Comprehensive Database of Circular Permutations: Systematic Detection and Analysis Using Deep Learning

Yue Hu[1#], Bin Huang[2#*](co-first and corresponding author)*

[1]College of Bioengineering, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250300, Shandong, China

[2] School of Life Sciences, Yunnan Normal University, Kunming 650500, Yunnan, China

#Co-first author

**Corresponding Author：*

Yue Hu, E-mail: huyue@qlu.edu.cn

Bin Huang, E-mail: huangbin@ynnu.edu.cn

Abstract

This study presents a comprehensive approach to detect circular permutations in Protein Data Bank up to date (PDB, 287081 proteins which sequence length is under 800 up to 20240101). We systematically analyzed the Protein Data Bank (PDB) to identify circular permutations, leveraging FoldSeek and MMseqs2 for structural and sequence similarity searches. The 143756535 candidate pairs were filtered by some threshold for corresponding analysis. TM-align, icarus or plmCP was used to align protein structures and refine detection accuracy, while facilitated the precise identification of circular permutations. Finally, we got 20801 candidate circular permutation pairs and 3351 circular permutation proteins (https://github.com/YueHuLab/Circular-permutation-in-PDB). Our methodology provides a robust framework for uncovering circular permutations in protein databases, enhancing our understanding of protein structural variations and evolutionary adaptations.

Introduction

Circular permutation[1] is a structural rearrangement in proteins where the N-terminal and C-terminal ends are swapped, and the protein sequence is "circularized" by connecting the original termini. This structural variant often retains the same overall fold and function but may exhibit differences in stability, activity, or regulation. These advances highlight the versatility of circular permutation as a tool for understanding protein structure and function, as well as its potential for creating novel biomolecules with practical applications.

Detecting circular permutations in proteins is an important task in bioinformatics and structural biology. Several computational tools have been developed to identify and analyse these rearrangements in protein sequences and structures. These tools[2, 3], developed by various researchers, provide robust methods for detecting and analysing circular permutations in proteins. They range from sequence-based approaches (like seqCP[4] and CPred[5]) to structure-based tools (like TMalign_CP[6]) and databases (like Circular Permutation Database) that integrate both types of data. These tools are essential for studying protein evolution, understanding structural diversity, and designing new proteins with desired functions.

The Circular Permutation Database[7] (CPDB) is a specialized resource dedicated to the study of circular permutations in proteins. This database serves as a crucial tool for the scientific community, providing insights into the structural and functional implications of circular permutations in proteins. The database not only facilitates research in this specialized area but also helps in advancing broader understanding in areas like protein engineering, structural biology, and evolutionary biology. However, the CPDB was published in 2009 and was a derivative database from the Protein Data Bank[8]. At its inception, it contained a curated set of circular permutation data but was limited by its relatively small size and the scope of available structural information at the time. Shortcomings include its static dataset, which may not

encompass more recent discoveries or integrate with newer computational methods, potentially limiting its usefulness for current research needs.

Analysing pairwise structural similarities for the entire Protein Data Bank (PDB) presents significant challenges due to the sheer volume of data and computational complexity. With over 300,000 protein structures, performing comparisons for every possible pair is immensely resource-intensive. For example, tools like TM-align, while effective for individual pairwise comparisons, become impractical for large-scale analysis due to their high computational and memory demands. The process requires calculating and storing similarity scores for every pair, which quickly becomes unmanageable as the number of structures increases. Foldseek[9], on the other hand, is designed to handle large-scale structural comparisons more efficiently, as it can perform comparisons in a more parallelized and optimized manner. However, even Foldseek faces limitations related to processing power and memory, and may require sophisticated infrastructure to handle the massive dataset of the PDB. Both approaches thus struggle with the massive computational load and memory requirements involved in analyzing structural similarities across such a vast database.

