Network Modeling Analysis in Health Informatics and Bioinformatics (2023) 12:3
https://doi.org/10.1007/s13721-022-00397-9

ORIGINAL ARTICLE

Structural analysis of SARS-CoV-2 Spike protein variants through graph embedding

Pietro Hiram Guzzi[1] · Ugo Lomoio[1] · Barbara Puccio[1] · Pierangelo Veltri[1]

Received: 12 July 2022 / Revised: 21 October 2022 / Accepted: 16 November 2022 / Published online: 2 December 2022

Abstract
Since December 2019, severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) has affected almost all countries. The unprecedented spreading of this virus has led to the insurgence of many variants that impact protein sequence and structure that need continuous monitoring and analysis of the sequences to understand the genetic evolution and to prevent possible dangerous outcomes. Some variants causing the modification of the structure of the proteins, such as the Spike protein S, need to be monitored. Protein contact networks (PCNs) have been recently proposed as a modelling framework for protein structures. In such a framework, the protein structure is represented as an unweighted graph whose nodes are the central atoms of the backbones (C-α), and edges connect two atoms falling in the spatial distance between 4 and 7 Å. PCN may also be a data-rich representation since we may add to each node/atom biological and topological information. Such formalism enables the possibility of using algorithms from graph theory to analyze the graph. In particular, we refer to graph embedding methods enabling the analysis of such graphs with deep learning methods. In this work, we explore the possibility of embedding PCN using Graph Neural Networks and then analyze in the embedded space each residue to distinguish mutated residues from non-mutated ones. In particular, we analyzed the structure of the Spike protein of the coronavirus. First, we obtained the PCNs of the Spike protein for the wild-type, α, β, and δ variants. Then we used the GraphSage embedding algorithm to obtain an unsupervised embedding. Then we analyzed the point of mutation in the embedded space. Results show the characteristics of the mutation point in the embedding space.

1 Introduction

Proteins play a prominent role in many biological processes. The molecular structure determines the function of each protein. Structural data about each protein are usually determined from experiments (e.g. X-ray crystallography or NMR; Petrey and Honig 2005). More recently, a set of computational prediction methods (e.g., Jumper and Pritzel 2021; Kukic et al. 2014; Palopoli et al. 2009; Gu et al. 2022) predicting protein structure has been introduced. Protein structures are finally stored in publicly available databases such as the Protein Databank (PDB) (Bittrich et al. 2022). Such data are also useful to unravel many biologically relevant problems, such as the structure-to-function relationship and the interaction among proteins (Eswar et al. 2003). The so-far introduced analysis requires the use of appropriate computational models to represent structures and enable the development of novel algorithms.

Protein contact networks (PCNs) emerged as a relevant paradigm for the analysis of protein molecular structures (Di Paola et al. 2013). PCN are networks whose nodes represent the $C-\alpha$ atoms of the backbone of proteins, while edges represent a relative spatial distance among 4 and 8 Å. Figure 1 depicts an example of a protein structure and fragment of the resulting graph. Topological descriptors of PCNs, such as centrality measures, are used to discover protein properties, even at the sub-molecular level. Protein modularity, for instance, is specifically suited to identify modular domains in a structure, whose mutual interactions are responsible for allosteric regulation, i.e., the protein structure adaptation to environmental cues (Khan and Ghosh 2015; Das et al. 2021). Existing approaches use spectral clustering to identify network modules that correspond to allosteric regions (Tasdighian et al. 2014).

Despite the relevance, these approaches are based only on network structure. Thus, they cannot gather biological and biochemical information broadly available

✉ Pietro Hiram Guzzi
hguzzi@unicz.it

[1] Department of Surgical and Medical Sciences, University of Catanzaro, Catanzaro, Italy

