BIOINFORMATICS ORIGINAL PAPER

Vol. 31 no. 3 2015, pages 363–372
doi:10.1093/bioinformatics/btu652

Systems biology

Advance Access publication October 4, 2014

LocalAli: an evolutionary-based local alignment approach to identify functionally *conserved* modules in multiple networks

Jialu Hu* and Knut Reinert

Department of Mathematics and Computer Science, Freie Universität Berlin, Takustrasse 9, 14195 Berlin, Germany

Associate Editor: Jonathan Wren

ABSTRACT

Motivation: Sequences and protein interaction data are of significance to understand the underlying molecular mechanism of organisms. Local network alignment is one of key systematic ways for predicting protein functions, identifying functional modules and understanding the phylogeny from these data. Most of currently existing tools, however, encounter their limitations, which are mainly concerned with scoring scheme, speed and scalability. Therefore, there are growing demands for sophisticated network evolution models and efficient local alignment algorithms.

Results: We developed a fast and scalable local network alignment tool called LocalAli for the identification of functionally conserved modules in multiple networks. In this algorithm, we firstly proposed a new framework to reconstruct the evolution history of conserved modules based on a maximum-parsimony evolutionary model. By relying on this model, LocalAli facilitates interpretation of resulting local alignments in terms of conserved modules, which have been evolved from a common ancestral module through a series of evolutionary events. A meta-heuristic method simulated annealing was used to search for the optimal or near-optimal inner nodes (i.e. ancestral modules) of the evolutionary tree. To evaluate the performance and the statistical significance, LocalAli were tested on 26 real datasets and 1040 randomly generated datasets. The results suggest that LocalAli outperforms all existing algorithms in terms of coverage, consistency and scalability, meanwhile retains a high precision in the identification of functionally coherent subnetworks.

Availability: The source code and test datasets are freely available for download under the GNU GPL v3 license at https://code.google.com/p/localali/.

Contact: jialu.hu@fu-berlin.de or knut.reinert@fu-berlin.de.

Supplementary information: Supplementary data are available at Bioinformatics online.

Received on April 25, 2014; revised on September 22, 2014; accepted on September 29, 2014

1 INTRODUCTION

A *functional module* is, by definition, a discrete entity whose function is separable from those of other modules (Hartwell *et al*., 1999). They are usually separated based on spatial localization (e.g. a ribosome) or chemical specificity (e.g. a signal transduction system) and composed of many types of molecules, such as proteins, DNA, RNA and small molecules. A deficiency of a comprehensive chart of functional modules within organisms becomes an obstacle to unravel the general design principles that govern the structure and behavior of modules, and understand the evolutionary constraints they must obey. Thus, the problem of identifying functional modules attracts many researchers in the fields of computational biology and systems biology. To come up to this problem, *network alignment* based on evolutionary cross-species comparison provides a valuable framework (Sharan, 2005; Sharan and Ideker, 2006). Despite the existence of many network alignment tools such as *PathBLAST* (Kelley *et al*., 2004), *NetworkBlast* (NB; Kalaev *et al*., 2008), *Natalie 2.0* (El-Kebir *et al*., 2011) and *AlignMCL* (Guzzi and Mina, 2012), the developments of faster and more efficient tools are of significance to cope with the rapidly growing biological data.

Network alignment aims at finding *one-to-one* or *many-to-many* node mapping tables by comparing networks based on information from sequence similarity, topology conservation, co-expression, co-evolution, etc. Nodes that are grouped into a same cluster in a node-mapping table constitute an equivalence class. Each equivalence class must bear at most one protein from each species in one-to-one tables, whereas it might receive more than one node from a same species in many-to-many tables. Generally, network alignment algorithms are categorized into *pairwise* and *multiple* network alignments according to the number of compared species (Clark and Kalita, 2014), and into *local* and *global* network alignments according to its target regions of interest. Pairwise approaches align two networks and multiple approaches three and more networks. *Local* alignment approaches detect node mapping tables for conserved subnetworks, which are usually independent and high-scoring local regions, each implying a putative functional module such as a protein complex (Sharan, 2005) or metabolic pathway (Kelley *et al*., 2004). *Global* alignment approaches determine an optimal global node mapping table for the compared networks (Huang *et al*., 2013; Li *et al*., 2007; Milenković *et al*., 2013; Singh *et al*., 2007, 2008), each set of matched nodes (i.e. proteins) implying a putative function-oriented ortholog group. Typically, *pairwise global* alignment methods attempt to provide one-to-one mapping tables including *NETAL* (Neyshabur *et al*., 2013), *MI-GRAAL* (Kuchaiev and Pržulj, 2011), *H-GRAAL* (Milenković *et al*., 2010), *SPINAL* (Aladağ and Erten, 2013) and *MAGNA* (Saraph and Milenković, 2014). *Multiple global* alignment and *pairwise*/*multiple local* alignment attempt to give many-to-many mapping tables, such as *SMETANA* (Sahraeian and Yoon, 2013) and *NetCoffee* (Hu *et al*., 2014). The scope of this paper focuses on the problem of *multiple local* alignment.

*To whom correspondence should be addressed.

Both *in silico* and *in vivo* studies suggest that *functional modules* of organisms tend to be conserved during the evolution history (Pellegrini *et al.*, 1999; Roguev *et al.*, 2008). Grounded on this hypothesis, *local* network alignment provides a general computational framework that searches for high-scoring conserved subnetworks to detect functionally conserved modules. The developments of *local* alignment tools or web servers have become a quite active field in the last decade. The most notable *pairwise local* alignment tools include *PathBlast* (Kelley *et al.*, 2004), *MaWISh* (MW; Koyutürk *et al.*, 2006), *NetworkBlast* (Kalaev *et al.*, 2008), *AlignNemo* (AN; Ciriello *et al.*, 2012) and *NetAligner* (Pache *et al.*, 2012). Just a few *multiple local* alignment tools have been developed. The currently existing *multiple local* alignment tools include *Graemlin* (Flannick *et al.*, 2006, 2009), *CAPPI* (Dutkowski and Tiuryn, 2007) and *NetworkBlast-M* (NBM; Kalaev *et al.*, 2009). In addition, there are also some works trying to detect functionally conserved modules by using a combination of clustering algorithms and global alignment algorithms, such as *PINALOG* (Phan and Sternberg, 2012). See more information about existing local and global network alignment tools in the supplementary material.

The currently existing *multiple local* alignment tools are concerned with three major issues. The first one is the scalability. To date, *CAPPI* has been applied to the analysis of three networks and only compatible with particularly designed data. *NBM* is unable to run on networks containing nodes with a large vertex degree (Hu *et al.*, 2014). Thus, the scalability of these tools is at a modest level. Another issue is the evolutionary relevance of the reported hits. To answer the question of how conserved modules of descendants have been evolved from their origin, the scoring schemes shall be more strongly rooted in an evolutionary model (Sharan and Ideker, 2006). But, neither the evolution history nor a probabilistic model of network growth was considered by *Graemlin* and *NBM*. The third issue is speed. The problem of aligning multiple networks is computationally intractable. Parallelization techniques can largely speed up local alignment algorithms because each target of interest can be searched by a single thread. Yet, none of the existing multiple local alignment tools support parallel computing.

To remedy these limitations, we developed a fast and scalable alignment tool *LocalAli* (LA) for the identification of functionally conserved modules in multiple networks. A new framework was firstly proposed for the inference of the evolution history of functional modules based on a *maximum-parsimony* evolutionary model. By relying on this model, LA facilitates the interpretation of resulting local alignments in terms of *conserved modules* that have been evolved from a common ancestral module through a series of evolutionary events.

