Bioinformatics Advances, 2023, **00**, vbad158
https://doi.org/10.1093/bioadv/vbad158
Advance Access Publication Date: 2 November 2023
Application Note

Structural bioinformatics

pyScoMotif: discovery of similar 3D structural motifs across proteins

Gabriel Cia[1,2]*, Jean Kwasigroch[1], Basile Stamatopoulos[3], Marianne Rooman[1,2,*,†], Fabrizio Pucci [1,2,*,†]

[1]Computational Biology and Bioinformatics, Université Libre de Bruxelles, Brussels, 1050, Belgium
[2]Interuniversity Institute of Bioinformatics in Brussels, Triomflaan, Brussels,1050, Belgium
[3]Laboratory of Clinical Cell Therapy, Jules Bordet Institute, Université Libre de Bruxelles, Brussels, 1070, Belgium

*Corresponding authors. Computational Biology and Bioinformatics, Université Libre de Bruxelles, Avenue F. Roosevelt 50, Brussels 1050, Belgium. E-mails: marianne.rooman@ulb.be (M.R.), fabrizio.pucci@ulb.be (F.P.), and gabriel.cia@ulb.be (G.C.)

†Equal contribution.

Associate Editor: Franca Fraternali

Abstract

Motivation: The fast and accurate detection of similar geometrical arrangements of protein residues, known as 3D structural motifs, is highly relevant for many applications such as binding region and catalytic site detection, drug discovery and structure conservation analyses. With the recent publication of new protein structure prediction methods, the number of available protein structures is exploding, which makes efficient and easy-to-use tools for identifying 3D structural motifs essential.

Results: We present an open-source Python package that enables the search for both exact and mutated motifs with position-specific residue substitutions. The tool is efficient, flexible, accurate, and suitable to run both on computer clusters and personal laptops. Two successful applications of pyScoMotif for catalytic site identification are showcased.

Availability and implementation: The pyScoMotif package can be installed from the PyPI repository and is also available at https://github.com/3BioCompBio/pyScoMotif. It is free to use for non-commercial purposes.

1 Introduction

The function and biophysical properties of structured proteins are determined by the conformation adopted by their amino acid sequence, and in particular by the presence of specific local geometrical arrangements of subsets of residues. These motifs generally consist of few residues and typically correspond to binding regions or catalytic sites. They are known as 3D structural motifs and are generally conserved in homologous proteins across a wide range of organisms (Fraser *et al.* 2002, Ribeiro *et al.* 2020). A variety of *in silico* tools to detect these motifs have been developed in the past decades, including brute-force methods (Ananthalakshmi *et al.* 2005, Debret *et al.* 2009), geometric hash-based techniques (Pennec and Ayache 1998, Moll *et al.* 2010), and graph-based approaches (Spriggs *et al.* 2003, Konc and Janežič 2010, Nadzirin *et al.* 2012, Kirshner *et al.* 2013). Recently, the Protein Data Bank (PDB) (Berman *et al.* 2000) made significant progress in this field by developing a novel and scalable structural motif search algorithm able to identify protein structures carrying a 3D motif that is similar to the queried motif (Bittrich *et al.* 2020). The algorithm, which is inspired by the inverted index approach used in search engines, relies on the indexing of all the pairs of residues in a set of target protein structures. Once the index has been built, the search for similar motifs across the set of structures can be performed rapidly by simply decomposing the query motif into a combination of residue pairs. Importantly, this approach does not require any prefiltering or clustering of proteins at either the sequence or structure level, unlike some other methods (Ananthalakshmi *et al.* 2005, Konc and Janežič 2010), thus enabling the discovery of 3D motifs even between evolutionary distant proteins.

The large-scale search for 3D structural motifs has recently attracted new interest thanks to the advances in protein structure prediction (Varadi *et al.* 2022). Despite the availability and the good performance of the PDB's motif search algorithm, there is currently a need to develop tools that are efficient, flexible, and easy to use and to install.

