research papers

Automated crystallographic system for high-throughput protein structure determination

Joseph S. Brunzelle,[a] Padram Shafaee,[b] Xiaojing Yang,[a,c] Steve Weigand,[d] Zhong Ren[c] and Wayne F. Anderson[a]*

[a]Northwestern University Feinburg School of Medicine, USA, [b]Hostway Corporation, USA, [c]Renz Research Inc, USA, and [d]Advanced Photon Source DuPont–Northwestern–Dow Collaborative Access Team, USA

Correspondence e-mail:
wf-anderson@northwestern.edu

High-throughput structural genomic efforts require software that is highly automated, distributive and requires minimal user intervention to determine protein structures. Preliminary experiments were set up to test whether automated scripts could utilize a minimum set of input parameters and produce a set of initial protein coordinates. From this starting point, a highly distributive system was developed that could determine macromolecular structures at a high throughput rate, warehouse and harvest the associated data. The system uses a web interface to obtain input data and display results. It utilizes a relational database to store the initial data needed to start the structure-determination process as well as generated data. A distributive program interface administers the crystallographic programs which determine protein structures. Using a test set of 19 protein targets, 79% were determined automatically.

1. Introduction

The rapid advance of genomic sequencing projects has already produced remarkable results and the quantity of sequence information is rapidly expanding. The sequencing effort is now being followed by a concerted structural genomics effort hosted by at least 13 different consortia worldwide. There are at least 309 structures that have already been deposited into the PDB (Berman *et al.*, 2000) from 1999 to present that are classified as 'Structural Genomics'. The current bottleneck in the process of structural genomics appears to be in the production of X-ray quality crystals, but the high-throughput determination of structures is likely to become a significant bottleneck as earlier stages attain high throughput.

Presently, there are several structure-determination software packages with built-in automation, *i.e. SOLVE/RESOLVE* (Terwilliger, 1999; Terwilliger & Berendzen, 1999), *AUTOSHARP/SHARP* (de la Fortelle & Bricogne, 1997), *BnP* (Weeks *et al.*, 2001) and *CHART* (Emsley, 1999). However, there is no single application that encompasses the results of multiple structure-determination software packages into one database. The system presented here utilizes a web interface to sequester the minimal initial data needed in determining a macromolecular structure *via* SAD (Wang, 1985; Dauter *et al.*, 2002) or MAD (Hendrickson & Ogata, 1997) techniques. The data is automatically entered into a database. At this point, an experiment is started. An experiment is defined as an arbitrary run through a selected structure-determination software package or a mixture of packages for a selected protein target with the associated diffraction, cell and crystal data. A process-management system distributes experiments for structure determination among an array of computers. Appropriate crystallographic software and input parameters are selected based on the initial data, data collected during the process and past results.

1.1. Design concept

Understanding the steps that are involved in determining the three-dimensional structure of a macromolecule (Fig. 1) is key to understanding the design concept of the Automated Crystallographic System (ACrS). The system will initially address those procedures following data processing and preceding manual model completion.

Initially, two shell scripts were written, one that ran *CNS* (Brünger *et al.*, 1998) and another that ran *SOLVE/RESOLVE* (Fig. 2). Once modified phases were calculated, *ARP/wARP* (Lamzin & Wilson, 1993) was used to autotrace the protein. A successful structure solution was defined as one that had at least 70% of the polypeptide backbone traced without manual model building. The scripts were run sequentially and manually edited for each data set to reflect the current structure parameters. Five test data sets from refined high-resolution structures were used for the initial trials. Overall, there was an 80% success rate for the correct structure determination.

