Supporting the CIF file format of proteins in molecular dynamics simulations
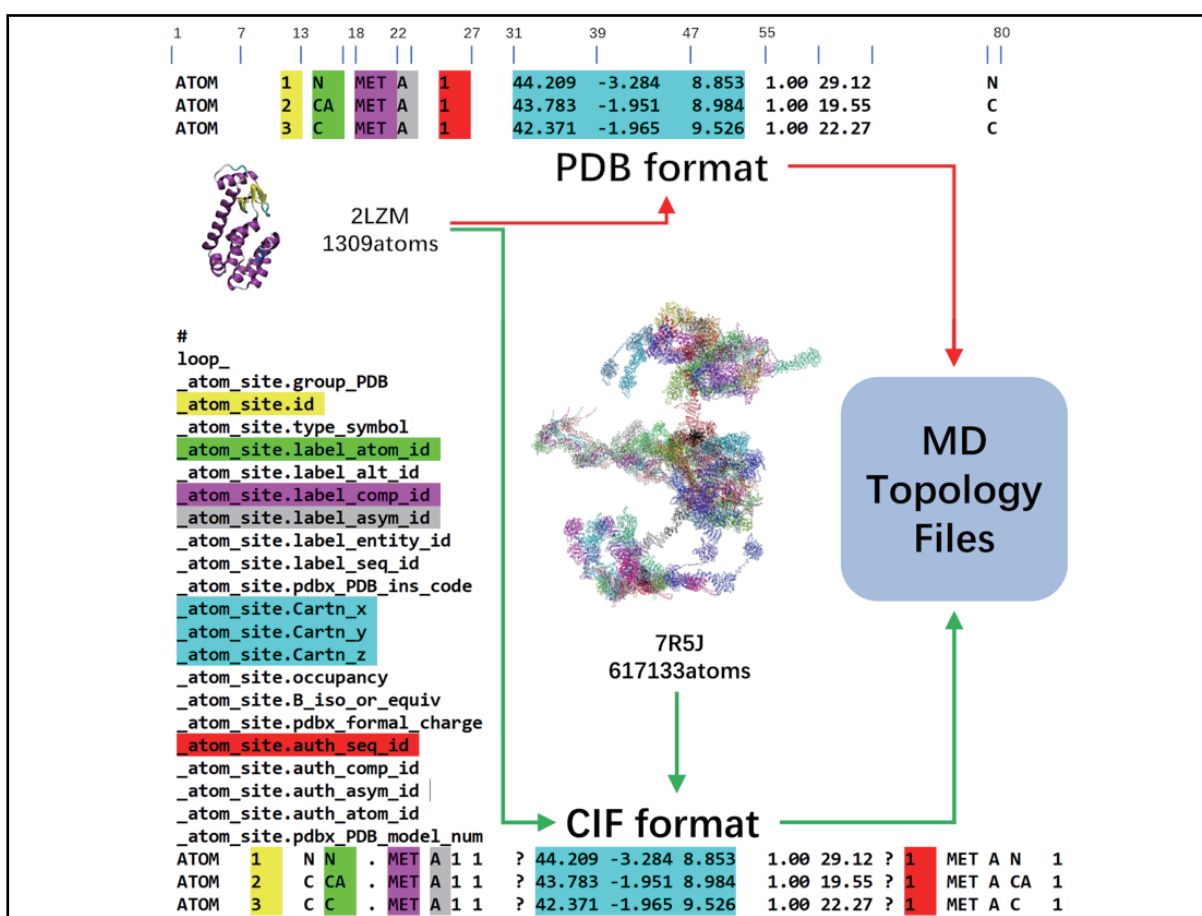
Hengyue Wang, and Zhiyong Zhang ✉

Department of Physics, University of Science and Technology of China, Hefei 230026, China

✉Correspondence: Zhiyong Zhang, E-mail: zzyzhang@ustc.edu.cn

Graphical abstract

The CIF file format of proteins can be directly used to generate topology files for molecular dynamics simulations.

Public summary

■ We modified the source code in one of the MD packages, GROMACS, which enables direct support of CIF files of proteins.

■ The modified program in GROMACS can read CIF files of proteins successfully and generate correct topology files.

■ This work simplifies the preprocessing of large protein complexes for MD simulations when only CIF files are available.

