Insights into Partial Folding State of Bovine Pancreatic Trypsin Inhibitor: A Combined Molecular Dynamics Simulations, Information Theory and Molecular Graph Theory Study

Hiqmet Kamberaj*

National Institute of Physics, Tirana, Albania and

International Balkan University, Faculty of Engineering,

Department of Computer Engineering,

Makedonsko-Kosovska Brigada BB, Skopje, North Macedonia

(Dated: October 9, 2023)

Abstract

Using a notably large amount of data in investigating physical and chemical phenomena demands new statistical and computational approaches; besides, the cross-validations require well-established theoretical frameworks. This study aims to validate the statistical efficiency of alternative definitions for the information-theoretic measures, such as transfer entropy, using the (α, q) theoretical framework. The primary goal is to find measurements of high-order correlations that preserve information-theoretic properties of information transfer between the components of a dynamical system due to local operations. Besides, this study aims to decode the information contained in the amino acid sequence establishing a three-dimensional protein structure by comparing their physical-chemical properties with their ranked role in the protein interaction network topology using new graph-theoretic measures based on the constructed digraph models of (α, q) information transfer. Moreover, this study aims to use the Deep Graph Convolution Neural Networks for classifying the amino acid role in a protein structure trained upon short equilibrium structure fluctuations at sub-nanosecond time scales.

In particular, this study examines the influence of disulphide bridges on the 3D protein structure of the Bovine Pancreatic Trypsin Inhibitor protein (wild type and its mutated analogue).

Keywords: Molecular Dynamics, Molecular Graphs, Heat and Information Flow, Deep Graph Convolution Neural Networks, (α, q) Transfer Entropy, Gibbs Free Energy

* Corresponding author: H. Kamberaj; Email: h.kamberaj@gmail.com

I. INTRODUCTION

Although slow conformational motions of large biological molecules studies are limited in time and size scales, atomistic models (AMs) combined with the molecular dynamics (MD) technique are widely used to understand the internal atomic motion of (bio)molecular systems [1, 2]. Furthermore, a considerably large amount of the data collected to investigate physical and chemical phenomena will require new statistical and computational approaches to be significantly efficient. Moreover, the cross-validation of these new computational approaches requires vast data collected from the simulations and well-established theoretical frameworks.

The problem of encoding the protein structure into amino acid sequences remains unsolved. Information contained in the amino acid sequences can be classified as overlapping of physical-chemical properties of amino acids. The statistical analysis frameworks have been established [3] (and the references therein) to study the amino acids network associated with the topology of the information flow in proteins. The weighted directed graph model was chosen to describe mathematically the topological network of the anisotropic diffusion of the heat flow in proteins. Furthermore, the information-theoretic measures, such as Shannon transfer entropy [4, 5], are used to describe the geometrical properties of the graphs, which in turn is defined in terms of the equilibrium structure fluctuations at sub-nanoseconds time scales from molecular dynamics simulations.

In this study, we aim to cross-validate the statistical efficiency of alternative definitions for the information-theoretic measures of the transfer entropy between time series, named as (α, q)-framework. The so-called (α, q) measures of mutual information and channel capacities, introduced in Ref. [6], were used. The primary goal is to find measurements of high-order correlations that satisfy information-theoretic properties of information transfer, mainly due to local operations [7]. Besides, they should be easy to compute using the existing algorithms [5, 8]. Furthermore, this study aims to decode the information contained in the amino acid sequence that establishes a protein structure by comparing the physical-chemical properties of amino acids with their role in the protein interaction network topology; for that, new graph-theoretic measures are introduced based on the constructed digraph models of (α, q) information transfer between amino acids [3]. Moreover, this study aims to use the so-called Deep Graph Convolution Neural Networks (DGCNN) for classifying the

amino acid role in a protein structure trained upon short equilibrium structure fluctuations at sub-nanoseconds time scales from molecular dynamics simulations.

In particular, this study aims to examine the influence of disulphide bridges on the 3D protein structure of the Bovine Pancreatic Trypsin Inhibitor protein (wild type and its mutated analogue).

II. MATERIALS AND METHODS

A. Markovian Processes

The Granger causality [9] concept is usually used to characterize the dependence of one variable Y measured over time on another variable X measured synchronously. This concept is initially being used to define the direction of interaction by estimating the contribution of X in predicting Y. However, there exist many other variations of this concept, such as the linear approaches in the time and frequency domain.

The information theory measure of transfer entropy quantifies the statistical coherence between two processes that evolve in time. Schreiber [4] introduced the transfer entropy as the deviation from the independence of the state transition (from the previous state to the next state) of an information destination X from the (previous) state of an information source Y.

To characterize the dynamics of the two random Markovian processes, we will use two vectors. For the first process, the vector is $\mathbf{X}^{\mu_x} = \{\mathbf{x}_k^{\mu_x}\}_{k=k_{0x}}^{T-1}$ where

$$\mathbf{x}_k^{\mu_x} = \left\{x_{k-(m_x-1)\tau_x}, x_{k-(m_x-1)\tau_x+\tau_x}, \cdots, x_{k-\tau_x}, x_k\right\}$$

Here, $k_{0x} = (m_x - 1)\tau_x$, and the dynamics of the second process is characterized by the vector $\mathbf{Y}^{\mu_y} = \{\mathbf{y}_k^{\mu_y}\}_{k=k_{0y}}^{T-1}$:

$$\mathbf{y}_k^{\mu_y} = \left\{y_{k-(m_y-1)\tau_y}, y_{k-(m_y-1)\tau_y+\tau_y}, \cdots, y_{k-\tau_y}, y_k\right\}$$

where $k_{0y} = (m_y - 1)\tau_y$. In the following discussion, $\mathbf{x}_k^{\mu_x+1}$ represents the vector

$$\mathbf{x}_k^{\mu_x+1} = \left\{x_{k-(m_x-1)\tau_x}, x_{k-(m_x-1)\tau_x+\tau_x}, \cdots, x_{k-\tau_x}, x_k, x_{k+\delta}\right\}$$

Similarly, $\mathbf{y}_k^{\mu_y+1}$ represents the vector

$$\mathbf{y}_k^{\mu_y+1} = \left\{y_{k-(m_y-1)\tau_y}, y_{k-(m_y-1)\tau_y+\tau_y}, \cdots, y_{k-\tau_y}, y_k, y_{k+\delta}\right\}$$

Note that m and τ are characteristics of each random process. The choice of m and τ is crucial in order to reconstruct the dynamical structure of the random processes as discussed above. For clarity, we denote $\mu + 1 \equiv (m+1, \tau)$ and $k_0 = \max\left((m_x - 1)\tau_x, (m_y - 1)\tau_y\right)$.

B. The (α, q)-Information Flow

1. *The α-Information Theoretic Measurements*

The so-called Rényi entropy is defined as [10]:

$$H_\alpha(X) = \log\left(\sum_k \left(p(\mathbf{x}_k^{\mu_x})\right)^\alpha\right) \tag{1}$$

which is non-additive from Penrose's viewpoint [11]. Here, $\log(\cdots)$ is the natural logarithm and thus the units are of *nats*. The $\alpha-$mutual information is defined as:

$$\begin{aligned} I_\alpha(X;Y) &= H_\alpha(X) + H_\alpha(Y) - H_\alpha(X,Y) \\ &= H_\alpha(Y) - H_\alpha(Y|X) \end{aligned} \tag{2}$$

which can be negative, in particular for weakly coupled systems and it is not equivalent to the definition (because of non-additivity property of $H_\alpha(X)$):

$$I_\alpha(X;Y) = \frac{1}{1-\alpha}\log\left(\sum_{k_x,k_y}\left[p(\mathbf{x}_{k_x}^{\mu_x}, \mathbf{y}_{k_y}^{\mu_y})\right]^\alpha \left[p(\mathbf{x}_{k_x}^{\mu_x})p(\mathbf{y}_{k_y}^{\mu_y})\right]^{1-\alpha}\right) \tag{3}$$

The $\alpha-$joint entropy is defined as:

$$H_\alpha(X,Y) = \log\left(\sum_{k_x,k_y}\left(p(\mathbf{x}_{k_x}^{\mu_x}, \mathbf{y}_{k_y}^{\mu_y})\right)^\alpha\right) \tag{4}$$

2. *The q-Information Theoretic Measurements*

The so-called Tsallis entropy is defined as [12]:

$$\begin{aligned} H_q(X) &= \frac{1}{1-q}\left(\sum_k \left(p(\mathbf{x}_k^{\mu_x})\right)^q - 1\right) \\ &= \sum_k p(\mathbf{x}_k^{\mu_x}) \log_q\left(\frac{1}{p(\mathbf{x}_k^{\mu_x})}\right) \end{aligned} \tag{5}$$

where $\log_q(\cdots)$ is defined in the following. $H_q(X)$ is also non-additive, based on Penrose's definition [11] and furthermore, it belongs to the class of non-extensive statistical mechanics [13] (and the references therein).

Here, we employed the Tsallis statistics to define a new information-theoretic measure, called $q-$mutual information as:

$$\begin{aligned} I_q(X;Y) &= H_q(X) + H_q(Y) - H_q(X,Y) \\ &= H_q(Y) - H_q(Y|X) \end{aligned} \tag{6}$$

Furthermore, we can also define an q-measure of the so-called Kullback-Leibler divergence as:

$$I_q(X;Y) = \sum_{k_x,k_y} p(\mathbf{x}_{k_x}^{\mu_x}, \mathbf{y}_{k_y}^{\mu_y}) \log_q \frac{p(\mathbf{x}_{k_x}^{\mu_x}, \mathbf{y}_{k_y}^{\mu_y})}{p(\mathbf{x}_{k_x}^{\mu_x}) p(\mathbf{y}_{k_y}^{\mu_y})} \tag{7}$$

Note that these two definitions are not equivalent because, in general, $\log_q(x/y) \neq \log_q(x) - \log_q(y)$, and they are equivalent for $q = 1$ only.

3. (α, q) *Mutual Information*

The so-called (α, q) information theoretic measures have already been formulated by Ilić and Djordjević [6]. In particular, the (α, q) mutual information and (α, q) capacities have been introduced, which are briefly summarised in the following, as in Ref. [6]:

$$\begin{aligned} I_{\alpha,q}(X;Y) &= H_{\alpha,q}(X) + H_{\alpha,q}(Y) - H_{\alpha,q}(X,Y) \\ &= H_{\alpha,q}(X) + H_{\alpha,q}(Y) - H_{\alpha,q}(X) \oplus_q H_{\alpha,q}(Y|X) \\ &= H_{\alpha,q}(X) + H_{\alpha,q}(Y) \\ &- [H_{\alpha,q}(X) + H_{\alpha,q}(Y|X) + (1-q) H_{\alpha,q}(X) H_{\alpha,q}(Y|X)] \\ &= H_{\alpha,q}(Y) - H_{\alpha,q}(Y|X) - (1-q) H_{\alpha,q}(X) H_{\alpha,q}(Y|X) \end{aligned}$$

where

$$\begin{aligned} H_{\alpha,q}(X,Y) &= H_{\alpha,q}(X) \oplus_q H_{\alpha,q}(Y|X) \\ &= H_{\alpha,q}(X) + H_{\alpha,q}(Y|X) + (1-q) H_{\alpha,q}(X) H_{\alpha,q}(Y|X) \\ H_{\alpha,q}(Y|X) &= \frac{H_{\alpha,q}(X,Y) - H_{\alpha,q}(X)}{1 + (1-q) H_{\alpha,q}(X)} = H_{\alpha,q}(X,Y) \ominus_q H_{\alpha,q}(X) \end{aligned}$$

which can also be negative, in particular for weakly coupled systems [7].

