Structural Biology on Mac OS X: Technology, Tools, and Workflows

By David W. Gohara, Ph.D.
Center for Computational Biology
Washington University School of Medicine

February 2007

Contents

Page 3 **Executive Summary**

Page 4 **Introduction**

Page 7 **Computational Requirements**

Page 12 **Technology Platforms**

Page 15 **Case Study: Workflow for X-ray Crystallography**

Page 26 **Conclusion**

Page 27 **References**

Executive Summary

Structural biology is important for relating molecular structure to biological function. The techniques and tools for solving the 3D structures of biological macromolecules is a computationally intensive process. Over the years, a variety of technology platforms have become de facto standards for macromolecular structure determination. In order for a technology platform to become widely adopted, it must provide features that either simplify the overall structural determination workflow by consolidating technologies into a single platform or offer performance-enhancing features that reduce the time spent waiting for calculations to finish and thereby increasing overall productivity, or both.

With the introduction of Mac OS X from Apple and the availability of computer systems that provide much of the critical functionality needed for structure determination, Mac OS X has emerged as a platform that is quickly becoming the new standard for structural biology work. In particular, through the combination of simple, elegant desktop computing, coupled with the full power of UNIX running on high-performance hardware, Mac OS X based systems provide a practically turnkey solution for macromolecular structure determination.

In this white paper, the techniques commonly used for structure determination are described. Additionally, some of the most important features in computer systems used for structural biology are discussed in detail, with a specific emphasis on technologies present in Mac OS X and available on Apple hardware. Combined into a single package, the technologies and tools discussed here greatly simplify the overall workflow of a structure determination. Finally, a case study outlining the software and hardware tools that can be used at various stages of a structure determination by X-ray crystallography is provided.

Introduction

During the last 20 years, 3D macromolecular structure determination has become a primary method for the elucidation of structure/function relationships of biological macromolecules. Large-scale mapping of organism genomes has flooded databases with a vast amount of information about putative genetic function, but for which no direct structural data is available. Combining structural information with genetic functional data opens up new paths for understanding how complex biological systems function genetically, biochemically, and structurally. Such analyses are important for advancing our fundamental knowledge of biological systems and ultimately our understanding of the root causes of disease and illness. Already the combination of genetic, biochemical, and structural analysis has yielded great advances in the identification of putative drug targets and the development of treatments using rational drug discovery methodologies.

While genetic and biochemical analyses have enjoyed a long history of study across a variety of disciplines, structural biology has been relegated to a relatively small number of specialists due to the complexity of the techniques involved and the equipment and computational power required. Additionally, structural biology as a discipline is somewhat of a misnomer, as there is no single technique that is employed for under-standing the relationship between molecular structure and function. The majority of experimentation typically occurs on the bench in the "wet lab" outside the realm of the structure determination itself. In fact, the actual structure determination encompasses only a relatively small, albeit important, portion of the information required to obtain a complete picture of how molecules interact in a biological system. And while the results obtained with a 3D structure often appear more exciting due to their visual nature, it would be a complete misrepresentation to imply that the structural component is truly meaningful in the absence of the biological analysis.

So how can we better define structural biology? 3D structure determination of biologically relevant macromolecules can be accomplished by a variety of techniques covering two broad classes: Comparative Modeling (CM) and Experimental Structure Determination[1]. CM-based techniques rely on the fact that although amino acid sequence conservation within related proteins can be low, there are a finite number of biologically relevant tertiary folds that any given sequence can adopt. And in functionally homologous proteins, evolution tends to favor structural conservation over sequence conservation. In other words, two homologous proteins related by function will tend to have similar structural folds, although their sequence similarity may be low. Homology modeling exploits this observation by using previously determined structures to construct putative 3D models of evolutionarily related molecules. A second CM-based technique is protein threading. Whereas homology modeling exploits sequence similarity of functionally related proteins, protein threading utilizes only the amino acid sequence in combination with a database of known, although possibly unrelated, structures to create possible 3D models of the sequence.

Although CM-based techniques can be powerful in their predictive capabilities of 3D molecular structures (especially when combined with other information such as biochemical analysis, site-directed mutagenesis, and genetic data), the de facto standard for three-dimensional structure determination is experimentally based. While there are a variety of empirical techniques that can be used to elucidate the 3D structure of biological macromolecules (either in part or as a whole), the primary methods are the following: X-ray crystallography, NMR, and cryo-electron microscopy. Like any methodology, each technique has its strengths and weaknesses, and they often complement one another.

For example, X-ray crystallography presupposes that molecules can be crystallized, whereas NMR doesn't have such a requirement. Crystallization also imposes some practical limits on the size of biological complexes that can be studied, as larger complexes are more difficult to purify to the levels required for crystallization to occur (in terms of both solution concentration and absolute amount for crystallization trials). Conversely, NMR relies on the ability to isotopically label particular atoms or amino acids in a molecule, which can be both difficult and expensive. Again, relatively high concentrations of sample are required. The practical range of molecular weights covered using one technique or the other is approximately 100 Daltons (Da) to 10 MDa (not including whole virus structures, which can be as large as 150 MDa or more). However, both techniques can be considered atomic resolution methods (where resolution refers to the ability to differentiate features at the atomic level, approximately 1–3 Angstroms (1 Angstrom = 10^{-10}m).

A third technique becoming more widely utilized is cryo-electron microscopy (cryo-EM). Cryo-EM provides several advantages over X-ray and NMR-based methods. Repeating assemblies, like those found in two-dimensional crystals and helical complexes, as well as randomly oriented single particles can all be studied, offering more flexibility for the types of macromolecules available for study. The range of sample sizes suitable for analysis is relatively wide (approximately 100 kDa–400 MDa). In theory, atomic resolution can also be achieved with this technique provided a powerful enough microscope, a high-resolution imaging camera, and enough samples for analysis. In practice, the current upper limit on resolution is approximately 7 Angstroms, with the average being between 10 and 25 Angstroms. At this resolution, the general fold of a molecule or organization of several molecules in a biological complex is observable. However, detailed information about atomic contacts, such as in the binding of a drug compound to a protein, is lost.

A cursory analysis of the Protein Data Bank reveals that the most commonly used method for structural studies is X-ray crystallography (~85% of deposited structures), followed by NMR (~15%), and cryo-EM (< 0.1%)[2]. Indeed, X-ray crystallography and NMR have been in use over 40 years for the study of biological macromolecules. It is not surprising that the two techniques have contributed the most in structure determinations. However, as high-resolution cameras, sample production, and data processing techniques improve, cryo-EM will take on a more dominant role for structural studies in the future.

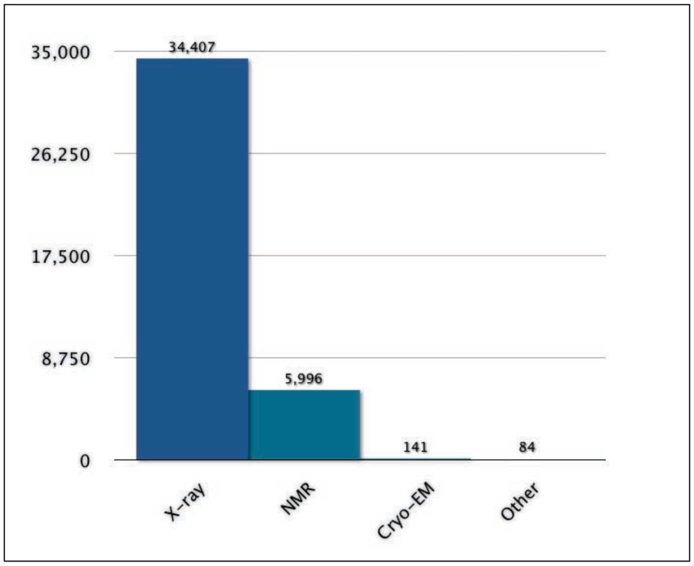


Figure 1. Distribution of Protein Data Bank entries by technique.

As with any analysis of a biological system, many of the technical challenges encountered are often related to the biological complexity of the study. Current trends are increasingly moving away from the isolated study of individual components in biological systems toward a systems biology approach, in other words, an attempt to understand how the various components act in concert to affect biological processes. With the increased complexity of the systems under study come new challenges for the analysis of these systems. A comprehensive discussion of analytical concerns is beyond the scope of this paper. However, it is appropriate and important to dig deeper specifically around computational requirements in modern structural biology settings since a significant portion of a structure determination depends on the computational aspect of a problem. This includes computational power, operating environment, and software and hardware necessary for 3D structure studies. The remainder of this article will focus on the computational requirements for empirical methods for structure determination with a specific focus on X-ray crystallography, NMR, and cryo-EM.