Supporting the CIF file format of proteins in molecular dynamics simulations

Hengyue Wang, and Zhiyong Zhang ✉

Department of Physics, University of Science and Technology of China, Hefei 230026, China

✉Correspondence: Zhiyong Zhang, E-mail: zzyzhang@ustc.edu.cn

Abstract: Molecular dynamics (MD) simulations can capture the dynamic behavior of proteins in full atomic detail and at very fine temporal resolution, so they have become an important tool in the study of protein dynamics. To date, several MD packages are widely used. An MD simulation starts from an initial structure that is generally taken from the Protein Data Bank (PDB). Until 2014, the PDB format was the standard file format for protein structures. However, there are certain intrinsic limitations in the PDB format, such as the storage of structural information in a fixed-width format, which is an issue for very large protein complexes. Therefore, the CIF (crystallographic information framework) format has been proposed, which is characterized by its superior expansibility. To our knowledge, the current mainstream MD packages support only the PDB format but do not support the CIF format directly. In this study, we modified the source code of one of the MD packages, GROMACS, which enables it to support CIF-formatted structure files as input and subsequently generate molecular topology files. This work simplifies the preprocessing of large protein complexes for MD simulations.

Keywords: Protein Data Bank; PDB format; CIF format; molecular dynamics simulation; GROMACS

CLC number: Q51   **Document code:** A

1 Introduction

Molecular dynamics (MD) simulations can predict how every atom in a protein will move over time based on a general model of physics governing interatomic interactions, thus revealing the positions of all of the atoms at femtosecond temporal resolution. These simulations can capture a wide variety of important biomolecular processes, such as conformational changes, ligand binding, and protein folding[1]. With the development of hardware and advanced algorithms, MD has become an important technique for studying the dynamic properties of proteins[2].

To date, several MD packages, such as GROMACS[3], AMBER[4], NAMD[5], and CHARMM[6], are available. To conduct an MD simulation, an initial protein structure, generally obtained from the Protein Data Bank (PDB), is needed. Taking GROMACS as an example, a preparatory step involves the use of "gmx pdb2gmx" to read atomic coordinates in a PDB format file; automatically determining chemical bonds, bond angles, and dihedral angles within the protein molecule; and subsequently generating a GROMACS-formatted coordinate file (gro), a molecular topology file (top), and one or more parameter files (itp) for the following processes.

The PDB is a widely used repository for protein structure data[7,8]. It provides two major file formats, namely, PDB and CIF (crystallographic information framework) formats. The PDB format, which originated in 1971, is characterized by the strict storage of protein structure data at fixed positions[9,10]. In a PDB file, the first six characters indicate the type of stored information, such as "HEADER", "ATOM", and

"HETATM". The "ATOM" and "HETATM" lines within a PDB file constitute the core content, encompassing details of all heavy atoms within a protein molecule. These data are tightly arranged within a single line. Despite the continued prevalence of the PDB format, it has intrinsic limitations. If the data content exceeds the predetermined limits, the PDB file format becomes invalid.

Presently, the upper limit of 99999 for atomic serial numbers in the PDB format has been surpassed in MD simulations because the MD software circumvents this issue by reading only the line numbers from the PDB file. In fact, there have been many MD simulations with more than millions of atoms[11,12]. However, another issue arises from the fact that the PDB format employs the "8.3f" format for storing atomic coordinates, with units in angstroms. Consequently, the PDB format can, at most, have storage coordinates ranging from −999.999 to 9999.999 Å. In recent years, with the development of high-resolution cryo-electron microscopy techniques, an increasing number of structures of large protein complexes have been solved[13]. Such a structure may contain a large number of atomic coordinates that would exceed the limit of the PDB format. Therefore, a new file format is needed to fix this problem.

The CIF format, introduced in 1990[14,15], has two fundamental distinctions from the PDB format. First, a CIF file is constructed by concatenating data blocks. Second, each data block comprises a label column followed by data columns. The positions of labels in the label column correspond to the positions of the data in the data columns, and the data are

separated by varying numbers of spaces[10]. The CIF file format offers unlimited expandability, demonstrated by the flexibility to augment the number of data blocks and label columns. Therefore, in 2014, the CIF format replaced the PDB format as the standard format in the PDB[16].

