-Chimique, 13 Rue Peirre et Marie Curie, 75005
Paris, France].
- September 6-9 British Biochemical Society Meeting. Canterbury meeting.
[Julie Staerck, The Meetings Office, The Biochemical Society,
59 Portland Place, London W1N 3AJ].
- September 8-10 "Biological Electron and Proton Transfer". Ashford.
[Dr P R Rich, Glynn Research Institute, Bodmin, Cornwall
PL30 4AU. Tel: 020-882482; Fax: 020-882572].
- September 8-13 10th International Conference on "Methods in Protein Structure
Analysis". Snowbird, Utah. [Dr M Z Atassi, Department of
Biochemistry, Baylor College of Medicine, Houston, TX 77030,
USA. Tel: (713) 798-6050. Fax: (713) 796-8040].
- September 19-25 FEBS Advanced Course. "Advanced Methods" DNA Sequencing
and Microinjection". Prague, Czech Republic. [Prof W Ansorge,
EMBL, Postfach 10.2209, 6900 Heidelberg, Germany].
- December 13-16 British Biochemical Society Meeting. Sussex meeting.
[Julie Staerck, The Meetings Office, The Biochemical Society,
59 Portland Place, London W1N 3AJ].

1995

- April 3-7 British Biochemical Society Meeting. Leicester meeting.
[Julie Staerck, The Meetings Office, The Biochemical Society,
59 Portland Place, London W1N 3AJ].
- July British Biochemical Society Meeting. Manchester meeting.
[Julie Staerck, The Meetings Office, The Biochemical Society,
59 Portland Place, London W1N 3AJ].
- September British Biochemical Society Meeting. University College, Dublin
meeting. [Julie Staerck, The Meetings Office, The Biochemical
Society, 59 Portland Place, London W1N 3AJ].

ESF NETWORK FOR THE CRYSTALLOGRAPHY OF BIOLOGICAL MACROMOLECULES



PHASE IMPROVEMENT AND MAP INTERPRETATION

WORKSHOP JANUARY, 1994

A practical workshop will be held at the University of York to look at methods of phase improvement, the effect of these calculations on maps and to gain experience in using the current electron density display software. This will include O, RAVE and QUANTA. Those attending are encouraged to bring their own data for analysis by the various methods. The workshop will start on the morning of Saturday, 8 January and end on the evening of Monday 10 January. This follows the Daresbury Workshop.

It is hoped tutors will include:

E. Dodson	University of York
P. Fitzgerald	Merck Sharp and Dohme
K. Henrick	Mill Hill
R. Hubbard	University of York
G. Kleywegt	University of Uppsala
V. Lamzin	EMBL, Hamburg
P. Main	University of York
Z. Otwinowski	Yale, New Haven
D. Stuart	Oxford

The number of places is restricted to a maximum of 16. The ESF network will cover costs of accommodation and subsistence. Some grants towards travel may be available but you are encouraged to arrange independent travel funds if possible. Please indicate on your application if travel support is essential for your attendance.

Please send applications by 31 October 1993 to:

Mrs Eleanor Dodson
University of York
Chemistry Department
Heslington,
York. YO1 5DD.

Fax: 0904 410519
email ccp4@uk.ac.york.yorvic

Include a one page curriculum vitae with your application. Pre-doctoral students should also enclose a short letter of recommendation from their supervisor.

EMBO Practical Course
Macromolecular Crystallographic Data
Collection

27th February - 5th March 1994

MRC Laboratory of Molecular Biology, Cambridge, UK

This course will cover the measurement of monochromatic X-ray diffraction data from crystals of macromolecules. Lectures will cover all aspects of data collection and processing. Students will collect and process data from a number of area detectors (electronic and image plates).

Topics to be covered:-

- X-ray sources, optics, collimation
- Crystal freezing
- Detectors: multi-wire chambers, TV, image plates
- Data collection strategies
- Crystal orientation determination
- Integration, profile fitting
- Data reduction

Instructors: P.R. Evans, A.G.W. Leslie (course organizers), U.W. Arndt, A.R. Faruqi, J.F.W. Mallet (Cambridge), Z. Dauter (Hamburg), E. Garman (Oxford), W. Kabsch (Heidelberg), Z. Otwinowski (Yale), J.W. Pflugrath (Cold Spring Harbor), P. Tucker (Heidelberg), A.J. Wonacott (Glaxo).

20 participants will be selected, with preference to post-docs or students currently (or about to be) involved in data collection, and to those from member countries of EMBO. Board and lodging will be provided, and some funds are available for travel. Applicants should send a CV, a statement of how they will benefit from the course, and a recommendation, by 3rd December 1993, to Dr. P.R. Evans or Dr. A.G.W. Leslie, MRC Laboratory of Molecular Biology, Hills Road, Cambridge CB2 2QH, UK, from whom further details may be obtained (tel: +44 223 248011, E-mail pre@mrc-lmb.cam.ac.uk).

DR P.J.RIZKALLAH
DARESBURY LABORATORY
INTERNAL

CS 10