Computational Requirements

X-ray crystallography, NMR, and cryo-EM represent distinct methods for structure determination, yet the fundamental computational requirements of each are similar, as are the computing environments of the researchers using these techniques. Following data acquisition, each technique requires significant computational power for data processing, storage, retrieval, and archiving. Additionally, each technique generates large amounts of information that need to be visualized in a variety of ways as part of the structure determination and analysis.

UNIX

UNIX Features

- Command-line interface
- gcc/gfortran
- configure/make
- Perl
- Python/PyObj-C
- Ruby/RubyCocoa
- Tcl/Tk
- bash/csh/tcsh
- rsync/awk/grep, etc.
- X11

UNIX is an important feature of any operating system used for structural biology work for the simple reason that most structural biology applications were originally developed on UNIX-based computers. UNIX-based systems have historically been used in the sciences owing in part to the fact that UNIX (and UNIX-like) operating environments drive most hardware that is designed for numerically intensive calculations.

Development of modern structural biology applications continues on UNIX-based systems. Much of the software used in biological structure determinations is developed by academic researchers and provided free for use in academic, government, and nonprofit work. As a result, some key features of UNIX or any platform to be used for structural biology applications development are cross-platform compatibility to enable a broad user base, support for industry-standard graphics networking and operating system protocols, all of which help reduce the development and support effort, and finally, developer tools like the GNU compiler collection (GCC) and associated utilities that are free and readily available on every major platform[3]. UNIX also provides a number of tools, commands, and utilities that are useful in constructing application workflows. These workflows often consist of a combination of several smaller specialized applications that can be combined via shell scripts or pipe commands into a single tool. The combined applications are more powerful and utilitarian than the individual component programs.

Performance

The types of problems addressed in structural biology are computationally and graphically intensive. As the size of the biological system being analyzed increases, so do the demands on the hardware for processing and viewing the data. While the computational methods used for structure determination vary, each has the need to complete numerically intensive calculations. Systems that provide, either in hardware or software, mechanisms for processing large amounts of data as rapidly as possible are, arguably, the most important component in the selection of a system for structural biology work. The primary elements for a system are high throughput/bandwidth (both processing and data shuttling), large CPU caches, optimized mathematical and numerical libraries (such as for performing FFTs and Linear Algebra operations), and the ability to hold massive amounts of data in memory.

64-Bit Addressing

System Features

- 32-bit/64-bit compatibility
- Floating-point performance
- High memory bandwidth
- Large CPU caches
- Large memory support (64-bit)
- Multiple CPUs
- Optimized numerical libraries
- Performance tools
- Vector processing unit

In the case of X-ray crystallography, memory requirements are typically at a maximum during the data processing phase; the primary data is then reduced to a smaller footprint for subsequent steps. In contrast, cryo-EM based methods continually need access to the primary data, the images collected during data acquisition, throughout the structure determination. Often the data is accessed from disk at program launch and for performance reasons needs to be held in memory during the course of a calculation, which can sometimes take days or weeks.

Although individual images are relatively small (often less than 1MB), the number of images required for a 3D reconstruction can be in the tens to hundreds of thousands depending on the target resolution. 64-bit addressing increases the upper limit of addressable memory per process, such that a single process can now theoretically access up to 18 exabytes of memory. More realistically, it is not uncommon to configure systems with 4–16GB of memory, and in such a system a single process would have access to all of the memory resources on a 64-bit machine. These factors make native support for 64-bit addressing essential for maximum performance in studies where large image sets are used. Without 64-bit addressing, large calculations must either be reduced in size, worked on in smaller pieces, or alternative methods for analysis employed, all of which add complexity to the overall process.

Graphics

Graphics Features

- 3D stereo graphics
- High-resolution monitor support
- Multiple displays
- Multiple video cards
- OpenGL/GLX/GLUT
- Stereo-in-a-window

The goal of a structure determination is to build a 3D representation of molecules that can be manipulated and analyzed. As the complexity of biological systems being studied increases, so do the requirements on the graphics hardware to display and interact with that data. Typical structures contain anywhere from 500–5000 atoms. Structures consisting of 20,000-40,000 atoms (or more) are becoming increasingly commonplace as more complex biological systems are studied.

For most simple viewing operations (e.g., single display, moderate screen resolution, average molecule size), the majority of basic consumer graphics cards are sufficient. However, the extensive amount of time spent working at graphics workstations necessitates use of technologies that maximize the usable desktop footprint and minimize fatigue on the eyes. These can include driving large, high-resolution graphics displays; multiple displays to increase visual "real estate" and thereby enabling side-by-side comparisons of structural or sequence/homology information; extremely fast graphics cards capable of handling large vertex counts (such as in viewing molecular surfaces); and the ability to transform the onscreen two-dimensional data into a format for viewing that is easier to interpret, such as 3D stereographic rendering.

A variety of methods exist for generating 3D stereo visuals on a 2D screen. Methods such as depth queuing and slabbing are traditionally implemented for assessing z-depth information. When combined with other 3D stereoscopic rendering techniques, a visual representation of the view port is produced that more closely mimics an actual 3D object from the viewer's perspective. Almost all complementary methods for enhancing the 3D view employ the use of specialized color or synchronizing glasses.

3D stereo-in-a-window support[4–5] is one technology that is becoming extremely common in the visual assessment of data onscreen. The primary benefit of stereo-in-a-window versus full-screen stereo, as an example, is that only the window being viewed is rendered in 3D. This leaves other windows, such as those that contain sequence information or program control menus, in 2D for side-by-side viewing or manipulation.

In order to use 3D stereo-in-a-window, applications must be enabled to display information in a manner consistent with the 3D stereo graphics hardware. The most common hardware-based method in structural biology settings is via infrared emitters and stereo goggles that deinterlace left/right pairs of the content being viewed on screen[4–5]. Stereo graphics enabled software and hardware, while not absolutely required, greatly simplify the analysis of electron density maps (which are essentially large 3D meshes), and the fitting of atomic information into those maps, during the building and analysis phases of a structure determination.

Interoperability/Networking

Networking Features

- NIS/LDAP/Kerberos
- NFS/SMB/Samba
- DHCP/wireless
- SSH/SFTP/SCP
- Software firewall
- Apache/Tomcat
- Printer and file sharing
- Backup (e.g., rsync)
- Remote management

There are a number of additional requirements that exist beyond the typical considerations about CPU, memory, and graphics performance. These requirements fall out of the need to build, maintain, and support computational infrastructure. For example, data acquisition computing systems are not necessarily the same as those used for processing and analysis; the systems used to drive data acquisition hardware are often dictated by the manufacturer of that hardware. Additionally, not all of the tools and online resources that are required may work together on the same platform. Any platform being used must be able to work in a heterogeneous computing environment. Support for standard file-sharing protocols such as NFS and networked authentication via NIS or LDAP ease the burden on system administrators and users by allowing files to reside in central locations that can be accessed universally across a variety of different systems, while providing a single set of credentials for authentication.

Data Storage and Management

Storage Features

- Cross-platform mountable
- Large storage on disk
- Redundancy
- High availability
- Backup to alternate media
- Upgradeable

Persistent data storage is a continual problem in scientific settings, and structural biology is no exception. Single projects can generate hundreds or thousands of individual files. Depending on the techniques involved, individual data files can be anywhere from a few kilobytes to several gigabytes. There is not only a need to have access to the data throughout the course of a structure determination, but funding organizations such as the National Institutes of Health mandate that all primary data be accessible for a period of at least 10 years. This requirement applies not only to data obtained on the bench, but computer-generated data as well. Indeed, the management of collected data represents one of the biggest shortcomings of the computational aspect of structural biology projects. The methods employed for data storage have changed somewhat as technology has improved. However, no clear system for indexing and organizing primary data has been developed or universally accepted.

Common methods of storing data in structural biology settings are backup to tape and storage on disk. Given the current costs, sizes, and options, hard disk based storage is becoming an increasingly popular method for persistent storage. The ability to purchase terabytes of disk space, off the shelf, as preconfigured RAID devices provides a significant amount of disk space for access and archival purposes. Further backup to an alternate media format, again such as tape, or perhaps optical disk, provides additional redundancy of that data.

A second aspect of data storage that needs consideration is the cataloging and management of digital information. There currently exists no standard method that is employed across laboratories or institutions for indexing primary data acquired in structural biology. Whereas the final processed data and structure files are deposited to central databases such as the Protein Data Bank[2], the primary data can exist anywhere, in a variety of formats and often labeled in a manner that is only obvious to the original researcher.

While a void clearly exists in data management, there are a number of applications and tools available that may be useful for cataloging data in a standard method that at the very least will make the information accessible to other researchers in the same laboratory. As an example, the open archive initiative DSpace is being developed with the goal of standardizing digital research content storage, indexing, and redistribution. Although just one example, projects such as DSpace may represent one possible means of using open source software, with an open standard, to develop a platform-neutral system for cataloging and archiving data in a manner consistent with the requirements set out by the funding agencies.