In this article, we combine these advanced tools to delve into circular permutation in the PDB. Foldseek, integrated with protein language models, facilitates database creation for any protein sequence. It transforms protein sequences into one-dimensional information with structural insights. After screening, we use pairwise structural comparison tools, such as TM-align, icarus[10], and plmCP[11], for further filtering and validation. We have simplified the classical process of discovering circular permutations, enabling the identification of circular permutations in the database with a single comparison of two databases (**Figure 1(C)**). Ultimately, we completed a systematic exploration of circular permutations in the PDB up to January 1, 2024, proving the effectiveness of the entire algorithm and providing the latest circular permutation database for the scientific community.

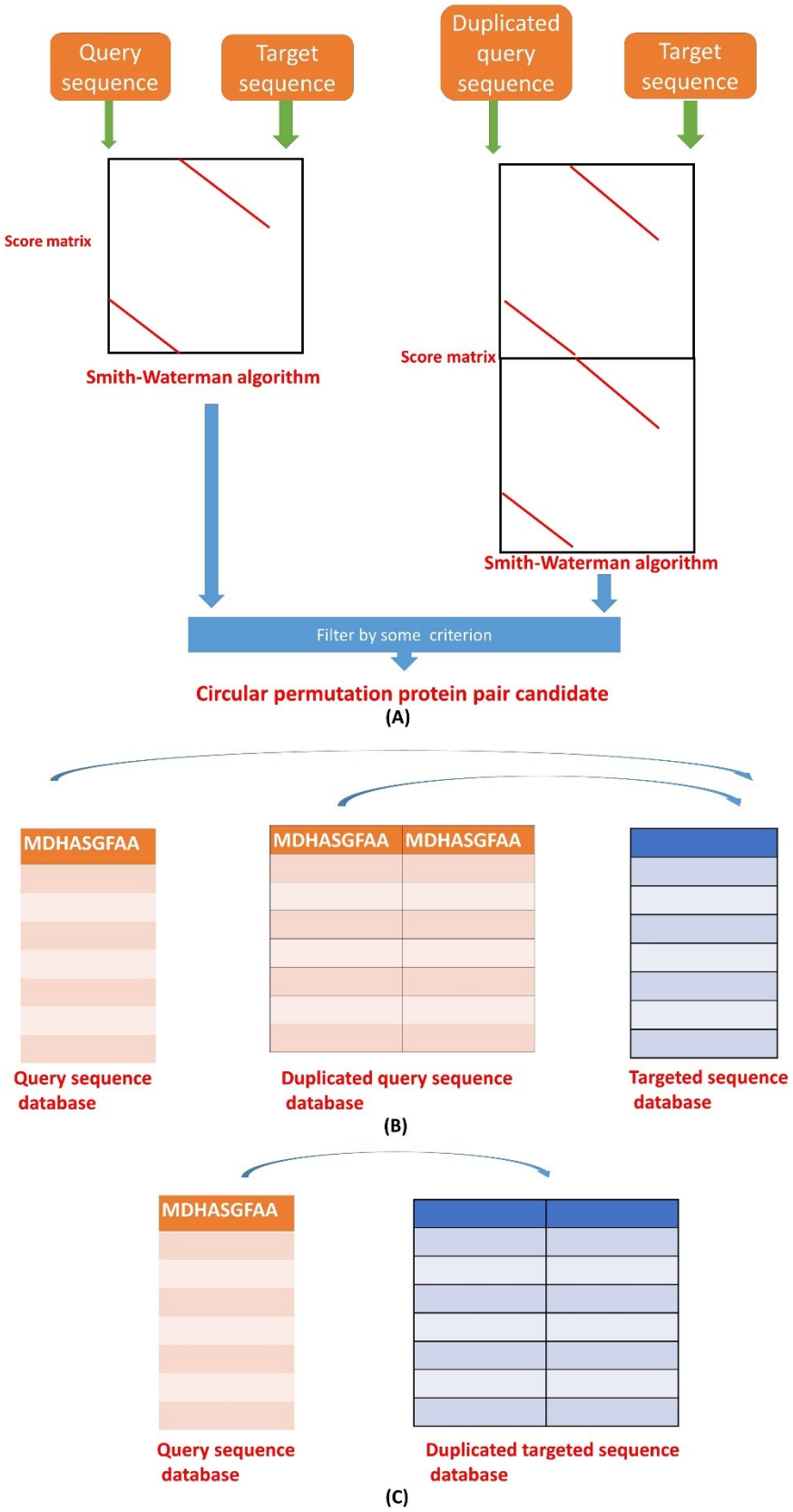

Figure 1. The overall algorithm of database search for the detection of circular permutation

