Coupling Graphs, Efficient Algorithms and B-Cell Epitope Prediction

Liang Zhao, Steven C.H. Hoi, Zhenhua Li, Limsoon Wong, Hung Nguyen, and Jinyan Li

Abstract—Coupling graphs are newly introduced in this paper to meet many application needs particularly in the field of bioinformatics. A coupling graph is a two-layer graph complex, in which each node from one layer of the graph complex has at least one connection with the nodes in the other layer, and vice versa. The coupling graph model is sufficiently powerful to capture strong and inherent associations between subgraph pairs in complicated applications. The focus of this paper is on mining algorithms of frequent coupling subgraphs and bioinformatics application. Although existing frequent subgraph mining algorithms are competent to identify frequent subgraphs from a graph database, they perform poorly on frequent coupling subgraph mining because they generate many irrelevant subgraphs. We propose a novel graph transformation technique to transform a coupling graph into a generic graph. Based on the transformed coupling graphs, existing graph mining methods are then utilized to discover frequent coupling subgraphs. We prove that the transformation is precise and complete and that the restoration is reversible. Experiments carried out on a database containing 10,511 coupling graphs show that our proposed algorithm reduces the mining time very much in comparison with the existing subgraph mining algorithms. Moreover, we demonstrate the usefulness of frequent coupling subgraphs by applying our algorithm to make accurate predictions of epitopes in antibody-antigen binding.

Index Terms—Coupling graph, epitope prediction, graph mining, graph transformation

1 Introduction

Graph representation and graph data analysis have been widely used in many bioinformatics studies. Protein-protein interaction (PPI) network is a well-known example; its nodes denote unique proteins and its edges represent physical contacts between the pairs of proteins [1]. Another example is genetic regulatory networks in which the nodes represent genes, and the edges stand for gene regulatory relations, such as a relation that gene A inhibits gene B, or a relation that gene B activates gene C [2].

More interesting graphs used in bioinformatics include those which contain two sets of nodes of different meanings. For example, a gene-phenotype association network contains two different sets of nodes. Nodes in one set represent genes, while nodes in the other set stand for phenotypes. The edges in such a network also have different meanings, and can be grouped into: (i) those relation edges within the genes only, (ii) those similarity edges within the phenotypes only, and (iii) the association edges between the genes and phenotypes [3]. An illustration of a gene-phenotype network is shown in Fig. 1a. It can be seen that the nodes in this network belong to two categories (gene and phenotype) and that the edges have different meanings (i.e., inter-gene interactions, inter-phenotype similarities, and gene-phenotype associations). This kind of two-layer graph complex is referred to as a *coupling graph* in this work. Each layer in a coupling graph is defined as a subgraph and every node in one layer has at least one edge connecting with a node in the other layer. A coupling graph is not necessarily a bipartite graph, as there usually exist many edges within each layer of a coupling graph. However, a coupling graph can be easily reduced to a bipartite graph by removing all of the edges in the same layer subgraph.

Many other bioinformatics problems also involve coupling graphs. For example, an antibody-antigen interaction complex [5] can form a coupling graph when the residues are represented by nodes, and the physical contacts between the residues are represented by edges. As shown in Fig. 1b, the interactions of some residues in the antibody-antigen complex (Protein Data Bank (PDB) entry 1TJG) forms a coupling graph, where the nodes are the contacting residues and the edges are the residue contacts. As another example, the expression regulation network of microRNAs and genes can be constructed as a coupling graph. One layer of this coupling graph represents the similarity network of the microRNAs' expression, while the other layer is a gene expression similarity network. The edges between these two networks are functional regulatory relationships [6], as shown in Fig. 1c.

Compared to generic bipartite graphs, the integrative notion of coupling graphs has advantages for deciphering biological associations, identifying structural motifs in protein complexes, predicting context-awareness binding sites of proteins, and constructing binding partners for an input

- L. Zhao is with the Department of Pediatrics, Baylor College of Medicine, 1100 Bates st, Houston, TX 77030. E-mail: s080011@e.ntu.edu.sg.
- S.C.H. Hoi and Z. Li are with the School of Computer Engineering, Nanyang Technological University, Singapore. E-mail: chhoi@ntu.edu.sg, lizh0021@e.ntu.edu.sg.
- L. Wong is with the School of Computing, National University of Singapore, 13 Computing Drive, Singapore 117417. E-mail: wongls@comp.nus.edu.sg.
- H. Nguyen is with the Center for Health Technologies, FEIT, University of Technology Sydney, 15 Briadway Road, Broadway, Sydney, NSW 2007, Australia. E-mail: hung.nguyen@uts.edu.au.
- J. Li is with the Advanced Analytics Institute, University of Technology Sydney, PO Box 123, Broadway, NSW 2007, Australia. E-mail: jinyan.li@uts.edu.au.

Manuscript received 4 Oct. 2013; accepted 15 Oct. 2013; date of publication 3 Nov. 2013; date of current version 7 May 2014.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TCBB.2013.136

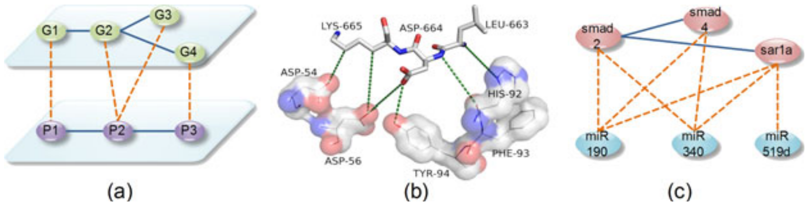


Fig. 1. Examples of coupling graphs in bioinformatics. (a) is a diagram of gene-phenotype association network, in which genes are represented by light green nodes and phenotypes are depicted by light purple nodes. The solid lines are interactions within the genes or phenotypes, while the dash lines are associations between the genes and phenotypes. (b) shows partial interactions between antigen gp41 and antibody 2F5. The interactions between this antibody and antigen are represented by dash lines. (c) illustrates the role of microRNAs in regulating TGFβ singaling pathway. The regulations between the microRNAs and their targets are represented as dash lines [4].

protein [7], [8]. Taking a paratope-epitope interacting complex as example, the coupling graph representation of this complex has several advantages. First, the two special subgraphs (the two layers) in this coupling graph can preserve topological information of paratope residues and epitope residues. Second, the edges between the two subgraphs of this coupling graph capture the contact details between the nodes of the two subgraphs. Note that the contacts between subgraphs have different meaning comparing with within-contacts in each subgraph. In this example, the between-contacts are mainly noncovalent bonds, while the within-contacts are mostly covalent bonds. Therefore, using coupling graph to distinguish them is informative and helpful. Third, the unification of between-contacts and within-contacts not only keeps the topology of the subgraph and inter-contacts between the subgraphs, but also uncovers the systematical structures of the contacts. For instance, a coupling graph can reveal the complementary core interaction between the epitope and the paratope in PDB complex 1AR1, where the epitope has a hydrophobic core surrounded by hydrophilic rim while the paratope has a hydrophilic core encompassed by neutral residues, as discovered in [9]. However, if bipartite graphs are used for the data representation, many important neighborhood and topological information as well as biological properties in the two subgraphs of coupling graphs may get lost.