To the best of our knowledge, until now, the main MD packages lack direct support for the CIF format. One solution is to convert the CIF format to the PDB format[17], but this may be unsuitable for a large protein complex exceeding the coordinate limit of the PDB format.

As GROMACS is an open-source software, in this work, we modified its source code. The updated "gmx pdb2gmx" program facilitates direct reading of a CIF file to generate the gro, top, and itp files. This modification simplifies the preprocessing of large protein complexes for MD simulations.

2   Materials and methods

2.1   How GROMACS reads the PDB format

Normally, GROMACS creates topology files from a PDB file through the command "gmx pdb2gmx -f x.pdb -o x.gro -p x.top -i x.itp". In short, the gro file contains coordinate information in the gmx format, the top file contains information on the molecular topology of the protein, and the itp file contains supplementary information on the topology. Together, they constitute the topology files. Our goal is for GROMACS to run the command "gmx pdb2gmx -f x.cif -o x.gro -p x.top -i x.itp" and generate the correct topology files. To enable direct CIF file support, the source code for how GROMACS supports the PDB file must be found. All related functions and their paths of the source files are listed in Table 1.

As shown in Fig. 1, in the process of the command "gmx pdb2gmx", the function "pdb2gmx::run" obtains the system path of the PDB file from the user's input. Then, the path is transmitted through the functions "read_pdball", "readConfAndAtoms", "get_stx_coordnum" and "read_stx_conf".

GROMACS not only supports PDB files but also accepts other structure files, such as the gro and g96 formats. To distinguish different file types, GROMACS has a function called "fn2ftp", which compares the filename extension with the built-in file type list. Therefore, "get_stx_coordnum" and "read_stx_conf" both use "fn2ftp" to identify file types and call corresponding functions. For a PDB file, "get_stx_coordnum" calls the function "get_pdb_coordnum" to acquire the atom number. Then, "get_stx_coordnum" initializes the built-in structure, which will store all the PDB information. Meanwhile, the PDB file's system path is transmitted through "read_stx_conf", "gmx_pdb_read_conf", and the function "read_pdbfile". It is worth noting that the PDB file has not yet been analyzed. The functions "read_pdbfile" and

"read_atom" will do that.

"read_pdbfile" analyzes the PDB file by the first six characters in each line. The most important "ATOM" and "HET-ATM" lines use the function "read_atom" with the whole line as input for further processing. In "read_atom", the whole line is strictly split by bits due to the rule of PDB files (Table 2). Then, each data part is stored in the corresponding variable contained by the built-in structure initialized in "get_stx_coordnum".

2.2   Comparison between the PDB and CIF formats

Table 2 illustrates the corresponding relationship between the PDB and the CIF file through an example (PDB ID 2LZM[18]). All information in the PDB file is also contained in the CIF file. Our idea is that if we store the corresponding CIF file's data into those variables in the function "read_atom", GROMACS will recognize them as the PDB input and continue its work. Therefore, we could create topology files directly from the CIF file by modifying the data-reading process in GROMACS.

2.3   Direct support of the CIF format in GROMACS

First, we added the CIF file type support to the source files "filetypes.cpp" and "filetypes.h". Then, we modified "get_stx_coordnum" and "read_stx_conf" to correctly identify the CIF file type and call the corresponding CIF functions. Next, we added the CIF functions "get_cif_coordnum", "gmx_cif_read_conf", "read_ciffile" and "read_atoms_cif" to the source file "pdbio.cpp". These new functions are parallel with the PDB functions, as shown in Fig. 1 and Table 3.

A CIF file is divided into several information boxes by "#". We only need the box containing PDB's "ATOM" and "HET-ATM" line information, which could be easily located by its unique label "_atom_site.group_PDB". To analyze the CIF file with the function "read_ciffile", we first searched the whole CIF file to find the unique label. Then, we checked all the labels and saved only the locations of the labels shown in Table 2. Finally, in the function "read_atoms_cif", we divided the line by whitespace. Because the location of the data and label is the same, we could store the data by its location in exactly the same variable as in the function "read_atom". For more details, check the source codes in the Data availability section.

3   Results and discussion

To verify the accuracy of the gro, top and itp files generated directly from the CIF file, we compared them with topology files generated from the PDB file.

For simplicity, we selected a relatively small protein, the bacteriophage T4 lysozyme (T4L, PDB ID 2LZM[18]). It is a

Table 1. PDB functions and the paths of source files.