Here, we present a Python package that implements a 3D structural motif search algorithm that is broadly inspired by the PDB algorithm Bittrich *et al.* (2020) with a series of improvements both in terms of search flexibility and user-friendliness.

2 Implementation

The pyScoMotif algorithm can be separated into two steps: indexing and motif search. We briefly describe the key elements of these two steps. Further details on the design and optimization choices as well as on the differences with respect

Received: May 9, 2023; **Revised:** October 12, 2023; **Editorial Decision:** October 28, 2023; **Accepted:** November 1, 2023

to the PDB algorithm (Bittrich *et al.* 2020) are given in Supplementary Material. A tutorial showing how to use pyScoMotif is available in our GitHub repository https://github.com/3BioCompBio/pyScoMotif.

2.1 Indexing

The objective of the indexing step is to extract and characterize the geometrical arrangement of all the residue pairs in a given set of protein structures, as shown in Fig. 1. The resulting residue pair index is organized in such a way that searching the set of structures for all the occurrences of a residue pair with a specific geometry is rapid.

More precisely, given a set of protein structures, the indexing step generates a lookup table for each of the 210 possible residue pairs (i.e. Ala-Ala, Ala-Cys, …, Tyr-Tyr), with information about the geometrical arrangement of each occurrence of that residue pair. Three structural descriptors are used to yield a rotation-invariant geometrical description: the distance between the two C_α atoms; the distance between the two side chain centroids, noted C_μ, calculated as the average of the coordinates of all the heavy side chain atoms; and the angle between the vectors $\overrightarrow{R} = \overrightarrow{C}_\alpha - \overrightarrow{C}_\mu$ of the two residues. For the special case of Gly that has no heavy side chain atoms, we define $C_\mu = C_\alpha$ and $\overrightarrow{R} = \overrightarrow{C}_\alpha - \overrightarrow{C}_{NC}$, where C_{NC} is the midpoint between the amino N atom and the carboxyl C atom of Gly. Since structural motifs are by definition small sets of spatially close residues, we limit the indexation to residue pairs with a $C_\alpha - C_\alpha$ distance ≤ 20 Å.

Each residue pair occurrence corresponds to a row in one of the lookup tables generated during indexation, and that row contains the identifier of the structure in which the pair occurs, the identifiers of the two residues, and the values of the three structural descriptors. Since the speed of the motif search is mainly limited by the loading speed of the lookup tables, it is advantageous to make them as small as possible (Bittrich *et al.* 2020). Therefore, each residue pair lookup table is split into smaller tables that contain the data for a given range of values of the three structural descriptors. We used a 1 Å bin size for the distance descriptors and a 20° bin size for the angle descriptor. These small lookup tables are implemented as simple CSV files. During the indexing process, each residue pair occurrence is appended at the end of its index file. Once all the occurrences of a residue pair have been processed, each index file is transformed into a pickled and compressed Pandas dataframe (McKinney 2010).

2.2 Motif search

The motif search step consists in finding all the 3D structural motifs in the set of indexed protein structures, called target motifs, that are similar to the motif given as input by the user, called query motif. Users must first provide a protein structure file (in PDB or mmCIF format) and the residue identifiers that make up the query motif. From this information, a fully connected, weighted, undirected graph is created to represent the motif, with nodes corresponding to residues and edges to the values of the three geometric descriptors of the residue pairs.

Searching the index tables for every single residue pair in the graph would be highly redundant and thus inefficient, especially for large graphs. Indeed, geometrical constraints between residue pairs have a transitive property [i.e. a constraint between residue pair (A, B) and (B, C) implies a constraint between residues A and C] which can be exploited to minimize the amount of data that is loaded from the index. Therefore, we prune graphs that have at least four nodes by applying Kruskal's algorithm, which returns a minimum spanning tree (MST) of the graph that covers all the residues in the query motif. Then, for each residue pair in the MST, we load the index tables that match the values of the geometrical descriptors within the tolerance ranges specified by the user. As we process each residue pair, we iteratively check that the hits identified in each candidate structure are connected by an edge, otherwise the structure is discarded. Finally, the subset of structures that contain *all* the residue pairs in the MST is determined.