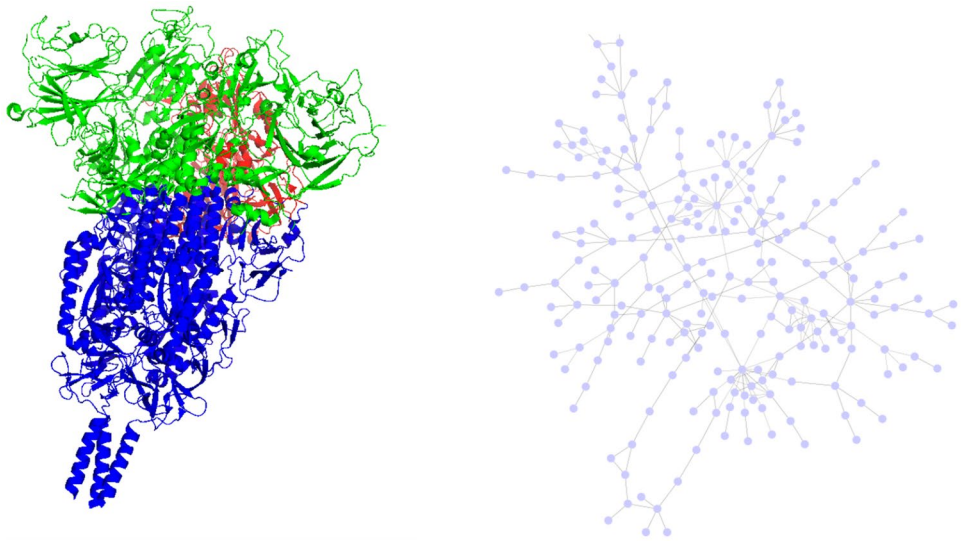

Fig. 1 Figure shows an example of a PCN. On the right part of the figure, the spatial structure of the 7sbk protein is depicted. After the structure analysis, a graph, as represented on the right, is obtained

for both nodes and edges. Moreover, classical clustering approaches are inherently transductive since they need to re-analyze all the networks in the presence of modification in both nodes and edges.

Many approaches have recently demonstrated that graph structures may be efficiently mapped into latent spaces and then analyzed using deep learning and data-mining methods. Such processes, known as *embedding methods*, map graph nodes into a so-called embedding space, and the transformation preserves node-similarity (Hamilton et al. 2017b). There exist different node-embedding methods. A large class of methods were based only on the analysis of the topology of the input graph [such as node2vec (Grover and Leskovec 2016), deep walk (Perozzi et al. 2014)]. These methods had two main drawbacks: (i) the impossibility of including information related to nodes; (ii) the need to recalculate the whole embedding in case of graph changes. Subsequent methods overcame these limitations by leveraging the computational power of an ad-hoc-developed neural network. GraphSage (Hamilton et al. 2017b) is a general framework that can leverage node features (e.g. attributes associated with each node) to generate node embeddings. It is based on an inductive process, so it can generate node embedding for unseen nodes without requiring the analysis of the whole graph again. It is based on learning a function that generates embeddings by the analysis of the neighbourhood of each node aggregating the features. It has been used for node classification and clustering tasks with good performances.

Here we integrate the previous methods into a single framework of analysis. Our framework is based on the integration of existing software modules for the whole process of the analysis: Creation of the Protein Contact Network Embedding and analysis of the network Visualization of the results

We apply the framework by presenting a case study of the analysis of the structure of the Spike protein of SARS-CoV-2 (Guzzi et al. 2020; Ortuso et al. 2021). SARS-CoV-2, which caused the recent pandemic, presents many sequence mutations that impact its protein structure. Among the others, mutations of Spike protein are particularly of interest since they impact the transmission of the virus. To the best of our knowledge, existing work does not face the analysis of such mutation on the embedding space.

The aims of this work are:

- Providing a mechanicist framework for the analysis of the structure of PCN in general;
- applying the framework to study structures of the variants of the Spike protein;
- contributing to elucidating the differences among variants of the SARS-CoV-2 proteins.

Therefore we first consider protein structures of the wild-type (i.e. unmutated) Spike protein and the structures of main variants: alpha, beta, delta, and omicron (Eskandarzade et al. 2022; Gordon et al. 2020; Kumar Das et al. 2021; Ortuso et al. 2022). Then we obtain the PCN representation for each structure. Finally, we map each structure into the embedding space using GraphSage and analyze the differences between mutated and unmutated residues. We train an unsupervised learning model, aiming to distinct in the topological space points of mutation from other points. Results evidence that PCN of variants are globally different, while more investigation needs to be performed to characterise these points better.