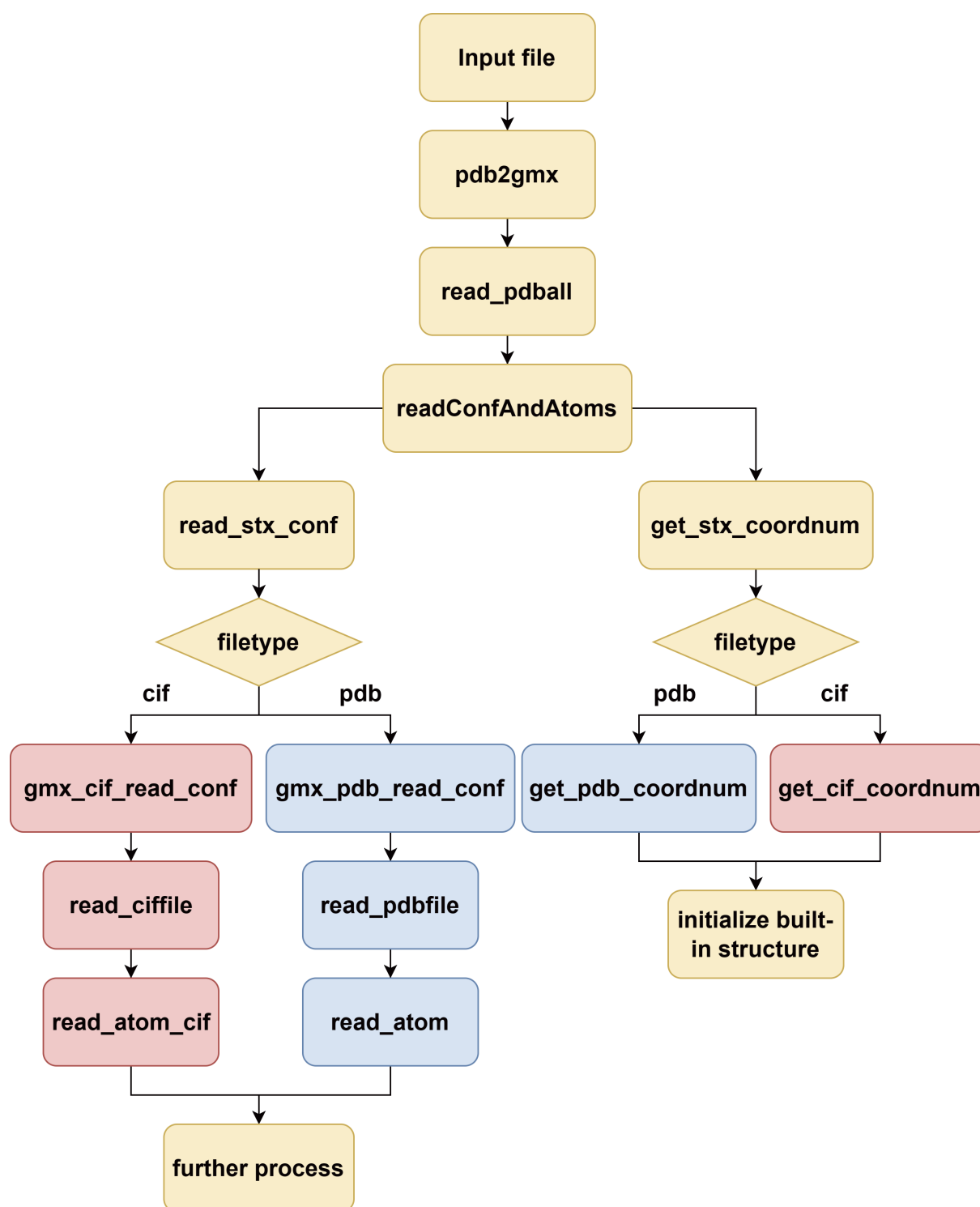| Function	Source file
pdb2gmx, read_pdball	..\src\gromacs\gmxpreprocess\pdb2gmx.cpp
readConfAndAtoms, get_stx_coordnum, read_stx_conf	..\src\gromacs\fileio\confio.cpp
fn2ftp, built-in filetype list	..\src\gromacs\fileio\filetypes.cpp, ..\src\gromacs\fileio\filetypes.h
gmx_pdb_read_conf, read_pdbfile, read_atom	..\src\gromacs\fileio\pdbio.cpp

Fig. 1. Flowchart of how GROMACS supports PDB and CIF files. Yellow: the same functions in processing PDB and CIF, red: PDB functions, and blue: CIF functions.

single-chain protein with only 164 amino acid residues and 1309 heavy atoms (Fig. 2a). The structure database provides both the PDB and the CIF files of T4L. Two groups of topology files are generated by the modified GROMACS using the PDB and the CIF files as inputs.

