BIOINFORMATICS

Vol. 25 ISMB 2009, pages i253–i258
doi:10.1093/bioinformatics/btp203

IsoRankN: spectral methods for global alignment of multiple protein networks

Chung-Shou Liao[1,2,3], Kanghao Lu[3,4], Michael Baym[3,4], Rohit Singh[3] and Bonnie Berger[3,4,*]

[1]Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, [2]Institute of Information Science, Academia Sinica, Nankang, Taipei 115, Taiwan, [3]Computer Science and Artificial Intelligence Laboratory and [4]Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

ABSTRACT

Motivation: With the increasing availability of large protein–protein interaction networks, the question of protein network alignment is becoming central to systems biology. Network alignment is further delineated into two sub-problems: local alignment, to find small conserved motifs across networks, and global alignment, which attempts to find a best mapping between all nodes of the two networks. In this article, our aim is to improve upon existing global alignment results. Better network alignment will enable, among other things, more accurate identification of functional orthologs across species.

Results: We introduce IsoRankN (IsoRank-Nibble) a global multiple-network alignment tool based on spectral clustering on the induced graph of pairwise alignment scores. IsoRankN outperforms existing algorithms for global network alignment in coverage and consistency on multiple alignments of the five available eukaryotic networks. Being based on spectral methods, IsoRankN is both error tolerant and computationally efficient.

Availability: Our software is available freely for non-commercial purposes on request from: http://isorank.csail.mit.edu/

Contact: bab@mit.edu

1 INTRODUCTION

Almost every biological process is mediated by a network of molecular interactions. A few examples of these include: genetic regulatory networks, signaling networks, metabolic networks and protein–protein interaction (PPI) networks. The structure of these networks is becoming increasingly well known, especially with the advent of high-throughput methods for network inference (Ito *et al*., 2001; Krogan *et al*., 2006; Uetz *et al*., 2000). As with the genome, there is significant conservation of network structure between organisms (Matthews *et al*., 2001; Yu *et al*., 2004). Thus, knowledge about the topology of a network in one organism can yield insights about not only the networks of similar organisms, but also the function of their components. A problem with accurate cross-species comparison of such networks is that the known networks, however, are both incomplete and inaccurate (Han *et al*., 2005; Huang *et al*., 2007).

The specific problem we address is that of global alignment of multiple PPI networks. A *PPI network* is an undirected collection of pairwise interactions on a set of proteins, where an edge represents interaction between two proteins. Given a pair of PPI networks, and a list of pairwise sequence similarities between proteins in the two networks, the pairwise alignment problem is to find an optimal mapping between the nodes of the two networks that best represents conserved biological function. We distinguish such global network alignment from local alignment where the goal is to find multiple network *motifs*, i.e. independent regions of localized network similarity. In the multiple global network alignment case, with k networks, the problem is extended to finding clusters of proteins across the networks such that these clusters best represent conserved biological function.

The search for such an alignment is motivated by the intuition that evolution of genes occurs within the context of the larger cellular system they are part of. Global network alignment can be interpreted as an evolutionary analysis done at this systems level rather than in a piecemeal, local fashion. Once a global network alignment has been estimated, we can analyze it to gather more localized, granular insights, e.g. estimating functional orthology across species.

Alignment of multiple networks poses two key problems. The first is that the computational complexity (i.e. the number of possible alignments) grows exponentially in the number of networks. The second is that the genomes corresponding to the various networks being aligned may vary widely in size (e.g. because of differing degrees of gene duplication). A multiple network alignment algorithm must thus efficiently identify a biologically appropriate mapping between the genes.

Here, we introduce IsoRankN (IsoRank-Nibble), which takes the approach of deriving pairwise alignment scores between every pair of networks, using the original IsoRank methodology (Singh *et al*., 2007, 2008, Box 1); then finds alignment clusters based on these scores. To find clusters, we use a spectral partitioning method that is both efficient and automatically adjusts to the wide variation in sizes of the species-specific networks. The algorithm is similar to the recently developed `PageRank-Nibble` algorithm (Anderson *et al*., 2006), which approximates the *Personalized PageRank* vector. A PageRank vector (i.e. one that describes a ranking of graph nodes for, say, search) is called a *Personalized PageRank* vector if, given a particular graph node, its preference scores are concentrated on a small set of vertices, the set being tailored to the given node. This notion of vertex-specific rankings is applied in IsoRankN to find dense, clique-like clusters of proteins when computing the global alignment of multiple PPI networks.