Therefore, here we considered another definition [6]:

$$I_{\alpha,q}(X;Y) = H_{\alpha,q}(Y) \ominus_q H_{\alpha,q}(Y|X) = \eta_q(I_\alpha(X;Y))$$

4. (α, q) Transfer Entropy

By definition [4], the Shannon transfer entropy, $T_{Y\to X}$, is given as:

$$T_{Y\to X} = \sum_k p(x_{k+\delta}, \mathbf{x}_k^{\mu_x}, \mathbf{y}_k^{\mu_y}) \log \frac{p(x_{k+\delta} \mid \mathbf{x}_k^{\mu_x}, \mathbf{y}_k^{\mu_y})}{p(x_{k+\delta} \mid \mathbf{x}_k^{\mu_x})} \tag{8}$$

where k is a time index and the information is in units of *nats*. Similarly, the transfer entropy, $T_{X\to Y}$, is given as:

$$T_{X\to Y} = \sum_k p(\mathbf{x}_k^{\mu_x}, \mathbf{y}_k^{\mu_y}, y_{k+\delta}) \log \frac{p(\mathbf{x}_k^{\mu_x}, y_{k+\delta} \mid \mathbf{y}_k^{\mu_y})}{p(y_{k+\delta} \mid \mathbf{y}_k^{\mu_y})} \tag{9}$$

This formulation of the transfer entropy represents a dynamical measure, as a generalization of the entropy rate to more than one element to form a mutual information rate [4]. In Eq. 8, $p(x_{k+\delta}, \mathbf{x}^{\mu_x}, \mathbf{y}_k^{\mu_y})$ is the joint probability distribution of observing future value $x_{k+\delta}$ and the histories of $\mathbf{x}_k^{\mu_x}$ and $\mathbf{y}_k^{\mu_y}$, $p(x_{k+\delta} \mid \mathbf{x}_k^{\mu_x}, \mathbf{y}_k^{\mu_y})$ is the conditional probability of observing the future value of X given the past values of both X and Y, and $p(x_{k+\delta} \mid \mathbf{x}_k^{\mu_x})$ is the conditional probability of observing the future of X knowing its past. From Eq. 8, we can write

$$\begin{aligned} T_{Y\to X} &= H(\mathbf{X}^{\mu_x+1}) - H(\mathbf{X}^{\mu_x}) - H(\mathbf{X}^{\mu_x+1}, \mathbf{Y}^{\mu_y}) + H(\mathbf{X}^{\mu_x}, \mathbf{Y}^{\mu_y}) \\ &\equiv I\left(X_{+\delta}; \mathbf{Y}^{\mu_y} \mid \mathbf{X}^{\mu_x}\right) = H(X_{+\delta} \mid \mathbf{X}^{\mu_x}) - H(X_{+\delta} \mid \mathbf{X}^{\mu_x}, \mathbf{Y}^{\mu_y}) \end{aligned} \tag{10}$$

where $I\left(X_{+\delta}; \mathbf{Y}^{\mu_y} \mid \mathbf{X}^{\mu_x}\right)$ is the conditional mutual information [5, 14, 15]. This indicates that the transfer entropy, $T_{Y\to X}$ can be interpreted as the average amount of information contained in the source about next state $X_{+\delta}$ of the variable X that was not already contained in the past of X.

Similarly, $T_{X\to Y}$ is given as

$$\begin{aligned} T_{X\to Y} &= H(\mathbf{Y}^{\mu_y+1}) - H(\mathbf{Y}^{\mu_y}) - H(\mathbf{X}^{\mu_x}, \mathbf{Y}^{\mu_y+1}) + H(\mathbf{X}^{\mu_x}, \mathbf{Y}^{\mu_y}) \\ &\equiv I\left(Y_{+\delta}; \mathbf{X}^{\mu_x} \mid \mathbf{Y}^{\mu_y}\right) = H(Y_{+\delta} \mid \mathbf{Y}^{\mu_y}) - H(Y_{+\delta} \mid \mathbf{X}^{\mu_x}, \mathbf{Y}^{\mu_y}) \end{aligned} \tag{11}$$

Here, we define (α, q) transfer entropies as:

$$\begin{aligned} T^{(\alpha,q)}_{Y\to X} &= I_{(\alpha,q)}\left(X_{+\delta}; \mathbf{Y}^{\mu_y} \mid \mathbf{X}^{\mu_x}\right) = H_{(\alpha,q)}(X_{+\delta} \mid \mathbf{X}^{\mu_x}) \ominus_q H_{(\alpha,q)}(X_{+\delta} \mid \mathbf{X}^{\mu_x}, \mathbf{Y}^{\mu_y}) \\ &= \left[H(\mathbf{X}^{\mu_x+1}) \ominus_q H(\mathbf{X}^{\mu_x})\right] \ominus_q \left[H_{(\alpha,q)}(\mathbf{X}^{\mu_x+1}, \mathbf{Y}^{\mu_y}) \ominus_q H_{(\alpha,q)}(\mathbf{X}^{\mu_x}, \mathbf{Y}^{\mu_y})\right] \\ T^{(\alpha,q)}_{X\to Y} &= I_{(\alpha,q)}\left(Y_{+\delta}; \mathbf{X}^{\mu_x} \mid \mathbf{Y}^{\mu_y}\right) = H_{(\alpha,q)}(Y_{+\delta} \mid \mathbf{Y}^{\mu_y}) \ominus_q H_{(\alpha,q)}(Y_{+\delta} \mid \mathbf{X}^{\mu_x}, \mathbf{Y}^{\mu_y}) \\ &= \left[H_{(\alpha,q)}(\mathbf{Y}^{\mu_y+1}) \ominus_q H_{(\alpha,q)}(\mathbf{Y}^{\mu_y})\right] \ominus_q \left[H_{(\alpha,q)}(\mathbf{X}^{\mu_x}, \mathbf{Y}^{\mu_y+1}) \ominus_q H_{(\alpha,q)}(\mathbf{X}^{\mu_x}, \mathbf{Y}^{\mu_y})\right] \end{aligned} \tag{12}$$

Here,

$$\eta_q(x) = \frac{\exp\left((1-q)x\right) - 1}{1-q} \tag{13}$$

$$x \ominus_q y = \frac{x-y}{1+(1-q)y} \tag{14}$$

$$x \oplus_q y = x + y + (1-q)xy \tag{15}$$

5. (α, q) *Local Transfer Entropy*

The so-called local Shannon transfer entropy measure is defined as [16]:

$$t_{Y\to X}(k+\delta) = h(x_{k+\delta} \mid \mathbf{x}_k^{\mu_x}) - h(x_{k+\delta} \mid \mathbf{x}_k^{\mu_x}, \mathbf{y}_k^{\mu_y}) \tag{16}$$

where the local condition entropy are as the following:

$$\begin{aligned} h(x_{k+\delta} \mid \mathbf{x}_k^{\mu_x}, \mathbf{y}_k^{\mu_y}) &= -\left(\log p(\mathbf{x}_k^{\mu_x+1}, \mathbf{y}_k^{\mu_y}) - \log p(\mathbf{x}_k^{\mu_x}, \mathbf{y}_k^{\mu_y})\right) \\ h(x_{k+\delta} \mid \mathbf{x}_k^{\mu_x}) &= -\left(\log p(\mathbf{x}_k^{\mu_x+1}) - \log p(\mathbf{x}_k^{\mu_x})\right) \end{aligned} \tag{17}$$

Similarly, for the local transfer entropy, $t_{X\to Y}(k)$, can be obtained

$$t_{X\to Y}(k+\delta) = h(y_{k+\delta} \mid \mathbf{y}_k^{\mu_x}) - h(y_{k+\delta} \mid \mathbf{x}_k^{\mu_x}, \mathbf{y}_k^{\mu_y}) \tag{18}$$

where

$$\begin{aligned} h(y_{k+\delta} \mid \mathbf{x}_k^{\mu_x}, \mathbf{y}_k^{\mu_y}) &= -\Big(\log p(\mathbf{x}_k^{\mu_x}, \mathbf{y}_k^{\mu_y+1}) - \log p(\mathbf{x}_k^{\mu_x}, \mathbf{y}_k^{\mu_y})\Big) \\ h(y_{k+\delta} \mid \mathbf{y}_k^{\mu_x}) &= -\Big(\log p(\mathbf{y}_k^{\mu_y+1}) - \log p(\mathbf{y}_k^{\mu_y})\Big) \end{aligned} \tag{19}$$

Here, we introduce to new measures of the local transfer entropy, named as $q-$local transfer entropy:

$$t^{(q)}_{X\to Y}(k+\delta) = h_q(y_{k+\delta} \mid \mathbf{y}_k^{\mu_x}) - h_q(y_{k+\delta} \mid \mathbf{x}_k^{\mu_x}, \mathbf{y}_k^{\mu_y}) \tag{20}$$

where

$$h_q(y_{k+\delta} \mid \mathbf{x}_k^{\mu_x}, \mathbf{y}_k^{\mu_y}) = \log_q \left(\frac{1}{p(\mathbf{x}_k^{\mu_x}, \mathbf{y}_k^{\mu_y+1})} \right) + \log_q \left(\frac{1}{p(\mathbf{x}_k^{\mu_x}, \mathbf{y}_k^{\mu_y})} \right) \tag{21}$$

$$h_q(y_{k+\delta} \mid \mathbf{y}_k^{\mu_x}) = \log_q \left(\frac{1}{p(\mathbf{y}_k^{\mu_y+1})} \right) + \log_q \left(\frac{1}{p(\mathbf{y}_k^{\mu_y})} \right)$$

and

$$\log_q x = \frac{x^{1-q} - 1}{1 - q} \tag{22}$$

Following Refs. [17–19], it can be shown that for processes close to the equilibrium, both reversible or irreversible, at constant temperature (see also [20]):

$$t_{X \to Y}^{(q)}(k + \delta) = -\frac{q_{X \to Y}^{\text{ext}}}{k_B T} \tag{23}$$

where $q_{X \to Y}^{\text{ext}}$ is the heat flow between the system Y and the surrounding in the context of the source X. The sign minus indicates the heat flow into the physical system Y attributed to the surrounding X and the local transfer entropy is the opposite. Eq. 24 gives the minimum amount of energy that a physical system (characterized by the dynamical component Y) must dissipate into the environment (characterized by the dynamical component X) to gain one bit of information. If $t_{X \to Y}^{(q)}(k + \delta) > 0$, and hence the physical system Y gains information, then the heat flows to the surrounding X from the system Y, $q_{X \to Y}^{\text{ext}} < 0$. In that case, a measure of the source X improves our predictions about the macroscopic transition state of Y. For thermally isolated transitions, that is $q_{X \to Y}^{\text{ext}} = 0$, then $t_{X \to Y}^{(q)}(k + \delta) = 0$, indicating that Y and X are independent, in other words, X and Y do not interact with each other.

This study concerns about the random walks of the heat flow in a graph; therefore, the direction of the heat flow between any two vertices X and Y is of interest. For that, the following relationship between the local information flow direction and the heat flow direction is defined

$$d_{X \to Y}^{(q)}(k + \delta) = t_{X \to Y}^{(q)}(k + \delta) - t_{Y \to X}^{(q)}(k + \delta) = -\frac{\Delta q_{X \to Y}^{\text{ext}}}{k_B T} \tag{24}$$

If $d_{X \to Y}^{(a,q)} < 0$, then $\Delta q_{X \to Y}^{\text{ext}} > 0$ and thus heat flow direction is from source X to target Y.