The paper is structured as follows: Sect. 2 introduces the Protein Interaction Network formalism. Section 3 discusses briefly the state-of-the-art approaches for node embedding. Then, Sect. 4 presents the proposed framework, its

architecture and main modules. Finally, Sect. 5 concludes the paper.

2 Protein interaction networks

A protein structure can be represented as a complex three-dimensional object, formally defined by its atoms' coordinates in 3D space. Despite the large availability of protein molecular structures data, there are yet many problems regarding the relationship between protein structures and their functions. For this reason, it is necessary to define simple descriptors that can describe protein structures with few numerical variables. Structure and function are based on the complex network of inter-residue interactions, where residues are identified by aminoacids sequences (Di Paola et al. 2013). Therefore, the residues interactions are used to define protein interaction networks. Protein interaction networks are thus used to study protein functions. The most simple choice to define networks is to represent the protein structure using α-carbon location. The spatial position of C_α is still reminiscent of the protein backbone, allowing us to highlight the three-dimensional structure's most important characteristics. Starting from the C_α spatial distribution, a distance matrix d is evaluated where each $d_{i,j}$ represents the Euclidean distance in the 3D space between the i-th and j-th residues, defined as

$$d_{i,j} = \sqrt{((x_i - x_j)^2) + ((y_i - y_j)^2) + (z_i - z_j)^2)} \quad (1)$$

where (x_i, y_i, z_i) and (x_j, y_j, z_j) respectively are the Cartesian coordinates of residue i and j. Matrix d is used to define a Protein Contact Network concept, which is an alternative and different representation of using graph-based models to represent protein structures. A graph is the most natural structure to represent proteins, where nodes (or vertices) are the protein residues and links (or edges) between the i-th and the j-th nodes (residues) represent residue contacts. In the graph representation, there exists a link between two residues i and j if the distance between two residues (i.e., $d_{i,j}$) is higher than 4 and lower than 8 Å. The lower end excludes all covalent bonds, which are not sensitive to environment change (so to protein functionality). In contrast, the upper end removes weaker non-covalent bonds (so not significant for protein functionality). At this point, it is possible to build up adjacency matrix A, whose generic element is defined as:

$$A_{ij} = \begin{cases} 1 & \text{if } 4\,\text{Å} \le d_{ij} \le 8\,\text{Å} \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

A graph's adjacency matrix is unique regarding the ordering nodes. In the case of proteins, in which the order of nodes (residues) corresponds to the residues sequence (primary structure) it can be said that its corresponding network is unique: this establishes a one-to-one correspondence between protein and its network.

3 Graph embeddings

Graph embedding approaches represented an answer to the primary challenge within machine learning: finding a way to represent or encode graph structure so that the machine learning model can easily exploit it. These approaches, that are usually referred to as *graph representation learning* or *graph-embedding*, automatically learn to encode graph structure into low-dimensional embeddings, using techniques based on deep learning and nonlinear dimensionality reduction (Hamilton et al. 2017a; Agapito et al. 2019; Guzzi and Zitnik 2022). The main purpose of graph embedding methods is to encode nodes in a latent vector space which means packaging the properties of each node into a vector with a smaller size. The embeddings learned can also support graph analysis much faster and more accurately compared to the direct execution of such tasks in the domains of complex high-dimensional graphs. Considering, for instance, node embedding, the aim of the mapping is to associate each node (and the associated features) to a low-dimensional vector.

These vector spaces correspond to a notion of similarity by preserving a graph's inherent properties and structure, i.e., similar nodes in the original graph space will be closer to each other in latent vector space. The generated embeddings reflect a network's updated features that carry the nonlinear graph information.

Node embedding techniques generate low-dimensional vectors by solving an optimization problem that follows an unsupervised learning schema. Based on the approach to generate embeddings, node embedding methods can be categorized into three major categories: (1) Matrix-Factorization, (2) Random walks, and (3) Graph Neural Networks. Embedding methods that fall under the matrix factorization and random-walk category are known as shallow embedding methods. They are hard-engineered, transductive and often fail to capture node attribute information. In contrast, the graph neural network-based embedding methods are known as deep graph encoders as they produce better representations by specifically involving in deep, multilayered approach for learning or training mechanisms.

