Index-Based Network Aligner of Protein-Protein Interaction Networks
Ahed Elmsallati, Abdulghani Msalati, and Jugal Kalita,

Abstract—Network Alignment over graph-structured data has received considerable attention in many recent applications. Global network alignment tries to uniquely find the best mapping for a node in one network to only one node in another network. The mapping is performed according to some matching criteria that depend on the nature of data. In molecular biology, functional orthologs, protein complexes and evolutionary conserved pathways are some examples of information uncovered by global network alignment. Current techniques for global network alignment suffer from several drawbacks, e.g., poor performance and high memory requirements. We address these problems by proposing IBNAL, Indexes-Based Network ALigner, for better alignment quality and faster results. To accelerate the alignment step, IBNAL makes use of a novel clique-based index and is able to align large networks in seconds. IBNAL produces a higher topological quality alignment and comparable biological match in alignment relative to other state-of-the-art aligners even though topological fit is primarily used to match nodes. IBNAL’s results confirm and give another evidence that homology information is more likely to be encoded in network topology than sequence information.

Index Terms—Biological networks, network alignment, graph isomorphism, protein-protein interactions, graph indexes

1 INTRODUCTION
As the amount of data available in molecular biology increases rapidly, the challenge of discovering and understanding relationships in such data is becoming harder and more time consuming [1]. Representing biological data as graphs or networks helps visualizing such data and facilitates analysis. Specifically, Protein-Protein Interactions (PPI) data is a type of biological information that can be modeled as networks, where nodes represent proteins and edges show that two proteins interact to perform a biological function. Understanding the topology of biological networks is getting increased attention and there are many studies concentrated on PPI networks [2]. Alignment of PPI networks, has attracted many researchers recently due to the importance of identifying potentially orthologous proteins across species, conserved pathways, and protein complexes [3].

Like many network or graph problems, network alignment is equivalent to the sub-graph isomorphism problem, which is known to be NP-Complete [4]. The network alignment approach, in general, aims to find common sub-networks among given networks. For example, if two networks are aligned, each node in the first network is matched to another node in the second network with a conserved edge [1], [5]. The differences among aligning algorithms are usually attributed to their search methodologies and the scoring functions used to identify similarities [6], [7].

Similar to sequence alignment, the alignment of PPI networks can be local or global. Local Network Alignment (LNA) tries to match small regions of similarity across networks. This kind of matching may lead to overlapped and ambiguous matching [1]. On the other hand, Global Network Alignment (GNA) aims to match each node of one network to only one node in the other network, trying to maximize the total number of nodes aligned [1], [7], [8]. While most early work considers the problem of LNA, such as PathBLAST [8], NetworkBLAST [9], NetAlign [10], MaWISh [11] and Graemlin [12], most recent work like IsoRank [1], SPINAL [13], PINALOG [14], NETAL [15], and L-GRAAL [16] focuses on solving the problem of GNA. Computationally, GNA is known to be NP-complete, which means there is no exact solution that can be used to obtain a perfect alignment in a tractable amount of time in the general case [17].

IsoRank was the first GNA algorithm developed. IsoRank uses an algorithm similar to Google’s PageRank [18] to estimate node similarity by recursively defining it in terms of the similarity of the node’s neighbors. The GRAAL family of algorithms [7], [16], [19], [20], [21], uses topological similarity, as measured by graphlet degree signatures [22], to align one network to another. PINALOG [14] uses a two-phase approach, which first extracts a set of communities from both networks and then maps the extracted communities of the first network to those extracted from the second network based on their sequence similarities. GHOST [23] takes advantage of spectral signature...
of graphs to align nodes to each other. METAL [15] iteratively builds a similarity matrix based on the neighborhood of nodes to be aligned. Then, a greedy seed-and-extend algorithm is used to construct the final alignment. SPINAL [13] is another two-phase aligner that uses both coarse-grained and fine-grained algorithms to iteratively expand the alignment. PISwap [24] is a heuristic aligner that uses a swap-based approach to improve the alignment iteratively. MAGNA [25] is a recently-published aligner that uses a genetic algorithm with a crossover and topological fit to build the resulting alignment. Optnetalign [26] is another recent aligner that optimizes multiple objectives using a memetic genetic algorithm to compare alignments in its population and decide which one to improve.

The main issue that needs to be addressed with the current aligners is running time. Most current aligners run very slowly, with some aligners taking hours to days to align mid-sized networks [17], [27]. In addition, in spite of the existence of resources that can be used to help align networks, such as function annotations, many aligners use only topological fit or sequence similarity for alignment.

