Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

AutoDep: a web-based system for deposition and validation of macromolecular structural information

Dawei Lin,[a]† Nancy O. Manning,[a]† Jiansheng Jiang,[a] Enrique E. Abola,[a]‡ David Stampf,[b] Jaime Prilusky[c] and Joel L. Sussman[a,d]*

[a]Biology Department, Brookhaven National Laboratory, Upton, NY 11973-5000, USA, [b]Information Technology Division, Brookhaven National Laboratory, Upton, NY 11973-5000, USA, [c]Bioinformatics Unit, Weizmann Institute of Science, Rehovot 76100, Israel, and [d]Department of Structural Biology, Weizmann Institute of Science, Rehovot 76100, Israel

† These authors contributed equally to this work.
‡ Present address, The Department of Molecular Biology, The Scripps Research Institute, La Jolla, CA 92037, USA.
§ Present address, The Department of Molecular Biology, The Scripps Research Institute, La Jolla, CA 92037, USA.

Correspondence e-mail: joel.sussman@weizmann.ac.il

Received 28 December 1999
Accepted 12 April 2000

This paper describes the design and full implementation of a new concept in data deposition and validation: *AutoDep* (copyright Brookhaven Science Associates LLC). *AutoDep* changes the traditional procedure for data acceptance and validation of the primary databases into an interactive depositor-driven operation which almost eliminates the delay between the acceptance of the data and its public release. The system takes full advantage of the knowledge and expertise of the experimenters, rather than relying on the database curators for the complete and accurate description of the structural experiment and its results. *AutoDep*, developed by the Protein Data Bank at Brookhaven National Laboratory (BNL) as a flexible and portable system, has already been adopted by other primary databases and implemented on different platforms/operating systems. *AutoDep* was introduced at BNL in 1996 [see Manning (1996), *Protein Data Bank Quart. Newslett.* **77**, 2 (ftp://ftp.rcsb.org/pub/pdb/doc/newsletters/bnl/newsletter96jul/newslttr.txt); Manning (1996), *Protein Data Bank Quart. Newslett.* **78**, 2 (ftp://ftp.rcsb.org/pub/pdb/doc/newsletters/bnl/newsletter96oct/newslttr.txt)].

1. Introduction

The Protein Data Bank (PDB) is a database containing experimentally determined three-dimensional structures of proteins, nucleic acids and other biological macromolecules (Abola *et al.*, 1987, 1997; Bernstein *et al.*, 1977; Sussman *et al.*, 1998). Established at Brookhaven National Laboratory (BNL) in 1971, the PDB has a 28-year history of service to a global community of researchers, educators and students in a wide variety of scientific disciplines. As of December 1999, there were over 11 000 structures archived by the PDB (Fig. 1; Table 1).

The explosive growth in the number of structural reports on biological macromolecules along with establishment of new initiatives for high-throughput protein crystallography highlight the critical need for the development of fast, efficient and powerful systems for acquiring data for deposition and archiving in structural databases. There was a tremendous need to design a tool that provided a direct means of entering and validating the extremely complex data while also facilitating rapid release of the submitted data in order to meet public demands. The tool had to be designed for the research scientist expert in his or her own field, if not in computing. It had to be simple to learn and use and should take advantage of previously submitted information wherever possible.

The Protein Data Bank harnessed emerging Internet technology and developed *AutoDep*, the first World Wide Web

Table 1
PDB archives, December 1999.

PDB archive contents as of December 1999	
Atomic coordinate entries	11207
Structure-factor files	3244
NMR restraint files	663
Molecule type	
Proteins, peptides and viruses	9944
Protein/nucleic acid complexes	483
Nucleic acids	762
Carbohydrates	18
Experimental technique	
Diffraction	9193
NMR	1776
Theoretical modeling	238

based database-submission tool. Released in October 1996, *AutoDep* quickly became the predominant method for submitting structural data to the Protein Data Bank. In just three and a half months, well over 50% of all new submissions were deposited *via AutoDep* (Sussman, 1997; ftp://ftp.rcsb.org/pub/pdb/doc/newsletters/bnl/newsletter97jan/newslttr.txt). By 1998, the time between submission and release of the data had dropped from 120 days to just a few days for data not placed on hold at the depositor's request; entries to be released upon publication were released virtually simultaneously with their appearance in print. *AutoDep* moves the burden of tedious data-formatting issues to computer programs, freeing the researcher who had been studying the protein's structure to focus on the completeness and accuracy of the structural entry and enabling the PDB's professional staff to deal with issues of representation and curation of the data. *AutoDep* also provides interactive feedback to the depositor, so that any errors or inadequacies of the data may be addressed immediately. Initially serving as an input tool with some syntax verification, *AutoDep* was enhanced and integrated with the PDB validation and release programs, forming a comprehensive macromolecular structure information deposition and release system. *AutoDep*'s generic approach to data input may be easily adapted by other resources.

2. Overview of the *AutoDep* system

2.1. Web-based system

The *AutoDep* system uses a client–server architecture running over the World Wide Web (WWW). HyperText Markup Language (HTML) forms are generated at the server side and sent to the user's site. On the user, or client side, depositors use their favorite web browser to access and complete the forms, then send their data through the Common Gateway Interface (CGI) to the server. The server's application programs process the information and send the results back to the users, again through their web browser. Therefore, *AutoDep* may be viewed using a browser of the user's choosing no matter which computer platform is used, resulting in a shallow learning curve. Researchers use web forms to submit their data over the web quickly and easily. Additionally, numerous hyperlinks put in-depth information at the fingertips of the user. Each interactive session is password protected and may be interrupted and resumed at will.

2.2. User-friendly interface

AutoDep was designed to present the complex submission process in a logical and easily understandable manner. Indeed, most first-time users work their way through the entire process with only minimal, if any, reading of the documentation or contact with the PDB Help Desk. Information is organized into separate sections, or pages, for easy data preparation. There are approximately 800 individual items in *AutoDep*, presented in 15 sections. Every question is displayed with a sample answer, a link to detailed help and an indication as to whether or not the item is mandatory for a complete submission. Questions are context-specific, dependent upon the particular structure or method used. There are many instances where a large portion of the questions can be preloaded with answers that need only be verified by the depositor. Color-coding and a logical system of red crosses and green check marks are used to indicate whether a question or entire section has been completed. Basic syntax verification is handled at the level of section completion and the user is immediately notified if a given answer is not consistent with the expected syntax. The web forms are customized for the experiment being represented, simplifying the submission process. For example, when submitting data from an X-ray crystallographic experiment, the user does not need to answer questions relevant only to NMR experiments. Transferring files to the PDB is also managed by *AutoDep*. The depositor merely follows the simple stepwise procedure for uploading coordinate and experimental data files.

2.3. *AutoDep* is an interactive system

AutoDep automatically verifies the syntax and completeness of the given values as the user fills out each section. If the given information is wrong or not appropriate, *AutoDep* returns a warning with hints to help the depositor fix the

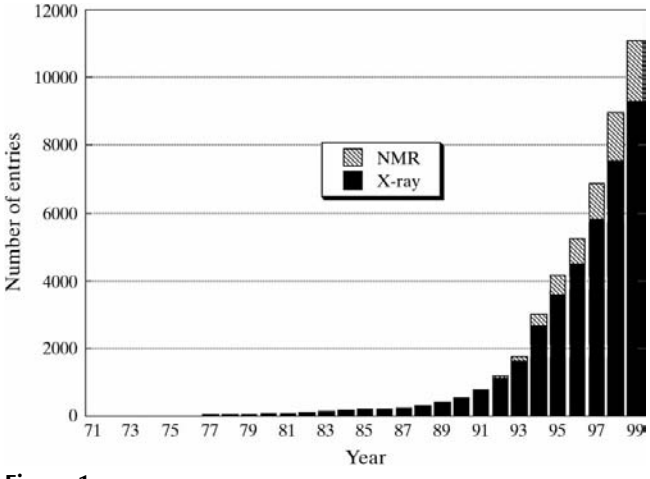


Figure 1
PDB coordinate entries available per year.

problem. In many cases, there are cross-checks performed with other data already provided. At any time, the user may view the PDB entry as it would appear with the given information. After the coordinate file has been transferred to the PDB and the entire form has been completely and correctly filled out, the user submits the structure for validation and is notified by e-mail when validation is complete. The user is then able to review the output of all validation checks and decide whether to complete submission at that time or to go back and further refine the structure before submission and release. If the user opts to further refine the structure, all of the information in the deposition is preserved and may be recalled instantly.