III. MOLECULAR DYNAMICS SIMULATIONS OF SYSTEM MODELLING

As initial structure for molecular dynamics simulations, the Bovine Pancreatic Trypsin Inhibitor (BPTI) with PDB ID 5PTI was considered. It contains 58 amino acids and its structure is investigated by joint X-ray and neutron experiments at a resolution of 1.00 Å [21]. The structure contains three disulphide bonds, namely Cys5-Cys55, Cys14-Cys38, and Cys30-Cys51, as depicted in Fig. 1.

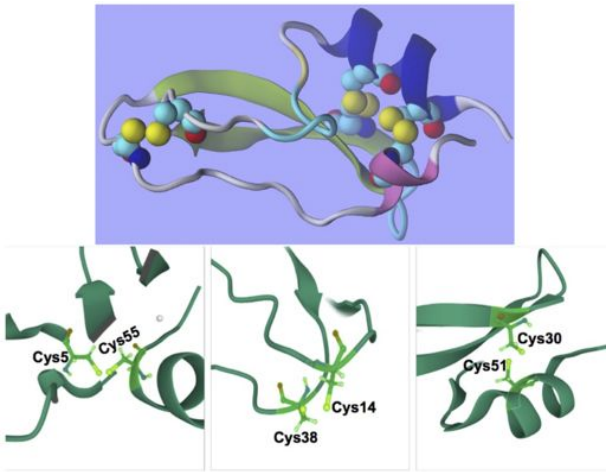

FIG. 1: Three-dimensional structure and disulphide bonds.

Next, a partially folded, namely BPTI[14-38]Abu, mutated-type [22] with PDB ID 1YKT was considered at a resolution of 1.70 Å. BPTI[14-38]Abu has a disulphide bond Cys14-Cys38, and Cys5, Cys30, Cys51, and Cys55 are mutated to isometric α-amino-n-butyric acid. The X-ray structure misses the amino acids Glu57 and Ala58, which are modelled using a three-dimensional structure alignment with BPTI wild-type.

CHARMM program [23] using the CHARMM36 force field [24–26] is used for MD simulations. Figure 2 shows the topology of CHARMM36 force field for isometric α-amino-n-butyric acid and partial charges calculated using quantum mechanics in the gas phase. Other force field parameters, such as bonds, angle bending, torsions, and van der Waals, were chosen based on the atom type as in CHARMM36 [24–26].

The hydrogen atoms and the protonation state of amino acids were determined using H++ web server. TIP3P model water shell is added around the protein with a salt (NaCl) concentration of 0.154 M/l, plus the counter ions to neutralise the systems. Equilibration

```
RESI ABU          0.00
GROUP
ATOM N    NH1    -0.47  !      |
ATOM HN   H       0.31  !  HN-N
ATOM CA   CT1     0.07  !      |   HB1 HG1
ATOM HA   HB1     0.09  !      |   |   |
GROUP                   !  HA-CA--CB--CG--HG3
ATOM CB   CT2    -0.18  !      |   |   |
ATOM HB1  HA2     0.09  !      |   HB2 HG2
ATOM HB2  HA2     0.09  !   O=C
GROUP                   !      |
ATOM CG   CT2    -0.18
ATOM HG1  HA2     0.06
ATOM HG2  HA2     0.06
ATOM HG3  HA2     0.06
GROUP
ATOM C    C       0.51
ATOM O    O      -0.51
```

FIG. 2: The topology of CHARMM36 force field for isometric α-amino-n-butyric acid.

protocol of the solvent and hydrogen atoms (with heavy atoms of proteins fixed at their X-ray structures) included energy minimisation (steepest descents and conjugate gradients algorithms [27] for about $5 \times 10^3 - 10 \times 10^3$ steps each), heating for about 32×10^3 (N; V; T) MD steps (from 50 K to 800 K in steps of 50 K) and cooling for about 40×10^3 (N; V; T) MD steps (from 800 K to 300 K in steps of 25 K). Next, the restraints (N; P; T) MD simulations at $T = 300$ K; $P = 1$ atm were performed for about 500 ps for each system to equilibrate both solvent and protein by gently lowering the restraints until the density of the solvent converged to the value of 1 gram/cm^3. The (N; P; T) MD simulations at $T = 300$ K; $P = 1$ atm continued for about 2.5 ns for each system without any restraints to finalise the equilibration stage. The production run (N; P; T) MD simulations at $T = 300$ K; $P = 1$ atm were performed for ten ns for each system. In total, about 30 ns of simulations and around 3 months of CPU time (4 CPUs, 2.8 GHz clock rate). A cutoff radius of $R_{\rm cut} = 12$ Å was used for non-bonded long-range interactions and the SHAKE algorithm [28] to constraint the covalent bonds involving the hydrogen atoms, allowing using a numerical integration time-step of $\Delta t = 2$ fs.

Particle-Mesh-Ewald was employed for long-range electrostatics and the potential shift for handling the short-range van der Waals interactions [20]. (Note that the van der Waals potential and the forces for this choice approach zero as the intra-distances become $R_{\rm cut}$, avoiding difficult computations of switching functions and steep forces at large distances, as suggested in Ref. [23]). A fixed dielectric constant of unity was taken and simple potential cutoff approach was used for handling the short-range electrostatic interactions. The Ewald summation was performed with $\kappa = 4.0/R_{\rm cut}$, as a trade-off between the computation cost

and accuracy [20].

The cubic simulation box was used for the periodic boundary conditions, the neighbour list algorithm, based on the atom list, with a spherical radius of 13 Å was used for keeping the non-bonded list of atoms, and the cutoff radius of the nearest neighbour is 11 Å. Here, a heuristic testing is done every time the energy and forces are calculated and a list update is done if necessary. (Note that this is both safer and more economical than frequency-updating [23]; furthermore, BYCBIM algorithm was employed to update the neighbour lists). MD structures were saved every 0.2 ps; for analysis, every 1 ps. Wild-type has around 19562 atoms and BPTI[14-38]Abu around 18244 atoms.

IV. RESULTS AND DISCUSSION

A. Free Energy Change Upon Mutations

Figure 3 presents the thermodynamic cycle for calculation of binding free energy of an amino acid in a protein. The differences in the solvation free energy ($\Delta G_{(k)}^{(\text{solvation})}$, $\Delta G_{(-k)}^{(\text{solvation})}$, and $\Delta G^{(\text{solvation})}$) in reference surrounding phase (often gas phase) and in a solvent (such as water and salt) are computed using Poisson-Boltzmann (PB) method [20] implemented in CHARMM [23]. Here, $\Delta G_k^{(\text{b,solv})}$ and $\Delta G_k^{(\text{b,ref})}$ denotes the binding free energies of the amino acid k in the solvent and gas phase, respectively. $\Delta G_k^{(\text{b,solv})}$ is computed as:

$$\Delta G_k^{(\text{b,solv})} = \Delta G_k^{(\text{b,ref})} + \Delta G_k^{(\text{solvation})} \tag{25}$$

where

$$\begin{aligned} \Delta G_k^{(\text{b,ref})} &= H^{(\text{MM})} - H_{(-k)}^{(\text{MM})} - T\Delta S_k \\ \Delta G_k^{(\text{solvation})} &= \Delta G^{(\text{solvation})} - \Delta G_{(-k)}^{(\text{solvation})} \end{aligned} \tag{26}$$

Here, $H^{(\text{MM})}$ and $H_{(-k)}^{(\text{MM})}$ represent the potential interaction energy of the entire system and subsystem without amino acid k, based on the CHARMM36 molecular force field. Furthermore, ΔS_k is the change on the entropy with T being the temperature, which is decomposed on three terms, namely rotational, translational and configurational entropy terms, implemented in CHARMM [23]. Note that here, Andricioaei & Karplus formula is used to compute configurational entropy [29, 30].

Moreover, $\Delta G_k^{(\text{solvation})}$ is decomposed into the polar (or electrostatic) and non-polar (or solvent-accessible-surface area, SASA) terms as:

$$\Delta G_k^{(\text{solvation})} = \Delta G_{\text{polar},k}^{(\text{solvation})} + \Delta G_{\text{SASA},k}^{(\text{solvation})} \tag{27}$$

In Eq. 27, $\Delta G_{\text{polar},k}^{(\text{solvation})}$ is calculated using PB method and $\Delta G_{\text{SASA},k}^{(\text{solvation})}$ is parametrised as

$$\Delta G_{\text{SASA},k}^{(\text{solvation})} = \gamma A_k \tag{28}$$

where A_k is the value of SASA for the amino acid k present in the protein and $\gamma = 0.006$ kcal $\times$ mol$^{-1} \times$ Å^{-2}. The coefficient γ and parametrisation of the surface effects are designed such to include short-range effects, dominated by the first solvation shell [31, 32]. There is wide range of the values of γ because of different models used to represent the molecular surface [31]. However, small values of γ indicate that molecular surface contribution has only a small impact on the free energy changes due to large-scale conformation changes. Note that all the computed quantities are time averaged ones over different configurations obtained from MD simulations; typically, 2000 configurations were used.

The calculations were done for each system (BPTI-WT and BPTI[14-38]ABU), then the differences between free energy changes due to mutations are calculated as:

$$\Delta\Delta G_{k\to k'}^{(\text{b,solv})} = \Delta G_{k'}^{(\text{b,solv})} - \Delta G_k^{(\text{b,solv})} \tag{29}$$

where k is the amino acid in BPTI-WT and k' is the amino acid with the same ID in BPTI[14-38]ABU.

Figure 5 shows graphically the results for $\Delta\Delta G_{k\to k'}^{(\text{b,solv})}$. In the calculations of the electrostatic Poisson-Boltzmann term the dielectric constants of the protein and water are, respectively, $\epsilon_p = 2$ and $\epsilon_w = 80$. The salt concentration was 0.154 M/l. From our calculations, mutation of Cys5 and Cys55 to ABU resulted in favour of the stability of the structure, respectively, $\Delta\Delta G_{\text{Cys5}\to\text{Abu5}}^{(\text{b,solv})} = -3.185 \pm 0.484$ kcal/mol and $\Delta\Delta G_{\text{Cys55}\to\text{Abu55}}^{(\text{b,solv})} = -3.701 \pm 0.470$ kcal/mol. On the other hand, the mutations of Cys30 and Cys51 to Abu favoured much less the stability of the BPTI[14-38]ABU; in particular, $\Delta\Delta G_{\text{Cys30}\to\text{Abu30}}^{(\text{b,solv})} = -0.724 \pm 0.463$ kcal/mol and $\Delta\Delta G_{\text{Cys51}\to\text{Abu51}}^{(\text{b,solv})} = -0.447 \pm 0.551$ kcal/mol. Interestingly, the most favoured mutation in enhancing the stability of BPTI[14-38]ABU structure is Ala27→Asp27 with a value of $\Delta\Delta G_{\text{Ala27}\to\text{Asp27}}^{(\text{b,solv})} = -13.784 \pm 0.441$ kcal/mol. In contrast, the mutation of Lys26 to Pro26 disfavoured the stability with a value of $\Delta\Delta G_{\text{Lys26}\to\text{Pro26}}^{(\text{b,solv})} = 2.856 \pm 0.606$ kcal/mol;

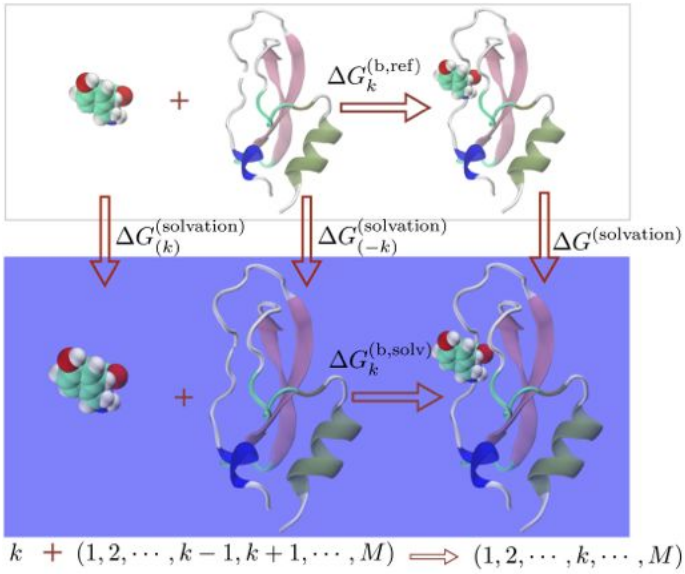

FIG. 3: Thermodynamic cycle for calculation of binding free energy of an amino acid in a protein. The differences in the solvation free energy, $\Delta G_{(k)}^{(\mathrm{solvation})}$, $\Delta G_{(-k)}^{(\mathrm{solvation})}$, and $\Delta G^{(\mathrm{solvation})}$ in reference surrounding phase (often gas phase) and in solvent (such as water and salt).

similarly the mutation of Arg15 to Lys15 with a value of $\Delta\Delta G_{\mathrm{Arg15\to Lys15}}^{(\mathrm{b,solv})} = 2.377 \pm 0.939$ kcal/mol.

