Published in final edited form as:

J Struct Biol. 2008 September ; 163(3): 224–228. doi:10.1016/j.jsb.2008.03.002.

Software Tools for Molecular Microscopy

Ross Smith¹ and Bridget Carragher²

¹*Department of Cell Biology, New York University School of Medicine, New York, New York*

²*The National Resource for Automated Molecular Microscopy, Department of Cell Biology, The Scripps Research Institute, La Jolla, California*

Abstract

In our role as the editors of a special edition of the Journal of Structural Biology published in 1996 and devoted to the development of software tools, we offer our view of past developments and future prospects in this area. The astonishing progress in computer hardware over the past decade has fueled a significant increase in computational power available for the solution of macromolecular structures. At the same time the relatively slow growth and development of the accompanying software reflects the difficulties of developing large, complex and very specialized analytical methods.

Introduction

The evolution of the field of electron microscopy applied to structural biology, and in particular that of cryo-electron microscopy (cryoEM), is intimately connected to the development of a wide array of complex software programs that support the required image processing and analysis functions. About twelve years ago we reviewed the available software packages (Carragher and Smith, 1996) in the context of the Journal of Structural Biology's first special issue on "Advances in Computational Image Processing for Microscopy" (Aebi et al., 1996). The significance of software tools to the field of cryoEM is reflected in the high citation rates of many of the papers that appeared in that special issue; five of the papers are amongst the top ten citations from JSB and twelve are in the top one hundred. Those papers have been acknowledged as part of the technical foundation of the developments in this area over the past decade. It is important to note that both the publisher and the members of the editorial board of the Journal of Structural Biology were exceptionally supportive, proactively promoted the publication of these critical technical papers, and have continued to support both technical software publications and regular special issues devoted to this topic (see for e.g. *J. Struct. Biol.*, **157**, (2007)).

In this review, on the occasion of the 50th Anniversary of the Journal of Structural Biology, we once again examine the state of the art in computational tools focused on structural studies using the electron microscope. In doing so we have naturally taken a retrospective look at the changes and progress that has been made by drawing on three sources as a starting point: a survey by Hegerl (1992) that summarized the software packages available at that time; a review of software tools written to accompany the Special Issue of 1996 (Carragher and Smith, 1996); and the continually updated Wikipedia pages¹ that were set up to accompany a more

Corresponding Author: Bridget Carragher, The Scripps Research Institute, 10550 North Torrey Pines Road, La Jolla, CA 92037, tel: (858) 784-9070; fax: (858) 784-9090; email: bcarr@scripps.edu.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

recent Special Issue on Software Tools (J. Struct. Biol., **157**, 2007). We acknowledge that this strategy has yielded a review that is far from exhaustive. We regret any errors of fact, or omissions of the mention of major contributors to the field.

Background

In the early 1990s, Reiner Hegerl, at the Max-Planck Institute for Biochemistry in Martinsreid, Germany, wrote “A brief survey of software packages for image processing in biological electron microscopy” (Hegerl, 1992). In that survey he reviewed 7 software packages that were then in use and had been documented in various publications. These included MDPP (Smith and Gottesman, 1996), SEMPER (Saxton et al., 1979), SPIDER (Frank et al., 1996), PIC (Trus et al., 1996), IMAGIC (van Heel et al., 1996), EM (Hegerl, 1996) and the MRC package (Crowther et al., 1996). It is interesting to note that all 7 of these packages appear to be in use today; 4 of them are described in the Wikipedia page listing currently available Software Tools for Molecular Microscopy. The MDPP, PIC and EM packages are still used by a few labs for selected projects, although the EM package has largely been absorbed into the underlying algorithms that now support the TOM toolbox (Nickell et al., 2005). The SEMPER package appears now to be used primarily by the geophysical community. In his review in 1992 Hegerl raised the issue as to whether there was a need for seven different program packages and whether the community would be better served by consolidating the existing packages into a modular and adaptable system supported by the entire community.