In particular, the Random Walks-based approaches to learn node embeddings described over a walk with a successive number of random steps in a network. The Random Walks incorporate local and higher-order topological neighbourhood information of a network. The key idea is to derive the similarity between two nodes based on the co-occurrence of nodes in the respective random walks by observing that

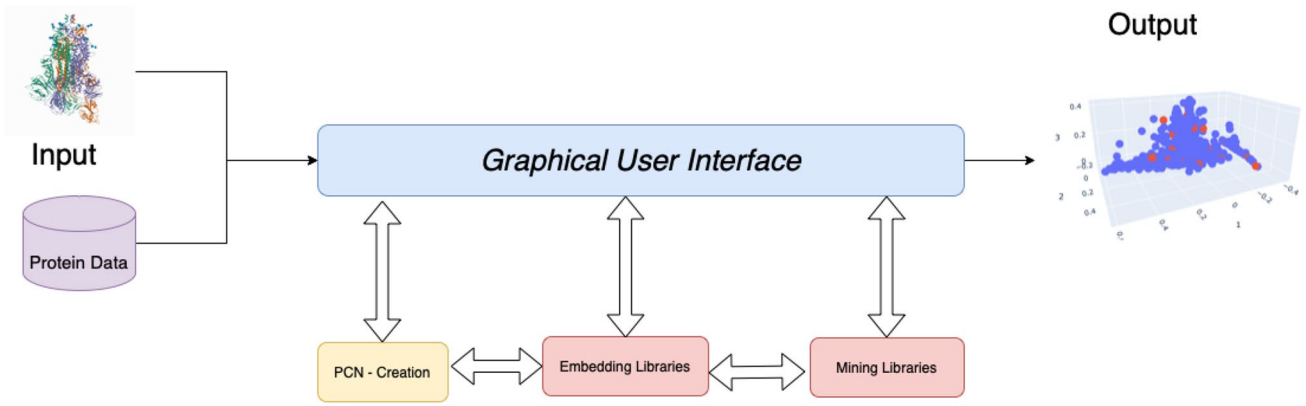

Fig. 2 Architecture of the proposed framework

two similar nodes have a greater chance of having similar random walks.

Different methods based on random walks have been developed depending on the strategy used to calculate similarity (e.g. the way to simulate random walks). *DeepWalk* (Perozzi et al. 2014) introduces unsupervised feature learning on graph data by incorporating truncated random walks to learn latent representations. The method exploits structural regularities and processes random walks equivalent to sentences in neural language modelling. *Node2Vec* (Grover and Leskovec 2016), which is a modified version of Deepwalk, samples the sequence of random walk based on DFS (depth-first-search) and BFS (Breadth-FirstSearch) strategies. *LINE* (Tang et al. 2015) can embed networks of large sizes and arbitrary types: undirected, directed, and weighted. The model carefully designs an objective function that optimizes and preserves both the local and the global structural information of graphs by testing the performance on word analogy, text classification, and node classification. *Struct2vec* (Ribeiro et al. 2017) model learning latent representations for classification task. The representations are generated via a biased random walk to produce node