Cost of Ownership

All too often, the term "lower cost of ownership" is used as a synonym for "cheaper." Hardware and software vendors try to expand on the concept of cost of ownership by providing metrics for the true cost of their products over their effective lifetimes. These metrics, such as reliability, performance, security, and power consumption, can be useful in making purchasing decisions. However, these metrics are often only valid for specific applications or a in a specific setting that may or may not represent the intended usage.

For scientific computing, the prevailing metric, performance per dollar, is still the most heavily weighted factor. This is mostly due to the fact that scientific computation resources are dedicated to a few target applications, and raw "number crunching" throughput is the most important consideration. In recent years, a new metric has come into common use: performance per watt. While there are valid reasons why power consumption is an important factor for consideration, especially when dealing with large data centers or computational clusters, in most structural biology settings, power consumption is at best a secondary thought.

From an end user's perspective, another metric of great importance is usability per dollar. Although subjective and biased heavily toward the specific application and user preference, it is nonetheless a significant component in determining how useful a system will be. Systems that ship with the greatest number of usable, and useful, features offer the best value not only from an economic standpoint but also from an efficiency standpoint. The more useful features for a given purpose that come prepackaged with the system, the less time the user has to spend finding alternatives or moving from system to system to complete a specific set of tasks.

Finally, it is important to ask the question, "How much is your time worth?" This is a valid consideration from the user's, developer's, and administrator's perspective. Unfortunately, the economic impact of inefficiency isn't typically considered in making a purchasing decision, probably due to the difficulty in quantifying the actual numeric cost. However, consideration for the time spent configuring, installing, and updating systems; switching from system to system to accomplish an individual task; or simply dealing with "secondary" issues that often arise from basing purchases solely on absolute dollar amount should not be dismissed. The primary purpose in any scientific computing purchase is to perform science, and time spent dealing with issues ancillary to the actual task is time not spent on research. Ease of use and reliability are just as important in this regard as is performance, especially when the system is serving multiple roles, such as a desktop and computation system.

To summarize, there is a common misconception that the inherent value of a system is dictated by purchase price alone. Rather, the true cost of ownership is a complicated mix of all of the issues discussed in this section. Systems that possess many of the features described above, thereby simplifying the overall process of using, maintaining, and managing these systems, tend to become dominant platforms for structural biology work, particularly in environments where there is a lack of dedicated support staff. A brief overview of some technology platforms that provide many of the features discussed above is presented in the next section.

Technology Platforms

A variety of computing platforms have been used for structural biology work over the past few decades. During the last 10 to 15 years, two platforms have dominated in this field, namely SGI and Linux-based systems, each filling a particular need that existed at the time. A brief description of some of the useful features of each platform and some of the problems each has solved are presented, followed by a detailed presentation of some of the features present in Mac OS X that make it an ideal platform for structural biology work.

SGI

SGI IRIX Features

- Unified package
- 64-bit
- Integrated GPU
- High performance
- Multiple CPUs
- Shared memory

SGI systems provided a number of important features for structural biologists that were difficult to find in other systems in a single package, in particular, the availability of a standardized OS environment (IRIX), high-performance integrated graphics and CPU design, a shared memory architecture for multiple CPUs, and one of the first standard 64-bit computing environments. These systems often provided high-performance numerical libraries, fast memory subsystems, and fast internal hard disk storage.

Linux

Linux Features

- Fully open source
- Commodity hardware
- Open standards
- Modern/high-performance OS
- Modern language support
- Package management
- Free development tools
- Free productivity apps

As the performance per dollar of commodity desktop computers increased, Linux became increasingly popular in the mid to late 1990s and remains so today. Benefiting from rapidly developing hardware designed for traditional desktop computing and high-performance graphics cards for gaming, Linux expands on many of the features present in SGI systems. The primary benefit from Linux is that it is a completely free and open source operating system, allowing a large number of developers to improve the base OS and offer further enhancements to the overall computing experience. Additionally, Linux supports a variety of open standards, provides standard open source developer tools, can be installed and run on off-the-shelf hardware (allowing users to build their own systems), and offers good performance for the price. Linux has provided a variety of important options to researchers looking for alternatives to features only previously available with “Big Iron” solutions. Additionally, Linux has, to a certain extent, forced commercial software vendors to continually improve and reassess the features and directions of their own offerings.

Mac OS X

Mac OS X Features

- 3D stereo support
- Batch scheduling software
- High-end graphics cards and displays
- High-performance systems
- Modern language support
- Modern OS
- Multiple CPUs
- Optimized numerical libraries
- Multiple operating system support
- Simple management
- Unified package

More recently a third platform, Mac OS X, has emerged that provides many of the features that scientists require for their research in both software and hardware, while offering additional functionality that can streamline their general workflow. The software and hardware comprise the best of both SGI and Linux.

First, Mac OS X is built on a stable, reliable UNIX subsystem, a version of UNIX derived from BSD developed at the University of California at Berkeley. Additionally, Mac OS X allows UNIX and desktop applications to work side by side, supports both open source and commercial application development and execution, and provides a number of specialized tools that are useful for scientists, such as the X11 window manager, Tcl/Tk; support for scripting/interpretive languages like Perl, Python, and Ruby; and 64-bit support.

The operating system is engineered around standards-based technology and is easily integrated into mixed computing environments, offers a variety of graphical-based and command-line administration tools, and is relatively simple for nonprofessional system administrators to manage in well-defined environments.

Each copy of Mac OS X ships with highly tuned numerical and vector libraries as part of the Accelerate framework that takes advantage of special hardware features such as the AltiVec and SSE SIMD (vector) units on the previously shipping PowerPC and newer Intel based systems, respectively. The Accelerate framework will automatically determine the system type and select the appropriate function type for that architecture. A variety of system configurations are available that offer high-performance computing solutions, which began with the PowerPC G4 and G5. The current offerings from Apple continue this trend with the newer 64-bit Dual-Core dual-CPU Intel Xeon processors (based on the new Intel Core 2 microarchitecture), large CPU caches, a fast front-side bus, and support for DDR2 RAM.

A number of features that ship with many of the systems increase the performance per dollar spent while minimizing overhead. First and foremost, whereas with Linux there exist a number of OS variations, Mac OS X is released as a single version and is the same operating system from client to server, reducing the number of possible conflicts in larger computing environments. Mac OS X systems are easily scaled either on the desktop or as a clustering solution using the Xserve server with either Xgrid[6] or with third-party solutions like Sun Grid Engine[7] or OpenPBS[8]. Because all new systems ship with multiple CPUs (or CPU cores), true multitasking is possible. Scientists can have one aspect of a project running at full speed in the background while performing other tasks interactively either on their desktop system or even with a mobile computer while traveling. And in the case of multithreaded applications, all of the system resources can be utilized simultaneously.

As of Mac OS 10.4.3, it is now possible to work in a 2D and 3D environment simultaneously due to support for 3D stereo-in-a-window on workstations that have the Quadro FX 4500 graphics card and stereo graphics hardware, such as CrystalEyes from Stereo Graphics Corporation[4]. Like many graphics cards that ship on the professional systems, the Quadro FX 4500 is capable of driving two displays simultaneously on a single system, consisting of two LCD displays, two CRT displays, or one LCD and one CRT display. In the case of a stereo graphics workstation, the latter combination is required.

The ability to drive multiple displays from a single computer offers a number of distinct advantages. First, the Apple Cinema Display provides high pixel density, high-resolution viewing (up to 2560 x 1600 pixel resolution on the 30-inch displays). Second, since it is not uncommon to have multiple files, projects, or windows open simultaneously, the extra “real estate” afforded by either a single large display or multiple displays minimizes the amount of window rearrangement required and provides ready access to information that may be needed throughout the workday. Additionally, Mac OS X offers native support for a number of common image formats (such as JPEG, PNG, GIF, and TIFF, to name a few) as well as native support for PDF and vector-based graphics.

With Mac OS X and Macintosh systems, both the operating system and hardware come from a single source. Therefore, software and hardware conflicts rarely occur. In addition, a unified package from a single vendor provides a lower cost of ownership by minimizing administration overhead, simplifies ease of use (as there is only one OS to learn and configure), and offers a simple, yet elegant, desktop computing experience while not sacrificing the underlying power of the UNIX-based portion of the OS. In terms of data storage, retrieval, and archiving, Apple provides simple add-on solutions with some of a lowest cost/terabyte of storage via Xserve and Xserve RAID.

Finally, with the switch to Intel-based systems on Mac OS X, two additional functionalities now exist that provide access to software that might not yet be available on the platform. First, Apple has provided Boot Camp[9] that allows users to dual boot Mac OS X and Windows on a single machine. Therefore, a mission-critical, Windows only application is no longer a reason to choose a PC over a Mac. Both operating systems can run on a single piece of hardware at fully native speeds.