In the previous step, no check is performed regarding the *exact* connectivity of the residue pairs. As a result, some of the identified structures can have all the residue pairs in the correct geometrical arrangement but with different edge connections than in the query motif. To filter out these unwanted hits, we build the fully connected graph formed by all the identified residue pairs in each candidate structure and check for the presence of a subgraph that matches the query motif. The matching is performed using a subgraph monomorphism algorithm (Hagberg *et al.* 2008) (see Supplementary Material for details).

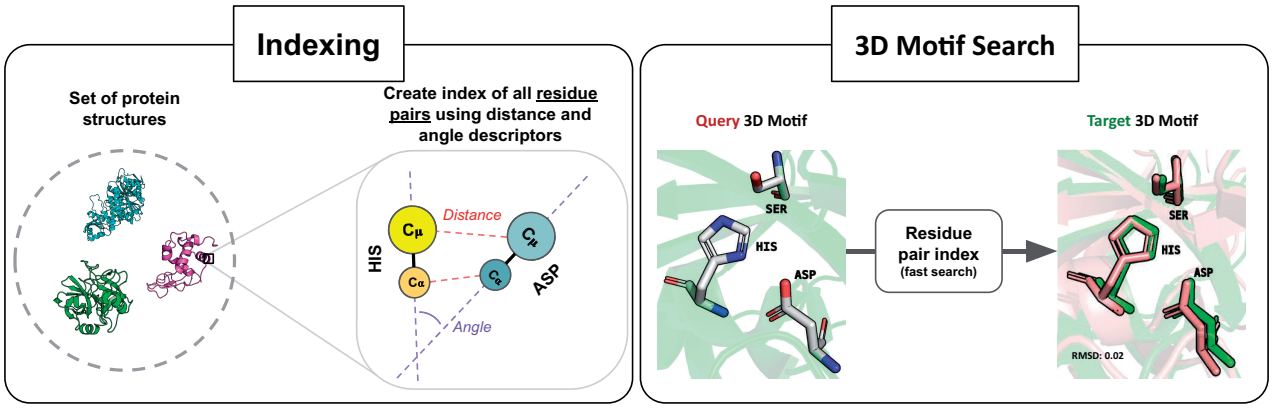

Figure 1. Schematic representation of the inverted index approach used in pyScoMotif to perform fast searches of similar 3D structural motifs in a set of protein structures. The first step involves precomputing an index containing all the residue pairs in a given protein structure dataset based on their spatial arrangement (left). Using this precomputed index, motif search can be performed rapidly by decomposing the query motif into its residue pairs and loading them from the index (right) (see Section 2 for implementation details).

Finally, to calculate the root mean square deviation (RMSD) between the query motif and each target motif found, we superimpose the motifs using the fast quaternion-based superimposition algorithm (Theobald 2005) from Biopython (Cock *et al.* 2009). Users can specify whether they want the RMSD to be calculated using the C_α coordinates, the C_μ coordinates, or both.

One of the main advantages of our implementation is that it allows users to specify different search strategies:

- Strict (by default): searches the index for target motifs that *exactly* match all the residues of the query motif.
- Position Specific Exchanges (PSE): searches the index for target motifs which may include mutated residues with respect to the query motif. Users have full control over which mutations should be allowed for each given residue in the query motif.
- Relaxed: similar to PSE search, but with possible mutations predefined based on physico-chemical amino acid properties: nonpolar (G, A, V, L, I), polar (S, T, P, N, Q), sulfur-containing (M, C), positively charged (K, R, H), negatively charged (D, E), and aromatic (F, Y, W).
- Fully relaxed: extension of the relaxed search where residues can be mutated into any of the 19 other amino acids.