To evaluate the performance, LA and four previous algorithms were tested on 26 real datasets and 1040 random datasets. We assessed the quality of the alignment results in terms of coverage, consistency, prediction of protein functions and prediction of protein complexes. The results show that LA outperforms all previous algorithms in terms of coverage, consistency and scalability, meanwhile retains a high precision in predicting functionally coherent subnetworks (FCS). Moreover, LA is also preferable to the *state-of-the-art* algorithm *NBM* in predicting protein functions and protein complexes.

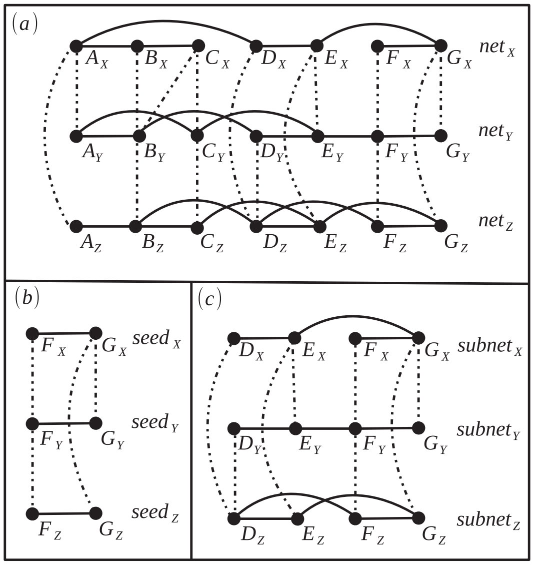
(a)
A_X B_X C_X D_X E_X F_X G_X net_X
A_Y B_Y C_Y D_Y E_Y F_Y G_Y net_Y
A_Z B_Z C_Z D_Z E_Z F_Z G_Z net_Z
(b)
F_X G_X $seed_X$
F_Y G_Y $seed_Y$
F_Z G_Z $seed_Z$
(c)
D_X E_X F_X G_X $subnet_X$
D_Y E_Y F_Y G_Y $subnet_Y$
D_Z E_Z F_Z G_Z $subnet_Z$

Fig. 1. An example of searching for a *d-subnet* from PPI networks of three species. (**a**) A three-layer (*k-layer* in general case) graph consisting of PPI networks and their bipartite graphs of three species *X*, *Y*, *Z*. In the graph, each layer is a PPI network, solid lines are interactions and dashed lines are edges of homologous proteins. (**b**) One of refined seeds consisting of two *k-spines*. (**c**) A *d-subnet* extended from the seed in (b)

2 DEFINITIONS

We use attributed undirected graphs $\{G_1, G_2, \cdots, G_k\}$ to represent k protein–protein interaction (PPI) networks. Each graph $G_i = (V_i, E_i, \mathcal{A}_i)$ corresponds to a species, where V_i represents all the proteins, E_i is the collection of interactions and $\mathcal{A}_i : V_i \rightarrow \Sigma^*$ is a labeling function that assigns protein sequences to their nodes. Further, a set of $\binom{k}{2}$ bipartite graphs $B_{ij} = (V_i \cup V_j, E_{ij})$ can be constructed by joining pairs of proteins between V_i and V_j if their sequences are sufficiently similar. To be clear, we refer to elements of E_i and E_{ij} as *interactions* (solid lines in Fig. 1a) and *edges* (dashed lines in Fig. 1a), respectively. A set of k proteins, each from one species, which are connected by *edges* is termed as a *k-spine* (Kalaev *et al.*, 2009), such as $\{A_X, A_Y, A_Z\}$ in Figure 1a. And a set of *d k-spines* connected by *interactions* form a *d-subnet*, such as the four *k-spines* in Figure 1c. Proteins that participate in a common structural complex or metabolic pathway are called *functionally linked* (Pellegrini *et al.*, 1999). These groups of functionally linked proteins are *functional modules*.

Local network alignment aims to detect high-scoring *d-subnets* that imply putative functional modules of the compared species, such as protein complexes. These *d-subnets* might be many-to-many node mapping tables because two *k-spines* within a *d-subnet* could probably overlap. In contrast, *global network alignment* aims to determine a best overall node mapping (i.e. with the highest alignment score), which consists of a set of mutually disjoint match-sets (Hu *et al.*, 2014; Liao *et al.*, 2009). Each match-set contains a group of proteins (≥ 2) and is considered as an equivalence class.

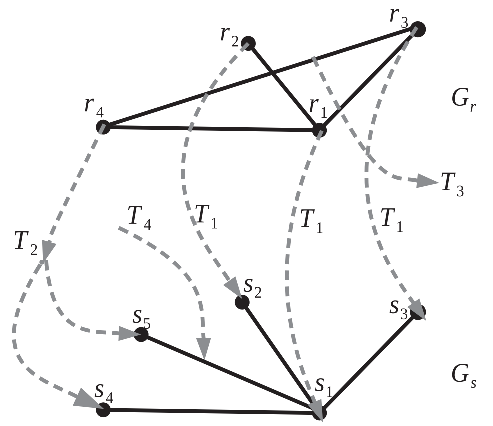

Fig. 2. Illustration of the evolutionary model. G_r and G_s are two functional modules. Proteins are represented by nodes, interactions by solid lines, evolutionary events from G_r to G_s by dashed arrows. T_1, T_2, T_3, T_4 refer to evolutionary events *protein mutation*, *protein duplication*, *interaction deletion* and *interaction insertion*, respectively. Suppose $t = 1$, $\alpha = 0.2$ and $\beta = 2$, by definition, the evolutionary distance can be calculated as follows: $f(G_r, G_s, \mathcal{M}_{rs}) = f_1 + f_2 + f_3 + f_4 = (e^{-0.2} + e^{-0.2\times2} + e^{-0.2\times3})+ e^{-0.2\times2} + e^{-0.2\times2} + e^{-0.2\times2} = 4.049$

3 METHODS

3.1 Overview

To identify functionally conserved modules from multiple networks, we proposed an evolutionary-based local alignment algorithm to heuristically search for high-scoring *d-subnets* with the information of interactions, homologous proteins and phylogenetic trees. First, the method constructs a set of k PPI networks and bipartite graphs with interactions and homologous proteins. These networks and bipartite graphs are integrated into a *k-layer* graph (Kalaev *et al.*, 2009) as illustrated in Figure 1a. Then, it heuristically searches for a set of refined seeds using a *seed-and-extend* approach (Fig. 1b) from the *k-layer* graph and extend them with a local search strategy to *d-subnets* (Fig. 1c), which are in a range of predefined *minimal* and *maximal* size. Afterward, the k-induced subnetworks of each *d-subnet* are set as the leaves of an evolutionary tree (see in Fig. 3b), which has the same topology and branch weights with its corresponding phylogenetic tree of the involved species (Fig. 3a). Under the *maximum parsimony* principle, the optimal or near-optimal inner nodes ($subnet_v$ and $subnet_w$ in Fig. 3b and c) are found by using *simulated annealing* (SA) such that the tree receives a minimal evolutionary distance according to our evolutionary model. Finally, an alignment score of each *d-subnet* is calculated and those scoring less than a threshold are filtered away.

