electronic reprint

Journal of
Applied Crystallography
ISSN 0021-8898
Editor: **Gernot Kostorz**

Shake-and-Bake on the grid

Russ Miller, Naimesh Shah, Mark L. Green, William Furey and Charles M. Weeks

Journal of
Applied Crystallography
ISSN 0021-8898

Received 27 February 2007
Accepted 15 July 2007

Shake-and-Bake on the grid

Russ Miller,[a,b]* Naimesh Shah,[a] Mark L. Green,[b] William Furey[c] and Charles M. Weeks[a]

[a]Hauptman-Woodward Medical Research Institute, 700 Ellicott Street, Buffalo, NY 14203, USA, [b]Department of Computer Science and Engineering, SUNY-Buffalo, Buffalo, NY 14260, USA, and [c]VA Medical Center and Department of Pharmacology, University of Pittsburgh, Pittsburgh, PA 15261, USA. Correspondence e-mail: miller@buffalo.edu

Computational and data grids represent an emerging technology that allows geographically and organizationally distributed resources (*e.g.* computing and storage resources) to be linked and accessed in a fashion that is transparent to the user, presenting an extension of the desktop for users whose computational, data and visualization needs extend beyond their local systems. The New York State Grid is an integrated computational and data grid that provides web-based access for users from around the world to computational, application and data storage resources. This grid is used in a ubiquitous fashion, where the users have virtual access to their data sets and applications, but do not need to be made aware of the details of the data storage or computational devices that are specifically employed. Two of the applications that users worldwide have access to on a variety of grids, including the New York State Grid, are the *SnB* and *BnP* programs, which implement the *Shake-and-Bake* method of molecular structure (*SnB*) and substructure (*BnP*) determination, respectively. In particular, through our grid portal (*i.e.* logging on to a web site), *SnB* has been run simultaneously on all computational resources on the New York State Grid as well as on more than 1100 of the over 3000 processors available through the Open Science Grid.

1. Introduction

The grid is a rapidly emerging and expanding technology that allows geographically distributed and independently operated resources to be linked together in a transparent fashion (www.gridcomputing.com, www.globus.org; Berman *et al.*, 2003; Foster & Kesselmann, 1999). These resources include CPU cycles, data storage systems, sensors, visualization devices and a wide variety of internet-ready instruments. The concept and terminology of the grid is borrowed from the electrical grid, where numerous utility companies have the ability to share and move resources (electricity) in a fashion that is transparent to the consumer. With rare exception, the view taken by a consumer is that they are able to plug a piece of equipment into a power outlet in order to obtain electricity and do not need to know, and in fact do not want to know, the details pertaining to the manner in which electricity makes its way to the outlet to provide power to some device. Similarly, the power of both computational grids (*i.e.* seamlessly connecting computer systems and their local storage) and data grids (*i.e.* seamlessly connecting large storage systems) lies not only on the aggregate computing power and data storage but also on its ease of use.

Numerous government-sponsored reports state that grid computing is a key to 21st century discovery by providing seamless access to computational systems that are required for revolutionary advances in contemporary science and engineering. In fact, National Science Foundation (NSF) Director Arden Bement states that ''leadership in cyberinfrastructure may determine America's continued ability to innovate, and thus our ability to compete successfully in the global arena''.

Grids are now a viable solution to certain computationally and data-intensive computing problems for reasons that include the following.

(i) Users can access a grid from virtually any web-based device anywhere in the world.

(ii) The internet is mature and able to serve as the fundamental infrastructure for network-based computing.

(iii) Storage capacity has now reached commodity levels, where one can purchase a terabyte of disk for roughly the same price as a high-end PC.

(iv) Many instruments are internet-aware.

(v) Clusters, supercomputers, storage and visualization devices are becoming more mainstream in terms of their ability to host scientific applications.

For these and other reasons, grids are starting to move out of the research laboratory and into early adopter production systems. Numerous grid projects have been initiated [*e.g.* GriPhyN (www.griphyn.org), PPDG (www.ppdg.net), EGEE (www.eu-egee.org), EU DataGrid (eu-datagrid.web.cern.ch/eu-datagrid), NASA's Information Power Grid (IPG) (www.gloriad.org/gloriad/projects/project000053.html), Tera-

Grid (www.teragrid.org), Open Science Grid (www.opensciencegrid.org) and iVDGL (www.ivdgl.org), to name a few]. However, the construction of a real general-purpose grid is in its infancy since a true grid requires coordinated resource sharing and problem solving in a dynamic multi-institutional scenario using standard open general-purpose protocols and interfaces that deliver a high quality of service. The immediate focus of grid deployment continues to be on the difficult issue of developing high-quality 'middleware' (www.nsf-middleware.org).