Furthermore, the mutation of Ala11 to Thr11 favoured the stability of BPTI[14-38]ABU structure with a change on the free energy of -3.708 ± 0.400 kcal/mol. It is interesting to note that other amino acids, which are not involved in the mutation, contribute to the stability of the protein structure. This indicates that the interaction network of the amino acids has changed upon the mutations. Therefore, the following graph network analysis of the information flow enables identification of the changes in the amino acids interaction networks of both proteins. The aim is to reveal the encoded information of the protein structure into amino acid sequence using the information-theoretic measures.

B. (a, q)-Information Flow

Next, the information flow direction is calculated based on the (a, q)-transfer entropies:

$$D_{i\to j}^{(a,q)} = T_{i\to j}^{(a,q)} - T_{j\to i}^{(a,q)}, \quad i, j = 1, 2, \cdots, M \tag{30}$$

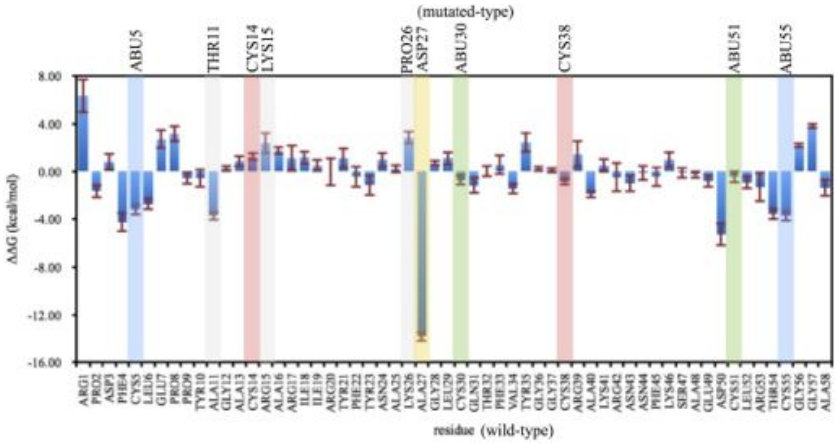

FIG. 4: The free energy changes for each amino acid upon mutation from protein BPTI-WT to BPTI[14-38]ABU.

where M is the number of amino acids in the structure. Here, if $D_{i\to j}^{(a,q)} > 0$, then information flows from source i to target j, or i influences j. Figure 5 presents heat map plots of $D_{i\to j}^{(a,q)}$ (in nats) for BPTI-WT and BPTI[14-38]Abu, where a red colour indicates a strong source of information and blue one indicates a target amino acid of information flow. Our results clearly identify the source and target residues and the change in propensity of an amino acid to being a source or target of information flow for BPTI-WT and BPTI[14-38]Abu.

In Fig. 6, we have plotted the differences in information flow for each amino acid in BPTI wild-type and BPTI[14-38]Abu structures (see Fig. 6(a)) calculated as:

$$\Delta D_{i\to j}^{(a,q)} = D_{i\to j}^{(a,q),\mathrm{wt}} - D_{i\to j}^{(a,q),\mathrm{mt}}, \quad i,j = 1,2,\cdots,M \tag{31}$$

Here, if $\Delta D_{i\to j}^{(a,q)} > 0$, then amino acid i has a greater propensity to be a source of information than j in wild-type structure compared with mutated protein, which is indicated in Fig. 6(a) by a red colour; otherwise, if $\Delta D_{i\to j}^{(a,q)} < 0$, then i has a greater propensity to be a target of information than j in wild-type structure compared with mutated protein, indicated in Fig. 6(a) by a blue colour. In Fig. 6, we have depicted the amino acids with greater positive impact as a source of information flow, such as Ala13, Phe22, Tyr23, Leu29, Gly37, Thr54, and Cys55(wild-type)/Abu55(mutated-type). In contrast, the amino acids with greater positive impact as a target of information flow include: Arg1, Phe4, Ala11(wild-type)/Thr11(mutated-type), Gly12, Cys14, Ala16, Gln31, Gly36, Cys38, and Asp50.

We also calculated the mean-square-deviation (msd) of $D_{i\to j}^{(a,q)}$ between BPTI-WT and

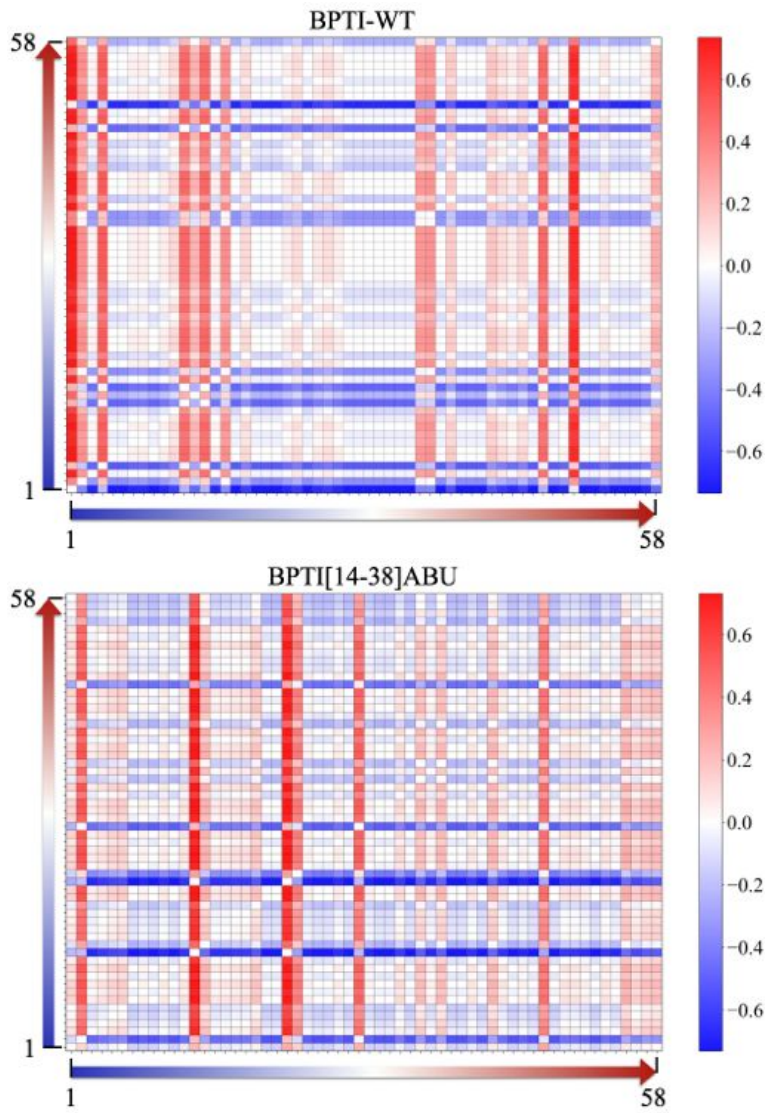

FIG. 5: The information flow for BPTI wild-type and BPTI[14-38]Abu.

BPTI[14-38]Abu as:

$$msd(i) = \frac{1}{M}\sum_{j=1}^{M}\left(D_{i\to j}^{(a,q),\mathrm{wt}} - D_{i\to j}^{(a,q),\mathrm{mt}}\right)^2 \tag{32}$$

The results are presented graphically in Fig. 6(b), where one can clearly identify the amino acids with the largest deviation in $D_{i\to j}^{(a,q)}$ from wild-type to mutated-type protein. Interestingly, there is a noticeable correlation between the msd and the changes on the free energies, indicating that the information-theoretic measure $D_{i\to j}^{(a,q)}$ (based on the fluctuations of encoded information) captures the nature of interactions within a protein structure, which are

highly non-linear based on the physics of molecular force field.

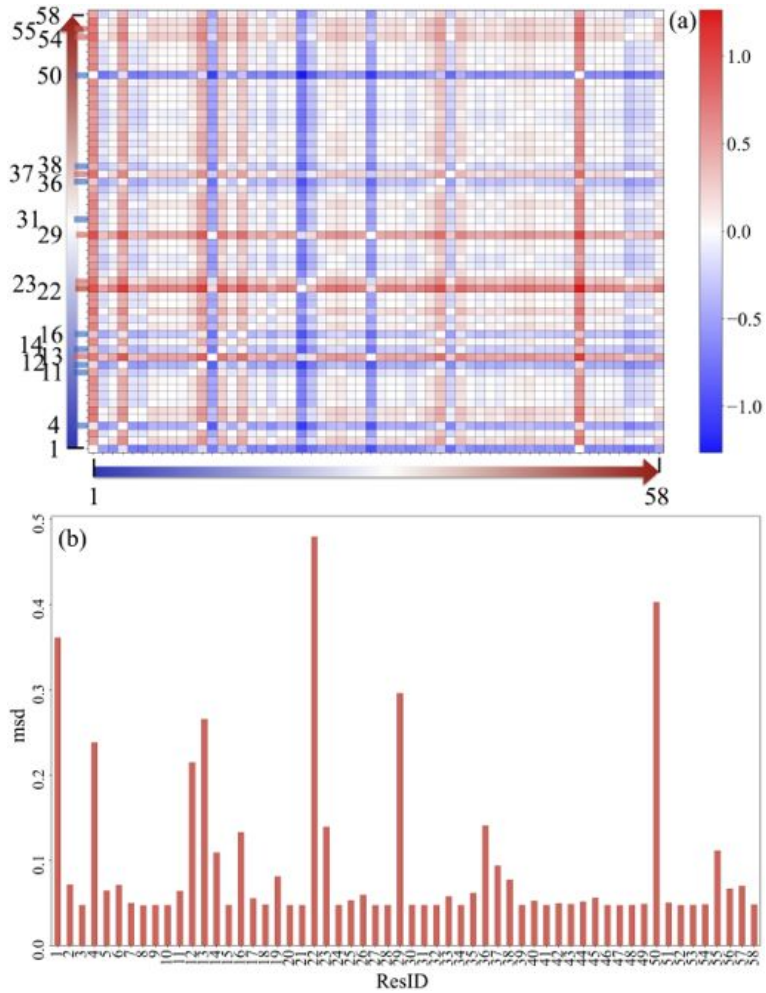

FIG. 6: (a) The information flow difference between BPTI wild-type and BPTI[14-38]Abu; (b) The msd of the information flow difference between BPTI wild-type and BPTI[14-38]Abu.