The focus of this work is on efficient mining of coupling subgraphs that occur frequently in coupling graph databases and its bioinformatics application. There exist efficient algorithms for mining frequent subgraphs from a generic graph database, including AGM [10], FSG [11], MoFa [12], gSpan [13], FFSM [14] and Gaston [15]. However, these algorithms cannot be directly used to mine frequent coupling subgraphs from a coupling graph database. If a coupling graph is treated as a generic graph, difficulties will arise when the aforementioned subgraph miners are used to find frequent coupling subgraphs. On the one hand, a frequent subgraph generated by these algorithms may contain nodes from only one layer of a coupling graph or include irrelevant subgraphs. For example, the frequent subgraph "1—3" in Fig. 2 is not a frequent coupling subgraph but it is a frequent subgraph, and the frequent subgraph "2—1—3" contains a subgraph "1—3" which is not a coupling graph. On the other hand, a coupling graph $A = (G_1^A, G_2^A, E^A)$ is isomorphic to a coupling graph $B = (G_1^B, G_2^B, E^B)$ if they are regarded as generic graphs, but their corresponding constituent graphs may not be isomorphic, i.e., G_1^A may not be isomorphic to G_1^B and G_2^A is not necessary to be isomorphic to G_2^B.

We propose new algorithms and make the following contributions to the efficient mining of frequent coupling subgraphs from coupling graph databases. We define and formulate the new concepts related to coupling graphs. We design an efficient algorithm to mine frequent coupling subgraphs from a coupling graph database by novel graph transformation and graph restoration techniques. We prove that the transformation and restoration are reversible. We also evaluate the efficiency of our algorithm by comparing it with the performance of generic subgraph mining algorithms on large-scale real data.

To show the usefulness of frequent coupling subgraphs in real bioinformatics problems, we apply our algorithm to predict antibody-specific B-cell epitopes. The representation of epitope-paratope interaction by the use of coupling graphs not only implements the context-awareness theories [16], it also builds a sound foundation to achieve better performance on epitope prediction according to our experimental results shown later.

2 Definition and Related Works

Coupling graph is a newly formulated concept, which is convenient and comprehensive to capture information of two related graphs. Coupling graph is related to, but different from bi-clique, quasi bi-clique and generic graph.

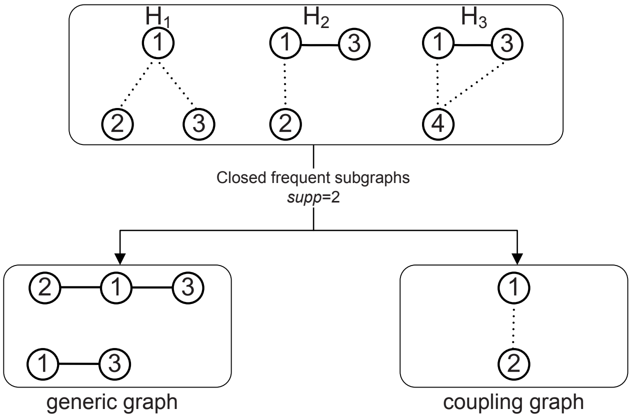


Fig. 2. Some frequent coupling subgraphs and frequent subgraphs of a graph data set. A solid line represents an edge within a layer subgraph, while a dash line represents an edge between the two layer subgraphs of a coupling graph.

2.1 Definition of Coupling Graph

A graph G is an ordered pair denoted by $G = (V, E)$, where V is a set of nodes and $E \subseteq V \times V$ is a set of edges. An edge e in E is denoted by $e = (v_i, v_j)$.

Definition 1. *A* coupling graph H *is a graph complex denoted by* $H = ((V^1, V^2), (E^1, E^2, E^{12}))$, *where* $E^{12} \subseteq V^1 \times V^2$, V^1 *and* E^1 *forms a subgraph* G^1, V^2 *and* E^2 *forms a subgraph* G^2, *and the two subgraphs satisfy that* $\forall v_i^1 \in V^1$, $\exists v_j^2 \in V^2$ *such that* $(v_i^1, v_j^2) \in E^{12}$, *and* $\forall v_j^2 \in V^2$, $\exists v_i^1 \in V^1$ *such that* $(v_i^1, v_j^2) \in E^{12}$.

We note that every node v^1 in G^1 is required by definition to connect to at least one node v^2 in G^2 for a coupling graph H. This constraint guarantees that all the nodes are involved in the interaction between the two subgraphs of a coupling graph. This modeling constraint is motivated by some real application needs. For example, to characterize antibody-antigen interactions, only paratope residues in an antibody and epitope residues in an antigen are needed, and the rest can be ignored.

According to the definition above, a coupling graph may contains several connected components, which is defined as:

Definition 2. *A* connected coupling graph $H_c = ((V_c^1, V_c^2), (E_c^1, E_c^2, E_c^{12}))$ *is a coupling graph, such that* $\forall u \in V_c^1, \exists v \in V_c^2$, *there is a path connecting* u *and* v, *and vice versa.*

A *coupling subgraph* is a coupling graph which is a subgraph of a coupling graph.

Definition 3. *A coupling graph* H *is frequent in a coupling graph database* $\mathbb{H}$ *if* H *is a coupling subgraph in not less than* δ *number of coupling graphs in* $\mathbb{H}$.

2.2 Relation to Bi-Clique, Quasi Bi-Clique and Generic Graph

Coupling graph has relation with bi-clique, quasi bi-clique and generic graph, but essentially it is different from various existing forms of graph.

A *bi-clique* is an *undirected graph* $G = (V, E)$, such that $V = (V_1, V_2)$, $V_1 \cap V_2 = \emptyset$, $V_1 \cup V_2 = V$, $\forall u \in V_1$ and $\forall v \in V_2$, $(u, v) \in E$ and $|V_1| \times |V_2| = |E|$. It is clear, from the two definitions, that a coupling graph differs from a bi-clique in two major points: (i) the edges between the two sets of the nodes in a bi-clique are complete, while no completeness restriction on the edges between two subgraphs of a coupling graph and; (ii) no edges within each set of the nodes in a bi-clique, but each subgraph of a coupling graph can have edges. Although differences exist, the two types of graph are related—both of them are two-layered graphs.

Regarding the completeness between graph connections, a coupling graph is more closer to a quasi bi-clique than a bi-clique. In a quasi bi-clique, the degree of a node $u \in V_1$, denoted as *deg(u)*, satisfies $\delta \leq deg(u) \leq |V_2|$, and the same constraint applies to any node of V_2; while for a coupling graph, the value δ can be considered as degenerated to 1 (excluding the degree formed from the edges within the same layer of a coupling graph).

A coupling graph is also quite different from a generic graph, in which all the nodes are considered within the same domain and thus no difference between edges as well.

2.3 Frequent Subgraph Mining

Due to the essential differences between coupling graphs and generic graphs, the frequent coupling subgraph mining is quite different from generic subgraph mining. However, several graph mining algorithms are closely related, and some of their ideas are useful for developing coupling graph mining algorithms.