We tested IsoRankN on the five known eukaryotic PPI networks, i.e. human, mouse, fly, worm, and yeast. Much of the related previous work has focused on local network alignment; hence, a direct comparative evaluation of our results was difficult. As a

*To whom correspondence should be addressed.

© 2009 The Author(s)
This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/2.0/uk/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Downloaded from https://academic.oup.com/bioinformatics/article/25/12/i253/189039 by guest on 06 April 2024

gold standard alignment does not yet exist, we instead evaluate our alignment method on a variety of indirect criteria, including number of clusters predicted, within-cluster consistency and GO/KEGG enrichment (Ashburner *et al*., 2000; Kanehisa and Goto, 2000). In order to measure within-cluster consistency, we introduce a novel metric based on the entropy of the GO/KEGG annotations of predicted clusters. We believe that the characteristic of a correct global network alignment would be to preserve the relative functions of various network parts; this can be well-measured by the various GO enrichment analyses described above.

A number of related techniques for PPI network alignment exist. Most notably, these include NetworkBLAST-M (Kalaev *et al*., 2008), Græmlin 2.0 (Flannick *et al*., 2008) and IsoRank (Singh *et al*., 2008), though a number of other techniques exist as well (Berg and Lässig, 2006; Dutkowski and Tiuryn, 2007; Kelley *et al*., 2003, 2004; Koyuturk *et al*., 2005; Sharan *et al*., 2005; Srinivasan *et al*., 2006). NetworkBLAST-M computes a local alignment by greedily finding regions of high local conservation based on inferred phylogeny. Græmlin 2.0, in contrast, computes a global alignment by training how to infer networks from phylogenetic relationships on a known set of alignments, then optimizing the learned objective function on the set of all networks.

IsoRank uses spectral graph theory to first find pairwise alignment scores across all pairs of networks, the details of which are provided later (Box 1); these pairwise scores, computed by spectral clustering on the product graph, work well in capturing both the topological similarity as well sequence similarity between nodes of the networks. However, to find multiple network alignments, IsoRank uses these scores in a time-intensive greedy algorithm. Instead, IsoRankN uses a different method of spectral clustering on the induced graph of pairwise alignment scores. The new approach provides significant advantages not only over the original IsoRank but also over other methods.

To test IsoRankN, we show that on the PPI networks from five different eukaryotic species, IsoRankN produces an alignment with a larger number of aligned proteins, higher within-cluster consistency and higher biological similarity than existing methods, as measured by GO/KEGG enrichment using GO TermFinder (Boyle *et al*., 2004). While other techniques for measuring GO enrichment exist (Segal *et al*., 2004; Schlicker *et al*., 2006), they did not apply directly to the context in which we work. Additionally, IsoRankN does not require training and does not rely on induced phylogeny; thus it is not sensitive to errors in the phylogenetic tree. While this is not a significant problem with eukaryotes, inference of accurate bacterial phylogeny has proven far more difficult.

Contributions: We introduce the IsoRankN algorithm which uses an approach similar to the `PageRank-Nibble` algorithm to align multiple PPI networks. In so doing, we bring a novel spectral clustering method to the bioinformatics community. We use IsoRankN to align the known eukaryotic PPI networks and find that it efficiently produces higher fidelity alignments than existing global multiple-alignment algorithms.

2 METHODS

2.1 Functional similarity graph

The central idea of IsoRankN is to build a multiple network alignment by local partitioning of the graph of pairwise functional similarity scores.

Box 1. The Original IsoRank Algorithm.