C. Amino Acids Ranking

Next, we considered graph-theoretic measures (such as PageRank, authority, and hub) as measures of the impact of the amino acids in the protein structure networks, constructed using information flow direction $D_{i\to j}^{(a,q)}$. For the PageRank, the algorithm introduced in Ref. [3] is used. For the authority and hub rank indices, a new algorithm is developed based on the so-called Google matrix, computed as in Ref. [3]. The algorithm uses the Google matrix L as an input, then iteratively the authority (a_i, for $i = 1, 2, \cdots, n$) and hub (h_i, for

$i = 1, 2, \cdots, n$) are computed for each vertex i running over all vertices of the graph G:

$$\mathbf{h}_{t+1} = \mathbf{L}^T \mathbf{a}_t \tag{33}$$

$$\mathbf{a}_{t+1} = \mathbf{L}\mathbf{h}_t, \quad t = 0, 1, \cdots, N_{\text{iter}} \tag{34}$$

where $()^T$ stands for the transpose, t is the iteration index, and N_{iter} is the number of iterations (maximum 100). At each iteration step, the vectors $\mathbf{a} = (a_1, a_2, \cdots, a_n)^T$ and $\mathbf{h} = (h_1, h_2, \cdots, h_n)^T$ are normalised to one. Here, the authority index indicates the rank of an amino acid as a source of information flow through the graph and the hub index ranks the vertices (and this the amino acids) based on their role as mediators of the information flow from the source to the target.

Our results of the PageRank and Hub ranks for each amino acid of BPTI wild-type and BPTI[14-38]Abu are shown graphically in Fig. 7. The PageRank and Hub indices showed a very interesting propensity to their values compared with the changes in the free energy from wild-type to the mutated-type. In particular, an increase in the PageRank index, compare with other residues, from BPTI-WT to BPTI[14-38]Abu resulted in a favourable free energy change of that amino acid in mutated structure. For instance, mutation of Ala27 to Asp27 resulted in an increase of the PageRank index and a favourable free energy change. Similarly, for amino acids with residue IDs 4, 11, 14, 15, 26, 38, and 50, as indicated in Fig. 7. Analogue tendency can be observed with the increase of the Hub index; in particular, the increase of the Hub rank, compared with other residues in the same structure, for the residues with ID 4, 5, 6, 30, 54, and 55 from wild- to mutated-type protein correlated with the change on the free energies. That might have a significant impact on prediction of those amino acids that their mutation could result in a favourable free energy change, which is directed to their cooperative contribution within the structure due to their role as mediator of information flow in protein network of amino acids.

The authority index of amino acids showed strong correlation with PageRank index, as presented in Fig. 9.

D. Amino Acids Information Flow Pathways

Graph-theoretic measures can also be used to identify the shortest and best paths between a source and target amino acid. Here, the algorithm employed in Ref. [15] is used

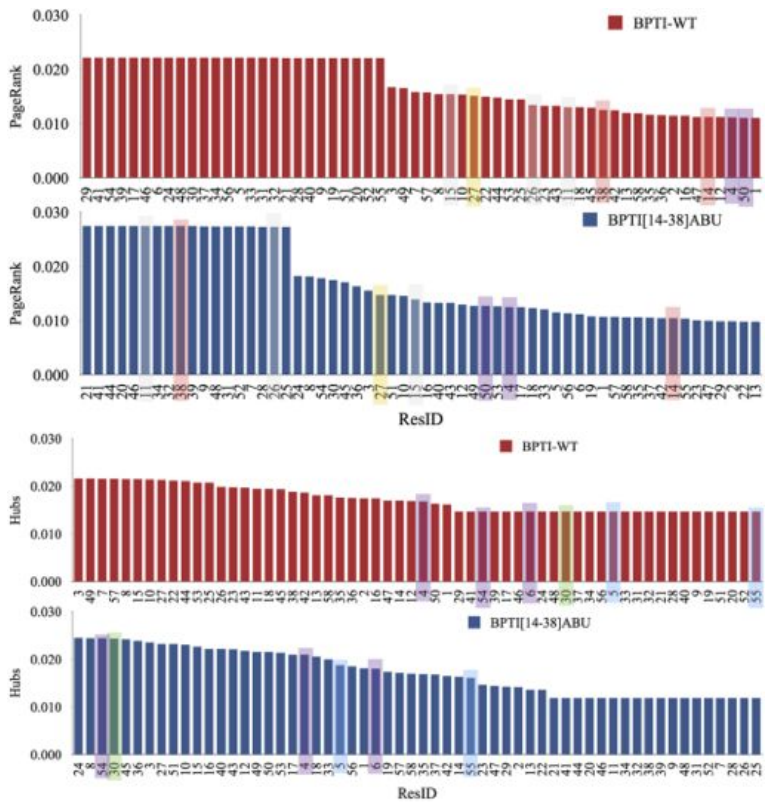

FIG. 7: The PageRank and Hub ranking for each amino acid of BPTI wild-type and BPTI[14-38]Abu.

to determine shortest/best paths of information flow in a graph. In Fig. 9, we present the shortest/best paths of amino acids between residues Cys14 and Cys38 for BPTI wild-type and BPTI[14-38]Abu. The algorithm given in Ref. [33] was used, as implemented in Ref. [15]. The paths are identified by giving the first and the last amino acids, representing, respectively, the target and the source of information; usually this is determined by comparing the PageRank indices, PageRank(first) $<$ PageRank(last), or information flows from the last to the first. Figure 9 shows different paths existing between a target and source amino acid for BPTI-WT and BPTI[14-38]Abu. All the intermediate residues serve as hubs (that is, mediator of information flow from the source to target). It is interesting to note that the paths for wild- and mutated-type proteins are passing through the same number of hubs; however, the role of hubs is undertaken by different amino acids.

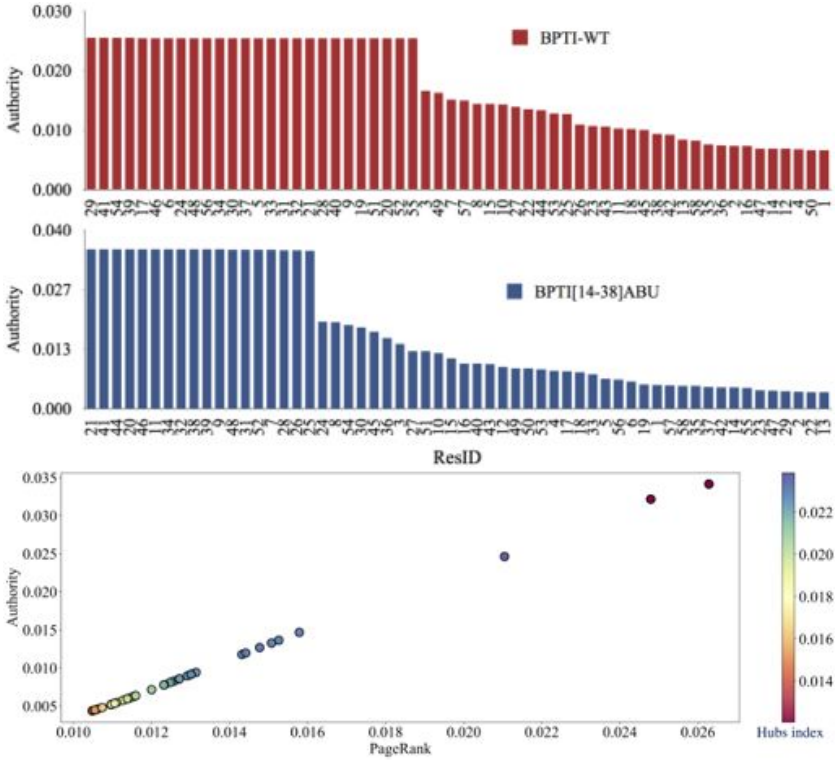

FIG. 8: The authority ranking for each amino acid of BPTI wild-type and BPTI[14-38]Abu.

E. Machine Learning of Amino Acids Ranks

1. *Heat Flow Kernel DiGraph Embedding*

A directed graph $G(V, E, \mathbf{A})$ with M vertices, $V = \{v_1, v_2, \cdots, v_M\}$, and m edges, $E = \{e_1, e_2, \cdots, e_n\}$, where $e_i = (u, v)$ is a directed edge from u to v characterised by the weight W_{uv} and an adjacency matrix $\mathbf{A}$:

$$A_{uv} = \begin{cases} W_{uv} & \text{if } (u, v) \in E \\ 0 & \text{otherwise} \end{cases} \tag{35}$$

Next, two diagonal matrices can be determined to characterise the in- and out-degrees of vertices, denoted $\mathbf{D}^{(\text{in})}$ and $\mathbf{D}^{(\text{out})}$, respectively, as:

$$\mathbf{D}^{(\text{in})} = diag\left(d_1^{(\text{in})}, \cdots, d_u^{(\text{in})}, \cdots, d_M^{(\text{in})}\right), \quad d_u^{(\text{in})} = \sum_{v=1}^{M} A_{vu} \tag{36}$$

$$\mathbf{D}^{(\text{out})} = diag\left(d_1^{(\text{out})}, \cdots, d_u^{(\text{out})}, \cdots, d_M^{(\text{out})}\right)^T, \quad d_u^{(\text{out})} = \sum_{v=1}^{M} A_{uv} \tag{37}$$

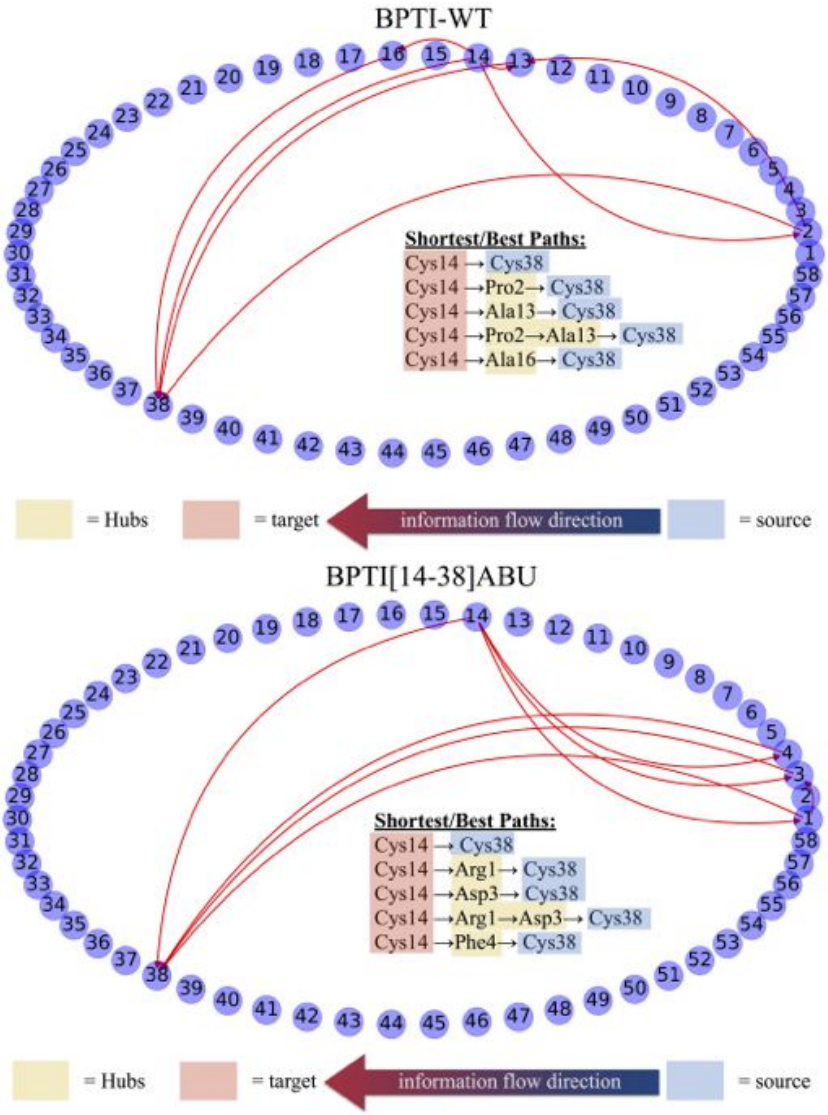

FIG. 9: The shortest/best paths of amino acids between residues Cys14 and Cys38 for BPTI wild-type and BPTI[14-38]Abu. The algorithm given in Ref. [33] and implemented in [15] was used.