A few years later when the Special Issue on Software Tools (J. Struct. Biol., **116**, 1996) was put together, the number of packages had proliferated to the extent that they were divided into four categories: General Packages, Specific Packages, Application Tools and Visualization Tools. This same loose and somewhat arbitrary organization is continued today on the Wikipedia web site. The general packages category included all of the original packages in the Hegerl survey plus three additional packages, only one of which has survived to have a presence on the Wikipedia pages. A fairly large number of specialized packages and specific application tools had also appeared, many of them repackaged subsets of the general packages. Most of these have not survived to the present but have been replaced by new sets of specialized tools. The current Wikipedia pages for Software Tools for Molecular Microscopy at last counting included 12 General Packages, 13 Specialized Packages, 13 Application Tools and 6 Visualization Tools. One extra category appeared on the Wikipedia pages, under the catchall title of “Utilities” and the sole package in this category is the file format conversion program called “em2em”. With a total of over 40 software packages currently available to the fairly small and specialized community devoted to structure determination by molecular microscopy, it has become even more urgent to address the challenge, initially posed by Hegerl in 1992, of consolidating and simplifying software development for our community.

A Decade of Change in the Computer Industry

The extraordinary technical developments in the computer industry over the last decade are too extensive and complex to be analyzed in any detail here. Three striking changes that have impacted the development of software tools for molecular microscopy (as well as all other software of course) demand mention, however. These are: (i) the growth in hardware performance (CPU and hard disk storage in particular); (ii) the dominance of the Internet; (iii) the advent of the Open Source movement.

The last few years have witnessed very significant increases in the power of the machines that investigators are able to purchase as personal computers. Even the most modest machines

¹http://en.wikipedia.org/wiki/Software_tools_for_molecular_microscopy

available for purchase today have the option for multiple, multi-core CPUs with main memory in the multiple gigabyte range. Additionally, desktop operating systems have become more "UNIX-like" from the developer's standpoint². Improvements in compiler technology leverage the new hardware with the consequence that all of the processing packages listed in Carragher and Smith (1996, Table I) that required server-class machines at that time could now be run on a desktop or laptop and deliver significantly increased performance.

From the standpoint of today's structural biological problems the notable common theme is that all studies (with a small number of possible exceptions) absolutely require significant computing resources for success. CryoEM of single particles has become the major area of development in the past decade. The technical challenges are to preserve native specimen structure through appropriate freezing technique and to limit beam-induced specimen degradation through low-dose imaging that nevertheless permits an adequate signal-to-noise ratio to be realized in the reconstructed structure. Meeting these challenges requires the collection of very large datasets and the processing of upwards of hundreds of thousands of individual particles in some cases (e.g. Stagg et al., 2007). These requirements have driven the development of software to provide automated data collection and highly automated processing of images as the sheer volume of data effectively precludes significant operator intervention at the individual particle level.

It seems likely that computational power will continue to increase over the next decade. This continuing growth will open up the possibility of more routine analysis of datasets containing on the order of 1 million particles. The increase in computational power will also enable promising techniques that are currently still limited by CPU capacity; examples include bootstrap methods (Penczek et al., 2006) and maximum likelihood methods (Sigworth, 1998, Scheres et al., 2005). These new software approaches are likely to play a critical part in a number of challenging problems, for example computing high resolution electron density maps of conformationally variable macromolecules.

Parallelization of the processing of such large datasets becomes very important if data analysis is to be feasible. This has been recognized by Yang et al (2007) who have devised a method to achieve coarse-grain parallelization of the SPIDER software package on distributed-memory parallel computers using the message-passing interface (MPI, e.g. Pacheco, 1996). The results show significant decrease in wall-clock time for parallelizable tasks that is limited by the bandwidth of the machine interconnects. There are also significant benefits expected (e.g. Tang et al., 2007) from fine-grain parallelization that will come from the continued development of CPUs with larger numbers of cores and the compilers that can utilize their potential. Similarly, the use of alternatives like GPU's (e.g. Castaño Díez, et al., 2007) or the advent of completely novel technologies like quantum computing will open new avenues for structure analysis.