In this paper, we present Index-Based Network Aligner (IBNAL), a novel algorithm for global pairwise network alignment of PPI networks. The design of IBNAL is built on two observations: (1) Proteins tend to form highly connected sub-networks [14], and (2) Sequence similarity does not usually help predict homology; instead, function similarity information is encoded in network topology [7]. Unlike other aligners, IBNAL is an algorithm that takes advantage of graph indexes and employs them to get a fast alignment. We build a novel clique-based index, which is used with function annotations to come up with a quick alignment. IBNAL’s results are compared to state-of-the-art aligners and its results are excellent when Symmetric Substructure Score ($S^3$) and Gene Ontology Consistency (GOC) are used. We also show that building time for the index is quick. The number of entries in the index is manageable; in particular, it does not take large size on disk or in memory, making IBNAL the fastest aligner developed up to date.

2 Method

In this section, we present our novel algorithm to align two PPI networks. The algorithm makes use of indices to align networks. Before describing our approach, we want to present some terminologies we use in this paper.

2.1 Preliminaries

2.1.1 Network

A network or graph $G$ is defined as a set of objects, known as nodes or vertices, that are connected via a set of relations, known as edges. Generally, a graph or network, $G$ can be written formally as $G=(V, E)$ where $V$ is the set of nodes, and $E$ is the set of edges such that $E \subseteq \{(u, v) : u, v \in V\}$.

2.1.2 Clique and Maximal Clique

A clique is a complete sub-graph that can be extracted from an original graph $G$. The extracted sub-graph contains a subset of nodes $V_i$, where $V_i \subseteq V$, such that, any two nodes in $V_i$ are connected. A clique is maximal if it cannot be extended by including any additional node from the original graph. We note here that a maximum clique is a clique that has the largest number of nodes extracted from the original graph. The clique number of any graph or network is the number of nodes in its maximum clique (See Figure 1).

2.1.3 Subordinate Node

Subordinate nodes can be defined, in the context of our problem, as the set of nodes that are not part of any cliques in a given graph. Such nodes usually, but not necessarily, connect two or more cliques or are just attached to one clique (see Table 1). If we denote the set of all subordinates as $V_s$ and the set of all nodes that are part of cliques as $V_c$, then $V = V_c \cup V_s$.

2.1.4 Clique Degree Signature And Signature Similarities

The Clique Degree Signature generalizes the idea of node’s degree and graphlet degree [22]. For each subordinate node, we count the number of different unique cliques it touches. If we denote cliques that have $k$ nodes as $k$-clique, the clique degree signature can be represented as a vector of length $l$, where $l$ is the clique number of the given network. Values of this vector are the number of $k$-cliques a subordinate node touches (See Figure 2).

The signature of a subordinate node provides a new and a novel way of summarizing the local topology.
scoring function alignment over all possible alignments as using a scoring function best alignment is the one that has the maximum score 

network. Formally, first network does not match any node in the second sometimes only 

V injective that 

teins also given a set of undirected edges, where each edge each vertex works, formally defined as follows: We are given two net-

2.1.5 Pairwise Network Alignment 

The problem of pairwise network alignment can be formally defined as follows: We are given two networks, \( G_1 = (V_1, E_1) \) and \( G_2 = (V_2, E_2) \), where \( G_1 \) and \( G_2 \) consist of \( n \) and \( m \) vertices respectively, with each vertex \( v \in V_1 \) representing a protein. We are also given a set of undirected edges, where each edge \( (u, v) \in E_i \) represents an interaction between two proteins \( u \) and \( v \). Without loss of generality, we assume that \( n \leq m \) so that each node in the smaller network can be aligned to a node in the larger network. The pairwise network alignment problem aims to find an injective function \( f : V_1 \rightarrow V_2 \) that aligns each node in \( V_1 \) to only one node in \( V_2 \). The injective function \( f \) is sometimes only partially defined when a node \( u \) in the first network does not match any node in the second network. Formally,

\[
f(u) = \{v, \text{where } u \in V_1 \text{ and } v \in V_2\}. \tag{1}
\]

The graph alignment problem aims to find the best alignment over the set of all possible alignments. The best alignment is the one that has the maximum score using a scoring function \( S \). Formally, if we denote the set of all possible alignments as \( A \), and the best alignment over all possible alignments as \( a \) using the scoring function \( S \), then \( a \) is given as

\[
a = \arg\max_{a \in A} S(a_i). \tag{2}
\]

