

## Structural bioinformatics

# PyPCN: protein contact networks in PyMOL

Serena Rosignoli <sup>1,\*</sup>, Luisa di Paola<sup>2</sup>, Alessandro Paiardini <sup>1,\*</sup>

<sup>1</sup>Department of Biochemical Sciences “A. Rossi Fanelli”, Sapienza University of Rome, 00185 Rome, Italy

<sup>2</sup>Unit of Chemical-Physics Fundamentals in Chemical Engineering, Department of Engineering, Università Campus Bio-Medico di Roma, 00128 Rome, Italy

\*Corresponding authors. Department of Biochemical Sciences “A. Rossi Fanelli”, Sapienza University of Rome, 00185 Rome, Italy.

E-mails: serena.rosignoli@uniroma1.it (S.R.) and alessandro.paiardini@uniroma1.it (A.P.)

Associate Editor: Lenore Cowen

### Abstract

**Motivation:** Protein contact networks (PCNs) represent the 3D structure of a protein using network formalism. Inter-residue contacts are described as binary adjacency matrices, which are derived from the graph representation of residues (as  $\alpha$ -carbons,  $\beta$ -carbons or centroids) and Euclidean distances according to defined thresholds. Functional characterization algorithms are computed on binary adjacency matrices to unveil allosteric, dynamic, and interaction mechanisms in proteins. Such strategies are usually applied in a combinatorial manner, although rarely in seamless and user-friendly implementations.

**Results:** PyPCN is a plugin for PyMOL wrapping more than twenty PCN algorithms and metrics in an easy-to-use graphical user interface, to support PCN analysis. The plugin accepts 3D structures from the Protein Data Bank, user-provided PDBs, or precomputed adjacency matrices. The results are directly mapped to 3D protein structures and organized into interactive diagrams for their visualization. A dedicated graphical user interface combined with PyMOL visual support makes analysis more intuitive and easier, extending the applicability of PCNs.