While CPU power is still sometimes a limiting factor in macromolecular microscopy, the once daunting challenge presented by the need for storing huge volumes of data has almost disappeared. With the price of disk storage now on the order of \$0.25 per Gbyte (compared to about \$1 per Mbyte in 1995), disk capacity presents almost no barrier at all. On the other hand, organizing, archiving and disseminating these data becomes ever more challenging as the number of projects grows along with the amount of data and the complexity of the processing steps.

The macromolecular microscopy field has traditionally relied on a flat file system for data organization. Information about the methods used to acquire and process the images is often distributed between lab notebooks, header information encoded into the data files, and output

²e.g. Macintosh OSX.5 (Leopard), is an Open Brand UNIX 03 Registered Product, conforming to the SUSv3 and POSIX 1003.1

text files, the format of which frequently changes over time. As a result it is easy to lose track of the native data and especially to lose the history of how the data was acquired and how the final maps were created. This makes it difficult to reprocess images in order to take advantage of new software developments or to understand which processing steps have been used when comparing similar structures.

We believe that the only feasible way of organizing the enormous quantity of data and tracking the myriad of processing steps that they undergo is with the aid of a relational database that records the location of all of the datasets and stores the metadata associated with them. The Leginon³/Appion⁴ database (Fellmann et al. 2002, Suloway et al, 2005) developed at the National Resource for Automated Molecular Microscopy (NRAMM) provides one example. This database keeps track of parameters associated with the sample, the grid preparation, the instrument and imaging conditions, the processing steps, etc. In this way once a 3D map is produced, the database can be queried to provide a complete provenance of the map. The Leginon database tracks data from ~2000 experiments associated with ~150 separate projects. This represents about 15 Terabytes of image data and about 10 Gbytes of metadata. Some of the experiments are over 5 years old but every imaging parameter associated with every image is still immediately available via a web based user interface. It is hard to imagine how a volume of data of this size could be managed using a flat file structure. Several other groups are developing databases along similar lines (for example the EMEN database⁵) and it is possible that the number and variety of databases will soon be equal to the number of available software packages. Communication between these databases will only be possible if the community can agree on a XML schema to describe essential metadata. The EMDB database⁶ (Henrick et al., 2003) that has been developed by the European Bioinformatics Institute seems to be the most rational and logical choice for a commonly accepted schema.

The second major change in computational environment that we want to briefly remark on is the advent of networks and the web. This can be readily discerned in how the method of dissemination of the software packages has changed over the last decade. In the original Hegerl article, the contact information was listed as a physical mail address. In the 1996 Special Issue, most contact information was in the form of an email address with an occasional ftp or http link available. In the 2007 Special Issue all of the contact information was consolidated onto a Wikipedia page with web links to the home pages of the software packages. The almost total interconnectedness of computers has enormously simplified the problems of moving data between computers and acquiring and installing new software packages. Web based documentation, wiki pages, bulletin boards and mailing lists for specialized user groups provide an extremely rich set of resources for software support. It is unfortunate that, as yet, this interconnectedness appears to have had a limited impact on consolidating and rationalizing the conventions used in the software packages that are available. One ray of light in this regard is a paper by Heymann et al. (2005) where they lay out a set of standardized conventions for most of the parameters that are common to the acquisition, processing and analysis steps. If all of the software packages currently in use were to adopt these conventions, or provide software to convert to and from them, it would greatly simplify the task of converting data between formats.