2.2 Our Approach

The design of IBNAL is based on two observed facts. First, proteins tend to form highly connected sub-networks when they interact [14]. Second, sequence similarity does not usually help predict function similarity; instead, homology information is encoded in network topology [7]. Therefore, aligning two PPI networks would be more efficient if proteins that are connected to similar dense sub-networks are aligned first along with sub-networks they are attached to; and extending the alignment to other proteins that are connected to similar sub-networks with the goal of discovering a pair of largest connected matched sub-structures between any two given networks. If \( G_1 \) and \( G_2 \) are two PPI networks of two different species, IBNAL aligns \( G_1 \) and \( G_2 \) in three stages: clique extraction, and computation of clique-degree signature similarity signatures, and finally subordinate mapping.

2.2.1 Clique Extraction

This is a preliminary step and is considered an off-line operation where all cliques from the networks to be aligned are extracted. To achieve this goal, we deploy the Bron-Kerbosch algorithm [28] for maximal clique finding to extract all cliques from the given networks, available implementation of Bron-Kerbosch algorithm in Java is capable to extract all overlapped cliques in \( O(3^n/3) \) as worst case scenario. We notice that the maximum common clique size among all PPI networks extracted from IsoBase [29] is 11. In this study, all cliques of size 2 are excluded, as, in reality, there are as many cliques of size 2 as the number of edges in the network. Since the maximum common clique number among all the PPI networks that we are interested in does not exceed 11, cliques that have more than 11 nodes are excluded. There is only one PPI network has such size in the dataset we consider and the number of cliques is few so that we can ignore them. Figure 3 shows the number and the size of cliques that are extracted from four different well-known PPI networks, Yeast, Human, Fly, and Worm.

![Fig. 2. An illustration of Clique Degree Signature of a subordinate node. The subordinate node labeled with number 6 touches 2 cliques of different sizes, \{1, 3, 4\} of size 3, and \{7, 8, 9, 10\} of size 4. All k-cliques that this node touches is presented in the clique degree signature as a vector.](image)

### TABLE 1

Statistics for subordinate nodes in Isobase PPI networks. This table shows total number of subordinate nodes, total number of nodes that are part of cliques, and total number of subordinate nodes that touch or connect cliques.

<table>
<thead>
<tr>
<th>PPI Network</th>
<th>Number of Subordinate Nodes</th>
<th>Clique Nodes</th>
<th>Subordinate Touch/Connect Cliques</th>
</tr>
</thead>
<tbody>
<tr>
<td>S.cerevisiae (Yeast)</td>
<td>2515</td>
<td>2984</td>
<td>2257</td>
</tr>
<tr>
<td>H.sapiens (Human)</td>
<td>5579</td>
<td>4054</td>
<td>4366</td>
</tr>
<tr>
<td>D. melanogaster (Fly)</td>
<td>6238</td>
<td>1280</td>
<td>4399</td>
</tr>
<tr>
<td>C.elegans Worm</td>
<td>2550</td>
<td>255</td>
<td>1209</td>
</tr>
</tbody>
</table>
Compared to the calculation of graphlet degree vector only
Moreover, we calculate the vector of clique-degrees once for the whole network as an operation. It is a very expensive operation, it is done
Additionally is a very expensive operation, it is done
The extraction of cliques from one network computationally is a very expensive operation, it is done
In our implementation, we used functional annotations. If a clique is completely or partially aligned, no
In matching cliques, we should note that clique nodes that are connected to subordinate nodes touch. Cliques of the same size are
The matching step continues till it finishes aligning
\[ Sim(a,b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2 } \] (3)
where \(a_i\) and \(b_i\) are the number of \(k\)-cliques that each subordinate touches and \(n\) is the length of the clique-degree signature. We should note here that candidates that have score of zero or low scores are likely to be aligned than those which have high scores. Note that, computing the signature of subordinate nodes gives a new and interesting way to summarize the local topology around a specific set of nodes. Even though the extraction of cliques from one network computationally is a very expensive operation, it is done only once for the whole network as an off-line operation. Moreover, we calculate the vector of clique-degrees only for subordinate nodes, which is more efficient compared to the calculation of graphlet degree vector
<table>
<thead>
<tr>
<th>Saccharomyces cerevisiae (Yeast)</th>
<th>Homo sapiens (Human)</th>
<th>Drosophila melanogaster (Fly)</th>
<th>Caenorhabditis elegans (Worm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of nodes: 5499</td>
<td>Total number of nodes: 9633</td>
<td>Total number of nodes: 7518</td>
<td>Total number of nodes: 2305</td>
</tr>
<tr>
<td>Total number of edges: 338198</td>
<td>Total number of edges: 69386</td>
<td>Total number of edges: 25830</td>
<td>Total number of edges: 4572</td>
</tr>
<tr>
<td>Number of Nodes in clique</td>
<td>Number of Cliques</td>
<td>Number of Nodes in clique</td>
<td>Number of Cliques</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------</td>
<td>---------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>2</td>
<td>13354</td>
<td>2</td>
<td>21246</td>
</tr>
<tr>
<td>3</td>
<td>4942</td>
<td>3</td>
<td>2136</td>
</tr>
<tr>
<td>4</td>
<td>2302</td>
<td>4</td>
<td>2289</td>
</tr>
<tr>
<td>5</td>
<td>1375</td>
<td>5</td>
<td>39</td>
</tr>
<tr>
<td>6</td>
<td>722</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>368</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>333</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>284</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>237</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>150</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>61</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>