As grid computing initiatives move forward, issues of interoperability, security, performance, management and privacy need to be carefully considered. In fact, security is concerned with various issues relating to authentication in order to ensure application and data integrity. Grid initiatives are also generating best-practice scheduling and resource management documents, protocols and API specifications to enable interoperability. Several layers of security, data encryption and certificate authorities already exist in grid-enabling toolkits such as Globus Toolkit (www.globus.org/toolkit).

2. Background

2.1. New York State Grid

The New York State Grid (NYS Grid) was designed and deployed by the SUNY-Buffalo Cyberinfrastructure Laboratory (www.cse.buffalo.edu/faculty/miller/CI). It includes resources from institutions throughout New York State and is available in a simple and seamless fashion to users worldwide. NYS Grid contains a heterogeneous set of resources and utilizes general-purpose IP networks (Green & Miller, 2003, 2004*a*,*b*,*c*). A major feature of this grid is that it integrates a computational grid (typically large computer clusters that have the ability to cooperate in serving the user) with a data grid (typically large storage devices that are similarly available to the user) so that the user may deploy computationally intensive applications that read or write large data files in a very simple fashion. In particular, NYS Grid was designed so that users do not need to know where the files are physically stored or where the application is physically run, while providing users with easy access to their files in terms of uploading, downloading, editing, viewing and so on. Of course, a user who wishes to manage more closely where the data are stored and where the applications are run also has the option to retain such control.

The core infrastructure for NYS Grid includes the installation of standard grid 'middleware', the use of an active web portal for deploying applications, dynamic resource allocation so that clusters and networks of workstations can be scheduled to provide resources on demand, a scalable and dynamic scheduling system, and a dynamic firewall, to name a few. NYS Grid transparently integrates a computational grid with a data grid, which is quite useful for the disciplinary scientist interested in scientific results, while ignoring the underlying grid intricacies. That is, from the users' perspective, they have convenient access to their data and to the computational resources required to process their data, but they do not need to know the location of the data or the computational resources.

Several key packages were used in the implementation of NYS Grid and other packages have been identified in order to allow for the anticipated expansion of the system. The Globus Toolkit (www.globus.org/toolkit) provides APIs and tools using Java SDK to simplify the development of OGSI-compliant services and clients. It supplies database services and Monitoring and Discovery System index services implemented in Java (www.globus.org/toolkit/mds), GRAM service implemented in C with a Java wrapper (www.globus.org/toolkit/docs/2.4/gram), GridFTP services implemented in C (www.globus.org/grid_software/data/gridftp.php) and a full set of Globus Toolkit components. The recently proposed Web Service-Resource Framework provides the concepts and interfaces developed by the OGSI specification exploiting the web services architecture (www.globus.org/wsrf).

NYS Grid has been used to support a campus grid involving a variety of independently run organizations at SUNY-Buffalo, a Western New York grid that provides a seamless and transparent mode of operation for grid users in the western New York region (*e.g.* the Hauptman-Woodward Medical Research Institute, Niagara University, and SUNY-Geneseo, to name a few), and the 17-member New York State Cyberinfrastructure Initiative (www.nysgrid.org).

NYS Grid supports a variety of applications and serves as a gateway to the Open Science Grid, TeraGrid, MCEER (mceer.buffalo.edu) and NEES (it.nees.org and nees.buffalo.edu), to name a few. In addition to the crystallographic *Shake-and-Bake* application, NYS Grid supports a wide variety of diverse scientific applications through its portal, as well as a worldwide set of applications that are submitted to NYS Grid in a more traditional command-line fashion.