2.4. *AutoDep* accepts electronic input

AutoDep was designed to facilitate easy and accurate data input. The best way to accomplish this is to electronically capture and transfer information wherever possible. We carried out a dialogue with the authors of several of the most popular refinement and viewing programs, including *CNS* (Brunger *et al.*, 1997, 1998), *SHELX* (Sheldrick, 1997), *TNT* (Tronrud *et al.*, 1987), *REFMAC* (Murshudov *et al.*, 1999), *X-PLOR* (Brünger, 1992) and *O* (Jones *et al.*, 1991). Our purpose was to identify the data output by their individual programs that might best assist the sophisticated user of the structural entry in evaluating the quality of that model. A major benefit of this exchange was that the program authors themselves carefully considered which refinement information should be archived, thereby guiding development of PDB content. We redesigned the contents of the PDB file to present refinement statistics unique to each refinement program (see the PDB Contents Guide, Abola *et al.*, 1996, ftp://ftp.rcsb.org/pub/pdb/doc/format_descriptions/Contents_Guide_21.html) and the programs were re-engineered in order to output these statistics along with the coordinates for easy submission to the PDB. As a result, depositors now easily transfer and merge the output of these programs, including refinement statistics, unit-cell data and some bonding information. An added benefit of this computer-to-computer transfer is that the entries are more uniformly complete and correct.

Another form of electronic input is available at the start of an *AutoDep* submission. Following the same spirit as income-tax preparation software programs, *AutoDep* can reuse archived information. This is a huge improvement over submission systems prior to *AutoDep*. *AutoDep* may be pre-populated from a released PDB structural entry, from one of a user's previous submissions (especially useful when submitting a related series of structures) or from a file submitted in PDB format.

2.5. *AutoDep* includes validation

AutoDep ensures that all mandatory data have been supplied by the depositor. As defined by the PDB, mandatory records are classified into four groups (see Abola & Manning, 1997; ftp://ftp.rcsb.org/pub/pdb/doc/newsletters/bnl/newsletter97oct/newslttr.txt).

(i) Administrative details, including user identification and special instructions.

(ii) Data needed for validation, such as the unit-cell parameters, space-group information, scale matrix and coordinates.

(iii) Data describing the molecular model and its fit to the experimental data. This includes a description of the molecule, its biological source, amino-acid sequence, catalytic site, resolution range, R value, free R value and r.m.s.d.s.

(iv) Description of the experiment used to generate the model, including experiment type, resolution range, completeness of data, $R_{\rm merge}$ or $R_{\rm sym}$ *etc*.

AutoDep is integrated into the PDB's validation suite, which includes PDB-developed checking programs and the *WHAT_CHECK* (Hooft *et al.*, 1996) implementation of the *WHAT IF* suite (Vriend, 1990). After passing completeness and syntax checking, submitted data are checked against these programs and generated reports are returned to the depositor over the web. *AutoDep* may be easily integrated with other commonly used structural validation programs.

2.6. *AutoDep* is dictionary driven

AutoDep's flexibility is a major advantage. CIF- (Hall *et al.*, 1991) like dictionaries completely define the user interface and specify verification rules. The web pages are dynamically generated based on these dictionaries. Changes to the interface are easily accomplished by editing the dictionaries. This modularity and flexibility simplifies development and maintenance. Indeed, the BioMagResBank (BMRB; http://www.bmrb.wisc.edu/; Seavey *et al.*, 1991) has adopted the *AutoDep* program suite to use as an NMR data deposition

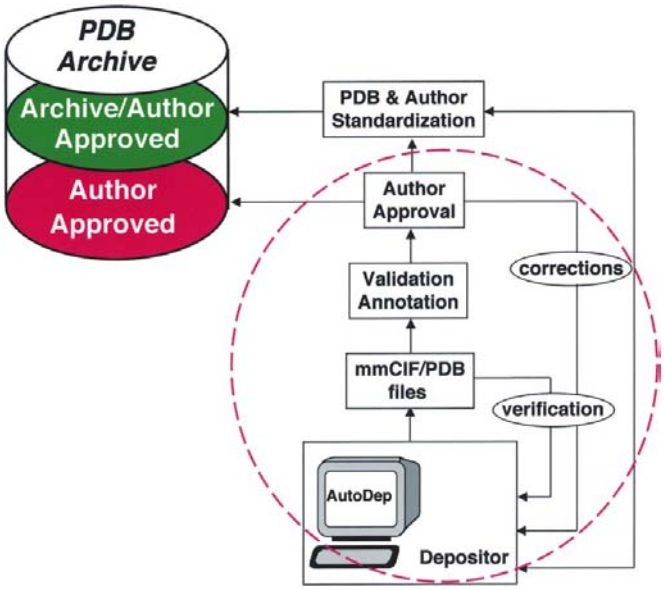


Figure 2
PDB web-based submisson *via AutoDep* facilitates releasing entries by using a layered approach making it possible to automatically release entries on publication, as indicated in the portion of the figure enclosed in a dashed circle.

tool. The dictionary also separates the user-interface issues from the content of the deposition, allowing new technologies such as Java to determine the user interface in the future while preserving the information content of the dictionary.

2.7. *AutoDep* is portable

AutoDep is designed to be fully portable. The EMBL Outstation European Bioinformatics Institute serves as a sister deposition site (http://autodep.ebi.ac.uk), and received approximately 25% of all submissions in June 1999. Additional deposition sites may be established easily.

The portable code of *AutoDep* allows programs to be updated easily. Updates are performed automatically by a mirror script between master site and mirror site. This mechanism ensures that all users enjoy exactly the same service, regardless of access site.

3. Implementation of *AutoDep*

There are four major functionalities that must be seamlessly handled by *AutoDep*. They are the initialization and security system, data entry, validation and release. Each of these will be described in detail in the following sections.

3.1. Initialization and security system

3.1.1. Initialization. There are four ways to begin an *AutoDep* submission. The first way is to initialize *AutoDep* by pre-loading the deposition form with information contained in a released PDB entry. This is useful in cases where the user is submitting a similar or related structure, such as the same enzyme with a different substrate or another mutant of a previously released structure. The second is to initialize *AutoDep* based on a previous submission. *AutoDep* pre-loads the new deposition form with data previously submitted by the user, greatly simplifying submission of a related set of structures. *AutoDep* may be started from scratch, beginning with a blank deposition form. Finally, a user may return to an incomplete submission from where he/she left off by use of the continuation option.

3.1.2. Security system. *AutoDep* issues a unique identifier, or deposition ID, to each deposition and the user chooses a password that is required for subsequent logins to the same deposition. *AutoDep*, which uses CGI scripts and HyperText Transfer Protocol (HTTP), uses this identifier and password to maintain consistent states between server and client during a deposition session. This is needed as HTTP is stateless, *i.e.* each interaction on a web server is completely independent from the interactions before and after it, and CGI programs are short-lived, ending when the web page is drawn. All variables are lost when the CGI program ends; consequently, no connection remains between client and server after each interaction. In other words, the *AutoDep* server would have no way of knowing which request belongs to which deposition. To solve this problem, an encrypted text string containing the password and deposition ID serves as a connection identifier.