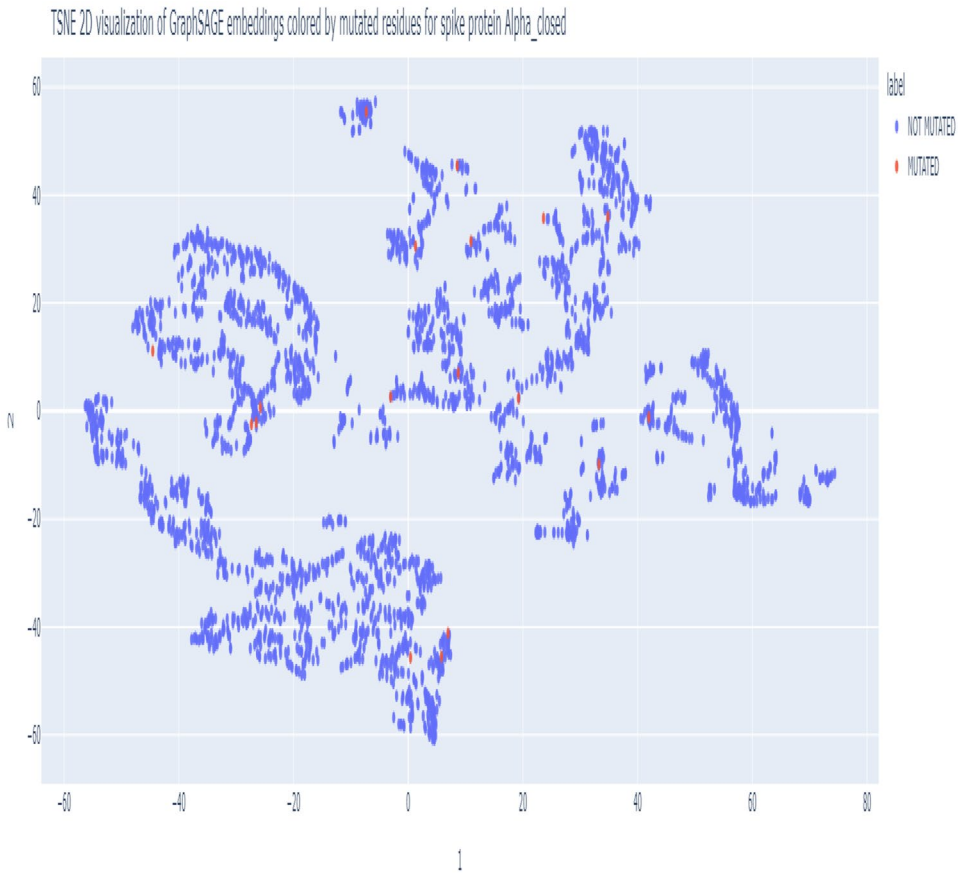

Fig. 3 t-sne visualisation of node embeddings obtained by GraphSAGE for Spike protein of the alpha variant. Red dots represent site of mutation (colour figure online)

sequences. All the proposed approaches have two main limitations: (i) they do not take into account data related to nodes (i.e. node features); (ii) they are inherently transductive, so they need to recalculate the whole embedding in case of any graph modification (e.g. node/edge insertion or removal). A set of different approaches have been introduced to overcome these limitation. The first method presented in literature is GraphSAGE (Hamilton et al. 2017a), which uses Graph Convolutional Networks to learn the mappings. GraphSAGE is based on an inductive framework that leverages node feature information (e.g., text attributes), so it can gather such information during the embeddings and efficiently generate node embeddings for previously unseen data without analysing the whole graph. For each node i of the graph, GraphSage can learn the embedding by analyzing the neighbours' information, e.g. all the nodes at a distance $K = 2$. After the determination of the neighborhood for each node, it use two functions for generating the embedding, *aggregation*, and *concatenation*. Aggregation functions accept a neighbourhood as input and combine each neighbour's embedding with weights to create a neighbourhood embedding. The aggregation function use weights that are shared among all the node of the graph, and such weights are either learned or fixed.

4 The proposed approach

4.1 Framework architecture

The proposed framework is based on some main modules, as represented in Fig. 2. Users may insert the input protein data, encoded as a PDB file through a Graphical User Interface. The GUI is responsible for invoking the building of the Protein Contact Network at first. The PCN-Creation module is realized by wrapping the PCN-Miner libraries (Guzzi et al. 2022a, b; Gu et al. 2022). After creating the network, the user may leverage embedding and subsequent mining libraries to analyse the embeddings. Both libraries are included in our framework by wrapping the Stellar Graph library (Data61 2018).

