This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2756879, IEEE/ACM Transactions on Computational Biology and Bioinformatics

Discovery of Functional Motifs from the Interface Region of Oligomeric Proteins using Frequent Subgraph Mining

Tanay Kumar Saha, Ataur Katebi, Wajdi Dhifli and Mohammad Al Hasan

***Abstract*—Modeling the interface region of a protein complex paves the way for understanding its dynamics and functionalities. Existing works model the interface region of a complex by using different approaches, such as, the residue composition at the interface region, the geometry of the interface residues, or the structural alignment of interface regions. These approaches are useful for ranking a set of docked conformation or for building scoring function for protein-protein docking, but they do not provide a generic and scalable technique for the extraction of interface patterns leading to functional motif discovery. In this work, we model the interface region of a protein complex by graphs and extract interface patterns of the given complex in the form of frequent subgraphs. To achieve this we develop a scalable algorithm for frequent subgraph mining. We show that a systematic review of the mined subgraphs provides an effective method for the discovery of functional motifs that exist along the interface region of a given protein complex.**

In our experiments, we use three PDB protein structure datasets. The first two datasets are composed of PDB structures from different conformations of two dimeric protein complexes: HIV-1 protease (329 structures), and triosephosphate isomerase (TIM) (86 structures). The third dataset is a collection of different enzyme protein structures from the six top-level enzyme classes, namely: Oxydoreductase, Transferase, Hydrolase, Lyase, Isomerase and Ligase. We show that for the first two datasets, our method captures the locking mechanism at the dimeric interface by taking into account the spatial positioning of the interfacial residues through graphs. Indeed, our frequent subgraph mining based approach discovers the patterns representing the dimerization lock which is formed at the base of the structure in 323 of the 329 HIV-1 protease structures. Similarly, for 86 TIM structures, our approach discovers the dimerization lock formation in 50 structures. For the enzyme structures, we show that we are able to capture functional motifs (active sites) that are specific for each of the six top-level classes of enzymes through frequent subgraphs.

***Index Terms*—Bio-Informatics, Functional Motifs, Interfacial Network, Frequent Subgraph Mining**

I. Introduction

STRUCTURAL dynamics and functions of many proteins are primarily controlled by the interaction of residues at the interface region. Because of this, studying and analyzing the interface region of a protein is crucial for understanding the underlying protein machinery [1]. In existing literatures, many research works have provided a detailed analysis of the interface region of various proteins. However, in the majority of these works protein interface region is represented through different spatial features; examples include interface area, interface polar residue abundance, hydrogen bonds, solvation free energy gain from interface formation, and binding energy [2]. Such a feature-based representation—although useful for ranking of predicted docked conformation of protein-protein complexes or for building scoring function for docking [3]–[5]—is not much useful for understanding protein machinery. This is due to the fact that a feature-based representation of interface region works like a black-box without providing much information regarding the functionalities of the protein. So, alternative representations of interface regions are needed for providing a better understanding of functional motifs, which are responsible for carrying out protein's intended functionalities.

Sequence motifs often correspond to the functional regions of a protein, such as, catalytic sites, binding sites, structural motifs, etc. and they are considered to be the building blocks of protein sequences [6]–[8]. These motifs are conserved across different proteins and possess highly discriminative features for predicting the functions of a protein [9]. However, sequence motifs are limited in their representation ability, so in recent years, networks are being used for representing biological data. Besides, network theories are also being used to gain insights into complex biological problems [10]–[13]. The concept of *network motif* has also emerged, which has been hypothesized to play an important role in carrying out the key functionalities that are performed by the entities in a biological network [14]–[17]. A very recent study [18] showed that the distribution of network motifs influences the organization of metabolic networks. However, the methodologies for network motif discovery [14], [17] yield sub-networks that are frequent in a given network, and hence they are not useful for finding conserved sub-networks at the interface of a set of proteins.

Mining frequent sub-networks (FSM) is an important and well studied task in data mining field; it is defined as finding all subgraphs that appear frequently in a graph dataset given a minimum frequency threshold. There are two variants of

Tanay Kumar Saha and Mohammad Al Hasan are with the computer and Information Science Department, Indiana University Purdue University, Indianapolis, Indiana, USA-46202. E-mail: {tksaha,alhasan}@iupui.edu

Ataur Katebi is a postdoctoral fellow in the Department of Veterinary Microbiology and Preventive Medicine at Iowa State University. E-mail: arkatebi@gmail.com

Wajdi Dhifli is with the institute of Systems and Synthetic Biology (iSSB), University of Evry Val d'Essonne, 91030, Evry, France. E-mail: wajdi.dhifli@univ-evry.fr

A short version of the paper was accepted in: 15th International Workshop on Data Mining in Bioinformatics (BIOKDD'16). The workshop paper can be downloaded from:http://cs.iupui.edu/∼tksaha/funcmotif.pdf

This is the author's manuscript of the article published in final edited form as:
Saha, T. K., Katebi, A., Dhifli, W., & Hasan, M. A. (2018). Discovery of Functional Motifs from the Interface Region of Oligomeric Proteins using Frequent Subgraph Mining. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 1–1. https://doi.org/10.1109/TCBB.2017.2756879

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2756879, IEEE/ACM Transactions on Computational Biology and Bioinformatics

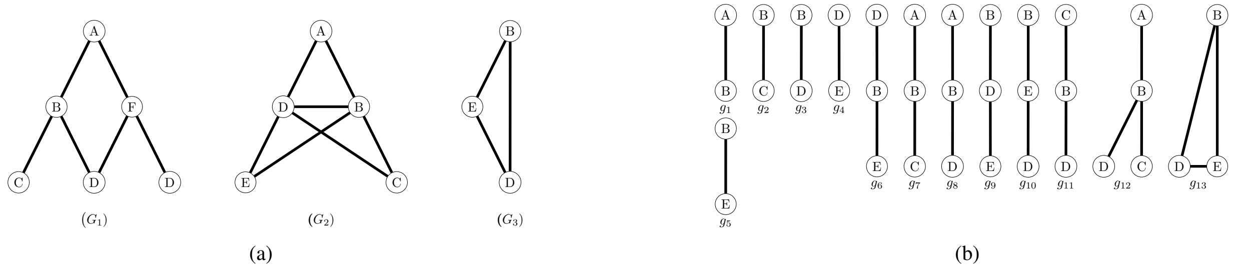


Fig. 1. (a) A graph database with 3 graphs (b) All the frequent subgraphs of the graph database in (a) using a minimum support value of 2. If we want to obtain only the induced frequent subgraphs, g_1-g_5, g_7, and g_{13} are frequent for a minimum support of 2.

this problem—in the first variant [19]–[25], the dataset has a collection of many graphs, and in the second variant [26]–[28], the dataset contains a single large graph. For the latter variant, the frequency of a subgraph is counted as its multiplicity in the large graph. On the other hand, the earlier variant of graph mining counts the frequency of a subgraph over the collection of graphs in the dataset. Thus, for this variant of graph mining, the overall frequency of a subgraph pattern is the number of distinct graphs in which the pattern appears. In this work, we represent the interface region of oligomeric proteins as a set of networks and then use a novel frequent sub-network mining algorithm for finding functional motifs in the interface region. As we discover patterns that span over a set of networks, the algorithms belonging to the first variant are relevant for our task and forthcoming references of frequent graph mining in this paper pertain to the first variant of FSM.

Mining sub-networks from a set of networks is defined as follows: Given a graph dataset $\mathcal{G}$, and a minimum support $\pi^{\textbf{min}}$, obtain the set of subgraphs whose frequency is higher than $\pi^{\textbf{min}}$. The set of frequent subgraphs are generally represented by $\mathcal{F}$. In Figure 1(a), we show a graph dataset with 3 graphs and in Figure 1(b) we show the frequent subgraphs of this dataset considering $\pi^{\textbf{min}} = 2$. Over the years, a good number of algorithms for frequent sub-network mining (FSM) have been proposed, examples include Subdue [19], AGM [20], FSG [21], gSpan [22], FFSM [29], DMTL [24], and Gaston [25]. Distributed solutions of FSM [28], [30] which runs on map-reduce platform have also been proposed.

Existing FSM algorithms are proven to be effective for finding frequent subgraphs from input graphs which are small and sparse. However, for general graphs, FSM task is not scalable due to the inherent complexity of this task. In fact, Horváth *et al.* have shown that FSM cannot be solved in output polynomial time [31]. The lack of scalability of FSM task has also been shown empirically. For instance, FSM has been applied on a small dataset (only 3 graphs) of protein-protein interaction (PPI) graphs, each graph having 2154 nodes on average; but the most efficient of the existing FSM algorithms cannot mine all the frequent subgraphs from this dataset in days of running even with 100% support value [32]. Distributed solution, such as [30] can successfully overcomes the lack of scalability issues arising from the large number of graphs in the dataset, but they still remains not scalable when the graphs in the dataset are dense and large. Our investigation finds that any reasonable construction of interface networks on real-life protein data yields large and dense graphs for which existing methods simply fail to find interface patterns in an effective manner.

Existing FSM methods suffer from some other serious limitations when they are used for mining interface patterns. First, existing subgraph mining methods require that the user selects a minimum support threshold value [22], [24], [25]. However, when the main objective of subgraph mining is to discover functional motifs from a number of protein conformations, this support value is generally unknown. This is due to the fact that the spatial orientation of the residues in a functional motif across the conformations fluctuates owing to the dynamics of the motif, and a part of the motif may be occluded in some subgraphs, making the motif infrequent. So, choosing a large support threshold may miss a significant part of a functional motifs; on the other hand, choosing a small support threshold may return too many random subgraphs that are frequent simply by chance. The second limitation is that existing algorithms [22], [24], [25] enumerate all the frequent subgraphs starting from size-1 and thus they return a large number of unnecessary patterns. But, for functional motifs, the subgraph size of interest is known in many cases; if not known, a reasonable initial guess of the motif size can be made from the knowledge of protein's family and functionalities. So, a novel frequent subgraph mining method is needed which is scalable, not dependent on the minimum support threshold, and able to return frequent subgraphs of a user-specified size.

In this work, we propose a graph mining framework which is particularly suited for the discovery of functional motifs from the interface graphs of a large collection of protein structures. Our proposed approach uses spatial proximity for creating the interfacial network dataset, so, the proteins in a dataset need to have high structural similarity (low structural diversity). For instance, these structures could either be structural conformations of the same protein (see Sections V-A and V-B) or they could represent multiple proteins from the same functional group (see Sections V-C). The proposed method first creates a dataset of interface graphs, each representing a structure from the database. It then uses a novel sampling based method for mining subgraphs of a given size which are frequent over the graph database with a high probability. In the proposed method, subgraph size is user-defined, which can be chosen from user's domain knowledge of the protein under

1545-5963 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2756879, IEEE/ACM Transactions on Computational Biology and Bioinformatics

investigation.