As shown in Fig. 3, the screen output from the CIF file (Fig. 3a) is exactly the same as that from the PDB file (Fig. 3b). Then, we further calculated the md5 value of the data

section in the topology files. The results from the PDB and the CIF file are identical, so we prove the correctness of the CIF-generated topology files for small proteins.

For complicated cases, we chose the dilated state of the human nuclear pore complex (PDB ID 7R5J[19]). It contains 25 different nucleoporins and 101 peptide chains with 617133 heavy atoms in total (Fig. 2b). This large protein complex exceeds the limit of the PDB format, so the structure database

Table 2. Corresponding relationship between the PDB and CIF formats. The same data are stored in both the fixed column in the PDB file and the label in the CIF file.

Data meaning	Example	Column in PDB	Label in CIF
Line type	ATOM	1–6	_atom_site.group_PDB
Atom serial number	986	7–11	_atom_site.id
Atom name	CD1	13–16	_atom_site.label_atom_id
Residue name	TRP	18–20	_atom_site.label_comp_id
Chain identifier	A	22	_atom_site.label_asym_id
Residue sequence number	126	23–26	_atom_site.auth_seq_id
X coordinates	31.655	31–38	_atom_site.Cartn_x
Y coordinates	−2.030	39–46	_atom_site.Cartn_y
Z coordinates	−6.788	47–54	_atom_site.Cartn_z
Occupancy	1.00	55–60	_atom_site.occupancy
Temperature factor	23.24	61–66	_atom_site.B_iso_or_equiv
Element symbol	C	77–78	_atom_site.type_symbol

Table 3. Usage of the PDB and CIF functions.

| PDB function	CIF function	Usage
get_pdb_coordnum	get_cif_coordnum	Get number of all atoms
gmx_pdb_read_conf	gmx_cif_read_conf	Open structure file
read_pdbfile	read_ciffile	Analyze structure file, and separate out lines containing atom information for further usage
read_atom	read_atom_cif	Analyze atom information line, and store useful part in corresponding variable
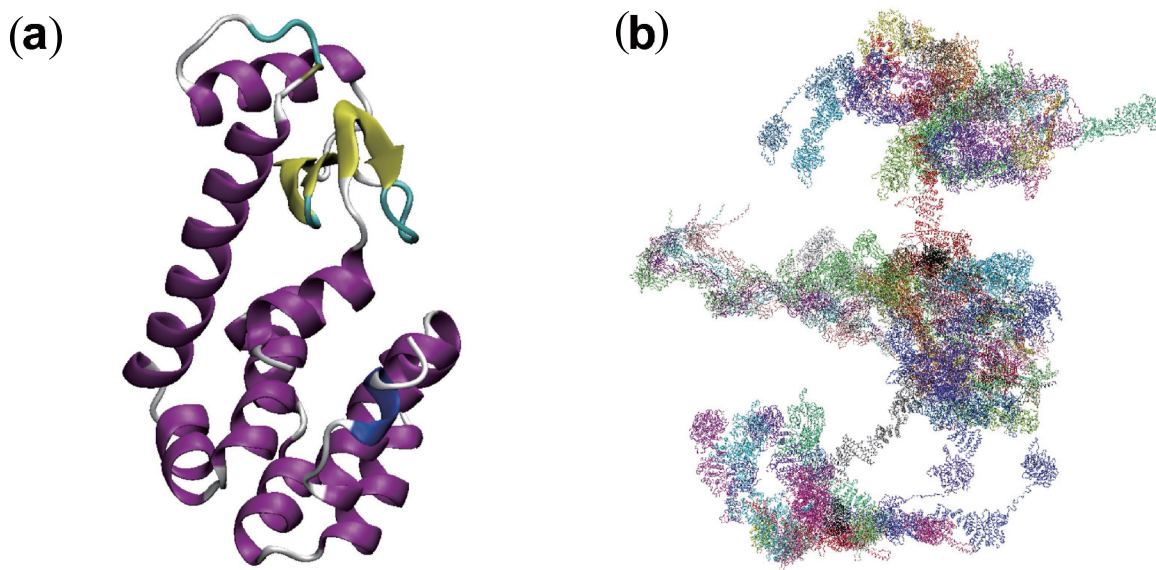
Fig. 2. The systems used to test the modified GROMACS. (a) The bacteriophage T4 lysozyme (2LZM). (b) The dilated human nuclear pore complex (7R5J). Since the whole complex has C8 symmetry, only one-eighth of the structure is included in the CIF file.

