

PREFACE

A Brief History of Protein Crystallographic Computing



Advances in protein crystallography have been fundamentally tied to advances in computing since Blow and Crick [1], working with Kendrew, used one of the earliest computers, EDSAC to phase myoglobin in the early 1950's. Protein crystallographic computing differs from small molecule crystallographic computing most fundamentally in the size of the problem, forcing protein crystallographers to chase after the largest computers available. A protein crystallographic problem may have 10,000 to 100,000 observed reflections and take up 0.4-4.0 megabytes of memory to store. Today such sizes seem small but, until this decade, they represented a significant fraction of the storage available on the university computer. Even with this large dataset, the number of atoms and, thus, the number of parameters to be fixed (i.e. three positional and one isotropic thermal), are greater than the number of data measured, making the problem under-determined. This is because the maximum resolution that could be obtained for most proteins was 2.0-2.5 Å.

In the last few years the extremely bright X-rays available at synchrotron light sources has allowed some protein crystals to achieve diffraction at atomic resolution ($< 1.2\text{Å}$). This year there will be an explosion in the number of proteins determined to atomic resolution and there are rumors of previously unsolved protein structures being determined ab initio by direct methods with such data. Paradoxically, as proteins data sets reach atomic resolution and grow to ever larger sizes, the small molecule program SHELXL, has become the preferred program for refinement and analysis. The explosion in structures is due in large measure to increases in the power of computers, parallel advances in software, and to advances in genetic engineering which will not be discussed in this article.

Every few years since 1969, the International Union of Crystallography has had a Crystallographic Computing School. The proceedings of these schools are interesting to read from the historical perspective and were eloquently summarized in a paper by Stewart and Bourne at the last school [2]. At the first school in 1969 an early macromolecular Fourier transform for ribonuclease by G. Kartha on an IBM 7040 mainframe was clocked at 52 minutes [3]. Stewart and Bourne report repeating the calculation on an IBM 6000 workstation at 1.5 minutes. Using a Fast Fourier Transform (FFT) and a DEC Alphastation 500/500, the same calculation takes about 1 second. In order to do this calculation in 1969, G. Kartha had to very carefully fit the problem into the 192 kilobytes of memory of an IBM 7040. Three-dimensional FFTs were possible but had to be very carefully coded in order to fit in the available memory, and a special version was used for each spacegroup. On my DEC Alpha, which is also orders of magnitude less expensive, I have 393,216 kilobytes available and no special treatment is needed-the problem is expanded to P1, which uses the maximum possible memory, but the least programming effort.

At the 1975 school [4], there was much debate on main-frame computers versus the emerging minicomputers. In 1980 [5] the feasibility of using interactive computing for all steps of the problem was discussed, although such methodology was a long way from implementation. In 1981 [6], the school had a session in which several successful programs for building protein models interactively were reported. For the first time, concerns of storage and timing have disappeared from concern. At that time, the modern era of workstations had dawned although it would take some time for mainframes and minicomputers to disappear from the scene. Throughout the 1980's, debates raged at the schools on the virtues of Digital's VMS versus UNIX as a computer operating system and which was preferable for protein crystallographic labs.

At the last school [2], there were reports of using graphical user interfaces, automated fitting and techniques for atomic resolution refinement of proteins. LINUX PC's were being used for all aspects of solving protein structures. Interestingly, Lynn Ten Eyck of the San Diego Supercomputer Center reported on some new techniques for analyzing the full normal matrix used in least-squares refinement to detect previously undetectable errors and cross-correlations. These analyses use several gigabytes of computer memory and take many hours on the fastest supercomputers available. As always, protein crystallographers continue to push the envelope of computing.

Throughout this process protein crystallographers, with some rare exceptions, have been users and not designers of computing systems. As such, they have had to adapt to whatever hardware was available at the time. Before the concept of portability became a prevalent buzzword (if not always achieved in reality), many man-hours were wasted by crystallographic programmers in porting complex software systems every time the university replaced the computer or changed the OS. Since computing advances so rapidly, often the computer would be changed every few years. Crystallographers who moved to a new institution might find themselves spending months getting software up and running before he could continue work on the new computer system. Fortunately, the other aspects of protein crystallography were proportionately slower-it was not uncommon for a protein to take a decade to be solved.