3.2 Models of functional module evolution

3.2.1 Existing evolutionary models In PPI networks, gene duplication and divergence are most probably the underlying biological mechanism for generating the scale-free topological feature (Vazquez *et al.*, 2003; Wagner, 2003). Among all existing *multiple local* alignment tools, only *CAPPI* uses a network growing model (i.e. duplication-divergence) to derive the posterior probabilities of interactions in ancestral PPI networks, whereas other tools are not strongly rooted in an evolutionary model. In addition, there are some other computational models applied to the problem of network history inference, namely, *maximum-likelihood* (Zhang and Moret, 2008) and *parsimonious-histories* (Patro *et al.*, 2012; Patro and Kingsford, 2013). Inspired by the latter approaches, we here introduce a similar parsimony-based model that aims to reconstruct the ancestral subnetwork for a set of conserved subnetworks. This model is designed based on a hypothesis that proteins that function together in a pathway or a structural complex are likely to evolve in a correlated fashion (Pellegrini *et al.*, 1999). It means that proteins of functional modules tend to be either preserved or eliminated all together during the evolution of the whole PPI network from their common ancestor. Unlike Dutkowski's algorithm (Dutkowski and Tiuryn, 2007), which gives a global view of the evolution of the whole networks, LA provides a new framework to reconstruct the evolution of conserved subnetworks.

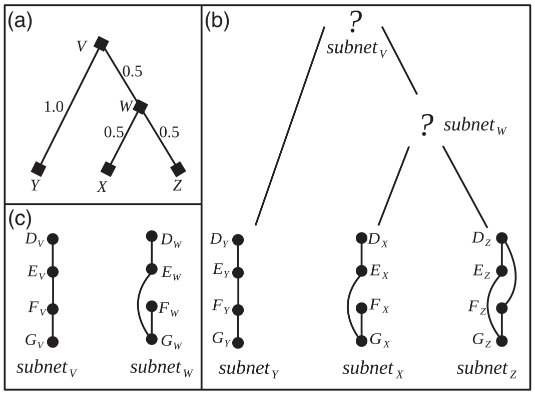

Fig. 3. Reconstruct an evolutionary tree of a *d-subnet*. (a) Give a phylogenetic tree of species X, Y, Z. (b) Set the k-induced subnetworks of a *d-subnet* as leaves of its evolutionary tree. This tree has the same topology and branch weight with its species tree. (c) Reconstruct optimal or near-optimal *inner nodes* of $subnet_V$ and $subnet_W$ such that this evolutionary tree has the minimal evolutionary distance. Let $\alpha = 0.2$, $\beta = 2$. The distance is calculated as follows: $f(V, Y, M_{VY}) + f(V, W, M_{VW}) + f(W, X, M_{WX}) + f(W, Z, M_{WZ}) = (2e^{-2\alpha} + 2e^{-\alpha}) \times 1.0 + (2e^{-2\alpha} + 2e^{-\alpha} + 2e^{-\alpha\beta}) \times 0.5 + (2e^{-2\alpha} + 2e^{-\alpha}) \times 0.5 + (4e^{-2\alpha} + e^{-\alpha\beta})\times 0.5 = 8.302$

3.2.2 The evolutionary tree To elucidate the phylogenetic relationships of functional modules, we use a binary tree to model the evolutionary process (see examples in Fig. 3b). In this tree, *external nodes*, which are also called *leaves* represent the observed functional modules of our compared species. *Internal nodes* represent the corresponding functional modules of the predecessor species. The *root* represents the corresponding functional module of the original species.

3.2.3 Evolutionary events and distances Evolutionary events are the basic building blocks of network evolution, and evolutionary distance describes how far a descendant subnetwork goes away from an ancestral subnetwork. To infer the evolution history, it is necessary to introduce the definition of evolutionary event and distance. Pellegrini's investigation and the scale-free topological features show that duplication and divergence are the major driving forces of network evolution (Pellegrini *et al.*, 1999; Wagner, 2003). Taking these evidences into consideration, we attempt to understand the evolution process using the following four types of evolutionary events:

(1) *Protein mutation*: The sequence change of two proteins in two species;

(2) *Protein duplication*: The duplication of a protein in an offspring species;

(3) *Interaction deletion*: The loss of an interaction from one network to another;

(4) *Interaction insertion*: The gain of an interaction from one network to another.

Let $G_r = (V_r, E_r, \mathcal{A}_r)$ and $G_s = (V_s, E_s, \mathcal{A}_s)$ be two functional modules. As illustrated in Figure 2, G_s descends from G_r according to a correspondence match $\mathcal{M}_{rs}: V_r \rightarrow V_s$. We denote as $f_i(G_r, G_s, \mathcal{M}_{rs})$ the evolutionary distance caused by type i events during the evolution from G_r to G_s. An investigation (Fraser *et al.*, 2002) showed that proteins with more interactions evolved more slowly because a greater proportion of proteins are directly involved in its function. In other words, proteins with different number of interactors have different evolutionary rates. Hence, we choose $e^{-\alpha \cdot deg(v)}$ as the function to calculate the evolutionary rate of a protein v, and $e^{-\alpha \cdot \beta}$ as the evolutionary rate of each interaction in the PPI networks (see an example in Supplementary Fig. S2). Consequently, the evolutionary distance function of each evolutionary event is written as follows:

$$f_i(G_r, G_s, \mathcal{M}_{rs}) = \begin{cases} \Sigma_{v \in T_i} e^{-\alpha \cdot deg(v)} t & i \in \{1, 2\} \\ \Sigma_{e \in T_i} e^{-\alpha \cdot \beta} t & i \in \{3, 4\} \end{cases}$$

where T_i is the collection of type i events, $deg(v)$ is the number of interactions connected with protein $v \in V_r$, α and β are parameters adjusting the evolutionary rates, t is the evolutionary time from G_r to G_s. The *evolutionary distance* between G_r and G_s, $f(G_r, G_s, \mathcal{M}_{rs})$, is defined as $f(G_r, G_s, \mathcal{M}_{rs}) = \Sigma_{i=1}^{4} f_i(G_r, G_s, \mathcal{M}_{rs})$. We chose proper values for α and β so that evolutionary distances caused by protein events and interaction events are in balance. Generally, the distances would be in balance if the following two requirements are fulfilled: (i) the evolutionary rate of interaction is similar with that of protein with 2 interactions; (ii) the evolutionary rate of protein is <0.2 when the protein has more than 10 interactions. If β is too small, type 3 and 4 events become unwelcome in searching for optimal evolutionary tree because these events would result a larger evolutionary distance. If β is too large, type 3 and 4 events become popular because they would not actually make a big effect on the evolutionary distance. For this reason, we tested a series of parameters and chose $\alpha = 0.2$ and $\beta = 2.0$ for all of our tests because it can make interaction distance and protein distance in balance (see more in Supplementary Fig. S2). We measure the evolutionary time t by the branch weight of the tree as shown in Figure 3a. The topology and branch weight of the evolutionary tree are calculated based on the common tree of the NCBI taxonomy database (Federhen, 2012). See more information about the tree in Figure 2 and the Supplementary Material.

3.3 Reconstruction of ancestral functional modules

To exactly answer how the extant functional modules evolved in the evolutionary history, the reconstruction of ancestral functional modules becomes a central problem. With this intention, we model it as an optimization problem of finding a series of optimal ancestral subnetworks that yield the smallest distance in the evolutionary tree. Subsequently, we use a meta-heuristic method *SA* (Kirkpatrick *et al.*, 1983) to find the optimal or near-optimal solution (Fig. 3).

3.3.1 The optimization problem To explain the descent of the extant functional modules, we estimate their ancestral functional modules (or internal tree nodes) using the *maximum parsimony* principle (Felsenstein, 2003; Fitch, 1971). Under this rule, the evolutionary tree requires the optimal internal tree nodes (i.e. the optimal ancestral functional modules) such that it yields the smallest evolutionary distance of the tree.