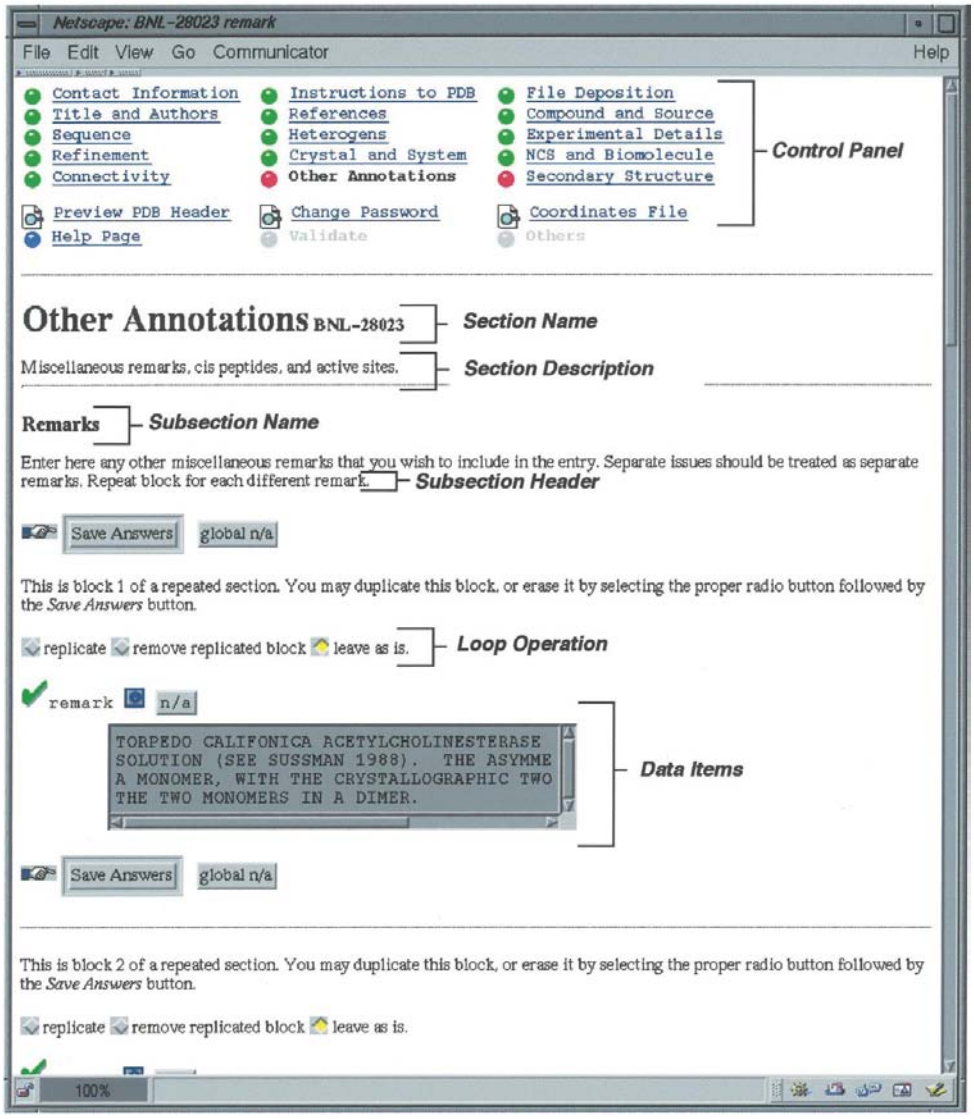


Figure 3
The layout of the *AutoDep* web pages facilitates and directs the deposition process. At the top of each page is the Control Panel that lists the sections, indicates the status of each section and contains several utility links.

This encrypted string is embedded into every HTML form throughout *AutoDep*. The program uses this string to match a deposition area on the server side, which has all the deposition and state information for each particular deposition, to the correct client. The use of this hidden ID in the HTML pages is transparent to users. Neither data nor 'cookies' are stored on the user's computer, preserving their privacy.

If an *AutoDep* session is inactive for more than 30 min, the user must re-input the correct password before continuing. This is to prevent other people from accessing a user's deposition form from the information cached in the user's web browser.

3.2. Data entry

3.2.1. Web-page layout and status control. The layout of the *AutoDep* web pages facilitates and directs the deposition process (see Figs. 2 and 3 and *Appendix A* for a guided tour of *AutoDep*). At the top of each page is the Control Panel that lists all the sections of the deposition form, indicates the status of each section and contains several utility links.

Colored status markers in the Control Panel indicate the progress of the *AutoDep* deposition. If a section is not active, its name is preceded by a gray dot. If a section is not complete or there is something unacceptable within it, its status marker is red. When a section is successfully completed, its marker turns green. Locked sections display a lock icon. The utility links 'Preview PDB Header', 'Change Password', 'View Coordinate file' and 'Help Page' are available from the Control Panel. There are also process links in the Control Panel. These links become active when a certain condition is satisfied. For example, when all sections are completed and have green dots, the 'Validate' link becomes active. Clicking on it begins the structure-validation process. Similarly, after validation is finished, the 'Report' link appears. Depositors follow that link to view all validation reports. During validation all deposition sections are locked, preventing any changes while validation proceeds. However, after validation the Control Panel is unlocked and the user may make modifications to any section as desired.

When beginning a new deposition, the user must complete the first three sections in order. These are 'Contact Information', 'Instructions to the PDB' and 'File Deposition'. These sections provide the PDB with information needed to assist users if they have any problem during the submission process, capture release information and any special instructions and transfer all files to the *AutoDep* server. After completing these three sections the user may proceed in any order; however, *AutoDep* automatically moves from one page to the next suggesting a logical order to the process.

Every page displays its section name, section header, deposition ID, lineage and warning information beneath the Control Panel. Each of the 15 data sections in *AutoDep* has a unique section name. The section header summarizes the type of information requested in that section. Deposition ID is the unique identifier for the submission. Lineage information indicates that the deposition was initialized from an existing PDB entry and gives the original entry's PDB ID code. Warning information may appear in red to alert the user to verify certain information. For example, if a user has previously deposited to the PDB, they may enter just their e-mail address and *AutoDep* will automatically fill out the remaining fields of the 'Contact Information' page from the PDB's user database. The user is asked to verify and update the inserted information.

Each section is composed of one or more subsections with its own title and header. Two buttons appear at the beginning and at the end of each subsection. These are the 'Save Answers' and 'global n/a' buttons. The 'Save Answers' button is used to transmit information from the client-side browser to the *AutoDep* server. The 'global n/a' button is a shortcut that inserts 'n/a', for non-applicable or non-available, into every empty text box in the section and sets untouched radio buttons and checkboxes to their default value. If a subsection captures looped information, such as multiple helices, it will have buttons to allow the user to control the number of loops.

At the bottom of every *AutoDep* page are help-desk and external resource links that may assist the user in completing that page.

3.2.2. *AutoDep* data format. *AutoDep* uses a dictionary-driven mechanism to dynamically generate the user interface. Specifications for the web pages are defined in the *AutoDep* CIF dictionaries. The *AutoDep* CIF is a self-defining data-archiving format, simple and straightforward to edit, making dictionary development and maintenance easy. An *AutoDep* web page may be changed merely by adding, deleting or modifying a dictionary item with no further programming needed. Data captured by *AutoDep* is also archived in this CIF-like format.

Although *AutoDep* CIF format is easy for humans to read owing to its liberal use of spacing, indentation and comments, and for programs to parse, it is not accessible to standard UNIX tools and web-scripting languages such as Perl (see http://www.perl.org; Schwartz & Wall, 1992), that deal much more easily with input formatted in single lines. To solve this problem, *AutoDep* uses Zinc format, which is not an interchange format as CIF is, but rather a piping format. Zinc represents CIF information as lines consisting of five tab-separated fields: block, name, index, value and loop-id. The first field is the name of the CIF data block and is repeated on each line where appropriate. The second field is the name of the data item. The third field is an index specifier which is empty for non-looped data and is a zero-based index for looped data. The fourth field is the data item itself and the fifth field is a loop identifier. The complete contents of the CIF files are maintained but are reorganized in such a way as to be easily manipulated. The Zinc format is therefore isomorphic to CIF but more amenable to be used to pipe CIF data to programs (for more information about Zinc, see Stampf, 1994; ftp://ftp.rcsb.org/pub/pdb/doc/newsletters/bnl/newsletter94oct/newslttr.txt).

When a script needs to read information from a CIF file, the file is first converted to a Zinc file by the general tool *CIF2Zinc* (Stampf, 1995; ftp://ftp.rcsb.org/pub/pdb/doc/news-

letters/bnl/newsletter95jan/newslttr.txt) and the Zinc file is read by the script. The programs easily convert this data into keyword–value pairs which are used by CGI for data transfer. Perl scripts on the server side parse the dictionaries which have been converted to Zinc format and dynamically generate each web page, or form, upon the user's request. When the depositor returns the filled-out form to the server, the data are transferred as CGI query strings in the form of keyword–value pairs. These keyword–value pairs are first written to a CIF file which is then automatically converted to its corresponding Zinc file for programs to read. These two files are used for state storage of the deposition, to populate new web forms and to generate the PDB entry.