In the early days concerns of speed and memory dominated much of the work on algorithms. It was necessary to use techniques such as overlays, where part of a program was overlaid onto other parts as they were needed. A modern program is always loaded into memory in its entirety. The advent of virtual memory eased this problem somewhat. With virtual memory the computer swaps pages of memory in and out transparently so that the total memory looks very large to the program even though the real memory was still small by today's standard. Programmers still had to be very careful to loop through an array in such a manner as to linearly address the memory or the computer could thrash loading pages in and out until it ground to a virtual halt. Thus, every programmer had to have intimate knowledge of the addressing system of the compilers. Early programs also used lookup tables for functions such as sine or logarithms to increase speed. In the early days a table was prepared with pre-computed sine values for perhaps every tenth of a degree. The program would then take each degree and multiply it by 10, truncate it to an integer and use that as an index to look up the pre-computed value. Now computers have built in floating point units that compute a sine as quickly as a multiplication and faster than this lookup process. To save space, programmers invented complicated bit-packing schemes so that several numbers could be represented as one. Whenever the program accessed one of these numbers it was unpacked, calculated, and repacked. A lot of overhead to save a few kilobytes of memory!

By and large most protein crystallographic software is, and has been, written in FORTRAN. FORTRAN was developed in 1958 at IBM with a crystallographer, David Sayre. Over the last decade software written

in C, C++, and Java is starting to emerge but FORTRAN is still the leader. The chief advantage of C over FORTRAN is ease of memory allocation. However, it is possible to use memory allocation in FORTRAN (e.g. XPLOR) or to use a heap management system (e.g. SHELX). Modern compilers are very good at taking any code and reducing it to the minimum number of machine instructions so that the choice of languages is mostly a matter of programmer taste. Java is coming into use for mini-applications on web pages to provide access to database information and provide mini-viewers.

It would be difficult to assemble a comprehensive list of protein crystallographic programs. Indeed many programs were written for a specific problem and used only once, and, as code has been passed from lab to lab, sometimes it is hard to pinpoint when a method was first used. At the risk of leaving out many who deserve mention, I have compiled a short table of landmark programs (Table 1). I used the criterion of popularity, long-life or first in a now widely used method.

A day in the life of a grad student - To illustrate how much computing has changed crystallography over the years we will look at a typical grad student day using the computer to calculate a map. Since I am not old enough to have experienced crystallography in the 1960's I will start in the 1970's.

1970's. The graduate student walks to the university computer center with several boxes of punched cards under his arm. The punched cards hold the data processed the day before from a paper tape punched by the Picker diffractometer and merged with phases previously solved. Another box contains the program to calculate a Fourier transform. At the computer center he would pull out the cards at the back of the program that represented the control cards and carefully key punch them for his problem. He might also pull out the space group routine and recode it for his space group and change the read format back to the one for his data (the back of the deck contains lots of old cards, so he only has to pick through them to find his format). He then hands the cards across a window to be run by the computer operator. Later he returns to pick up his output from a bin. If there was an error he would repeat the process the next day. (Actually he would probably be doing this after midnight when the rates are cheap enough to actually run such a large job as a Fourier within his boss's computing budget.) He plots the map onto large glass sheets and installs it in the Richards Box - a device with a half-silvered mirror that allows simultaneous viewing of a map and a brass Kendrew model - and wishes for one of those state-of-the-art computer graphics systems.

1980's. The lab now uses a newer faster minicomputer but it is still administered by the university. His boss has bought him a terminal (glass-teletype) that can connect to the computer through a 300 baud modem from home. This way he can log in at night when the computer is free from word-processing and other jobs in order to get enough time to run such a large job as a Fourier. He plots his output onto clear plastic, when his map looks convincing, he signs up to use the E&S PS300 graphics system in the basement of the adjacent building on Saturday morning (the only time it is free). He wishes he could have one of those state-of-the-art workstations that he has read about.

