J. Appl. Cryst. (1998). **31**, 282–295

CIF Applications. VIII. *pdb2cif*: translating PDB entries into mmCIF format†

HERBERT J. BERNSTEIN,[a]* FRANCES C. BERNSTEIN[b] AND PHILIP E. BOURNE[c,d] *at* [a]*Bernstein + Sons, 5 Brewster Lane, Bellport, NY 11713-2803, USA,* [b]*Biology Department, Brookhaven National Laboratory, Upton, NY 11973-5000, USA,* [c]*San Diego Supercomputer Center, PO Box 85608, San Diego, CA 92186-9784, USA, and* [d]*Department of Pharmacology, University of California, San Diego, CA 92093-0365, USA. E-mail: yaya@bernstein-plus-sons.com*

(*Received 8 May 1997; accepted 14 May 1997*)

Abstract

pdb2cif is a new version of an awk script originally written by P. E. Bourne in 1993 to translate from the 1992 Protein Data Bank (PDB) format to the then-emerging macromolecular Crystallographic Information File (mmCIF) definition. This new version of *pdb2cif* translates from all current PDB formats, including the 1992 PDB format and the 1996 PDB Atomic Coordinate Entry Format, Version 2.0, to the 1997 mmCIF format as defined in the mmCIF dictionary 1.0.00. The program is provided as an m4 script from which both perl and awk versions can be produced. The program identifies mmCIF entities implicitly by sequence homology among PDB SEQRES records. With minor additions to the dictionary, the resultant mmCIF data-sets are substantially compliant with the mmCIF 1.0.00 dictionary.

1. Introduction

The program *pdb2cif* reads entries in Protein Data Bank (PDB) format (Bernstein *et al.*, 1977) or PDB Atomic Coordinate Entry format (Protein Data Bank, 1996) and converts them to macromolecular Crystallographic Information File (mmCIF) format (Fitzgerald *et al.*, 1996; Bourne *et al.*, 1997). All valid PDB record types are converted, but most PDB REMARK records are carried forward as text, rather than being parsed any further. The resulting entries are substantially compliant with dictionary definition language version 2 (DDL2) (Berman & Westbrook, 1994; Westbrook & Hall, 1995) and mmCIF rules with the addition of a small number of new token definitions.

The Protein Data Bank format has been used for over 20 years to archive macromolecular data, is produced by many refinement programs and is used as an input format by many applications. The adoption of the mmCIF dictionary (Fitzgerald *et al.*, 1996) by the IUCr, in response to the need to represent explicitly a larger amount of data that can be parsed by computer (necessary as the number of structures continues to grow exponentially), has made translation from PDB format to mmCIF format a pressing issue.

In this paper we review the techniques used in *pdb2cif* to move from structures represented in PDB format to mmCIF format. Some data items have direct mapping with minor syntactic adjustment, such as for author names and journal references. Other data items, however, require us to recast our thinking along new lines. For example, the PDB format works with chains and heterogen groups, while mmCIF uses entities (discrete chemical components). Proper identification of entities in a PDB entry may require looking for sequence homology. As another example, consider beta sheets. The PDB format treats a bifurcated sheet as two distinct sheets that happen to have certain strands in common, while mmCIF allows all the strands involved to be represented as a single sheet. This requires strand matching and alignment to go from PDB format to mmCIF. What has currently been automated in *pdb2cif* and what still requires human intervention is discussed.

2. Outline of the PDB format

The Protein Data Bank describes a macromolecular structure using a format containing records with fixed fields that are order dependent. In this context, a record is a line of text. The first six characters of each record contain a left-justified string of from three to six upper-case characters that specifies a particular PDB 'record type'. The record type implies the layout of the information in that record. For some record types fields of information may span multiple records of the same type. For many such record types, continuation is indicated by an integer in columns 9–10. In most cases, the meaning of the information found in specific columns of a record type is fixed and is specified in external documents rather than within the PDB entry itself. In most cases, all records of a given record type are grouped together. The order of presentation of different record types within an entry is fixed. In all cases the records are no more than 80 characters long and, in entries conforming to the PDB 1992 or earlier formats, there is no structural information past column 72. There are several variations of the PDB format which have been used since its adoption. Table 1 is a composite of the 1992 and 1996 versions, since data-sets in both formats are still in use.

In the coordinate section, ATOM records are used for 'standard' residues and HETATM records are used for atoms in heterogens. For any given atom in a given conformation, the ordering of records is [ATOM|HETATM][SIGATM] [ANISOU][SIGUIJ] as a single group of records. Groups of records specifying the alternate conformations for the same atom follow immediately. These groups of records are then organized in an ordering determined by templates for standard residues or heterogens. Records in the coordinate section associated with a particular model in an NMR entry with multiple models are delimited by a MODEL/ENDMDL pair. Chains are terminated by TER records. Except within the coordinate section, all records for any given record type are grouped together.

† This paper is one of a series on CIF applications. Offprints are available from The Managing Editor, International Union of Crystallography, 5 Abbey Square, Chester CH1 2HU, England. See text of paper for availability of program(s) by e-mail.

Table 1. *Outline of the PDB data format*

An entry comprises sections (**bold**) containing the record types (*italic* for record types in both 1992 and 1996 formats, plain text for record types introduced in 1996) listed in the following order. Optional record types are contained in square brackets. It should be pointed out that the records REMARK 3-999 have more internal structure in the 1996 format than in previous formats.

Title section			
HEADER	[*OBSLTE*]	TITLE	[CAVEAT]
COMPND	*SOURCE*	KEYWDS	[*EXPDTA*]
AUTHOR	*REVDAT*	[*SPRSDE*]	[*JRNL*]
REMARK 1	*REMARK* 2	*REMARK* 3	[*REMARK* 4-999]
Primary structure section			
[MODRES]	DBREF	[SEQADV]	[*SEQRES*]
Footnote section (1992 format only)			
FTNOTE			
Heterogen section			
[*HET*]	[HETNAM]	[HETSYN]	[*FORMUL*]
Secondary structure section			
[*HELIX*]	[*SHEET*]	[*TURN*]	
Connectivity annotation section			
[*SSBOND*]	[LINK]	[HYDBND]	[SLTBRG]
[CISPEP]			
Miscellaneous features section			
[*SITE*]			
Crystallographic and coordinate transformation section			
CRYST1	*ORIGXn*	*SCALEn*	[*MTRIXn*]
[*TVECT*]			
Coordinate section			
[*MODEL*]	[*ATOM*]	[*SIGATM*]	[*ANISOU*]
[*SIGUIJ*]	[*TER*]	[*HETATM*]	[*ENDMDL*]
Connectivity section			
[*CONECT*]			
Bookkeeping section			
MASTER	*END*		

Fig. 1 shows part of the coordinate section from the PDB entry 4INS (pig insulin) (Baker *et al.*, 1988) in the format in use in 1989 and how the same information would be presented in the 1996 format. Note that columns 78–79 now contain the right-justified element symbol in the 1996 format.

3. Outline of mmCIF

The new mmCIF format is one of a family of STAR (Self-defining Text Archive and Retrieval) file formats which uses a tag-value style of presentation and has very little sensitivity to the ordering of the information (Hall, 1991; Hall & Spadaccini, 1994). Since no fixed positions for fields are defined in mmCIF, the format of mmCIF data-sets (Fitzgerald *et al.*, 1996; Bourne *et al.*, 1997) is much less rigidly defined than is the case for fixed-field formats such as the PDB format. Information is presented either in tag-value pairs or in column-headed tabular form. Tags are distinguished from values by an initial underscore. Information is constrained to 80-column lines, but spacing between fields is arbitrary. In mmCIF, tags are organized into category groups and categories. Individual tag-value pairs from different categories may be placed anywhere within a data-set, but it is considered good practice to group the tag-value pairs from a given category together. When the STAR construct '`loop_`' is used to introduce a table, all the data items within that table must have tags from the same category and all the data items for that category for which any information is being presented should be placed in the same table. The category is a name for the table. The category groups and associated categories defined in the mmCIF dictionary are given in Table 2.