The StellarGraph library offers state-of-the-art algorithms for machine learning on graphs. It provides algorithms of representation learning for nodes and edges; Classification of nodes and edges, and link prediction. StellarGraph is built

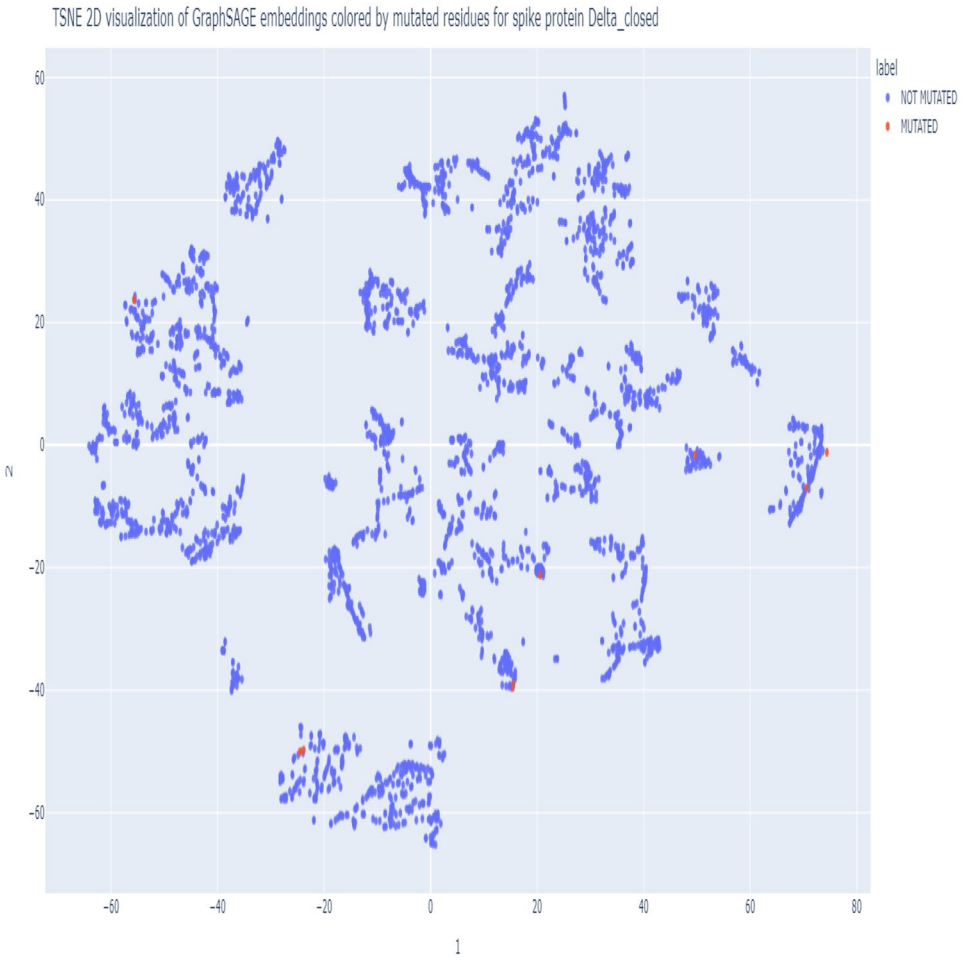

Fig. 4 t-sne visualisation of node embeddings obtained by GraphSAGE for Spike protein of the Delta variant. Red dots represent site of mutation (colour figure online)

on TensorFlow 2 and its Keras high-level API, as well as Pandas and NumPy.

The framework is implemented in the Python Programming Language.

The first step is the creation of a Protein Contact Network from structural data. The *PCN-Creation* module is responsible for this task. It reads a Protein Data Bank File, and it can build a PCN. For each atom of the C-α backbone the module adds a node into the PCN. Then, all the pairwise distances are calculated. Finally, for each distance among two atoms i, j that fall into the range 4–8 Å, the module adds an edge among nodes i, and j into the network. This step adds to each node the information about the centrality values of the node itself. The *PCN-enrichment* module is responsible for this task. Currently, we calculate closeness, eigenvector, and betweenness centrality (Guzzi and Milenković 2018). These values are a set of node features we add to each node through the networkX library.