1990's. The student sits at one of the lab SGI workstations, clicks on the file name to load the data, presses FFT after confirming the resolution desired. The map displays on his computer and he begins fitting it. He thinks to himself "when are they going to automate this bloody fitting so I don't have to spend so many hours doing it manually?".

2050. No one needs a grad student. The researcher sends his crystals out to the service crystallographer who solves the structure completely automatically with his table-top Rigaku synchrotron light source and 3000 MHz PC with 128 gigabytes of RAM (its an old out-of-date model). Of course, the researcher only

does this to confirm the structure already determined by the folding prediction algorithms on his laptop before publication.

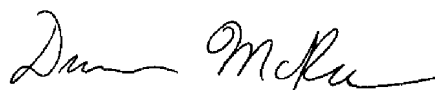
Future prospects - We have entered into a golden age of software development for protein crystallography. From the user's point of view, there are a wide choice of programs and packages to choose from, all of which work well. The user now finds himself in the enviable position of being courted by competing groups and heretofore unimportant details, such as the users time and effort, have become major concerns for developers. In the early days, no cardcarrying protein crystallographer could do without some programming knowledge. These days, protein structures are routinely solved by people who have no idea how to program a computer but only how to run programs. While I suspect it will still be some years before solving a protein structure will be as simple as solving a 50-atom small molecule structure is today, the day is rapidly approaching where only the quality of the sample will matter in achieving a solution.

Table 1 Some protein crystallography software landmarks.

HALSQ	M. Rossman [7]	1969 MIR phasing program which forms basis of many successor programs and is still in use.
MOSFLM	A. Leslie [8]	Film integration finding new life with image plates and CCD detectors, GUI interface.
DENZO	Z. Otwinowski [9]	Data Integration added reliable space group determination.
XENGEN	A. Howard [10]	Data integration with automated indexing, 3-D profiles and fine-slicing.
PROLSQ	W. Hendrickson and J.. Konnert [11]	Refinement program that combined reciprocal space and protein stereochemistry.
FFT	L. Ten Eyck [12]	Practical large Fast Fourier Transform used in many programs.
ENVELP	B.C. Wang [13]	Automated solvent boundary determination.
Phases	W. Furey [14]	Combined phasing and density modification methods in an easy to use package.
Merlot	P. Fitzgerald [15]	A package that brought molecular replacement to the masses.
SOLVE	T. Terwilliger [16]	Totally automated phasing from MIR or MAD data that actually works.
CCP4	Many [17]	Widely used and versatile collaborative softwarepackage used to solve innumerable protein structures.
SIGMAA	R. Read [18]	Weighting scheme to help remove phase bias.
SHELX	G. Sheldrick [19]	Small molecule package becomes best way to refine very high resolution protein structures.
FRODO	A. Jones [20]	Popular fitting program still used today after a decade and a half.
XPLOR	A. Brünger [21]	Combines dynamics with refinement and a powerful control language.
PROTEIN	W. Steigemann [22]	Macromolecular structure solution package.
XtalView	D. McRee [23]	First protein structures completely solved interactively using only a GUI.