AGM [10] is a representative Apriori-based approach for mining frequent subgraphs, which can identify both connected and unconnected graphs. It employs an adjacency matrix to represent graphs, and breadth-first search (BFS) to discover frequent graph patterns. Other Apriori-based algorithms have also been proposed for mining frequent subgraphs, including FSG [11], gFSG [17] and DPMine [18]. Although the same strategy is adopted by these algorithms, different graph representation and repeat count ideas are used. The BFS search strategy performs strong pruning during subgraph expansion; however, it consumes huge volume of memory. Therefore, the depth-first search (DFS) method, which takes less memory, is developed. MoFa [12] uses a fragment-local numbering scheme to expand subgraphs. Besides, structural pruning and molecular knowledge are used to reduce support calculation, which thus dedicates to chemical molecules exploration. Another well-established algorithm for frequent subgraph mining based on pattern growth is gSpan [13]. gSpan uses the minimum DFS code to represent each graph and only expands a frequent subgraph with minimum DFS code. The canonical adjacency matrix (CAM) graph representation is used by FFSM [14] to mine frequent subgraphs. This algorithm uses an embedding list to record the discovered frequent patterns in CAM format, which avoids graph isomorphism testing. Gaston [15] incorporates a progressive model, from path, tree to graph, to reduce the mining time. Graph isomorphism testing is only performed on subgraphs instead of trees and paths. Various graph expansion and support counting methods have been proposed to mine frequent subgraphs; which, however, cannot be directly used to mine frequent coupling subgraphs as the edges in a coupling graph have different meanings.

2.4 Correlated Graph Pattern Mining

Besides frequent subgraph mining, attempts have been made on correlated graph pattern mining. The *correlated graph search* is formulated by Ke et al. [19], in which Pearson's correlation coefficient is used to measure the correlation between graphs. Later on, the *frequent correlated subgraph pairs* mining algorithm is established by Ke et al. [20], in which a theoretical bound on the minimum correlation is determined to discover correlated subgraph pairs. HSG [21] is proposed to discover frequent *hyperclique patterns* in graph databases, where a hyperclique pattern is defined as a set of items with high affinity measured by h-confidence [22]. Another related work is *pairs of graph pattern* mining, which discovers rules to classify graph pairs by estimating the tight upper bound on a statistical metric. An attempt has also been made on frequent subgraph-subsequence pair mining [23]. However, these problems are different from coupling graph mining—the correlated graphs are separate in the former, while they are tightly connected in the latter.

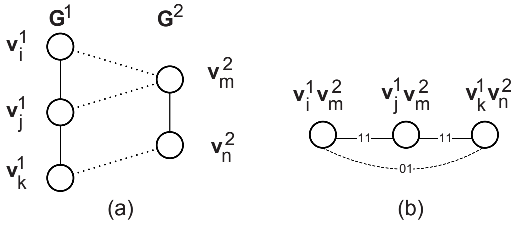


Fig. 3. Coupling graph transformation. (a) is the original coupling graph, where solid lines represent edges within G^1/G^2 and dash lines represent edges between G^1 and G^2. (b) is the transformed generic graph.

3 Algorithms for Mining Frequent Coupling Subgraphs from a Graph Database

We take the following three steps to mine frequent coupling subgraphs: (i) transform a coupling graph into a generic graph; (ii) mine frequent subgraphs from the transformed generic graphs by using an existing graph mining method; and (iii) restore the coupling graphs from the set of transformed frequent subgraphs. The detailed description for each step is presented in the following sections.

3.1 Transformation of Coupling Graphs into Generic Graphs

For a coupling graph $H = ((V^1, V^2), (E^1, E^2, E^{12}))$, we transform it into a generic graph $H' = (V', E')$ in two steps: node construction and edge construction.

- *Node transformation*. For each edge $e_i^{12} = (v_i^1, v_i^2)$ in E^{12}, we use $v_i^1 v_i^2$ as a label to form a new node of V';
- *Edge transformation*. Two nodes $v_i' = v_i^1 v_i^2$ and $v_j' = v_j^1 v_j^2$ of V' are connected by an edge with a label l defined as:

$$l = \begin{cases} 11, & \text{iff } v_i^1 = v_j^1 \ \& \ (v_i^2, v_j^2) \in E^2, \qquad (a) \\ 11, & \text{iff } (v_i^1, v_j^1) \in E^1 \ \& \ v_i^2 = v_j^2, \qquad (b) \\ 11, & \text{iff } (v_i^1, v_j^1) \in E^1 \ \& \ (v_i^2, v_j^2) \in E^2, \qquad (c) \\ 01, & \text{iff } (v_i^1, v_j^1) \notin E^1 \ \& \ (v_i^2, v_j^2) \in E^2, \qquad (d) \\ 10, & \text{iff } (v_i^1, v_j^1) \in E^1 \ \& \ (v_i^2, v_j^2) \notin E^2. \qquad (e) \end{cases}$$

For the label "l" of an edge in E', the first code "1" means that the edge (v_i^1, v_j^1) exists in E^1, and the first code "0" represents that there is no edge between v_i^1 and v_j^1, and similarly for the meaning of the two second codes. Condition (a) represents that one node in G^1 connects with two different nodes v_i^2 and v_j^2 in G^2 between which there is an edge, while condition (b) is a similar situation of condition (a) differing in that which layer the node(s) belong to. In condition (c), v_i^1 and v_j^1 in G^1 have an edge, and the same situation for v_i^2 and v_j^2 in G^2. Condition (d) and (e) are situations where only one edge is present. If none of the two pairs of nodes is connected in G^1 or G^2, then there is no edge between the newly constructed nodes.

Fig. 3 shows an example using the above definition to transform a coupling graph (Fig. 3a) into a generic graph (Fig. 3b). For ease of presentation, we use superscript 1, 2, or 12 to represent coupling graphs before our transformation and use those with superscript $'$ to represent generic graphs after the transformation.

Theorem 1. *Transformation from H to H' is precise and complete. Preciseness means that all the edges and nodes in H' correspond to some nodes and/or edges in H. Completeness means that all the edges and nodes information in H is contained in H' without information loss.*

The correctness of the theorem is proofed in the following section, where restoration is presented.

3.2 Restoration of Coupling Graphs from Transformed Generic Graphs

For a transformed generic graph $H' = (V', E')$, we take the following steps to restore its coupling graph $H = ((V^1, V^2), (E^1, E^2, E^{12}))$:

- *Node restoration*. For each node $v' = v^1 v^2$ of V', we add v^1 to V_1 and v^2 to V_2;
- *Edge restoration*. For each node $v' = v^1 v^2$ of V', we add (v^1, v^2) to E^{12}; for each edge $e' = (v_i', v_j', l)$ of E', we add (v_i^1, v_j^1) to E^1 if l is "10" or "11" and add (v_i^2, v_j^2) to E^2 if l is "01" or "11", where $v_i' = v_i^1 v_i^2$, $v_j' = v_j^1 v_j^2$ and $l \in \{01, 10, 11\}$.