Finally, we learn the representation of each PCN through Graphsage in an unsupervised mode. GraphSAGE learns node embeddings in this modality by solving a classification task: nodes are subdivided into positive and negative groups. Positive nodes are generated from the analysis of simulated random walks (i.e. nodes that frequently co-occur in random walks), while negative nodes are randomly selected pairs. The binary classifier predicts the likelihood of co-occurrence in a random walk performed on the graph. The classification task is used to learn an inductive mapping from attributes of nodes and their neighbours to node embedding.

We downloaded three protein structures, specified with a code (PDB code), from the Protein Data Bank (PDB https://www.rcsb.org/), an archive of 3D structure data: 7FET (variant *alpha*), 7SBK (variant *Delta*), and 7WK2 (variant *Omicron*). Coordinates of the Carbon $-\alpha$ atoms were used to get PCNs. Starting from 3D structure, we obtained the corresponding PCN for each structure using PCN-Miner. This tool (Zitnik et al. 2018) allows to import the structure in .pdb format, to extract structural information and to build the corrispective PCN. Protein network nodes are built to represent single residues. Links between nodes (residues) are defined if the distance between corresponding residues lies between 4 and 8 Å. This threshold, that PCN-miner allows to set, is chosen to map connections only for relevant non-covalent intra-molecular interactions. Then we calculated for each node following centrality measures: Degree, Eigenvector, Closeness and Betweenness. We then apply GraphSAGE with the standard parameters and we these points using t-sne transformation. Figures 3, 4, and 5, depict the embedding of the *alpha*, *delta*, and *omicron* variant. Each point on the figure

Fig. 5 t-sne visualisation of node embeddings obtained by GraphSAGE for Spike protein of the Omicron variant. Red dots represent site of mutation (colour figure online)

represent a residue. Red dots evidences points in which a mutation has been occurred.

The analysis of each map reveals that there is a substantial difference in the PCN of variants. Conversely, there are no appreciable differences in the topological parameters of the mutated variants with respect to those that are preserved. Hence, more deep investigations need to be performed.

5 Conclusion

Protein contact networks (PCNs) are a modelling framework for protein structures. In such a framework, the protein structure is represented as an unweighted graph. Graph embedding methods enable to map of nodes into a latent space, including a set of information to each node, and then to analyze such graph with deep learning methods. In this work, we proposed a framework to perform such analysis and to study the mutation of the S protein of SARS-CoV-2.

Acknowledgements Authors thank Luisa di Paola and Alessandro Giuliani for fruitful discussion and collaboration during the preparation of this paper

Author Contributions For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "Conceptualization, PHG. and PV.; methodology, PHG.; software, BP and UL.; data curation, UL; writing—original draft preparation, PHG, PV, and BP.; writing—review and editing, PHG and PV. Funding acquisition, PV. All authors have read and agreed to the published version of the manuscript."

Data availability Data are available upon request.

References

Agapito G, Guzzi PH, Cannataro M (2019) Parallel extraction of association rules from genomics data. Appl Math Comput 350:434–446

Bittrich S, Rose Y, Segura J, Lowe R, Westbrook JD, Duarte JM, Burley SK (2022) RCSB Protein Data Bank: improved annotation, search and visualization of membrane protein structures archived in the PDB. Bioinformatics 38(5):1452–1454

Das JK, Roy S, Guzzi PH (2021) Analyzing host-viral interactome of SARS-CoV-2 for identifying vulnerable host proteins during COVID-19 pathogenesis. Infect Genet Evol 93:104921

Data61 C (2018) Stellargraph machine learning library. https://github.com/stellargraph/stellargraph. Accessed Sept 2022

Di Paola L, De Ruvo M, Paci P, Santoni D, Giuliani A (2013) Protein contact networks: an emerging paradigm in chemistry. Chem Rev 113(3):1598–1613

Eskandarzade N, Ghorbani A, Samarfard S, Diaz J, Guzzi PH, Fariborzi N, Tahmasebi A, Izadpanah K (2022) Network for network concept offers new insights into host-SARS-CoV-2 protein interactions and potential novel targets for developing antiviral drugs. Comput Biol Med 105575

Eswar N, John B, Mirkovic N, Fiser A, Ilyin VA, Pieper U, Stuart AC, Marti-Renom MA, Madhusudhan MS, Yerkovich B et al (2003) Tools for comparative protein structure modeling and analysis. Nucleic Acids Res 31(13):3375–3380