Manually editing scripts is acceptable when performing calculations for only a few proteins, but problems arise when high throughput is desired. The task of editing input files becomes tedious and standard paths do not work for all protein targets. In some cases, it is better to use alternative or hybrid paths constructed by crossing between various software packages (Fig. 3). A once simple script can quickly evolve into a complicated application that tries to accommodate for geometric expansion. This method is neither truly automated
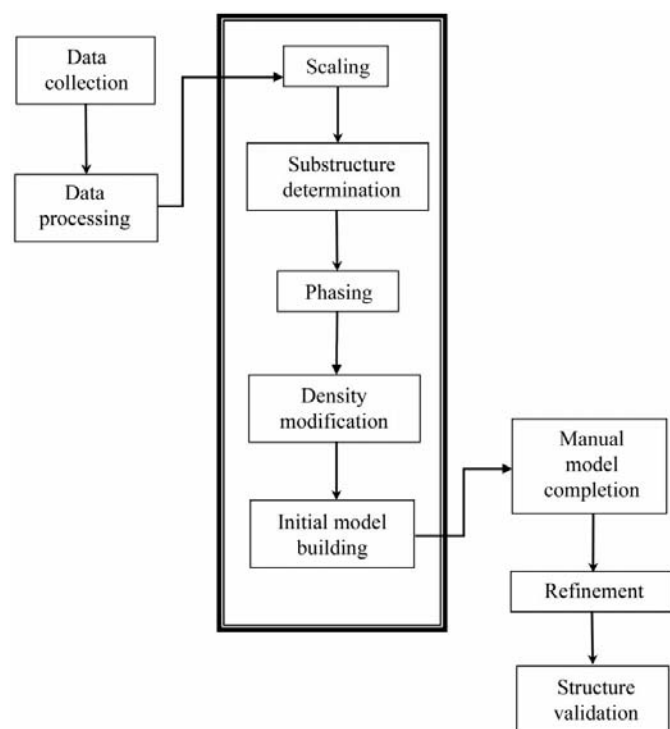
nor capable of high-throughput structure determination. These limitations led to the development of a software package that facilitates the determination of protein structures in an automated and high-throughput manner.

ACrS was designed to utilize existing crystallographic software whenever possible. However, ACrS is flexible: when new algorithms or new implementations become available, they can easily be incorporated to keep the system updated. An essential requirement of the design was to minimize user input and to eliminate user intervention during the structure-determination process. Finally, the system was designed to obtain high throughput by distributing the steps of many structure determinations across a network of processors rather than attempting to speed up the individual steps.

1.2. Objective

The objective of this work is to develop an automated high-

Figure 1
Steps of structure determination. Steps from scaling through initial model building (highlighted by the double frame) are addressed by the ACrS and run without user intervention. Data collection, processing, manual model completion, refinement and structure validation are not administered by the ACrS.
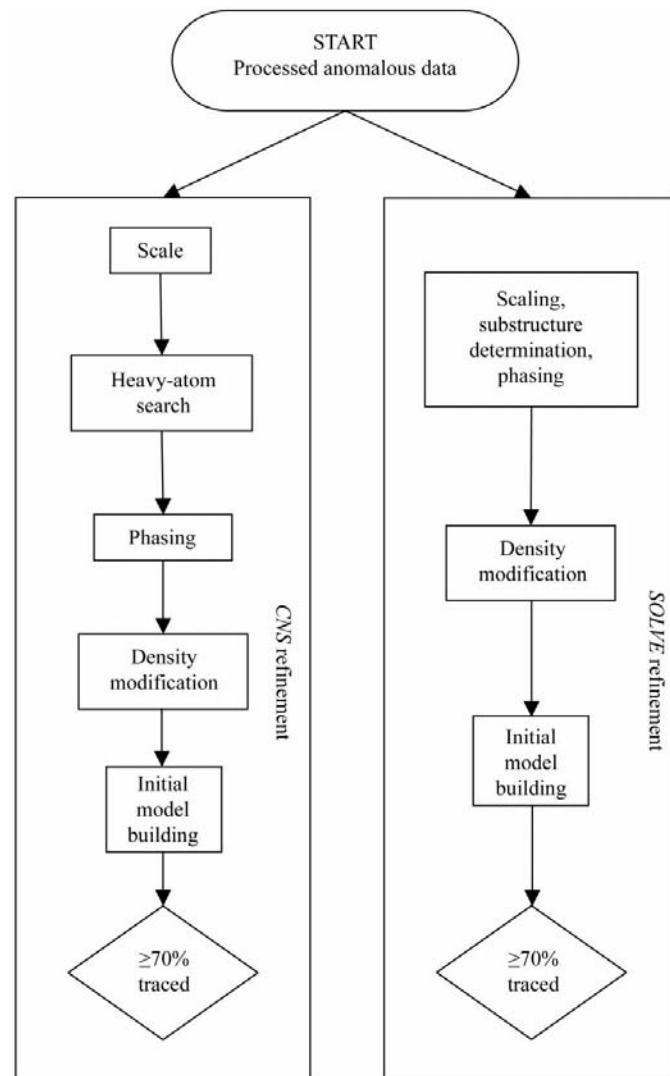

Figure 2
Simple pathways. Illustration of two simple pathways: one utilizes *CNS* software and the second uses *SOLVE/RESOLVE*. Each pathway follows the steps outlined in Fig. 1 per the specified program.

throughput system to determine the three-dimensional structure of proteins that is an application service provider (ASP). Elements of this application include the following.