The second is the ability to virtualize alternative computing environments using applications such as Parallels Desktop[10]. Parallels allows users, without rebooting, to run multiple (virtual) instances of a number of operating systems, including Windows, all Linux variants, FreeBSD, and OS/2, simultaneously and at near native speeds. In other words, users can run any number of operating systems on a single piece of hardware, taking advantage of software applications that may not yet run on Mac OS X. These technologies, combined with the features discussed above, mean that Mac OS X practically provides a turnkey solution for scientists, and especially for structural biologists, who need high-performance, reasonably priced, full-featured, versatile systems, in an easy-to-use, well-engineered, major manufacturer supported package.

Case Study: Workflow for X-ray Crystallography

The goal in any structure determination performed using X-ray crystallography is to obtain a "picture" of the electron density surrounding a molecule that is of high enough quality that a 3D atomic model, representing the atoms in the molecule of interest, can be built. The final atomic model provides a snapshot of the molecule as it exists in the crystal lattice and presumably represents at least one state in which the molecule can exist within a biological context.

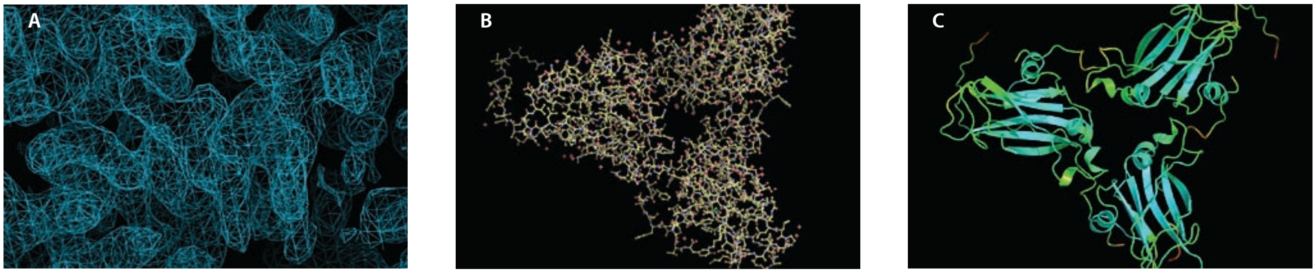


Figure 2. Image of an electron density map (A), final model in ball-and-stick representation (B), and a cartoon representation (C). Map and model courtesy of Jeff Speir. Speir et. al., Journal of Virology, Apr. 2006, p. 3582–3591.

As mentioned earlier, X-ray crystallography is the most common technique employed for the determination of macromolecular structures. This section will focus on some of the common tools and present a general computational workflow for X-ray crystallographic studies using Mac OS X. Because the stages prior to data processing can vary significantly between projects, this workflow will begin by assuming that the bench top steps have been completed and usable data has already been collected at either a home or synchrotron X-ray source.

The workflow presented here is broken down into five stages:

1. Data Processing and Reduction
2. Phasing
3. Model Building, Phase Improvement, and Refinement
4. Visualization and Analysis
5. Final Validation and Publication.

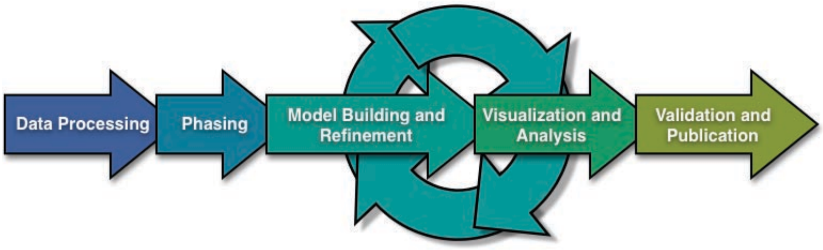

Data Processing
Phasing
Model Building and Refinement
Visualization and Analysis
Validation and Publication

Figure 3. Schematic of the workflow.

In this workflow, examples of open source and commercial software available on the Mac OS X platform will be highlighted. Please note that because this article is not an exhaustive review, there are a large number of applications not shown here that can be used at various stages of the workflow and that offer additional features and functionality that could be used as alternatives to the applications described in this paper. The examples provided have been chosen because they are the more commonly used applications. These applications run under the UNIX environment, the native Mac OS X windowing system, or the X11 windowing system. Almost all of the software can be downloaded as source, precompiled binaries or installed via package management systems such as Fink[11].

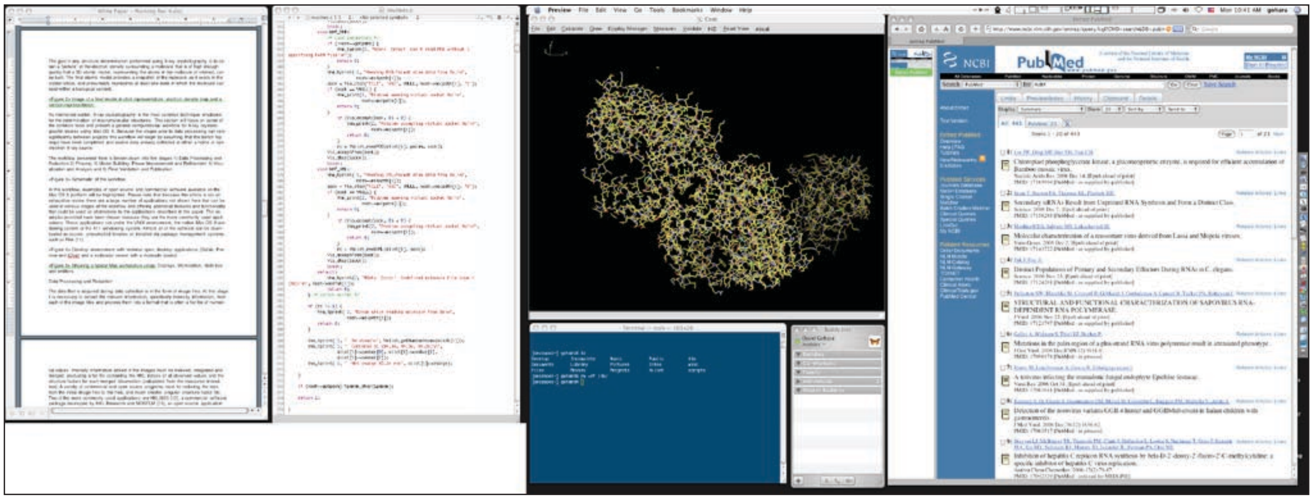


Figure 4. Desktop/UNIX computing environment.

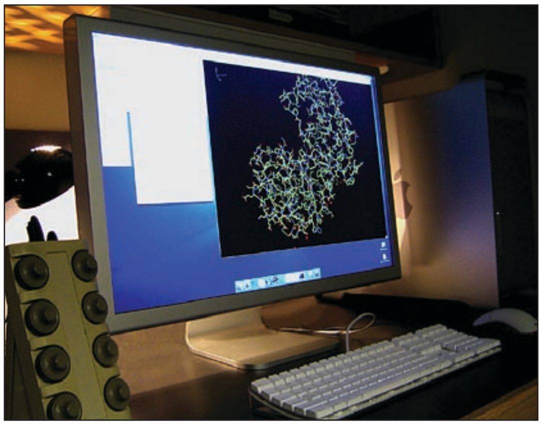


Figure 5. Workstation for X-ray crystallography.

Data Processing and Reduction

Data Processing Applications

- HKL2000
- MOSFLM
- XDS

The data that is acquired during data collection is in the form of image files. At this stage, it is necessary to extract the relevant information, specifically intensity information, from each of the image files and process it into a format that is often a flat file of numerical values. Intensity information stored in the images must be indexed, integrated, and merged, producing a flat file containing the HKL indices of all observed values and the structure factors for each merged observation (calculated from the measured intensities). A variety of commercial and open source programs exist for reducing the data from the initial image files to the final, and much smaller, singular structure factor file. Two of the more commonly used applications are HKL2000[12], a commercial software package developed by HKL Research, and MOSFLM[13], an open source application currently being developed by Harry Powell at the MRC-LMB, Cambridge. Each of these programs can be run via an X11 GUI, at the command line interactively, or using a script. A new interface for MOSFLM has been developed that uses Tcl/Tk[14].