The final extraordinary development has been the success of the Open Source movement. An astonishingly rich variety of sophisticated software environments, applications and support libraries are now freely available and well supported by a large community of expert users and

³<http://legion.org>

⁴<http://appion.org>

⁵<http://ncmi.bcm.tmc.edu/software/emen>

⁶<http://www.ebi.ac.uk/msd-srv/docs/emdb/index.html>

developers. Some excellent examples include: python (a high-level, object-oriented, interpreted programming language), MySQL (a multithreaded, multi-user SQL database management system), PHP (a scripting language designed for producing dynamic web pages), and Apache (a web server). These packages provide the underlying architecture for the Leginon (Suloway et al., 2006) and Appion software environments developed at NRAMM and have vastly extended the capabilities of a small programming team.

Standardization?

Graphics standardization

At the time that Hegerl (1992) reviewed image-processing software, graphics output was typically obtained on raster graphics displays that were separate, costly peripheral devices requiring their own unique drivers and for which there was no standardized graphical user interface software. Graphically driven software could be written for these raster graphics devices, but it was time-consuming and difficult to do and there was little expectation that it would be portable to any other system. With the advent of X-windows (Nye, 1990), which became the basic display system for workstations in the '90s, programmers had something consistent to work with that has broad, platform independent support. At this time all of the packages listed in the Wikipedia software pages are able to use X-windows for image display, directly or indirectly, and the latest release of X-windows is supported on PCs and Macintosh desktops.

Carragher and Smith (1996) also pointed to the potential importance of OpenGL⁷ for complex graphics and indeed it has emerged as today's most widely adopted graphics standard. Ironically, the emergence of OpenGL is probably linked to the dramatic decline in use of SGI workstations and their proprietary GL window system, an outcome that was entirely unforeseen in 1996 when the growing dominance of SGI appeared to be inevitable. Future software development is likely to use OpenGL over X-windows due to the very broad support it enjoys.

Software standardization

As most software developers realize, writing core analysis code to implement an algorithm within a software package is often only a small part of the job. The significantly larger task is programming the data management and user interaction with that core, and to do it in a way that makes it easily maintainable. This reality has had a significant impact on software development. Amongst the twelve "general packages" identified in the Wikipedia pages as available for download today, six were numbered amongst the ten packages listed in (Carragher and Smith, 1996). Of these, the MRC package (Crowther et al., 1996) and Spider (Frank et al., 1996) have become *de facto* standards in many labs for a range of image processing tasks; 2D crystal, helical and icosahedral particle reconstruction in the case of the MRC package, and single particle and tomographic reconstruction in the case of Spider. A large community of users has also built up around the EMAN (Ludtke et al., 1999) software suite, which is particularly focused on a streamlined approach to single particle processing.

In several labs new developments have consisted of repackaging existing software to improve data management, user accessibility or to provide an elegant graphical user interface. Examples of these approaches can be seen in some of the more recently developed packages including "2dx" (Gipson et al., 2007), which repackages selected components of the MRC suite, and "SPIRE" (Baxter et al., 2007), which repackages SPIDER's reconstruction tools using a Python wrapper. In contrast Nickell et al. (2005) created TOM, the "Tomography Toolbox" to support cryoEM in their lab in part by reprogramming the tools provided in the EM package (Hegerl,

⁷<http://www.opengl.org>

1996) that it now largely replaces, but with the support of the large commercial MATLAB application. In this case, MATLAB (The MathWorks, Natick, MA) is used as the core analysis engine for high-performance image processing and also to control the FEI *Tecnai* electron microscope during data collection (FEI Company, Eindhoven, The Netherlands). The use of a commercial, sophisticated package like MATLAB with a large user base provides many advantages including an enormous range of algorithms and features and a community of educated developers and users. The obvious disadvantage of course is the cost, which can be prohibitive for a small lab especially if the package must be run on a cluster architecture requiring a separate license for each node.

A few of the “General” packages that are listed on the Wikipedia pages are ambitious projects that have the goal of consolidating several of the standard packages into a single environment; an example is the SPARX software system (Hohn et al., 2007). While this is a laudable goal it is a daunting task to persuade an existing user community to adopt a new system and it is not clear if any of these efforts will be able to accumulate a large enough user base to be effective, let alone become acceptable standards for the community.