IsoRank works on the principle that if two nodes of different networks are aligned, then their neighbors should be aligned as well. In lieu of sequence similarity information, the functional similarity score R_{ij} between vertex v_i and v_j is the set of positive scores which satisfies:

$$R_{ij} = \sum_{\substack{v_u \in N(v_i) \\ v_w \in N(v_j)}} \frac{1}{|N(v_u)||N(v_w)|} R_{uw},$$

where $N(v_i)$ is the neighborhood of v_i within its own network. This can also be viewed as the steady-state distribution of a random walk on the direct product of the two networks.
To integrate a vector of sequence homologies, E, IsoRank takes a parameterized average between the network-topological similarity and the known sequence homology. It uses the power method to find the unique positive R satisfying

$$R = \alpha AR + (1-\alpha)E, \text{ with } 0 \leq \alpha \leq 1,$$

where

$$A_{ij,uw} = \begin{cases} \frac{1}{|N(v_u)||N(v_w)|}, & v_u \in N(v_i), v_w \in N(v_j), \\ 0, & \text{otherwise.} \end{cases}$$

Given the resulting vector of pairwise functional similarity scores, R, a discrete network alignment is then greedily generated.

Specifically, given k PPI networks, $G_1, G_2, \ldots, G_k$, we first compute the functional similarity scores of every pair of cross-species proteins $(v_i, v_j) \in (G_l, G_m)$. This is done using the original IsoRank algorithm (Box 1), but without the final step of greedily selecting an alignment. The scores generated by IsoRank have the advantage of being highly noise tolerant, a result of using a spectral approach.

The result is a *functional similarity graph*, a weighted complete k-partite graph on the k sets of proteins, where each edge is weighted by its functional similarity score. If the PPI networks were complete and exact, the multiple alignment problem would simply be to find maximally weighted cliques. As the networks are not, we introduce the *star spread* method to find highly similar near cliques, which yields a multiple alignment. In addition, in contrast to the *seed-path extension* method used by NetworkBLAST-M, our method is similar to the *star aligned* approach in multiple sequence alignment introduced by Lipman *et al*. (1989) and CLUSTAL W (Thompson *et al*., 1994).

2.2 Star spread

We first compute, for every protein v in a chosen species, every neighbor connected to v by an edge with weight greater than a threshold; this is the *star*, S_v of the protein (Fig. 1a). We greedily order the proteins v by the total weight of S_v and for each find the subset $S_v^* \subset S_v$ such that S_v^* is a highly weighted neighborhood of v (Fig. 1b). This is done using a spectral local graph partitioning algorithm with approximate *Personalized PageRank vectors*, similar to the `PageRank-Nibble` algorithm. The resulting S_v^* represents a *functionally conserved interaction cluster*, a set of network-aligned proteins. This is repeated for every protein in all species not already assigned to an S_v^*, yielding assignments for all vertices. While it is not clear exactly how the order of vertex choice affects the results, this ordering performs better empirically than others we have tried, including random ordering. The ordering of species is discussed below.

Downloaded from https://academic.oup.com/bioinformatics/article/25/12/i253/189039 by guest on 06 April 2024