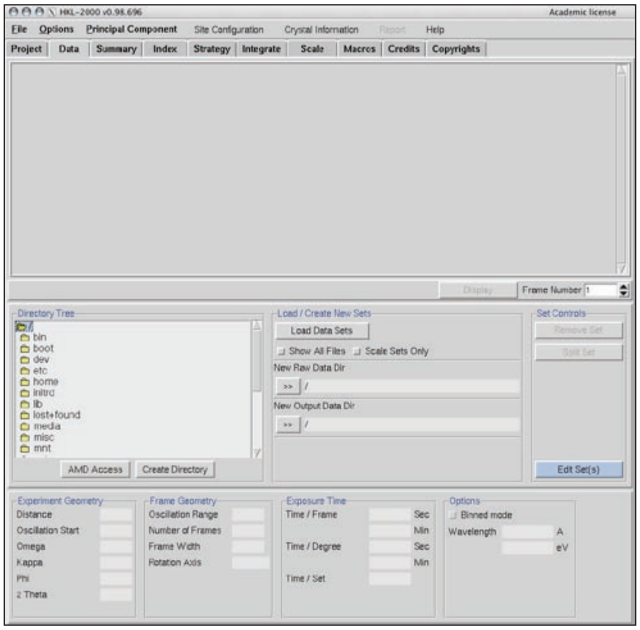


Figure 6. HKL2000 GUI.

Provided with a stack of images containing the data to be processed and some initial information, both applications will attempt to determine the basic crystal cell parameters for the data set and autoindex, integrate, and merge the data for the user. However, this process only provides one-half of the information required for a structure determination. The first piece of information, the amplitudes, provides the intensity of the x-rays scattered by the crystal. The second component, the phase information, provides the position of the scattered x-rays' maximum relative to a fixed position. The phase information is lost during data collection and needs to be reconstructed using alternate methods. A variety of techniques exist for obtaining initial phase estimates suitable for structure determination and include molecular replacement, heavy atom derivitization, and Multiwavelength Anomolous Diffraction (MAD). Each of these techniques and the software available for implementing each method are discussed in the following section.

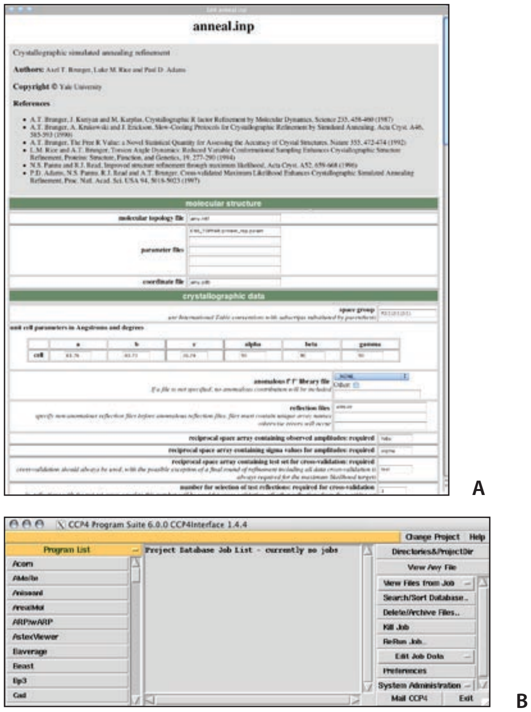
Phasing

Phasing Applications

- BnP
- CCP4
- CNS Solve
- Phaser
- Replace
- Shake-N-Bake
- Sharp
- Shelx
- Solve/Resolve

Arguably the most important information required for a structure determination is the phase information. In order to create a 3D reconstruction of the electron density surrounding the molecule (referred to as a "map"), an initial phase estimate must somehow be determined from other sources of information. This requirement is due to the fact that there currently exists no method to directly focus, and therefore image, X-rays, as you would focus light in a light microscope, for example. Without any phasing information, the amplitudes are essentially meaningless. Two primary methods for obtaining phase information exist: model phasing and experimental phasing.

In situations where the molecule of interest is related by functional or sequence homology to a molecular structure already determined, or where a previously determined structure is a subcomponent of the current molecule of interest, the previous models can often be used to provide an initial phase estimate for the structure determination. This technique, referred to as molecular replacement, requires only the newly acquired data and the PDB file of the "related" molecule, which can often be downloaded from the Protein Data Bank. There are a variety of applications that can perform molecular replacement, including CNS Solve[15], Phaser[16], and MOLREP[17] (part of the CCP4[58] suite of programs). Each program approaches the problem of molecular replacement differently, and where one application may fail to produce a suitable answer, another might succeed. All of these programs are run primarily from the command line, although there is a web interface for CNS Solve script editing and a Tcl/Tk-based GUI job manager (as part of CCP4, called CCP4i) for both Phaser and MOL-REP.



anneal.inp

Crystallographic simulated annealing refinement

Authors: Axel T. Brunger, Luke M. Rice and Paul D. Adams

Copyright © Yale University

References

- A.T. Brunger, J. Kuriyan and M. Karplus, Crystallographic R factor Refinement by Molecular Dynamics. Science 235, 458-460 (1987)
- A.T. Brunger, A. Krukowski and J. Erickson, Slow-Cooling Protocols for Crystallographic Refinement by Simulated Annealing. Acta Cryst. A46, 585-593 (1990)
- A.T. Brunger, The Free R Value: a Novel Statistical Quantity for Assessing the Accuracy of Crystal Structures. Nature 355, 472-474 (1992)
- L.M. Rice and A.T. Brunger, Torsion Angle Dynamics: Reduced Variable Conformational Sampling Enhances Crystallographic Structure Refinement. Proteins: Structure, Function, and Genetics, 19, 277-290 (1994)
- N.S. Pannu and R.J. Read, Improved structure refinement through maximum likelihood. Acta Cryst. A52, 659-668 (1996)
- P.D. Adams, N.S. Pannu, R.J. Read and A.T. Brunger, Cross-validated Maximum Likelihood Enhances Crystallographic Simulated Annealing Refinement. Proc. Natl. Acad. Sci. USA 94, 5018-5023 (1997)

molecular structure

molecular topology file
parameter files
coordinate file

crystallographic data

space group
unit cell parameters in Angstroms and degrees

	a	b	c	alpha	beta	gamma
cell	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

anomalous f' f'' library file
reflection files
reciprocal space array containing observed amplitudes: required
reciprocal space array containing sigma values for amplitudes: required
reciprocal space array containing test set for cross-validation: required
number for selection of test reflections: required for cross-validation

CCP4 Program Suite 6.0.0 CCP4Interface 1.4.4

Project Database Job List - currently no jobs

Figure 7. Interface for CCP4 (A) and the web interface for CNS Solve scripts (B).

The caveat to molecular replacement is that it requires that a model with sufficient homology already exist in order to obtain the initial phase estimate. In situations where no previous structural information is available, experimentally determined phase estimates will need to be obtained. Experimentally determined phases, while sometimes more difficult to generate, offer the benefit of not introducing model bias immediately into the generated electron density maps. As mentioned above, two common techniques for obtaining experimentally derived phases are heavy atom derivitization and MAD. Again, the specifics of each technique are beyond the scope of this article, but it should suffice to say that, when successful, each method is extremely powerful in helping to obtain phases of the molecule of interest.

For experimentally determined phases, a number of programs exist that are capable of determining an initial phase estimate. In addition to the program CNS Solve, others, such as Shelx[18], BnP[19], and MOLREP are commonly used for this task. Another program suite capable of performing a variety of phasing tasks is Solve/Resolve[20–22]. In the case of Solve/Resolve, and with good data, several steps of the structure determination including phasing, map calculation, and initial model building can be completed automatically with little to no user intervention. Again, all of these programs are run from the command line via scripts, with the exception of BnP, which has a Java front end that executes command line based executables on the back end. BnP can even interface with Solve/Resolve or ARP/wARP[23] or graphics programs such as O[24].

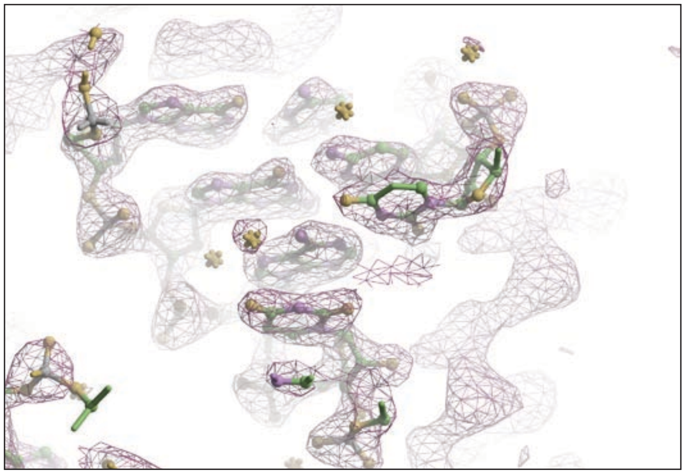


Figure 8. Electron density map. Map courtesy of William Scott. Martick and Scott. Cell, 126:309-320 (2006).