When performing searches with potential mutations, users can control the maximum number of simultaneous mutations, thus allowing them to search for a large number of mutated target motifs with a single search command.

3 Performance

We assessed pyScoMotif's performance in terms of accuracy and speed and compared it with the PDB implementation (Bittrich *et al.* 2020), using the catalytic sites of serine protease, aminopeptidase and enolase, and the zinc-binding motif of the zinc-finger binding domain; details are given in Supplementary Sections S4 and S5.

In terms of accuracy, the pyScoMotif and PDB motif search implementations return very similar results: when using each method's default distance and angle thresholds and an RMSD threshold of 1 Å, we found an overlap of more than 95% between the results of the two methods on four different motifs (see Supplementary Section S5 for details). Note that we chose to use the side chain centroids C_μ rather than C_β atoms to represent residue sidechains, which makes our algorithm much more sensitive to correct side chain positioning than the PDB implementation.

pyScoMotif is highly parallelizable, thus making it highly efficient for both indexing and motif search. In terms of indexing speed, pyScoMotif is faster than the PDB implementation, as shown in Supplementary Section S4. In terms of motif search speed, pyScoMotif is slightly slower than the PDB implementation, in large part due to the use of Python, which is inherently slower than Java. However, the bottleneck of the method is index creation, and pyScoMotif is able to perform motif searches in a few seconds, as shown in Supplementary Table S2, which in absolute terms remains fast.

4 Applications

We showcase a practical application of pyScoMotif: the identification of isoenzymes. Another application related to proteome-wide search for catalytic residues can be found in Supplementary Section S6.

We performed isoenzyme identification on alcohol dehydrogenase (ADH), an extensively studied class of enzymes that are well annotated in the Mechanism and Catalytic Site Atlas (Ribeiro *et al.* 2018). To check pyScoMotif's sensitivity, we applied it to search for each ADH active site. We compared the results with the UniProt (UniprotConsortium 2023) annotations on all the structures in the index that belong to the ADH family.

In humans, alcohol metabolism is primarily achieved through the ADH enzyme family. There are seven ADH genes that code for seven ADH isozymes. The three genes ADH1A, ADH1B, and ADH1C encode class I ADH isoenzymes, which have different substrate preferences and kinetics and are abundantly present in the liver (Yin 1994). As shown in Fig. 2, the catalytic site is composed of three residues (Cys46, Cys174, His67) that bind the catalytic zinc ion, and two residues (Ser/Thr48, His51) that are involved in the catalytic reaction. In addition to the three class I ADH isoenzymes, humans encode four additional ADH classes (Rajendram *et al.* 2016): class II (ADH4), class III (ADH5), class IV (ADH7), and class V (ADH6); although all the classes share a very similar catalytic site, some of them carry distinctive mutations.

We performed the search for motifs similar to the catalytic site of PDB:1HSO (the experimental protein structure of human ADH1A) on the human proteome, whose structures were taken from the AlphaFold database (Varadi *et al.* 2022). We used default search parameters, except for the residue type which was set to fully relaxed in order to detect the different ADH isoenzymes. pyScoMotif successfully reported the structures of the seven isoenzymes as the top hits, with RMSD values ranging from 0.21 for ADH1B (Uniprot ID: P00325) to 1.02 for ADH5 (Uniprot ID: P11766).

5 Discussion

This paper introduces pyScoMotif, an efficient, flexible, and user-friendly Python implementation of a 3D structural motif search algorithm that has been inspired by Bittrich *et al.* (2020). The method allows searching for such motifs in large sets of protein structures in seconds, which otherwise could take hours using a brute force method. This is made possible by the use of an inverted index which organizes all the relevant structural information of the given set of protein structures in order to make 3D structural motif search fast. The key elements of pyScoMotif are summarized below.

pyScoMotif is designed to be easy to install and use for researchers with no structural bioinformatics background. Its implementation balances speed and disk memory efficiency, thus allowing it to be used on both laptop computers and computer clusters. The code is highly parallelizable, which makes it suitable even for searches in large protein structure datasets. Since generating the index is pyScoMotif's bottleneck in terms of time, we provide precomputed indexes of the full PDB as well as AlphaFold's global health and human proteomes at http://babylone.ulb.ac.be/pyScoMotif/data/.