Theorem 2. *Transformation from H to H' is reversible, i.e., all the nodes and edges of H can be recovered from H' without introducing additional nodes or edges.*

Proof. Preciseness is obvious by construction of generic graph from coupling graph. As to completeness, we show all five components $(V^1, V^2, E^{12}, E^1, E^2)$ of H are captured in H'. Wrt V^1, by definition of coupling graphs, for each v^1 in V^1, there is a v^2 in V^2 such that the edge (v^1, v^2) is in E^{12}. By the node transformation step, there is a node $v^1 v^2$ in V', thus capturing v^1. A similar argument shows that every node in V^2 is also captured by a node in V'. Wrt E^{12}, for each edge (v^1, v^2) in E^{12}, by the node transformation step, it is captured by the node $v^1 v^2$ in V'. Wrt E^1, for an edge (v_i^1, v_j^1) in E^1, by definition of coupling graphs, there are nodes v_i^2 and v_j^2 in E^2, not necessarily distinct, such that (v_i^1, v_i^2) and (v_j^1, v_j^2) are in E^{12}. By the edge transformation step, there is an edge $(v_i^1 v_i^2, v_j^1 v_j^2)$ in E' with label "10" or "11". This implies (v_i^1, v_j^1) is captured. Wrt E^2, a similar argument shows that every edge in E^2 is also captured by an edge in E'. Therefore, the transformation from H to H' is also complete. Based on the procedure of constructing transformed graph, it is obvious that the transformation from H to H' is reversible. □

3.3 Frequent Coupling Subgraph Mining

For a coupling graph database $\mathbb{H}$, we first transform each coupling graph into a generic graph, then we use subgraph mining algorithms to obtain frequent subgraphs from the transformed graph database, finally the transformed frequent subgraphs are restored to obtain the frequent coupling subgraphs. The pseudocode for mining frequent coupling subgraphs is shown in Algorithm 1.

Algorithm 1 Mining frequent coupling subgraphs

```
Procedure mineCouplingGraph($\mathbb{G}$, $\delta$)
  $\mathbb{G}^{'}$:=$\emptyset$, $\mathbb{H}$:=$\emptyset$, $\mathbb{H}^{'}$:=$\emptyset$
  for $G$ := $((V^1, V^2), (E^1, E^2, E^{12}))$ in $\mathbb{G}$ do
      $\mathbb{G}^{'} \leftarrow$ transformGraph($G$)
  end for
  $\mathbb{H}^{'}$ := frequentSubgraphMining($\mathbb{G}^{'}$, $\delta$)
  for $H^{'}$ in $\mathbb{H}^{'}$ do
      $\mathbb{H} \leftarrow$ restoreGraph($H^{'}$)
  end for
  return $\mathbb{H}$
```

```
Function transformGraph($H$)
  $V^{'}$:=$\emptyset$, $E^{'}$:=$\emptyset$
  for $e^{12}$=$(v^1, v^2)$ in $E^{12}$ do
      $V^{'} \leftarrow v^1 v^2$
  end for
  for $e^1$=$(v_i^1, v_j^1)$ in $E^1$ do
      for $e^2$=$(v_i^2, v_j^2)$ in $E^2$ do
          if $(v_i^1, v_i^2) \in E^{12}$ && $(v_j^1, v_j^2) \in E^{12}$ then
              $E^{'} \leftarrow (v_i^1 v_i^2, v_j^1 v_j^2, 11)$
          else if $(v_i^1, v_i^2) \in E^{12}$ && $(v_j^1, v_j^2) \notin E^{12}$ then
              $E^{'} \leftarrow (v_i^1 v_i^2, v_j^1 v_j^2, 10)$
          else if $(v_i^1, v_i^2) \notin E^{12}$ && $(v_j^1, v_j^2) \in E^{12}$ then
              $E^{'} \leftarrow (v_i^1 v_i^2, v_j^1 v_j^2, 01)$
          end if
      end for
  end for
  return $H^{'}$=$(V^{'}, E^{'})$
```

```
Function restoreGraph($H^{'}$)
  $V^1$:=$\emptyset$, $V^2$:=$\emptyset$, $E^1$:=$\emptyset$, $E^2$:=$\emptyset$, $E^{12}$:=$\emptyset$
  for $v^{'}$=$v^1 v^2$ in $V^{'}$ do
      $V^1 \leftarrow v^1$, $V^2 \leftarrow v^2$, $E^{12} \leftarrow (v^1, v^2)$
  end for
  for $e^{'}$=$(v_i^{'}, v_j^{'}, l)$ in $E^{'}$ do
      if l == "11" then
          $E^1 \leftarrow (v_i^1, v_j^1)$, $E^2 \leftarrow (v_i^2 v_j^2)$
      else if l == "10" then
          $E^1 \leftarrow (v_i^1, v_j^1)$
      else if l == "01" then
          $E^2 \leftarrow (v_i^2 v_j^2)$
      end if
  end for
  return $H$=$((V^1, V^2), (E^1, E^2, E^{12}))$
```

The time complexity of subgraph mining is in proportion to the product between the total number of subgraphs and the complexity of graph isomorphism testing. The main part of the time cost of subgraph mining is for subgraph isomorphism testing, which is NP-complete [24]. The proposed algorithm of coupling graph mining significantly reduces the time cost and memory consumption by using graph transformation which avoids the generation of many irrelevant subgraphs. The time complexity of graph transformation for a data set with n coupling graphs is in proportion to $\sum_i^n N_1^i \cdot N_2^i$, where N_1^i is the number of edges of graph G_1^i and N_2^i is the number of edges of graph G_2^i.

3.4 Transformation and Restoration with Duplicate Node Labels

In the above study, we assume that all the node labels in G^1 or in G^2 of a coupling graph H' are unique but allowing some identical labels between some nodes in G^1 and G^2. In practice labels usually have duplicates in V^1 or in V^2. For example, an interface of protein-protein interacting complex is composed of residues which have twenty types only in nature, hence duplicate residues usually exist in interfaces.

Duplicate labels do not affect coupling graph transformation and transformed generic graph mining, but it does impede graph restoration because whether a new node should be created or not is unknown when a node with a duplicate label is brought in. We take some additional steps to solve coupling graph mining with duplicate labels: (i) map each node in V^1 or in V^2 to a unique label and transform the relabeled coupling graph into generic graph; (ii) mine frequent subgraphs from the transformed generic graph with new labels; (iii) restore each transformed frequent subgraph into a coupling graph and recover the original labels according to the mapping table.

4 Protein Complex Coupling Graph Database and Efficiency Results

In this section, we report the performance of our algorithm. We also report the number of irrelevant subgraphs generated by existing subgraph mining algorithms to understand why the high efficiency of our algorithm is achieved by the graph transformation approach. The coupling graphs we used in the evaluation are real data compiled from the Protein Data Bank [25]. The purpose is to comprehend to what extent the new algorithm is better than the existing algorithms when dealing with real-world problems.

4.1 Coupling Graph Database Compilation

As mentioned in Section 1, when one protein interacts with another protein, the interacting part of the two proteins can be represented as a coupling graph by using nodes to represent the contacting residues and using edges to represent the close contacting distance. Protein-protein interaction complexes are stored at the widely used PDB database where the three-dimensional co-ordinates information of atoms in every residues is available.