Data File Format Standardization

In Carragher and Smith (1996) we stressed the value of having a generally accepted standard for image storage to facilitate data exchange and collaboration. More recently Heymann et al. (2005) have emphasized the importance of standardized conventions for image interpretation and presentation that their paper codifies under the name “3DEM Image Conventions”. So far no standard file format has been created: the nearest that we have come is the em2em⁸ package (available at no cost from Image Science, Berlin, Germany) that provides image data inter-conversion between packages. However, the general convergence on the MRC file format has achieved standardization of a sort: of all the packages listed in the Wikipedia pages, almost all can read the MRC⁹, or the closely related CCP4¹⁰ formatted files.

At this point, therefore, the MRC/CCP4 format is the *de facto* standard. Its benefits (as described on the documentation web pages⁹) are that the relationship of the map to the crystal cell, and other information useful in crystallographic calculations (e.g. crystal symmetry) are all stored in the header, the file organization permits the writing and reading of the file section by section or line by line, and the format is suitable for both crystallographic work and for image processing so that Fourier and plotting programs can be used for both purposes. Its support for machine stamping, which identifies the byte-order and format of “reals” written in a file so that it can be read on machines with different architectures, was a relatively recent addition that has permitted the MRC format file to become a generally accepted cross-platform file format.

With the huge increase in data volumes required to solve structures in cryoEM there are potential benefits in using a file structure that supports large-scale datasets. HDF5¹¹, already supported by EMAN and EMAN2 (Ludtke et al., 1999), is a data format developed over the past 20 years to support very large datasets, very fast access requirements, or very complex datasets. Increased awareness of HDF5 could well provide opportunities for improved data management in cryoEM projects in the future.

⁸<http://www.imagescience.de/em2em/index.htm>

⁹<http://www2.mrc-lmb.cam.ac.uk/image2000.html>

¹⁰<http://www.ccp4.ac.uk/html/maplib.html#description>

¹¹<http://hdf.ncsa.uiuc.edu/>

Software consolidation

The wish expressed by Hegerl (1992) and echoed in (Carragher and Smith, 1996) was that investigators would collaborate and, in the interests of greater software maintainability and ease of development, would settle on a single model for overall software design and a single, shared library of analysis tools. The last few years have shown that this goal is unlikely to be realized. Given that most investigators understand the benefits of such a strategy in general it is interesting to ask why people build rather than borrow or buy for their own specific requirements.

Probably the biggest disincentive to using code from other labs is a lack of control over the details of the analysis. For this reason, while basic routines such as a multi-radix FFT (e.g. FFTW¹²), general Fourier filtering routines or CTF corrections are very likely to be common to many software developments and obtained from a standard library, the details of particle alignment and data merging are likely to be part of the specific algorithm development that a lab uses as part of its “competitive edge” in tackling complex problems and winning funding. Despite the efforts that developers have made to make standard packages “easy to modify and adapt”, breaking open a black-box library routine to re-build functionality can be daunting: it often seems simpler to start from scratch where all aspects of the software development task for a specific problem can be addressed directly and transparently.

Another disincentive is the culture of a laboratory. If a lab has invested years working with a particular software package and its development and has become productive and comfortable with its use, the transition to a package with a completely different mode of operation will be very challenging. In such cases re-writing code taken from another source to integrate the new functionality into the lab’s workhorse software could be the fastest and least disruptive way to get access to the new technology.

Finally, computing paradigms change. Twelve years ago, the idea that incurring the additional overhead of using an interpreted language such as PHP, Python or Java as the “glue” to build an image-processing package would have seemed quite odd. With the increased power of the hardware and the scale of the computational tasks, this overhead is now only a minor component of the overall cost of running an analysis (e.g. Ludtke et al., 1999). The benefits of significantly more flexible programming using PHP, for example, include an accelerated development schedule for new software and a shorter time to get novice investigators up-to-speed using the new tools. New developments tend, therefore, to favor use of new tools rather than to continue to build using an “obsolete” paradigm that is no longer in favor and is thought to limit development options.