Another strength of pyScoMotif that sets it apart from other algorithms is its ability to perform flexible searches of the query motif. Indeed, motif search parameters such as tolerance thresholds of the geometrical arrangement of residues allow users to make their search as specific or loose

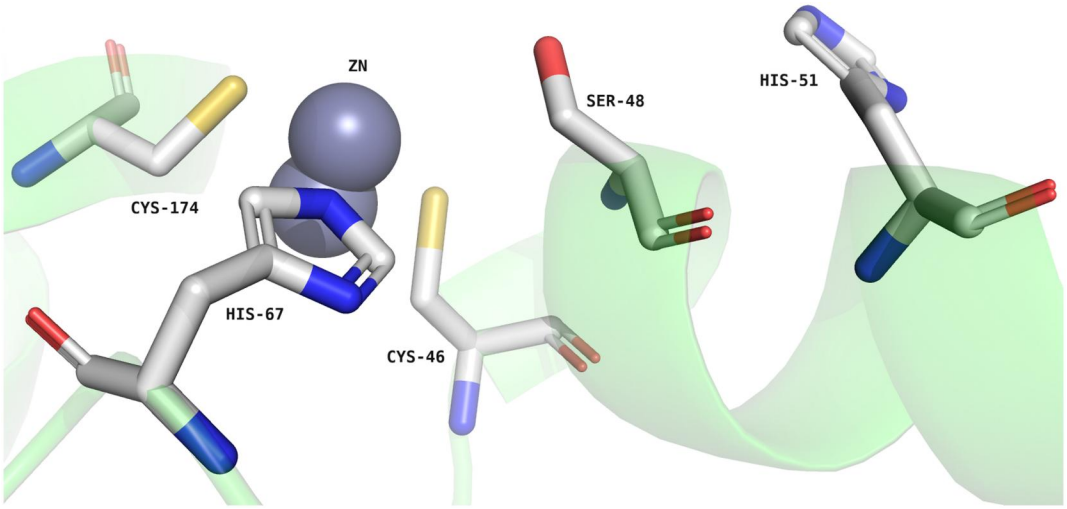

Figure 2. Catalytic site residues of the alcohol dehydrogenase structure PDB:1HSO. Generated with PyMOL (DeLano 2002).

as they wish. Moreover, users can search for possibly mutated versions of their motif by providing position-specific substitutions, and can even perform complex searches with multiple residues of the motif mutated simultaneously. The latter feature is particularly convenient as it allows users to search for a large number of similar motifs with a single command.

In summary, the rapid increase in modeled protein 3D structures now enables large-scale structural analyses that would not have been possible previously. In this evolving context, pyScoMotif emerges as a valuable tool for researchers interested in analyzing newly available structural information and performing large-scale 3D structural motif searches.

Author contributions

Gabriel Cia (Conceptualization [equal], Data curation [lead], Methodology [equal], Software [lead], Validation [lead], Writing—original draft [equal], Writing—review & editing [equal]), Jean Marc Kwasigroch (Data curation [equal], Investigation [supporting], Resources [lead]), Basile Stamatopoulos (Funding acquisition [equal], Investigation [equal], Supervision [supporting]), Marianne Rooman (Conceptualization [lead], Funding acquisition [lead], Investigation [equal], Project administration [lead], Supervision [lead], Writing—original draft [equal], Writing—review & editing [equal]), and Fabrizio Pucci (Conceptualization [lead], Funding acquisition [lead], Investigation [equal], Project administration [lead], Supervision [lead], Writing—original draft [equal], Writing—review & editing [equal])

Supplementary data

Supplementary data are available at *Bioinformatics Advances* online.