Let T be the evolutionary tree that includes a set of leaves $L = \{P_1, P_2, \cdots, P_k\}$, internal nodes $I = \{P_{k+1}, P_{k+2}, \cdots, P_{k+m}\}$. We refer to $B \subset N \times N$ as all branches of T where $N = I \cup L$, and Γ as the collection of all possible I. We define $\mathcal{M}_{ij}$ as the node correspondence match of P_i and P_j. On the basis of the *maximum parsimony* rule, we reconstruct the set of internal nodes by solving an optimization problem $\min_{I \in \Gamma} \sum_{i,j} f(P_i, P_j, \mathcal{M}_{ij}) \delta_{ij}$, where $\delta_{ij} = 1$ iff $(P_i, P_j) \in B$ and $i, j \in \{1, 2, \cdots, m+k\}$.

3.3.2 Search for optimal internal tree nodes With a tree topology B and its leaves L, the computation of exhaustively searching for the optimal internal tree nodes I^* is numerically intractable. Hence, we use a meta-heuristic algorithm SA to detect optimal or near-optimal answers. Annealing is known as a thermal process for obtaining a minimum energy state of solid in a heat bath, which includes two steps: (i) raising the temperature to melt the solid metal; (ii) decreasing the temperature in a proper strategy so that the inner particles arrange themselves in a state of lower energy.

For each observed *d-subnet*, the SA approach starts with a series of non-interaction subnetworks as the initial internal tree nodes and specifies the initial temperature to its maximum (see the pseudocode in the *Procedure* S7). Let $\mathbf{x} = (e_0, e_1, e_2, \cdots)$ be a series of binary variables that represent the appearance of interactions in the inner nodes. Then, $\mathbf{x}$ can describe the current state of the evolutionary tree. For instance, the initial state can be written as $\mathbf{x}_0 = (0, 0, 0, \cdots)$ since the absence of interaction in all ancestral modules. Then, we use $\Theta(\mathbf{x})$ as our objective function $\sum_{i,j} f(P_i, P_j, \mathcal{M}_{ij}) \delta_{ij}$. In the following phase, we diminish the temperature and repeatedly perturb the current state $\mathbf{x}$ with a *Metropolis* scheme (Metropolis *et al.*, 1953) using $\pi_i \propto \exp(\Theta(\mathbf{x})/(sT_i))$ as the Boltzmann probability distribution (Kirkpatrick *et al.*, 1983). It is noted that it allows the alteration of only one interaction from one state $\mathbf{x}_j$ to its neighbor state $\mathbf{x}_{j+1}$ (i.e. $|\mathbf{x}_j - \mathbf{x}_{j+1}| = 1$). This process continues until the temperature T_i decreases to T_{min}. Eventually, all the internal nodes I^* are reconstructed according to the final solution $\mathbf{x}^*$. A more detailed workflow of this method is described in the Supplementary Material.

3.4 Search for high-scoring *d-subnets*

To search for high-scoring local alignments, it is necessary to find a suitable scoring scheme that assigns each *d-subnet* an alignment score. The alignment scores reflect the fit of *d-subnets* to functionally conserved modules.

3.4.1 Scoring function We introduce a scoring function that can foretell how likely a *d-subnet* could be functionally conserved modules. As mentioned before, each *d-subnet* can be put an evolutionary distance. However, it is not enough to calibrate *d-subnets* of various sizes because the evolutionary distance of a *d-subnet* tends to be linearly related to the number of *k-spines* within it. Supplementary Figure S3 gives the distance of 48 364 *d-subnets* sampled from our datasets. However, it is obvious that *functional modules* are not biased toward the one of a bigger size. So, we assigned each *d-subnet* an alignment score in the following way. Let $\mathcal{U}$ be a *d-subnet*, which includes a set of d *k-spines* and k induced subnetworks of the PPI networks. Regarding the k subnetworks as the leaves $L = \{P_1, P_2, \cdots, P_k\}$ of the evolutionary tree T, we set the scoring function for the *d-subnet* $\mathcal{U}$ as

$$\varphi(\mathcal{U}) = \frac{d}{\min_{I \in \Gamma} \sum_{i,j} f(P_i, P_j, \mathcal{M}_{ij}) \delta_{ij}}.$$

Hence, the score of each *d-subnet* is a positive value that indicates the fit of the observed *d-subnet* to a certain conserved functional module. The distribution of alignment scores for *d-subnets* sampled from our datasets is given in Supplementary Figures S4–S6.

3.4.2 Searching algorithm Using the scoring function, the problem of identifying conserved functional modules is reduced to the problem of searching for high-scoring *d-subnets*. However, the task of enumerating all *d-subnets* is computationally hard (Kalaev *et al.*, 2009) because the complexity of the fastest known algorithm is $O(n^{kd})$. Hence, we use a widely used heuristic approach *seed-and-extend* (Kalaev *et al.*, 2009; Sharan, 2005). Although the method does not reduce the complexity of enumeration, it practically reduces the computational time.

Let $G_H = \{V_H, E_H\}$ be a k-layer graph, where $V_H = \cup_{i=1}^{k} V_i$ and $E_H = \cup_{i,j}(E_{ij} \cup E_i)$ (see an example in Fig. 1). The pseudocode of our

Table 1. Proteins and interactions of our five observed species, which are collected from the databases of IntAct and Uniprot/Swiss-Prot

Species	Proteins	Interactions
H.sapiens	11 258	47 031
C.elegans	9302	15 669
D.melanogaster	8725	27 053
S.cerevisiae	5494	54 163
E.coli	2985	14 467

searching algorithm is described in Algorithm 1. To find a *k-spine* around a given node in G_H, we adapt an efficient subgraph searching algorithm ESU (Wernicke, 2006) to our implementation. In the ESU algorithm, each *k-spine* can be accessed by a starting node because its nodes are visited in a fixed order, but several *k-spines* might have a common starting node (see more in Procedure S1 and S3). As the beginning of our search, we collect all starting nodes (line 1). Then we search for a set of strongly connected small *d-subnets* as refined seeds (line 2). Subsequently, each refined seed is extended to a *d-subnet* of a size between *minSize* and *maxSize* by using local search (line 6–11). Lastly, the alignment score of each *d-subnet* is calculated by using *SA* (line 14), and these *d-subnets* scoring lower than a threshold value are filtered away (line 17–19). More detailed description of our algorithm is provided in the Supplementary Material (Procedure S1–S7).

Algorithm 1 Search for high-scoring *d-subnets*.

```
1: $startNodes[] \leftarrow collectStartNodes(G_H)$;
2: $refinedSeeds[] \leftarrow searchSeeds(G_H, \mathbb{N}, seedSize)$;
3: $m \leftarrow refinedSeeds.size()$;
4: $minSize \leftarrow seedSize + minExt$;
5: $maxSize \leftarrow seedSize + maxExt$;
6: for $i := 1$ to $m$ do
7:     $subnet \leftarrow expandSeed(G_H, \mathbb{N}, refinedSeeds[i], ext)$;
8:     if $subnet.size \geq minSize || subnet.size() \leq maxSize$ then
9:         $subnetList.push_{back}(subnet)$;
10:    end if
11: end for
12: $n \leftarrow subnetList.size()$;
13: for $i := 1$ to $n$ do
14:    $\mathbf{x}^* \leftarrow simulatedAnnealing(subnetList[i], T)$;
15:    $d \leftarrow subnetList[i].size()$;
16:    $score \leftarrow d/\Theta(\mathbf{x}^*)$;
17:    if $score > threshold$ then
18:        output $subnetList[i]$;
19:    end if
20: end for
```

