Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006



# Building a Structured PDB: The RS-PDB Database

Zoltán Szabadka and Vince Grolmusz

**Abstract—A method for automatically analyzing structures deposited in the Protein Data Bank is presented. The method is capable to detect missing atoms, bond length deviations, atom bumps and to correctly identify protein-ligand complexes. The results are organized into a database, called the Rich Structure PDB (RS-PDB in short) from which one can easily select PDB entries satisfying diverse sets of requirements. The newer and richer mmCIF format of both the PDB and its Chemical Component Dictionary (formerly the HET Group Dictionary) were used in the construction, and the International Chemical Identifier (InChI) of IUPAC played a main role in correctly identifying distinct ligands.**

## I. Introduction

The Protein Data Bank [1] is the definitive collection of proteins, nucleic acids and their complexes of known 3D structure. Its organization and history implies that the deposited files are often heterogeneously organized, labels and comments are often missing or incorrect. Without careful pre-processing, it seems to be a hopeless task to derive reliable information automatically from the entries in the whole databank. We suggested an algorithm for repairing the entries in the PDB in [2].

In this paper we present a new pre-processing algorithm that works on the mmCIF (macromolecular Crystallographic Information File) format of the PDB, and uses the InChI chemical identifier of the IUPAC [3]. Our method checks the entries for errors and inconsistencies, marks missing atoms, decomposes the structures into protein-, nucleic acid- and polysaccharide chains as well as various types of ligand molecules (e.g. peptides, cofactors/coenzymes, metals, etc.), distinguishes between covalently and non-covalently bound ligands, and identifies different binding sites. The result is a strictly structured, homogeneous database, adequate for processing diverse queries and serving intricate data-mining applications.

Manuscript received on April 2, 2006. This work was supported in part the European Commission FP6 project No. LSHP-CT-2005-012127 and the Hungarian OTKA grant No. T046234. Parts of this work was done in cooperation with Uratim Ltd. and with Math-for-Health LLC.

Zoltán Szabadka and Vince Grolmusz are with the Department of Computer Science, Eötvös University, Pázmany P. stny. 1/C, H-1117 Budapest, Hungary. E-mail: {sinus, grolmusz}@cs.elte.hu.

## II. Methods

The RS-PDB database consists of three parts (c.f., Figure 1).

### A. The Small Molecule Database

The first part is the small molecule database, where molecules in a format similar to that used in SD files are stored. For any atom only its element symbol, its formal charge, and its model coordinates are stored. The bonds can be of order one, two or three, and we store the *Kekule* representation of the aromatic systems. The main difference from the SD format is that the numbering of the atoms in the connection table of the molecule is not arbitrary. We use the canonical numbering generated together with the IUPAC/NIST chemical identifier of the molecule (InChI). To generate this, our program uses the API that was obtained from IUPAC's web-site [3]. If a new molecule is inserted into this part of the database, then first its InChI is generated, and its presence is checked. The key of this part of the database is a sequence number called *molid* that is incremented each time a new InChI is found.

### B. PDB Chemical Component Dictionary (formerly the HET Group Dictionary)

The second part of the database (Figure 1) contains the PDB Chemical Component Dictionary. Its mmCIF format, *components.cif* can be downloaded from RCSB's web-site [4]. It consists of the connection tables of the molecules and functional groups associated with the three letter monomer codes found in the PDB. These connection tables are inserted into the small molecule database after conversion and error-checking. Since the mmCIF format contains four types of bonds: single, double, triple and aromatic, the latter had to be converted to the *Kekule* representation. Because the aromatic bonds are special types of ring bonds, each bond in a molecule marked as aromatic is checked to be a part of a ring. This was done with a depth-first-search graph-traversal of the molecule. To our surprise, we found that there were even a few monomers, where this condition was not met. These errors were corrected manually. Next, with a graph matching algorithm, we selected from the aromatic bonds those ones that were converted to double bonds. Before inserting a monomer into the database the standard "valence rules" are verified as described in the InChI Technical Manual [3].