To validate the effectiveness of our method we perform three independent experiments. In the first two experiments, we use two different datasets of protein conformations: (1) HIV-1 protease (329 conformations) and (2) Triosephosphate isomerase (TIM) (86 conformations) and find frequent subgraphs of appropriate size from the given conformations of these proteins. The subgraphs that we mine from the interface networks enable us to discover the functional motifs in the above pair of proteins. The first protein, HIV-1 protease is essential for the life cycle of human immunodeficiency virus (HIV) which causes acquired immunodeficiency syndrome (AIDS) in humans. The second protein, TIM is the fifth enzyme in the glycolysis pathway that produces energy in all living organisms. For both proteins, the large number of structures represent a sample of different conformational states of the proteins that are solved experimentally and they can explain the functional dynamics and functional motifs of the protein [33]. The 10 most frequent subgraphs mined from the HIV-1 protease using our proposed method collectively capture a 16-residue functional motif, named dimerization lock (shown in Figure: 2(a)) that exists in the interface of the protein. Among these frequent subgraphs, our method retrieves 15 out of 16 residues in 6 subgraphs, 14 residues in 2 subgraphs and 13 residues in the remaining 2 subgraphs. Similarly, frequent subgraphs from TIM retrieve dimerization lock that exists in TIM conformations (shown in Figure: 2(b)).

In the third experiment, we use the Dobson and Doig (D&D) benchmark dataset for enzymes (691 enzymes out of 1178 protein structures) [34]. As enzymes are known to be macromolecular catalysts, discovering functional motifs at the interface region of these proteins is paramount to understanding how they bind and interact with other macromolecules to perform their functions. The subset of enzymes in D&D is composed of groups of proteins from the six top-level classes of enzymes namely: Oxydoreductase, Transferase, Hydrolase, Lyase, Isomerase and Ligase. We use our approach for mining function specific motifs for each of these classes of enzymes. Specifically, for each class, We mine up to 200 most frequent patterns within a size range of 5, 6, 7 and 8 nodes per pattern. By checking the overlap between the set of patterns mined from each class, we show that our approach discovers function specific patterns from each functional class of enzymes. We also show that these patterns include catalytic sites of enzymes that have been identified in the literature.

We claim the following contribution in this paper:

- We propose a method to map the interfacial region of a protein as a network for the discovery of functional motifs by using a sampling based frequent subgraph (FSM) mining method.
- We validate the utility of the proposed FSM method by capturing the locking mechanism at the dimeric interface from different conformations of HIV and TIM protein structures.
- We also observe that our sampling based FSM method enables us to capture function specific patterns at the interface region of 3D structures of proteins belonging to the same functional group.

II. Background

Let $G(V, E)$ be an *interfacial network*, where V is the set of vertices and E is the set of edges. For our problem, the vertices are set of residues and the edges are connection among the residues based on their pair-wise physical proximity. Specifically, if the inter- and intra-chain distance between a pair of residue is smaller than a user defined distance threshold, an edge is added between the corresponding pair of vertices. By construction, the interfacial networks are simple graph which do not have self-loops or multi-edges. Besides, these graphs are undirected, because the Euclidean distance is a symmetric metric. Finally, for all reasonable choices of inter and intra chain distance threshold, these graphs are connected.

A *labeled graph* $G(V, E, L, \Psi)$ is a graph[1] for which the vertices and the edges have labels that are assigned by a labeling function, $\Psi : V \cup E \rightarrow L$ where L is a set of labels. In our case, only vertices have labels, which is a value between 1 to 20, corresponding to the 20 amino acid residues of proteins.

A graph $G' = (V', E')$ is a *s*ubgraph of G (denoted as $G' \subseteq G$) if $V' \subseteq V$ and $E' \subseteq E$. A graph $G' = (V', E')$ is a *v*ertex-induced subgraph of G if G' is a subgraph of G, and for any pair of vertices $v_a, v_b \in V'$, $(v_a, v_b) \in E'$ if and only if $(v_a, v_b) \in E$. In other words, a *vertex-induced* subgraph of G is a graph G' consisting of a subset of G's vertices together with all the edges of G whose both endpoints are in this subset. In this paper, we have used the word *subgraph* for abbreviating vertex-induced subgraph. If G' is a (induced or non-induced) subgraph of G and $|V'| = \ell$, we call G' a ℓ-subgraph of G.

Let $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$ be an interfacial network database, where each $G_i \in \mathcal{G}, \forall i = \{1 \ldots n\}$ represents a labeled, undirected and connected graph. The *support-set* of the graph g is $\mathbf{t}(g)$, and $\mathbf{t}(g) = \{G_i : g \subseteq G_i \in \mathcal{G}\}, \forall i = \{1 \ldots n\}$. This set contains all the graphs in $\mathcal{G}$ that have a subgraph isomorphic to g. The cardinality of the *support-set* is called the *support* of g. g is called frequent if $support \geq \pi^{\textbf{min}}$, where $\pi^{\textbf{min}}$ is predefined/user-specified *minimum support (minsup)* threshold. Given the graph database $\mathcal{G}$, and minimum support $\pi^{\textbf{min}}$, the task of a frequent subgraph mining (FSM) algorithm is to obtain the set of frequent subgraphs (represented by $\mathcal{F}$). While computing support, if an FSM algorithm enforces induced subgraph isomorphism, it obtains the set of frequent induced subgraphs (represented by $\mathcal{F}_I$). It is easy to argue that $\mathcal{F}_I \subseteq \mathcal{F}$.

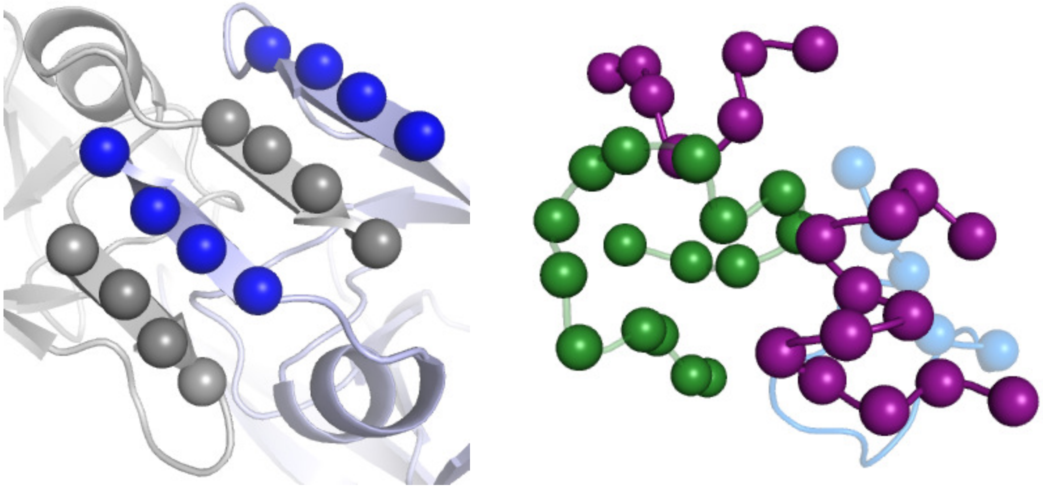
III. Related Work

There are several works that represent a protein structure as a network consisting of a set of nodes and the relationship between the nodes. However, the way different works model the network differs. Across these works, the nodes can represent amino acid residues [1], [12], [35]–[38], functional atoms from the side chains [39], [40], secondary structure elements [41]–[43], proteins [44], [45], protein complexes [46], and interaction pseudoatoms [47]. Edges also has different connotations

[1] We have used the terms graph and network interchangeably.

1545-5963 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2756879, IEEE/ACM Transactions on Computational Biology and Bioinformatics



(a) HIV

(b) TIM

Fig. 2. (a) Retrieved frequent patterns representing the dimerization lock at the base of HIV- 1 protease structure and (b) along the dimeric interface of triosephosphate isomerase.

in different works. For instances, edges connect nodes if they interact with each other [35], [36], or if they are nearer to each other spatially [1], [40], or if they are within the interacting distance of each other [39]. Some works create edges between two nodes if the nodes are part of a functional unit in a pathway or in a biological process [44], [45], or if side-chains interact with each other [13]. Our work differs in the method of construction and analysis of these networks from previous studies. In our work, we use C_α carbon (backbone carbon) of a particular residue as a node. So, the C_α carbons from all the residues of a particular protein represent the set of nodes and we connect two nodes if their C_α carbons are spatially nearer to each other. Existing works use a graph to capture the entire protein structure, but in this work we capture dense interfacial region between different subunits of the same structure.

In existing works, network representation of proteins has been used for various purposes; for example, to study the evolution of protein-protein interactions [1], to summarize how central network elements are enriched in active centers and ligand binding sites directing the dynamics of entire protein [40], to classify protein 3D-structures [12], [38], to characterize the topological role of residues [37], to offer a comprehensible view of critical residues and to facilitate the inspection of their organization [48], to detect cancer-associated functional residues [44], to uncover distinct cancer-specific functional modules [45], to document functional components and sub-components of proteins [49], and to compare two networks (Oligomeric vs Monomeric) [36] for getting insight into the protein association. Greene *et al.* [11] authored a good review article which surveys several key advances in the expanding area of protein structure and folding research using network approaches. To the best of our knowledge we are the first to develop graph mining methodologies for mining interfacial networks to discover important functional units (such as, lock structure in HIV 2(a) and hugging point 2(b) in TIM structure), or to find family specific active sites from enzymes.

IV. Methods

In Figure 3, we provide a pictorial depiction of the proposed method. Given a set of structures of a protein, we first convert each structure into an interfacial network, which is our collection of graphs in the graph dataset. Then we use our designed frequent pattern mining method for mining a set of fixed-size (user defined) subgraphs, which are the most frequent (probabilistically) over the graphs in the graph database. For each of the mined frequent subgraphs, we find their structural embedding in the host graphs, and identify those patterns for which the nodes in a pattern consistently map to a fixed set of residues in all the conformations. We consider these structural patterns as possible candidates of being a functional motif, and study whether these residues correspond to any known oligomerization mechanism. In this work, we use these set of steps to study the dimerization interfaces of HIV-1 and TIM proteins, and also to discover family-specific active sites of various enzyme families. Below, we discuss each of the steps of our method in details.