The use of CIF and Zinc format in *AutoDep* is shown in Fig. 4. It can be seen that information from all sources, such as the dictionaries, PDB format files and data captured from the web, is first translated to keyword–value pairs. From these keyword–value pairs *AutoDep* generates web pages and archive data files. The bottom branch in Fig. 4 shows how files are merged into *AutoDep*. CIF dictionaries, which are converted to Zinc files, specify the file formats that may be merged and map values to the keyword–value pairs recognized by *AutoDep*. At the time a merge takes place, values are preferentially saved from the merged source. This procedure is also used by the initialization option 'Based on a released PDB entry', which locates the user-specified entry on the PDB FTP server, and the option 'Based on a previous submission', which copies the state storage file of the user-specified deposition. In both cases, the mapped information is inherited by the new deposition.

3.2.3. Dictionaries. *AutoDep* dictionaries describe the web pages. Each data item requires its prompt, example, display style, validation details, the help text and a method to assign values from other sources such as PDB-format files. Display style refers to details such as indentation level and the conditions which specify whether an item is displayed, dependent on experiment type, refinement method *etc*. Style also defines whether the token is looped or a single value. Dictionaries contain the rules followed in validating data items, the relationship between two or more data items, and the level of verification, such as whether the item is mandatory. Web pages may be modified and validation rules updated by simply editing the corresponding dictionaries. Details of the various types of *AutoDep* dictionaries are found in *Appendix B*.

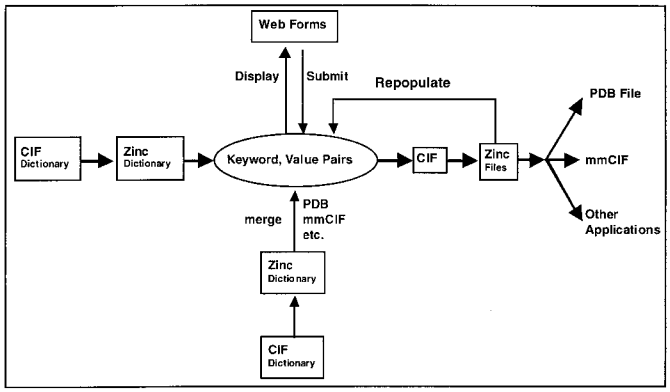


Figure 4
AutoDep's use of CIF and Zinc files.

Table 2
PDB's data validation checks.

Class	What is checked
Stereochemistry	Bond distances and angles, Ramachandran plot (dihedral angles), planarity of groups, chirality
Bonded/non-bonded interactions	Crystal packing, unspecified inter- and intra-residue links
Crystallographic information	Matthews coefficient, Z value, cell transformation matrices
Non-crystallographic transformation	Validity of non-crystallographic symmetry
Primary sequence data	Discrepancies with sequence databases
Secondary structure	Generated automatically or visually checked
Heterogen groups	Identification, geometry and nomenclature
Atomic coordinates	Syntax, nomenclature, missing atoms or residues, occupancies, thermal factors
Miscellaneous checks	Solvent molecules outside the hydration sphere, syntax checks, internal data consistency checks, R value and free R value

3.3. File upload

AutoDep allows users to transfer files using the web's CGI file upload protocol. For older browser versions that do not support CGI file upload, *AutoDep* uses an HTML form and FTP to accomplish file uploading. Here, we describe how to upload files with browsers that do support CGI file upload. File upload is as easy as 1, 2, 3 (see *Appendix A* for a step-by-step view of the process). The depositor first selects the type of file being uploaded, such as coordinate file, structure-factor file or NMR constraints file. They then enter the file's full pathname and click on the 'Upload' button to send the file to the *AutoDep* server. The status box lists the size and type of each transferred file. Data in the file that is in PDB format may be automatically merged into the deposition form by selecting the 'merge' option. Files can be deleted from the *AutoDep* server after uploading or they can be overwritten, which is useful when sending an updated coordinate file.

3.4. Validation and report system

3.4.1. Validation system. There are several layers of validation handled by *AutoDep*. During the active deposition session, on-line validation verifies that the user provides complete and correct data for every question. Results are written directly to the web page. Acceptable answers are given green check marks, while warnings and examples are provided to help the user correct any mistakes. Syntax, allowed value ranges and relationships between certain data items are checked at this level. These checks are specified in *AutoDep*'s verification dictionary. After on-line checking passes the entire deposition form, the structure is extensively checked by the PDB validation suite which includes in-house programs

and *WHAT_CHECK* (Table 2). This stage requires several minutes and the depositor receives an e-mail notification when validation is completed. The validation output is provided to the depositor, who can then take appropriate action. These diagnostic messages fall into three major categories, which the depositor must address before completing the submission and receiving a PDB ID code.

(i) Diagnostics requiring corrections and re-submission of coordinate data.

(ii) Diagnostics requiring annotations and/or comments to be provided by the depositors if the data are not corrected. A CAVEAT record will be added before release.

(iii) Diagnostics that may be indicative of unusual structures or possible problems.

For details of these messages see Table 3.

Tests valid only for diffraction experiments are not applied to entries reporting NMR experiments or model-building studies.

Heterogen groups are checked against the PDB Het Dictionary (see ftp://ftp.rcsb.org/pub/pdb/data/monomers/het_dictionary.txt) to see if the HET ID and the atom nomenclature used are consistent with the dictionary. Groups not in the dictionary and for which there is no conflict with the HET ID are accepted as is and are checked and standardized as part of the normalization process.

3.4.2. Report system. Following validation, *AutoDep* generates up to four different reports for the user. They are the Serious Error Report for violations which require correction and re-submission of data, the Caveat Report for items deemed serious that require annotation, the Summary Report which summarizes the results of the PDB validation checks and the *WHAT_CHECK* report. These reports must be reviewed before completing submission. The depositor has the option of modifying the model and submitting new coordinates for validation before completing submission.

Table 3
AutoDep's validation system.

Diagnostics requiring corrections and re-submission of coordinate data
More than one polypeptide or nucleotide chain assigned the same chain name
Heterogen group specified by HET and FORMUL records not present in the ATOM/HETATM records
More than 10% of the atoms involved in unusually close crystal packing interactions (this check also covers the case in which a non-standard space-group setting is used and the correct set of symmetry operators is not provided)
Violation of atom nomenclature for standard amino and nucleic acids
Duplicate ATOM or HETATM records in the same residue with the same atom name or the same coordinates
ATOM/HETATM records not formatted as described in the PDB Contents Guide
Heterogen ID provided in the coordinate file conflicts with the PDB Het Group Dictionary
Diagnostics requiring annotations or comments to be provided by the depositors if the data are not corrected. A CAVEAT record will be added before release
For polypeptides, φ–ψ angles for more than 20% of the residues outside the allowed region
Unexpected chirality at C^{α} center
Diagnostics that may be indicative of unusual structures or possible problems
R.m.s.d. of bond lengths greater than 0.08 Å from ideal values
R.m.s.d. of bond angles greater than 5.0°
Breaks in the chain (*e.g.* owing to disorder)
Differences between amino-acid sequences given by the ATOM records and those given in the appropriate sequence database entry
Amino-acid sequences not reported in any sequence database
CIS-peptides and peptide bonds that deviate significantly from the expected *trans* conformation
Individual bond lengths differing by more than 0.1 Å from standard values
Individual bond angles differing by more than 15° from standard values
Atoms too close to symmetry axes
Atoms involved in unusually close crystal packing interactions
Occupancy less than or equal to 0.0 or occupancy greater than 1.0
Atom occupancies less than 1.0 and for which no alternate location ATOM record is provided
Missing residues, missing atoms
Thermal factors greater than 100 $Å^2$
Unexpected deviations from planarity
Non-standard SCALE matrix
OXT atom record in the middle of a chain (flagged as extra atom) typically occurring before a gap in the coordinates
R value greater than 30%
Free *R* value greater than 35%
Free *R* and *R* value differing by more than 10%
R.m.s.d. between atoms related by an NCS MTRIX record greater than 3.0 Å

3.5. Release system

Depositors complete the submission process only after the structure data have passed validation. They must review the completed PDB entry and its companion report file and review and accept the PDB release policy. *AutoDep* returns an acknowledgment letter with the assigned four-letter PDB ID code by e-mail. If the depositor specifies that they want to release their data immediately, *AutoDep* places the data into the release stream. These data are referred to as 'author approved'. Following this, the PDB staff normalizes the data against other structure data in the database, producing the final version of the structural data that will be released as 'archive/author approved'. In this way, *AutoDep* maximizes the usefulness and timeliness of the structure data produced by research scientists independent of any work by the PDB staff, thereby enabling deposition centers the ability to keep up with an ever-increasing flow of data.