**Availability and implementation:** <https://github.com/pcnproject/PyPCN>.

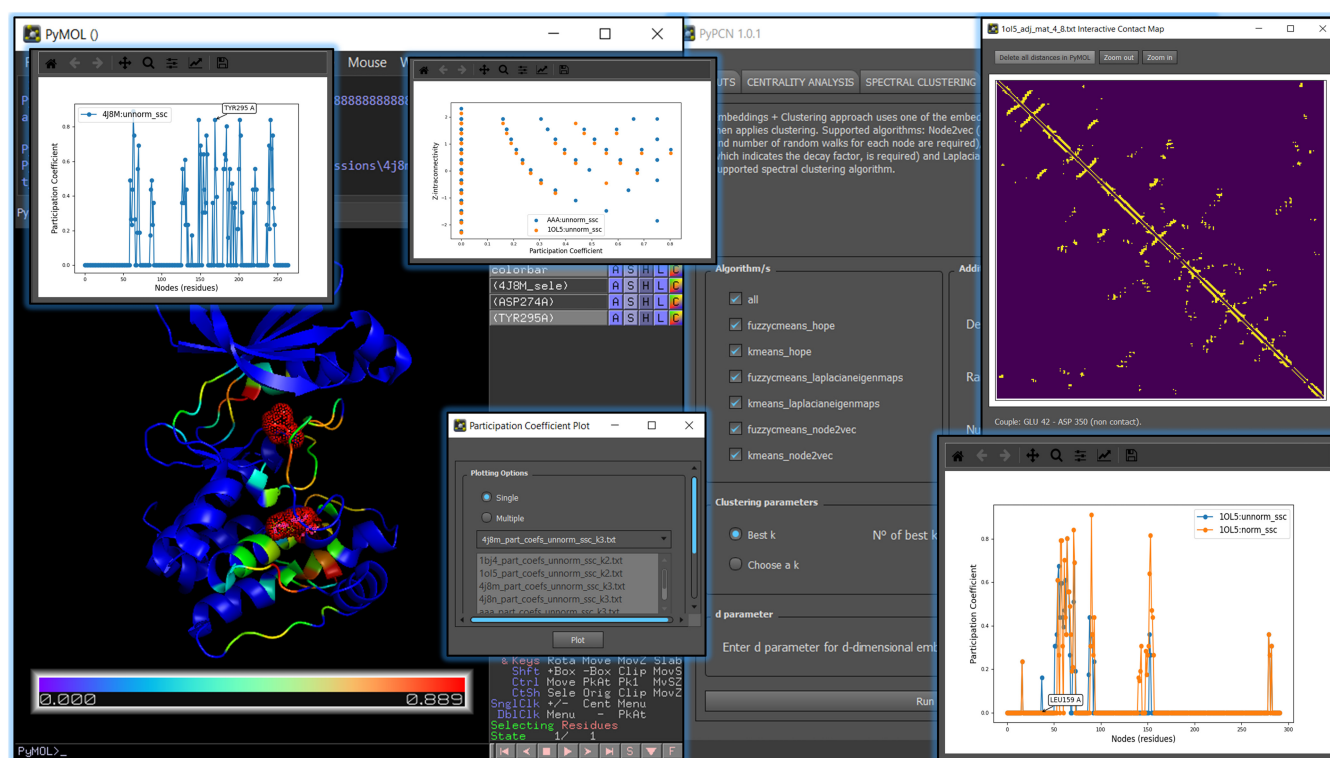
## 1 Introduction

In recent years, outstanding breakthroughs have been made in the field of structural biology, thanks to the increasing amount of data obtained using innovative experimental and computational techniques (Jumper *et al.* 2021, Porta-Pardo *et al.* 2022, Burley *et al.* 2023). Although a detailed understanding of protein function and dynamics can be gained with these data, many of the unanswered questions regarding the intimate relationship between structure and function could benefit from a simplified representation of protein structural information. The graph representation of protein inter-residue contacts is an application area of graph theory termed protein contact network (PCN), which is gaining appraisal as an informative schematization of 3D protein structures (Krishnan *et al.* 2008, Di Paola *et al.* 2013). To build a PCN, the structural information of the  $\alpha$ -carbons ( $C\alpha$ ) is extracted to create binary matrices describing the relationships between the protein residues. The  $C\alpha$  are the nodes, whilst the significant contacts, according to defined thresholds, are the edges. A topological representation, in contrast to a geometric description, is considered an efficient and accurate means for structural characterization, with low computational cost (Doncheva *et al.* 2012, Tasdighian *et al.* 2014). In this regard, many of the best-known metrics for graph analyses, i.e. clustering techniques and centrality measures (Doncheva *et al.* 2011, Brysbaert and Lensink 2021), found valid application in structural biology, with proven relevance in protein

allosteric, dynamic, and interaction analyses (Liu and Nussinov 2016, Das *et al.* 2021, Dubanevics and McLeish 2021, Di Paola *et al.* 2022, Guzzi *et al.* 2023).

Despite the abundance and diversity of graph analysis modules, the state of the art in the applicability of PCNs suffers from poor usability. This leads to a lack of workflows for creating and analyzing PCNs, unless there is an expert-driven integration of different tools and implementations. To address this issue and facilitate visual analysis of PCNs, some web-servers were developed in recent years (Seeber *et al.* 2015, Chakrabarty and Parekh 2016, Felling *et al.* 2020). However, despite their ease of use, the major limitation of these tools is their inherent design for specific analyses and the resulting lack of flexibility, which limits their applicability, extensibility, and integration with other protein analysis tools. On the other hand, the use of an open-source tool such as PCN-Miner (Guzzi *et al.* 2022), which offers the possibility of accessing a large number of different algorithms from a large user base, is limited by a very basic and essential interface that is not extensible and embeddable in complex analysis workflows. With this in mind, we were motivated to develop a new and more efficient environment to support PCN analysis, keeping pace with the latest modern, unified user interfaces.