A. Modeling Protein as Interfacial Network

For each structure, we first retrieve the C_α carbons along with their 3D co-ordinates from the residues of a pair of chains U_i and U_j. We then construct an interfacial network of the structure by connecting the subset of C_α residues that are in the interface region of either of the chains. We consider a residue (say, v_a) in a chain (U_i) to be at the interface region if it is within a maximum spatial distance (γ) of any C_α residue (say, v_b) in the other chain (U_j), with respect to a distance measure (Δ) that is the Euclidean distance in our case. The interface C_α carbons are the set of nodes in our interface network. Within each chain, we connect pairs of residues if they are within a spatial proximity of at most δ. We label the nodes from 1 to 20 based on the amino acid types of the

1545-5963 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2756879, IEEE/ACM Transactions on Computational Biology and Bioinformatics

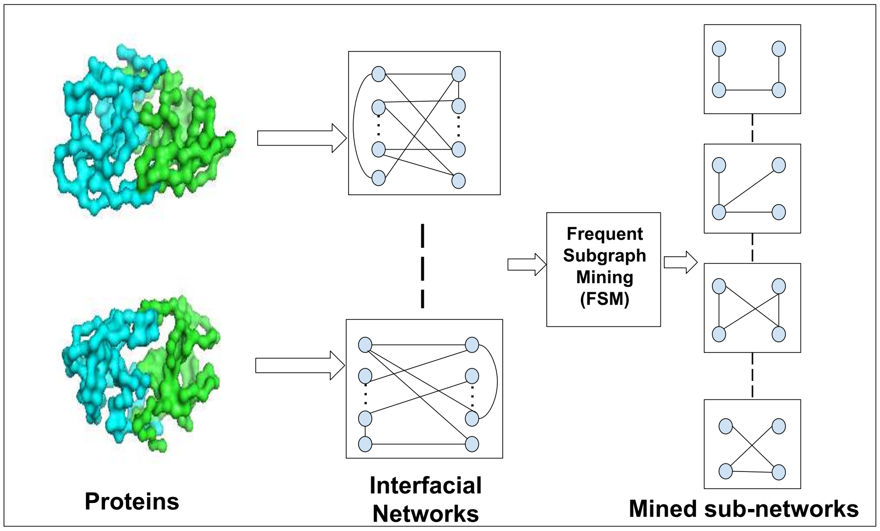


Fig. 3. A pictorial depiction of the proposed method. Given a set of structures of a protein, we first convert each structure into an interfacial network. Then we use a frequent pattern mining method for mining a set of fixed-size (user defined) subgraphs. Finally, for each of the mined frequent subgraphs, we find their structural embedding in the host graphs.

corresponding residues. Then, we form edges between nodes (residues) of different chains if they are spatially close to each other. After this step, we obtain an undirected vertex-labeled graphs— corresponding to interfacial network of the input protein structure. Equation (1) formally describes the graph modeling process. Note that for interfacial networks, the intra-chain distance threshold (δ) should be made low while the inter-chain distance threshold (γ) should be kept high. This will make the graph model emphasize the interfacial region at the surface between the different chains of the structure (at the 3D level) while making the intra-chain network very sparse to approximately contain at most the connections between amino acids at the primary structure level.

$$e(v_a, v_b) = \begin{cases} 1, \ if \ \Delta(v_a, v_b) \leq \delta \mid v_a \in U_i, v_b \in U_j, i = j \\ 1, \ if \ \Delta(v_a, v_b) \leq \gamma \mid v_a \in U_i, v_b \in U_j, i \neq j \\ 0, \ otherwise \end{cases} \tag{1}$$

It is important to note that having a larger distance threshold (values of δ, and γ) makes the interfacial networks denser and thus makes it more likely to find frequent subgraph patterns across different structures. However, the patterns that are discovered using a large threshold are less precise because the edges of these patterns cover a larger range of distances between a pair of residues. On the other hand, if we consider smaller distance threshold we get more precise patterns, but the mining process is less likely to find a frequent pattern. This is similar to precision-recall trade-off in information retrieval. For larger values of δ and γ, the recall increases but precision deteriorates, and for smaller values, the precision improves with a loss of recall.

B. Frequent Subgraph Mining with FS^3

For mining a fixed size frequent subgraph we use a sampling based graph mining algorithm, called FS^3, which we have proposed in one of our recent works [50]. FS^3 is based on sampling of subgraphs of a fixed size [2]. Given a graph dataset $\mathcal{G}$, and a size value ℓ, FS^3 samples subgraphs of size-ℓ from $\mathcal{G}$. The distribution from which the size-ℓ subgraphs is sampled is biased such that the sampling process over-samples the graphs that are likely to be frequent over the graphs in $\mathcal{G}$. FS^3 runs the above sampling process for many times, and uses an innovative priority queue to hold a small set of most frequent subgraphs, which it returns at the end of the sampling process. The unique feature of FS^3 is that unlike earlier works which are based on sampling [51], FS^3 does not perform any subgraph isomorphism (SI) test, so it is scalable to large graphs. By choosing different values of ℓ, user can find a succinct set of frequent subgraphs of different sizes. Also, as the number of samples increases, FS^3's output progressively converges to the top-k most frequent subgraphs of size ℓ. So user can run the sampler as long as he wants to obtain more precise results.

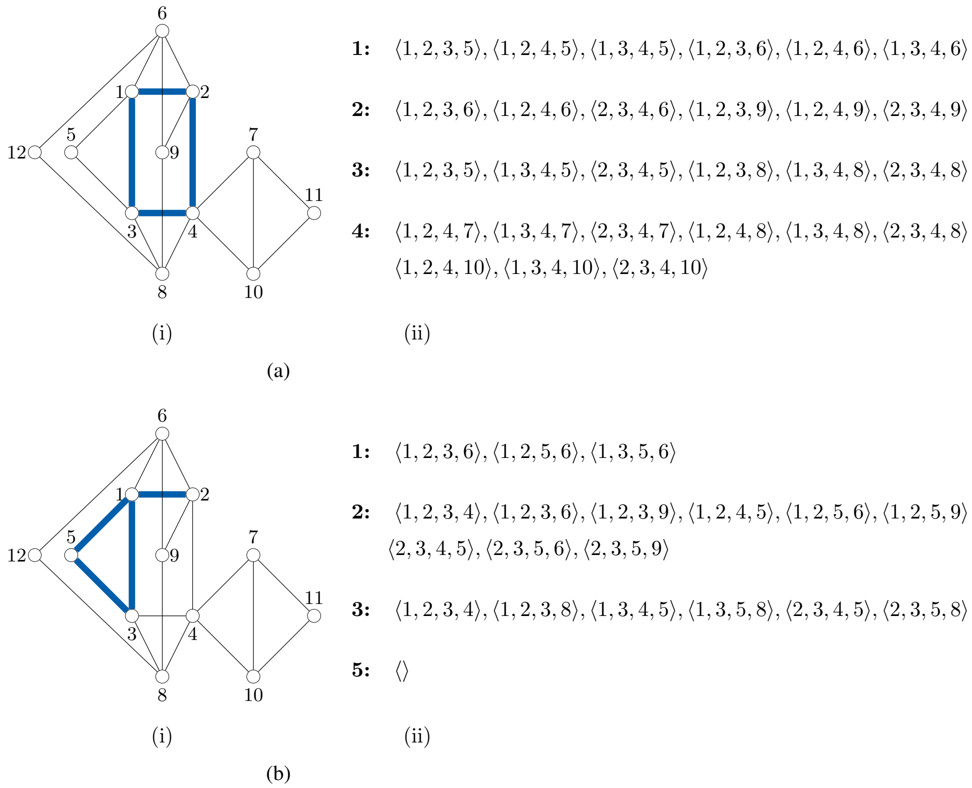
A detail discussion of FS^3 algorithm is out of scope for this paper. However, to make this paper self-sufficient, We describe below some key concepts of FS^3 algorithm. Interested readers are encouraged to read the original FS^3 paper [50] for more details.

Subgraph sampling by FS^3 Algorithm: At each sampling iteration, FS^3 performs a 2-stage sampling process. In the first stage, FS^3 chooses one of the graphs in $\mathcal{G}$ (say, G_i) uniformly, and in the second stage it samples a size-ℓ subgraph of G_i and returns. For the second stage, FS^3 performs a Markov chain Monte Carlo (MCMC) sampling over the ℓ-subgraphs of G_i. The main idea of MCMC sampling is to perform a random walk over the sampling space and subsequently return the sample the walk visits. The transitional probability of the

[2] The name FS^3 should be read as *F-S-Cube*, which is a compressed representation of the 4-gram composed of the bold letters in **F**ixed **S**ize **S**ubgraph **S**ampler.

1545-5963 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2756879, IEEE/ACM Transactions on Computational Biology and Bioinformatics



1: $\langle 1,2,3,5\rangle, \langle 1,2,4,5\rangle, \langle 1,3,4,5\rangle, \langle 1,2,3,6\rangle, \langle 1,2,4,6\rangle, \langle 1,3,4,6\rangle$

2: $\langle 1,2,3,6\rangle, \langle 1,2,4,6\rangle, \langle 2,3,4,6\rangle, \langle 1,2,3,9\rangle, \langle 1,2,4,9\rangle, \langle 2,3,4,9\rangle$

3: $\langle 1,2,3,5\rangle, \langle 1,3,4,5\rangle, \langle 2,3,4,5\rangle, \langle 1,2,3,8\rangle, \langle 1,3,4,8\rangle, \langle 2,3,4,8\rangle$

4: $\langle 1,2,4,7\rangle, \langle 1,3,4,7\rangle, \langle 2,3,4,7\rangle, \langle 1,2,4,8\rangle, \langle 1,3,4,8\rangle, \langle 2,3,4,8\rangle$
$\langle 1,2,4,10\rangle, \langle 1,3,4,10\rangle, \langle 2,3,4,10\rangle$

1: $\langle 1,2,3,6\rangle, \langle 1,2,5,6\rangle, \langle 1,3,5,6\rangle$

2: $\langle 1,2,3,4\rangle, \langle 1,2,3,6\rangle, \langle 1,2,3,9\rangle, \langle 1,2,4,5\rangle, \langle 1,2,5,6\rangle, \langle 1,2,5,9\rangle$
$\langle 2,3,4,5\rangle, \langle 2,3,5,6\rangle, \langle 2,3,5,9\rangle$

3: $\langle 1,2,3,4\rangle, \langle 1,2,3,8\rangle, \langle 1,3,4,5\rangle, \langle 1,3,5,8\rangle, \langle 2,3,4,5\rangle, \langle 2,3,5,8\rangle$

5: $\langle\rangle$

Fig. 4. State Transition of the random walk for substructure sampling. (a)(i) A database graph G_i with the current state of FS^3's random walk (a) (ii) Neighborhood information of the current state $\langle 1,2,3,4\rangle$. (b)(i) The state of random walk on G_i (Figure 4(a)) after one transition (b) (ii) Updated Neighborhood information.