4 RESULTS AND DISCUSSION

4.1 Test data

All experimentally determined interactions of five species were collected from the IntAct database (Kerrien *et al.*, 2012) as the test data of our evaluation (downloaded on February 10, 2014). The five species were *Homo sapiens* (hsa), *Caenorhabditis elegans* (cel), *Drosophila melanogaster* (dme), *Saccharomyces cerevisiae* (sce) and *Escherichia coli* (eco). The protein sequences were downloaded from a reviewed and manually annotated database, UniprotKB/Swiss-Prot (Magrane and Consortium, 2011). All-against-all protein sequence similarities were calculated with the program BLASTP (Altschul *et al.*, 1997), and these with $E\text{-}value \leq 1.0e^{-7}$ were selected as homologous proteins. The phylogenetic relationship of the five species was obtained from the NCBI taxonomy database (Federhen, 2012). With the real-world knowledge of the five species (Table 1), we performed LA and several existing algorithms on 26 real datasets including all possible combinations of the test species. To test the statistical significance of our alignment results, LA were also tested on 1040 random datasets (40 random k-layer graphs for each combination). All these random k-layer graphs remained the same number of interactions and edges as the real k-layer graphs. Moreover, high-quality associated Gene Ontology (GO) annotations which were downloaded from the *Uniprot-GOA* database (on March 14, 2014) and a reference dataset CORUM (Ruepp *et al.*, 2010) were used to help assess the biological quality of the results.

4.2 Experimental setup

We implemented LA in C + + using the *LEMON Graph Library* (Dezső *et al.*, 2011) version 1.2.3 and OpenMP (Chapman *et al.*, 2007). The implementation supports multicore parallelism in the search for high-scoring *d-subnets*. LA provides many user-specified parameters that are used to determine the topological feature of target regions and the scoring scheme, such as *seedSize*, *minExt*, *maxExt*, α and β. The default values are now $seedSize = 2$, $minExt = 3$, $maxExt = 13$, $\alpha = 0.2$ and $\beta = 2$. More elaborate information about the other specific parameters is described in Supplementary Table S1. We first performed LA 20 times with a single core, and then ran it 20 times again with 16 cores in parallel on each real dataset. The best, average and worst results were applied to assess the performance. *NBM* was subsequently performed on the same datasets with the extension scheme of *relaxed order*. In addition, three pairwise local alignment tools *NetworkBlast*, *AN* and *MW* were applied to all of our *2-way* alignments. However, another two multiple local alignment tools *Graemlin* and *CAPPI* were not taken into consideration in our assessment, as *Graemlin* did not compile successfully (the current available version is outdated), and *CAPPI* was only compatible with particularly designed data. For a fair comparison, the solutions that were highly overlapped (i.e. >0.5) were filtered out after the search for high-scoring *d-subnets* (see more in the Supplementary Material). All experiments mentioned in the following parts were carried out on the same machine, an Intel(R) Xeon(R) CPU X5550 with 2.67 GHz.

4.3 Cross-validation

We assessed the quality of the alignment results in four ways: coverage, consistency, prediction of protein functions and prediction of protein complexes. Coverage indicates the amount of input data the algorithm can explain. Consistency implies the functional coherence of identified *d-subnets*. Our goal is to find a series of *d-subnets* that have good consistency while reporting as many *d-subnets* as possible (i.e. a high coverage) within reasonable time. Consistency can be well accomplished by sacrificing coverage and vice versa. Further, to determine how much our alignment results agree with known biological knowledge,

Table 2. Coverage, consistency and running time on the two-way alignments

Dataset	Measure	*LA* (best)	*LA* (average)	*LA* (worst)	*NB*	*AN*	*NBM*	*MW*
A-B	Precision (%)	99.6	98.9	97.85	99.4	/	99.7	90
	Time(s) × 1	59.19	62.33	65.7	16 260	/	24	276
	Time(s) × 16	15.43	16.4	18.1	/	/	/	/
A-C	Precision (%)	100	97.7	93.7	96.1	/	100	96.6
	Time(s) × 1	54.8	56.84	59.06	36 750	/	55	889
	Time(s) × 16	16.34	16.97	17.61	/	/	/	/
A-D	Precision (%)	93.5	92.1	91	/	/	/	95.2
	Time(s) × 1	126	129.7	135.4	> 24 h	/	/	1587
	Time(s) × 16	32.04	33.38	34.61	/	/	/	/
A-E	Precision (%)	100	99.8	99.4	83.4	91.2	96.4	80.6
	Time(s) × 1	76.39	81.05	83.93	10	/	6	1
	Time(s) × 16	20.27	21.01	21.82	/	4	/	/
B-C	Precision (%)	99.1	98	96.15	87.5	76.9	99.2	73.7
	Time(s) × 1	30.57	31.63	32.71	17	/	5	4
	Time(s) × 16	9.652	10.41	11.12	/	11	/	/
B-D	Precision (%)	91.1	87.5	84.4	100	73.1	/	72.6
	Time(s) × 1	116.9	120.3	125	59	/	/	10
	Time(s) × 16	24.42	26.71	28.52	/	7	/	/
B-E	Precision (%)	97.3	94.4	90.35	/	100	100	100
	Time(s) × 1	50.8	53.22	54.67	4	/	1	0
	Time(s) × 16	11.5	11.95	13.44	/	1	/	/
C-D	Precision (%)	94.85	93.6	92.2	96.5	70.6	/	69.8
	Time(s) × 1	130.3	144.3	201.7	1558	/	/	40
	Time(s) × 16	30.33	32.38	35.69	/	43	/	/
C-E	Precision(%)	97.75	95.9	93.55	100	100	100	57.1
	Time(s) × 1	58.95	61.5	63.98	6	/	1	0
	Time(s) × 16	14.61	15.28	18.28	/	1	/	/
D-E	Precision (%)	97.75	94.8	91.5	88.1	84.6	/	51.8
	Time(s) × 1	57.74	59.43	60.63	91	/	/	4
	Time(s) × 16	20.63	21.16	21.81	/	20	/	/

Note: The five species *Human*, *Worm*, *Fruit Fly*, *Yeast* and *E.coli* are rereperesented by *A*, *B*, *C*, *D* and *E*. The five algorithms *LocalAli*, *NetworkBlast*, *AlignNemo*, *NetworkBlastM* and *MaWISh* are shortly represented by *LA*, *NB*, *AN*, *NBM* and *MW*, respectively.

LA was also applied to predict protein functions and protein complexes. Finally, we compared the performance of the alignment tools in terms of scalability and running time.

4.3.1 Coverage and consistency The coverage was measured in two ways. First, we measured it by the number of reported *d-subnets* (or hits) after the elimination of redundant solutions. Second, the coverage was measured by the *percentage of proteins value* (*PPV*), which calculated the percentage of proteins covered by the identified hits over all the proteins. We performed functional enrichment analyses based on GO annotation data (Ashburner *et al.*, 2000) to assess the functional coherency of each subnetworks in the reported hits. A powerful package *GO-TermFinder* (Boyle *et al.*, 2004) was used to calculate the statistical significance of GO annotations. Those subnetworks that had one or more enriched *GO* terms (i.e. corrected *P*-value ≤ 0.01) were regarded as FCS and likely to be functional modules. Therefore, we measured consistency by the number of reported FCS and the portion of FCS over all identified subnetworks (i.e. precision). All the results of the 26 real datasets including 10 two-way alignments, 10 three-way alignments, 5 four-way alignments and 1 five-way alignment were analyzed as shown in Tables 2 and 3 and Supplementary Table S3–S27.