4. Discussion

AutoDep established a highly complex but flexible system for untrained user input of macromolecular structure information into a database. *AutoDep* changes the traditional procedure for data acceptance and validation of the primary databases into an interactive depositor-driven operation. The system leverages the knowledge and expertise of the experimenters, rather than relying on the database curators, for the complete and accurate description of the structural experiment and its results, and almost eliminates the delay between the acceptance of the data and its public release. *AutoDep* also greatly accelerates the experiment data-deposition process. Deposition of structure factors and other experimental data signifi-

Table 4
PDB structure-factor (SF) submissions, as of November 1998.

Year	Number of X-ray structure submissions	Number of SF submissions (%)
1994	804	205 (25.0)
1995	963	343 (36.0)
1996	1124	546 (49.0)
1997	1484	932 (62.8)
1998	1616	868 (53.7)
Total	5991	2894 (48.3)

cantly increased after the release of *AutoDep* in October 1996 (see Table 4). *AutoDep* is the first graphic user interface for macromolecular structure information deposition and validation. By taking advantage of the World Wide Web, it permits users around the world to use standard client tools, *i.e.* web browsers such as Netscape's *Communicator* or Microsoft's *Internet Explorer*, to access *AutoDep*.

AutoDep's set of mandatory items required for submission of a three-dimensional structure and the rules used for validation were adopted after the accumulation of many years experience at the Brookhaven PDB and with input from dozens of scientists in the field. This is the first explicit declaration of what constitutes a valid report to a macromolecular database. The rules themselves also set a *de facto* standard for structure checking and can be used by implementations other than *AutoDep* for automatic and objective structure verification.

Although *AutoDep* was originally developed as a tool for submission to the PDB, it also serves as a general tool for macromolecular structure validation. Users could download the *AutoDep* software suite onto their local machines and go through the entire validation procedure locally, producing a high-quality structure. High-throughput structure determination centers could automate the structure submission process. The *AutoDep* system is currently being used for deposition of macromolecular structures to the PDB at the European Molecular Biology Laboratory, European Bioinformatics Institute at Cambridge, UK (http://autodep.ebi.ac.uk/).

AutoDep is a powerful and effective first step for complex macromolecular structure data collection and validation. *AutoDep* was designed to be easily extensible. As *AutoDep* was used in production, frequently encountered user errors and other user-requested extensions were simply addressed and corrected/added by editing the corresponding dictionaries. Because *AutoDep* runs automatically with no staff involvement during the submission session, certain descriptive information is not fully validated until the archive normalization stage. For example, *AutoDep* compares the entered molecule and organism names to those in the SWISS-PROT Database (Bairoch & Boeckmann, 1994). If the name is not found in the database, *AutoDep* presents a warning to the user, asking them to verify it. Following this, even if the user ignores the warnings, *AutoDep* accepts the information as entered. Data collection should include the data existing in high-quality biological databases. It would be easy to implement a tool to run sequence comparisons with an external database such as SWISS-PROT and merge into *AutoDep* information related to the deposited structure, *i.e.* molecule name, source, sequence, function and macromolecule description. This would not only make deposition easier, but would also improve the quality of the data.

It would also be easy to improve the normalization of small-molecule structures and their nomenclature with the existing database *via* web-based tools.

Several more sophisticated features were planned, but with the relocation of the PDB from Brookhaven to the Research Collaboratory for Structural Bioinformatics (RCSB; http://www.rcsb.org/pdb/) this work was interrupted.

AutoDep's flexible and portable system has already been adopted by other primary databases and implemented on different platforms/operating systems. Sharing dictionaries and validation rules will make it easy to share data between databases in the future. The automatic dictionary-driven data deposition and validation process has a secondary important effect on the area of data mining and data collection: the standardization of every step of macromolecular structure information collection. *AutoDep* enforces standardization of the submitted information, facilitating *a posteriori* data interchange with other databases and data harvesting.

We wish to thank D. Xue, M. Libeson and J. McCarthy for their programming work on *AutoDep*, its associated programs and integration into the PDB processing flow and Dr Kim Henrick of the EMBL, European Bioinformatics Institute, Cambridge, UK, for collaboration in porting *AutoDep* to the EBI. We thank Dr Gert Vriend and Rob Hooft for aid in implementing *WHAT_CHECK* into *AutoDep*, Dr Manfred Hendlich for great assistance with heterogen groups and Dr S. Swaminathan for his great assistance throughout all aspects of *AutoDep*'s development. We also wish to thank Drs Maria Raves, Michal Harel, Clifford Felder, Harry Greenblatt and Gitay Kryger for their assistance in alpha testing of *AutoDep* and the many beta testers and users of the production system of *AutoDep* for their invaluable comments and suggestions. Brookhaven National Laboratory is operated by Brookhaven Science Associates under contract with the US Department of Energy. The Brookhaven PDB was supported by a combination of Federal Government Agency funds (work supported by the US National Science Foundation, the US Public Health Service, National Institutes of Health, National Center for Research Resources, National Institute of General Medical Sciences and National Library of Medicine and the US Department of Energy under contract DE-AC02-98CH10886) and user fees.

APPENDIX *A*

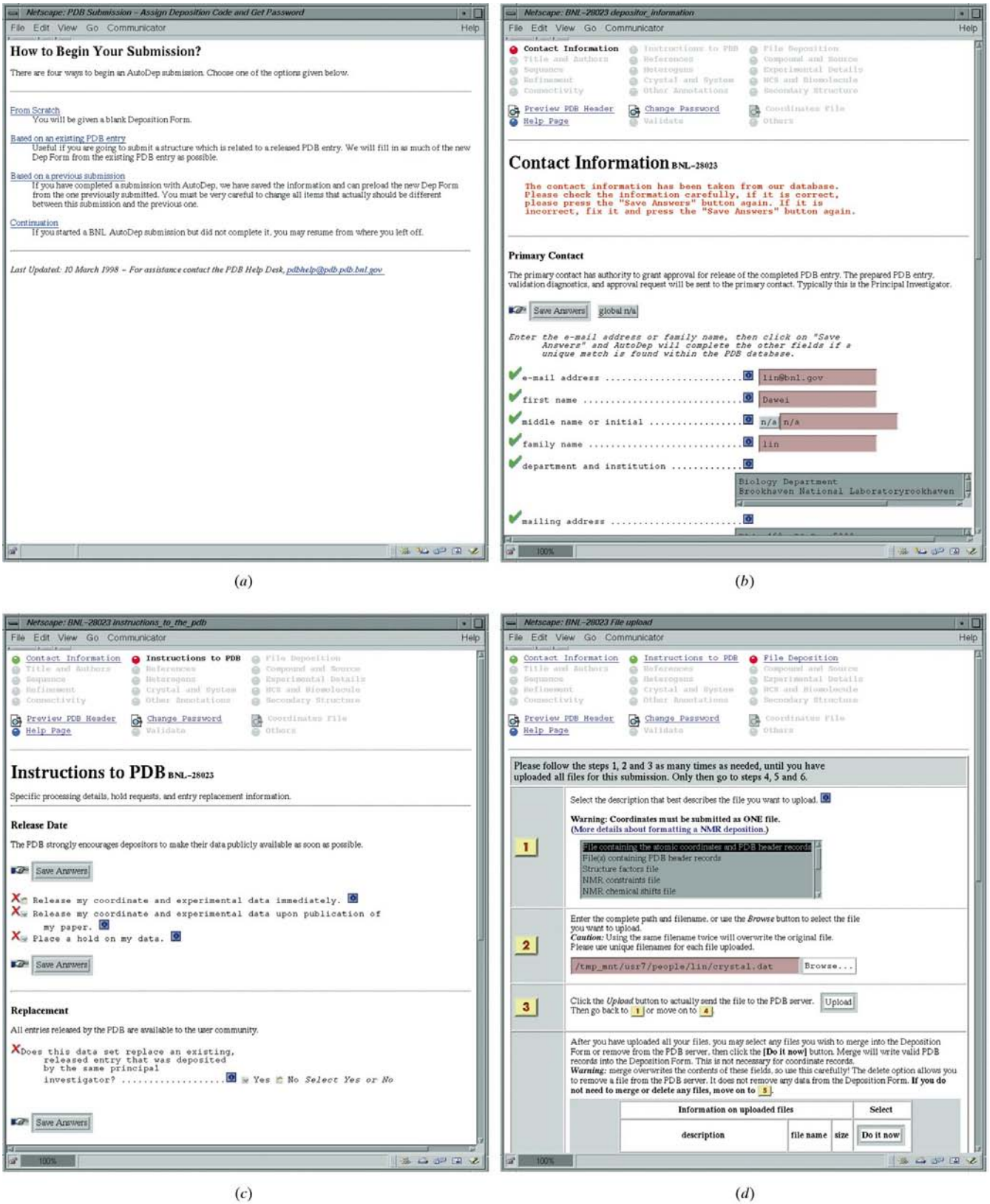


Figure 5
(*a*) Initialize deposition by one of the four options. (*b*) Enter user information. This can be populated from the user database. (*c*) Enter hold request, entry replacement information and special instructions. (*d*) Upload files according to stepwise instructions. After the upload, *AutoDep* activates all other sections.