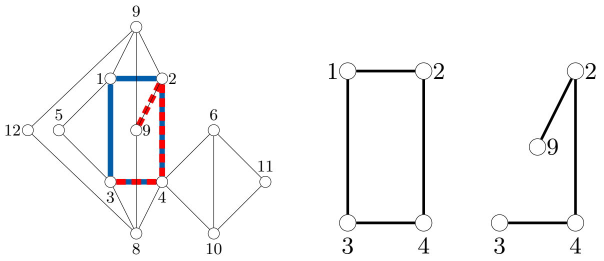
random walk is chosen so that the stationary distribution of the random walk matches with a user-chosen target distribution. FS^3's target distribution favors ℓ-subgraph so that the sampling process can predominantly sample frequent subgraphs. FS^3's MCMC walk changes state by walking from one ℓ-subgraph (say g) to a neighboring ℓ-subgraph. In our neighborhood definition, for a ℓ-subgraph all other ℓ-subgraphs that have $\ell - 1$ vertices in common are its neighbor subgraph/state. To obtain a neighbor subgraph of g, FS^3 simply replaces one of the existing vertices of g with another vertex which is not part of g but is adjacent to one of g's vertices. Also, note that in g, FS^3 includes all the edges of G_i that are induced by the set of the selected vertices, so the sampled subgraph of FS^3 is always a connected induced subgraph of the graph G_i. For a given graph G_i in $\mathcal{G}$, the currently sampled ℓ-subgraph is saved so that the random walk over G_i can be resumed in a later iteration if the graph G_i is again selected in the first stage of the sampling iteration. Below, we show an example of state transition of FS^3.

Example: Suppose FS^3 is sampling 4-subgraphs from the graph G_i shown in Figure 4(a)(i) using MCMC sampling. Let, at any given time the 4-subgraph, $\langle 1,2,3,4\rangle$ (shown in bold lines) is the current state of this random walk. In Figure 4(a)(ii), we list its neighbor states as four comma-separated lists, one in each row. The neighbor-list in the top row is labeled by '1', which indicates that these neighbors can be obtained from the current 4-subgraph $\langle 1,2,3,4\rangle$ by retaining the vertex 1 and replacing exactly one of the remaining vertices ($\{2,3,4\}$) with a new vertex which is adjacent to vertex 1, ensuring connectedness. Similarly, the neighbors in the second list are obtained by retaining the vertex 2 and replacing one of the remaining vertices with a vertex from 2's adjacency list. The information in the third and fourth lists are populated in a similar manner. As shown in the top-list, $\langle 1,2,3,5\rangle$ is a neighbor of $\langle 1,2,3,4\rangle$; if the random walk transitions to this state, the current state becomes $\langle 1,2,3,5\rangle$, which is shown in Figure 4(b)(i). In Figure 4(b)(ii), we show the updated neighbor lists considering the new state. Note that, here also we have 4 set of neighbors corresponding to 4 vertices of $\langle 1,2,3,5\rangle$. The neighbor-list corresponding to vertex 5 is empty, as besides 1 and 3 (which are part of current state), 5 has no other adjacent vertices that can be used as a replacement vertex to build a new state.

C. *Finding Sub-Network Embedding in the Interface Graph*

Note that FS^3 samples ℓ-node induced subgraphs from the database graphs using a sampling-based method. It makes FS^3 scalable over large networks, but to achieve scalability it also loses completeness, i.e., for a given frequent subgraph, its support-list i.e. relative support-list may miss some of the graphs in $\mathcal{G}$ in which the pattern occurs. Therefore, at the end of the sampling process, for each of the top-k frequent

1545-5963 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.



(a) An interface graph (b) Subnetwork patches

Fig. 5. Subnetwork patches embedded in an interface graph.

subgraph patterns, we use a subgraph isomorphism algorithm for finding the embedding of the pattern in all the graphs in the database. This step completes the relative support-list of a frequent subgraph pattern and we get the actual support-list. Besides, it provides a mapping between the pattern nodes and a subset of interface graph nodes such that the mapping respects the vertex label. Thus, the embedding process enables us to inspect the subgraph pattern within the native context of residue contact graph.

Additionally, we observe that, in some cases most of the top-frequent subgraphs are almost identical except one or two nodes. After embedding, they map to a patch of the functional motif, such that super-imposition of the embedded patches of multiple top-frequent patterns cover the entire motif. For visualizing this step, we present Figure 5. In Figure 5(a), we show an example interface graph. The node labels in the figure represent residue ids. In Figure 5(b), we list two top-frequent patterns. Bold blue and dashed red lines in Figure 5(a) show that super-imposing the embedding of the top two patterns retrieves the entire motif consisting of residues 1, 2, 3, 4 and 9 (shown in color).

For HIV-1 protease, we consider only 10 of the most frequent subgraphs, and the embedding of these subgraphs discovers the entire 16-residue dimeric lock motif in 323 out of 329 patterns. Similar treatment for the TIM protein using 20 most frequent subgraphs finds the dimeric lock in 50 out of 86 structures.

D. Statistical Significance Test of Discovered Patterns

Statistical significance test of a frequent subgraph g determines the probability (p-value) of observing g as a frequent pattern at equal or a higher support value in a database of random graphs, where the random graphs are constructed from a null model. The subgraph pattern g is statistically significant when it is highly unlikely for g to be frequent under the null model. In existing works [14], statistical significance test has been used to calculate the p-value of network motifs, which are mined from a single large graph. In these works, a set of random graphs are generated from the input graph under a specified random graph model and the subgraphs which appear in the input graph at a much higher frequency than in the random graphs are considered as significant. But, this method does not apply for our task, because in our task the we have a database of input graphs instead of a single graph. So, we generalize the above method as below. First, we generate a set [3] of clone graph databases each containing the same number of random graphs as our input graph database. The random graphs in the clone databases are generated using a null model, details of which is discussed in the next paragraph. Then, we run our algorithm on the input graph database and on each of the clone graph databases to discover the top-k patterns and their frequencies in these datasets. Finally, we compute the z-score of a mined subgraph pattern. If the support of a subgraph pattern g in an input graph database is $s_{real}(g)$ and the average support and standard deviation in an ensemble of random graph database are $s_{avg}(g)$ and $s_{dev}(g)$, then z-score of g is calculated as shown below:

$$z\text{-score}(g) = \frac{s_{real}(g) - s_{avg}(g)}{s_{dev}(g)} \tag{2}$$

Then we obtain the p-value of g by considering that the support of a top-k pattern under the null hypothesis is distributed as a normal distribution. A small p-value confirms that the null hypothesis is discarded and the subgraph pattern g is statistically significant.

Random Graph Generation for Null Model As we have discussed earlier, for significant test we build a set of clone graph databases, each containing the same number of random graphs as the input graph database. Under the null model, the random graphs in the clone databases have the same degree distribution and vertex label distribution. The null hypothesis is that a frequent subgraph g is also frequent in the clone databases.

Generating a random graph (i.e. generating random 0-1 matrices) by keeping the degree distribution the same is a well studied problem. We use switching method proposed by [52]. In this method, for a given adjacency matrix of a particular graph, all the adjacency matrices which can be obtained by switching alternating 1's and 0's along the alternative rectangles or the alternating hexagons are considered to be the neighbor states. A Markov chain can be formed from this state transition and [52] has shown that if we take a particular state after p or less transitions we sample a random graph uniformly at random where p represents the minimum of the total number of zero's and one's in the random network. This algorithm samples correctly in the limit of long run and in practice is found to give good results compared to other methods [53]. In Figure 6, we show an example. Figure 6 (a) represents the input network (a line graph) whereas Figure 6(b) and Figure 6 (c) show two random graphs generated using the switching technique. From the figures, we can see that randomization has rewired the nodes by preserving the degree of all the nodes in the input graph. We do not alter the vertex labels, so the vertex label distribution is identical to the original graph.

For both the TIM and HIV-1 protease structures (discussed in Section V-A and V-B, respectively), we generate 20 (chosen arbitrarily) clone databases containing random graphs, i.e., for each graph in the host database, we generate 20

[3] size of this set can be anything between 10 and 100, the higher the size the better is the estimates.

1545-5963 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

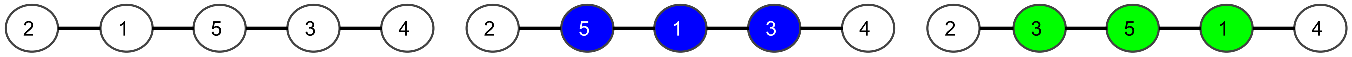


Fig. 6. Random graph generation from a particular graph. Figure (a) is the input graph, Figures (b) and (c) are random graphs using switching algorithm described in Section IV-D. Interchanges are shown in blue and green color.

random copies of that graph using the method described in the above paragraph. Then we apply FS^3 on both the host (input) database and each of the random graph databases separately with the *same configuration* (size-ℓ) used for the input database. Our experiments show that all our frequent patterns (size 16 for HIV-1, and size 20 for TIM) are highly significant as their frequency in the database of random graphs is zero, but the average support of HIV-1 frequent patterns is 320 (for a database size 329) and the average support of TIM frequent patterns is 50 (for a database size 86). This yields a p-value less than 0.00001 using Laplace correction for the denominator, thus making all the discovered frequent patterns in both datasets highly significant. Interestingly, no frequent patterns exist in the clone databases of random graphs; in fact, the highest support of any subgraph in each of these clone databases is exactly one, that is each subgraph appears in only one random graph.

V. Experimental Results

In this section, we present our experimental findings. Section V-A and V-B shows that our graph-mining method retrieves the dimerization locks in each of the protein structure with multiple conformations whereas in Section V-C, we show that our approach captures class specific active sites for the six top-level classes of enzymes each composed of multiple protein structures with a single conformation. In Section V-A and V-B, we report average pairwise RMSD (Root Mean Square Deviation) distance among conformers[4]. For calculating RMSD distance, we use Kabsch algorithm [54] and Quaternion algorithm [55]. Kabsch algorithm [54] is a simple procedure which determines a best rotation of a given vector set into a second vector set by minimizing the weighted sum of squared deviations. On the other hand, Quaternion algorithm [55] solves for the orientation and the position of an object by minimizing a single cost function associated with the sum of the orientation and position errors.

