# Protein Repair and Analysis Server: A Web Server to Repair PDB Structures, Add Missing Heavy Atoms and Hydrogen Atoms, and Assign Secondary Structures by Amide Interactions

Osita Sunday Nnyigide, Tochukwu Olunna Nnyigide, Sun-Gu Lee* and Kyu Hyun*

School of Chemical and Biomolecular Engineering, Pusan National University, Busan 46241, Korea

*Correspondence to kyuhyun@pusan.ac.kr or sungulee@pusan.ac.kr

## HOW TO USE PRAS SERVER WITH .pdb STRUCTURE FORMAT

To use PRAS server, simply upload a syntactically correct PDB file (.pdb format) and click the submit button. This normally means a protein structural data obtained from the worldwide PDB database (wwPDB) or its member organizations (i.e., PDBe, PDBj, RCSB, and BMRB).

User is encouraged to read the content of the log.txt file generated by the server to see if there are errors in the PDB structure file and their fixes.

To use a PDB file obtained from other means, user must ensure that the file columns and data correspond to the description given below (*the easiest way to meet this specification is to edit the PDB structure using PyMOL, i.e., open the structure in PyMOL and save a copy. All missing fields will be inserted by PyMOL*). The PRAS program is hard coded to expect/use these specifications and failure to conform to the standard will either raise exception or generate weird results.

The server may work with any PDB file that contains all the columns in **fig.1** with corresponding data as given in **table 1** (including the "TER" and "END" records!). The line for TER record may be complete as in standard PDB file from rcsb.org or just as is in this figure.

**Note that the order the atoms are written is important also (i.e., main atoms followed by beta, gamma, delta, epsilon and zeta sidechain atoms). So, if CYS has no missing atoms, then the 6$^{th}$ atom must be SG otherwise the output PDB may contain cysteine instead of cystine.**

**PRAS assumes that the chains are not bonded covalently and thus, the residue before TER will have OXT added if not present. The number of proteins in the protein data bank whose chains are bonded covalently is negligible, so it is not worth the effort to cover it programmatically. Therefore, if the protein submitted to PRAS has chains that are covalently bonded, user should delete the unwanted atoms added by PRAS. One example is 5DK3.pdb where several CYS are bonded inter-chain, and user should delete extra OXT or HG atoms added to CYS.**

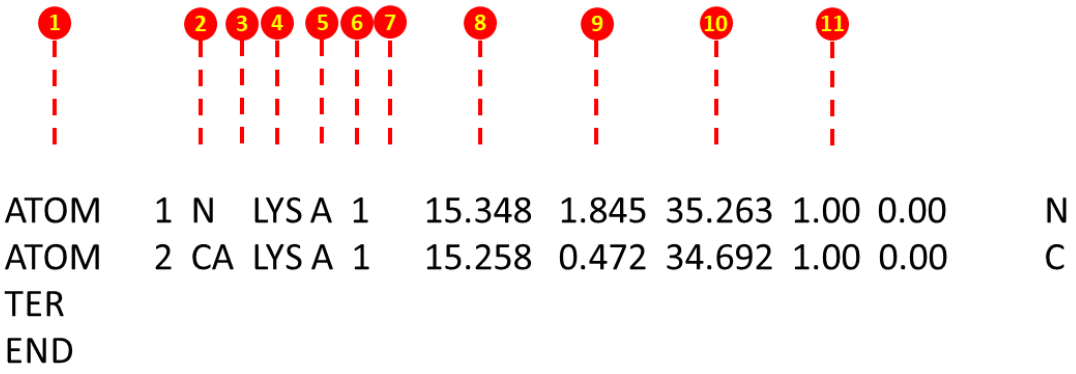


**Fig.1**. Annotation of a sample PDB file. The numbers, "TER" and "END" correspond to columns of the file as summarized in table below

**Table.1**. Summary of **fig.1** (the columns and data they contain)

| S/N | Columns | Data |
|---|---|---|
| 1 | 1-4 | "ATOM" |
| 1 | 1-3 | "TER" |
| 1 | 1-3 | "END" |
| 2 | 13-16 | Atom name |
| 3[a] | 17 | Alternate location indicator |
| 4 | 18-20 | Residue name |
| 5 | 22 | Chain identifier |
| 6 | 23-26 | Residue sequence number |
| 7[b] | 27 | Code for insertions of residues |
| 8 | 31-38 | X coordinate (Å) |
| 9 | 39-46 | Y coordinate (Å) |
| 10 | 47-54 | Z coordinate (Å) |
| 11 | 55-60 | Occupancy |

[a] This column is only used if there's rotamer or point mutation otherwise it is empty.

[b] This column is only used if there's residue insertion otherwise it is empty.

## HOW TO USE PRAS SERVER WITH .cif (PDBx/mmCIF ) STRUCTURE FORMAT

In the case of .cif structure format, the data written in the file should be syntactically correct and have the compulsory category name and attribute name described below. The combination of category and attribute name is known as an mmCIF token. Strictly speaking, the data categories are given in two styles only: key-value and tabular. In the case of key-value style, the mmCIF token is followed directly by a corresponding value. PRAS does not need the data category in key-value style as the information contained therein is not related to atom coordinates. On the other hand, the tabular style is used when the data item presented includes multiple values for each token. The category (in tabular style) that describes the identities and atomic coordinates is the critical or compulsory section that PRAS reads in order to repair or analyze the structure data. As seen below, there are 21 tokens in this category, and 4 out of 21 (highlighted in bold letters) are additional tokens provided by the author. Notice that these are already present (**auth_seq_id** vs label_seq_id). These additional 4 tokens are not essential to PRAS, and PRAS will repair a structure that does not contain those, but the other 17 are essential. For instance, label_seq_id is a positive integer indicating the sequence number while **auth_seq_id** is an optional numbering presented by the author. The **auth_seq_id** is not necessarily a number and the values do not have to be positive or correspond to the value of _atom_site.label_seq_id.

Note that the 17 tokens described must have their corresponding data values in the following line that starts with "ATOM" (thus, group_PDB = ATOM, site.id = 1, type_symbol = N and so on). Notice that label_alt_id = "." While pdbx_PDB_ins_code = "?". The "." means that the value has been omitted intentionally, and the "?" means that the value is missing.

_atom_site.group_PDB

_atom_site.id

_atom_site.type_symbol

_atom_site.label_atom_id

_atom_site.label_alt_id

_atom_site.label_comp_id

_atom_site.label_asym_id

_atom_site.label_entity_id

_atom_site.label_seq_id

_atom_site.pdbx_PDB_ins_code

_atom_site.Cartn_x

_atom_site.Cartn_y

_atom_site.Cartn_z

_atom_site.occupancy

_atom_site.B_iso_or_equiv

_atom_site.pdbx_formal_charge

**_atom_site.auth_seq_id**

**_atom_site.auth_comp_id**

**_atom_site.auth_asym_id**

**_atom_site.auth_atom_id**

_atom_site.pdbx_PDB_model_num

**ATOM 1 N N . ASP A 1 5 ? 32.778 -20.767 16.955 1.00 63.36 ? 453 ASP A N 1**