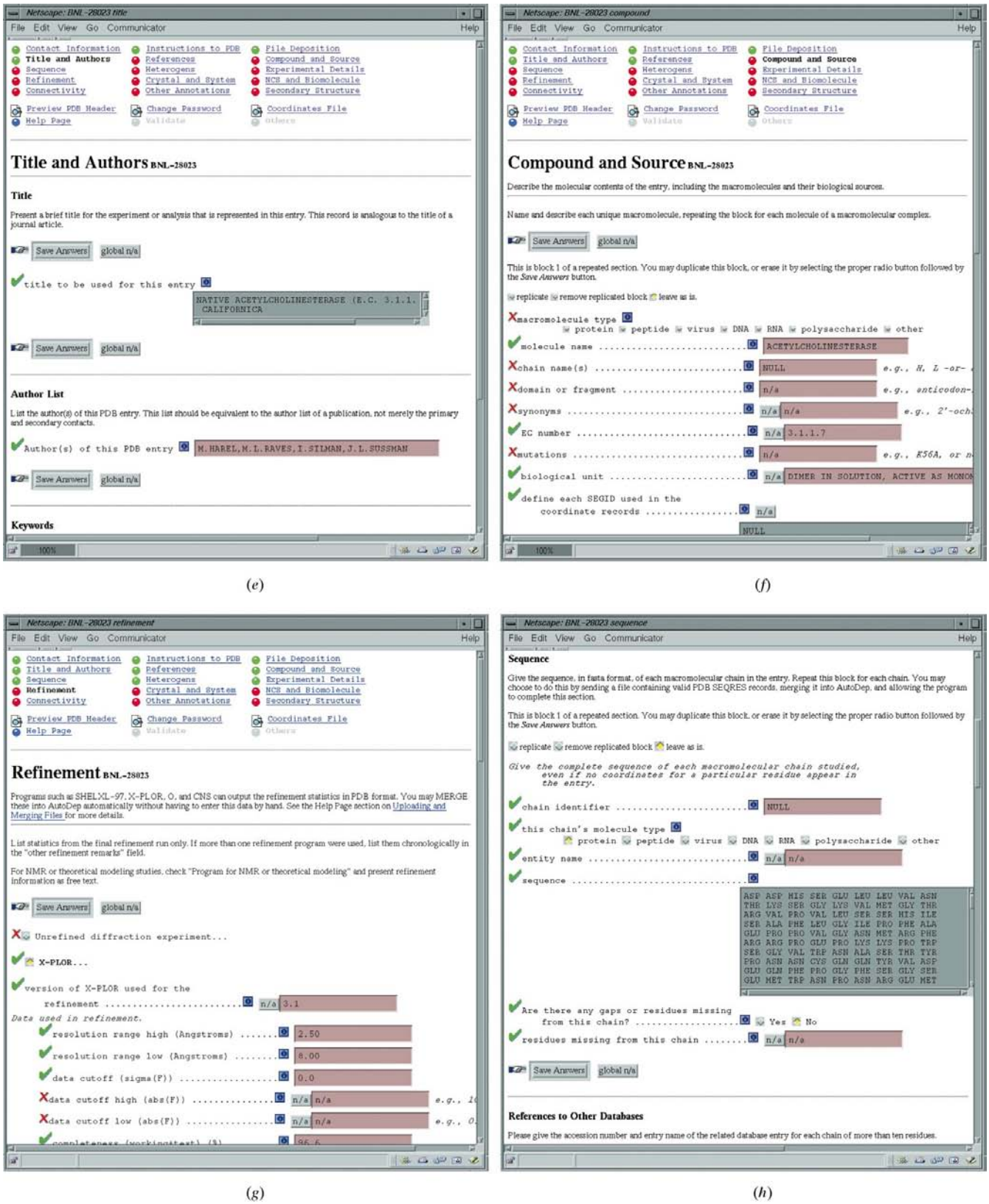
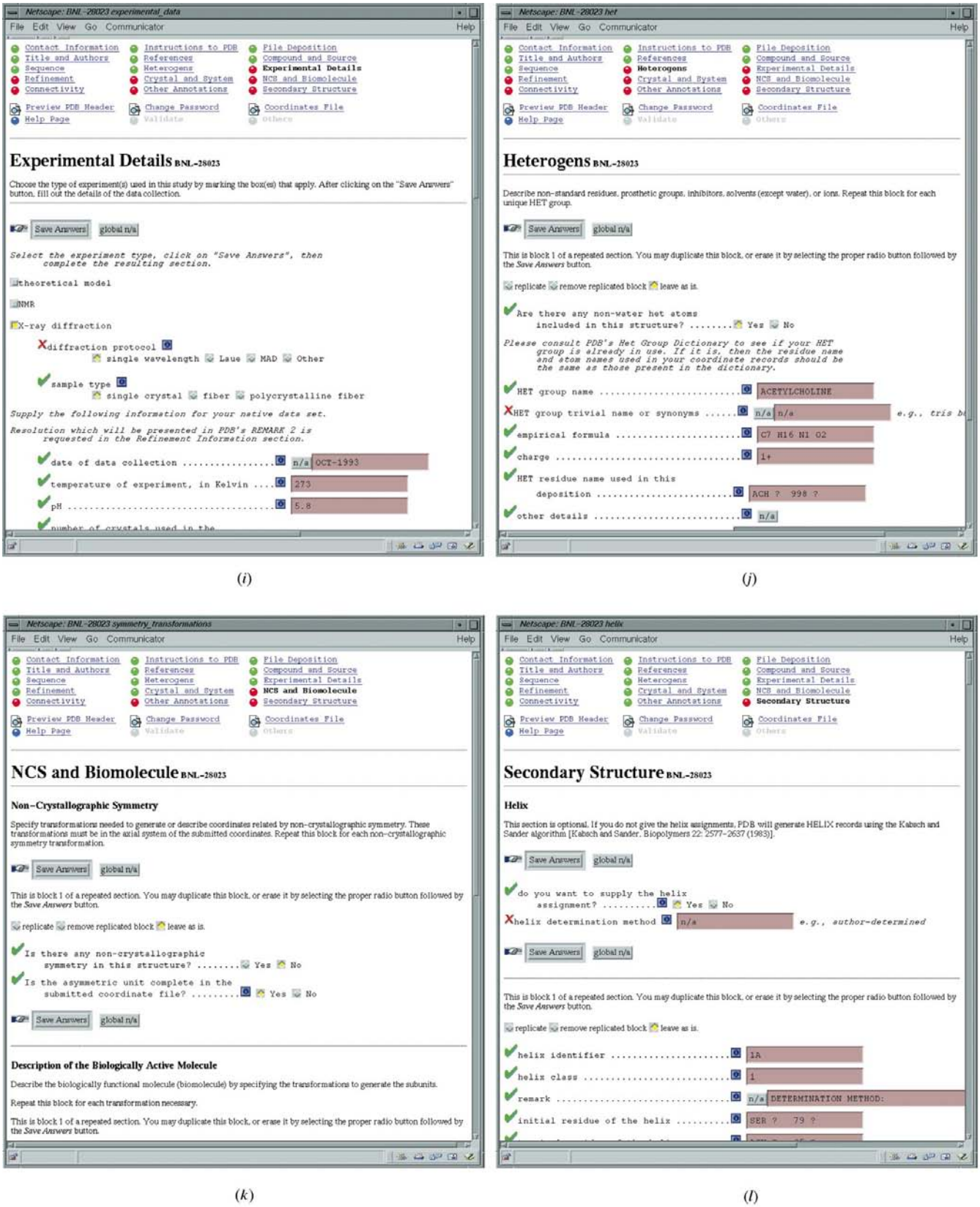