In comparison with NB, NBM, AN and MW, LA basically outperformed all existing algorithms in the aspect of coverage. As shown in Supplementary Table S3, for instance, LA reported 477, 408 and 348 hits in the best, average and worst case in the two-way alignment of hsa-cel, while merely 367160 and 252 hits were reported by NB, NBM and MW. The worst PPV value of LA was also upto 10.8%, which was obviously more advanced than that of other algorithms. This was not a unique instance in the 10 two-way alignments as shown in Supplementary Tables S5–S11. However, LA reported less hits than NB, NBM, MW in the two-way alignment of hsa-dme. The reason was that the threshold of the alignment score was too high for this dataset. More than 90% d-subnets were filtered

Table 3. Coverage, consistency and running time on the 3-, 4- and 5-way alignments

Dataset	Measure	*LA* (best)	*LA* (average)	*LA* (worst)	*NBM*
A-B-C	Precision (%)	97.8	96.4	94.4	99.6
	Time(s) × 1	3143.79	3292.036	3390.03	227
	Time(s) × 16	265.702	284.66345	304.984	/
A-B-D	Precision (%)	94.8	93.9	93	/
	Time(s) × 1	1029.38	1055.189	1073.14	/
	Time(s) × 16	247.493	257.9092	270.716	/
A-B-E	Precision (%)	100	95	86.7	90.5
	Time(s) × 1	1047.39	1161.1855	1624	5
	Time(s) × 16	865.333	1119.06825	1272.58	/
A-C-D	Precision (%)	91.4	90.8	90.2	/
	Time(s) × 1	2200.25	2281.7855	2383.87	/
	Time(s) × 16	204.366	211.43815	219.9	/
A-C-E	Precision (%)	97.5	96	94.7	96
	Time(s) × 1	2405.05	2593.7125	2816.24	16
	Time(s) × 16	1350.6	1746.3	1958.26	/
A-D-E	Precision (%)	97.4	97.1	96.5	/
	Time(s) × 1	1677.12	1800.295	1924.56	/
	Time(s) × 16	271.142	295.1718	316.262	/
B-C-D	Precision (%)	96.9	95.9	94.9	/
	Time(s) × 1	931.271	1852.4552	2588.25	/
	Time(s) × 16	192.827	199.7882	210.676	/
B-C-E	Precision (%)	93.3	79.5	58.3	94.4
	Time(s) × 1	1087.89	1225.4395	1339.38	1
	Time(s) × 16	226.042	272.53595	303.416	/
B-D-E	Precision (%)	97.5	96.3	94.8	/
	Time(s) × 1	3277.73	3482.7515	3655.78	/
	Time(s) × 16	250.955	252.15815	275.12	/
C-D-E	Precision (%)	99.4	99.2	98.8	/
	Time(s) × 1	2316.76	2450.2115	2534.97	/
	Time(s) × 16	204.455	211.4621	218.722	/
A-B-C-D	Precision (%)	98.1	96.5	95	/
	Time(s) × 1	1389.4	1473.8075	1575.09	/
	Time(s) × 16	363.119	445.35695	514.922	/
A-B-C-E	Precision (%)	97.9	95.6	93.4	91.1
	Time(s) × 1	2576.42	2919.1695	3333.32	152
	Time(s) × 16	1721.29	2137.6375	2924.53	/
A-B-D-E	Precision (%)	95.6	93.4	90.3	/
	Time(s) × 1	506.98	579.72535	642.928	/
	Time(s) × 16	157.4	197.66365	259.277	/
A-C-D-E	Precision (%)	92.6	90.9	88.4	/
	Time(s) × 1	706.675	850.19905	954.841	/
	Time(s) × 16	261.356	375.35105	544.023	/
B-C-D-E	Precision (%)	99.8	98.5	95.8	/
	Time(s) × 1	760.734	884.5283	968.757	/
	Time(s) × 16	125.965	160.45365	204.812	/
A-B-C-D-E	Precision (%)	98.9	97.3	95.5	/
	Time(s) × 1	8457.52	10 344.525	12 424.8	/
	Time(s) × 16	3867.05	6535.725	9886.51	/

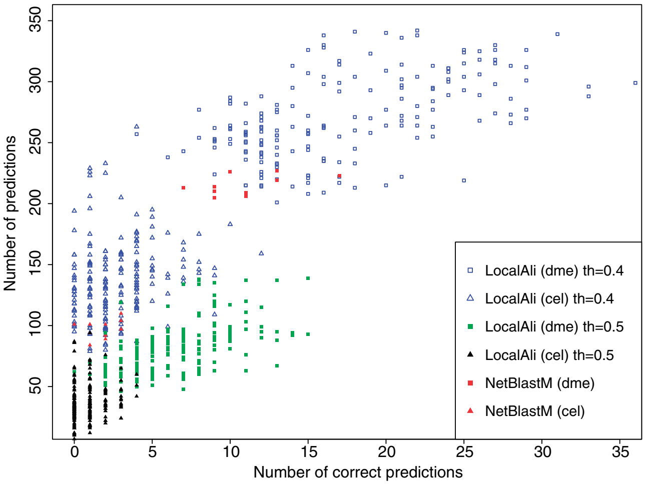

Fig. 4. Ten-fold cross-validation for function predictions on the cel-dme alignment using LA and *NetworkBlast-M* (NetBlastM). The parameter of threshold (th) is used to filter out *d-subnets* with a lower alignment score

away. Comparing with NBM in multiple alignments, LA also reported more hits and higher PPV in many cases such as the hsa-cel-dme alignment (Supplementary Table S13), the hsa-dme-eco (Supplementary Table S17) and the four-way alignment of hsa-cel-dme-eco (Supplementary Table S23). NBM failed to report any hit in many other multiple alignments such as the three-way alignment of hsa-dme-sce (Supplementary Table S16) and hsa-sce-eco (Supplementary Table S18) because of its limited scalability.

In the aspect of consistency, LA identified much more FCS than NB, NBM, AN, MW in both of the pairwise and multiple alignments, meanwhile retained a high precision. For instance, LA found 1628, 1535 and 1402 FCS in the best, average and worst case in the hsa-eco alignment (Supplementary Table S6), whereas only 5, 31, 81 and 79 were found by NB, AN, NBM and MW, respectively. Meanwhile, the worst success rate of identifying FCS was also upto 99.4%, which was higher than all other algorithms (the A–E alignment in Table 2). Similar results could be found in many other pairwise alignments such as the hsa-eco alignment (Supplementary Table S6), the cel-dme (Supplementary Table S7) and the cel-sce (Supplementary Table S8). In the alignment of multiple networks, it showed that LA had a competitive advantage in FCS over NBM, as well as a comparable precision. For example, it resulted 360 FCS in the worst case of the hsa-dme-eco alignment (Supplementary Table S17), which was five times as many as these reported by NBM. At the same time, it got the same average precision with NBM (the A-C-E alignment in Table 3). More importantly, LA successfully aligned many datasets, such as the hsa-cel-sce (Supplementary Table S14) and the hsa-dme-sce (Supplementary Table S16), in which NBM, however, reached its limitation. NBM nevertheless got a higher precision than LA in the cel-dme-eco alignment (Supplementary Table S20).

Moreover, we executed LA on random datasets of each possible combination of input species to verify the statistical significance of our results. As a result, we found all these data about hits, FCS and precision were non-random and statistically significant. As shown in Supplementary Figures S7–S16, the random results (blue triangles) were far away from the real-world data (red points). It indicated that these results of hits and FCS in real-world data were unlikely to happen in the random data. Further, we found most of the red points stand quite close to the oblique line, while the blue triangles were far away from the line. This evidence implies that precision is also statistically significant because the closer the points are, the higher precision they have. There is no figure illustrating multiple alignments of the random datasets, since LA can hardly find any *d-subnet* in multiple alignments of random datasets.