provides only a CIF file and PDB format-like files. The latter are individual PDB files, each including parts of the data to circumvent limitations.

The modified GROMACS can successfully generate topology files through 7R5J.cif. As shown in Fig. 3c, all 101 peptide chains were identified, and 617273 missing hydrogen atoms were correctly added to the 78030 amino acid residues. Next, we selected 5 different peptide chains in 7R5J. The same data in PDB format-like files are merged into one PDB file as the input of the modified GROMACS to generate topology files. As in the case of 2LZM, we separated the two groups of topology file data and proved that they were identical through the md5 values.

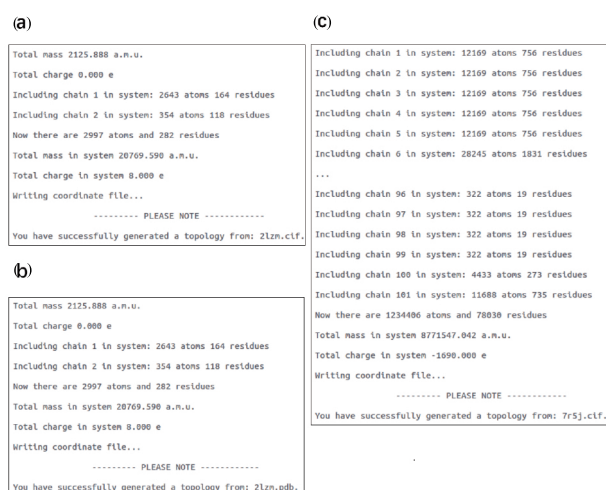The results indicate that the modified GROMACS could generate correct topology files from the CIF files.

Fig. 3. Screen output from the modified GROMACS. (a) Output when generating topology files from 2LZM.cif. (b) Output when generating topology files from 2LZM.pdb. (c) Output when generating topology files from 7R5J.cif.

4　Conclusions

To enable direct support for the CIF file format in MD simulations of proteins, in this work, we modified the open-source software GROMACS to allow for the utilization of CIF files as input to generate topology files of proteins. To validate the accuracy of the generated topology files, we selected two systems: a relatively small protein (bacteriophage T4 lysozyme) and a large protein complex (asymmetric unit of the dilated human nuclear pore complex). A comparison between topology files generated from the CIF format and those generated from the PDB format was conducted, thereby confirming their correctness.

For relatively small protein systems, the Protein Data Bank offers protein structure files in two formats, PDB and CIF, both containing the same information. Generally, PDB-format files serve as initial structures for MD simulations. In this case, our work does not make an additional contribution because CIF-format files are not used. However, for very large protein complexes lacking PDB-format files, our contribution streamlines the preprocessing phase of their MD simulations.

For other open-source MD packages, we may also modify the source code to support the CIF format directly. However, another solution is to generate topology files via modified GROMACS and then use tools such as amb2gro_top_gro.py[20] to convert the GROMACS topology files into other MD package formats.

Data availability

The installation guidance, source codes and examples are available at https://github.com/zyzhangGroup/Gromacs-CIF.

Conflict of interest

The authors declare that they have no conflict of interest.

Preprint statement

Research presented in this article was posted on a preprint server prior to publication in JUSTC. The corresponding pre-print article can be found at https://doi.org/10.1101/2023.09.01.555884.

Biographies

Hengyue Wang　is currently a graduate student in the School of Physics, University of Science and Technology of China, under the supervision of Prof. Zhiyong Zhang. His research mainly focuses on computer simulations of large biomolecular complex assemblies.