Model Building, Phase Improvement, and Refinement

Building/Refining Applications

- ARP/wARP
- BUSTER-TNT
- CCP4
- CNS Solve
- Coot
- DINO
- MolMol
- O
- PMV
- Refmac
- Spock
- UCSF Chimera
- VMD
- X-PLOR

Following the generation of an initial phase estimate, an electron density map is calculated. At this initial stage, a number of important technologies discussed earlier become extremely important. Molecular visualization is a key first step to this process; a significant amount of time is spent looking over electron density maps in molecular viewer applications. The primary goal at this stage is the identification of atomic features of the molecule such as amino acid side chains, nucleic acid features (if present), backbone carbon traces, or in the case of low-resolution information, identifying domains of the molecule(s).

In the absence of a starting model, the initial map will appear as a series of blobs to the untrained eye (and, depending on the resolution, the trained eye as well). Looking at the initial map on a 2D screen, it's often difficult to extract features of the molecule that can be used for placing atoms initially. The availability of 3D graphics workstations alleviates some of this problem by providing additional depth information that would be unavailable otherwise. Molecular viewer programs such as O, Coot[25], and UCSF Chimera[53] can be used to read the electron density maps in a number of common formats (O, CCP4, CNS) for initial evaluation and subsequently used for model building and validation. Electron density maps can be improved using a variety of programs such as DM[27] or Solomon[28].

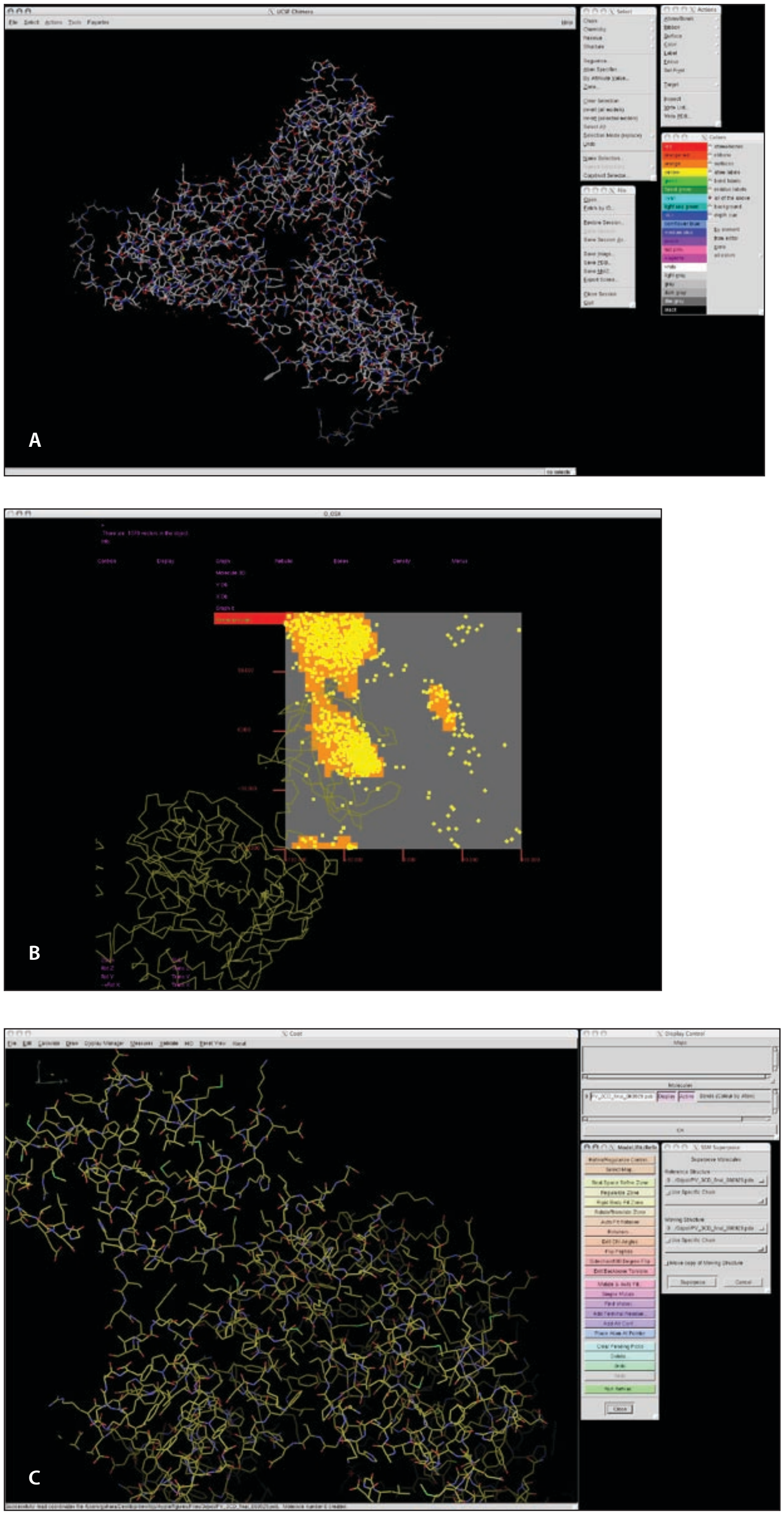


Figure 9. Interfaces for Chimera (A), O (B), and Coot (C).

Provided that the initial map is of sufficient quality to begin placing atoms, model building proceeds. The model building stage is the most time-consuming portion of a structure determination since the process of building and refining the model is an iterative process that is both labor intensive on the researcher and computationally demanding. Programs such as ARP/wARP can often be used to assist in map interpretation and automatic model building. As the model improves, the estimate of the phases improves as well.

The time spent in this phase of the project can be categorized into two domains. The first, the user component, is where the bulk of time is spent building in atoms, manually fitting the atoms to the electron density, and assessing that the molecular trace of the molecule makes chemical sense. In X-ray crystallography, two of the more commonly used programs for this task are O and Coot, although a variety of others exist as well. Each application has a number of features that aid in the overall process, including real-space refinement, real-time structure validation, and bond idealization. Both have been modified to support external hardware devices such as dials boxes under Mac OS X. Additionally, O and Coot support 3D stereo-in-a-window viewing capabilities.

The second time-consuming portion of this stage resides in the computational domain. Following a round of building atoms into the model, the atom positions are refined (fit to the experimentally observed data). Refinement is extremely computationally intensive, as a number of parameters are often minimized simultaneously, and depending on the size of the structure being built, a single refinement calculation can take anywhere from hours to days. Again, programs such as CNS Solve or the CCP4 suite of programs, which provides applications such as Refmac, can be used. Additionally, applications such as ARP/wARP can also be used at this stage.

To minimize the amount of time researchers spend waiting for a calculation to finish, a number of the CPU-intensive applications have been optimized for Mac OS X, taking advantage of a variety of technologies that come prepackaged on each system. Where possible, applications have been modified to use the FFT and BLAS libraries that are distributed as part of the Accelerate framework in Mac OS X. Applications are reengineered to call faster versions of performance-demanding math functions; custom routines are written that utilize the vector processing units (such as AltiVec and SSE) and sometimes modified to exploit the presence of multiple CPUs simultaneously using OpenMP, pthreads, or MPI-based multithreading techniques. Because a number of these applications are open source, the modifications can be made available to other researchers for use in their own work either as precompiled binaries or via source, when appropriate.

For applications that are not amenable to fine-grain optimization strategies, the UNIX subsystem in Mac OS X provides mechanisms for job distribution/automation via shell or interpretive language scripting or coarsely processed in a distributed manner across multiple systems by batch methods using Xgrid or Sun Grid Engine. A number of applications function primarily by providing a graphical environment that acts as a wrapper for a set of scripts and command-line applications to drive program execution. As an example, the program interface CCP4i functions in this manner for the CCP4 suite of programs.

Following a round of model building and refinement, validation of the output model is performed by assessing "difference" electron density maps (e.g., fo-fc) using CCP4 to generate the maps and Coot or O for visualization. Composite omit electron density maps, which help reduce bias in the electron density introduced by the atoms being built into the model, can be constructed using CNS Solve (either serially or in a distributed manner). Difference and composite omit electron density maps provide a qualitative visual assessment of the progress of a structure refinement. A quantitative analysis of bond angles and torsions via Ramachandran plots can be accomplished with programs such as PROCHECK[28-29], O, and Coot.

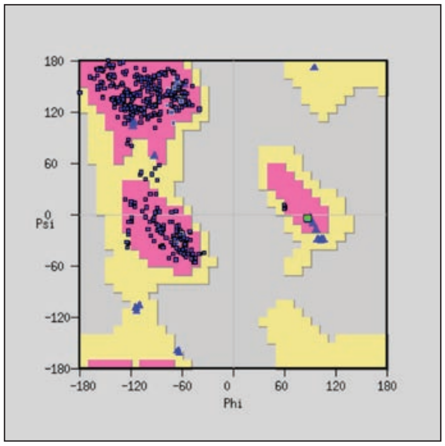


Figure 10. Ramachandran plot generating in Coot.