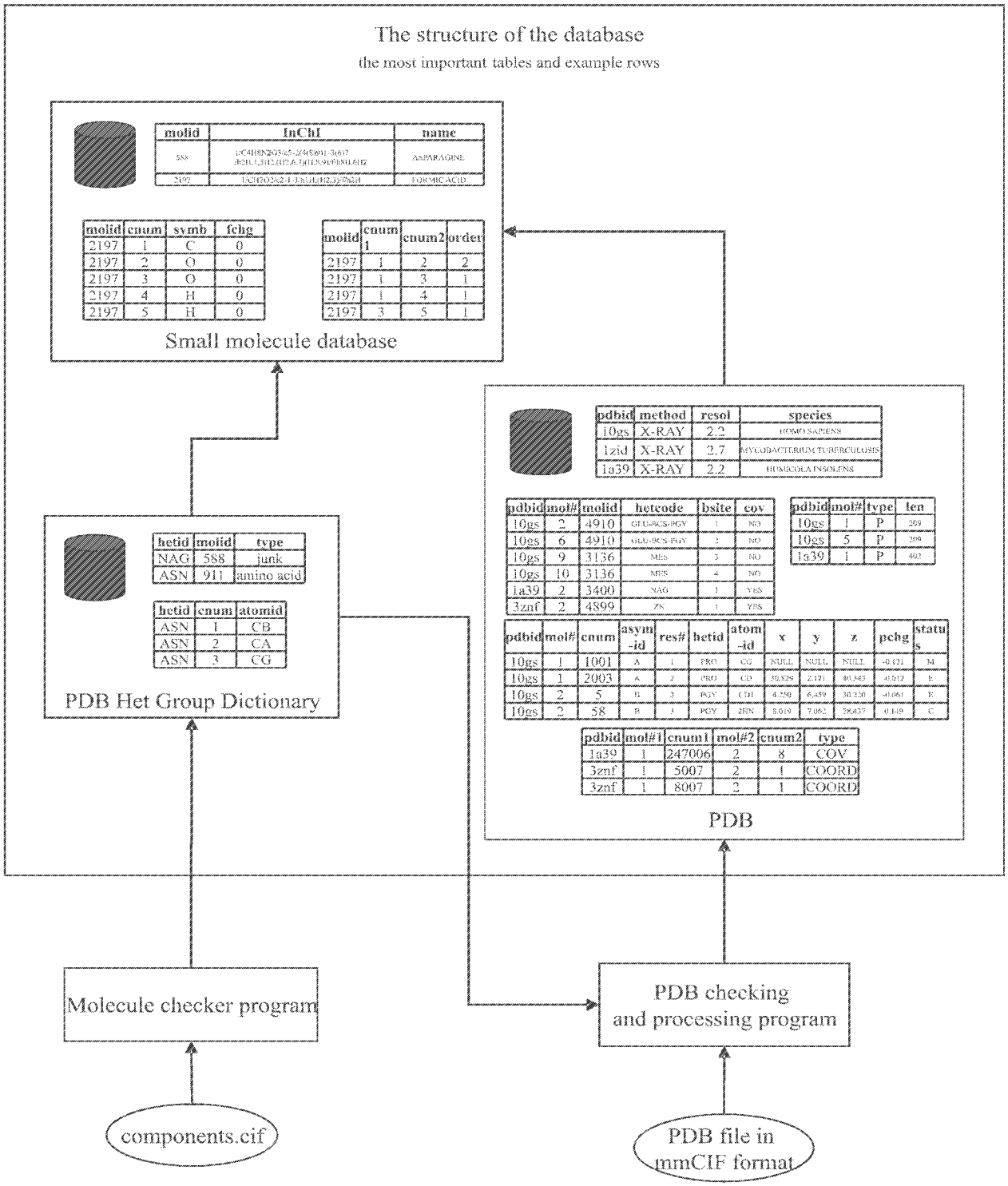


Figure 1: The structure of the RS-PDB database

### C. The Main Part of the Database

The results of processing of the PDB entries are stored in the third, and most important part of the database. After an entry is successfully read by our program, it will consist of polymer and ligand molecules, and each one will have a unique number within the entry. This number is denoted by *mol#* in the database tables. For each ligand molecule, we create its InChI identifier and insert it into the small molecule database as well. In the case that was already there, then we obtain its molid. For every ligand it is recorded whether they bind covalently to a polymer chain, and also the id of the binding site they are in. For the polymers we record only their type ('P'=protein chain and 'N'=DNA/RNA chain) and their length in amino acids. The table listing the atomic coordinates will contain an attribute called the status of the atom. This can be 'M'=missing, 'E'=experimentally determined and 'C'=computed (as in the case of hydrogen atoms). The *cnum* attribute in each table means the canonical number of the atom in the given molecule. For the polymer chains, we define the canonical number of an atom by adding 1000*(residue sequence number) to its canonical number within the monomer residue. Because the covalent bonds within the ligands are stored in the small molecule database, we only have to record the bonds between the ligands and the polymer chains here, as well as the disulphide bonds between the polymer chains. The type of these bonds can be covalent and coordinated covalent denoted by COV and COORD.

### D. Building the Database: Processing the mmCIF Files

We explain step-by-step how an entry, given in the mmCIF format, is processed, checked for errors and is finally decomposed to polymer chains and ligand molecules. When processing an entry, we rely not only on the information given in the structure file itself, but also on the previously processed PDB Chemical Component Dictionary, that contains the covalent structure of each monomer.

The mmCIF format is a kind of database description language, logically, a file consists of many tables, called 'data categories', and the attributes in a table are called 'data items'.

The most important mmCIF data categories:

**struct_asym:** List of the components in the asymmetric unit. Each component has an asym id.

**pdbx_poly_seq_scheme:** Describes the sequence of monomers in a polymer entity.

**pdbx_nonpoly_scheme:** List of the monomers belonging to the non-polymer entities.

**atom_site:** Coordinate data for atoms, whose positions could be experimentally determined.

Presently the following entry header information is stored in the RS-PDB database:

The species of the source organism(s) from which the structure was obtained. This can be found in the *entity_src_gen* table if the source was genetically manipulated, otherwise in the *entity_src_nat* table.

The method used in the experiment ( *exptl.method* ), eg. X-Ray Diffraction, NMR, etc.

The resolution of the structure ( *refine.ls_d_res_high* ).

In an mmCIF file, the contents of the asymmetric unit are listed in the table *struct_asym*. Each item (also called entity) in this list has an asym id. The type of an entity can be polymer, non-polymer or water. Each polymer entity has also a polymer type.

We define a protein chain as a polymer entity of type "polypeptide(L)", if it is at least ten monomers long, and a DNA/RNA chain as a polymer entity, which is at least 5 monomers long and its type is either "polydeoxiribonucleotide", "polyribonucleotide", or more than half of its monomers are nucleic acids (A,C,G,I,T,U monomer id).

At this point the list of monomers that make up a polymer chain is identified. The covalent structure of these monomers (the so-called "connection table") is read from the PDB Chemical Component Dictionary (formerly HET Group Dictionary, HGD). The next step, i.e., connecting the monomers to obtain the covalent structure of the whole chain is performed by adding the monomers to the chain one-by-one.

After creating the connection table for the polymer chains, we read the list of monomers from the table *pdbx_nonpoly_scheme*. The initial set of ligand molecules will be these, plus the monomers from the polymer entities that were not long enough (these will form the oligopeptide ligands, for example). Their connection table is obtained from the HGD.

### E. Building the Database: Inserting Atomic Coordinates

The coordinates of the atoms can be found in the table *atom_site*. We go through this table, and for each row, we try to identify the atom in our previously built covalent model that this row is referring to. This is not always an easy task, because there are four different numbering schemes used simultaneously in mmCIF.

After reading the table *atom_site,* there will be several atoms, whose coordinates are known. However, there will still be several atoms whose coordinates are unknown (e.g., hydrogens, flexible chain segments, residues at the ends of the chain, etc.). We will refer to them as "missing atoms" hereafter.

Next the bond-lengths will be checked for correctness by taking all pairs of atoms that are in the same monomer, and

checking whether their distance is in accordance with the connectivity information in the HGD. We record the deviances in a separate table in our database, so that we can use this information, when selecting the most precise structures.