2.2. Shake-and-Bake

Shake-and-Bake is a powerful algorithmic formulation of direct methods that, given accurate diffraction data to 1.2 Å or better resolution, has made possible the *ab initio* phasing of complete crystal structures containing as many as ∼2000 independent non-H atoms (Frazão *et al.*, 1999). It has also been used to determine the anomalously scattering substructures of selenomethionyl-substituted proteins containing as many as 160 selenium sites using 3–4 Å data (Von Delft *et al.*, 2003). *Shake-and-Bake* belongs to the class of phasing methods known as 'multisolution' procedures (Germain & Woolfson, 1968) in which multiple sets of trial phases are generated in the hope that one or more of the resultant combinations will lead to a solution. Solutions, if they occur, are identified on the basis of the value of a suitable figure of merit, such as the minimal function (DeTitta *et al.*, 1994) or the crystallographic *R* value. Since each of the sets of trial phases can be processed independently, the *Shake-and-Bake* algorithm can be easily adapted to a coarse-grained parallel processing approach and implemented on a computational grid.

The distinctive feature of *Shake-and-Bake* is the repeated and unconditional cyclical alternation of reciprocal-space phase refinement with a complementary real-space process that seeks to improve phases by imposing constraints through a physically meaningful interpretation of the electron density (Miller *et al.*, 1993; Weeks *et al.*, 1994). First, a random-number generator is used to assign initial coordinates to the atoms comprising the trial structures, and structure-factor calculations are performed to generate the corresponding sets of starting trial phases. Then, phases are refined either by the tangent formula (Karle & Hauptman, 1956), or by constrained minimization of the so-called minimal function (DeTitta *et al.*, 1994) using the parameter-shift algorithm (Bhuiya & Stanley, 1963). Following Fourier transformation to real space and computation of an electron-density map, peak picking is used to impose the atomicity constraint. Peaks are located on the map, an appropriate number of the largest of these maxima are assumed to be atoms, and the cycle is completed by using inverse Fourier transformation (in the form of a structure-factor calculation) to generate phases for another round of refinement. The entire process is repeated for a predetermined (by the user) number of cycles. The success rate of this process (*i.e.* the percentage of trial structures that converge to solution) depends on data quality and the size of the structure.

The *Shake-and-Bake* procedure has been implemented in a computer program, *SnB*, in a manner convenient for both protein substructures and complete structures (Miller *et al.*, 1994; Weeks & Miller, 1999; Rappleye *et al.*, 2002). [The *Shake-and-Bake* algorithm has also been implemented independently in the program *SHELXD* (Schneider & Sheldrick, 2002).] The *SnB* graphical user interface, written in Java, controls not only the main phasing program but also the *DREAR* program suite (Blessing & Smith, 1999) that computes the normalized structure-factor magnitudes ($|E|$) required for direct-methods calculations. In addition, the two-step process of substructure determination and protein phasing has been combined in the program *BnP* (Weeks *et al.*, 2002), which provides a common interface for *SnB* and components of the *PHASES* suite (Furey & Swami-