Fig. 3. Statistics for clique extraction from PPI networks of four different species. This figure shows the total number of nodes, total number of edges, and total number of \(k\)-cliques for 4 different PPI networks of different species. The statistics are obtained using IsoBase dataset used in [26].

2.2.2 Clique-Degree Signature and Signature Similarities With Index Building
In this step, we calculate the clique-degree signature for all subordinate nodes, construct the similarity matrix, and build our clique-based index. For each subordinate node, we count the number of different \(k\)-cliques it touches. These values are represented as a vector of length 9, which is the total number of \(k\)-cliques from size 3 to size 11. Synchronized with this calculation, we build our index that contains all cliques that a subordinate node touches. Figure 4 shows a sample index for a couple of subordinate nodes of the network shown in Figure 1. We should note that the signature similarity matrix is calculated as the distance between clique-degree signature vectors of any two subordinate nodes. Formally, if \(a\) and \(b\) are two subordinate nodes belonging to \(V_1\) and \(V_2\) respectively, the distance between them is given as:

![Figure 4: A Sample Index](image)

Fig. 4. A Sample Index. This figure shows a sample index for the sample network given in Figure 1. The index is divided into two parts. The first part keeps track of all cliques in the network as shown in (a). The second part of the index keeps track of all cliques that a subordinate node touches (b).
all subordinate nodes that the smaller network has. If the next subordinate node that has the lowest score is already aligned, we just skip it to pop the next candidate pair from the priority queue. We should note that IBNAL allows gapped alignment. For example, if two subordinate nodes are matched but they have different total number of touched cliques, then some of these cliques are not matched. Algorithm 1 gives the pseudocode for the subordinate mapping step of our approach.

We should mention here that the maximum number of entries in the priority queue is \( n_1 + n_2 \), where \( n_1 \) is the total number of subordinate nodes in the first network and \( n_2 \) is the total number of subordinate nodes in the second network. For each pair popped from the priority queue, IBNAL may match up to \( c \) cliques that the subordinate node touches. However, the development of IBNAL for aligning two PPI networks takes in the worst case \( O(n^2c^2) \), where \( n \) is the largest number of subordinate nodes and \( c \) is the maximum number of cliques that subordinate nodes touch (See Section 3). The number of cliques that a node touches is usually a small number less than the number of edges in the worst case.

### 3 Experimental Results

In this section, we present IBNAL’s results and discuss them with a comparison against a set of state-of-the-art aligners. IBNAL is implemented in Java with the use of JGraphT library.

The implementation of IBNAL is available at [http://www.cs.uccs.edu/~linclab/IBNAL/Documentation.html](http://www.cs.uccs.edu/~linclab/IBNAL/Documentation.html). All the experiments with IBNAL are conducted on a desktop machine, runs on OS X Yosemite Version 10.10.5, with Quad Core 2.7GHz Intel Core i7 and 16GB 1600MHz DDR3 memory.

#### 3.1 Datasets and Evaluating The Quality of Alignment

To examine our implementation of IBNAL, we evaluated both real and synthetic PPI networks, where the true alignment is known. For real networks, we run our algorithm on four extensively studied species: *Saccharomyces cerevisiae* (Yeast), *Drosophila melanogaster* (Fly), *Caenorhabditis elegans* (Worm), and *Homo sapiens* (Human). All data is retrieved from the IsoBase [29] dataset and it is the same as the one used in [26]. Figure 3 shows the number of proteins, nodes, and interactions, edges, of PPI networks we use.