Possibly the only real incentive likely to encourage the consolidation of software is funding. We point out two efforts in this direction. Over the past 5 years the EEC has funded the “3D-EM Network of Excellence¹³”. The program is a consortium of most of the major European labs with expertise in macromolecular microscopy. One of the original three major goals of the network was the creation of a standardized software platform. A more modestly funded project in the United States had as one of its goals the consolidation of some of the most popular packages used for single particle analysis (e.g. EMAN, Spider and Phenix¹⁴). This has resulted in the SPARX¹⁵ image-processing environment. Whether the developments of either of these large efforts towards a unified software platform succeed will probably depend on ongoing

¹²<http://www.fftw.org/>

¹³<http://www.3dem-noe.org/index.html>

¹⁴<http://www.phenix.bnl.gov/WWW/software/>

¹⁵<http://macro-em.org/sparxwiki/SparxWiki>

funding and the cooperation between all of the interested parties both across the Atlantic and around the rest of the world.

In speculating how software development for macromolecular microscopy might change over the next 10 years we expect that it will follow the trend of all other technology developments. That is, we are likely to over-estimate what can be achieved in the short term but under estimate the changes that will occur over the long term.

Acknowledgments

The authors wish to thank the Editor-in-Chief for the invitation to prepare this paper, and the reviewers for their helpful input. They also gratefully acknowledge salary support from the NCRR from grant M01 RR00096 (RS) and RR17573 (BC)

References

- Aebi U, Carragher B, Smith PR. EDITORIAL. *J. Struct. Biol* 1996;116:1. [PubMed: 8812974]
- Baxter WT, Leith A, Frank J. SPIRE: The SPIDER Reconstruction Engine. *J. Struct. Biol* 2007;157:56–63. [PubMed: 17055743]
- Carragher B, Smith PR. Advances in Computational Image Processing for Microscopy. *J. Struct. Biol* 1996;116:2–8. [PubMed: 8742716]
- Castaño Díez D, Mueller H, Frangakis AS. Implementation and performance evaluation of reconstruction algorithms on graphics processors. *J. Struct. Biol* 2007;157:288–295. [PubMed: 17029985]
- Crowther RA, Henderson R, Smith JM. MRC Image Processing Programs. *J. Struct. Biol* 1996;116:9–16. [PubMed: 8742717]
- Fellmann D, Pulokas J, Milligan RA, Carragher B, Potter CS. A relational database for cryoEM: experience at one year and 50 000 images. *J. Struct. Biol* 2002;137:273–282. [PubMed: 12096895]
- Frank J, Radermacher M, Penczek P, Zhu J, Li Y, Ladjadj M, Leith A. SPIDER and WEB: Processing and Visualization of Images in 3D Electron Microscopy and Related Fields. *J. Struct. Biol* 1996;116:190–199. [PubMed: 8742743]
- Gipson B, Zeng X, Zhang ZY, Stahlberg H. 2dx--User-friendly image processing for 2D crystals. *J. Struct. Biol* 2007;157:64–72. [PubMed: 17055742]
- Hegerl R. A brief survey of software packages for image processing in biological electron microscopy. *Ultramicroscopy* 1992;46:417–423.
- Hegerl R. The EM Program Package: A Platform for Image Processing in Biological Electron Microscopy. *J. Struct. Biol* 1996;116:30–34. [PubMed: 8812976]
- Henrick K, Newman R, Tagari M, Chagoyen M. EMDep: a web-based system for the deposition and validation of high-resolution electron microscopy macromolecular structural information. *J. Struct. Biol* 2003;144:228–237. [PubMed: 14643225]
- Heymann JB, Chagoyen M, Belnap DM. Common conventions for interchange and archiving of three-dimensional electron microscopy information in structural biology. *J. Struct. Biol* 2005;151:196–207. [PubMed: 16043364]corrigendum (2006): 153, 312
- Hohn M, T, Tang G, Goodyear G, Baldwin PR, Huang Z, Penczek PA, Yang Ch, Baldwin PR, Adams P, Ludtke SJ. SPARX, a new environment for cryo-EM image processing. *J. Struct. Biol* 2007;157:47–55. [PubMed: 16931051]
- Ludtke SJ, Baldwin PR, Chiu W. EMAN: Semiautomated Software for High-Resolution Single-Particle Reconstructions. *J. Struct. Biol* 1999;128:82–97. [PubMed: 10600563]
- Macias JR, Jimenez-Lozano N, Carazo JM. Integrating electron microscopy information into existing Distributed Annotation Systems. *J. Struct. Biol* 2007;158:205–213. [PubMed: 17400476]
- Nickell S, Forster F, Linaroudis A, Net WD, Beck F, Hegerl R, Baumeister W, Plitzko JM. TOM software toolbox: acquisition and analysis for electron tomography. *J. Struct. Biol* 2005;149:227–234. [PubMed: 15721576]
- Nye, A., editor. X-Window System. Sebastapol, CA: O'Reilly; 1990.
- Pacheco, PS. Parallel programming with MPI. Morgan Kaufmann Publishers Inc; 1996.