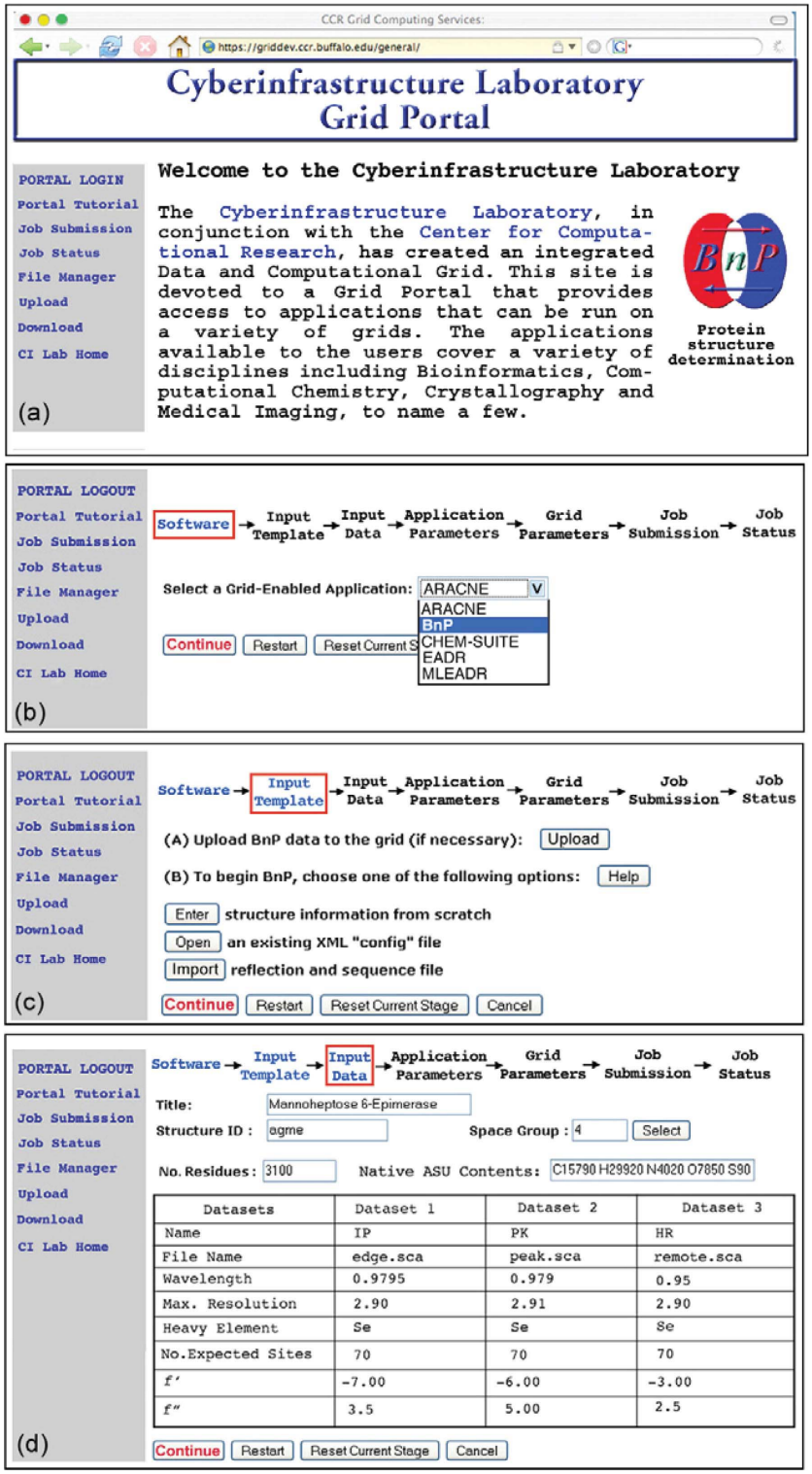

Figure 1
Job submission, monitoring and examination of results using the grid portal. (*a*) Login to the SUNY-Buffalo Cyberinfrastructure Laboratory grid portal. (*b*) Select *BnP* as the software application to be used. (*c*) Select execution options and upload files from the home computer. (*d*) Input additional information about the structure. (*e*) Assign values to program parameters that will be varied in different jobs. Some additional parameters (constant for all jobs submitted simultaneously) are accessible from a pop-up window. (*f*) Supply information controlling job execution on the grid. (*g*) Review all parameters and start the jobs. (*h*) Check whether the jobs are still running. (*i*) 'Drilldown' to check the execution status of individual components of a multiple job set. (*j*) 'Drilldown' to inspect the results of an individual job. (*k*) Figure-of-merit histogram for the selected job. (*l*) Download the output files to the home computer.

nathan, 1997). Thus, *BnP* provides an automated pathway from processed intensities to an unambiguous protein electron-density map. This pathway includes *SnB* substructure determination, heavy-atom site validation, enantiomorph determination, substructure and protein phase refinement, and solvent flattening.

The repetitive shuttling of trial structures between real and reciprocal space gives the *Shake-and-Bake* algorithm its power, but the need to perform two Fourier transformations in each cycle yields a computationally intensive procedure. In fact, the running time for *SnB* or *BnP* varies widely, from just a few seconds to many hours for large structures or structures with diffraction data of marginal quality that typically require a large number of trial structures to be examined before a solution is found. In such cases, the ability to increase throughput by processing many trial structures simultaneously on a cluster or a computational grid is invaluable.

3. Results

NYS Grid may be accessed *via* a browser through a web portal, as shown in Fig. 1(*a*), or through a standard command-line submission. Versions of the *SnB* and *BnP* programs capable of parallel processing have been equipped with web-compatible PHP interfaces and incorporated into the web portal for ease of use using grid-enabling application templates (Green & Miller, 2004*c*). The traditional stand-alone Java versions and the new grid-enabled PHP versions are compatible in the sense that files created in both versions are interchangeable and can be uploaded to and downloaded from the grid as needed. As mentioned previously, *SnB* has been run on NYS Grid and Open Science Grid through the web portal, as well as through command-line submission, to compute hundreds of thousands of trial structures using an aggregate of nearly 100000 CPUs.