Each category contains multiple tags. The name of each tag begins with its category followed by a period. In STAR, a table of information is created by the special token '`loop_`' followed by the tags that head the columns of the table, followed by the rows of values. If a table is given at all, certain tags are mandatory and certain values cannot be missing. Each row of a table must have a unique key, consisting of the values of certain designated columns within that row. In addition, some information is mandatory in mmCIF data-sets to ensure a complete coherent presentation of information

1989 format:

```
ATOM      1  N   GLY A   1      -8.863  16.944  14.289  1.00 21.88   1  4INS 235
ATOM      2  CA  GLY A   1      -9.929  17.026  13.244  1.00 22.85   1  4INS 236
ATOM      3  C   GLY A   1     -10.051  15.625  12.618  1.00 43.92   1  4INS 237
ATOM      4  O   GLY A   1      -9.782  14.728  13.407  1.00 25.22   1  4INS 238
ATOM      5  N   ILE A   2     -10.333  15.531  11.332  1.00 26.28   1  4INS 239
ATOM      6  CA  ILE A   2     -10.488  14.266  10.600  1.00 20.84   1  4INS 240
ATOM      7  C   ILE A   2      -9.367  13.302  10.658  1.00 11.81   1  4INS 241
ATOM      8  O   ILE A   2      -9.580  12.092  10.969  1.00 20.31   1  4INS 242
ATOM      9  CB  ILE A   2     -10.883  14.493   9.095  1.00 40.00   1  4INS 243
ATOM     10  CG1 ILE A   2     -11.579  13.146   8.697  1.00 36.74   1  4INS 244
```

1996 format:

```
ATOM      1  N   GLY A   1      -8.863  16.944  14.289  1.00 21.88   1       N
ATOM      2  CA  GLY A   1      -9.929  17.026  13.244  1.00 22.85   1       C
ATOM      3  C   GLY A   1     -10.051  15.625  12.618  1.00 43.92   1       C
ATOM      4  O   GLY A   1      -9.782  14.728  13.407  1.00 25.22   1       O
ATOM      5  N   ILE A   2     -10.333  15.531  11.332  1.00 26.28   1       N
ATOM      6  CA  ILE A   2     -10.488  14.266  10.600  1.00 20.84   1       C
ATOM      7  C   ILE A   2      -9.367  13.302  10.658  1.00 11.81   1       C
ATOM      8  O   ILE A   2      -9.580  12.092  10.969  1.00 20.31   1       O
ATOM      9  CB  ILE A   2     -10.883  14.493   9.095  1.00 40.00   1       C
ATOM     10  CG1 ILE A   2     -11.579  13.146   8.697  1.00 36.74   1       C
```

Fig. 1. Example of the coordinate section from PDB entry 4INS in the format in use in 1989 and in the 1996 format.

Table 2. *Category groups (**bold**) and categories in the mmCIF dictionary (version 1.0.00)*

atom_group (properties of atoms)			
atom_site	atom_site_anisotrop	atom_sites	atom_sites_alt
atom_sites_alt_ens	atom_sites_alt_gen	atom_sites_footnote	atom_type
audit_group (dictionary maintenance and identification)			
audit	audit_author	audit_conform	audit_contract_author
cell_group (unit cell)			
cell	cell_measurement	cell_measurement_refln	
chemical_group (chemical properties and nomenclature)			
chemical	chemical_conn_atom	chemical_conn_bond	chemical_formula
chem_comp_group (components of chemical structure)			
chem_comp	chem_comp_angle	chem_comp_atom	chem_comp_bond
chem_comp_chir	chem_comp_chir_atom	chem_comp_plane	chem_comp_plane_atom
chem_comp_tor	chem_comp_tor_value		
chem_link_group (linkages between components of chemical structure)			
chem_comp_link	chem_link	chem_link_angle	chem_link_bond
chem_link_chir	chem_link_chir_atom	chem_link_plane	chem_link_plane_atom
chem_link_tor	chem_link_tor_value	entity_link	
citation_group (bibliographic references)			
citation	citation_author	citation_editor	
compliance_group (categories included to comply with previous dictionaries)			
database			
computing_group (computational details of the experiment)			
computing	software		
database_group (references to other databases with related information)			
database_2	database_PDB_caveat	database_PDB_matrix	database_PDB_remark
database_PDB_rev	database_PDB_rev_record	database_PDB_tvect	
diffrn_group (details of the diffraction experiment)			
diffrn	diffrn_attenuator	diffrn_detector	diffrn_measurement
diffrn_orient_matrix	diffrn_orient_refln	diffrn_radiation	diffrn_radiation_wavelength
diffrn_refln	diffrn_reflns	diffrn_scale_group	diffrn_source
diffrn_standard_refln	diffrn_standards		
entity_group (chemical entities)			
entity	entity_keywords	entity_link	entity_name_com
entity_name_sys	entity_poly	entity_poly_seq	entity_src_gen
entity_src_nat			

about a macromolecular structure. Such additional mandatory tags in a table need not have distinct values row to row. For example, in the `atom_site` category, the key is the data item `_atom_site.id`, which uniquely identifies each row in the `atom_site` table. Values for `_atom_site.type_symbol` (*e.g.* C, N, O) are also mandatory, but, naturally, they are not unique in each row.

In mmCIF format, once the tag heading a column is given, values must be given for that column in every row. When the information to be given is not known, a question mark is used in place of the required value. When a value is otherwise intentionally not given, a period is used in place of the required value. In translating from PDB format to mmCIF, it is often necessary to recognize blank fields in PDB records and to find a value to use in the equivalent mmCIF table. With some exceptions noted below, a period is an appropriate equivalent to the PDB blank.

Fig. 2 gives an extract from an mmCIF conversion of PDB entry 4INS showing the beginning of the table giving the tags and values in the `atom_site` category. Because tags are always given, the same information can be presented in different orderings. Note that the mmCIF format does not depend on the columns shown in Fig. 2, just on a consistent ordering of tags *versus* data values. Also note that a period had to be given in each row as a place-holder for the unspecified values of `_atom_site.label_alt_id`. The period is a 'metacharacter' in mmCIF denoting an unspecified value. A question mark, which has the slightly different meaning of a missing value, could also have been used.

4. Relationship between mmCIF and PDB format

The relationship between mmCIF and PDB format is complex. There are differences both in syntax and in content. These differences are summarized in Table 3.

Handling the syntactic differences between PDB format and mmCIF format involves attention to detailed information relating various PDB fields to appropriate mmCIF tags and is a straightforward translation using specific rules. However, handling the differences in content requires much more from a translation program. Translation of PDB polypeptide and polynucleotide chains into mmCIF chemical entities is a case in point. While nonpolymeric heterogens are assigned an explicit 'component number' in PDB format, which is essentially equivalent to an mmCIF `_entity.id`, more analysis is needed when dealing with chains. In general, the most difficult issues arise from the concept of 'normalization' (see below). Other areas are less troublesome. PDB and mmCIF formats agree simply and directly for some data items, such as cell parameters, and permit a simple tabular mapping, as shown in Fig. 3, by an extract from the concordance which is available as part of the *pdb2cif* program release. Other important

Table 2. (*cont.*)

entry_group (the entire data block)			
entry	entry_link		
exptl_group (details of the experimental conditions)			
exptl	exptl_crystal	exptl_crystal_face	exptl_crystal_grow
exptl_crystal_grow_comp			
geom_group (internal coordinates)			
geom	geom_angle	geom_bond	geom_contact
geom_hbond	geom_torsion		
iucr_group (internal processing and manuscript submission by the IUCr staff)			
journal	journal_index	publ	publ_author
publ_body	publ_manuscript_incl		
pdb_group (pertaining to PDB file format or data processing codes, overlaps much of database_group)			
database_PDB_caveat	database_PDB_matrix	database_PDB_remark	database_PDB_rev
database_PDB_rev_record	database_PDB_tvect		
phasing_group (phasing)			
phasing	phasing_averaging	phasing_isomorphous	phasing_mad
phasing_MAD_clust	phasing_MAD_expt	phasing_MAD_ratio	phasing_MAD_set
phasing_mir	phasing_MIR_der	phasing_MIR_der_refln	phasing_MIR_der_shell
phasing_MIR_der_site	phasing_MIR_shell	phasing_set	phasing_set_refln
refine_group (describe refinement)			
refine	refine_analyze	refine_B_iso	refine_hist
refine_ls_restr	refine_ls_restr_ncs	refine_ls_shell	refine_occupany
refln_group (details of reflection measurements)			
refln	reflns	reflns_scale	reflns_shell
struct_group (crystallographic structure)			
struct	struct_asym	struct_biol	struct_biol_gen
struct_biol_keywords	struct_biol_view	struct_conf	struct_conf_type
struct_conn	struct_conn_type	struct_keywords	struct_mon_details
struct_mon_nucl	struct_mon_prot	struct_mon_prot_cis	struct_ncs_dom
struct_ncs_dom_lim	struct_ncs_ens	struct_ncs_ens_gen	struct_ncs_oper
struct_ref	struct_ref_seq	struct_ref_seq_dif	struct_sheet
struct_sheet_hbond	struct_sheet_order	struct_sheet_range	struct_sheet_topology
struct_site	struct_site_gen	struct_site_keywords	struct_site_view
symmetry_group (symmetry information)			
symmetry	symmetry_equiv		

macromolecular data descriptors, because of the very different views of the same data, require complex transformations.

