Acta Crystallographica Section F
Structural Biology and Crystallization Communications
ISSN 1744-3091

Deposit3D: a tool for automating structure depositions to the Protein Data Bank

J. Badger,*‡ J. Hendle, S. K. Burley and C. R. Kissinger

SGX Inc., 10505 Roselle Street, San Diego, CA 92121, USA

‡ Current address: ActiveSight and Molecular Images, 4045 Sorrento Valley Boulevard, San Diego, CA 92121, USA.

Correspondence e-mail: jbadger@active-sight.com

Received 25 April 2005
Accepted 13 August 2005
Online 31 August 2005

Almost all successful protein structure-determination projects in the public sector culminate in a structure deposition to the Protein Data Bank (PDB). In order to expedite the deposition proces, *Deposit3D* has been developed. This command-line script calculates or gathers all the required structure-deposition information and outputs this data into a mmCIF file for subsequent upload through the RCSB PDB *ADIT* interface. *Deposit3D* might be particularly useful for structural genomics pipeline projects because it allows workers involved with various stages of a structure-determination project to pool their different categories of annotation information before starting a deposition session.

1. The crystallographic problem

Three-dimensional macromolecular structure data in the public domain is available through the Protein Data Bank (PDB; Berman *et al.*, 2000), a corpus which currently contains ∼30 300 entries and is growing at a rate of ∼100 new structures per week. Structure deposition to the PDB is now a condition of acceptance for most journals and is a requirement for funding by the NIH. A consequence of advances in automated structure-solution methodologies is that protein structures are frequently completed with a few days once suitable data has been collected. In particular, many of the structural genomics centers have developed computational platforms for rapid structure determination and would be assisted by the availability of a streamlined mechanism for structure deposition. Clearly, the time-scale and level of effort involved in depositing a structure to the PDB should remain commensurate with accelerating levels of throughput.

At the present time, two different interfaces are available for depositing structure data to the PDB. Over the past year, ∼85% of new structures were entered through the *ADIT* system (Westbrook *et al.*, 2003) managed by the RCSB. The remaining structures were entered through the *AUTODEP* system (Keller *et al.*, 1998) managed by the EBI. Manual entry of data through the *ADIT* interface is relatively slow and, since the amount of time a depositor is willing to spend on a deposition is limited, unavoidably restricts the quantity of information that is provided. Errors and inconsistencies are likely to result from this mode of data entry which, when detected, require resolution through dialogue with the PDB annotator. Both *ADIT* and *AUTODEP* provide some degree of automated data capture by parsing the headers of the submitted coordinate files. However, the information that is obtained in this way is usually limited to structure statistics obtained from the final round of refinement. In addition, not all of the currently available refinement software create PDB file headers that meet the required specification. More recently, 'data-harvesting' approaches have been implemented to allow automated data capture over more of the structure-solution process. In the data-harvesting methodology which has been implemented throughout the *CCP*4 suite of programs, special standardized results files are created from a program run. The *AUTODEP* interface provides a means for uploading data-harvesting files as part of the deposition process (Keller *et al.*, 1998). For use with *ADIT*, a comprehensive parsing program that extracts information from both data-harvesting and various program log files has been made available (Yang *et al.*, 2004). The mmCIF file containing this information that is produced by this

program may be uploaded through the *ADIT* interface. Although clearly beneficial in many ways, data-harvesting approaches to automated structure deposition require that the crystallographer responsible for the structure determination creates and maintains the necessary data-harvesting files and takes some care to ensure that the correct file set is used for structure deposition.

2. Method of solution

To automate the deposition of protein structures, we have (i) developed a mmCIF deposition file, into which all necessary annotation data may be entered prior to upload through the RCSB/PDB *ADIT* deposition interface, and (ii) written an efficient tool (*Deposit3D*) for generating this deposition file.

The data items used in the deposition file (see supplementary data[1]) employ the mmCIF dictionaries and standards (Bourne *et al.*, 1997) as augmented by the PDB exchange dictionary (http://mmcif.pdb.org/dictionaries/mmcif_pdbx.cif/index/index.html). Once all information in this file has been entered, it provides an operationally complete basis for the RCSB/PDB to process the structure deposition (*i.e.* it contains all mandatory deposition information). A huge number of data items could be included in the deposition file, but to make it a practical process it is necessary to select the scientifically most useful items. The data items that we have included are generally those that allow the user to connect the structure with other sequences/data and that allow some evaluation of the structure quality. Should the depositor wish to add information not included in the deposition file, this may be entered through the graphical interface in the RCSB/PDB *ADIT* deposition session.