Then, the transition probability matrix $\tilde{\mathbf{P}}$ of rank $M \times M$ of a random walk on a digraph is given as:

$$\tilde{\mathbf{P}} = \left[\mathbf{D}^{(\mathrm{out})}\right]^{-1} \mathbf{A} \tag{38}$$

For a strongly connected digraph the random walk characterised by $\tilde{\mathbf{P}}$ is an irreducible and aperiodic Markovian transition process, which is satisfied if there are no vertices with $d_u^{(\mathrm{out})} = 0$ [3]. To guarantee the irreducibility and aperiodicity, the following transition

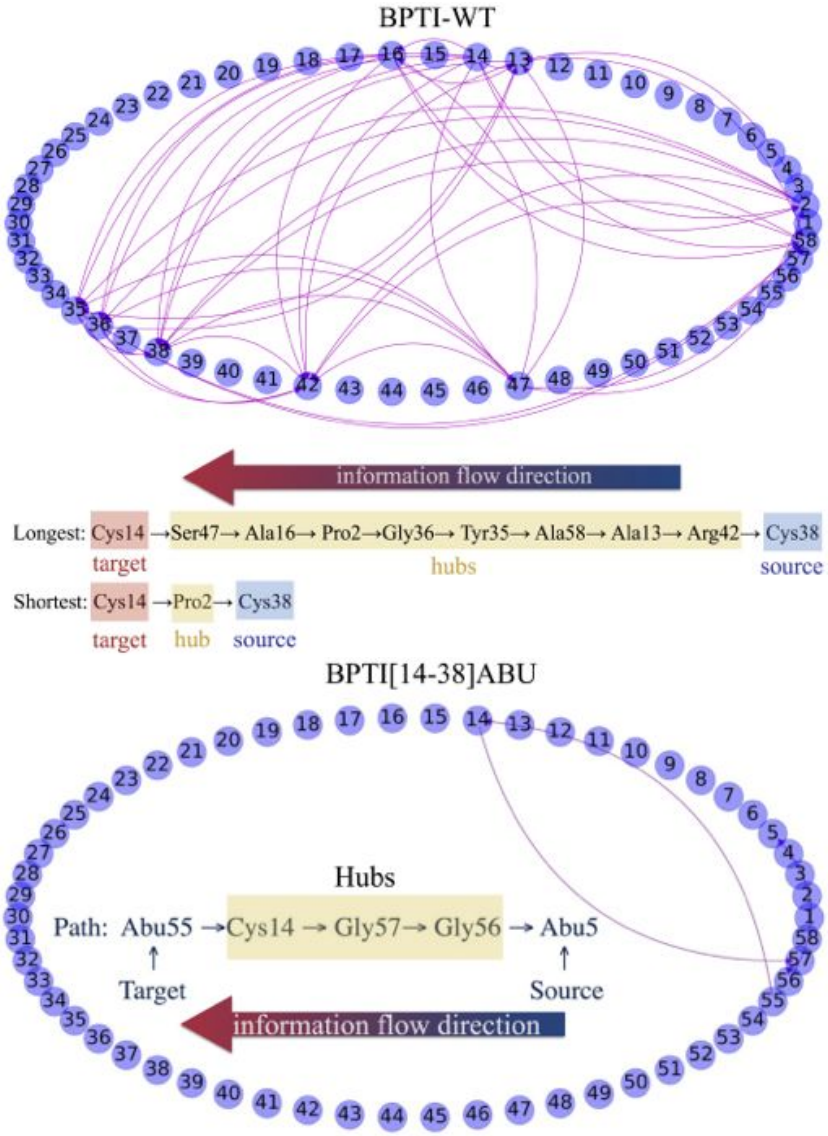

FIG. 10: A long path of amino acids between residues Cys14 and Cys38 for BPTI wild-type and a short path between residues Abu55 and Abu5 for BPTI[14-38]Abu.

probability matrix is suggested

$$\mathbf{P} = (1-\alpha)\left[\mathbf{D}^{(\mathrm{out})}\right]^{-1}\mathbf{A} + \alpha\mathbf{J} \tag{39}$$

where $\mathbf{J}$ is a jumping matrix given in terms of unitary diagonal matrix $\mathbf{1}_M$ of rank $M \times M$ as:

$$\mathbf{J} = \frac{1}{M}\mathbf{1}_M \tag{40}$$

Here, α is a number between 0.1 and 0.2 [3] (and the references therein). Then, the normalised Laplacian matrix is constructed as:

$$\begin{aligned}\mathbf{L} &= \left[\mathbf{D}^{(\text{out})}\right]^{-1/2}\left[\mathbf{D}^{(\text{out})} - \mathbf{D}^{(\text{out})}\mathbf{P}\right]\left[\mathbf{D}^{(\text{out})}\right]^{-1/2} \\ &= \left[\mathbf{D}^{(\text{out})}\right]^{+1/2}\left[\mathbf{1}_M - \mathbf{P}\right]\left[\mathbf{D}^{(\text{out})}\right]^{-1/2}\end{aligned} \tag{41}$$

Spectral decomposition of $\mathbf{L}$ matrix is given as:

$$\mathbf{L} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^T \tag{42}$$

where $\mathbf{E}$ is the matrix of ordered eigenvectors, $\mathbf{e}_u = (E_{1u}, \cdots, E_{Mu})^T$ as columns and $\mathbf{\Lambda}$ is a diagonal matrix of ordered eigenvalues ($\lambda_1 < \lambda_2 < \cdots < \lambda_M$), which are all non-negative.

The vertices of the digraph G are mapped into a vector space using the heat kernel such that for each vertex we determine a so-called embedded vector $\mathbf{z}_u^{(t)}$ of M dimensions at any time t (for $u = 1, 2, \cdots, M$):

$$\mathbf{z}_u^{(t)} = \left(\exp\left(-\frac{1}{2}\lambda_1 t\right)E_{1u}, \exp\left(-\frac{1}{2}\lambda_2 t\right)E_{2u}, \cdots, \exp\left(-\frac{1}{2}\lambda_M t\right)E_{Mu}\right)^T \tag{43}$$

where t is the random walk diffusive time of heat flow on the graph. That represents a kernel mapping

$$\mathcal{F} : V \longrightarrow R^M, \tag{44}$$

which embeds each vertex of digraph in a vector space R^M. Thus, the Euclidean distance between any two vertices u and v is defined as

$$\ell(u, v) = \sqrt{(\mathbf{z}_u^{(t)} - \mathbf{z}_v^{(t)})^T(\mathbf{z}_u^{(t)} - \mathbf{z}_v^{(t)})} \tag{45}$$

2. Deep Graph Convolution Neural Network Model

First, we describe the graph representation of the dataset created by computation of the heat flow $q_{u\to v}^{\text{ext}}$ between two amino acids u and v, representing the vertices of a graph. This is done at regular time frame intervals of MD simulations; therefore, a set of such graph representations will be stored in the dataset with $t = 1, 2, \cdots, T$ entries where T is the number of snapshots.

Figure 11(a) illustrates a graph representation of the heat flow at some time frame t, $G_t(V, E, \mathbf{A})$, in a protein where vertices represent amino acids and the arrow lines indicate

the direction of heat flow; that is, if $q^{\text{ext}}_{u\to v} > 0$, an edge is added from u to v with weight $W_{uv} = q^{\text{ext}}_{u\to v}$. This constructs an adjacency matrix $\mathbf{A}$, as illustrated in Fig. 11(b). Each vertex u is associated with an embedding vector $\mathbf{z}_u$, constructing the embedding matrix of rank $M \times M$ (see Fig. 11(c)), which is computed from spectral decomposition of the Laplacian $\mathbf{L}$ matrix using Eq. 43 and $\mathbf{L}$ is computed using Eq. 41. Besides, there could be assigned an embedding matrix to the edge attributes; here, the probability transition matrix $\mathbf{P}$, given by Eq. 39, is used as the edge embedding matrix, hence $n = 1$.

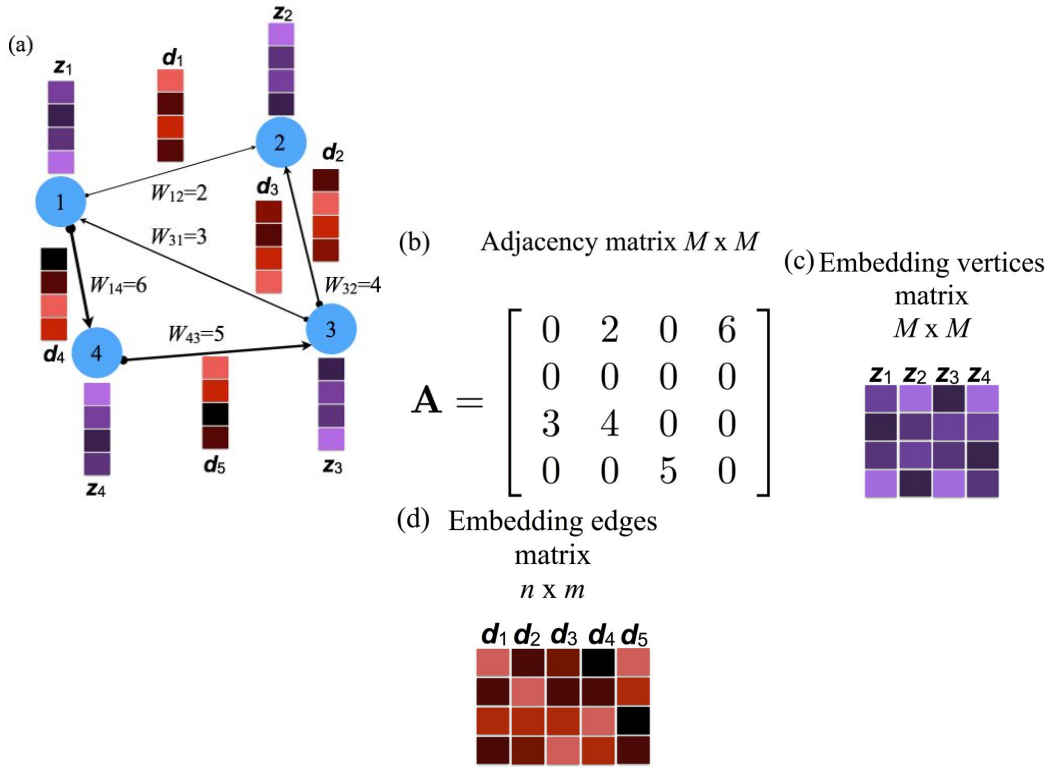

FIG. 11: a) Illustration of a graph representation indicating the vertices set, edges set, edges weights, edges embeddings, vertex embeddings; (b) Adjacency matrix $M \times M$; (c) Embedded vertices matrix $M \times M$; and (d) Embedded edges matrix $n \times m$.