(i) A graphical user interface (GUI) for accessing all aspects of the structure-determination process.

(ii) A relational database for warehousing and harvesting of initial data, run-time parameters and results that can be used for final optimized structure determination.

(iii) A distributive process system providing flexibility for the following.

(*a*) A variety of pathways utilizing different algorithms from various software packages.

(*b*) Non-linear paths with crossovers between packages.

(*c*) Easy incorporation of new methods and new program versions.

(iv) Automatic decision making coupled to a relational database for optimum structure determination.

Figure 3
Complex pathways. Illustration of two previously shown simple pathways and three complex pathways mapped in red, blue and green.

Along with the stated design concept and objectives, ACrS is being developed for immediate use with other structural genomic centres. The long-term goal is for the ACrS to be utilized by conventional laboratories with in-house collected X-ray data and data collected at the synchrotron source.

2. Application service provider

2.1. Web interface

A web interface to the ACrS was constructed to give the user capabilities to access, submit and retrieve data from the ASP. The interface was constructed using a web development framework called Slither. Slither (Thiruvathukal *et al.*, 2002) is simple yet powerful and has many features found in competing frameworks such as PHP, Zope, Active Server Pages *etc.* A key benefit to Slither is that it is flexible and easy to implement as it is based on the 'higher level language' Python (Python, version 2.1). Slither is interfaced to the web-server program Apache, version 1.3.22. For these reasons, Slither was chosen as the tool to build the GUI for the ASP. The current interface has been constructed with the following modules: Access Control and Administration, Data Deposition, Search and View of Result Reports and Advanced Experimental Setup.

An access-control module was developed to keep track of user access and deposited data sets for each protein target. This module has a general account for access to all deposited protein targets and computational results.

The data-deposition process is divided into three stages: Primary Macromolecular Target Description, File Binding and Submission, and Data Confirmation. The Primary Macromolecular Description stage acquires descriptive information about the macromolecule target (such as sequence) and a tarball file (a compressed Unix tar archived file) that contains 1–*n* *SCALEPACK* (Otwinowski & Minor, 1997) merged diffraction data files. The selected tarball file is uploaded from a local computer to the server. The File Binding and Submission stage contains fields that describe the data and the manner in which the data were collected, including wavelength, scattering factors, scattering-atom number and type, unit-cell parameters and space group. Once all the parameters have been entered that describe the diffraction experiments the final stage, Data Confirmation, is entered. The Data Confirmation stage allows the user to confirm that the data entered is correct and has been received into the database. At each stage, checks are performed to ensure data validity and integrity, *i.e.* that all fields are filled or that data is not duplicated into the database.