BnP combines the *Shake-and-Bake* substructure determination step, which can easily take advantage of a parallel computing environment, with protein phasing calculations that are applied to a successful *Shake-and-Bake* substructure trial. The approach that has been taken is to spread the processing of trial substructures among a large number of computational nodes. In most cases it is possible for *BnP* to determine when a substructure solution has occurred and to move on to the next step. Jobs that complete the first step without identifying a solution can (optionally) be terminated. Since jobs on different machines are independent once they have been spawned, multiple solutions are possible,

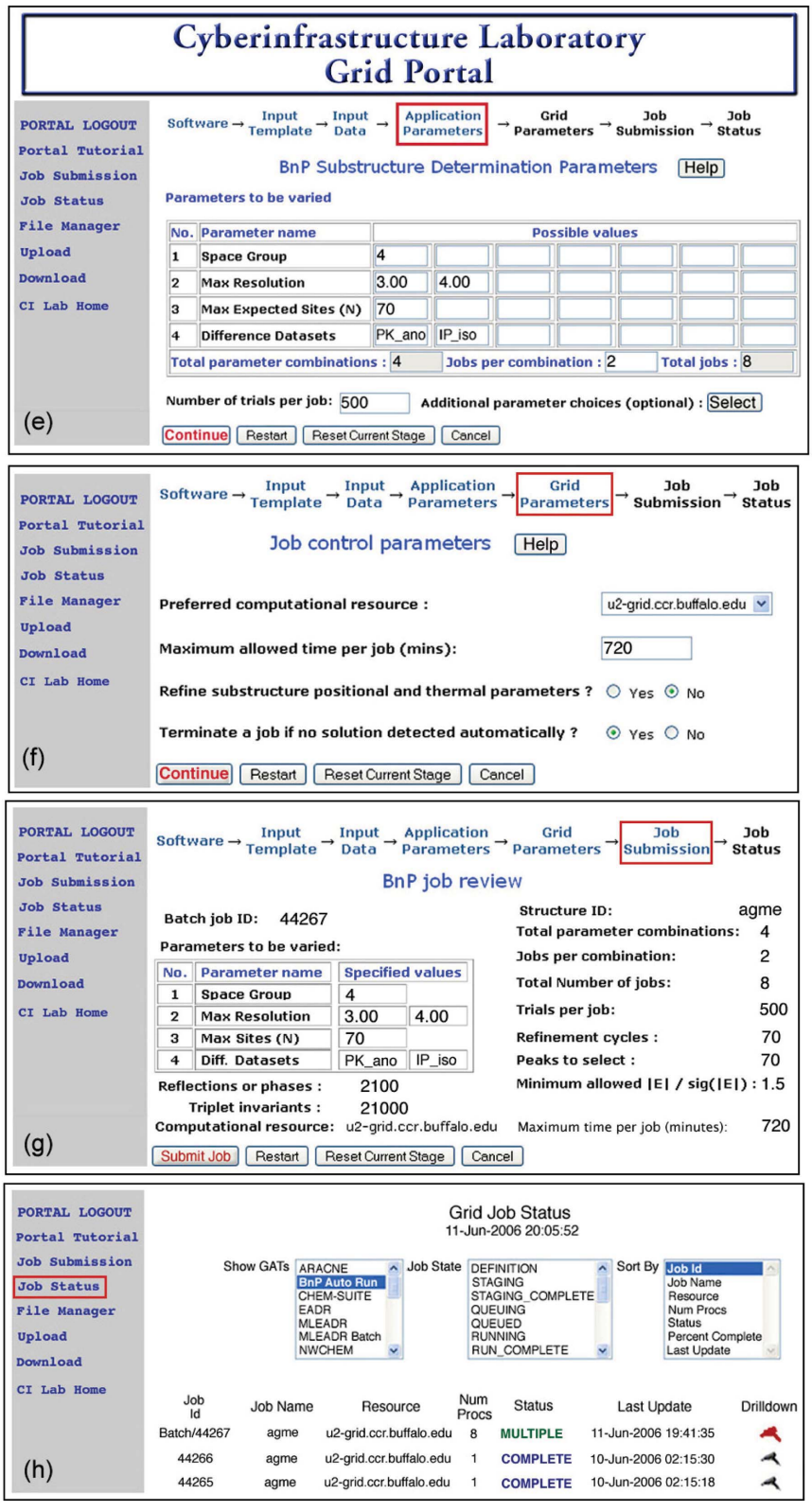

Figure 1 (continued)

which is an advantage if any jobs produce false solutions or solutions of marginal quality. It was decided that this execution scenario maximized the probability that at least one solution would be found for a difficult structure in a modest amount of time.