A graph neural network (GNN) is a computation model that uses the graph representation, described above, as an input (that is, the vertex embedding vectors $\mathbf{z}_u$, $u = 1, 2, \cdots, M$, adjacency matrix $\mathbf{A}$, edge embedding matrix $\mathbf{P}$) and transfers it into a set of L layers. The vertex embedding vectors $\mathbf{z}_u$ at every layer update to construct the so-called intermediate *hidden* representations $\mathbf{h}_{u,l}$ with the output vertex embedding vector $\mathbf{h}_{u,L}$. Note that the GNN uses matrix representation; for instance, $\mathbf{Z}$ is the matrix of vertex embeddings with

the columns the vertex embeddings $\mathbf{z}_u$. Similarly, the layer embeddings $\mathbf{H}_l$ and the output matrix $\mathbf{Y}$. The input layer of the GNN contains the information about the embeddings of each vertex u in $\mathbf{Z}$ and the last layer (layer L) contains the information about the vertices and their context within the graph in $\mathbf{H}_L$.

We assign to each vertex u of the graph an output vector $\mathbf{y}_u$, which represents the set of measurements to be predicted using the so-called *regression* at a vertex-level task; here, we used the amino acids ranks (PageRank, authority, and hub indices):

$$\mathbf{y}_u^{(t)} = (\text{PageRank}(u), \text{Authority}(u), \text{Hub}(u))^T \tag{46}$$

That is done using the graph structure and vertex embeddings. Besides, the loss functions are independently defined at each vertex u:

$$\mathbf{y}_u^{(\text{pred})} = \sigma\left(\boldsymbol{\gamma}_L^{(1)} + \boldsymbol{\gamma}_L^{(2)}\mathbf{h}_{u,L}\right) \tag{47}$$

where σ is a sigmoid or softmax function and the column vector $\boldsymbol{\gamma}_L^{(1)}$ and a matrix $\boldsymbol{\gamma}_L^{(2)}$ are learned parameters. (Note that sometimes a final linear transformation is needed to map vertex embeddings to the expected output feature dimensionality). The loss function was the mean-square-error (mse) ones. One can also use the same network to predict the edge attribute embeddings between at any two vertices u and v, such as the transition probability matrix elements:

$$P^{(\text{pred})}(v|u) = \sigma\left((\mathbf{h}_{u,L})^T\mathbf{h}_{v,L}\right) \tag{48}$$

In this study, the graph convolutional network (GCN) is used among the other GNNs as computational model. In a GCN model, the vertices are updated by aggregation of the information from the neighbouring vertices by considering a relational inductive bias. Here, the spacial-based GCN model is implemented that applies convolution spacial domain based on the initial graph structure. A nonlinear activation function f (such as a sigmoid or softmax) applies at each layer for each vertex u with parameters $(\mathbf{b}_l, \boldsymbol{\Gamma}_l)$ (for $l = 0, \cdots, L$), where the inputs are the vertex embeddings and adjacency matrix. The activation function applies to every vertex embeddings as:

$$\begin{aligned} \mathbf{h}_{u,1} &= f\left(\mathbf{b}_0 + \boldsymbol{\Gamma}_0^T\left[\mathbf{z}_u + \mathbf{a}(u,1)\right]\right) \\ \mathbf{h}_{u,l} &= f\left(\mathbf{b}_{l-1} + \boldsymbol{\Gamma}_{l-1}^T\left[\mathbf{h}_{u,l-1} + \mathbf{a}(u,l-1)\right]\right), \quad l = 2, \cdots, L \end{aligned} \tag{49}$$

where $\mathbf{a}(\cdots)$ is the aggregation function:

$$\mathbf{a}(u,l) = \sum_{v \in N(u)} \mathbf{h}_{v,l} \tag{50}$$

where $N(u)$ is the list of the neighbouring vertices of u. The outputs are again predicted as:

$$\mathbf{y}_u^{(\mathrm{pred})} = \sigma\left(\boldsymbol{\gamma}_L^{(1)} + \boldsymbol{\gamma}_L^{(2)} \mathbf{h}_{u,L}\right) \tag{51}$$

In training, we considered, as a dataset, T graphs $G_t(V, E, \mathbf{A}, \mathbf{P})$ and the measurements $\mathbf{Y}^{(t)}$ $(t = 1, 2, \cdots T)$. The dataset is randomly split into training dataset (80 %) T_{train}, validation dataset (10 %) T_{valid}, and test dataset (10 %) T_{test}. The test dataset is such that it does not share any information with training or validation datasets. In this way, the set of training parameters consists of $\{\mathbf{b}_l, \boldsymbol{\Gamma}_l\}$ for $l = 0, 1, \cdots, L$. We used the so-called *Adam* algorithm as optimiser and the MSE loss. The mini batches are chosen as a random subset of vertices of the graph at each training iteration. Other aggregation function are also chosen for testing, such as the mean or maximum pooling aggregation functions:

$$\mathbf{a}(u,l) = \frac{1}{\mid N(u) \mid} \sum_{v \in N(u)} \mathbf{h}_{v,l} \tag{52}$$

$$\mathbf{a}(u,l) = \max_{v \in N(u)} \mathbf{h}_{v,l} \tag{53}$$

A jump knowledge connection algorithm [34] was used according to this, the model will additionally apply a final linear transformation to transform vertex embeddings to the expected output feature dimensionality. Here, the long short-term memory (lstm) algorithm [35] was used. Furthermore, a DeeperGCNLayer model [36] was implemented that includes the pre-activation residual connection, the residual connection, the dense connection, and no connections.

3. Test Case

First test system was a system of five vertices, with a local transfer entropy given as:

$$t_{i \to j}^{(0)} = \begin{bmatrix} 0.0 & 6.1 & 7.5 & 6.5 & 6.7 \\ -0.2 & 0.0 & -0.2 & -0.2 & 0.2 \\ 6.0 & 7.1 & 0.0 & 4.5 & 7.0 \\ -0.1 & -0.2 & -0.2 & 0.0 & 0.2 \\ 2.2 & 0.8 & 0.2 & 0.2 & 0.0 \end{bmatrix} \tag{54}$$

At every time iteration, a white noise $\mathcal{N}(0,1)$ is added to every element as:

$$t_{i\to j}^{(t)} = t_{i\to j}^{(0)} + 0.01 \cdot \mathcal{N}(0,1) \tag{55}$$

It is easily to judge that the vertex 1 has the highest rank and then vertex 3 the second highest rank.

Figure 12(a) shows the loss as a function of epochs for the training dataset and Fig. 12(b) shows the performance metrics for the training, validation, and test datasets. Besides, in Fig. 12(c-e), we present the predicted and the true (measured) PageRank values based on the train dataset, the validation dataset, and the test dataset. The size of dataset was $T = 2000$ frames and length of training, validation, and test datasets was 80 %, 10 %, and 10 %, respectively. The batch sizes were $(100, 1, 1)$ and number of epochs 250. Here, the number of layers was 6 and number of channels at each layer was 32. The *adam* optimiser was chosen, the activation function for the hidden layers was *relu*, and the sigmoid function for the output layer. The aggregation function was *maximum* and the jumping knowledge mode was *lstm*. The learning rate parameter was 0.001.

4. BPTI[14-38]Abu Protein

Figure 13(a) shows the loss as a function of epochs for the training dataset and Fig. 13(b) shows the performance metrics for the training, validation, and test datasets. Besides, in Fig. 13(c-e), we present the predicted and the true (measured) PageRank values based on the train dataset, the validation dataset, and the test dataset. The size of dataset was $T = 1000$ frames and length of training, validation, and test datasets was 80 %, 10 %, and 10 %, respectively. The batch sizes were $(100, 1, 1)$ and number of epochs 1000. Here, the number of layers was 10 and number of channels at each layer was 16. The *adam* optimiser was chosen, the activation function for the hidden layers was *relu*, and the sigmoid function for the output layer. The aggregation function was *maximum* and the jumping knowledge mode was *lstm*. The learning rate parameter was 0.001.

V. CONCLUSIONS

This study discussed some alternative definitions for the information theoretic-measures of transfer entropy between time series using (α, q) measures to validate statistical efficiency.

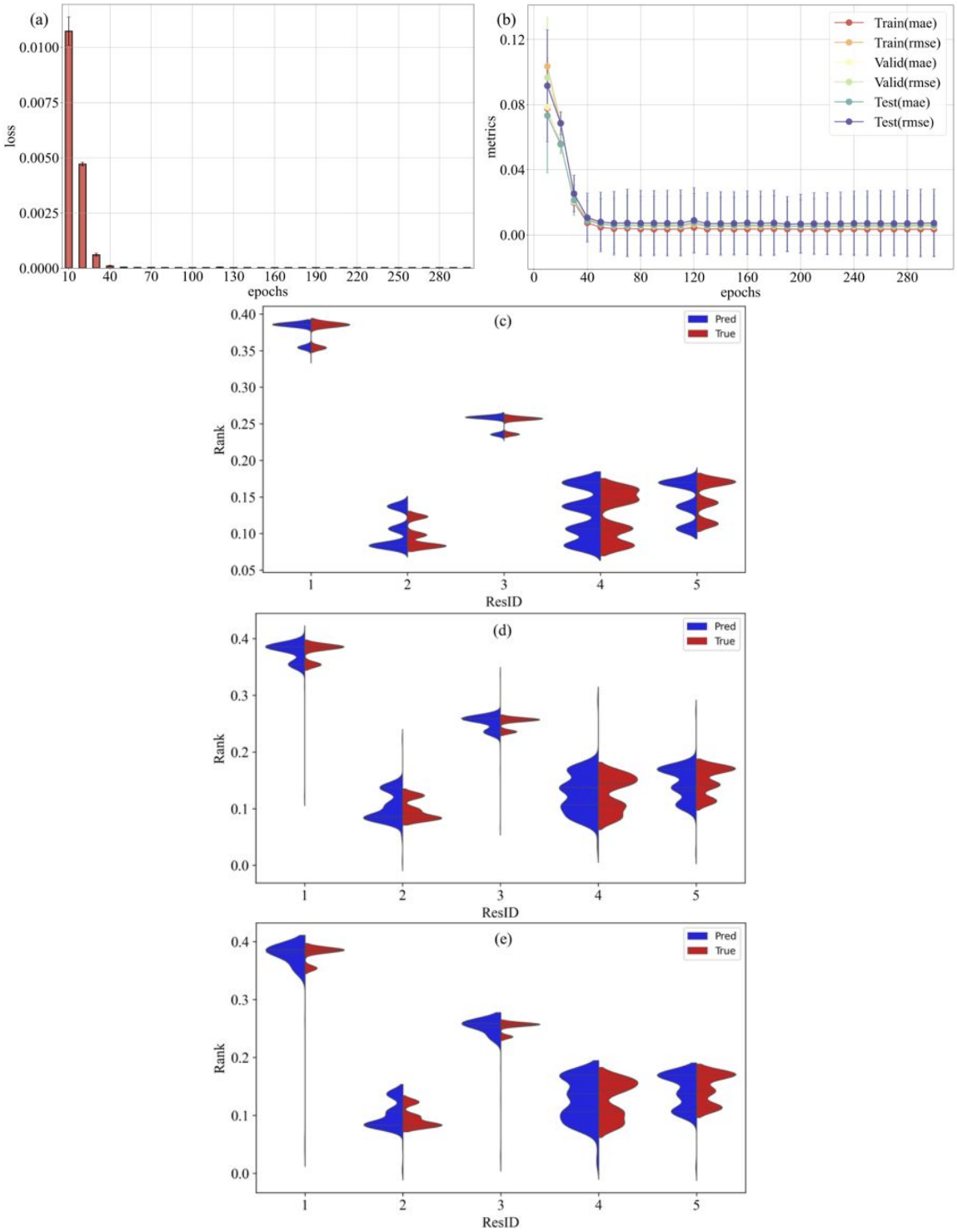

FIG. 12: a) Loss as a function of epochs for the training dataset; (b) Performance metrics for the training, validation, and test datasets; (c) Predicted and the true (measured) PageRank values based on the train dataset; (d) Predicted and the true (measured) PageRank values based on the validation dataset; (e) Predicted and the true (measured) PageRank values based on the test dataset. The length of training, validation, and test datasets was 80 %, 10 %, and 10 %, respectively, and the size of dataset was 1000 frames. The batch sizes were $(100, 1, 1)$ and number of epochs 1000.