A search and view results module allows the user to search for results in the system (the results may either be from a completed structure determination or one in the process of being determined) and view the results from scaling, substructure determination, phasing, density modification and autotracing. The user also has the ability to download the current coordinates, unmodified phases and modified phases (the phases and reflection data used for autotracing) to view on a local graphics station.
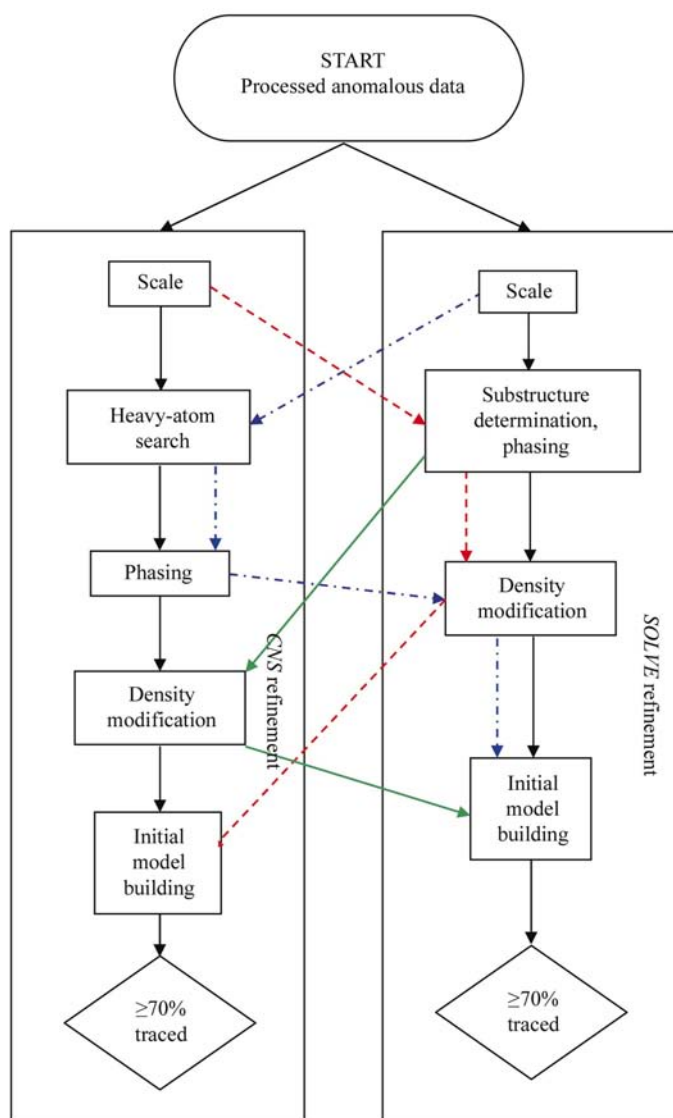
For those cases that are more difficult or for the user who desires more control of the process, an expert interface has been added. The expert interface is used to set up and run experiments in which the user can set parameters such as choice of wavelength, resolution limits, space group, number of substructure sites to determine, number of molecules per asymmetric unit and which software package to use. The initial parameters are first derived from the original experimental parameters (a database query is executed) and the user is then able to modify the parameters. The web-form interface can then submit the experiment.

2.2. Database

A relational database engine is used to warehouse all data related to structure determination. A relational database consists of data organized into logical tables. Data in one table is related to data in another table by references to key columns (ID tags). The database is designed around the mmCIF dictionary (Bourne *et al.*, 1997) and the process of directing high throughput for structure determination. For example, in the current database schema there is a table termed cell and this table corresponds to the cell category in the mmCIF dictionary. The data items (data field names) in the cell table are directly related to the items in the cell category. There are also data items that are specific to ACrS and are not defined in the mmCIF dictionary. One way to conceptualize the data schema designed is that the schema tells a story, as follows.

(i) Basic protein target information is used to uniquely track collected data.

(ii) Protein target data are associated with one or more diffraction data sets, which are associated with one or more crystal and cell properties.

(iii) An experiment is an arbitrary run through a selected structure-determination software package or a mixture of packages for the selected protein target with the associated diffraction, cell and crystal properties.

(iv) During processing of the experiment, all generated data (reflection files, maps, NCS *etc.*) are associated with the experiment.

(v) All actions taken in processing an experiment are stored in a series of log tables. Data tracked includes which applications were used, in what order *etc.*

(vi) Results of the experiment are stored in the structure statistics tables.

The database system used is MySQL, version 3.23.41, and the database interface used is MySQL-Python DBI, version 0.9. The database schema is available at the web address http://acrs.structure.northwestern.edu.

2.3. Process distribution

To increase throughput, ACrS utilizes a cluster of computers and an integrated process-distribution system to allocate processes of a related nature. The related processes are managed in an orderly fashion to enhance computer performance over the entire cluster.

2.3.1. The AntPharm system. The AntPharm system is a distributed process for managing computations on a cluster of servers. The design of AntPharm components is analogous to the inner workings of an ant colony. There is a Queen Ant and her Worker Ants. There are various types of Worker Ants who perform specific tasks in a project for the Queen Ant. When an Ant is done with its task, the next Ant will continue with its task and the succession of Worker Ants will follow until the project is completed.

An Ant is a Python routine specialized to performing a single task, such as preparing and running the crystallographic program *SOLVE* (Terwilliger & Berendzen, 1999). Each node in a computer cluster runs a collection of various types of Worker Ants that run in cycling daemon mode. Each Ant requests jobs to work on from a managing server. The managing server contains an internal table of queued jobs and their status; the Queen Ant manages this table. Each job goes through a series of states and a collection of states is referred to as a path. Initially, a job starts on the first state of a prescribed path. Jobs are created when a new experiment is started; a job is defined as a crystallographic computational experiment. Once the job is made available, Ants compete for work. The competition-based framework helps distribute the load over the entire cluster, as a working Ant will not request work for the same experiment on the same path. This type of system is readily scalable both locally and remotely and provides high throughput.