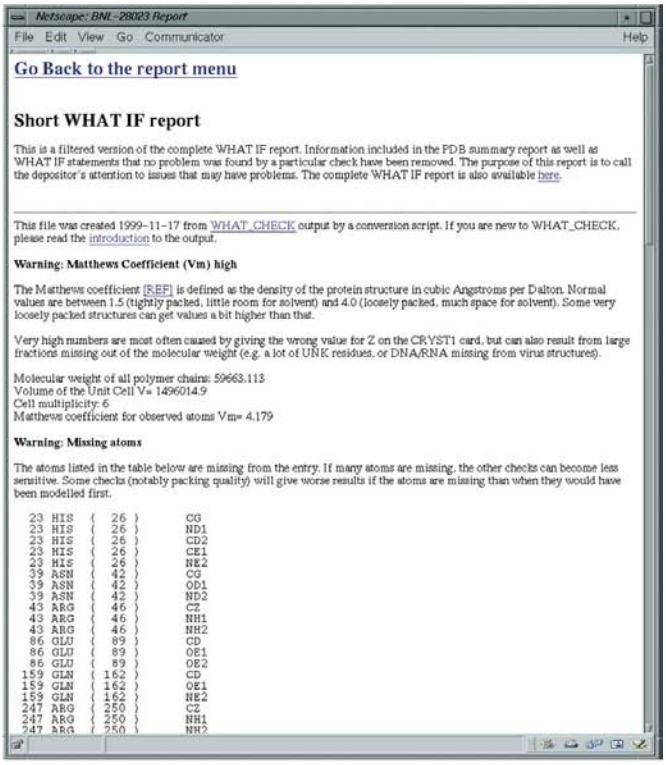
Figure 5 (continued)
(*e*) Enter the entry's title and authors. (*f*) Describe the molecular contents of the entry. (*g*) Enter or merge the refinement details. (*h*) Enter sequence and related database information. Three-letter codes are used in order to avoid non-standard name ambiguities.



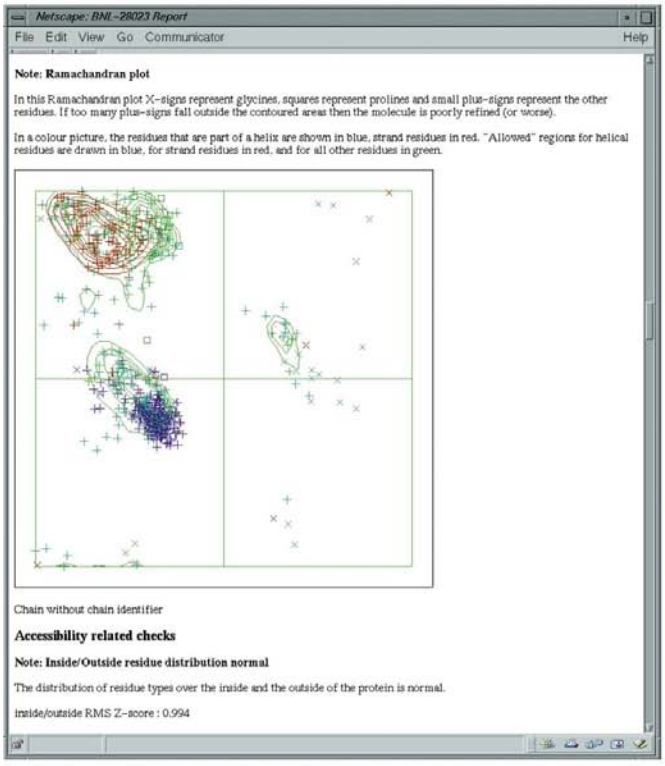
(*i*) (*j*) (*k*) (*l*)

Figure 5 (continued)
(*i*) Descibe the experiment and the details of the data collection. (*j*) Descibe any non-amino-acid residues, prosthetic groups, inhibitors, solvents (except water) or ions. (*k*) Enter non-crystallographic symmetry description. (*l*) Describe secondary structure.

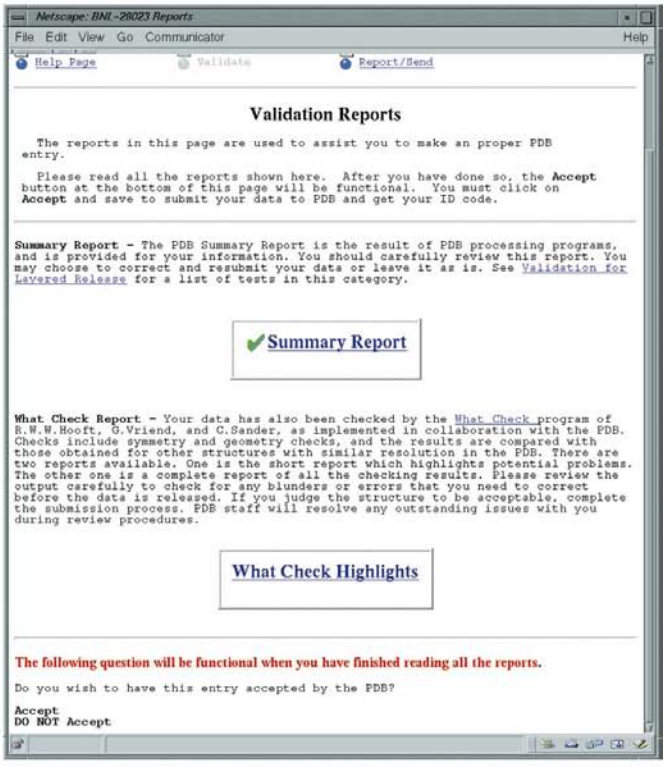
(*m*)



(*n*)



(*o*)



(*p*)

Figure 5 (continued)
(*m*) After completing all sections, the user can begin validation. (*n*) After validation, the user can inspect the summarized reports. Pictured is the automatically generated *filtered* version of the *WHAT IF* report. (*o*) The complete *WHAT IF* report is also available for more details. (*p*) After inspecting all reports, the user may submit the entry to the PDB.

APPENDIX *B*

```
data_sections
        loop_
              _name           # link to other tables
              _title          # text to appear at the top
              _mandatory      # true or false
              _header         # text to begin the section
              _status         # new/valid/dirty
depositor_information
"Contact Information"
true
""
new
```

(*a*)

```
data_useful_urls
        loop_
              _name           # link to the other table
              _text           # lead in
              _url            # the link


depositor_information
"The 3DB Browser"
"http://www.pdb.bnl.gov/cgi-bin/pdbmain"


depositor_information
"The PDB WWW Browser"
"http://www.pdb.bnl.gov/cgi-bin/browse"
```

(*b*)

```
data_subsections
     loop_
        _name               # name of the subsection (used as a link to other
                            # CIF loops)
        _section_name       # name of the section in which the subsection is
                            # contained, this links to the sections loop
        _header             # text to appear at the top of the subsection
        _is_loop            # true/false to indicate if the depositor is
                            # permitted to duplicate the subsection

primary_contact
depositor_information
;
Primary Contact
The primary contact has authority to grant approval for release of the
completed PDB entry. The prepared PDB entry, validation diagnostics, and
approval request will be sent to the primary contact. Typically this is the
Principal Investigator.

;
false
```