Fig. 1 illustrates the use of the grid-enabled version of *BnP* to phase the protein mannoheptose 6-epimerase (Deacon *et al.*, 2000) after first solving the 70-Se substructure. Fig. 1(*a*) shows the grid portal as it would initially appear to a user. After login, the user will be presented with screens representing the various stages of the workflow, which are defined at the top of the screen. In each case the current stage (Figs. 1*b*–1*g*) is indicated by the red rectangle. To move to the next stage, the user must click the 'Continue' button at the bottom of the screen. Several 'Help' buttons are available along the way.

First, the *BnP* software is selected (Fig. 1*b*). At the next stage (Fig. 1*c*), the user is reminded to upload input data files to the grid (not illustrated) and then instructed to choose how additional required information is to be supplied. If 'Enter' (or 'Continue') is selected, a blank version of screen in Fig. 1(*d*) will appear, and the information requested about the structure (*e.g.* space group) and its data sets must be entered manually. If 'Import' is chosen, most or all of the information will be extracted from the headers of the input data files, and 'Open' will restore information saved from a previous job.

The next step is to supply the values for the parameters required to execute the *BnP* application (Fig. 1*e*). Experience has shown that it is sometimes important to vary the values of certain key parameters when performing a *Shake-and-Bake* calculation in order to ensure that a solution will be found. These parameters include the space group (specified by giving its number in *International Tables for Crystallography*), the maximum resolution of the diffraction data to be used for the substructure determination, the maximum number of expected heavy-atom or anomalously scattering sites, and the type of normalized difference data to be used for *Shake-and-Bake* (*e.g.* peak wavelength anomalous differences or maximum dispersive differences). The interface computes the number