Method

Inspired by classical circular permutation detection methods (**Figure 1(A)&(B)**), we have developed a more resource-efficient algorithm that requires only a single systematic search (**Figure 1(C)**). The drawback of this method is that it may easily identify identical or similar structures, which can also be viewed as a form of circular permutation, where the N- and C-termini of a protein are connected and then cleaved at similar positions. Fortunately, such cases can be easily checked and filtered based on alignment information (such as positions). After filtering, we use low-throughput but more accurate structural and sequence alignment tools for confirmation.

First, we use Foldseek and MMseqs[12, 13] to build and search the database. Starting with the sequences, we use the ProstT5 protein language model module of Foldseek to construct both the query sequence database and the duplicated target database. We then perform searches using Foldseek and convert the results into a readable format with MMseqs. Foldseek is a fast and accurate tool designed for comparing protein structures. It is particularly useful for large-scale structural comparisons, such as identifying structural homologs in protein structure databases. Foldseek is known for its ability to handle large datasets efficiently while providing high sensitivity in detecting structural similarities. MMseqs2 is a fast, ultra-sensitive tool for sequence searching, clustering, and annotation. It is designed to handle large-scale sequence datasets, offering a balance between speed and sensitivity. MMseqs2 is widely used for tasks that involve searching large sequence databases, clustering sequences, or annotating sequences with functional information. Subsequently, we use the alignment results to filter and select data for further analysis. Next, we use TM-align, Icarus, or plmCP to validate the filtered results. We select TM-align with the CP parameter, and consider those results with an increased TM score as the final results.TM-align is a structural alignment tool that compares the 3D structures of proteins. It is widely recognized for its ability to align protein structures and assess their similarity using the TM-score[14], which is a measure of structural similarity that accounts for differences in protein size and alignment length.