An example of a prescribed path is the CNS Path for performing computations using the *CNS* application (Brünger *et al.*, 1998). The CNS Path is composed of three distinct states: ExpSetup Ant, CNS Ant and TrailCleaner Ant. The ExpSetup Ant looks for experiments to be worked on in the database and then creates a directory to store all subsequently generated data. The CNS Ant first creates all the required input files that are generated from a set of default templates that are filled *via* database queries and then executes the calculations. When the CNS Ant has completed its work, the TrailCleaner Ant is employed to mark the current experiment completed and releases the protein target from the current path.

This system is similar to other queuing systems, in that when multiple jobs are submitted an administrator builds a queue. However, the system differs because the Ants request work when they become available instead of the queue administrator assigning the job to an available Ant(s). In traditional queue systems, scripts are used to execute a program and distribute the calculation as defined by the queue rules. The AntPharm is more advanced because it is not only a queuing and distributing system, but is also a task and project manager with the integration of a relational database. The system is also extremely flexible. The AntPharm can be modified readily to fit the tasks and projects of the computer environment.

ACrS is a fairly simple system compared with the more complex computing environment entailed in the concept of GRID computing (Foster *et al.*, 2001); an example of the latter is the Globus Project (Foster & Kesselman, 1997). The Globus Project is a massive computing environment that utilizes a 'GRID' algorithm to distribute computations across a wide

Table 1
Test data set characteristics.

APC_ID	Protein	Residues per molecule	Space group	Molecules per AU	Sites per AU	No. wavelengths
	Groel apical domain	145	$C222_1$	1	3 SeMet	4
APC250	Ac CoA transferase α subunit	220	$P6_2$	2	14 SeMet	3
	DAPK	285	$P2_12_12_1$	1	1 Hg	1
APC074	ITPase	197	$P4_32_12$	1	2 SeMet	3
APC117	Yrdc	190	$P2_1$	2	6 SeMet	3
APC078	Hypothetical	157	$P2_12_12$	2	20 SeMet	3
APC127	Cyanase	156	$P1$	10	40 SeMet	4
APC0066	Hypothetical	258	$C2$	1	4 SeMet	3
APC121	MAF	189	$P2_12_12_1$	2	12 SeMet	4
APC038	Hypothetical	299	$P2_13$	2	10 SeMet	3
APC1056	Hypothetical	185	$P6_522$	2	18 SeMet	2
APC5007	EC1988	336	$P4_12_12$	1	14 SeMet	2
	DNA Pol II	783	$P2_12_12$	1	5 Hg	1
APC047	Spermidine synthase	296	$P2_12_12$	4	40 SeMet	3
APC5006	EC29	134	$P6_222$	1	6 SeMet	3
APC20010	TA726	255	$P2_1$	6	78 SeMet	2
APC20012	TA649	207	$P2_12_12$	4	40 SeMet	2
APC409	Hypothetical	271	$P2_1$	4	18 Br	2
	ZnPt	141	$P2_12_12_1$	1	∼6 Pt/Zn	1

Table 2
Test data set results: percent of structure completed.

NT, not tested; DNW, did not work. Percentage values in bold are structures that have >70% of their main chain traced and percentages in italics are structures where the main-chain trace is <70% but the structure is essentially correct.

APC_ID	Protein	Overall resolution (Å)	Phasing resolution (Å)	SOLVE/RESOLVE1.2/ARP	CNS/ARP	SOLVE/RESOLVE2.02
	Groel apical domain	20–1.7	20–2.5	**94**	**96**	NT
APC250	Ac CoA transferase α subunit	20–1.9	20–3.3	DNW	DNW	**95**†
	DAPK	20–1.8	20–2.5	**88**	**90**	NT
APC074	ITPase	20–1.8	20–2.5	**84**	**88**	**87**
APC117	Yrdc	20–1.95	20–2.5	DNW	**81**	**83**
APC078	Hypothetical	20–2.0	20–2.5	**73**	*56*	**82**
APC127	Cyanase	20–1.65	20–2.5	DNW	DNW	**79**†
APC0066	Hypothetical	20–2.5	20–2.5	NT	*68*	**79**
APC121	MAF	20–1.8	20–2.5	**73**	DNW	*65*
APC038	Hypothetical	20–1.9	20–2.3	NT	DNW	*69*
APC1056	Hypothetical	20–2.7	20–2.7	NT	*39*	*59*†
APC5007	Ec1988	20–2.6	20–2.6	NT	*58*†	*64*†
	DNA Pol II	20–2	20–2.5	NT	NT	*39*
APC047	Spermidine synthase	20–2.3	20–2.5	DNW	*40*	*35*
APC5006	EC29	20–2.2	20–2.6	NT	DNW	*58*
APC20010	TA726	20–2.28	20–2.5	NT	DNW	DNW
APC20012	TA649	20–2.1	20–2.5	NT	DNW	DNW
APC409	Hypothetical	20–2.23	20–2.5	NT	DNW	DNW
	ZnPt	20–1.9	20–2.7	DNW	DNW	DNW