Gordon DE, Jang GM, Bouhaddou M, Xu J, Obernier K, White KM, O'Meara MJ, Rezelj VV, Guo JZ, Swaney DL et al (2020) A SARS-CoV-2 protein interaction map reveals targets for drug repurposing. Nature 583(7816):459–468

Grover A, Leskovec J (2016) node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pp 855–864

Gu S, Jiang M, Guzzi PH, Milenković T (2022) Modeling multi-scale data via a network of networks. Bioinformatics 38(9):2544–2553

Guzzi PH, Milenković T (2018) Survey of local and global biological network alignment: the need to reconcile the two sides of the same coin. Brief Bioinform 19(3):472–481

Guzzi PH, Zitnik M (2022) Editorial deep learning and graph embeddings for network biology. IEEE/ACM Trans Comput Biol Bioinform 19(2):653–654

Guzzi PH, Mercatelli D, Ceraolo C, Giorgi FM (2020) Master regulator analysis of the SARS-CoV-2/human interactome. J Clin Med 9(4):982

Guzzi PH, Di Paola L, Giuliani A, Veltri P (2022a) Design and development of PCN-miner: a tool for the analysis of protein contact networks. arXiv preprint. arXiv:2201.05434

Guzzi PH, Di Paola L, Giuliani A, Veltri P (2022b) PCN-miner: an open-source extensible tool for the analysis of protein contact networks. Bioinformatics 38(17):4235–4237

Hamilton WL, Ying R, Leskovec J (2017a) Representation learning on graphs: methods and applications. arXiv preprint. arXiv:1709.05584

Hamilton W, Ying Z, Leskovec J (2017b) Inductive representation learning on large graphs. Adv Neural Inf Process Syst 30

Jumper JE, Pritzel A et al (2021) Highly accurate protein structure prediction with AlphaFold. Nature 596:583–589

Khan T, Ghosh I (2015) Modularity in protein structures: study on all-alpha proteins. J Biomol Struct Dyn 33(12):2667–2681

Kukic P, Mirabello C, Tradigo G, Walsh I, Veltri P, Pollastri G (2014) Toward an accurate prediction of inter-residue distances in proteins using 2D recursive neural networks. BMC Bioinform 15(1):1–15

Kumar Das J, Tradigo G, Veltri P, Guzzi PH, Roy S (2021) Data science in unveiling COVID-19 pathogenesis and diagnosis: evolutionary origin to drug repurposing. Brief Bioinform 22(2):855–872

Ortuso F, Mercatelli D, Guzzi PH, Giorgi FM (2021) Structural genetics of circulating variants affecting the SARS-CoV-2 spike/human ace2 complex. J Biomol Struct Dyn 1–11

Ortuso F, Mercatelli D, Guzzi PH, Giorgi FM (2022) Structural genetics of circulating variants affecting the SARS-CoV-2 spike/human ACE2 complex. J Biomol Struct Dyn 40(14):6545–6555

Palopoli L, Rombo SE, Terracina G, Tradigo G, Veltri P (2009) Improving protein secondary structure predictions by prediction fusion. Inf Fusion 10(3):217–232

Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 701–710

Petrey D, Honig B (2005) Protein structure prediction: inroads to biology. Mol Cell 20(6):811–819

Ribeiro LF, Saverese PH, Figueiredo DR (2017) Struc2vec: learning node representations from structural identity. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. KDD '17. ACM, New York, pp 385–394. https://doi.org/10.1145/3097983.3098061

Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q (2015) LINE. In: Proceedings of the 24th international conference on world wide web. International world wide web conferences steering committee. https://doi.org/10.1145/2736277.2741093

Tasdighian S, Di Paola L, De Ruvo M, Paci P, Santoni D, Palumbo P, Mei G, Di Venere A, Giuliani A (2014) Modules identification in protein structures: the topological and geometrical solutions. J Chem Inf Model 54(1):159–168

Zitnik M, Agrawal M, Leskovec J (2018) Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics 34(13):i457–i466. https://doi.org/10.1093/bioinformatics/bty294