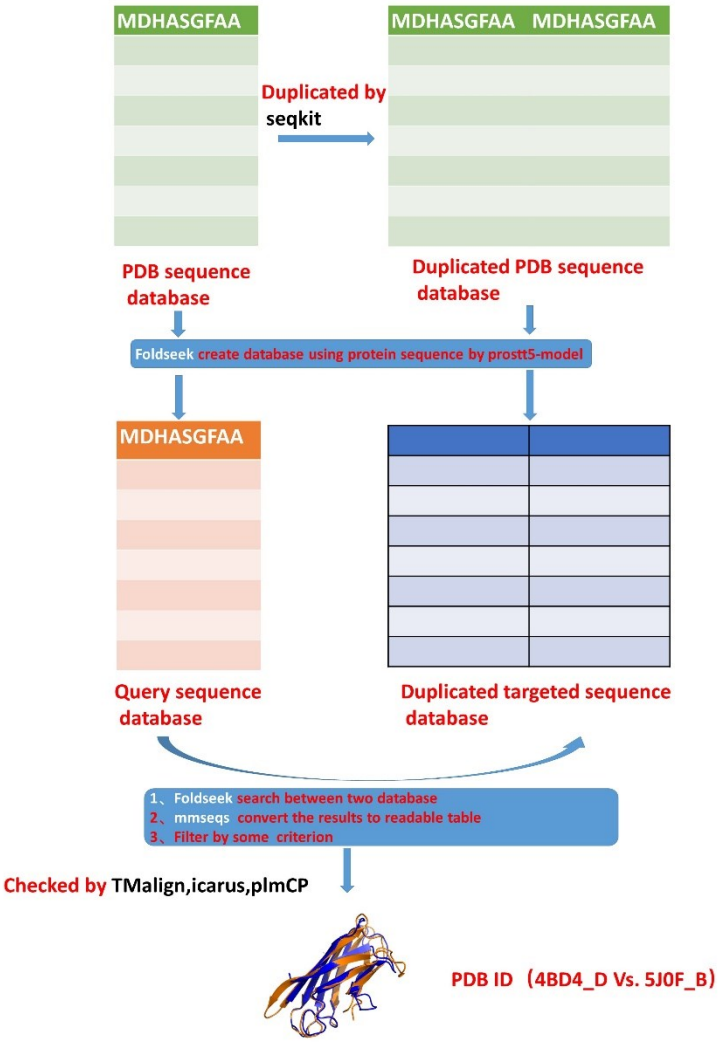

Figure 2. The details of algorithm of the circular permutation detect for database

Results and Discussion

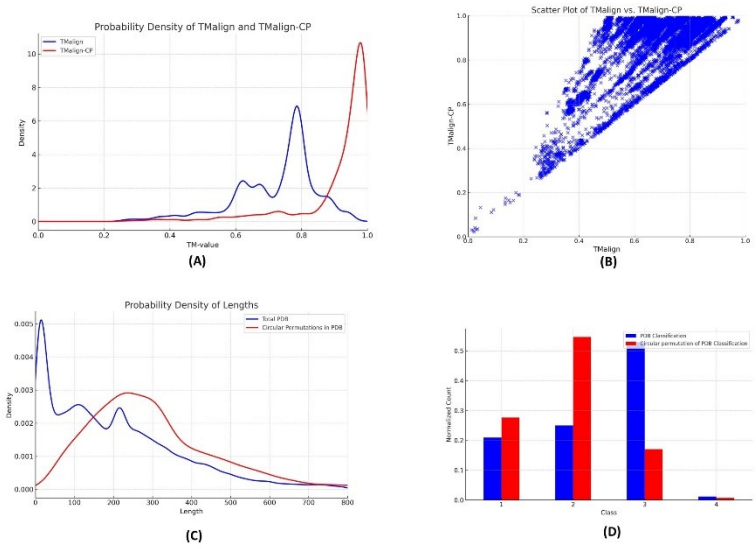

Figure 3. Statistic of the circular permutation proteins

We have obtained 20801 pairs of circular permutation proteins, totaling 3351 proteins. We have made this data available on GitHub for public download (https://github.com/YueHuLab/Circular-permutation-in-PDB). Upon further examination of this data, we first looked at the protein pairs and compared the TM-scores calculated using TM-align without the circular permutation parameter and with the circular permutation parameter (**Figure 3(A)&(B)**). We then plotted probability density plots and scatter plots, which clearly show significant differences between the two. This indicates that the circular permutation we identified is quite prominent. We also compared the lengths of the circular permutation proteins we obtained with the lengths of all proteins in the PDB database that are shorter than 800 residues (**Figure 3(C)**). We found significant differences between the two, with fewer circular permutation proteins being either much shorter or much longer. The lengths of the circular permutation proteins follow a normal distribution, centered between 200 and 300 residues. Finally, we examined the proportion of the four CATH[15] classifications (Mainly Alpha (α): Mainly Beta (β); Alpha-Beta (α/β); Few Secondary Structures:) in the circular permutation proteins we obtained and compared it with the proportion of the four CATH classifications in the PDB database using histograms (**Figure 3(D)**). Observations and Fisher's exact test revealed significant differences. Circular permutations are predominantly

concentrated in Mainly Alpha (α) and Mainly Beta (β), with very few in the mixed Alpha-Beta (α/β).

We utilized Foldseek in combination with AI models to generate a one-dimensional string representation of protein structures from sequences. This approach is no longer limited to existing protein structures, as it leverages protein language models to effectively provide structural information, ensuring accuracy. We also improved traditional circular permutation methods, simplifying the process of library-to-library searches and completing the task of searching the PDB database. After filtering and further structural alignment, we ultimately identified a substantial number of circular permutation pairs and proteins. These findings were statistically analyzed and visualized, laying a solid foundation for discovering circular permutations in the protein universe.