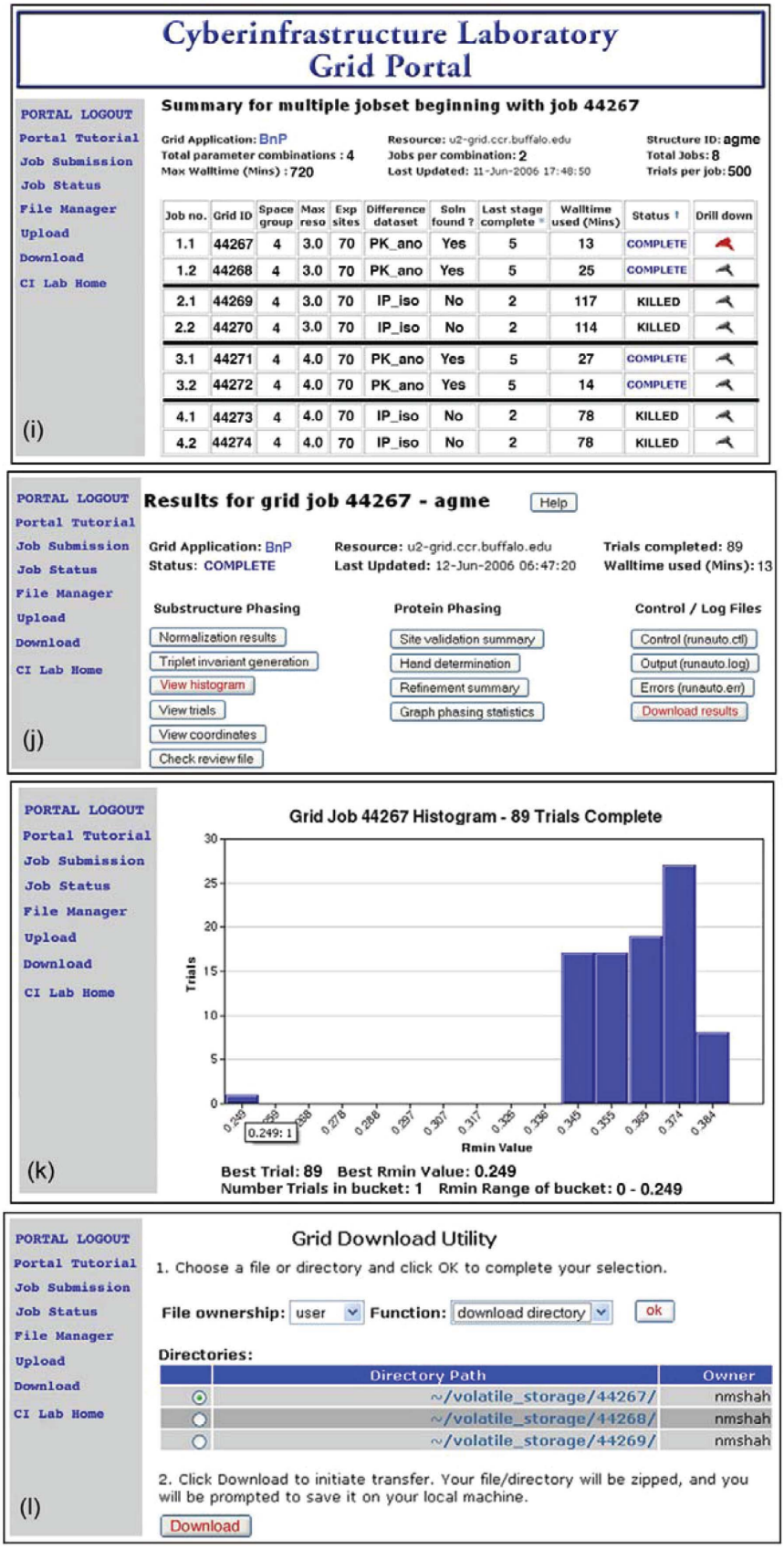

Figure 1 (continued)

of different parameter combinations that have been specified, and the user must then indicate the number of different substructure trials that are to be processed on the grid for each set of parameter values. The final choice that the user has to make at this stage is to decide whether the default values of other *BnP* parameters (*e.g.* the numbers of phases, peaks and refinement cycles), that are to be fixed for all jobs, should be changed (option not illustrated).

Parameters such as the computational resource (*e.g.* a particular computer cluster) on which the jobs are to be run and the maximum time allowed per job are specified at the 'Grid Parameters' stage (Fig. 1*f*). The user can also specify whether jobs are to be terminated after the substructure determination (*Shake-and-Bake*) step if no solution has been detected automatically based on the figure-of-merit calculation. Otherwise, the job will continue using the trial substructure with the best figure of merit even though the program does not regard it as a solution. Making the decision to terminate frees resources for other users, but there is a small risk that an actual solution might be missed.

The next screen (Fig. 1*g*) permits the user to review all parameter choices before the job(s) are submitted. The progress of the jobs can then be monitored from the 'Job Status' screen (Fig. 1*h*). A 'drilldown' button (shown in red) provides access to more information about an individual job or multiple job set, and Fig. 1(*i*) shows details about the status of the multiple job (number 44267) setup (Fig. 1*e*) and selected for viewing (Fig. 1*h*). The values of the variable parameters for each job are given in the table, which indicates that the jobs were grouped in pairs (differing by the random-number seed). In the case illustrated, all jobs that used the peak anomalous difference data gave recognizable solutions and proceeded to perform all *BnP* tasks up to and including protein phasing and solvent flattening (note indications that stage 5 was completed). The other jobs did not find solutions and were killed (terminated) after the *Shake-and-Bake* stage (#2) had processed all 500 trial structures assigned to them. If any jobs had failed because of *BnP* errors or for other reasons, this would have been indicated in the status column.

'Drilling down' one level further (indicated here by the red symbol on the line for job 1.1) provides access to the results for a single job (Fig. 1*j*). Those results include, for example, a *Shake-and-Bake* histogram (Fig. 1*k*) of minimal function values for the 89 trials that were processed in job 1.1 before the program automatically detected that a probable solution had been found. Finally, clicking the 'Download results' button (on Fig. 1*j*) causes the screen shown in Fig. 1(*l*) to be displayed with options for downloading the files resulting from the *BnP* calculations to the home computer.

4. Conclusions

The *SnB* and *BnP* applications have been adapted to a grid environment. They are available through a web-based portal and currently run on both the NYS Grid and the Open Science Grid, providing the user with access to many thousands of processors. In order to use either of these programs on a grid, the user simply points their browser to https://grid.ccr.buffalo.edu/new_user/ and clicks on 'Request a Portal Account.' After following a simple set of instructions, the user will be provided with a grid account and access to the requested applications, usually within a week (a grid account requires security that includes user verification). Once the user has been issued an account, they simply point their browser to https://grid.ccr.buffalo.edu/ and initiate either an *SnB* or *BnP* application. Note that there is no requirement that the user be at an institution that is a member of a particular grid or that a user install any special software on their system. The web-based grid portal provides convenient access to these programs and users do not even need to know details of where their files are maintained or where the computations are performed, although details are always available for those who wish to be so informed. The grid-enabled versions allow for a much shorter time to solution for large and difficult structures. Finally, feel free to contact the authors if you would like to be able to run either *SnB* or *BnP* on another grid. Typically, this can be readily arranged.