For synthetic datasets, we use NAPAbench, which has been created mainly to evaluate and compare network aligners’ performance. NAPAbench has three suites of datasets and one of them is designed for pairwise alignment. Each suite contains three subcategories based on the growth model used and each model has 10 generated networks of different sizes, with 3000 to 4000 nodes. We should mention here that most recent network aligners use NAPAbench along

Algorithm 1: Pseudocode of IBNAL

1: Set $A$ initially to $\emptyset$ \Comment{the final alignment}
2: Set $S_1$ as the set of all subordinate nodes in $V_1$
3: Read in the similarity matrix, the index of subordinate nodes, and GeneOntology terms.
4: Construct the priority queue, $Q$, of subordinate node pairs ordered by confidence scores $Sim(a, b)$.
5: function DOALIGNMENT($A$, $Q$, $S_1$)
6: while ($S_1 \neq \emptyset$) do
7:    $<u, v> \leftarrow Q.pop();$
8:    if ($u$ and $v$ are not aligned) then
9:        $A.add(u, v);$  
10:       $S_1 \leftarrow S_1 \setminus \{u\};$
11:       $C_1 \leftarrow u.get Cliques();$
12:       $C_2 \leftarrow v.get Cliques();$
13:       MATCHCLIQES($C_1$, $C_2$);  
14: end if
15: end while
16: return $A$;
17: end function
18: procedure MATCHCLIQES($A$, $C_1$, $C_2$)
19: for each $i \in C_1$ do
20:    maxShared $\leftarrow 0$;
21:    for each $j \in C_2$ and $i.size() = j.size()$ do
22:        if (sharedGO($i,j$) $> maxShared$) then
23:            $a \leftarrow i;$
24:            $b \leftarrow j;$
25:            maxShared $\leftarrow$ sharedGO($i,j$);
26:        end if
27:    end for
28:    $A.add(a, b);$  
29: end for
30: end procedure

For biological assessment, we evaluate IBNAL’s performance using the Gene Ontology Consistency (GOC) [13] which uses Gene Ontology (GO) [31] annotations to enhance the measurement of the alignment quality. GOC is defined formally as

$$GOC(G_1, G_2, f) = \sum_{(u,v) \in a} \frac{|GO(u) \cap GO(v)|}{|GO(u) \cup GO(v)|}$$

where $u$ and $v$ are two nodes matched in the resulting alignment $a$ and $u \in V_1$, $v \in V_2$. The set of GO annotations of a protein $u$ is denoted by $GO(u)$, where all terms are extracted from the GO database with the exclusion of the root terms for each protein. To minimize redundancy, we limit the GO terms used to a maximum distance of five from the root of the ontology.

3.2 Discussion

3.2.1 Alignment Evaluation

We examined IBNAL on two different datasets, the Isobase and NAPAbench suite datasets. We selected state-of-the-art aligners to compare our algorithm based on availability and approach used. First, we selected C-GRAAL and PINALOG as these two aligners use a community-based topological approach similar to our algorithm. MAGNA and GHOST are two recently published aligners that have the best performance among topological aligners [27]. In addition, we choose NETAL because it is one of the fastest topological aligners [27]. All parameters and setting used for running other aligners is shown in Table 2.

<table>
<thead>
<tr>
<th>Aligner</th>
<th>Settings and Parameters Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>NETAL</td>
<td>-a 0.0001 -b 0 -c 0.5 -i 2</td>
</tr>
<tr>
<td>PINALOG</td>
<td>No user-provided parameters</td>
</tr>
<tr>
<td>C-GRAAL</td>
<td>No user-provided parameters</td>
</tr>
<tr>
<td>GHOST</td>
<td>For the extractor, $k = 3$, except for H. sapiens, where the extractor ran out of memory, in which case we used $k = 2$. For the aligner, matcher: linear, nneighbors: all, beta: 1.0, ratio:8.0, search: 10.</td>
</tr>
<tr>
<td>MAGNA</td>
<td>-m s3 -p 15000 -n 2000</td>
</tr>
</tbody>
</table>

Isobase Dataset: The first benchmark results that we present here are alignments based upon the PPI networks included in the IsoBase [29] dataset. We report the results of aligning PPI networks based upon the evaluation metrics mentioned above. IBNAL achieved the best $S^3$ score among all aligners for 5 of the 6 alignments examined. There is only one alignment, the one produced by NETAL that has a score, 0.149, a bit higher the score that IBNAL produces,
0.106 (See Table 3). IBNAL aligns only identical substructures and there are no sparse-to-dense or dense-to-sparse alignments that can be penalized. We note here that GHOST crashed when testing on the sc-dm and sc-hs dataset pairs and it appears in Figure 6 (a) as missing data.