Here, we introduce PyPCN (Fig. 1), a user-friendly PyMOL (Schrödinger, LLC) plugin for computation, visualization and analysis of PCNs. The plugin aims to bridge the gap between knowledge of graphs and structure–function characterization



**Figure 1.** Overview of PyPCN. Some of the analyses supported by PyPCN, and its integration with PyMOL, are shown. It showcases interactive participation coefficient plots for both single structures (upper left) and multiple structures (upper right and bottom right). The upper-right window displays the interactive contact map. The PyMOL viewer demonstrates how the results are mapped onto the 3D structure, including the highlighting of residues when selected from the interactive plots. For more comprehensive information regarding all the analyses that have been implemented, refer to the User's Guide ([Supplementary Material](#)).

of proteins, bypassing tedious command line implementations or inaccessibility of source codes for specific protocols.

## 2 Implementation

PyPCN is written in Python3 with Object-Oriented logic and the graphical user interface (GUI) development is supported by PyQT-5 ([Summerfield 2008](#)). It has been tested and works on Linux, Windows, and macOS. PyPCN works as a plugin of the popular Molecular Graphics Viewer PyMOL, which is required in its version 2.3 or higher, either Incentive or Open-source. At the first usage, the user is asked to choose a working directory, and guided to proceed with the installation of external dependencies ([Supplementary Material](#)), for which PyPCN offers an automatic process.

PyPCN computes, starting from a PDB file, the binary adjacency matrix, unless it is provided with a precomputed one in a properly formatted file ([Supplementary Material](#)). The PDB file can be provided as a valid PDB-ID in the GUI input form; a use-case in which the protein to be analyzed is downloaded from Protein Data Bank ([Berman \*et al.\* 2000](#)). The alternative option is to provide PyPCN for a user-customized protein structure by loading it in PyMOL.

Once the setup is complete, the workflow of the analysis is divided into 4 modules for the (i) centrality analysis; (ii) spectral clustering; (iii) embedded clustering; and (iv) community extraction. The centrality of a node (i) in PyPCN can be computed either by means of its degree of connectivity, or with path-related metrics, such as the betweenness, closeness, and eigenvector ([Brinda and Vishveshwara 2005](#), [De Ruvo \*et al.\*](#)

[2012](#), [Di Paola \*et al.\* 2013](#)) ([Supplementary Material](#)). With regards to the clustering and community extraction methods (ii–iv) PyPCN supports 20 algorithms and the related parameters. Along with every clustering partition, PyPCN quantifies the role of connectivity of each node ([Supplementary Material](#)), both overall and in a specific cluster, by computing the Participation Coefficient ( $P$ ) and the Intramodular Connectivity  $z$ -score ( $z$ ) ([Di Paola and Giuliani 2015](#), [Hu \*et al.\* 2017](#)). Using PyPCN as an integrated platform for these algorithms serves as a foundational tool for analyzing outputs beyond the plugin's scope, i.e. graph representations of conformational ensembles ([Yao \*et al.\* 2018](#)).

In addition to saving the results in the previously setup working directory, PyPCN stores all information of any analysis in a dedicated window, allowing the user to conveniently proceed with multiple and comparative analysis avoiding any loss of the results.

Thanks to the implementation in PyMOL, the gap between data retrieval and visualization is filled. The per-atom computed values are color-mapped onto the 3D protein structure and straightly visualized in PyMOL. The adjacency matrices can be visualized as interactive contact maps, by clicking on which the information stored in the 2D plot is highlighted in the 3D structure. The per-residue centrality calculations, as well as  $z$ - $p$  correlations, of one or more proteins, can be visualized as editable and interactive line plots.

## Supplementary data

[Supplementary data](#) are available at *Bioinformatics* online.

## Conflict of interest

None declared.

## Funding

This work was supported by Associazione Italiana Ricerca sul Cancro [MFAG id. 20447 to A.P.]; and Sapienza University [RP12017275CED09F, RM1221815D52AB32].