(*c*)

```
data_crystal
   loop_
      _name               # CIF name given to the data item. The value provided by the
                          # depositor is placed in a CIF-formatted file for later
                          # processing.
      _level              # integer indicating the indentation level. 1 is left
                          # justified.

      _type               # questions and answers are presented in many ways:
                              # array-3x1 - the expected answer is a 3 by 1 vector
                              # array-3x3 - the expected answer is a 3 by 3 matrix
                              # Boolean - the answer is yes or no
                              # checkbox - similar to Boolean, but allows for a
                              # different
                              # presentation to the user
                              # enum - a list of values is presented
                              # label - no answer expected, text written to the web
                              # page
                              # longtext - a longer text answer is expected
                              # radio-<text> - set of values, one of which may be
                              # selected
                              # text - a shorter text answer is expected
      _prompt             # prompt that poses the question to the depositor
      _verification       # short segment of Perl code or function name
      _example            # string with clue as to the type of answer required
      _mandatory          # 0 for non-mandatory, or 1 for mandatory
      _help               # longer paragraph to provide detailed help
      _display            # This is a Perl function that decides whether or not the
                          # question should be posed. For example, if an entry is based
                          # upon an NMR experiment, it is not necessary to ask dozens
                          # of x-ray related questions, so this allows them to be
                          # skipped.
      _xlate_from_pdb     # Perl code (or call to a Perl function) that instructs
                          # AutoDep how to insert a value that comes from a PDB file.
      _xlate_from_msd     # Perl code (or call to a Perl function) that instructs
                          # AutoDep how to insert a value that comes from a database
                          # entry.
Crystal_a
1
Text
"a (Angstroms)"
Verify_crystal_a
"41.67"
" (! &crystal_sys_mand())? 0:1"
"Cell length of side a of the unit cell."
1
'(! &crystal_sys_mand()) ? "n/a": $pdbValue{pdb_unit_cell_a}'
"$msdValue{_a}"
```

(*d*)

```
data_pdb_records
        loop_
              _pdb_record                 # first 6 columns
              _pdb_record_type            # single, multiple, etc.
              _pdb_record_cont            # continuation field begins here
              _pdb_record_overview        # text to begin the section
              _pdb_record_details         # minutia
              _pdb_record_vvv             # verification, validation,
                                          # value control
              _pdb_record_relationship    # relationship to other records
              _pdb_record_dep
              _pdb_record_example
              _pdb_record_bugs
HEADER single "" "" "" "" "" "" "" "" ""
TITLE continued 8 "" "" "" "" "" "" ""
COMPND keyword1 8 "" "" "" "" "" "" ""
```

(*e*)

```
data_pdb_items
        loop_
              _pdb_record             # link to data_pdb_records
              _pdb_item_name          # name link to the format description
              _pdb_item_start         # the column it begins in
              _pdb_item_width         # field width
              _pdb_loop               # is this a looped item?
              _pdb_keyword            # keyword prompt
              _pdb_item_definition    # definition from the format
                                      # description
HEADER pdb_functional_classification 11 40 no "" ""
HEADER pdb_deposition_date 51 9 no "" ""
HEADER pdb_idcode 63 4 no "" ""
TITLE pdb_title 11 -1 no "" ""
```

(*f*)

Figure 6
(*a*) Section dictionary, data_sections. *AutoDep* is divided into 15 sections, with each section presented as a separate web page or pages. Information about each section is contained in an *AutoDep* CIF section dictionary which has two looped data blocks. The first looped data block is 'data_sections' which defines each section. (*b*) Section dictionary, data_useful_urls. The second looped data block of the section dictionary contains uniform resource locators (URLs) that might assist the depositor in completing a submission. (*c*) Subsection dictionary, data_subsections. Each section is divided into one or more subsections which are defined in this looped data block. Subsections may be repeated if the data being submitted require it. For example, if there are two different molecular species in one deposition, each will be described in a separate subsection. (*d*) Question dictionary. Each subsection consists of a sequence of questions, or data items. Each of these is described by 11 fields in the dictionary. Dictionary items may contain text or regular expressions that are parsed and evaluated by the Perl program. It is possible to specify very complicated conditions in this manner. For example, as NMR experiments do not use crystals, the display condition may be '`(! &is_crystal_sys()) ? "n/a": $pdbValue{pdb_unit_cell_a}`' which stands for "If the condition of not having a crystal system is true, the value 'n/a' is automatically inserted for length of side *a* in the unit cell". Regular expressions are used throughout *AutoDep* to specify validation and merging rules. Each question in *AutoDep* has its own verification routine contained in the verification dictionary. Each routine returns a passed status, an error message or a warning message. The name of the routine appears in the question dictionary. These routines may be as simple as ensuring that the entered value is in the form of a real number or date string or as complex as checking that unit-cell parameters are consistent with the transformation matrix. (*e*) PDB Record type dictionary, data_pdb_records. *AutoDep* uses this dictionary to populate fields of a new deposition form by using data in a PDB file. This dictionary consists of two looped data blocks. The first and simplest loop is data_pdb_records which stores the PDB format description. (*f*) PDB Record items specification dictionary, data_pdb_items. The second loop of the PDB Record type dictionary is data_pdb_items which assigns a CIF identifier to each item stored in the PDB file and details how each item is extracted from the file.

References

Abola, E. E., Bernstein, F. C., Bryant, S. H., Koetzle, T. F. & Weng, J. (1987). *Crystallographic Databases – Information Content, Software Systems, Scientific Applications*, edited by F. H. Allen, G. Bergerhoff & R. Sievers, pp. 107–132. Bonn: Data Commission of the International Union of Crystallography.

Abola, E. E., Bernstein, F. C., Callaway, J., Cummings, M., Deroski, B., Esposito, P., Forman, A., Langdon, P., Libeson, M., Manning, N. O., McCarthy, J., Shea, R., Sikora, J., Stampf, D., Xue, D. & Sussman, J. L. (1996). *The Protein Data Bank Contents Guide: Atomic Coordinate Entry Format Description*. ftp://ftp.rcsb.org/pub/pdb/doc/format_descriptions/Contents_Guide_21.html.

Abola, E. E. & Manning, N. O. (1997). *Protein Data Bank Quart. Newslett.* **82**, 2.

Abola, E. E., Sussman, J. L., Prilusky, J. & Manning, N. O. (1997). *Methods Enzymol.* **277**, 556–571.

Bairoch, A. & Boeckmann, B. (1994). *Nucleic Acids Res.* **22**, 3578–3580.

Bernstein, F. C., Koetzle, T. F., Williams, G. J. B., Meyer, E. F. Jr, Brice, M. D., Rodgers, J. R., Kennard, O., Shimanouchi, T. & Tasumi, M. (1977). *J. Mol. Biol.* **112**, 535–542.

Brünger, A. T. (1992). *X-PLOR Version* 3.1. *A System for Crystallography and NMR*. Yale University, New Haven, USA.

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

Brünger, A. T., Adams, P. D. & Rice, L. M. (1997). *Structure*, **5**, 325–336.

Hall, S. R., Allen, F. H. & Brown, I. D. (1991). *Acta Cryst.* A**47**, 655–685.

Hooft, R. W. W., Vriend, G., Sander, C. & Abola, E. E. (1996). *Nature (London)*, **381**, 272.

Jones, T. A., Zou, J.-Y., Cowan, S. W. & Kjeldgaard, M. (1991). *Acta Cryst.* A**47**, 110–119.

Manning, N. O. (1996). *Protein Data Bank Quart. Newslett.* **77**, 2.

Manning, N. O. (1996). *Protein Data Bank Quart. Newslett.* **78**, 2.

Murshudov, G. N., Vagin, A. A., Lebedev, A., Wilson, K. S. & Dodson, E. J. (1999). *Acta Cryst.* D**55**, 247–255.

Protein Data Bank (1971). Crystallography, Protein Data Bank [announcement]. *Nature New Biol.* **233**, 223.

Schwartz, R. L. & Wall, L. (1992). *Programming Perl.* Sebastopolo: O'Reilly & Associates, Inc.

Seavey, B. R., Farr, E. A., Westler, W. M. & Markley, J. L. (1991). *J. Biomol. NMR*, **1**, 217–236.

Sheldrick, G. M. (1997). *Proceedings of the CCP4 Study Weekend. Recent Advances in Phasing*, edited by K. S. Wilson, G. Davies, A. Ashton & S. Bailey, pp. 147–157. Warrington: Daresbury Laboratory.

Stampf, D. (1994). *Protein Data Bank Quart. Newslett.* **70**, 1.

Stampf, D. (1995). *Protein Data Bank Quart. Newslett.* **71**, 7.

Sussman, J. L. (1997). *Protein Data Bank Quart. Newslett.* **79**, 1.

Sussman, J. L., Lin, D., Jiang, J., Manning, N. O., Prilusky, J., Ritter, O. & Abola, E. E. (1998). *Acta Cryst.* D**54**, 1078–1084.

Tronrud, D. E., Ten Eyck, L. F. & Matthews, B. W. (1987). *Acta Cryst.* A**43**, 489–501.

Vriend, G. (1990). *J. Mol. Graph. Model.* **8**, 52–56.