Conflict of interest

None declared.

Funding

This work was supported by the FNRS with a PDR grant; M. R. is a FNRS research director and G.C. benefits from a FNRS-FRIA PhD grant.

References

Ananthalakshmi P, Kumar CK, Jeyasimhan M *et al.* Fragment Finder: a web-based software to identify similar three-dimensional structural motif. *Nucleic Acids Res* 2005;**33**:W85–8.

Berman HM, Westbrook J, Feng Z *et al.* The protein data bank. *Nucleic Acids Res* 2000;**28**:235–42.

Bittrich S, Burley SK, Rose AS *et al.* Real-time structural motif searching in proteins using an inverted index strategy. *PLoS Comput Biol* 2020;**16**:e1008502.

Cock PJA, Antao T, Chang JT *et al.* Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* 2009;**25**:1422–3.

Debret G, Martel A, Cuniasse P *et al.* RASMOT-3D PRO: a 3D motif search webserver. *Nucleic Acids Res* 2009;**37**:W459–64.

DeLano WL. Pymol: an open-source molecular graphics tool. *CCP4 Newslett Protein Crystallogr* 2002;**40**:82–92.

Fraser HB, Hirsh AE, Steinmetz LM *et al.* Evolutionary rate in the protein interaction network. *Science* 2002;**296**:750–2.

Hagberg A *et al.* Exploring network structure, dynamics, and function using NetworkX. *Technical report*. Los Alamos National Lab, 2008.

Kirshner DA, Nilmeier JP, Lightstone FC *et al.* Catalytic site identification—a web server to identify catalytic site structural matches throughout PDB. *Nucleic Acids Res* 2013;**41**:W256–65.

Konc J, Janežič D. ProBiS algorithm for detection of structurally similar protein binding sites by local structural alignment. *Bioinformatics* 2010;**26**:1160–8.

McKinney W. Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*, 2010; **445**:51–6.

Moll M, Bryant DH, Kavraki LE *et al.* The LabelHash algorithm for substructure matching. *BMC Bioinformatics* 2010;**11**:555–15.

Nadzirin N, Gardiner EJ, Willett P *et al.* SPRITE and Assam: web servers for side chain 3D-motif searching in protein structures. *Nucleic Acids Res* 2012;**40**:W380–6.

Pennec X, Ayache N. A geometric algorithm to find small but highly similar 3D substructures in proteins. *Bioinformatics (Oxford, England)* 1998;**14**:516–22.

Rajendram R, Rajendram R, Preedy VR. Ethanol metabolism and implications for disease. In: *Neuropathology of Drug Addictions and Substance Misuse*. San Diego: Academic Press, 2016:377–88.

Ribeiro AJM, Holliday GL, Furnham N *et al.* Mechanism and Catalytic Site Atlas (M-CSA): a database of enzyme reaction mechanisms and active sites. *Nucleic Acids Res* 2018; **46**:D618–23.

Ribeiro AJM, Tyzack JD, Borkakoti N *et al.* A global analysis of function and conservation of catalytic residues in enzymes. *J Biol Chem* 2020;**295**:314–24.

Spriggs RV, Artymiuk PJ, Willett P *et al.* Searching for patterns of amino acids in 3D protein structures. *J Chem Inf Comput Sci* 2003; **43**:412–21.

Theobald DL. Rapid calculation of RMSDs using a quaternion-based characteristic polynomial. *Acta Crystallogr A* 2005;**61**:478–80.

UniprotConsortium. UniProt: the Universal Protein knowledgebase in 2023. *Nucleic Acids Research* 2023;**51**:D523–31.

Varadi M, Anyango S, Deshpande M *et al.* AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Res* 2022; **50**:D439–44.

Yin S-J. Alcohol dehydrogenase: enzymology and metabolism. *Alcohol Alcohol (Oxford, Oxfordshire)* 1994;**2**:113–9.