References

- [1] D. M. Blow and F. H. C. Crick. The treatment of errors in the isomorphous replacement method. *Acta Cryst.*, **12**: 794-802, 1959.
- [2] Crystallographic Computing 7. Proceedings from the Macromolecular Crystallography Computing School. Eds. Philip E. Bourne and Keith Watenpaugh. Held August 17-22, 1996, Western Washington University, WA. <http://www.sdsc.edu/projects/Xtal/IUCr/CC/School96/IUCr.html>.
- [3] Crystallographic Computing, Proceedings of the 1969 International Summer School, Ed. F. R. Ahmed, S. R. Hall, and C. P. Huber, Munksgaard, 1970.
- [4] Crystallographic Computing Techniques, Proceedings of the 1975 International Summer School, Ed. F. R. Ahmed, K. Huml, and B. Sedlacek, Munksgaard, 1976.
- [5] Computing in Crystallography, Proceedings of the 1980 International Winter School, Ed. R. Diamond, S. Ramaseshan, and K. Venkatesan, Indian Academy of Sciences, Bangalore, 1980.
- [6] Computational Crystallography, Proceedings of the 1981 International Summer School, Ed. David Sayre, Clarendon Press, 1982.
- [7] Adams, M. J., D. J. Haas, B. A. Jeffery, A. McPherson, Jr., H. L. Mermall, M. G. Rossmann, R. W. Schevitz, A. J. Wonacott. 1969. Low resolution study of crystalline L-lactate dehydrogenase. *J. Mol. Biol.* **41**: 159-188. Rossmann, M. G. 1976. The refinement of heavy-atom parameters in the presence of non-crystallographic symmetry. *Acta Crystallogr.* **A32**: 774-777.
- [8] A.G.W. Leslie (1990) in 'Crystallographic Computing', Oxford University Press.
- [9] Zbyszek Otwinowski, 'Oscillation Data Reduction Program' in Proceedings of the CCP4 Study Weekend: 'Data Collection and Processing', 29-30 January 1993, Compiled by: L. Sawyer, N. Isaacs and S. Bailey, SERC Daresbury Laboratory, England, pp. 56-62.
- [10] Howard, Andrew J., Gilliland, Gary L., Finzel, Barry C., Poulos, Thomas L., Ohlendorf, Douglas H. & Salemme, F. Ray (1987) The use of an imaging proportional counter in macromolecular crystallography. *J. Appl. Crystallogr.* **20**, 383-387.
- [11] Hendrickson, W. A., & Konner, J. H. (1980) Incorporation of stereochemical information into crystallographic refinement In "Computing in Crystallography", Diamond, R., Rameshan, S., Venkatesan, K. eds. Indian Academy of Sciences, 10.01-10.23.
- [12] L. F. Ten Eyck, *Acta Cryst.*, **A29**, 486, (1973).
- [13] Wang, B.C. (1985). Resolution of phase ambiguity. *Methods in Enzymology* **115**, 90-112.
- [14] Furey, W., University of Pittsburgh, PA
- [15] P. M. D. Fitzgerald. Molecular replacement. In D. Moras, A. D. Podjarny, and J. C. Thierry, editors, Crystallographic computing 5, pages 333-347. Oxford University Press, New-York, 1991.
- [16] Terwilliger, T., Los Alamos National Laboratory, Los Alamos, NM. <http://www.solve.lanl.gov/>.
- [17] Collaborative Computational Project, Number 4. 1994. "The CCP4 Suite: Programs for Protein Crystallography". *Acta Cryst.* **D50**, 760-763. <http://www.dl.ac.uk/CCP/CCP4/main.html>.
- [18] Read, R.J. (1986) Improved fourier coefficients for maps using phases from partial structures with errors. *Acta Cryst.* **A42**, 140-149.
- [19] SHELX. Sheldrick, G. Institute of Inorganic Chemistry, Georg-August-Universität, Göttingen. Germany. Sheldrick GM, Schneider TR: 'SHELXL: High Resolution Refinement', *Methods in Enzymology* (R.M. Sweet and C.W. Carter Jr., eds.), Academic Press; Orlando, Florida, 277: 319-343 (1997). <http://linux.uni-ac.gwdg.de/SHELX/>
- [20] Jones, T. A. (1978) A graphics model building and refinement program system for macromolecules. *J. Appl. Cryst.* **11**, 268.
- [21] Brunger, A. T., Kuriyan J. & Karplus, M. (1987) Crystallographic R-factor refinement by molecular dynamics. *Science* **235**, 458. <http://atb.csb.yale.edu/>.
- [22] The PROTEIN program system. W. Steigemann, Rechenzentrum, Max-Planck-Institut für Biochemie, D-82152 Martinsried, Germany. <http://www.biochem.mpg.de/PROTEIN/>.
- [23] McRee, Duncan E. (1992) A Visual Protein Crystallographic Software System for Xll/Xview. *J. Mol. Graphics* **10**, 44-46. <http://www.scripps.edu/pub/deni-web/toc.html>.



Duncan E. McRee, Ph. D.

Assistant Professor
Protein Crystallography
Department of Molecular