For example, beta sheets are built from beta strands. In mmCIF, all the strands in all sheets are listed in one `struct_sheet_range` table. The relative ordering and orientation of all strands in all sheets are given in one `struct_sheet_order` table. The hydrogen bonding among all strands in all sheets is listed in one `struct_sheet_hbond`

```
loop_
_atom_site.label_seq_id
_atom_site.group_PDB
_atom_site.type_symbol
_atom_site.label_atom_id
_atom_site.label_comp_id
_atom_site.label_asym_id
_atom_site.auth_seq_id
_atom_site.label_alt_id
_atom_site.cartn_x
_atom_site.cartn_y
_atom_site.cartn_z
_atom_site.occupancy
_atom_site.B_iso_or_equiv
_atom_site.footnote_id
_atom_site.label_entity_id
_atom_site.id
1 ATOM  N  N    GLY A   1  .  -8.863  16.944  14.289  1.00 21.88   1   1      1
1 ATOM  C  CA   GLY A   1  .  -9.929  17.026  13.244  1.00 22.85   1   1      2
1 ATOM  C  C    GLY A   1  . -10.051  15.625  12.618  1.00 43.92   1   1      3
1 ATOM  O  O    GLY A   1  .  -9.782  14.728  13.407  1.00 25.22   1   1      4
2 ATOM  N  N    ILE A   2  . -10.333  15.531  11.332  1.00 26.28   1   1      5
2 ATOM  C  CA   ILE A   2  . -10.488  14.266  10.600  1.00 20.84   1   1      6
2 ATOM  C  C    ILE A   2  .  -9.367  13.302  10.658  1.00 11.81   1   1      7
2 ATOM  O  O    ILE A   2  .  -9.580  12.092  10.969  1.00 20.31   1   1      8
2 ATOM  C  CB   ILE A   2  . -10.883  14.493   9.095  1.00 40.00   1   1      9
2 ATOM  C  CG1  ILE A   2  . -11.579  13.146   8.697  1.00 36.74   1   1     10
```

Fig. 2. Beginning of the `atom_site` table for mmCIF conversion of PDB entry 4INS.

Table 3. *Major differences in syntax and content between PDB format and mmCIF format*

mmCIF	PDB
Syntax	
Tag-value definitions	Fixed fields
Little order dependence	Strong order dependence
Strict table structure	Some information nontabular
Upper/lower case	Upper case only
yyyy-mm-dd dates	dd-mmm-yy dates (dd-mmm-yyyy in some REMARKS)
Family-name-first author names	Family-name-last author names
Related items may have to appear in separate tables	
Content	
Extensive normalization	Less normalization
Structures defined using entities (unique chemical components)	Structures defined using chains and heterogen groups

table. The general characteristics of all sheets *per se* is given in one `struct_sheet` table. In PDB format, sheets are described by one set of sheet records for each simple, nonbifurcated sheet. To convert from PDB format to mmCIF format, a list of all strands must be extracted from the SHEET records, sorted to remove duplicates, and the information placed in a `struct_sheet_range` table. All strand-to-strand relationships are extracted and placed in a `struct_sheet_order` table, *etc.* A diagram of PDB entry 2ACE (native acetylcholinesterase) (Raves *et al.*, 1997) is given in Fig. 4 showing the strands forming sheets. Other secondary structure is not shown. A small sheet of three strands is on the top left and a larger sheet of 11 strands is on the right. Residue 16 is common to both sheets. The SHEET information from this PDB entry is given in Fig. 5(*a*). The same information converted to mmCIF format by *pdb2cif* is given in Figs. 5(*b*) and 5(*c*).

The scattering of information from a PDB SHEET record type into various tables is an example of 'normalization' (Codd, 1970, 1972). Normalization is a concept from the design of databases in which data are organized into the rows and columns of tables with a single data item in each table position, with unique keys to identify each row, and minimal repetition of the same information, so that it is easier to update, check and retrieve data reliably. Although not developed explicitly for such database considerations, mmCIF is database oriented. This causes information from PDB records such as SHEET or JRNL to be distributed across multiple mmCIF categories and information from separate PDB records to be gathered into common mmCIF tables. For example, the PDB-record-to-mmCIF-category mapping of the primary structure section used in *pdb2cif* is shown in Fig. 6.

One last issue that arises in conversion from PDB format to mmCIF is selection of an ordering of the information in an mmCIF data-set. There is no required ordering. One common practice is to order tag-value pairs and tables alphabetically, but this places the table of atomic coordinates in the `atom_site` category first, placing a large block of information before categories that identify the data-set. For readability, it is helpful to place information from the `_entry.id`, `_struct.title`, the contents of the `struct_keywords`, `audit_author`, `citation`, `citation_editor`, `citation_author`, `rens`, `database_PDB_remark`, `cell`, `symmetry`, `audit`, `entity_poly_seq`, `entity`, `struct_asym`, `chem_comp`, `database_PDB_matrix`, `atom_sites` and `atom_sites_footnote` categories before the `atom_site` table. We follow this practice in *pdb2cif*.

5. The program *pdb2cif*

pdb2cif converts PDB entries into mmCIF data-sets. (The term 'data-set' refers to the comments and mmCIF information presented as a single document describing some set of data. At present, each data-set produced by mmCIF contains one CIF `data_block`, even if multiple NMR models are described.) Most, but not all, common PDB record types are converted. The exceptions are the new structured PDB REMARK records introduced in April 1996 (Protein Data Bank, 1996), which, as of this writing, are still evolving. These REMARK records are preserved as text associated with the `database_PDB_remark.text` tag, rather than being parsed internally to provide values for tags in other categories. The program also cannot resolve some of the ambiguities involved in the analysis of the new keyword fields for the PDB COMPND and SOURCE records and treats those as text as well. The program has gone through extensive changes since 1993 as both mmCIF and the PDB format have evolved. The program, which was initially written as an awk script, is now available as an m4 (Kernighan & Ritchie, 1977) macro document that produces either perl or awk versions. The perl version is recommended.

The `pdb2cif.m4` macro document contains approximately 6500 lines of text, which generates a similarly sized awk script or over 10 000 lines of perl code (due to in-lining of certain critical functions). On modern processors with sufficient memory (32 to 64 Mbytes of available RAM), the conversion

```
PDB Field          Content  Type of Transformation  Related mmCIF field

CRYST1[1-6]        CRYST1   NA
CRYST1[7-15]       a        equivalent to           _cell.length_a
CRYST1[16-24]      b        equivalent to           _cell.length_b
CRYST1[25-33]      c        equivalent to           _cell.length_c
CRYST1[34-40]      alpha    equivalent to           _cell.angle_alpha
CRYST1[41-47]      beta     equivalent to           _cell.angle_beta
CRYST1[48-54]      gamma    equivalent to           _cell.angle_gamma
CRYST1[56-66]      sGroup   equivalent to           _symmetry.space_group_name_H-M
CRYST1[67-70]      Z        equivalent to           _cell.Z_PDB
```

Fig. 3. Example of a simple concordance between the PDB CRYST1 record type and mmCIF format.

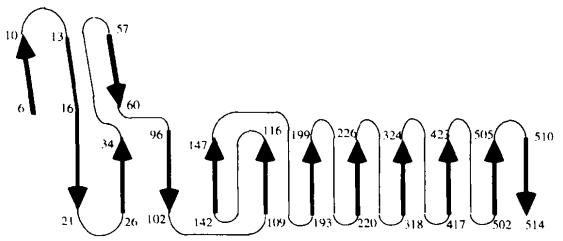

Fig. 4. Diagram of PDB entry 2ACE showing strands forming sheets.

of a PDB entry to an mmCIF data-set takes from several seconds to a few minutes depending on the size of the PDB entry. The longest processing times are, for example, in NMR entries with multiple models. The mmCIF data-sets produced are approximately the same size as the original PDB entries. Table 4 provides the statistics for some conversions done on an SGI R8000 Power Indigo-2 Extreme with 128 Mbytes of memory.