- Penczek PA, Yang C, Frank J, Spahn CMT. Estimation of variance in single-particle reconstruction using the bootstrap technique. *J. Struct. Biol* 2006;154:168–183. [PubMed: 16510296]
- Saxton WO, Pitt TJ, Horner M. Digital image processing: The semper system. *Ultramicroscopy* 1979;4:343–353.
- Scheres SHW, Valle M, Nunez R, Sorzano COS, Marabini R, Herman GT, Carazo J-M. Maximum-likelihood Multi-reference Refinement for Electron Microscopy Images. *J. Mol. Biol* 2005;348:139–149. [PubMed: 15808859]
- Sigworth FJ. A Maximum-Likelihood Approach to Single-Particle Image Refinement. *J. Struct. Biol* 1998;122:328–339. [PubMed: 9774537]
- Smith PR, Gottesman SM. The Micrograph Data Processing Program. *J. Struct. Biol* 1996;116:35–40. [PubMed: 8742720]
- Stagg SM, Lander GC, Pulokas J, Fellmann D, Cheng A, Quispe JD, Mallick SP, Avila RM, Carragher B, Potter CS. Automated cryoEM data acquisition and analysis of 284 742 particles of GroEL. *J. Struct. Biol* 2006;155:470–481. [PubMed: 16762565]
- Suloway C, Pulokas J, Fellmann D, Cheng A, Guerra F, Quispe J, Stagg S, Potter CS, Carragher B. Automated molecular microscopy: The new Legimon system. *J. Struct. Biol* 2005;151:41–60. [PubMed: 15890530]
- Tang G, Peng L, Baldwin PR, Mann DS, Jiang W, Rees I, Ludtke SJ. EMAN2: An extensible image processing suite for electron microscopy. *J. Struct. Biol* 2007;157:38–46. [PubMed: 16859925]
- Trus BL, Kocsis E, Conway JF, Steven AC. Digital Image Processing of Electron Micrographs: The PIC System-III. *J. Struct. Biol* 1996;116:61–67. [PubMed: 8742724]
- van Heel M, Harauz G, Orlova EV, Schmidt R, Schatz M. A New Generation of the IMAGIC Image Processing System. *J. Struct. Biol* 1996;116:17–24. [PubMed: 8742718]
- Yang C, Penczek PA, Leith A, Asturias FJ, Ng EG, Glaeser RM, Frank J. The parallelization of SPIDER on distributed-memory computers using MPI. *J. Struct. Biol* 2007;157:240–249. [PubMed: 16859923]