### Table 3

<table>
<thead>
<tr>
<th>Aligner</th>
<th>ce-dm</th>
<th>ce-hs</th>
<th>ce-sc</th>
<th>dm-hs</th>
<th>sc-dm</th>
<th>sc-hs</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBNAL</td>
<td>0.611</td>
<td>0.612</td>
<td>0.604</td>
<td>0.264</td>
<td>0.106</td>
<td>0.142</td>
</tr>
<tr>
<td>C-GRAAL</td>
<td>0.142</td>
<td>0.131</td>
<td>0.009</td>
<td>0.119</td>
<td>0.118</td>
<td>0.107</td>
</tr>
<tr>
<td>GHOST</td>
<td>0.139</td>
<td>0.148</td>
<td>0.118</td>
<td>0.115</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MAGNA</td>
<td>0.261</td>
<td>0.245</td>
<td>0.246</td>
<td>0.107</td>
<td>0.107</td>
<td>0.116</td>
</tr>
<tr>
<td>NETAL</td>
<td>0.412</td>
<td>0.396</td>
<td>0.323</td>
<td>0.187</td>
<td>0.149</td>
<td>0.141</td>
</tr>
<tr>
<td>PINALOG</td>
<td>0.006</td>
<td>0.006</td>
<td>0.005</td>
<td>0.005</td>
<td>0.004</td>
<td>0.005</td>
</tr>
</tbody>
</table>

With respect to GOC, IBNAL has unstable performance. While IBNAL outperforms other aligners in some experiments, it has comparable results in other experiments. To be more specific, IBNAL does not show good GOC scores when aligning C.elegans to different species, even though it scores the best S^3 among all aligners in these experiments. On the other hand, PINALOG and GHOST perform better than IBNAL in these tests with respect to GOC score, but they have a very low S^3 score. While GHOST crashed analyzing the last two alignments, IBNAL scored the best GOC scores aligning S.cerevisiae to D.melanogaster and S.cerevisiae to H.sapiens. As IBNAL allows gap alignment, we report the GOC scores in two different ways. The first is the actual score of resulting alignments; the second is the percentage of GOC score to the total number of proteins aligned (See Table 4 for the total number of pairs aligned). In Table 5, we report the actual GOC score of aligning sc-hs. Even though IBNAL allows gap alignment and it aligns fewer nodes than other aligners, it has the best GOC score among all aligners except PINALOG. It is worth noting that PINALOG has a very low scores with respect to S^3 which suggests most of the proteins aligned are sparse-to-dense or dense-to-sparse penalized.

### Table 4

Total number of pairs aligned for different IsoBase tests. Missing data indicates the failure of aligner to produce alignment.

<table>
<thead>
<tr>
<th>Aligner</th>
<th>ce-dm</th>
<th>ce-hs</th>
<th>ce-sc</th>
<th>dm-hs</th>
<th>sc-dm</th>
<th>sc-hs</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBNAL</td>
<td>2797</td>
<td>2797</td>
<td>2765</td>
<td>6814</td>
<td>3409</td>
<td>4515</td>
</tr>
<tr>
<td>C-GRAAL</td>
<td>2805</td>
<td>2805</td>
<td>2805</td>
<td>7518</td>
<td>5499</td>
<td>5499</td>
</tr>
<tr>
<td>GHOST</td>
<td>2805</td>
<td>2805</td>
<td>2805</td>
<td>7518</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MAGNA</td>
<td>2805</td>
<td>2805</td>
<td>2805</td>
<td>7518</td>
<td>5499</td>
<td>5499</td>
</tr>
<tr>
<td>NETAL</td>
<td>2805</td>
<td>2805</td>
<td>2805</td>
<td>7518</td>
<td>5499</td>
<td>5499</td>
</tr>
<tr>
<td>PINALOG</td>
<td>2088</td>
<td>2255</td>
<td>1427</td>
<td>5771</td>
<td>3646</td>
<td>3909</td>
</tr>
</tbody>
</table>

We should mention here that the low GOC scores of IBNAL in some of the experiments mentioned above do not reflect the true performance of IBNAL. These low scores can be due to: (1) a lack of identical substructures that can be aligned with the experiments of C.elegans; (2) the GO terms are not precise enough or largely assigned [32]; also (3) C.elegans has one of the smallest PPI networks in the IsoBase dataset and it does not contain as many cliques as found in the other PPI networks, although the total number of subordinate nodes is high compared to other PPI networks.