To generate a deposition file for *ADIT*, we developed a command-line tool (*Deposit3D*) that only requires that the depositor provide a PDB-format coordinate file, an X-ray data file in the *CCP*4 MTZ format and a sequence file in FASTA format. Optionally, the user may complete the population of numerical data in the deposition file by providing a log file from the data-merging program *SCALA* from *CCP*4 or *SCALEPACK* (Otwinowski & Minor, 1997). A special template file has been devised to allow the depositor to enter 'non-electronic' information that is not readily obtainable through automated processes (for example, authorship and citation data as well as gene and protein names). The use of the template file with *Deposit3D* is optional (the missing information may be completed in the *ADIT* interface), but it provides a particularly efficient approach for depositing sets of closely related structures, since most of this content is identical for each structure. For structures determined through strongly pipelined processes (for example, structural genomics initiatives) it may be the case that not all of the required deposition information pertaining to the structure is known to a single individual. In this case, the template file provides a place in which the structure-determination staff can pool all required information prior to starting the PDB deposition session.

A novel feature of *Deposit3D* is that it applies a concept originating in our earlier work on a structure-validation/deposition system for internal use at SGX (Badger & Hendle, 2002): most of the annotation information (for example, the crystallographic *R* factor) is calculated from the input data rather than entered by the depositor. Most obviously, this strategy ensures that the resulting statistics are self-consistent and reliably relate the coordinate set and diffraction data that are being deposited. Although *Deposit3D* uses *CCP*4 as an engine for structure calculation (see §3), it is emphasized that this does not limit the system to use with structures solved using the *CCP*4 program suite. A slightly earlier version of the deposition-file format and the forerunner of this tool were used to deposit to the Protein Data Bank ∼70 of the structures resulting from the SGX bacterial genomics project (Badger *et al.*, 2005).

The uses of the mmCIF files produced by *Deposit3D* may extend beyond deposition of a structure though the RCSB PDB *ADIT* interface. Since the file contains a self-contained and relatively detailed account of a completed structure, it could potentially form a computer-generated basis for a short structure report in (for example) *Acta Crystallographica Section F* (http://journals.iucr.org/f/services/structuralcommunications/). Any data items required for publication that are not contained with the current file could be added relatively easily by including additional mmCIF tags from the appropriate dictionaries. In addition, *Deposit3D* could be used to provide convenient records of completed structures within corporate structure-determination environments.

3. Software and hardware environment

Deposit3D is a self-contained ∼3300-line Python script. The operation of the script includes the execution of the *CCP*4 (Collaborative Computational Project, Number 4, 1994) programs *LIBCHECK*, *MATTHEWS_COEF*, *MTZDMP*, *MTZ2VARIOUS*, *REFMAC*5 and *UNIQUE* in a C-shell environment. *Deposit3D* was developed on RedHat Linux operating systems, but should also run without modification in any Unix-like environment. A valid *CCP*4 installation is required to run *Deposit3D*; testing was carried out with *CCP*4 v.5.0.2.

The run time for the process depends on the resolution of the input data and the size of the input structure, but does not usually exceed 5 min on desktop machines typically used for protein crystal structure determination.

4. Documentation and availability

Transparency of operation, maintainability and extensions to *Deposit3D* are facilitated by the non-compiled nature of the Python script. Built-in user documentation includes a prompt for the required command-line input files if *Deposit3D* is executed without any inputs and a 'help' command that supplies more information on the input file formats. Explanatory error messages are directed to the user's terminal window in the event of an operational failure. The template file used for non-electronic data entry is self-documented with explanations and examples of the required data fields.

The *Deposit3D* script and the associated template file are freely available as supplementary material to this paper (corresponding to the files deposit3d.py and deposit3d.template[1]) or by email request to jbadger@active-sight.com. Except for prohibition against removing information citing authorship, the script and file may be modified as required. After downloading *Deposit3D*, the only required configuration is to ensure that the script is executable (UNIX command `chmod +x deposit3d.py`) and to edit the *ccp4installation* variable to create a path to a local *CCP*4 installation.

We thank John Westbrook and Kyle Burkhardt of the RCSB PDB for their assistance in developing the deposition-file format for applications to structures resulting from the SGX bacterial structural genomics program project (Badger *et al.*, 2005).

References

Badger, J. *et al.* (2005). *Proteins*, **60**, 787–796.
Badger, J. & Hendle, J. (2002). *Acta Cryst.* D**58**, 284–291.

[1] Supplementary material is available from Crystallography Journals Online (Reference: HI5569).

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., McMahon, B., Watenpaugh, K. D., Westbrook, J. D. & Fitzgerald, P. M. D. (1997). *Methods Enzymol.* **277**, 571–590.
Collaborative Computational Project, Number 4 (1994). *Acta Cryst.* D**50**, 760–763.
Keller, P. A., Henrick, K., McNeil, P., Moodie, S. & Barton, G. J. (1998). *Acta Cryst.* D**54**, 1105–1108.
Otwinowski, Z. & Minor, W. (1997). *Methods Enzymol.* **276**, 307–326.
Westbrook, J., Zeng, K., Burkhardt, K. & Berman, H. M. (2003). *Methods Enzymol.* **374**, 370–385.
Yang, H., Guranovic, V., Dutta, S., Feng, Z., Berman, H. M. & Westbrook, J. D. (2004). *Acta Cryst.* D**60**, 1833–1839.