The time is approximately linear in the file size and dominated by the processing time of the atom list. The times given in Table 4 are wall-clock times and approximate the processor

(*a*)

```
SHEET    1   A 3 LEU     6  THR     10  0
SHEET    2   A 3 GLY    13  MET     16 -1  N  VAL     15  O  VAL      8
SHEET    3   A 3 VAL    57  ALA     60  1  N  TRP     58  O  LYS     14
SHEET    1   B11 MET    16  PRO     21  0
SHEET    2   B11 HIS    26  PRO     34 -1  O  ALA     29  N  THR     18
SHEET    3   B11 TYR    96  PRO    102 -1  N  ILE     99  O  PHE     30
SHEET    4   B11 VAL   142  SER    147 -1  N  LEU    143  O  TRP    100
SHEET    5   B11 THR   109  TYR    116  1  N  MET    112  O  VAL    142
SHEET    6   B11 THR   193  GLU    199  1  O  THR    195  N  VAL    113
SHEET    7   B11 ARG   220  SER    226  1  N  ILE    223  O  ILE    196
SHEET    8   B11 GLN   318  ASN    324  1  N  GLY    322  O  LEU    224
SHEET    9   B11 GLY   417  PHE    423  1  N  TYR    421  O  LEU    321
SHEET   10   B11 PHE   502  LEU    505  1  N  ILE    503  O  LEU    420
SHEET   11   B11 MET   510  GLN    514 -1  N  HIS    513  O  PHE    502
```

(*b*)

```
loop_
_struct_sheet.id
_struct_sheet.number_strands
  A     3
  B    11

loop_
_struct_sheet_hbond.sheet_id
_struct_sheet_hbond.range_id_1
_struct_sheet_hbond.range_id_2
_struct_sheet_hbond.range_1_beg_auth_seq_id
_struct_sheet_hbond.range_1_beg_label_atom_id
_struct_sheet_hbond.range_2_beg_auth_seq_id
_struct_sheet_hbond.range_2_beg_label_atom_id
_struct_sheet_hbond.range_1_end_auth_seq_id
_struct_sheet_hbond.range_1_end_label_atom_id
_struct_sheet_hbond.range_2_end_auth_seq_id
_struct_sheet_hbond.range_2_end_label_atom_id
_struct_sheet_hbond.range_1_beg_label_seq_id
_struct_sheet_hbond.range_2_beg_label_seq_id
_struct_sheet_hbond.range_1_end_label_seq_id
_struct_sheet_hbond.range_2_end_label_seq_id
  A     1_A     2_A     8  O     15  N      8  O     15  N
                        8        15         8        15
  A     2_A     3_A    14  O     58  N     14  O     58  N
                       14        58        14        58
  B     1_B     2_B    18  N     29  O     18  N     29  O
                       18        29        18        29
  B    10_B    11_B   502  O    513  N    502  O    513  N
                      502       513       502       513
  B     2_B     3_B    30  O     99  N     30  O     99  N
                       30        99        30        99
  B     3_B     4_B   100  O    143  N    100  O    143  N
                      100       143       100       143
  B     4_B     5_B   142  O    112  N    142  O    112  N
                      142       112       142       112
  B     5_B     6_B   113  N    195  O    113  N    195  O
                      113       195       113       195
  B     6_B     7_B   196  O    223  N    196  O    223  N
                      196       223       196       223
  B     7_B     8_B   224  O    322  N    224  O    322  N
                      224       322       224       322
  B     8_B     9_B   321  O    421  N    321  O    421  N
                      321       421       321       421
  B     9_B    10_B   420  O    503  N    420  O    503  N
                      420       503       420       503
```

Fig. 5. (*a*) SHEET information from PDB entry 2ACE. (*b*) mmCIF `struct_sheet` and `struct_sheet_hbond` tables converted by *pdb2cif* from SHEET information in PDB entry 2ACE. (*c*) mmCIF `struct_sheet_order` and `struct_sheet_range` tables converted by *pdb2cif* from SHEET information in PDB entry 2ACE.

time on larger machines (assuming exclusive use). For large NMR entries processed on small machines, the wall-clock time can become very large due to extensive page swapping for the arrays used to hold the atom list.

The program produces summary warnings as comments at the end of each mmCIF data-set it produces. If a record is found with an unrecognized PDB record type it is reported in the AUDIT category. Warnings and converted records should be examined carefully, especially for the following record types.

COMPND, SOURCE, TITLE and CAVEAT are merged into `_struct.title` without further parsing. Additional information could be derived from PDB entries that follow the PDB 1996 format description when sufficient information for mapping of the PDB MOL_ID to mmCIF entities is available.

EXPDTA records use values that do not have a direct mapping to enumerated values for `_exptl.method`.

(*c*)
```
loop_
_struct_sheet_order.sheet_id
_struct_sheet_order.range_id_1
_struct_sheet_order.range_id_2
_struct_sheet_order.offset
_struct_sheet_order.sense
  A     1_A     2_A +1 anti-parallel
  A     2_A     3_A +1 parallel
  B     1_B     2_B +1 anti-parallel
  B    10_B    11_B +1 anti-parallel
  B     2_B     3_B +1 anti-parallel
  B     3_B     4_B +1 anti-parallel
  B     4_B     5_B +1 parallel
  B     5_B     6_B +1 parallel
  B     6_B     7_B +1 parallel
  B     7_B     8_B +1 parallel
  B     8_B     9_B +1 parallel
  B     9_B    10_B +1 parallel

loop_
_struct_sheet_range.sheet_id
_struct_sheet_range.id
_struct_sheet_range.beg_label_comp_id
_struct_sheet_range.beg_label_asym_id
_struct_sheet_range.beg_auth_seq_id
_struct_sheet_range.end_label_comp_id
_struct_sheet_range.end_label_asym_id
_struct_sheet_range.end_auth_seq_id
_struct_sheet_range.beg_label_seq_id
_struct_sheet_range.end_label_seq_id
  A     1_A LEU *     6  THR *    10
                      6             10
  A     2_A GLY *    13  MET *    16
                     13             16
  A     3_A VAL *    57  ALA *    60
                     57             60
  B     1_B MET *    16  PRO *    21
                     16             21
  B    10_B PHE *   502  LEU *   505
                    502            505
  B    11_B MET *   510  GLN *   514
                    510            514
  B     2_B HIS *    26  PRO *    34
                     26             34
  B     3_B TYR *    96  PRO *   102
                     96            102
  B     4_B VAL *   142  SER *   147
                    142            147
  B     5_B THR *   109  TYR *   116
                    109            116
  B     6_B THR *   193  GLU *   199
                    193            199
  B     7_B ARG *   220  SER *   226
                    220            226
  B     8_B GLN *   318  ASN *   324
                    318            324
  B     9_B GLY *   417  PHE *   423
                    417            423
```

Fig. 5 (*cont.*)

Table 4. *pdb2cif conversion times (s) on an SGI R8000 Power Indigo-2*

	Size in characters (× 1000)		
PDB entry	PDB	mmCIF	Conversion time (s)
4INS	117	130	2.7
1CTJ	170	179	2.7
2ACE	393	433	7.3
4HIR	1753	1896	28.8

ATOM/HETATM records in PDB entries conforming to the 1996 PDB format have a field for a segment ID. The field is mapped to the mmCIF data item `_atom_site.auth_asym_id`, but the data type used in the dictionary does not permit embedded blanks, which may occur in the field. The problem is side-stepped for totally blank fields by mapping them to a period. Nonblank segment IDs are presented in the mmCIF data-set in quotation marks, *e.g.* as 'VH 1', but, strictly speaking, if the rule in the mmCIF dictionary for this data item is not relaxed, the embedded blank should be replaced to make a valid mmCIF data-set.

It must be noted, even though the documentation of the program includes a partial concordance between PDB format and mmCIF, the program itself is not table driven. At present the relationship between PDB format and mmCIF is too complex to be handled by use of a table. However, it may be helpful in understanding the discussion that follows to refer to the extract from the concordance given in Fig. 7. The full concordance can be found at http://ndbserver.rutgers.edu/NDB/mmcif/software/pdb2cif/concord.html.