4.3.2 Prediction of protein functions Proteins that function in a pathway or structural complex are functionally related. It spontaneously leads us to the tentative functional assignments, which can be called by applying the method of *annotation transfer* (Sharan, 2005). Given a set of proteins, we predicted new protein functions whenever all the following four requirements were fulfilled: (i) the set of proteins was significantly enriched for a particular GO annotation (corrected P-value ≤ 0.01); (ii) at least three of the proteins were annotated with the GO annotation; (iii) the percentage of proteins annotated with this GO annotation over all characterized proteins was >0.5; (iv) the GO annotation was at a GO level of three or higher in the GO tree. All the remaining proteins will be considered to have the annotation if all the four demands are satisfied. If there are several GO annotations fulfilling the four requisites, just the one with the lowest corrected P-value will be applied for the prediction. According to the four requirements, all the cel-dme alignments that reported by NB, NBM, AN and LA were analyzed for predicting gene-associated ontology with the aspect of *biological process*. As a result, LA recognized 214.9 predictions of new GO annotations for proteins in *cel*, 286.2 predictions for proteins in *dme* in the average case. In contrast, NB reported 26 predictions in *cel*, 31 predictions in *dme*; AN found 18 in *cel*, 55 in *dme*; NBM found 165 in *cel*, 229 in *dme*.

To validate the quality of the predicted functions, we estimated the success rate of our predictions using a method of 10-fold cross-validation, in which we equally separated the annotation data into 10 parts, iteratively hid one part and used the remaining data to predict the held-out annotations (Sharan, 2005). The prediction will be considered correct if the protein has some true annotation that lies on a path in the GO tree from the root to a leaf that visits the predicted annotation. According to this rule, the number of correct predictions obtained from NBM and LA were illustrated in Figure 4 on the 10-fold cross-validation. The blue points of LA were much more than that of NBM in the figure since 20×10 samples are plotted. Then, we tried it again after increasing the threshold to 0.5 (th = 0.5) to verify whether our scoring scheme was indeed closer to the truth of biology. As indicated in the figure, LA was preferable to NBM in predicting the correct protein functions with th = 0.4 for both *cel* and *dme*, though it also made some false positive points (i.e. these tended to travel to the left upper corner) for *cel*. In the case of th = 0.5, it was more clear to see that LA had similar number of correct functions with NBM by using less number of predictions. The average success rates of NBM were 1.83 and 5.05% for *cel* and *dme*, respectively. They were less than that of LA with th = 0.4, which are 1.96 and

6.35%. They increased to 2.26 and 7.67% when th = 0.5. To sum up, we can conclude that LA, in comparison with NBM, is more precise in the prediction of functional annotations, and the higher-scoring *d-subnets* are more favorable for the prediction of protein functions.

4.3.3 Validation of predicted functional modules To validate the predicted functional modules, we collected a benchmark set of protein complexes that belonged to *hsa* as annotated in *CORUM* (Ruepp *et al.*, 2010; released in February 2012). Overall, there were 1283 protein complexes consisting of three or more proteins in our benchmark set. Then, we compared these identified conserved subnetworks with the benchmark set of complexes. Let S represent proteins of a conserved subnetwork, C be proteins of a known protein complex. We will consider S to be a successful prediction of C if and only if two requirements are fulfilled: (i) $|S \cap C| \geq 3$; (ii) $\frac{|S \cap C|}{\max \{|C|, |S|\}} \geq 0.2$. If S corresponds to a protein complex in *CORUM*, it will be a *pure module*. As a result, NBM successfully recognized 29 *pure modules* from the human PPI network with a success rate of 11.9%. In contrast, LA recognized 55.8 *pure modules* on average with a success rate of 17.4%. It indicates that LA is more accurate than NBM in recognizing biologically meaningful modules.

4.3.4 Scalability Scalability is a bottleneck problem that limits the applications of existing alignment tools. Many pairwise alignment tools attempting to search for strongly connected subgraphs in an alignment graph are difficult to extend to multiple networks because alignment nodes in the graphs will grow exponentially when the number of networks increases. In comparison with other algorithms in our tests, LA demonstrated the best performance in the aspect of scalability. It was the only algorithm that favorably ran on all the 26 datasets. However, NBM encountered its limitation when some network had a protein connected to a large number of other proteins, such as PPI networks of *sce* in Table 1.

4.3.5 Running time Parallelization is a key technique that enables LA to speed up. We first performed LA on each real dataset 20 times with a single core, and then ran it 20 times again with 16 cores in parallel. In comparison with NB, NBM, AN and MW, LA was the most favorable alignment tool in the pairwise alignments. As shown in Supplementary Tables S3–S12, LA finished all the pairwise alignments within several minutes ($\leq$3) using a single core. The parallelism yielded a speedup of LA. Generally, it could be three to six times faster in the pairwise alignments. In contrast, NB spent about 5 h on the hsa-cel alignment, 10 h on hsa-dme, >24 h on hsa-sce and 0.5 h on dme-sce. MW spent 15 min on hsa-dme, 26 min on hsa-sce. Although, NB, NBM, AN and MW were faster than LA in some alignment such as hsa-eco and cel-eco, they accomplished the advancement with a serious sacrifice of coverage. In the multiple alignment, NBM was faster than LA in many cases but with a smaller number of reported hits and a limited scalability (Supplementary Tables S13–S27).

5 CONCLUSION

In this article, we developed a fast and scalable local alignment tool LA to identify functionally conserved modules across multiple species. It overcomes several limitations of existing algorithms by using a scoring scheme strongly rooted in a *maximum-parsimony* evolutionary model, scaling to multiple networks with tens of thousands of proteins and interactions and parallel computing. By relying on this model, LA can provide an inference of the evolutionary history of observed functional modules with a series of evolutionary events. With a rigorously designed scoring function, we reduced the problem of identifying functionally conserved modules to a problem of searching for high-scoring *d-subnets*. LA solves the problem in three steps as follows: (i) it searches for a set of *d-subnets* with a heuristic approach *seed-and-extend*; (ii) it reconstructs the evolution history of each *d-subnets* and calculates its alignment score; (iii) it filters away these *d-subnets* with an alignment score below a threshold.

To compare LA with other existing algorithms, we tested these algorithm on 26 real-world datasets and analyzed their output in terms of several criteria. In a short conclusion, LA has a superiority of coverage, consistency and scalability over *NB*, *AN*, *NBM* and *MW*, meanwhile retains a high precision in identifying *functional coherent subnetworks*. Furthermore, it predicted >500 new functional annotations for proteins of *worm* and *fruit fly*, and identified 55 *pure modules*, which were known protein complexes that belonged to *human* as annotated in *CORUM*. It reported many significant functional modules that were missed by other alignment tools. The results demonstrate that LA provides substantial improvements to multiple local alignment and might give helpful suggestions to the research community that attempts to determine phylogeny, function annotations and functional modules.

ACKNOWLEDGEMENTS

We are grateful to Xiao Liang and René Rahn for proofreading the manuscript and their helpful suggestions.

Funding: This work is financially supported by the China Scholarship Council.

Conflict of interest: none declared.

REFERENCES

Aladağ,A.E. and Erten,C. (2013) Spinal: scalable protein interaction network alignment. *Bioinformatics*, **29**, 917–924.

Altschul,S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Ashburner,M. *et al.* (2000) Gene ontology: tool for the unification of biology. The gene ontology consortium. *Nat. Genet.*, **25**, 25–29.

Boyle,E.I. *et al.* (2004) Go::termfinderopen source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. *Bioinformatics*, **20**, 3710–3715.