Looking ahead, this work opens up exciting possibilities for exploring the vast diversity of protein structures, functions, and evolutionary relationships. By combining deep learning and protein language models[16, 17], we can continue to refine our understanding of complex protein rearrangements, potentially uncovering new insights into protein biology and contributing to advancements in bioinformatics and computational biology. This approach could lead to more efficient and accurate detection of circular permutations, even in cases with low sequence similarity or subtle rearrangements, paving the way for new discoveries in the field.

Reference

1. Lindqvist, Y. & Schneider, G. Circular permutations of natural protein sequences: structural evidence. *Current opinion in structural biology* **7**, 422-427 (1997).
2. Lo, W.-C. & Lyu, P.-C. CPSARST: an efficient circular permutation search tool applied to the detection of novel protein structural relationships. *Genome Biol* **9**, R11 (2008).
3. Carpentier, M., Brouillet, S. & Pothier, J. YAKUSA: a fast structural database scanning method. *Proteins: Structure, Function, and Bioinformatics* **61**, 137-151 (2005).
4. Chen, C.-C., Huang, Y.-W., Huang, H.-C., Lo, W.-C. & Lyu, P.-C. SeqCP: A sequence-based algorithm for searching circularly permuted proteins. *Computational and Structural Biotechnology Journal* **21**, 185-201 (2023).
5. Lo, W.-C. et al. CPred: a web server for predicting viable circular permutations in proteins. *Nucleic acids research* **40**, W232-W237 (2012).

6. Zhang, Y. & Skolnick, J. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic acids research* **33**, 2302-2309 (2005).
7. Lo, W.-C., Lee, C.-C., Lee, C.-Y. & Lyu, P.-C. CPDB: a database of circular permutation in proteins. *Nucleic acids research* **37**, D328-D332 (2009).
8. Bank, P.D. Protein data bank. *Nature New Biol* **233**, 10-1038 (1971).
9. van Kempen, M. et al. Foldseek: fast and accurate protein structure search. *Biorxiv*, 2022.2002. 2007.479398 (2022).
10. Cretin, G., Périn, C., Zimmermann, N., Galochkina, T. & Gelly, J.-C. ICARUS: flexible protein structural alignment based on Protein Units. *Bioinformatics* **39**, btad459 (2023).
11. Hu, Y. & Huang, B. Detection of circular permutations by Protein Language Models. *arXiv preprint arXiv:2404.15087* (2024).
12. Hauser, M., Steinegger, M. & Söding, J. MMseqs software suite for fast and deep clustering and searching of large protein sequence sets. *Bioinformatics* **32**, 1323-1330 (2016).
13. Steinegger, M. & Söding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology* **35**, 1026-1028 (2017).
14. Xu, J. & Zhang, Y. How significant is a protein structure similarity with TM-score= 0.5? *Bioinformatics* **26**, 889-895 (2010).
15. Orengo, C.A. et al. CATH–a hierarchic classification of protein domain structures. *Structure* **5**, 1093-1109 (1997).
16. Llinares-López, F., Berthet, Q., Blondel, M., Teboul, O. & Vert, J.-P. Deep embedding and alignment of protein sequences. *Nature methods* **20**, 104-111 (2023).
17. Kaminski, K., Ludwiczak, J., Pawlicki, K., Alva, V. & Dunin-Horkawicz, S. pLM-BLAST: distant homology detection based on direct comparison of sequence representations from protein language models. *Bioinformatics* **39**, btad579 (2023).

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 32301061 to B.H.) and was supported by the Yunnan Fundamental Research Projects (Grant No. 202401CF070034 to B.H.) and was supported by Shandong Provincial Natural Science Foundation Committee (Grant No. ZR2016HB54 Y.H.) and was supported by Shandong Provincial Key Laboratory of Microbial Engineering (SME).

Author contributions

Hu Yue and Huang Bin secured funding and designed the algorithm, who are also guarantors of this work and, as such, had full access to all of the data in the study and takes responsibility for the integrity of the data and the accuracy of the data analysis; Hu Yue, Huang Bin performed computation and analyzed the data.

Competing interests

All authors declare no competing interests.