Visualization and Analysis

Analysis Applications

- APBS
- AutoDock
- CCP4mg
- Coot
- Delphi
- DINO
- DOCK
- MegaPOV/POV-Ray
- MolMol
- Molscript/Bobscript
- Naccess
- NUCPLOT
- O
- PMV
- PROCHECK
- PyMOL
- RasMol
- Raster3D
- Ribbons
- Spock
- UCSF Chimera
- VMD
- Whatcheck

As described throughout this article, data visualization is an important component in any structure determination. One useful technology is 3D stereo graphics support, which aids in building molecular models. Support for multiple displays is also beneficial. There is yet another aspect to visualization that occurs once the structure is mostly complete, and it ties in with the notion of analyzing the structure in the context of genetic, biological, or biochemical data. Often the analysis will focus on determining where mutations, molecular interfaces, electrostatic interactions, and protein/ligand interactions occur. This analysis is key to understanding how a molecule or molecular complex functions or for ascertaining how a drug compound works in the context of a therapeutic drug target.

A variety of tools exist that help to quickly visualize and report back important information about a structure that a cursory examination of the molecule might not reveal. Programs such as APBS[30] and DelPhi[31], when used in conjunction with molecular viewer applications PyMOL[32], VMD[33], or PMV[34], are useful for ascertaining the charge distribution across the surface of a molecule. Protein/ligand interactions (e.g., a putative drug target in the active site of an enzyme) can be analyzed using docking applications such as UCSF Dock[35] or Autodock[36]. Additionally, information about the structure itself can also be extracted. The programs Surfrace[37], Naccess[38], and ArealMol[39] are useful in determining the extent of protein/protein interactions (e.g., in multiprotein complexes), or NUCPLOT[40] can be used for generating diagrams of protein/nucleic-acid interactions that can be quickly used to determine what atoms of a protein are contacting a nucleic acid substrate or small molecule. Diagrams created by NUCPLOT can also be used for publications.

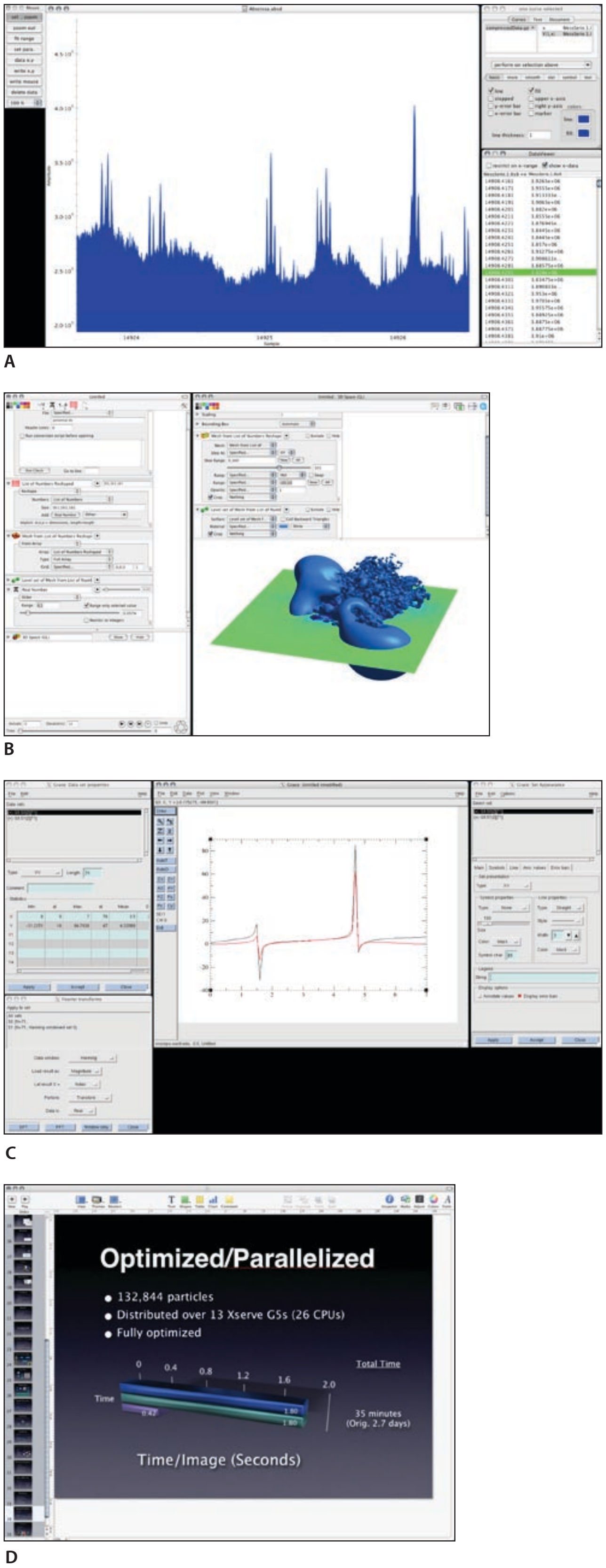
A number of applications not only create static images, but also 3D animations of the structure. An analysis such as this serves a descriptive purpose, as it allows viewing of the structure from multiple angles. The programs PyMOL and Chimera can generate a series of images via scripting or plug-in interfaces that can then be combined using QuickTime Pro[41] into a movie file that is then easily shared with others. When done properly, animations are often more revealing than a static picture and can expose important aspects of a structure that would otherwise be lost using a traditional medium.

Final Validation and Publication

Of course, the ultimate goal in any structure determination is the presentation of the results, usually via publication in a peer-reviewed journal. While the requirements for publication vary significantly from journal to journal, there are a number of steps that are assumed to be taken prior to publication, and they often begin with submission of the structure (and possibly structure factors) to one of the online structure repositories such as the Protein Data Bank.

The second and more intellectually interesting aspect is obviously the manuscript preparation. There are a variety of tools that are available to assist in this step, owing in large part to the fact that Mac OS X provides both the UNIX and desktop computing environments in a single package. Common productivity applications such as Microsoft Word and Excel and Pages all run under Mac OS X. Additionally, open source document preparation tools such as LaTeX[42] and OpenOffice[43] are useful for high-quality typesetting in conjunction with bibliography applications like EndNote (for Word)[44] or the open source program BibDesk[45].

There are a number of plotting and data analysis programs that are useful for generating charts and tables, including Aabel[46], DataTank[47], Kaleidagraph[48], and IGOR Pro[49]. And a variety of free applications such as GNUplot[50], Grace[51], and Abscissa[52] exist as well. And, of course, applications like PowerPoint and Keynote are useful for generating not only presentations, but also for creating schematic diagrams for publications.



Optimized/Parallelized

- 132,844 particles
- Distributed over 13 Xserve G5s (26 CPUs)
- Fully optimized

35 minutes
(Orig. 2.7 days)

Time/Image (Seconds)

Figure 11. Data analysis, graphing, and presentation packages. Abscissa (A), DataTank (B), Grace (C), and Keynote (D).

An important aspect of manuscript preparation is the preparation of figures. Again, there are a variety of applications and tools that run on Mac OS X that are useful for this purpose. Programs such as Adobe Illustrator and Adobe Photoshop can be used to produce vector and raster graphics files. Aperture can also be used to manage and generate cropped images of figures without altering the content of the original image files (this is particularly important for experimentally derived figures such as accompanying electrophoresis gels, where journals have become extremely wary of photo manipulation). Chimera, Molscript/Raster3D[54-55], PyMOL, CCP4mg[56], and Ribbons[57] are all capable of generating cartoon representations of biological molecules that highlight specific features of the structure. These applications produce high-quality renderings that support shading, surface displays, and transparent surface overlays for publication.

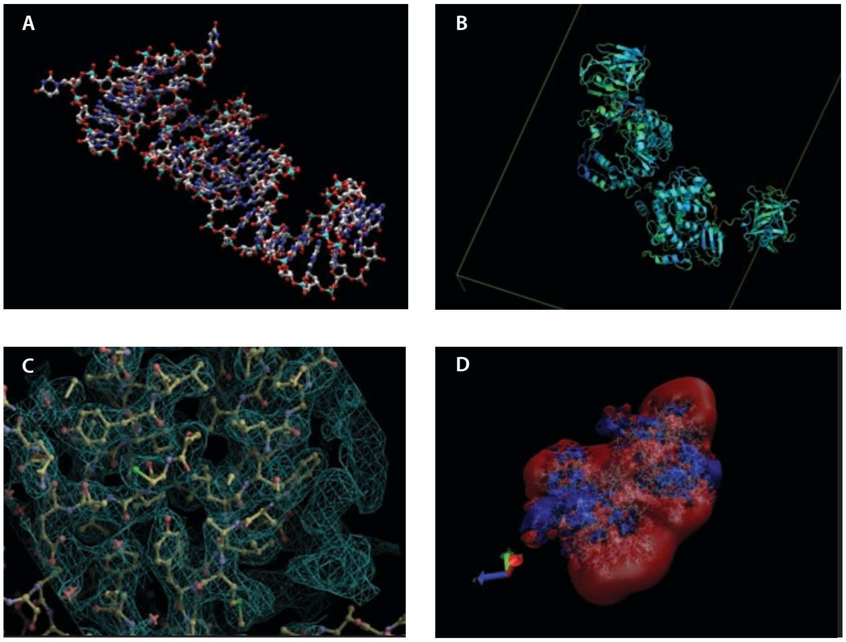


Figure 12. Image generation programs. Images were created using Chimera (A), PyMOL (B), Raster3D (C), and VMD (D). Data for panel A courtesy of William Scott. Martick and Scott. *Cell*, 126:309-320 (2006).