Chapman,B. *et al.* (2007) *Using OpenMP: Portable Shared Memory Parallel Programming (Scientific and Engineering Computation)*. The MIT Press, Cambridge, Massachusetts, London, England.

Ciriello,G. *et al.* (2012) Alignnemo: a local network alignment method to integrate homology and topology. *PLoS One*, **7**, e38107.

Clark,C. and Kalita,J. (2014) A comparison of algorithms for the pairwise alignment of biological networks. *Bioinformatics*, **30**, 2351–2359.

Dezső,B. *et al.* (2011) LEMON–an open source C++ graph template library. *Electron. Notes Theor. Comput. Sci.*, **264**, 23–45; Proceedings of the Second Workshop on Generative Technologies (WGT) 2010.

Dutkowski,J. and Tiuryn,J. (2007) Identification of functional modules from conserved ancestral protein-protein interactions. *Bioinformatics*, **23**, i149–i158.

El-Kebir,M. *et al.* (2011) Lagrangian relaxation applied to sparse global network alignment. In: Loog,M. *et al.* (eds) *Pattern Recognition in Bioinformatics, volume 7036 of Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, pp. 225–236.

Federhen,S. (2012) The ncbi taxonomy database. *Nucleic Acids Res.*, **40**, D136–D143.

Felsenstein,J. (2003) *Inferring Phylogenies*. 2 edn. Sinauer Associates, Inc., Sunderland, Massachusetts.

Fitch,W.M. (1971) Toward defining the course of evolution: minimum change for a specific tree topology. *Syst. Biol.*, **20**, 406–416.

Flannick,J. *et al.* (2006) Graemlin: general and robust alignment of multiple large interaction networks. *Genome Res.*, **16**, 1169–1181.

Flannick,J. *et al.* (2009) Automatic parameter learning for multiple local network alignment. *J. Comput. Biol.*, **16**, 1001–1022.

Fraser,H.B. *et al.* (2002) Evolutionary rate in the protein interaction network. *Science*, **296**, 750–752.

Guzzi,P.H. and Mina,M. (2012) Alignmcl: Comparative analysis of protein interaction networks through markov clustering. In: *Proceedings of the 2012 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)*. BIBMW'12. IEEE Computer Society, Washington, DC, USA, pp. 174–181.

Hartwell,L.H. *et al.* (1999) From molecular to modular cell biology. *Nature*, **402** (**Suppl. 6761**), C47–C52.

Hu,J. *et al.* (2014) Netcoffee: a fast and accurate global alignment approach to identify functionally conserved proteins in multiple networks. *Bioinformatics*, **30**, 540–548.

Huang,Q. *et al.* (2013) Corbi: a new r package for biological network alignment and querying. *BMC Syst. Biol.*, **7** (**Suppl. 2**), S6.

Kalaev,M. *et al.* (2008) Networkblast: comparative analysis of protein networks. *Bioinformatics*, **24**, 594–596.

Kalaev,M. *et al.* (2009) Fast and accurate alignment of multiple protein networks. *J. Comput. Biol.*, **16**, 989–999.

Kelley,B.P. *et al.* (2004) PathBLAST: a tool for alignment of protein interaction networks. *Nucleic Acids Res.*, **32** (**Suppl. 2**), W83–W88.

Kerrien,S. *et al.* (2012) The intact molecular interaction database in 2012. *Nucleic Acids Res.*, **40**, D841–D846.

Kirkpatrick,S. *et al.* (1983) Optimization by simulated annealing. *Science*, **220**, 671–680.

Koyutürk,M. *et al.* (2006) Pairwise local alignment of protein interaction networks guided by models of evolution. *J. Comput. Biol.*, **13**, 182–199.

Kuchaiev,O. and Pržulj,N. (2011) Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics*, **27**, 1390–1396.

Li,Z. *et al.* (2007) Alignment of molecular networks by integer quadratic programming. *Bioinformatics*, **23**, 1631–1639.

Liao,C.-S. *et al.* (2009) IsoRankN: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, **25**, i253–i258.

Magrane,M. and Consortium,U. (2011) Uniprot knowledgebase: a hub of integrated protein data. *Database*, **2011**, bar009.

Metropolis,N. *et al.* (1953) Equation of state calculations by fast computing machines. *J. Chem. Phys.*, **21**, 1087–1092.

Milenković,T. *et al.* (2010) Optimal network alignment with graphlet degree vectors. *Cancer Inform.*, **9**, 121–137.

Milenković,T. *et al.* (2013) Global network alignment in the context of aging. In: *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*. BCB'13. ACM, New York, NY, pp. 23:23–23:32.

Neyshabur,B. *et al.* (2013) Netal: a new graph-based method for global alignment of proteinprotein interaction networks. *Bioinformatics*, **29**, 1654–1662.

Pache,R.A. *et al.* (2012) Netalignera network alignment server to compare complexes, pathways and whole interactomes. *Nucleic Acids Res.*, **40**, W157–W161.

Patro,R. and Kingsford,C. (2013) Predicting protein interactions via parsimonious network history inference. *Bioinformatics*, **29**, i237–i246.

Patro,R. *et al.* (2012) Parsimonious reconstruction of network evolution. *Algorithms Mol. Biol.*, **7**, 25.

Pellegrini,M. *et al.* (1999) Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc. Natl Acad. Sci. USA*, **96**, 4285–4288.

Phan,H.T.T. and Sternberg,M.J.E. (2012) Pinalog: a novel approach to align protein interaction networks implications for complex detection and function prediction. *Bioinformatics*, **28**, 1239–1245.

Roguev,A. *et al.* (2008) Conservation and rewiring of functional modules revealed by an epistasis map in fission yeast. *Science*, **322**, 405–410.

Ruepp,A. *et al.* (2010) Corum: the comprehensive resource of mammalian protein complexes-2009. *Nucleic Acids Res.*, **38** (**Suppl. 1**), D497–D501.

Sahraeian,S.M.E. and Yoon,B.-J. (2013) Smetana: accurate and scalable algorithm for probabilistic alignment of large-scale biological networks. *PLoS One*, **8**, e67995.

Saraph,V. and Milenković,T. (2014) Magna: Maximizing accuracy in global network alignment. *Bioinformatics*, **30**, 2931–2940.

Sharan,R. (2005) Conserved patterns of protein interaction in multiple species. *Proc. Natl Acad. Sci. USA*, **102**, 1974–1979.

Sharan,R. and Ideker,T. (2006) Modeling cellular machinery through biological network comparison. *Nat. Biotech.*, **24**, 427–433.

Singh,R. *et al.* (2007) Pairwise global alignment of protein interaction networks by matching neighborhood topology. In: *Proceedings of the 11th annual international conference on Research in computational molecular biology, RECOMB'07*. Springer-Verlag, Berlin, Heidelberg, pp. 16–31.

Singh,R. *et al.* (2008) Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proc. Natl Acad. Sci. USA*, **105**, 12763–12768.

Vazquez,A. *et al.* (2003) Modeling of protein interaction networks. *ComPlexUs*, **1**, 38–44.

Wagner,A. (2003) How the global structure of protein interaction networks evolves. *Proc. R. Soc. Lond. Ser. B Biol. Sci.*, **270**, 457–466.

Wernicke,S. (2006) Efficient detection of network motifs. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **3**, 347–359.

Zhang,X. and Moret,B. (2008) Boosting the performance of inference algorithms for transcriptional regulatory networks using a phylogenetic approach. In: Crandall,K. and Lagergren,J. (eds) *Algorithms in Bioinformatics, volume 5251 of Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, pp. 245–258.