A. HIV-1 protease structures

HIV-1 PR dimerization occurs at the interface between two homologue structures- each subunit having 99 residues. Each subunit structure can be divided into functionally important components (Fig. 7A): 1) Terminal domains (blue, NT strand: residues 1-4 and CT strand: residues 96-99) that form the base of the protease structure. 2) Flap domain (orange, residues 37-58) that opens and closes the structure for substrate recruitment and product release. The coordination of motion between 3) Fulcrum (red, residues 9-21) and 4) Cantilever (green, residues 59-75) controls the opening/closing motion of the Flap domain.

NT (residues 1-4) and CT (residues 96-99) strands from one subunit form a ridge where the CT strand from the partner subunit gets inter-digitated, and vice versa (Fig 7B). This interlocked configuration of the terminal strands forms a strongly-bound dimeric base which facilitates the opening-closing motion of the flap tips of the Flap domains.

We selected 329 HIV-1 structures from PDB [56] such that each structure has no missing residues. Subsequently, we have created an interfacial network (connected graph) for each structure considering the interfacial residues that are within 8 Angstrom (Å) distance from any residue from the partner subunit. We also connect two residues within the same subunit if their distance is within 4 angstrom, i.e., we set γ = 8Å and δ = 4Å. The average number of nodes and edges for these networks are 64.00 and 242.00 respectively. Then, we mined these 329 connected graphs using FS^3, our graph mining method. If the proteins are structurally similar, the frequent subgraphs are more likely to form; so one may opt for more precise results by setting smaller values of δ and γ. For this purpose, structural similarity of a collection of proteins should be obtained by optimally superimposing the proteins one on top of another, and then computing The average RMSD distance. We perform the same by using both the Kabsch and the Quaternion algorithm on our HIV-1 dataset. The median RMSD value was 0.7305 (minimum =0.0, maximum=2.74) when the statistics was calculated over all the 329 conformers of HIV-1.

Figure 7C labels the 16 residues of four strands that form the dimeric lock at the base - four residues in each strand. For a pattern of size 16, our method retrieves 13 of these base forming residues. Three residues (I3 on NT B, I3 and T4 on NT A) shown in red were not included, rather K5 and T6 on the coil connecting NT B and Fulcrum and T91 on the helical region at the N-terminal end of CT A got included.

B. TIM structures

TIM is the fifth enzyme in the glycolysis pathway that produces energy in all living organisms. The functional oligomeric state of TIM is a homo-dimeric structure in most mesophilic organisms. A TIM subunit has a central barrel formed by eight strands (β1$-$ β8) which is surrounded by eight helices (α1-α8). Eight back loops (BL1-BL8) connect from helix to strand and eight front loops (FL1-FL8 or simply Loop 1$-$Loop 8) connect from strand to helix. Details can be found in [57] (Fig. 8A). Two monomeric subunits form the dimeric TIM structure through interaction of a pair of symmetric locks at their interface. We construct interfacial network for each

[4] https://github.com/charnley/rmsd

1545-5963 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

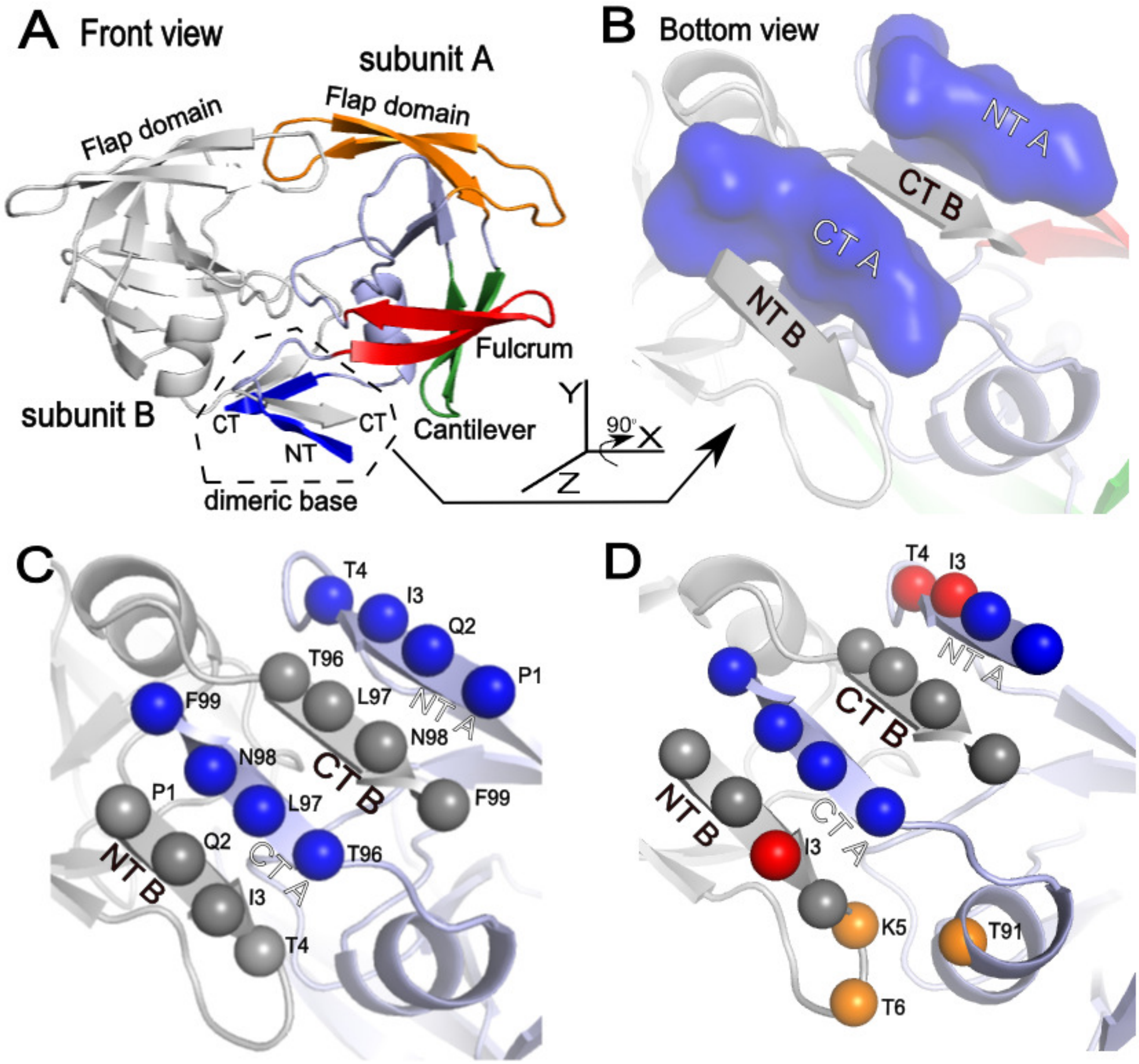


Fig. 7. HIV-1 protease (HIV-1 PR) functional components, interface formation, and computationally retrieved residues from the interface residue network. Panel A shows the macromolecular architecture of the protease (based on PDB: 1a30, a closed conformation), Panel B show the lock formation at the base, Panel C shows the residues in spheres at the dimeric base, and Panel D shows the computationally retrieved residues from the interface networks. (A) Front view of HIV-1 PR dimeric structure (modified from Fig. 2 of [33]). The functionally important components are colored and labeled in subunit A. N-terminal (NT) and C-terminal (CT) strands are colored blue: NT residues 1-4 and CT residues 96-99. NT and CT strands of one subunit form a ridge where CT strand of the other subunit is locked, and vice versa. Fulcrum (red, residues 9-21) - at one end of this component is the C-terminus and on the other end there are the active site region. Flap domain (orange, residues 37-58) has three main regions. Cantilever (green, residues 59-75) is located at the C-terminal end of the Flap domain. (B) Lock formation at the base of the structure - NT and CT strands of chain A form a ridge where CT from B is inserted and vice versa. (C) The residues on NT and CT of each chain forming the lock are identified (PDB 1a30). (D) Blue ones are the correctly recognized interface residues by graph mining. Three residues forming the lock shown in the panels B and C that the mining algorithm failed to identify are colored red. Instead, the mining included the orange residues in the pattern that are not part of the lock pair.

of the 86 triosephosphate isomerase (TIM) PDB structures with γ = 8Å and δ = 4Å. The average number of nodes and edges for these networks are 158.50 and 884.75 respectively. The average RMSD distance using both the Kabsch and the Quaternion algorithm is: 5.76 (min=0.0, max=24.64) and it was calculated over 25 TIM structures for which the number of C_α carbons were the same.

A dimer of two subunits is formed by two symmetric locks at the interface: Loop 1 and Loop 4 of one subunit form a ridge wherein Loop 3 of the partner subunit gets engaged, and vice versa. Figure 8A shows such a pair of locks at the dimeric interface of a TIM structure (PDB 1ypi). The space-filled view in Fig. 8B illustrates one of these locks more clearly. Figure 8C illustrates the residues of the involved loops in spheres (L1 of chain B: $F_{11}K_{12}$ $L_{13}N_{14}G_{15}$ S_{16}, L4 of chain B: G_{94} $H_{95}S_{96}E_{97}$ $R_{98}R_{99}$ $S_{100}Y_{101}$ $F_{102}H_{103}$ $E_{104}D_{105}$, L3 of chain A: $Q_{64}N_{65}$ $A_{66}Y_{67}L_{68}$ $K_{69}A_{70}\mathbf{S_{71}}$ $\mathbf{G_{72}A_{73}F}$ $\mathbf{_{74}T_{75}G_{76}}$ $\mathbf{E_{77}N_{78}S_{79}}$).