One of the most challenging parts of the conversion done by *pdb2cif* is the identification of chemical entities. *pdb2cif* does this by scanning SEQRES and ATOM PDB records for sequence homology indicating homologous chains and therefore equivalence as chemical entities. Doubtful cases are reported by warning comments in the mmCIF output. In mmCIF, an appropriate entity must be assigned to each unique structural element in the asymmetric unit. This includes polypeptide chains, polynucleotide chains, solvent, counter ions and other discrete chemical components such as inhibitors. If the same chemical entity appears more than once it must be given the same entity identification. This differs from PDB format in which chains are not explicitly associated with particular chemical entities. Let us first consider the handling of heterogens.

In the PDB format there is effectively an explicit identification of heterogeneous molecular entities by means of PDB FORMUL records. Each heterogen that is not integrated into

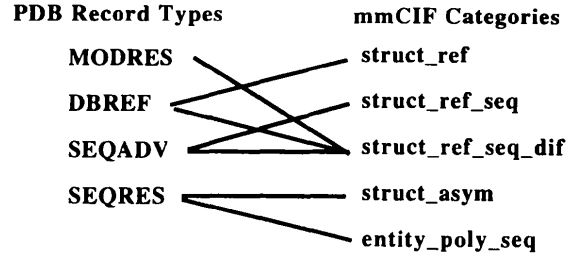

Fig. 6. Mapping of PDB primary structure section records to mmCIF categories.

the backbone of a chain has a component number in columns 9–10 of the associated FORMUL record that may be used as a value for the mmCIF `_entity.id`. Within an entry this number uniquely identifies the particular heterogen as a chemical entity. Alternatively, the PDB three-letter heterogen ID (HetID) in columns 13–15 of the FORMUL record and columns 8–10 of the HET record could equally well be used to identify uniquely the entity for a heterogen. While the HetID has the singular advantage of being an identifier with global meaning valid for all PDB entries, the mmCIF token `_struct_asym.id` can be used to hold the HetID just as well. Therefore, for heterogens, we assign the FORMUL component number as the mmCIF `_entity.id` for heterogens, so that the PDB assignments will not be lost.

The identification of molecular entities for macromolecular chains is more complex and requires the use of implicit, rather than explicit, information from the PDB entry. Consider the sequence and heterogen information from PDB entry 4INS (Baker *et al.*, 1988) given in Fig. 8(*a*). There are four polypeptide chains (*A*, *B*, *C* and *D*), two zinc ions and 350 solvent molecules. On inspection of the sequences, it is clear that chains *A* and *C* are identical and chains *B* and *D* are identical. The program *pdb2cif* makes the same inspection by representing each residue by a single letter, converting each chain sequence into a character string and then performing substring matching to identify the chains that agree. The program insists on an exact match to declare two chains to be the same chemical entity, but warns of chains that show a match of more than 85% and less than 100% of the sequence. In the mmCIF produced by *pdb2cif* the `entity` category is used to report the distinct entity types and the `struct_asym` category is used to report the entity assigned to each chain or heterogen in the asymmetric unit. The `chem_comp` category is used to hold the chemical information. Note that, in order to satisfy mmCIF requirements for complete information about all the chemical components used, we list the amino acids as well as the heterogens. The resulting entity assignments made by *pdb2cif* are shown in Fig. 8(*b*).

Subsequence matching is then used to assign positions within the mmCIF `entity_poly_seq` table to each residue in the atom list. On each matching pass, an attempt is made to match the entire length of the remaining unmatched sequence and then the matching window is reduced by factors of the square root of two until we are working with a sequence fragment of length 16 or less, and then the window is reduced one residue at a time. The PDB ATOM list does not directly associate a residue with a position on the chain sequence, since the residue numbering used in the PDB ATOM list can have deletions, insertions, or be numbered in any arbitrary manner (even backwards or with negative numbers) prescribed by the author. Therefore, the residue numbers in the PDB ATOM list cannot be used for this assignment. However, *pdb2cif* does issue a warning message if the sequence matches the implicit ATOM list sequence for less than 90% of a chain. In the case of NMR entries, a cross-comparison is also made between the implicit sequences of each of the models and a warning is issued if any mismatches are found. Consider PDB entry 1CWP (cowpea chlorotic mottle virus) (Speir *et al.*, 1995). The sequence information in the SEQRES records shown in Fig. 9(*a*) implicitly defines a single entity for chains *A*, *B* and *C*, starting MET, SER, THR, but the ATOM list starts with residue 42. *pdb2cif* correctly makes the necessary sequence number assignments despite only 78% homology for chain *A*. This information is analyzed

PDB Record Type	**PDB Field Name**		**mmCIF data item name**
SEQRES[1-6]	SEQRES	NA	
SEQRES[9-10]	serialNum	NA	
SEQRES[12]	chainID	complex	_struct_asym.id
	used to obtain		_entity_poly_seq.entity_id
SEQRES[14-17]	numRes		
SEQRES[20-22]	resName	==	_entity_seq_mon_id
SEQRES[24-26]	resName	==	_entity_seq_mon_id
[...]			
SEQRES[64-66]	resName	==	_entity_seq_mon_id
SEQRES[68-70]	resName	==	_entity_seq_mon_id
HET[1-6]	HET		
HET[8-10]	hetID	==	_chem_comp.id
HET[13]	ChainID		
HET[14-17]	seqNum		
HET[18]	insertCode		
HET[21-25]	numAtoms	complex	_chem_comp.number_atoms_all or _chem_comp.number_atoms_nh
HET[31-70]	Text	==	_chem_comp.details
FORMUL[1-6]	FORMUL	NA	
FORMUL[9-10]	Component number	complex	_chem_comp.entity_id
FORMUL[13-15]	hetID	related	_chem_comp.id
FORMUL[18]	contin.	NA	
FORMUL[19]	'*' for water		
FORMUL[20-70]	Chemical Formula	~	_chem_comp.formula

Fig. 7. Extract from the partial concordance of PDB format and mmCIF. The concordance shows some of the information needed to understand the mapping from PDB SEQRES records to mmCIF entities. (The notation '==' means 'equivalent to'; '~' means 'approximately equivalent to'; 'complex' means that a complex transition is involved; 'related' means that there is a relationship; and 'NA' means 'not applicable'.)

to find one entity for polypeptide chains *A*, *B* and *C*, a second entity for polynucleotide chains *D* and *F*, and a third entity for polynucleotide chain *E*. When the first entity sequence is matched to the ATOM list, only 78% homology is found for chain *A*, and 86% for chains *B* and *C*. The entity/sequence assignments (Fig. 9*b*) are then applied to the ATOM list without use of the author-assigned residue numbers or insertion codes, but purely from sequence homology. The result, shown in Fig. 9(*c*) is the same identification as made by the authors of 1CWP.

(*a*)

```
SEQRES   1 A   21  GLY ILE VAL GLU GLN CYS CYS THR SER ILE CYS SER LEU  4INS 170
SEQRES   2 A   21  TYR GLN LEU GLU ASN TYR CYS ASN                      4INS 171
SEQRES   1 B   30  PHE VAL ASN GLN HIS LEU CYS GLY SER HIS LEU VAL GLU  4INS 172
SEQRES   2 B   30  ALA LEU TYR LEU VAL CYS GLY GLU ARG GLY PHE PHE TYR  4INS 173
SEQRES   3 B   30  THR PRO LYS ALA                                      4INS 174
SEQRES   1 C   21  GLY ILE VAL GLU GLN CYS CYS THR SER ILE CYS SER LEU  4INS 175
SEQRES   2 C   21  TYR GLN LEU GLU ASN TYR CYS ASN                      4INS 176
SEQRES   1 D   30  PHE VAL ASN GLN HIS LEU CYS GLY SER HIS LEU VAL GLU  4INS 177
SEQRES   2 D   30  ALA LEU TYR LEU VAL CYS GLY GLU ARG GLY PHE PHE TYR  4INS 178
SEQRES   3 D   30  THR PRO LYS ALA                                      4INS 179

HET     ZN      1       1     ZINC ION ON 3-FOLD CRYSTAL AXIS           4INS 191
HET     ZN      2       1     ZINC ION ON 3-FOLD CRYSTAL AXIS           4INS 192
FORMUL   5   ZN    2(ZN1 ++)                                           4INS 193
FORMUL   6  HOH   *350(H2 O1)                                          4INS 194
```