† MAD data collected, but solved as SAD.

3. Results

A series of experiments have been executed using the ACrS. A total of 19 MAD/SAD data sets have been submitted to the AntPharm. The protein targets cover a range of complexities, with varying numbers of substructure sites, space groups and numbers of molecules per asymmetric unit (Table 1 lists the properties of the data sets). Experiments were performed using three different paths: one path utilized *SOLVE/RESOLVE* (non-building version) and *ARP*, another path used *CNS* and *ARP*, and the last path used *SOLVE/RESOLVE* (building version). The default parameters distributed with the various software packages were used for all calculations and only resolution values were changed depending on the data-set limitations as shown in Table 2.

The success of an experiment is broken down into two levels. The first level of success was judged by how complete the initial model was when compared with the total number of expected residues (calculated from the total number of residues multiplied by the expected number of molecules per asymmetric unit). If an initial model had reached the arbitrary level of at least 70% of the polypeptide backbone automatically built with the building software, that experiment was marked as being successful and the model could move on to the next stage of manual model completion. The second level is when the substructure model and phases are essentially correct but are not good enough for the model-building programs to build 70% of the model. This scenario can happen when the substructure is incomplete, there is a lack of data or when there is disorder in the protein which cannot be interpreted by the building software. Overall, 79% of the protein targets tested fit into the two levels of success; 60% of these targets are at least 70% complete and proceeded to the manual model-building stage. Table 2 lists the corresponding results from all of the automated experiments. The table is ordered from most to least successful.

In the case of APC127 (cyanase), the structure was determined after using only the peak-energy data (SAD experiment). With the use of high-resolution native data, the phases

array of computers that can be local or remote. Globus also relies on a modified version of MPI, a parallel interface, to take a single computation task, split it into subtasks and run them in parallel on the 'GRID' and then later merge the subtasks to complete the initial computation. The AntPharm and ACrS could potentially be tailored to work within a system like Globus, although at present Globus is beyond the scope of this current project.

were then extended and NCS electron-density averaging assisted the automated model-building step, resulting in a structure trace that was 79% complete. The case of APC038 ended with only 69% of the structure automatically traced; the lack of more complete residues was a consequence of domain disorder as was subsequently determined from viewing the electron-density maps. Protein target APC1056 had good phasing statistics (high figure of merit and sites that fit in the experimental anomalous Patterson map), but the initial model was only 59% complete. The incomplete model is a consequence of the lack of experimental measured data (resolution limits 20–2.7 Å). The protein target ZnPt apparently did not succeed owing to the presence of multiple anomalous scattering atoms that misled Patterson map interpretation. The substructure of targets APC20010, APC20012 and APC409 was not determined correctly with the current implemented substructure programs within ACrS.

4. Discussion

A total of 19 MAD/SAD data sets have been submitted to the ACrS. In each case, the process began with the deposition of minimal data into the database from the ACrS web interface. From the initial data, the molecular weight, Matthews coefficient, solvent content and number of molecules per asymmetric unit are estimated and stored in the database automatically. The processed data (*SCALEPACK* format) is then analyzed to determine the optimal resolution for scaling, phasing and map interpretation (model building). These values are reported back to the database for later use. The data are scaled for the experiment type, either MAD or SAD, using the local scaling algorithm within *SOLVE* (Terwilliger & Berendzen, 1999). The scaled data are then passed to both *CNS* and *SOLVE* for substructure determination, phasing and density modification. The last three steps are carried out with each of the two packages. The final step is autotracing, currently performed with the *RESOLVE* program. During this process, the application output (logs, results *etc.*) files are parsed for results to be automatically updated into the database. A solution is judged to be correct if 70% of the polypeptide backbone was autotraced.