Our graph-mining method retrieves the key residues of the locking mechanism. When the pattern-size is 12, the retrieved residues are: L1 (chain A): 10, 12; L4 (chain A): 95, 97, 98; L3

1545-5963 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2756879, IEEE/ACM Transactions on Computational Biology and Bioinformatics

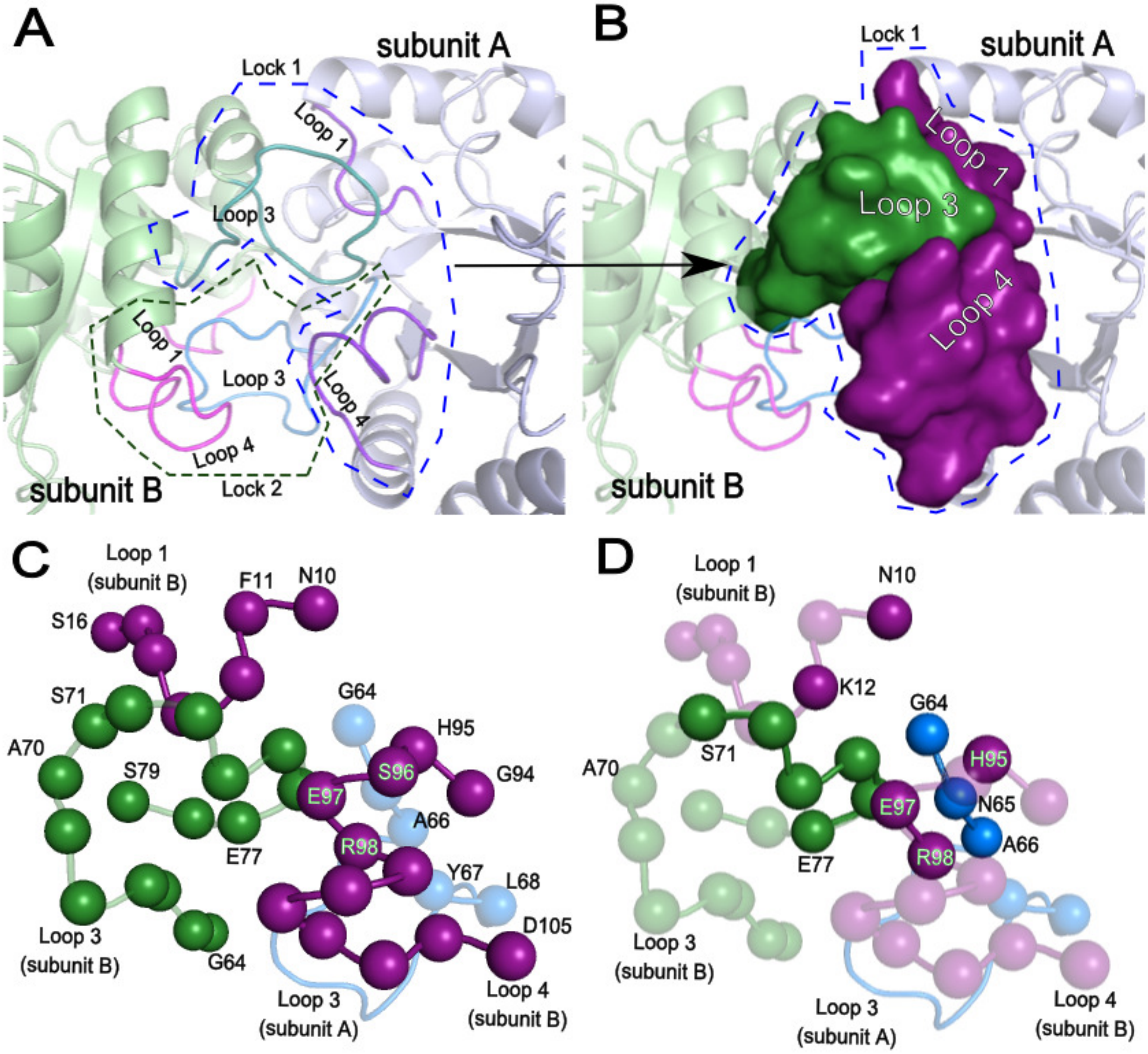


Fig. 8. **Type 1 interface of TIM dimeric structure. (A) Loop 3 from subunit A and Loop 1 and Loop 4 from subunit B form a lock at the interface, and vice versa. (B) Surface view of Lock 1. (C) Residues of the loops involved in Lock 1 are shown in spheres. (D) Retrieved residues in Lock 1 are shown in bright color and others are deemed.**

(chain B): 72-77; N-terminal base of L3 (chain A): 64. And, when the pattern-size is 12 are: L1 (chain A): 10, 12; L3 (chain B): 72,..., 77; L4 (chain A): 95, 97, 98; and N-terminal base of L3 (chain A): 63, 64, 65, 66. The residues from the interlocking mechanism that are retrieved by our method are shown in bright spheres (Fig. 8D). Interestingly, all the residues of Loop 3 ($S_{71}G_{72}$ $A_{73}F_{74}T_{75}$ $G_{76}E_{77}$ - 7 residues [58]) are successfully retrieved. Moreover, the retrieved patterns reveal that a few residues at the N-terminal region of Loop 3 from chain A (residues G_{64}, N_{65}, and A_{66}) engages in the lock formation.

C. *Enzymes*

Enzymes are known to be macromolecular catalysts that speed up biochemical reactions by providing an alternative reaction pathway of lower activation energy. In the absence of enzymatic catalysis, most biochemical reactions are so slow that they would not occur under the mild conditions of temperature and pressure that are compatible with life [59]. Enzymes accelerate the rates of such reactions by well over a million-fold, so reactions that would take years in the absence of catalysis can occur in fractions of seconds if catalyzed by the appropriate enzyme. Enzymes bind their reactants or substrates at a small portion of their structure that is known as the active site. Active sites are substructures on the surface of an enzyme, usually composed of amino acids from different parts of the polypeptide chain that are brought together in the tertiary structure of the folded protein [59]. Hence, mining functional motifs (active sites) from the interface region of enzymes is important for understanding the underlying mechanisms that allow them to interact with other molecules and perform their vital functions that sustain life in the cells. The International Union of Biochemistry and

1545-5963 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2756879, IEEE/ACM Transactions on Computational Biology and Bioinformatics

Molecular Biology[5] has developed a classification system for enzymes[6] that, at its top-level, divides them into six groups namely:

- *Oxydoreductase (EC1)*: catalyze oxidation/reduction reactions.
- *Transferase (EC2)*: transfer of a chemical group from substrate to product.
- *Hydrolase (EC3)*: cleavage of bonds by hydrolysis.
- *Lyase (EC4)*: elimination of various bonds by means other than hydrolysis and oxidation.
- *Isomerase (EC5)*: catalyze isomerization changes within a single molecule.
- *Ligase (EC6)*: join two molecules with covalent bonds.

Unlike the previous two experiments where we mined patterns that are shared across the different conformations of the same protein, in this experiment, we are interested in mining functional motifs that are shared by multiple protein structures within the same group of enzymes but not across the different classes. That is to say, class specific active sites that allow each of the enzyme classes to exert a specific function. Since enzymes need to bind to their substrates at their active sites to perform their biological functions, mining class-wise frequent patterns at the interface region of enzymes could help to unravel class specific active sites. We use enzymes from the Dobson and Doig (D&D) protein structure dataset [34] which originally consists of 1178 proteins divided into a group of 691 enzymes and a second group of 487 non-enzymes. We consider only the subset of oligomeric protein structures from the enzymes, $i.e$, structures with at least two sub-units. The remaining set of enzymes is composed of 326 protein structures. Table I shows the number of protein structures in each class, the number of EC subclasses and sub-subclasses in each group, as well as, the average number of nodes and edges from the derived graphs.

TABLE I
INTERFACIAL NETWORK STATISTICS FOR OUR SUBSET OF ENZYMES FROM THE DOBSON AND DOIG (D&D) PROTEIN STRUCTURE DATASET [34]

Class	#structures	#Subclasses	#Sub-subclasses	Avg. #Nodes	Avg. #Edges
$EC1$	76	16	35	151	487
$EC2$	84	8	21	102	318
$EC3$	91	8	29	82	262
$EC4$	40	4	9	128	414
$EC5$	21	5	10	118	367
$EC6$	14	4	6	134	403

We have constructed an interfacial network for each PDB structure of the set, based on Equation (1) with γ = 10Å and δ = 4Å. After this step, we obtained a set of undirected, vertex-labeled graphs—each corresponding to one enzyme protein structure. We used FS[3] to discover function specific subgraph motifs across the six different classes of enzymes. We mined 200 most frequent patterns for each of the following sizes [7] 5, 6, 7 and 8 from each of the six enzyme classes.

We first validate whether the discovered patterns are abundant across all six enzyme classes or they are frequent only within an enzyme class. Since the enzyme classes are derived from their function, patterns that are frequent only within an enzyme class are functional motif for that class of enzyme. For this validation, we count the number of patterns that occurs over multiple classes of enzymes. For a clean presentation, in Table II we only show the number of patterns which overlap over a pair of enzyme classes. Along the rows we list $\binom{6}{2} = 15$ pairs of enzyme classes and along the column we list the size of patterns. Each cell entry shows the number of overlapping patterns across the corresponding pair of enzyme classes for the given pattern size. For example, there are 14 patterns (out of 200 most frequent patterns) of size-5 which overlaps across enzyme class 1 and enzyme class 2. We notice that the overlap between the sets of patterns mined from each class is very small for the sizes 5, 6 and 7, and that there is no overlap at all at the size 8. Thus the number of overlaps decreases while increasing the size of patterns. This shows that our modeling and mining method allows to unravel class specific patterns at the interfacial region. Besides, the fact that each of the classes performs a particular function also suggests that the discovered patterns are active sites and they are specific to functions performed by the enzymes in that class.

In fact, the active site of an enzyme is composed of two components. The first component is the catalytic site that is known to be small ($2 - 4$ amino acids [60], [61]), highly conserved, and allows the enzyme to perform its function. The second component is the binding site which allows the recognition and precise positioning of an enzyme's substrate in proximity to the chemically active catalytic residues and lower the energy of the transition state, which aids catalysis [60]. Figure 9 shows an example of a protein structure namely the *L-3-hydroxyacyl-CoA dehydrogenase* (PDB IDs: 1F14) from the EC1 class of our dataset and the mapping of a frequent pattern of size 8 that we discovered using our subgraph mining method. The pattern contains a catalytic site composed of the residues ”Glutamine”, ”Asparagine”, and ”Serine” that have been identified at the same structure in the Catalytic Site Atlas[8] [60], [62], a database of both hand-curated and automatically annotated catalytic sites in enzyme structures. Since the catalytic and binding sites co-occur together as part of the same active site, we consider the five remaining residues (”Lysine”, ”Leucine”, and 3 ”Alanine”) from the pattern as of the binding site. Figure 9 shows the catalytic and binding site in red and blue respectively.