Protein-protein interaction complexes that satisfy the following criteria are retrieved from PDB: (i) the macromolecular type is protein only, without DNA and RNA; (ii) the number of protein chains is larger than two; (iii) the length of each protein (chain) is larger than or equal to 30; and (iv) the X-ray resolution of one complex is less than 3 Å. As a result, 29,418 PDB entries with 129,305 protein-protein interaction pairs are obtained. With the removal of those similar chains under BLAST [26] maximum pair-wise sequence similarity threshold of 90 percent, 9,781 PDB entries containing 10,511 protein-protein interaction complexes are left and used for our algorithm efficiency study.

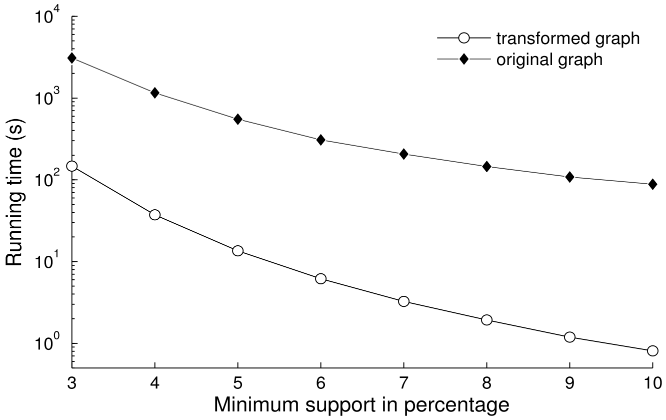


Fig. 4. Running time comparison of mining frequent coupling subgraphs from the original coupling graphs and from the transformed coupling graphs.

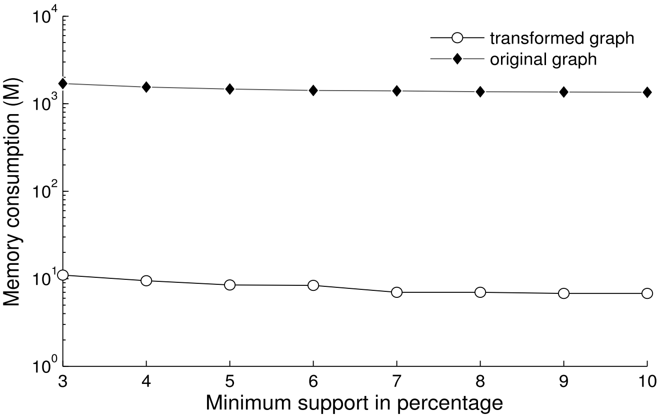


Fig. 5. Memory consumption comparison of mining frequent coupling subgraphs from the original coupling graphs and from the transformed coupling graphs.

The coupling graph database for the 10,511 protein-protein interaction complexes are built in two steps: (i) determine interfacial residues (i.e., the nodes of a coupling graph) and connections between the two interfacial surfaces (edges between the two layer subgraphs of a coupling graph) from a PPI complex by using Euclidian distance of 2.75 Å plus residues' radii [27]; (ii) build connections of residues within each interfacial surface (i.e., edges within each of the two subgraphs of a coupling graph) by using qhull [28]. The average number of nodes and the average number of edges for the coupling graphs in our graph database are 65.3 ± 43.2 and 205.9 ± 155.8, respectively.

Our experiments were carried out on a platform with Ubuntu 11.04 operating system, 4G physical memory and eight cores with each of 2.67 GHz.

4.2 Efficiency Results

Frequent subgraphs of the coupling graph database without graph transformation are mined using gSpan [13] which is implemented in the ParMol package [29], while frequent coupling subgraphs with graph transformation are mined by using LCM [30].

LCM is feasible to mine frequent coupling subgraphs because of the following reasons: (i) the transformation makes the label sparser, i.e., theoretically from n to n^4 (each item is a transformed node pair connected by an edge); (ii) duplicate items are allowed due to the relabelling of repeat labels and; (iii) post-comparison on restoration with duplicate labels guarantees that the repeat nodes are properly handled. In the extreme case, i.e., all the nodes have the same label, although very unlikely to happen, however LCM is not a good choice for our purpose. But considering the real cases, it is still competent to handle.

To mine frequent coupling subgraph partially by using LCM, we take a transactional database to represent the coupling graph database. Each transaction represents a transformed coupling graph and the items in this transaction are the entire set of nodes and edges of the transformed graph (duplicate items are preserved and are relabeled in order). Each frequent item set corresponds to a transformed coupling graph, which can be restored to its equivalent original coupling graph form. The equivalence between a coupling graph and its transformed generic graph has been proved in the above section.

Fig. 4 shows the running time of mining frequent subgraphs from the database with 10,511 coupling graphs on the original graphs and also on the transformed graphs. It is clear that mining coupling subgraphs from the transformed graphs is remarkably faster than mining subgraphs from original coupling graphs. For example, mining frequent subgraphs from the original coupling graph database costs 3,084 seconds at the minimum support of 3 percent, while the cost is only 147 seconds on the transformed graphs with the same support level. In addition, Fig. 5 also indicates that using graph transformation consumes significantly less memory.

4.3 Irrelevant Frequent Subgraphs Generated by gSpan

We note that the frequent subgraphs mined from the coupling graph database by using gSpan [13] covers a large number of frequent non-coupling subgraphs. For instance as shown in Fig. 2, the frequent subgraphs generated by gSpan with support of 2 are "1", "2", "3", "1—2", "1—3", "2—1—3"; however, only "1—2" is frequent coupling subgraphs. Therefore, to eliminate these irrelevant frequent subgraphs still takes plenty of time, especially when an extremely huge number of frequent subgraphs are produced. In contrast, every frequent subgraph generated from the transformed graphs is an equivalent of a coupling subgraph thus, no such tremendous cost is needed.

Fig. 6 shows the number of connected frequent subgraphs generated from the coupling graph database as well as from its transformed graph database. The average number of frequent subgraphs generated by gSpan is about eight times the number of connected frequent subgraphs produced from the transformed graph database. Therefore, about 88 percent of the frequent subgraphs generated by gSpan are irrelevant frequent subgraphs, not to say the removal of irrelevant frequent subgraphs is a very heavy task, especially when the minimum support is low.

4.4 Statistics on the Frequent Coupling Subgraphs

A coupling graph can be connected or disconnected. For example, the coupling graphs shown in Figs. 7a and 7c are

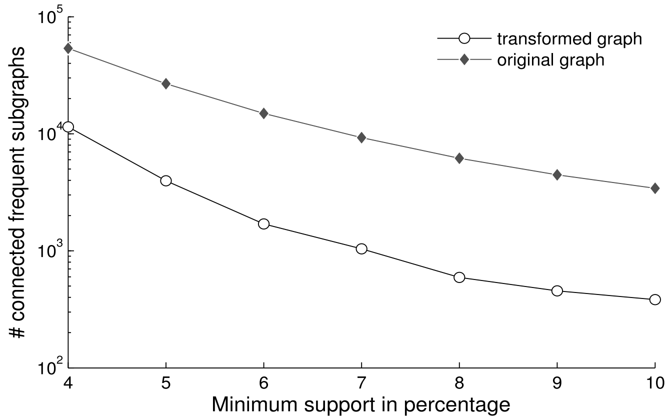


Fig. 6. Numbers of connected frequent subgraphs generated from the original graphs and from the transformed graphs.

connected coupling graphs, while the coupling graph shown in Fig. 7b is disconnected. The number of frequent coupling subgraphs of a coupling graph database can be extremely large, partially because some frequent connected coupling subgraphs can be combined to form new and frequent coupling subgraphs.