The development of NYS Grid and the grid-enabled versions of *SnB* and *BnP* were supported by NSF grant ACI-0204918, NSF CRI grant CNS-0454114 and NIH grant EB002057. The authors would like to thank members of Dr Miller's Cyberinfrastructure Laboratory (including Steve Gallo, Jason Rappleye, Cathy Ruby, Jon Bednasz, Tony Kew, Sam Guercio, Adam Koniak, Martins Innus, Dori Macchioni, Amin Ghadersohi and Cynthia Cornelius) for their contributions to the efforts described in this paper. Much of this research was performed while Dr Miller was Director of the Center for Computational Research.

References

Berman, F., Hey, A. J. G. & Fox, G. C. (2003). Editors. *Grid Computing: Making the Global Infrastructure a Reality*. New York: John Wiley.
Bhuiya, A. K. & Stanley, E. (1963). *Acta Cryst.* **16**, 981–984.
Blessing, R. H. & Smith, G. D. (1999). *J. Appl. Cryst.* **32**, 664–670.
Deacon, A. M., Ni, Y. S., Coleman, W. G. Jr & Ealick, S. E. (2000). *Structure*, **8**, 453–462.
DeTitta, G. T., Weeks, C. M., Thuman, P., Miller, R. & Hauptman, H. A. (1994). *Acta Cryst.* A**50**, 203–210.
Foster, I. & Kesselmann, C. (1999). Editors. *The Grid: Blueprint for a New Computing Intrastructure*. San Francisco: Morgan Kaufmann.
Frazão, C., Sieker, L., Sheldrick, G. M., Lamzin, V., LeGall, J. & Carrondo, M. A. (1999). *J. Biol. Inorg. Chem.* **4**, 162–165.
Furey, W. & Swaminathan, S. (1997). *Methods Enzymol.* **277**, 590–620.
Germain, G. & Woolfson, M. M. (1968). *Acta Cryst.* B**24**, 91–96.
Green, M. L. & Miller, R. (2003). *Ann. Eur. Acad. Sci.* pp. 191–218.
Green, M. L. & Miller, R. (2004*a*). *Parallel Comput.* **30**, 1001–1017.
Green, M. L. & Miller, R. (2004*b*). *Parallel Comput.* **30**, 1057–1071.
Green, M. L. & Miller, R. (2004*c*). *Parallel Process. Lett.* **14**, 241–253.
Karle, J. & Hauptman, H. (1956). *Acta Cryst.* **9**, 635–651.
Miller, R., DeTitta, G. T., Jones, R., Langs, D. A., Weeks, C. M. & Hauptman, H. A. (1993). *Science*, **259**, 1430–1433.
Miller, R., Gallo, S. M., Khalak, H. G. & Weeks, C. M. (1994). *J. Appl. Cryst.* **27**, 613–621.
Rappleye, J., Innus, M., Weeks, C. M. & Miller, R. (2002). *J. Appl. Cryst.* **35**, 374–376.

Schneider, T. R. & Sheldrick, G. M. (2002). *Acta Cryst.* D**58**, 1772–1779.
Von Delft, F., Inoue, T., Saldanha, S. A., Ottenhof, H. H., Schmitzberger, F., Birch, L. M., Dhanaraj, V., Witty, M., Smith, A. G., Blundell, T. L. & Abell, C. (2003). *Structure*, **11**, 985–996.
Weeks, C. M., Blessing, R. H., Miller, R., Mungee, R., Potter, S. A., Rappleye, J., Smith, G. D., Xu, H. & Furey, W. (2002). *Z. Kristallogr.* **217**, 686–693.
Weeks, C. M., DeTitta, G. T., Hauptman, H. A., Thuman, P. & Miller, R. (1994). *Acta Cryst.* A**50**, 210–220.
Weeks, C. M. & Miller, R. (1999). *J. Appl. Cryst.* **32**, 120–124.