In the current implementation, the degree of decision making performed by the system is limited. Presently, decisions are made for selecting the optimized resolution settings, the number of molecules and substructure sites per asymmetric unit and an appropriate solvent content. The ScaleAnt was designed to encompass multiple scaling programs into one module. This module is capable of running scaling routines from *CNS* or *SOLVE* and variations within these crystallographic packages. The ScaleAnt has built-in decision-making abilities to perform data rejection and resolution cutoff, which can be determined from analysis of individual reflections, data completeness, anomalous differences and dispersive differences. These statistics are reported to the database for evaluation and selection of the optimal data parameters. The utilized crystallographic software does not provide the decision-making engine in the ACrS; all decision making is currently handled within the ACrS code or the Crystallographic Protocol Library (CPL is an internal library that will be distributed with the ACrS).

Over time, additional crystallographic Ants will be developed. For example, a SubstructureAnt is being developed which will continue to use the programs *SOLVE* and *CNS* but also incorporate *SHELXD* and *SnB*. This Ant will determine which software package to use and which wavelength(s) are selected for substructure determination. Targets APC20010, APC20012 and APC409 are proteins that failed during automated anomalous substructure determination and would greatly benefit with the implementation of a more robust SubstructureAnt.

This logical decision making is guided by parameters and results that are stored in the relational database. The process of formulating the decision-making criteria is being established empirically from the existing test data. Although ACrS currently is restricted to MAD or SAD phasing, it could readily be extended to incorporate other phasing techniques.

5. Conclusion

In conclusion, a new system, ACrS, has been developed that will meet the requirements of automation for high-throughput structure determination. The system utilizes a web-based GUI, a relational database for all data warehousing, data harvesting and eventually structure-determination optimization, and a distributed process manager that is simple, flexible and allows for the integration of new crystallographic routines. In the current implementation, some low-level decision making has been incorporated. Within this initial framework, more crystallographic Ants are being developed (*i.e.* SubstructureAnt, RefinementAnt) and optimized.

research papers

References

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. & Bourne, P. E. (2000). *Nucleic Acids Res.* **28**, 235–242.

Bourne, P. E., Berman, H. M., Watenpaugh, K., Westbrook. J. D. & Fitzgerald, P. M. D. (1997). *Methods Enzymol.* **277**, 571–590.

Brünger, A. T., Adams, P. D., Clore, G. M., Gros, P., Grosse-Kunstleve, R. W., Jiang, J.-S., Kuszewski, J., Nilges, N., Pannu, N. S., Read, R. J., Rice, L. M., Simonson, T. & Warren, G. L. (1998). *Acta Cryst.* D**54**, 905–921.

Dauter, Z., Dauter, M. & Dodson, E. J. (2002). *Acta Cryst.* D**58**, 494–506.

Emsley, P. (1999). *CCP4 Newsl.* **36**, http://www.ccp4.ac.uk/newsletter36/08_chart.html.

Fortelle, E. de la & Bricogne, G. (1997). *Methods Enzymol.* **276**, 472–494.

Foster, I. & Kesselman, C. (1997). *Int. J. Supercomp. Appl. High Perf. Comput.* **11**, 115–128.

Foster, I., Kesselman, C. & Tuecke, S. (2001). *Int. J. Supercomp. Appl. High Perf. Comput.* **15**, 200–222.

Hendrickson, W. A. & Ogata, C. M. (1997). *Methods Enzymol.* **276**, 494–523.

Lamzin, V. S. & Wilson, K. S. (1993). *Acta Cryst.* D**49**, 129–147.

Otwinowski, Z. & Minor, W. (1997). *Methods Enzymol.* **276**, 307–326.

Terwilliger, T. C. (1999). *Acta Cryst.* D**55**, 1863–1871.

Terwilliger, T. C. & Berendzen, J. (1999). *Acta Cryst.* D**55**, 849–861.

Thiruvathukal, G. K., Christopher, T. W. & Shafaee, J. P. (2002). *Web Programming in Python: Techniques for Integrating Linux, Apache and MySQL.* Upper Saddle River: Prentice Hall.

Wang, B.-C. (1985). *Methods Enzymol.* **115**, 90–112.

Weeks, C. M., Rappeleye, J., Furey, W., Miller, R., Potter, S. A., Smith, G. D. & Xu, H. (2001). Annu. Meet. Am. Crystallogr. Assoc., Abstract W0264.