VI. CONCLUSION & FUTURE WORK

In this work, we proposed a method for the discovery of functional motifs from the interface region of dimeric protein structures. Our method uses a graph representation of the interface region of these structures, and mines a fixed-size highly frequent subgraphs over those graphs. We then use a small collection of subgraphs to discover functional motifs at the interface region of the structures. In our experiments, we showed that our method discovers the oligomeric lock motif

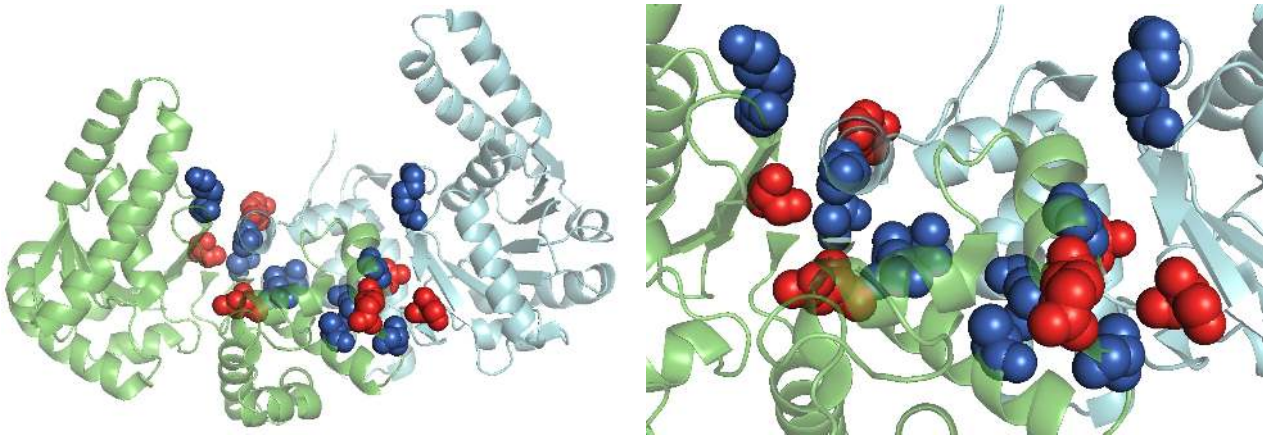
[5] http://iubmb.org/

[6] http://www.enzyme-database.org/

[7] Size of a subgraph pattern is the number of vertices in that pattern.

[8] http://www.ebi.ac.uk/thornton-srv/databases/CSA/

1545-5963 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.



(a) Front view of the L-3-hydroxyacyl-CoA dehydrogenase protein structure

(b) Zoomed view of the active site of the structure: in red and blue are residues from the catalytic and binding sites respectively

Fig. 9. Retrieved frequent pattern representing active site at the L-3-hydroxyacyl-CoA dehydrogenase protein structure. (a) A front view of the entire structure with the active site and (b) a zoomed view of the active site with the catalytic site (in red) and binding site (in blue).

TABLE II
THE NUMBER OF PATTERNS OVERLAPS WITHIN DIFFERENT GROUPS FOR A SPECIFIC SIZE, ℓ AND TOP-200 PATTERNS.

	# Overlaps for a specific ℓ			
Classes	$\ell = 5$	$\ell = 6$	$\ell = 7$	$\ell = 8$
$EC1 - EC2$	14	8	7	0
$EC1 - EC3$	0	0	0	0
$EC1 - EC4$	20	2	0	0
$EC1 - EC5$	10	0	0	0
$EC1 - EC6$	5	4	1	0
$EC2 - EC3$	17	2	2	0
$EC2 - EC4$	14	0	0	0
$EC2 - EC5$	11	3	0	0
$EC2 - EC6$	8	0	0	0
$EC3 - EC4$	12	0	0	0
$EC3 - EC5$	0	0	0	0
$EC3 - EC6$	3	0	0	0
$EC4 - EC5$	6	0	0	0
$EC4 - EC6$	8	0	0	0
$EC5 - EC6$	3	0	0	0

in the majority of the structures for both HIV-1 protease and TIM protein. We also showed that our method discovers class specific active sites at the interfacial region of the six top-level classes of enzymes.

There are significant scopes for extending this work. First, we plan to make our FS^3 software a stand-alone tool for the functional motif discovery at the interfacial region of proteins. As we have observed highly frequent patterns of a given size although captures the functional motifs, each such patterns sometimes misses a few residues of a functional motifs. At this stage, we manually patch together a collection of patterns to identify the entire functional motifs. One immediate future work is to identify a cluster of similar patterns which overlap the core of a functional motif and then automatically patch them together to discover the functional motifs. Also, we are planning to extend the functionality of our FSM based functional motif discovery tool. Currently, our FSM method counts the frequency of a pattern by its identical occurrences over different graphs. As future work, we are planning to extend our approach with a selection module that accounts for amino acids similarity as in [38], [63] for counting occurrences of a pattern.

VII. Acknowledgments

We thank all anonymous reviewers from BIOKDD'16 and TCBB for their thoughtful feedback that helped us to improve this manuscript. A part of funding for this research is made available through Mohammad Hasan's NSF CAREER Award (IIS-1149851) grant.

References

[1] X. Zhang, T. Perica, and S. A. Teichmann, "Evolution of protein structures and interactions from the perspective of residue contact networks," *Current opinion in structural biology*, vol. 23, no. 6, pp. 954–963, 2013.

[2] A. Tomovic and E. J. Oakeley, "Computational structural analysis: multiple proteins bound to dna," *Plos One*, vol. 3, no. 9, pp. 32–43, 2008.

[3] R. Chen and Z. Weng, "A novel shape complementarity scoring function for protein-protein docking," *Proteins: Structure, Function, and Bioinformatics*, vol. 51, no. 3, pp. 397–408, 2003.

[4] R. Chen, L. Li, and Z. Weng, "Zdock: An initial-stage protein-docking algorithm," *Proteins: Structure, Function, and Bioinformatics*, vol. 52, no. 1, pp. 80–87, 2003.

[5] I. S. Moreira, J. M. Martins, J. T. Coimbra, M. J. Ramos, and P. A. Fernandes, "A new scoring function for protein–protein docking that identifies native structures with unprecedented accuracy," *Physical Chemistry Chemical Physics*, vol. 17, no. 4, pp. 2378–2387, 2015.

[6] N. Hulo, A. Bairoch, V. Bulliard, L. Cerutti, E. De Castro, P. S. Langendijk-Genevaux, M. Pagni, and C. J. Sigrist, "The prosite database," *Nucleic acids research*, vol. 34, no. suppl 1, pp. D227–D230, 2006.

[7] C. G. Nevill-Manning, T. D. Wu, and D. L. Brutlag, "Highly specific protein sequence motifs for genome analysis," *Proceedings of the National Academy of Sciences*, vol. 95, no. 11, pp. 5865–5871, 1998.

[8] J. Y. Huang and D. L. Brutlag, "The emotif database," *Nucleic acids research*, vol. 29, no. 1, pp. 202–204, 2001.

[9] R. Saidi, M. Maddouri, and E. M. Nguifo, "Protein sequences classification by means of feature extraction with substitution matrices," *BMC bioinformatics*, vol. 11, no. 1, p. 1, 2010.

1545-5963 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2756879, IEEE/ACM Transactions on Computational Biology and Bioinformatics

[10] M. Vendruscolo, N. Dokholyan, E. Paci, and M. Karplus, "Small-world view of the amino acids that play a key role in protein folding," *Physical Review E*, vol. 65, no. 6, p. 061910, 2002.

[11] L. H. Greene, "Protein structure networks," *Briefings in functional genomics*, vol. 11, no. 6, pp. 469–478, 2012.

[12] W. Dhifli and A. B. Diallo, "Protnn: fast and accurate protein 3d-structure classification in structural and topological space," *BioData Mining*, vol. 9, no. 1, p. 30, 2016.

[13] M. Bhattacharyya, S. Ghosh, and S. Vishveshwara, "Protein structure and function: Looking through the network of side-chain interactions," *Current Protein and Peptide Science*, vol. 17, no. 1, pp. 4–25, 2016.

[14] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, "Network motifs in the transcriptional regulation network of escherichia coli," *Nature Genetics*, vol. 31, pp. 1061–4036, 2002.

[15] E. Wong, B. Baur, S. Quader, and C.-H. Huang, "Biological network motif detection: principles and practice," *Briefings in bioinformatics*, p. bbr033, 2011.

[16] B. K. Kamapantula, M. L. Mayo, E. J. Perkins, and P. Ghosh, "The structural role of feed-forward loop motif in transcriptional regulatory networks," *Mobile Networks and Applications*, pp. 1–15, 2016.

[17] T. K. Saha and M. A. Hasan, "Finding network motifs using MCMC sampling," in *Complex Networks VI: Proceedings of the 6th Workshop on Complex Networks CompleNet 2015*, G. Mangioni, F. Simini, M. S. Uzzo, and D. Wang, Eds. Springer International Publishing, 2015, pp. 13–24.

[18] S. Gao, A. Chen, A. Rahmani, J. Zeng, M. Tan, R. Alhajj, J. Rokne, D. Demetrick, and X. Wei, "Multi-scale modularity and motif distributional effect in metabolic networks," *Current Protein and Peptide Science*, vol. 17, no. 1, pp. 82–92, 2016.

[19] D. Cook and L. Holder, "Substructure discovery using minimal description length and background knowledge," *Journal of Artificial Intelligence Research*, vol. 1, pp. 231–255, 1994.

[20] A. Inokuchi, T. Washio, and H. Motoda, "An apriori-based algorithm for mining frequent substructures from graph data," in *Proc. of PKDD*, 2000, pp. 13–23.

[21] M. Kuramochi and G. Karypis, "An Efficient Algorithm for Discovering Frequent Subgraphs," *IEEE Trans. on Knowledge and Data Engineering*, vol. 16, no. 9, pp. 1038–1051, 2004.

[22] X. Yan and J. Han, "gspan: Graph-based substructure pattern mining," in *Proc. of ICDM*, 2002, pp. 721–724.

[23] L. T. Thomas, S. R. Valluri, and K. Karlapalem, "Margin: Maximal frequent subgraph mining," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 4, no. 3, p. 10, 2010.

[24] V. Chaoji, M. Hasan, S. Salem, and M. Zaki, "An Integrated, Generic Approach to Pattern Mining: Data Mining Template Library," *Data Mining and Knowledge Discovery Journal*, vol. 17, no. 3, pp. 457–495, 2008.

[25] S. Nijssen and J. N. Kok, "The gaston tool for frequent subgraph mining," *Electr. Notes Theor. Comput. Sci.*, vol. 127, no. 1, pp. 77–87, 2005.

[26] M. Elseidy, E. Abdelhamid, S. Skiadopoulos, and P. Kalnis, "Grami: Frequent subgraph and pattern mining in a single large graph," *Proceedings of the VLDB Endowment*, vol. 7, no. 7, pp. 517–528, 2014.