## References

- Berman HM, Westbrook J, Feng Z *et al.* The protein data bank. *Nucleic Acids Res* 2000;**28**:235–42.
- Brinda KV, Vishveshwara S. A network representation of protein structures: implications for protein stability. *Biophys J* 2005;**89**:4159–70.
- Brysbaert G, Lensink MF. Centrality measures in residue interaction networks to highlight amino acids in Protein-Protein binding. *Front Bioinform* 2021;**1**:684970.
- Burley SK, Bhikadiya C, Bi C *et al.* RCSB protein data bank (RCSB.org): delivery of experimentally-determined PDB structures alongside one million computed structure models of proteins from artificial intelligence/machine learning. *Nucleic Acids Res* 2023;**51**:D488–508.
- Chakrabarty B, Parekh N. NAPS: network analysis of protein structures. *Nucleic Acids Res* 2016;**44**:W375–82.
- Das JK, Roy S, Guzzi PH *et al.* Analyzing host-viral interactome of SARS-CoV-2 for identifying vulnerable host proteins during COVID-19 pathogenesis. *Infect Genet Evol* 2021;**93**:104921.
- De Ruvo M, Giuliani A, Paci P *et al.* Shedding light on protein-ligand binding by graph theory: the topological nature of allostery. *Biophys Chem* 2012;**165**–166:21–9.
- Di Paola L, De Ruvo M, Paci P *et al.* Protein contact networks: an emerging paradigm in chemistry. *Chem Rev* 2013;**113**:1598–613.
- Di Paola L, Giuliani A. Protein contact network topology: a natural language for allostery. *Curr Opin Struct Biol* 2015;**31**:43–8.
- Di Paola L, Poudel H, Parise M *et al.* A statistical journey through the topological determinants of the  $\beta_2$  adrenergic receptor dynamics. *Entropy (Basel)* 2022;**24**:998.
- Doncheva NT, Klein K, Domingues FS *et al.* Analyzing and visualizing residue networks of protein structures. *Trends Biochem Sci* 2011;**36**: 179–82.
- Doncheva NT, Assenov Y, Domingues FS *et al.* Topological analysis and interactive visualization of biological networks and protein structures. *Nat Protoc* 2012;**7**:670–85.
- Dubanevics I, McLeish TCB. Computational analysis of dynamic allostery and control in the SARS-CoV-2 main protease. *J R Soc Interface* 2021;**18**:20200591.
- Felline A, Seeber M, Fanelli F *et al.* webPSN v2.0: a webserver to infer fingerprints of structural communication in biomacromolecules. *Nucleic Acids Res* 2020;**48**:W94–103.
- Guzzi PH, Di Paola L, Giuliani A *et al.* PCN-Miner: an open-source extensible tool for the analysis of protein contact networks. *Bioinformatics* 2022;**38**:4235–7.
- Guzzi PH, di Paola L, Puccio B *et al.* Computational analysis of the sequence-structure relation in SARS-CoV-2 spike protein using protein contact networks. *Sci Rep* 2023;**13**:2837.
- Hu G, Di Paola L, Liang Z *et al.* Comparative study of elastic network model and protein contact network for protein complexes: the hemoglobin case. *Biomed Res Int* 2017;**2017**:2483264.
- Jumper J, Evans R, Pritzel A *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* 2021;**596**:583–9.
- Krishnan A, Zbilut JP, Tomita M *et al.* Proteins as networks: usefulness of graph theory in protein science. *Curr Protein Pept Sci* 2008;**9**: 28–38.
- Liu J, Nussinov R. Allostery: an overview of its history, concepts, methods, and applications. *PLoS Comput Biol* 2016;**12**: e1004966.
- Porta-Pardo E, Ruiz-Serra V, Valentini S *et al.* The structural coverage of the human proteome before and after AlphaFold. *PLoS Comput Biol* 2022;**18**:e1009818.
- Seeber M, Felling A, Raimondi F *et al.* WebPSN: a web server for high-throughput investigation of structural communication in biomacromolecules. *Bioinformatics* 2015;**31**:779–81.
- Summerfield M. *Rapid GUI Programming with Python and QT: the Definitive Guide to PyQt Programming*. Upper Saddle River, NJ: Prentice Hall, 2008.
- Tasdighian S, Di Paola L, De Ruvo M *et al.* Modules identification in protein structures: the topological and geometrical solutions. *J Chem Inf Model* 2014;**54**:159–68.
- Yao X-Q, Momin M, Hamelberg D *et al.* Elucidating allosteric communications in proteins with difference contact network analysis. *J Chem Inf Model* 2018;**58**:1325–30.