(*b*)

```
loop_
_entity.id
_entity.type
_entity.details
     1     polymer
; Protein chain: A, C
;
     2     polymer
; Protein chain: B, D
;
     5     non-polymer 'het group  ZN'
     6     water       'HOH'

loop_
_struct_asym.entity_id
_struct_asym.id
     1 A
     2 B
     1 C
     2 D
     5  ZN
     6 HOH

loop_
_chem_comp.id
_chem_comp.mon_nstd_flag
_chem_comp.formula
_chem_comp.name
 ZN no
; 2(ZN1 ++)
;
; ZINC ION ON 3-FOLD CRYSTAL AXIS
;
HOH no
; 350(H2 O1)
;
;
;
ALA yes 'C3 H7 N1 O2'             'Alanine'
ARG yes 'C6 H14 N4 O2'            'Arginine'
ASN yes 'C4 H8 N2 O3'             'Asparagine'
CYS yes 'C3 H7 N1 O2 S1'          'Cysteine'
GLN yes 'C5 H10 N2 O3'            'Glutamine'
GLU yes 'C5 H9 N1 O4'             'Glutamic acid'
GLY yes 'C2 H5 N1 O2'             'Glycine'
HIS yes 'C6 H9 N3 O2'             'Histidine'
ILE yes 'C6 H13 N1 O2'            'Isoleucine'
LEU yes 'C6 H13 N1 O2'            'Leucine'
LYS yes 'C6 H14 N2 O2'            'Lysine'
PHE yes 'C9 H11 N1 O2'            'Phenylalanine'
PRO yes 'C5 H9 N1 O2'             'Proline'
SER yes 'C3 H7 N1 O3'             'Serine'
THR yes 'C4 H9 N1 O3'             'Threonine'
TYR yes 'C9 H11 N1 O3'            'Tyrosine'
VAL yes 'C5 H11 N1 O2'            'Valine'
```

Fig. 8. (*a*) Sequence and heterogen information from PDB entry 4INS. (*b*) Entity assignments made by *pdb2cif* for PDB entry 4INS.

(*a*)
```
SEQRES   1 A  190  MET SER THR VAL GLY THR GLY LYS LEU THR ARG ALA GLN  1CWP 114
SEQRES   2 A  190  ARG ARG ALA ALA ALA ARG LYS ASN LYS ARG ASN THR ARG  1CWP 115
SEQRES   3 A  190  VAL VAL GLN PRO VAL ILE VAL GLU PRO ILE ALA SER GLY  1CWP 116
SEQRES   4 A  190  GLN GLY LYS ALA ILE LYS ALA TRP THR GLY TYR SER VAL  1CWP 117
SEQRES   5 A  190  SER LYS TRP THR ALA SER CYS ALA ALA ALA GLU ALA LYS  1CWP 118
SEQRES   6 A  190  VAL THR SER ALA ILE THR ILE SER LEU PRO ASN GLU LEU  1CWP 119
SEQRES   7 A  190  SER SER GLU ARG ASN LYS GLN LEU LYS VAL GLY ARG VAL  1CWP 120
SEQRES   8 A  190  LEU LEU TRP LEU GLY LEU LEU PRO SER VAL SER GLY THR  1CWP 121
SEQRES   9 A  190  VAL LYS SER CYS VAL THR GLU THR GLN THR THR ALA ALA  1CWP 122
SEQRES  10 A  190  ALA SER PHE GLN VAL ALA LEU ALA VAL ALA ASP ASN SER  1CWP 123
SEQRES  11 A  190  LYS ASP VAL VAL ALA ALA MET TYR PRO GLU ALA PHE LYS  1CWP 124
SEQRES  12 A  190  GLY ILE THR LEU GLU GLN LEU ALA ALA ASP LEU THR ILE  1CWP 125
SEQRES  13 A  190  TYR LEU TYR SER SER ALA ALA LEU THR GLU GLY ASP VAL  1CWP 126
SEQRES  14 A  190  ILE VAL HIS LEU GLU VAL GLU HIS VAL ARG PRO THR PHE  1CWP 127
SEQRES  15 A  190  ASP ASP SER PHE THR PRO VAL TYR                      1CWP 128
SEQRES   1 B  190  MET SER THR VAL GLY THR GLY LYS LEU THR ARG ALA GLN  1CWP 129
SEQRES   2 B  190  ARG ARG ALA ALA ALA ARG LYS ASN LYS ARG ASN THR ARG  1CWP 130

[ ... portions of chains B and C omitted here ...]

SEQRES  13 C  190  TYR LEU TYR SER SER ALA ALA LEU THR GLU GLY ASP VAL  1CWP 156
SEQRES  14 C  190  ILE VAL HIS LEU GLU VAL GLU HIS VAL ARG PRO THR PHE  1CWP 157
SEQRES  15 C  190  ASP ASP SER PHE THR PRO VAL TYR                      1CWP 158
SEQRES   1 D    4    A   U   A   U                                      1CWP 159
SEQRES   1 E    2    A   U                                              1CWP 160
SEQRES   1 F    4    A   U   A   U                                      1CWP 161
```

(*b*)
```
loop_
_entity_poly_seq.entity_id
_entity_poly_seq.num
_entity_poly_seq.mon_id
     1    1 MET     1    2 SER     1    3 THR     1    4 VAL     1    5 GLY
     1    6 THR     1    7 GLY     1    8 LYS     1    9 LEU     1   10 THR
     1   11 ARG     1   12 ALA     1   13 GLN     1   14 ARG     1   15 ARG
     1   16 ALA     1   17 ALA     1   18 ALA     1   19 ARG     1   20 LYS
     1   21 ASN     1   22 LYS     1   23 ARG     1   24 ASN     1   25 THR
     1   26 ARG     1   27 VAL     1   28 VAL     1   29 GLN     1   30 PRO
     1   31 VAL     1   32 ILE     1   33 VAL     1   34 GLU     1   35 PRO
     1   36 ILE     1   37 ALA     1   38 SER     1   39 GLY     1   40 GLN
     1   41 GLY     1   42 LYS     1   43 ALA     1   44 ILE     1   45 LYS
     1   46 ALA     1   47 TRP     1   48 THR     1   49 GLY     1   50 TYR
     1   51 SER     1   52 VAL     1   53 SER     1   54 LYS     1   55 TRP

[ ... portions of entity 1 sequence omitted ... ]

     1  176 GLU     1  177 HIS     1  178 VAL     1  179 ARG     1  180 PRO
     1  181 THR     1  182 PHE     1  183 ASP     1  184 ASP     1  185 SER
     1  186 PHE     1  187 THR     1  188 PRO     1  189 VAL     1  190 TYR
     4  191   A     4  192   U     4  193   A     4  194   U
     5  195   A     5  196   U
# *** WARNING *** only    78% homology to chain   A
# *** WARNING *** only    86% homology to chain   B
# *** WARNING *** only    86% homology to chain   C

loop_
_entity.id
_entity.type
_entity.details
     1     polymer
; Protein chain: A, B, C
;
     4     polymer
; Nucleic Acid chain: D, F
;
     5     polymer
; Nucleic Acid chain: E
;

loop_
_struct_asym.entity_id
_struct_asym.id
     1 A
     1 B
     1 C
     4 D
     5 E
     4 F
```

Fig. 9. (*a*) Sequence information from PDB entry 1CWP. (*b*) Entity assignments made by *pdb2cif* for PDB entry 1CWP. (*c*) Entity assignments in the `atom_site` table made by *pdb2cif* for PDB entry 1CWP.

The program accepts all current PDB record types. Figs. 10(*a*) and 10(*b*) show examples for the DBREF and ANISOU PDB record types from the PDB entry 1CTJ (cytochrome c6) (Frazao *et al.*, 1995). *pdb2cif* inserts the necessary tags and values into the `atom_site` table, but uses a different ordering, as shown in Fig. 10(*c*). Also, note the change in scaling, because the values for anisotropic U in mmCIF are not multiplied by 10 000 as in PDB entries.