[27] D. Aparicio, P. Paredes, and P. Ribeiro, "A scalable parallel approach for subgraph census computation," in *Euro-Par 2014: Parallel Processing Workshops*. Springer, 2014, pp. 194–205.

[28] C. H. Teixeira, A. J. Fonseca, M. Serafini, G. Siganos, M. J. Zaki, and A. Aboulnaga, "Arabesque: a system for distributed graph mining," in *Proceedings of the 25th Symposium on Operating Systems Principles*. ACM, 2015, pp. 425–440.

[29] J. Huan, W. Wang, and J. Prins, "Efficient mining of frequent subgraphs in the presence of isomorphism," in *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE, 2003, pp. 549–552.

[30] M. Bhuiyan and M. A. Hasan, "An Iterative MapReduce Based Frequent Subgraph Mining Algorithm," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 3, pp. 608–620, 2015. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2014.2345408

[31] T. Horváth, B. Bringmann, and L. De Raedt, "Frequent hypergraph mining," in *Inductive Logic Programming*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, vol. 4455, pp. 244–259.

[32] V. Chaoji, M. Hasan, S. Salem, J. Besson, and M. Zaki, "ORIGAMI: A Novel and Effective Approach for Mining Representative Orthogonal Graph Patterns," *Statistical Analysis and Data Mining*, vol. 1, no. 2, pp. 67–84, June 2008.

[33] A. R. Katebi, K. Sankar, K. Jia, and R. L. Jernigan, "The use of experimental structures to model protein dynamics," in *Molecular Modeling of Proteins*. Springer, 2015, pp. 213–236.

[34] P. D. Dobson and A. J. Doig, "Distinguishing enzyme structures from non-enzymes without alignments," *Journal of molecular biology*, vol. 330, no. 4, pp. 771–783, 2003.

[35] G. Amitai, A. Shemesh, E. Sitbon, M. Shklar, D. Netanely, I. Venger, and S. Pietrokovski, "Network analysis of protein structures identifies functional residues," *Journal of molecular biology*, vol. 344, no. 4, pp. 1135–1146, 2004.

[36] K. Brinda and S. Vishveshwara, "Oligomeric protein structure networks: insights into protein-protein interactions," *Bmc Bioinformatics*, vol. 6, no. 1, p. 296, 2005.

[37] A. Giuliani, A. Krishnan, J. P. Zbilut, and M. Tomita, "Proteins as networks: usefulness of graph theory in protein science," *Current Protein and Peptide Science*, vol. 9, no. 1, pp. 28–38, 2008.

[38] W. Dhifli, R. Saidi, and E. M. Nguifo, "Smoothing 3d protein structure motifs through graph mining and amino acid similarities," *Journal of Computational Biology*, vol. 21, no. 2, pp. 162–172, 2014.

[39] P. P. Wangikar, A. V. Tendulkar, S. Ramya, D. N. Mali, and S. Sarawagi, "Functional sites in protein families uncovered via an objective and automated graph theoretic approach," *Journal of molecular biology*, vol. 326, no. 3, pp. 955–978, 2003.

[40] C. Böde, I. A. Kovács, M. S. Szalay, R. Palotai, T. Korcsmáros, and P. Csermely, "Network analysis of protein dynamics," *Febs Letters*, vol. 581, no. 15, pp. 2776–2782, 2007.

[41] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. Vishwanathan, A. J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels," *Bioinformatics*, vol. 21, no. suppl 1, pp. i47–i56, 2005.

[42] A. Harrison, F. Pearl, R. Mott, J. Thornton, and C. Orengo, "Quantifying the similarities within fold space," *Journal of molecular biology*, vol. 323, no. 5, pp. 909–926, 2002.

[43] E. Krissinel and K. Henrick, "Protein structure comparison in 3d based on secondary structure matching (ssm) followed by c-alpha alignment, scored by a new structural similarity function," in *Proceedings of the 5th International Conference on Molecular Structural Biology*, vol. 88. Vienna, 2003.

[44] R. Shen, N. C. Goonesekere, and C. Guda, "Mining functional subgraphs from cancer protein-protein interaction networks," *BMC systems biology*, vol. 6, no. Suppl 3, p. S2, 2012.

[45] R. Shen, X. Wang, C. Guda, and C. B. Guda, "Discovering distinct functional modules of specific cancer types using protein-protein interaction networks," *BioMed Research International*, vol. 2015, 2015.

[46] C. Ding, X. He, R. F. Meraz, and S. R. Holbrook, "A unified representation of multiprotein complex data for modeling interaction networks," *Proteins: Structure, Function, and Bioinformatics*, vol. 57, no. 1, pp. 99–108, 2004.

[47] J. Desaphy, E. Raimbaud, P. Ducrot, and D. Rognan, "Encoding protein–ligand interaction patterns in fingerprints and graphs," *Journal of chemical information and modeling*, vol. 53, no. 3, pp. 623–637, 2013.

[48] N. Tuncbag, F. S. Salman, O. Keskin, and A. Gursoy, "Analysis and network representation of hotspots in protein interfaces using minimum cut trees," *Proteins: Structure, Function, and Bioinformatics*, vol. 78, no. 10, pp. 2283–2294, 2010.

[49] N. W. Lemons, B. Hu, and W. S. Hlavacek, "Hierarchical graphs for rule-based modeling of biochemical systems," *BMC bioinformatics*, vol. 12, no. 1, p. 45, 2011.

[50] T. K. Saha and M. A. Hasan, "FS^3: A sampling based method for top-*k* frequent subgraph mining," *Statistical Analysis and Data Mining*, vol. 8, no. 4, pp. 245–261, 2015.

[51] M. Al Hasan and M. J. Zaki, "Output space sampling for graph patterns," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 730–741, 2009.

[52] A. R. Rao, R. Jana, and S. Bandyopadhyay, "A markov chain monte carlo method for generating random (0, 1)-matrices with given marginals," *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 225–242, 1996.

[53] R. Milo, N. Kashtan, S. Itzkovitz, M. E. Newman, and U. Alon, "On the uniform generation of random graphs with prescribed degree sequences," *arXiv preprint cond-mat/0312028*, 2003.

[54] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 32, no. 5, pp. 922–923, 1976.

[55] M. W. Walker, L. Shao, and R. A. Volz, "Estimating 3-d location parameters using dual number quaternions," *CVGIP: image understanding*, vol. 54, no. 3, pp. 358–367, 1991.

[56] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic acids research*, vol. 28, no. 1, pp. 235–242, 2000.

1545-5963 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

[57] A. R. Katebi and R. L. Jernigan, “The critical role of the loops of triosephosphate isomerase for its oligomerization, dynamics, and functionality,” *Protein Science*, vol. 23, no. 2, pp. 213–228, 2014.

[58] E. Lolis, T. Alber, R. C. Davenport, D. Rose, F. C. Hartman, and G. A. Petsko, “Structure of yeast triosephosphate isomerase at 1.9-. ang. resolution,” *Biochemistry*, vol. 29, no. 28, pp. 6609–6618, 1990.

[59] G. M. Cooper, *The Cell - A Molecular Approach 2nd Edition*. Sunderland (MA): Sinauer Associates, 2000.

[60] C. T. Porter, G. J. Bartlett, and J. M. Thornton, “The catalytic site atlas: a resource of catalytic sites and residues identified in enzymes using structural data,” *Nucleic acids research*, vol. 32, no. suppl 1, pp. D129–D133, 2004.

[61] J. P. Nilmeier, D. A. Kirshner, S. E. Wong, and F. C. Lightstone, “Rapid catalytic template searching as an enzyme function prediction procedure,” *PloS one*, vol. 8, no. 5, p. e62535, 2013.

[62] N. Furnham, G. L. Holliday, T. A. de Beer, J. O. Jacobsen, W. R. Pearson, and J. M. Thornton, “The catalytic site atlas 2.0: cataloging catalytic sites and residues identified in enzymes,” *Nucleic acids research*, vol. 42, no. D1, pp. D485–D489, 2014.

[63] W. Dhifli, S. Aridhi, and E. M. Nguifo, “Mr-simlab: Scalable subgraph selection with label similarity for big data,” *Information Systems*, vol. 69, pp. 155 – 163, 2017.

Tanay Kumar Saha is a PhD candidate in Purdue University, West Lafayette. He has finished his Bachelor and Masters from Bangladesh University of Engineering and Technology (BUET) and Indiana University - Purdue University Indianapolis (IUPUI), respectively. His works are at the intersection of Networks Theory and Machine Learning. His research interest includes developing data mining and machine learning algorithms for novel applications in various domains, such as, text mining, biology and security. He had the opportunity to work with a number of industrial research labs including NEC Labs, Data Analytics Team at QCRI, and iControl ESI. He has authored 2 peer-reviewed journal articles, 6 conference papers and 2 workshop papers so far.

Ataur Katebi is a postdoctoral fellow in the Department of Veterinary Microbiology and Preventive Medicine at Iowa State University. Previously, he obtained his doctoral degree in Bioinformatics and Computational Biology from Iowa State University and was a visiting fellow at National Cancer Institute. His research involves understanding the architectures, dynamics, and functions of proteins as biological machines and their interactions in biological networks. He is also interested in the correlation between gene expression and DNA conformational changes. He employed computer algorithms and biophysical methods in his investigations. His research has been funded by both NSF grants and NIH fellowship.

Wajdi Dhifli is a postdoctoral research associate in machine learning and computational biology at the institute of Systems and Synthetic Biology (iSSB) at the University of Evry-Val-d’Essonne in Evry, France. He previously worked also as a postdoctoral research associate and lecturer at the University of Quebec at Montreal (UQAM) and he received a Ph.D in computer science from Blaise Pascal University (UBP), France. His research interests include data mining, machine learning, big data analytics and bioinformatics.

Mohammad Al Hasan received his PhD degree in computer science from Rensselaer Polytechnic Institute, NY, in 2009. He is an associate professor of computer science at Indiana University- Purdue University, Indianapolis (IUPUI). Before that, he was a senior research scientist at eBay Research Labs, San Jose, CA. His research interest focuses on developing novel algorithms in data mining, data management, information retrieval, machine learning, social network analysis, and bio-informatics. He has published more than 30 research articles in top-tier data mining conferences and journals. He has received various awards, including PAKDD conference best paper award in 2009, SIGKDD doctoral dissertation award in 2010, NSF CAREER award in 2012, and IUPUI School of Science Pre-tenure Research award in 2013.

1545-5963 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.