### Table 5

GOC scores for different IsoBase tests. Missing data indicates the failure of aligner to produce alignment.

<table>
<thead>
<tr>
<th>Aligner</th>
<th>ce-dm</th>
<th>ce-hs</th>
<th>ce-sc</th>
<th>dm-hs</th>
<th>sc-dm</th>
<th>sc-hs</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBNAL</td>
<td>20.37</td>
<td>18.51</td>
<td>17.60</td>
<td>74.01</td>
<td>72.63</td>
<td>160.1</td>
</tr>
<tr>
<td>C-GRAAL</td>
<td>75.95</td>
<td>41.21</td>
<td>53.48</td>
<td>110.5</td>
<td>100.5</td>
<td>123.2</td>
</tr>
<tr>
<td>GHOST</td>
<td>181.5</td>
<td>84.85</td>
<td>95.43</td>
<td>166.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MAGNA</td>
<td>12.96</td>
<td>18.64</td>
<td>25.26</td>
<td>37.54</td>
<td>48.84</td>
<td>79.59</td>
</tr>
<tr>
<td>NETAL</td>
<td>12.01</td>
<td>18.64</td>
<td>25.26</td>
<td>37.54</td>
<td>48.84</td>
<td>79.59</td>
</tr>
<tr>
<td>PINALOG</td>
<td>229.8</td>
<td>96.62</td>
<td>136.5</td>
<td>302.3</td>
<td>363.7</td>
<td>320.4</td>
</tr>
</tbody>
</table>

**NAPAbench Dataset:** As we see in the previous benchmark, IBNAL has the best S^3 scores among selected aligners and competitive GOC scores for data where true alignment is unknown. We also test IBNAL on the NAPAbench dataset [33] where the true alignment is known. For the Crystal Growth (CG) suite, which consists of 10 pairs of families, IBNAL maps approximately 52.6% of nodes to true orthologs. For other growth models, Duplication Mutation Completion (DMC) and Duplication with Random Mutation (DMR), IBNAL, surprisingly, gets only 3% nodes mapped to true orthologs. These low scores of DMC and DMR require further investigation. We analyzed the NAPAbench dataset and we compared it with IsoBase in terms of the total number of subordinate nodes and the sizes of cliques extracted. We did find that there is a clear distinction between the real and synthetic PPI networks. The real PPI networks of IsoBase have between 13% to 45% of nodes as subordinate nodes. On the other hand, the NAPAbench suite of CG has a very low number of subordinate nodes which means that the networks of this suite are highly connected and most of nodes are parts of cliques. The DMC and DMR suites have a very high percentage of subordinate nodes, between 71% to 95%, which means that these two suites are highly sparse (See Table 6). In addition, all cliques extracted from NAPAbench dataset have clique sizes of 3, 4, and 5. Only the DMR suite has cliques of size 10. As we can see, the number of subordinate nodes and the sizes of cliques extracted from the synthetic dataset are completely different compared to those extracted from real networks (See Figure 3). This distinction...
between synthetic and real dataset supports claims in our previous work, [27], stating that NAPAbench networks tend to be sparse, such as DMC and DMR suites, or highly connected, like the CG suite, in a way that affects the performance of aligners. To conclude, as we mentioned in the introduction, the design of IBNAL is based on observations from analyzing real PPI networks.

### TABLE 6
The percentage of subordinate nodes and clique sizes for NAPAbench synthetic dataset. Data shown in this table is extracted using all families of pairwise suite of NAPAbench.