Table 1 shows the total number of frequent coupling subgraphs and frequent connected coupling subgraphs with respect to different minimum support from our data set containing 10,511 coupling graphs. It can be seen that when the support level is set as minimum 10 percent, there are still hundreds of connected frequent coupling subgraphs in our graph database. It implies that there are many regular coupling graph patterns in the protein-protein interactions.

5 Application: Pattern Discovery and Epitope Prediction in Antibody-Antigen Complexes

Frequent coupling subgraphs within protein-protein complexes can reveal important patterns shared by multiple complexes. These patterns have potential to discover contact residues or to construct binding partners with the property of “coupling”. In this section, we show an application of using coupling graphs for detecting significant patterns shared by antibody-antigen interacting complexes to identify antibody-specific B-cell epitopes.

5.1 Frequent Coupling Subgraph Patterns in Antibody-Antigen Complexes

We collected 156 antibody-antigen structural complexes from the PDB with antigen pair-wise sequence similarity less than 0.5 and the number of mutated antibody residues larger than 30. By using the coupling graph mining algorithm described in this study, we obtained 2,472 frequent coupling subgraphs from the 156 antibody-antigen complexes with the minimum support of 5 percent. Fig. 7 shows three examples of significant structural patterns that are common in antibody-antigen complexes. Among these examples, only Figs. 7a and 7c can be found by the existing subgraph mining algorithm, while Fig. 7b cannot be identified by them, but it can be found by our algorithm.

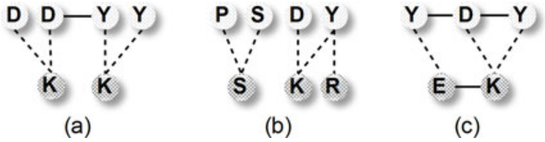


Fig. 7. Examples of frequent coupling patterns shared by antibody-antigen complexes.

TABLE 1
The Numbers of Frequent Coupling Subgraphs and Frequent Connected Coupling Subgraphs in the Database

min supp	# frequent coupling subgraphs	# frequent **connected** coupling subgraphs
4%	6,545,268	11,445
5%	1,379,285	3,966
6%	410,476	1,697
7%	153,045	1,040
8%	66,991	593
9%	32,357	455
10%	17,060	383

One of our findings from our experiments in coupling subgraph mining is that the residue Tyrosine (Y) in the antibodies is predominantly preferred in partnership with a hydrophilic residue to perform antigen binding. However, in the antigens the favored residues for antibody binding are charged residues (both positively charged and negatively charged), especially residues Arginine (R), Lysine (K), Aspartate (D) and Glutamate (E). Although the preferences of residue contacts within antibodies or within antigens have been explored elsewhere [8], none of them can be used to discover structural patterns between antibodies and antigens.

5.2 Epitope Prediction Using Frequent Coupling Graphs in Antibody-Antigen Complexes

As mentioned in Section 1, a protein antigen is a string of residues in the primary representation of proteins. An *epitope* of an antigen is a subset of residues of this antigen which physically contact each other tightly at the surface of the antigen and which is the binding area for an antibody in interaction. Similarly, the *paratope* site of an antibody is a subset of residues of this antibody which physically contact each other tightly at the surface of the antibody and which is the area binding to an epitope of an antigen. An interaction between an epitope and a paratope can be represented by a coupling graph when the residues are denoted by nodes and the physical contacts are denoted by edges for the pairs of residues in the antigen or in the antibody or in the both.

For a new antigen, its epitopes are usually unknown. Thus, epitope prediction is an important research for many applications in bioinformatics [31]. However, existing methods for epitope prediction overlook the principle of context-awareness in antibody-antigen interactions, and thus may not reflect biological reality [16], [32]. Therefore, we built a model incorporating frequent coupling subgraphs within antibody-antigen complexes to predict antibody-specific epitopes. The main idea is using frequent coupling subgraphs of antibody-antigen complexes from a training data set to identify the seeds of antibody-specific epitope residues of the testing data set, and then the true epitope residues are completely determined by some statistical measures.

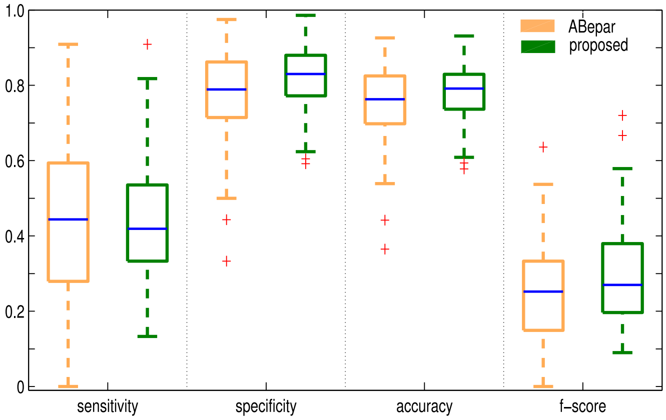


Fig. 8. Performance comparison between the proposed model, coupling graph based, and ABepar, association based, on antibody-specific B-cell epitope prediction.

Experimental results conducted on the data set of [16], which is the only existing data set for antibody-specific epitope prediction, show that our coupling graph-based model is much better than the association-based model [16] on epitope prediction. Fig. 8 shows the performances comparison between the coupling graph-based and the two-dimensional association-based methods for antibody-specific epitope prediction. The t-test p-values between the two models on averaged sensitivity, accuracy and f-score are 3.0e-3, 4.5e-3 and 7.8e-4, respectively. These significant p-values suggest that our method is indeed more accurate on epitope prediction than the association-based model.

As an example, the antigen lysozyme C with PDB entry 1P2C, as shown in Fig. 9, contains 129 residues in which 16 are epitope residues and 113 are non-epitope residues. The coupling graph model can successfully identify 11 epitope residues while only introducing 10 non-epitope residues; however, the association model includes 35 non-epitope residues although 12 epitope residues are correctly predicted. The prediction accuracy of the coupling graph-based method and association-based method on this antigen are 0.884 and 0.698, respectively. Frequent connected coupling subgraphs which are used to identify these epitope residues are shown in Fig. 10. Interestingly, the seed epitope residues are mainly introduced by the frequent coupling subgraphs with paratope residues D and Y.