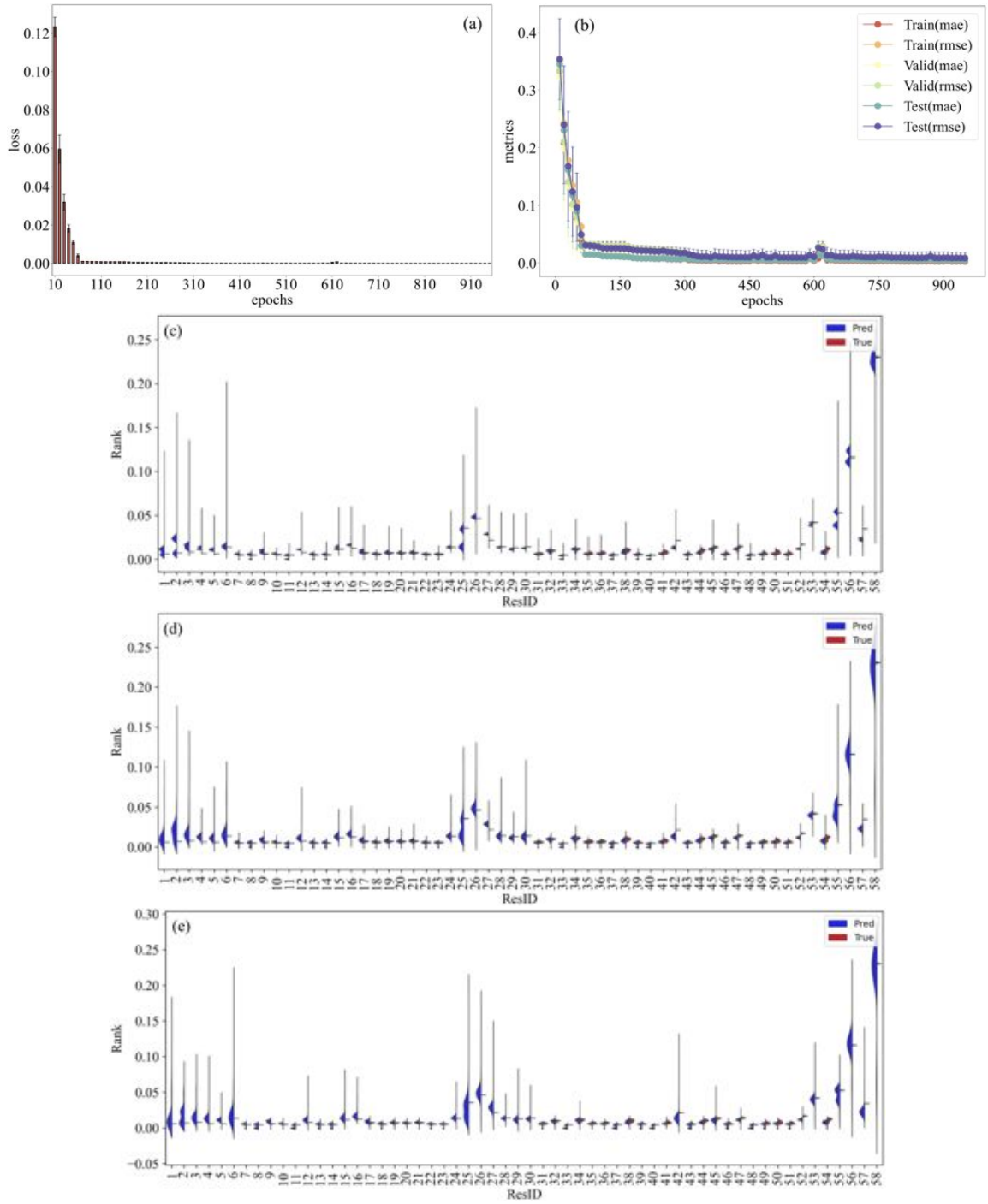

FIG. 13: a) Loss as a function of epochs for the training dataset; (b) Performance metrics for the training, validation, and test datasets; (c) Predicted and the true (measured) PageRank values based on the train dataset; (d) Predicted and the true (measured) PageRank values based on the validation dataset; (e) Predicted and the true (measured) PageRank values based on the test dataset. The length of training, validation, and test datasets was 80 %, 10 %, and 10 %, respectively, and the size of dataset was 1000 frames. The batch sizes were $(100, 1, 1)$ and number of epochs 1000.

The primary goal was at finding new measures of high-order correlations that preserved information-theoretic properties of information transfer due to local operations.

Besides, this study revealed the decoded information contained in an amino acid sequence establishing an 3D protein structure by comparing the physical-chemical properties with the ranked role in the protein interaction network topology using new graph-theoretic measures based on the constructed digraph models of (α, q) information transfer.

Moreover, in this study, the DGCNN approach was used for classifying the amino acid role in a protein structure trained upon short equilibrium structure fluctuations at sub-nanoseconds time scales.

We examined the influence of disulphide bridges to the 3D protein structure of the Bovine Pancreatic Trypsin Inhibitor protein (wild type and its mutated analogue).

---

[1] M. Karplus and J.A. McCammon. *Nat. Struct. Biol.* 9(9), 646–652 (2002). With corrigenda in Nat. Struct. Biol. 9(10),788 (2002).

[2] W.F. van Gunsteren, D. Bakowies, R. Baron, I. Chandrasekhar, M. Christen, X. Daura, P. Gee, D.P. Geerke, A. Glättli, P.H. Hünenberger, M.A. Kastenholz, C. Oostenbrink, M. Schenk, D. Trzesniak, N.F.A. van der Vegt, and H.B. Yu. *Angew. Chem. Int. Ed.* 45(25), 4064–4092 (2006).

[3] H. Kamberaj. *J. Mol. Graph. Model.* 104, 107838 (2021).

[4] T. Schreiber. *Phys. Rev. Lett.* 85, 461–464 (2000).

[5] H. Kamberaj and A. der Vaart. *Biophys. J.* 97, 1747–1755 (2009).

[6] V. Ilić and I. B. Djordjević. *Entropy* 23, 702–725 (2021).

[7] S. O. Scalet, A. M. Alhambra, G. Styliaris, and J. I. Cirac. *Quantum* 5, 541 (2021).

[8] Dh. Nebiu and H. Kamberaj. *SoftwareX* 11, 100470 (2020).

[9] J. Granger. *Acta Physica Polonica B* 37, 424–438 (1969).

[10] A. Rényi. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, 547–561. University of California Press, (1961).

[11] O. Penrose. *Foundations of Statistical Mechanics: A Deductive Treatment.* Dover Publications, Mineola, NY, USA, (1970).

[12] C. Tsallis. *J. Stat. Phys.* 52, 479–487 (1988).

[13] C. Tsallis. *Entropy* 13, 1765–1804 (2011).

[14] B. Gourévitch and J. Eggermont. *J. Neurophysiol.* 97, 2533–2543 (2007).

[15] H. Kamberaj. *J. Mol. Graph. Model.* 114, 108199 (2022).

[16] J. T. Lizier, M. Prokopenko, and A. Y. Zomaya. *Phys. Rev. E* 77, 026110 (2008).

[17] T. D. Schneider. *J. Theor. Biol.* 148, 125–137 (1991).

[18] M. Prokopenko, J. T. Lizier, and D. C. Price. *Entropy* 15, 524–543 (2013).

[19] M. Prokopenko and J. T. Lizier. *Sci. Rep.* 4, 5394 (2014).

[20] H. Kamberaj. *Molecular Dynamics Simulations in Statistical Physics. Theory and Applications.* Scientific Computation Series. Springer Nature, (2020).

[21] A. Wlodawer, J. Walter, R. Huber, and L. Sjolin. *J. Mol. Biol.* 180, 301–329 (1984).

[22] I. V. Getun, C. K. Brown, J. Tulla-Puche, D. Ohlendorf, C. Woodward, and G. Barany. *J. Mol. Biol.* 375, 812 (2008).

[23] B. R. Brooks, C. L. Brooks, A. D. MacKerell, L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, and M. Karplus. *J. Comput. Chem.* 30(10), 1545–1614 (2009).

[24] J. A. D. MacKerell, D. Bashford, M. Bellott, Jr. R. L. Dunbrack, J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. Reiher, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, and M. Karplus. *J. Phys. Chem. B* 102, 3586–3616 (1998).

[25] A. D. Jr. MacKerell, M. Feig, and C. L. Brooks III. *J. Am. Chem. Soc.* 126, 698–699 (2004).

[26] B. Robert, R. B. Best, X. Zhu, J. Shim, P. Lopes, J. Mittal, M. Feig, and A. D. Jr. MacKerell. *J. Chem. Theory Comput.* 8, 3257–3273 (2013).

[27] H. Kamberaj. *Computer Simulations in Molecular Biology.* Scientific Computation Series. Springer Nature, (2023).

[28] J. P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen. *J. Comput. Phys.* 23, 327–341 (1977).

[29] M. Karplus and J. Jushick. *Macromolecules* 14, 325–332 (1981).

[30] I. Andricioaei and M. Karplus. *J. Chem. Phys.* 115, 6289–6292 (2001).

[31] J. Chen and C. L. Brooks III. *Phys. Chem. Chem. Phys.* 10, 471–481 (2008).

[32] J. Chen and C. L. Brooks III. *Phys. Chem. Chem. Phys.* 10, 471–481 (2008).

[33] J. Y. Yen. *Management Science* 17(11), 712–716 (1971). Theory Series.

[34] K. Xu, C. Li, Y. Tian, T. Sonobe, K. -i. Kawarabayashi, and S. Jegelka. *International Conference on Machine Learning* 267, 5453–5462 (2018).

[35] S. Hochreiter and J. Schmidhuber. *Neural computation* 9(8), 1735–1780 (1997).

[36] G. Li, M. Müller, G. Qian, I. C. Delgadillo, A. Abualshour, A. Thabet, and B. Ghanem. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45(6), 6923–6930 (2023).