Zhiyong Zhang　is currently a Professor in the Department of Physics, University of Science and Technology of China (USTC). He received his Ph.D. degree in Biochemistry and Molecular Biology from USTC in 2003. His research interests include method development on multiscale modeling and integrative modeling of large biomolecular complexes.

References

[1] Hospital A, Goñi J R, Orozco M, et al. Molecular dynamics simulations: advances and applications. *Advances and Applications in Bioinformatics and Chemistry*, **2015**, *8*: 37–47.

[2] Hollingsworth S A, Dror R O. Molecular dynamics simulation for all. *Neuron*, **2018**, *99* (6): 1129–1143.

[3] Van Der Spoel D, Lindahl E, Hess B, et al. GROMACS: fast, flexible, and free. *Journal of Computational Chemistry*, **2005**, *26* (16): 1701–1718.

[4] Case D A, Cheatham III T E, Darden T, et al. The Amber biomolecular simulation programs. *Journal of Computational Chemistry*, **2005**, *26* (16): 1668–1688.

[5] Phillips J C, Hardy D J, Maia J D C, et al. Scalable molecular dynamics on CPU and GPU architectures with NAMD. *The Journal of Chemical Physics*, **2020**, *153* (4): 044130.

[6] Brooks B R, Brooks III C L, Mackerell Jr A D, et al. CHARMM: the biomolecular simulation program. *Journal of Computational Chemistry*, **2009**, *30* (10): 1545–1614.

[7] Berman H, Henrick K, Nakamura H. Announcing the worldwide Protein Data Bank. *Nature Structural & Molecular Biology*, **2003**, *10* (12): 980.

[8] wwPDB consortium. Protein Data Bank: the single global archive for 3D macromolecular structure data. *Nucleic Acids Research*, **2019**, *47* (D1): D520–D528.

[9] Callaway J, Cummings M, Deroski B, et al. Protein Data Bank contents guide: Atomic coordinate entry format description. Upton: Brookhaven National Laboratory, **1996**.

[10] Westbrook J D, Fitzgerald P M D. The PDB Format, mmCIF Formats, and Other Data Formats. In: Bourne P E, Weissig H, editors. Structural Bioinformatics. Hoboken: John Wiley & Sons, Inc. , **2003**.

[11] Zhao G, Perilla J R, Yufenyuy E L, et al. Mature HIV-1 capsid structure by cryo-electron microscopy and all-atom molecular dynamics. *Nature*, **2013**, *497* (7451): 643–646.

[12] Khalid S, Brandner A F, Juraschko N, et al. Computational microbiology of bacteria: Advancements in molecular dynamics simulations. *Structure*, **2023**, *31* (11): 1320–1327.

[13] Chua E Y D, Mendez J H, Rapp M, et al. Better, faster, cheaper: recent advances in cryo–electron microscopy. *Annual Review of Biochemistry*, **2022**, *91*: 1–32.

[14] Fitzgerald P M D, Berman H, Bourne P, et al. The mmCIF dictionary: community review and final approval. *Acta Crystallographica Section A*, **1996**, *52*: C575.

[15] Hall S R, Allen F H, Brown I D. The crystallographic information file (CIF): a new standard archive file for crystallography. *Acta Crystallographica Section A*, **1991**, *47*(6): 655–685.

[16] Berman H M, Kleywegt G J, Nakamura H, et al. The Protein Data Bank archive as an open data resource. *Journal of Computer-Aided Molecular Design*, **2014**, *28* (10): 1009–1014.

[17] van Ginkel G, Pravda L, Dana J M, et al. PDBeCIF: an open-source mmCIF/CIF parsing and processing package. *BMC Bioinformatics*, **2021**, *22* (1): 383.

[18] Weaver L H, Matthews B W. Structure of bacteriophage T4 lysozyme refined at 1.7 Å resolution. *Journal of Molecular Biology*, **1987**, *193* (1): 189–199.

[19] Mosalaganti S, Obarska-Kosinska A, Siggel M, et al. AI-based structure prediction empowers integrative structural analysis of human nuclear pores. *Science*, **2022**, *376* (6598): eabm9506.

[20] Case D A, Aktulga H M, Belfon K A A, et al. Amber 2021. San Francisco: University of California, **2021**.