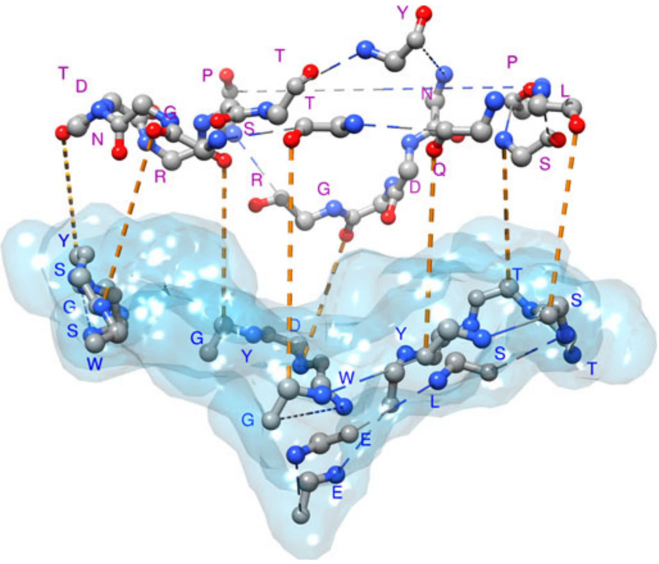


Fig. 9. An antibody-antigen interacting coupling graph extracted from the PDB entry 1P2C, where the paratope and epitope residues are shown. The epitope residues of the antigen are rendered as stick, while paratope residues of the antibody are represented by surface. The inter-edges between paratope and epitope are represented by dash orange lines.

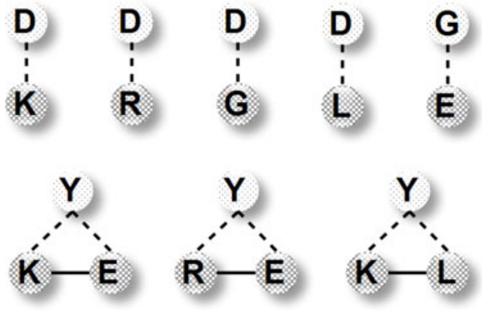


Fig. 10. Frequent connected coupling subgraphs which are used for identifying antibody-specific epitope residues of the antigen in PDB entry 1P2C.

6 Conclusion

Coupling graph is a new and very useful graphical model for representing intrinsic associations between pairs of subgraphs in a complex. In bioinformatics, coupling graphs can be used to reveal the structural interactions of protein-protein interacting complexes, gene-phenotype association networks, microRNA-gene expression regulatory networks, and so on. The frequent coupling subgraphs of these coupling graph databases play an important role in discovering the essential patterns hidden in the coupling graph databases. However, mining the frequent coupling subgraphs from a coupling graph database is very challenging, as existing subgraph mining algorithms perform poorly on coupling subgraph mining. The huge number of irrelevant subgraphs generated by the existing algorithm is the big hurdle to the efficiency. To overcome this obstacle, we have introduced a new algorithm by using a novel graph transformation and restoration technique. In this work, a coupling graph is transformed into a generic graph, and then subgraph mining is conducted on the transformed coupling graphs. We have proved that the transformation and restoration are equivalent. Experimental results carried out on a data set containing 10,511 coupling graphs have demonstrated that the proposed algorithm not only shortens the mining time, but also reduces the memory usage. The usefulness of frequent coupling subgraphs has also been demonstrated on identifying antibody-specific B-cell epitopes.

Acknowledgments

This work was partially supported by Nanyang Technological University Tier-1 Grant RG66/07.

REFERENCES

[1] M. Pellegrini, D. Haynor, and J.M. Johnson, “Protein Interaction Networks,” *Expert Rev. Proteomics*, vol. 1, no. 2, pp. 239-249, 2004.

[2] E. Davidson and M. Levin, “Gene Regulatory Networks,” *Proc. Nat'l Academy of Sciences USA*, vol. 102, no. 14, p. 4935, 2005.

[3] J.O. Korbel, T. Doerks, L.J. Jensen, C. Perez-Iratxeta, S. Kaczanowski, S.D. Hooper, M.A. Andrade, and P. Bork, “Systematic Association of Genes to Phenotypes by Genome and Literature Mining,” *PLoS Biology*, vol. 3, no. 5, p. e134, Apr. 2005.

[4] V.A. Gennarino, G.D'Angelo, G. Dharmalingam, S. Fernandez, G. Russolillo, R. Sanges, M. Mutarelli, V. Belcastro, A. Ballabio, P. Verde, M. Sardiello, and S. Banfi, “Identification of microRNA-Regulated Gene Networks by Expression Analysis of Target Genes,” *Genome Research*, vol. 22, no. 6, pp. 1163-1172, 2012.

[5] D.R. Davies and E.A. Padlan, “Antibody-Antigen Complexes,” *Ann. Rev. Biochemistry*, vol. 59, pp. 439-473, 1990.

[6] R.J. Jackson and N. Standart, “How Do microRNAs Regulate Gene Expression?” *Science STKE*, vol. 2007, no. 367, p. re1, 2007.

[7] L. He, F. Vandin, G. Pandurangan, and C. Bailey-Kellogg, “BALLAST: A Ball-Based Algorithm for Structural Motifs,” *Proc. 16th Ann. Int'l Conf. Research in Computational Molecular Biology (RECOMB)*, pp. 79-93, 2012.

[8] L. Zhao and J. Li, “Mining for the Antibody-Antigen Interacting Associations that Predict the B Cell Epitopes,” *BMC Structural Biology*, vol. 10, no. Suppl 1, article S6, 2010.

[9] L. Zhao, L. Wong, L. Lu, S.C.H. Hoi, and J. Li, “B-cell Epitope Prediction through a Graph Model,” *BMC Bioinformatics*, vol. 13, no. Suppl 17, article S20, 2012.

[10] A. Inokuchi, T. Washio, and H. Motoda, “An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data,” *Proc. Fourth European Conf. Principles of Data Mining and Knowledge Discovery (PKDD)*, pp. 13-23, 2000.

[11] M. Kuramochi and G. Karypis, “Frequent Subgraph Discovery,” *Proc. IEEE Int'l Conf. Data Mining (ICDM)*, pp. 313-320, 2001.

[12] C. Borgelt and M.R. Berthold, “Mining Molecular Fragments: Finding Relevant Substructures of Molecules,” *Proc. IEEE Int'l Conf. Data Mining (ICDM '02)*, 2002.

[13] X. Yan and J. Han, “gSpan: Graph-Based Substructure Pattern Mining,” *Proc. IEEE Int'l Conf. Data Mining (ICDM '02)*, 2002.

[14] J. Huan, W. Wang, and J. Prins, “Efficient Mining of Frequent Subgraphs in the Presence of Isomorphism,” *Proc. Third IEEE Int'l Conf. Data Mining*, pp. 549-552, 2003.

[15] S. Nijssen and J.N. Kok, “Frequent Graph Mining and Its Application to Molecular Databases,” *Proc. IEEE Int'l Conf. Systems Man and Cybernetics*, vol. 5, pp. 4571-4577, 2004.

[16] L. Zhao, L. Wong, and J. Li, “Antibody-Specified B-Cell Epitope Prediction in Line with the Principle of Context-Awareness,” *IEEE/ACM Trans. Computational Biology and Bioinformatics*, vol. 8, no. 6, pp. 1483-1494, Nov./Dec. 2011.