<table>
<thead>
<tr>
<th>Growth Model</th>
<th>% of Subordinate Nodes</th>
<th>Clique Sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crystal Growth (CG)</td>
<td>Less than 1.0%</td>
<td>3, 4, and 5</td>
</tr>
<tr>
<td>Duplication Mutation Complementation (DMC)</td>
<td>71% to 76%</td>
<td>3, 4, and 5</td>
</tr>
<tr>
<td>Duplication with Random Mutation (DMR)</td>
<td>91% to 95%</td>
<td>3, 4, 5, 6 and 10</td>
</tr>
</tbody>
</table>

### 3.2.2 Index Performance
Now we examine the performance of our index in terms of building time, number of entries, and index size as shown in Figure 7. Each sub-figure depicts results of four different PPI networks of the species mentioned in Figure 3. As we can see, the building time of the index is almost twice the time needed to extract all cliques from the network (Figure 7a). The building time is still fast, compared to the time taken by other aligners to calculate the similarity matrix and to perform the alignment, which may takes hours to days for some aligners [17] [27]. We note that building the index for *C. elegans* (Worm) does not take more than 5 seconds, even though the PPI network for this species has around 3,000 proteins. This short building time is due to the *C. elegans* PPI network containing approximately 5,000 interactions. The number of entries in our index reflects the efficiency of IBNAL. Comparing the number of entries to the original number of nodes in the network, the alignment process is done quickly; however, IBNAL uses most of its runtime analyzing and aligning subordinate nodes and attached cliques. The number of subordinate nodes is roughly one half of the number of all nodes for the largest PPI networks. Only one small network has nearly as many subordinate nodes as the original number of nodes, due to the sparse nature. Precisely, IBNAL reduces the time of aligning networks to the time needed for aligning subordinate nodes only. Also, all cliques attached to any subordinate nodes aligned are reached faster as we keep links to those cliques using our index. Moreover, Figure 7(c) shows that the size of our clique-based index ranges between tens of kilobytes, for smaller networks, to 1.2 megabytes, for larger networks. We can see that this index size range easily allows for the entire index file to be resident in memory which permits faster performance. To conclude, IBNAL does not take more than a minute to align any two PPI networks, mentioned above, when all off-line data, index files and similarity matrices, are fed. Aligning any two networks in such time makes IBNAL one of the fastest aligners available.

### 4 Conclusion
In this paper, we present a new and novel methodology for local topology of PPI networks where functional orthology is usually encoded in network topology instead of sequence similarity. We employ a methodology to build a clique-based index, which is used in our novel aligner IBNAL. IBNAL’s results are comparable to other state-of-the-art aligners but IBNAL beat other aligners in the time taken to extract the alignment and also with respect to topological quality of its alignment. IBNAL also provides another proof that homology information is encoded in the network topology instead of sequence similarity.

The implementation of IBNAL and its results show that NAPAbench dataset is a quite different in its topological properties than the real PPI networks. For some suites, it is denser than the real networks, and for the rest is substantially more sparse than the real PPI networks.
Fig. 7. The Impact of Indexing PPI Networks. This figure shows the impact of indexing PPI networks for four different species in terms of (a) time needed to build the index, (b) number of entries in each index, and (c) the size taken by each index on disk.

ACKNOWLEDGMENTS

This work was performed in the (LINC Lab) at University of Colorado, Colorado Springs. The Ministry of Higher Education and Scientific Research, Libya (MOHESR) and University of Tripoli supported this work under grant 972-2007. Special thanks to Nabeeh Hasan, National Jewish Health, for his helpful discussion and comments.

REFERENCES


Ahed Elmsallati received the BSc degree in Computer Science in 1998 from University of Tripoli, formerly known as Al-Fatah University, Libya, the Master of Science degree in Computer Science in 2011 from New Mexico State University, USA, and joined University of Colorado at Colorado Springs in 2013 as a graduate student. His research area include databases, graph algorithms, and biological computations. His current work focuses on large biological networks alignment and matching.

Abdulghani Msalati is associate professor of Biochemistry at the University of Tripoli, School of Medicine Tripoli, Libya. He received his Bachelor of Science in Biology from the University of Tripoli, Libya. He obtained his Master of Science in Biochemistry from the University of Manchester Institute of Science and Technology (UMIST), United Kingdom, and his Ph.D from University of Salford, United Kingdom. His research interests include protein structure and components, DNA cleavage, signal process, bioinformatics, and molecular biology.

Jugal Kalita is a professor of Computer Science at The University of Colorado at Colorado Springs, received his Bachelor of Technology degree from the Indian Institute of Technology in Kharagpur, India, his Master of Science degree from the University of Saskatchewan, Canada, and a Master of Science and PhD from the University of Pennsylvania. He is also the founder of the Bioinformatics and Biotechnology Symposium (BIOT). His research interests are in artificial intelligence, bioinformatics, information retrieval, and natural language processing and machine learning. He is the author of 150 papers in reputed conferences and journals.