Finally, it should be noted that many of the programs described above are capable of reading and/or writing out files into formats supported by the others. This is important, as interoperability and compatibility are critical in a collaborative environment and to ensure compliance with journal requirements for publication.

Conclusion

With regard to structural biology, Mac OS X is becoming an increasingly popular platform for macromolecular structure determinations. Workflows are greatly simplified in large and small computing environments by allowing all of the required functionality, such as computationally demanding tasks and desktop applications, to run on a single system. A number of additional technologies like Xgrid and highly optimized numerical libraries offer simple, yet powerful solutions for computationally demanding tasks. Applications that are written, optimized, or extended to use these features can be easily distributed to other users, as these technologies come as part of the system package.

As outlined in the preceding sections, Mac OS X on Apple systems provides a nearly complete operating environment for the determination of biologically relevant macromolecular structures. Utilizing several metrics, Mac OS X provides excellent value for the cost. In terms of system configuration and usability, Mac OS X is nearly a turnkey solution for structural biology applications, owing in part to the combination of an underlying UNIX subsystem, high-performance system design, integrated hardware and software features, high-end graphics, standard desktop and networking functionality, and ease of configuration and use. The features integrated into the operating system and hardware make it an ideal platform not only for structural biology, but for scientific work in general.

References

1. Marti-Renom, M.A., Stuart, A.C., Fiser, A., Sanchez, R., Melo, F., Sali, A. (2000). "Comparative protein structure modeling of genes and genomes." Annu Rev Biophys Biomol Struct 29: 291–325.
2. http://www.rcsb.org
3. http://gcc.gnu.org/
4. http://www.reald.com/scientific/
5. http://www.nuvision3d.com/
6. http://www.apple.com/macosx/features/xgrid/
7. http://gridengine.sunsource.net/
8. http://www.openpbs.org/
9. http://www.apple.com/macosx/bootcamp/
10. http://www.parallels.com/
11. http://fink.sourceforge.net/
12. Otwinowski, Z. and Minor, W. "Processing of X-ray Diffraction Data Collected in Oscillation Mode." Methods in Enzymology, Volume 276: Macromolecular Crystallography, part A, p. 307–326, 1997.
13. Leslie, A.G.W. (1992). Joint CCP4 and ESF-EAMCB Newsletter on Protein Crystallography, No. 26.
14. http://www.mrc-lmb.cam.ac.uk/harry/imosflm/index.html
15. Brunger, A.T. et. al. "Crystallography & NMR system: A new software suite for macromolecular structure determination."
16. McCoy, A.J., Grosse-Kunstleve, R.W., Storoni, L.C., and Read, R.J. "Likelihood-enhanced fast translation functions." Acta Cryst. (2005). D61, 458–464.
17. Vagin, A., Teplyakov, A. "MOLREP: an automated program for molecular replacement." Appl. Cryst. (1997) 30, 1022–1025.
18. Sheldrick, G. and Schneider, T. (1997). Methods Enzymol. 277, 319–343.
19. http://www.hwi.buffalo.edu/BnP/
20. Terwilliger, T.C. and J. Berendzen. (1999). "Automated MAD and MIR structure solution." Acta Crystallographica D55, 849¬861.
21. Terwilliger, T.C. (2000). "Maximum likelihood density modification." Acta Cryst. D56, 965–972.

22. Terwilliger, T.C. (2002). "Automated main-chain model-building by template-matching and iterative fragment extension." Acta Cryst. D59, 34–44.

23. Perrakis, A., Morris, R.M. and Lamzin, V.S. (1999). "Automated protein model building combined with iterative structure refinement." Nature Struct. Biol. 6, 458–463.

24. Jones, T.A. et. al. (1991). "Improved methods for the building of protein models in electron density maps and the location of errors in these models." Acta Cryst. A47 110–119.

25. Emsley P., Cowtan K. "Coot: model-building tools for molecular graphics." Acta Crystallogr D Biol Crystallogr. 2004 Dec;60(Pt 12 Pt 1):2126-32.

26. Cowtan, K. (1994). Joint CCP4 and ESF-EACBM Newsletter on Protein Crystallog-raphy, 31, p. 34–38.

27. Abrahams, J.P. and Leslie, A.G.W. Acta Cryst. D52, 30–42 (1996).

28. Laskowski, R.A, MacArthur, M.W., Moss, D.S., Thornton, J.M. (1993). "PROCHECK: a program to check the stereochemical quality of protein structures." J. Appl. Cryst., 26, 283–291.

29. Morris, A.L., MacArthur, M.W., Hutchinson, E.G., Thornton, J.M. (1992). "Stereo-chemical quality of protein structure coordinates." Proteins, 12, 345–364.

30. Baker, N.A., Sept, D., Joseph, S., Holst, M.J., McCammon, J.A. "Electrostatics of nanosystems: application to microtubules and the ribosome." Proc. Natl. Acad. Sci. USA 98, 10037–10041 (2001).

31. Gilson, M.K. and Honig, B. "Calculation of the Total Electrostatic Energy of a Macromolecular System: Solvation Energies, Binding Energies and Conformational Analysis." Proteins, 4, 7–18 (1988).

32. DeLano, W.L. "The PyMOL Molecular Graphics System" (2002). http://www.pymol.org

33. Humphrey, W., Dalke, A., and Schulten, K. "VMD—Visual Molecular Dynamics." J. Molec. Graphics, 1996, vol. 14, p. 33–38.

34. Sanner, M.F. "Python: A Programming Language for Software Integration and Development." J. Mol. Graphics Mod., 1999, Vol. 17, February. p. 57–61.

35. http://dock.compbio.ucsf.edu/

36. Morris, G.M., Goodsell, D.S., Halliday, R.S., Huey, R., Hart, W.E., Belew, R.K., and Olson, A.J. (1998). "Automated Docking Using a Lamarckian Genetic Algorithm and Empirical Binding Free Energy Function." J. Computational Chemistry, 19: 1639–1662. http://autodock.scripps.edu/

37. Tsodikov, O.V., Record, M.T., Jr., and Sergeev, Y.V. (2002). "A novel computer program for fast exact calculation of accessible and molecular surface areas and average surface curvature." J. Comput. Chem., 23, 600–609. http://monte.biochem.wisc.edu/~tsodikov/surface.html

38. http://wolf.bms.umist.ac.uk/naccess/

39. Lee, B. and Richards, F.M., J. Mol. Biol., 55, 379–400 (1971).

40. Luscombe, N.M., Laskowski, R.A, Thornton, J.M. (1997). "NUCPLOT: a program to generate schematic diagrams of protein-DNA interactions." Nucleic Acids Research, 25, 4940¬4945. http://www.biochem.ucl.ac.uk/bsm/nucplot.html

41. http://www.apple.com/quicktime/mac.html

42. http://www.latex-project.org/

43. http:// www.openoffice.org

44. http://www.endnote.com/

45. http://bibdesk.sourceforge.net/

46. http://www.gigawiz.com/

47. http://www.visualdatatools.com/

48. http://www.synergy.com/

49. http://www.wavemetrics.com/

50. http://www.gnuplot.info/

51. http://plasma-gate.weizmann.ac.il/Grace/

52. http://iapf.physik.tu-berlin.de/DZ/bruehl/

53. http://www.cgl.ucsf.edu/chimera/

54. http://www.avatar.se/molscript/

55. Merritt, E.A. and Bacon, D.J. (1997). "Raster3D: Photorealistic Molecular Graphics." Methods in Enzymology 277, 505–524. http://skuld.bmsc.washington.edu/raster3d/

56. http://www.ysbl.york.ac.uk/~ccp4mg/

57. http://www.cbse.uab.edu/ribbons/

58. "The CCP4 Suite: Programs for Protein Crystallography." Acta Cryst. D50, 760–763. http://www.ccp4.ac.uk/index.php