[17] M. Kuramochi and G. Karypis, “Discovering Frequent Geometric Subgraphs,” *Proc. IEEE Int'l Conf. Data Mining (ICDM '02)*, pp. 258-265, 2002.

[18] N. Vanetik, E. Gudes, and S.E. Shimony, “Computing Frequent Graph Patterns from Semistructured Data,” *Proc. IEEE Int'l Conf. Data Mining (ICDM '02)*, pp. 458-465, 2002.

[19] Y. Ke, J. Cheng, and W. Ng, “Correlation Search in Graph Databases,” *Proc. 13th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '07)*, pp. 390-399, 2007.

[20] Y. Ke, J. Cheng, and J.X. Yu, “Efficient Discovery of Frequent Correlated Subgraph Pairs,” *Proc. Ninth IEEE Int'l Conf. Data Mining (ICDM '09)*, pp. 239-248, http://dx.doi.org/10.1109/ICDM.2009.54, 2009.

[21] T. Ozaki and T. Ohkawa, “Mining Correlated Subgraphs in Graph Databases,” *Proc. 12th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining (PAKDD '08)*, pp. 272-283, 2008.

[22] H. Xiong, P.-N. Tan, and V. Kumar, “Mining Strong Affinity Association Patterns in Data Sets with Skewed Support Distribution,” *Proc. Third IEEE Int'l Conf. Data Mining (ICDM '03)*, pp. 387-394, 2003.

[23] I. Takigawa, K. Tsuda, and H. Mamitsuka, “Mining Significant Substructure Pairs for Interpreting Polypharmacology in Drug-Target Network,” *PLoS ONE*, vol. 6, no. 2, p. e16999, Feb. 2011.

[24] S.A. Cook, “The Complexity of Theorem-Proving Procedures,” *Proc. Third Ann. ACM Symp. Theory of Computing (STOC '71)*, pp. 151-158, 1971.

[25] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. 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.

[26] M. Johnson, I. Zaretskaya, Y. Raytselis, Y. Merezhuk, S. McGinnis, and T.L. Madden, “NCBI BLAST: A Better Web Interface,” *Nucleic Acids Research*, vol. 36, no. suppl 2, pp. W5-W9, July 2008.

[27] Z. Li, Y. He, L. Wong, and J. Li, “Progressive Dry-Core-Wet-Rim Hydration Trend in a Nested-Ring Topology of Protein Binding Interfaces,” *BMC Bioinformatics*, vol. 13, no. 1, article 51, 2012.

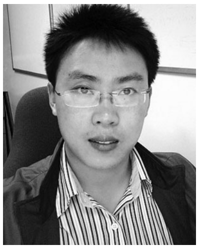
[28] C.B. Barber, D.P. Dobkin, and H. Huhdanpaa, “The Quickhull Algorithm for Convex Hulls,” *ACM Trans. Math. Software*, vol. 22, no. 4, pp. 469-483, Dec. 1996.

[29] T. Meinl, M. Wölein, O. Urzova, I. Fischer, and M. Philippsen, “The ParMol Package for Frequent Subgraph Mining,” *Electronic Comm. EASST*, vol. 1, pp. 1-12, 2006.

[30] T. Uno, M. Kiyomi, and H. Arimura, “LCM ver.3: Collaboration of Array, Bitmap and Prefix Tree for Frequent Itemset Mining,” *Proc. First Int'l Workshop Open Source Data Mining (OSDM '05)*, pp. 77-86, 2005.

[31] S.E.C. Caoili, “B-Cell Epitope Prediction for Peptide-Based Vaccine Design: Towards a Paradigm of Biological Outcomes for Global Health,” *Immunome Research*, vol. 7, no. 2, p. 2, 2011.

[32] J.A. Greenbaum et al., “Towards a Consensus on Datasets and Evaluation Metrics for Developing B-Cell Epitope Prediction Tools,” *J. Molecular Recognition*, vol. 20, no. 2, pp. 75-82, 2007.



Liang Zhao received the BS degree from Wuhan University, China, and the PhD degree from Nanyang Technological University, Singapore. His current research interests include statistical genetics, immunoinformatics, computational biology, graph theory, data mining, and machine learning.

Steven C.H. Hoi received the bachelor's degree from Tsinghua University, P.R. China, in 2002, and the PhD degree in computer science and engineering from the Chinese University of Hong Kong in 2006. He is an associate professor in the School of Computer Engineering at Nanyang Technological University, Singapore. His research interests include machine learning and data mining and their applications to multimedia information retrieval (image and video retrieval), social media and web mining, and computational finance. He has published more than 100 referred papers in top conferences and journals in related areas. He has served as general co-chair for ACM SIGMM Workshops on Social Media (WSM'09, WSM'10, WSM'11), program co-chair for the Fourth Asian Conference on Machine Learning (ACML'12), book editor for *Social Media Modeling and Computing*, guest editor for *ACM TIST*, technical PC member for many international conferences, and external reviewer for many top journals and worldwide funding agencies, including US National Science Foundation (NSF) and RGC in Hong Kong. He is a member of the IEEE and ACM.

Zhenhua Li studied computer science at Wuhan University where he received the BEng and MEng degrees in 2007 and 2009, respectively. He received the PhD degree in bioinformatics and computational biology from Nanyang Technological University in 2013. His current research interests include medical data analysis, bioinformatics, and data mining.

Limsoon Wong received the BSc(Eng) degree from Imperial College London in 1988 and the PhD degree from the University of Pennsylvania in 1994. He is a KITHCT professor of computer science and professor of pathology at the National University of Singapore. His research interests include knowledge discovery technologies and their application to biomedicine. He serves/served on the editorial boards of *Information Systems*, *Journal of Bioinformatics and Computational Biology, Bioinformatics, Biology Direct*, *IEEE/ACM Transactions on Computational Biology and Bioinformatics, Drug Discovery Today, Journal of Biomedical Semantics, and Methods*. He is a scientific advisor to Semantic Discovery Systems (United Kingdom), Molecular Connections (India), and CellSafe International (Malaysia).

Hung Nguyen received the PhD degree from the University of Newcastle, Australia, in 1980. He is a professor of electrical engineering at the University of Technology, Sydney (UTS). He is dean of the Faculty of Engineering and Information Technology and director of the Centre for Health Technologies. His research interests include biomedical engineering, advanced control, and artificial intelligence. He has developed biomedical devices for diabetes, disability, and cardiovascular diseases. He is a senior member of the IEEE, and a fellow of the Institution of Engineers, Australia, the British Computer Society, and the Australian Computer Society.

Jinyan Li received the bachelor's degree of science from the National University of Defense Technology, the master's degree of engineering from the Hebei University of Technology, and PhD degree from the University of Melbourne. He is an associate professor and core member at the Advanced Analytics Institute and Center for Health Technologies, Faculty of Engineering and IT, University of Technology, Sydney, Australia. His research is focused on fundamental data mining algorithms, machine learning, gene expression data analysis, structural bioinformatics, and information theory. He is known for the notion of emerging patterns in data mining, and is known for double water exclusion hypothesis in bioinformatics.