The organization of the `atom_site` records into lines was dictated by the limit of 80 characters per line in mmCIF, a desire to keep related information together and organized into columns that could easily be scanned by eye. It would have made an equally valid mmCIF data-set to have removed most of the white space and presented the three lines of data which are the first row of the table as:

```
1 . ATOM N N GLU * 1 A 4.127 26.179 −7.903
0.49 57.53 . 1 1 0.9336 0.0004 0.2737 0.7394
0.2771 0.4591
```

6. Dealing with blanks

A PDB entry may have many blank fields and omitted records, but mmCIF format does not permit blank or skipped fields. This restriction in mmCIF is necessary in order to retain the correct alignment of the name–value mapping between the column headings and the values within tables. For example, in the ATOM records of 1CTJ above, the chain identifier, the insertion code and footnote fields are blank. In most cases, *pdb2cif* translates a blank field in PDB format to a period, to denote an intentionally blank field. In some cases, question marks are used instead of periods in, for example, some fields in citations, because there is a possibility that some of the information could be filled in from other sources (*e.g.* `_citation.journal_issue`). Blank insertion codes are ignored rather than converted, since *pdb2cif* appends the insertion code to the residue number to form `_atom_site.auth_seq_id`. There is no possibility of unintentional duplications in this field in recent PDB entries, since the PDB does not use numerical insertion codes. However, it is possible that some old PDB entries might contain numeric insertion codes. In those few cases, it is possible that residue '9' with insertion code '2' might be confused with residue '92'. When the PDB converts its older entries to the current format, any numeric insertion codes will be changed to alphabetic characters.

The most difficult question of blank fields arises from blank chain identifiers in PDB entries. The PDB uses a blank as the chain identifier in almost all entries with only one chain. In this case, a quoted blank or a question mark as the mmCIF translation of the blank PDB chain identifier might have the wrong connotation. Therefore, except when translating the chain identifiers for heterogens in a structure with multiple chains, we replace a blank chain identifier with an asterisk. An asterisk is not a special character in mmCIF, but is a character that is never used in PDB entries for a chain identifier. This provides a valid chain identifier in the mmCIF data-set while preserving the information that the original chain identifier in the PDB entry was blank. Therefore, `atom_site` records for heterogens in which the PDB chain identifier is blank are given as a period for `atom_site.asym_id` unless the PDB entry has only one chain that had a blank chain identifier in the PDB entry. This avoids any implications about chain assignments for heterogens in a PDB entry with multiple chains for which the PDB entry did not make any chain assignment.

(*c*)

```
loop_
_atom_site.label_seq_id
_atom_site.group_PDB
_atom_site.type_symbol
_atom_site.label_atom_id
_atom_site.label_comp_id
_atom_site.label_asym_id
_atom_site.auth_seq_id
_atom_site.label_alt_id
_atom_site.cartn_x
_atom_site.cartn_y
_atom_site.cartn_z
_atom_site.occupancy
_atom_site.B_iso_or_equiv
_atom_site.footnote_id
_atom_site.label_entity_id
_atom_site.id
42
  ATOM  N  N   LYS A  42  .  72.004 -56.695  52.682  1.00 20.00  .  1  1
42
  ATOM  C  CA  LYS A  42  .  72.198 -55.311  52.149  1.00 20.00  .  1  2
42
  ATOM  C  C   LYS A  42  .  73.687 -55.156  51.846  1.00 20.00  .  1  3
42
  ATOM  O  O   LYS A  42  .  74.532 -55.589  52.633  1.00 20.00  .  1  4
42
  ATOM  C  CB  LYS A  42  .  71.786 -54.251  53.201  1.00 20.00  .  1  5
42
  ATOM  C  CG  LYS A  42  .  70.359 -54.405  53.774  1.00 20.00  .  1  6
42
  ATOM  C  CD  LYS A  42  .  70.073 -53.380  54.893  1.00 20.00  .  1  7
42
  ATOM  C  CE  LYS A  42  .  68.635 -53.512  55.436  1.00 20.00  .  1  8
42
  ATOM  N  NZ  LYS A  42  .  68.255 -52.521  56.514  1.00 20.00  .  1  9
43
  ATOM  N  N   ALA A  43  .  74.000 -54.638  50.668  1.00 20.00  .  1  10
43
  ATOM  C  CA  ALA A  43  .  75.388 -54.404  50.294  1.00 20.00  .  1  11
```

Fig. 9 (*cont.*)

(*a*)
```
DBREF  1CTJ      1    89  SWS    Q09099   CYC6_MONBR       1     89
```

is converted into two tables by *pdb2cif*:

```
loop_
_struct_ref.id
_struct_ref.entity_id
_struct_ref.biol_id
_struct_ref.db_name
_struct_ref.db_code
_struct_ref.seq_align
_struct_ref.seq_dif
_struct_ref.details
     1    1    *      SWS  'Q09099 CYC6_MONBR'  partial   no .

loop_
_struct_ref_seq.align_id
_struct_ref_seq.ref_id
_struct_ref_seq.seq_align_beg
_struct_ref_seq.seq_align_end
_struct_ref_seq.db_align_beg
_struct_ref_seq.db_align_end
_struct_ref_seq.details
     1     1     '1'    '89'     '1'    '89' .
```

(*b*)
```
ATOM      1  N  AGLU     1       4.127  26.179  -7.903  0.49 57.53           N
ANISOU    1  N  AGLU     1     9336   7394   4591      4   2737   2771       N
ATOM      2  N  BGLU     1       3.535  25.488 -12.889  0.51 54.52           N
ANISOU    2  N  BGLU     1     8406   5015   6783   -887   3093    161       N
ATOM      3  CA AGLU     1       5.490  26.607  -8.207  0.49 52.50           C
ANISOU    3  CA AGLU     1     9283   5563   4611   -256   2331   1241       C
ATOM      4  CA BGLU     1       2.754  26.395 -12.051  0.51 51.27           C
ANISOU    4  CA BGLU     1     7663   5124   6212   -653   2258    184       C
ATOM      5  C  AGLU     1       5.550  27.734  -9.233  0.49 47.55           C
ANISOU    5  C  AGLU     1     8593   4752   4275   -880   1820    625       C
```

(*c*)
```
loop_
_atom_site.label_seq_id
_atom_site.auth_asym_id
_atom_site.group_PDB
_atom_site.type_symbol
_atom_site.label_atom_id
_atom_site.label_comp_id
_atom_site.label_asym_id
_atom_site.auth_seq_id
_atom_site.label_alt_id
_atom_site.cartn_x
_atom_site.cartn_y
_atom_site.cartn_z
_atom_site.occupancy
_atom_site.B_iso_or_equiv
_atom_site.footnote_id
_atom_site.label_entity_id
_atom_site.id
_atom_site.aniso_U[1][1]
_atom_site.aniso_U[1][2]
_atom_site.aniso_U[1][3]
_atom_site.aniso_U[2][2]
_atom_site.aniso_U[2][3]
_atom_site.aniso_U[3][3]

1 .
ATOM  N    N    GLU *    1  A   4.127  26.179  -7.903  0.49 57.53  .    1      1
                         0.9336  0.0004  0.2737  0.7394  0.2771  0.4591
1 .
ATOM  N    N    GLU *    1  B   3.535  25.488 -12.889  0.51 54.52  .    1      2
                         0.8406 -0.0887  0.3093  0.5015  0.0161  0.6783
1 .
ATOM  C    CA   GLU *    1  A   5.490  26.607  -8.207  0.49 52.50  .    1      3
                         0.9283 -0.0256  0.2331  0.5563  0.1241  0.4611
1 .
ATOM  C    CA   GLU *    1  B   2.754  26.395 -12.051  0.51 51.27  .    1      4
                         0.7663 -0.0653  0.2258  0.5124  0.0184  0.6212
1 .
ATOM  C    C    GLU *    1  A   5.550  27.734  -9.233  0.49 47.55  .    1      5
                         0.8593  -0.088   0.182  0.4752  0.0625  0.4275
```