At this point, we still have the initial set of ligands. A molecule in the final set can consist of two or more such monomers, bound covalently. To identify such covalent bonds, we select all pairs of atoms in the entry that are closer to each other than 6 Angstroms. We achieve this by building a Kd-tree on the atoms, thus avoiding the examination of all pairs, and saving a lot of computational time. After a detailed examination (to be given in the full version of this work), all possible covalent bond is created.

Finally, the coordinates of the hydrogen atoms are computed using standard hybridization rules. We also calculate partial charges for the ligand molecules using the Gasteiger algorithm, and assign the Kollman partial charges for the atoms in protein chains.

## III. Analysis and Results

As an application we selected a set of PDB entries that contain valid protein-ligand complexes, and also have correct structure. To define the term "valid", we adopted and slightly extended the monomer characterization of Wang et al.[5,6]. They have a list of "special" monomers, consisting of biologically important cofactors/coenzymes, as well as "junk" molecules that were found in many PDB entries. Further types of monomers were also defined, such as modified amino acids, modified nucleic acids, metals, and organo-metallic molecules. The modified amino/nucleic acids were identified with the help of the mon_nstd_parent_comp_id attribute of the monomers in the mmCIF format of the HGD. If this was one of the standard amino/nucleic acids, then the monomer was assumed to be its modified form. The monomers that had less than six heavy atoms and were not classified before were assigned the "tiny" type. The other monomers were classified as standard ligand molecules. We defined a ligand to be valid, if it had at least one ligand type monomer or modified amino acid monomer, and if it had no cofactor/coenzyme monomer.

Another straightforward application is identifying and counting the missing atoms of the protein chains in the PDB entries. It is anticipated that atoms from the flexible regions of the protein chains are missing from the crystallographic data. Findings on Figure 2 correspond to this assumption: too short or too long segments cannot be flexible in crystal structures.

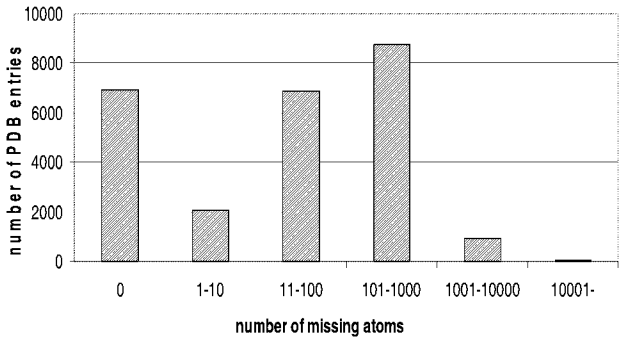


Figure 2: The distribution of the number of missing atoms from protein chains in the PDB entries. Note, that there are relatively few entries, where only a few atoms are missing.

## IV. Conclusion

Using physicochemical rules with graph-theoretical algorithms a new, well-structured version of the Protein Data Bank was built. Two examples were presented for demonstrating the applicability of the database. The most important application is the reliable, automatic identification of protein-ligand complexes in the PDB.

## References


[1] H.M.Berman, J.Westbrook, Z.Feng, G.Gilliland, T.N.Bhat, H.Weissig, I.N.Shindyalov, P.E.Bourne The Protein Data Bank *Nucleic Acids Research*, **28** pp. 235-242 (2000)

[2] Z. Szabadka, V. Grolmusz: High Throughput Processing of the Structural Information of the Protein Data Bank, DIMACS Workshop on Information Processing by Protein Structures in Molecular Recognition, June 13 - 14, 2005, DIMACS Center, Rutgers University

[3] http://www.iupac.org/inchi/index.html

[4] The PDB Chemical Component Dictionary (formerly the Het Group Dictionary) is available at http://deposit.pdb.org/public-component-erf.cif.

[5] Wang et. al, The PDBbind Database: Methodologies and Updates, J. Med. Chem., 2005

[6] Wang et al., The PDBbind Database: Collection of Binding Affinities for Protein-Ligand Complexes with Known Three-Dimensional Structures, J. Med. Chem., 2004