Fig. 10. (*a*) DBREF information from 1CTJ translated to the `struct_ref` and `struct_ref_seq` tables. (*b*) Anisotropic temperature factors from PDB entry 1CTJ. (*c*) Translation of anisotropic temperature factors from PDB entry 1CTJ to appropriate values in the `atom_site` table. Note the change in scaling.

```
save__struct_conn.ptnr1_atom_site_id
    _item_description.description
;               The id of an atom site for the first partner in a bond

                This data item is a pointer to _atom_site.id in the
                ATOM_SITE category.
;
    _item.name                      '_struct_conn.ptnr1_atom_site_id'
    _item.mandatory_code             no
    _item.category_id                struct_conn
    _item_linked.child_name         '_struct_conn.ptnr1_atom_site_id'
    _item_linked.parent_name        '_atom_site.id'
     save_

save__struct_conn.ptnr2_atom_site_id
    _item_description.description
;               The id of an atom site for the second partner in a bond

                This data item is a pointer to _atom_site.id in the
                ATOM_SITE category.
;
    _item.name                      '_struct_conn.ptnr2_atom_site_id'
    _item.mandatory_code             no
    _item.category_id                struct_conn
    _item_linked.child_name         '_struct_conn.ptnr2_atom_site_id'
    _item_linked.parent_name        '_atom_site.id'
     save_

save__atom_site.label_model_id
    _item_description.description
;               A component of the macromolecular identifier for this atom site.
                The value of _atom_site.label_model_id associates the atom
                site with a particular nmr model.

;
    _item.name                      '_atom_site.label_model_id'
    _item.mandatory_code             no
    _item.category_id               'atom_site'
    _item_type.code                  code
     loop_
    _item_linked.child_name
    _item_linked.parent_name
         '_struct_mon_prot.label_model_id'          '_atom_site.label_model_id'
         '_struct_mon_prot_cis.label_model_id'      '_atom_site.label_model_id'
     save_


save__struct_mon_prot.label_model_id
    _item_description.description
;               This data item is a pointer to _atom_site.label_model_id in the
                ATOM_SITE category.
;
    _item.name                      '_struct_mon_prot.label_model_id'
    _item.mandatory_code             no
    _item.category_id                struct_mon_prot
     save_

save__struct_mon_prot_cis.label_model_id
    _item_description.description
;               This data item is a pointer to _atom_site.label_model_id in the
                ATOM_SITE category.
;
    _item.name                      '_struct_mon_prot_cis.label_model_id'
    _item.mandatory_code             no
    _item.category_id                struct_mon_prot_cis
     save_

save__struct_ref_seq_dif.db_seq_num
    _item_description.description
;               The sequence position in the referenced database entry
                corresponding to this point difference position.

                The use of . for _struct_ref_seq_dif.db_seq_num when
                a value has been given for _struct_ref_seq_dif.seq_num
                indicates that there has been an insertion at this
                position.

                The use of . for _struct_ref_seq_dif.seq_num when
                a value is given for _struct_ref_seq_dif.db_seq_num
                indicates that there has been a deletion at this
                position.
;
    _item.name                      '_struct_ref_seq_dif.db_seq_num'
    _item.mandatory_code             no
    _item.category_id                struct_ref_seq_dif
     loop_
    _item_range.maximum
    _item_range.minimum
                                     .    1
                                     1    1
    _item_type.code                  int

     save_
```

Fig. 11. Definitions used by *pdb2cif* which are not in the mmCIF dictionary.

7. mmCIF compliance

The program *pdb2cif* can translate a PDB entry into a data-set that is substantially compliant with the mmCIF dictionary, although careful checking of the results is suggested. This version is intended to produce mmCIF files conforming to mmCIF version 1.0.00 and above. Full compliance is not possible in some areas. In particular, most of the values used for `_exptl.method`, and some of the values used for `_struct_conf_type.id` do not conform to the enumerations in the dictionary. Full compliance would require agreement between the PDB and COMCIFS (the IUCr committee that oversees the CIF dictionaries) on equivalent lists of values. In addition, the PDB has released some entries with truncated author lists, using 'ET AL.' to indicate the missing authors. This practice does not conform to mmCIF requirements and *pdb2cif* does not have access to the information necessary to complete the list of authors.

In order to translate PDB records completely without information loss, *pdb2cif* uses a few tokens that are not in the dictionary. If strict dictionary validation is done, the definitions shown in Fig. 11 would have to be appended to the mmCIF dictionary for validation of *pdb2cif* output.

8. Future plans

Plans call for an extension of the parsing of the internal fields of COMPND and SOURCE and of the newer, more structured remarks (Protein Data Bank, 1996) and compliance with the mmCIF dictionary as it evolves. Ultimately, our goal is to convert from PDB format to mmCIF in sufficient detail as to extract all information for which mmCIF tokens exist and for which information was provided in an entry, while preserving the names and relationships that existed in the PDB entry. In this way, all records of the original entry can be reconstructed from the new mmCIF data-set.

9. Distribution

The latest version of this software is available at any of the following WWW servers:

http://www.sdsc.edu/pb/pdb2cif/pdb2cif
http://ndbserver.rutgers.edu/NBD/mmcif/software
http://www.ebi.ac.uk/NDB/mmcif/software
http://ndbserver.nibh.go.jp/NDB/mmcif/software
http://www.iucr.org/iucr-top/cif/software/pdb2cif

pdb2cif is distributed as `pdb2cif.cshar.Z`, a compressed C-shell self-extracting archive. The structure of this file permits automatic unpacking on Unix systems using the C shell, `csh`, but, unlike the more commonly used 'shar' format, also permits unpacking with a text editor. A `pdb2cif.shar.Z` version is also available.

If an mmCIF data-set produced from a particular PDB entry is required, the 3DB browser (Abola *et al.*, 1996) available at http://www.pdb.bnl.gov has an interface to *pdb2cif* as an output option. Alternatively, the MOOSE database (Shindyalov *et al.*, 1995) available at http://www.sdsc.edu/moose also has an option to display the mmCIF version of any PDB-formatted file. For further information, e-mail yaya@bernstein-plus-sons.com.

This work was supported in part by US NSF, PHS, NIH, NCRR, NIGMS, NLM and DOE under contract DE-AC02-76CH00016 (for FCB), US NSF grant no. BIR 9310154 (for PEB), and the IUCr (for HJB).

References

Abola, E. E., Prilusky, J., Manning, N. O. & Sussman, J. L. (1996). *Acta Cryst.* **A52** Supplement, C-586.

Baker, E. N., Blundell, T. L., Cutfield, J. F., Cutfield, S. M., Dodson, E. J., Dodson, G. G., Crowfoot Hodgkin, D. M., Hubbard, R. E., Isaacs, N. W., Reynolds, C. D., Sakabe, K., Sakabe, N. & Vijayan, N. M. (1988). *Philos. Trans. R. Soc. London*, **319**, 369–456.

Berman, H. M. & Westbrook, J. D. (1994). *A Gentle Introduction to One Working Alternative DDL for Macromolecular Structure*. In *European Macromolecular Crystallographic Information (mmCIF) Workshop*, edited by S. D. Wodak. Free University of Brussels, European Commission.

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.

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

Codd, E. F. (1970). *Commun. ACM*, **13**, 377–387.

Codd, E. F. (1972). *Data Base Systems*, edited by R. Rustin, pp. 33–64. Englewood Cliffs, NJ: Prentice-Hall.

Fitzgerald, P. M. D., Berman, H. M., Bourne, P. E., McMahon, B., Watenpaugh, K. & Westbrook, J. (1996). *Acta Cryst.* **A52** Supplement, C-576.

Frazao, C., Soares, C. M., Carrondo, M. A., Pohl, E., Dauter, Z., Wilson, K. S., Hervas, M., Navarro, J. A., De La Rosa, M. A. & Sheldrick, G. M. (1995). *Structure (London)*, **3**, 1159–1169.

Hall, S. R. (1991). *J. Chem. Inf. Comput. Sci.* **31**, 326–333.

Hall, S. R. & Spadaccini, N. (1994). *J. Chem. Inf. Comput. Sci.* **34**, 505–508 (see http://www.crystal.uwa.edu.au/cc_star.html).

Kernighan, B. W. & Ritchie, D. M. (1977). *The M4 Macro Processor*. Murray Hill, NJ: Bell Laboratories.

Protein Data Bank (1996). *The Protein Data Bank Contents Guide: Atomic Coordinate Entry Format Description*, Version 2.1, http://www.pdb.bnl.gov/format.doc/format_home.html.

Raves, M. L., Harel, M., Pang, Y.-P., Silman, I., Kozikowski, A. P. & Sussman, J. L. (1977). *Nat. Struct. Biol.* **4**, 57–63.

Shindyalov, I. N., Cooper, J., Chang, W. & Bourne, P. E. (1995). *Proceedings of the 28th Hawaii International Conference on System Sciences*, pp. 207–217. Los Alamitos, CA: IEEE Computer Society Press.

Speir, J. A., Munshi, S., Wang, G., Timothy, S., Baker, J. E. & Johnson, J. E. (1995). *Structure (London)*, **3**, 63–78.

Westbrook, J. & Hall, S. R. (1995). *A Dictionary Description Language for Macromolecular Structure, Draft DDL V 2.1.0*, IUCr COMCIFS, Chester, England. Available from http://ndbserver.rutgers.edu/NDB/mmcif/ddl/index.html.