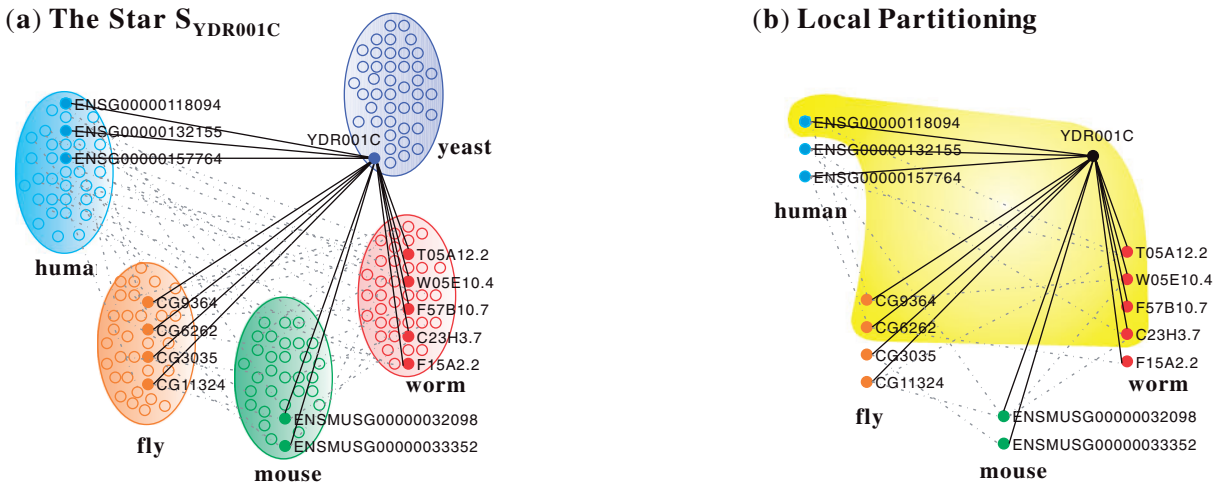

Fig. 1. An example of star spread on the five known eukaryotic networks. (**a**) S_{YDR001C}, the set of all neighbors of YDR001C with a similarity bounded by a threshold $\beta = 0.01$. The illustration emphasizes the key idea of star spread, that the neighborhood of a single protein, YDR001C, has many high-weight neighbors in other networks, each of which are connected to others with varying weights. As the data are noisy, we seek a highly weighted subset of this neighborhood, as opposed to a clique. (**b**) The shaded area is the resulting conserved interaction cluster S^*_{YDR001C}, containing YDR001C, as generated by our local graph partition algorithm.

2.3 Spectral partitioning

The main algorithmic challenge in obtaining functionally conserved interaction clusters S_v^* is uncertainty introduced by the incomplete and inaccurate PPI network data. Thus instead of finding a maximally weighted clique containing v, we find a low-conductance set containing v.

The *conductance*, $\Phi(S)$, of a subset S of a graph G is the ratio of the size of the edge cut to separate S to the number of edges in the larger of the two remaining sets, providing a very natural measure of 'clusterness' of a subset of vertices. Formally, $\Phi(S) = \frac{\sigma(S)}{\min\{\text{vol}(S), 2m - \text{vol}(S)\}}$, where $\sigma(S) = |\{(v_x, v_y); v_x \in S, v_y \notin S\}|$, $\text{vol}(S) = \sum_i \deg(v_i)$, and m is the number of edges in G.

Anderson *et al.* (2006) showed that a low-conductance set containing v can be computed efficiently via the personalized PageRank vector of v. A *personalized PageRank vector* $Pr(\gamma, v)$ is the stationary distribution of the random walk on S_v in which at every step, with probability γ, the walk 'teleports' back to v and otherwise performs a lazy random walk with transition probabilities proportional to R, the vector of pairwise interaction scores (i.e. with probability 1/2, the walk does not move). Thus in this case, a personalized PageRank vector is the unique solution to:

$$Pr(\gamma, v) = \gamma \chi_v + (1-\gamma) Pr(\gamma, v) W, \tag{1}$$

where $\gamma \in (0, 1]$, $\chi_v(x) = \delta_{x,v}$ is the indicator vector of v, $W = \frac{1}{2}(I + D^{-1}R)$ is the lazy random walk transition matrix and D is the diagonal of column sums of R. For the purposes of this article, we instead use an efficient approximation $p :\approx Pr(\gamma, v)$, the details of which can be found in (Anderson *et al.*, 2006).

To compute the minimal conductance cut, we consider the sets $T_j^p = \left\{ v_i \middle| \frac{p(v_i)}{\sum_k R_{ik}} \geq \frac{p(v_j)}{\sum_k R_{jk}} \right\}$, or those vertices which contain at least as much of the mass of p, normalized by R. As in (Anderson *et al.*, 2006), we then find the set S_v^* as:

$$S_v^* = \min_j \Phi\left(T_j^p\right). \tag{2}$$

2.4 Star merging

While highly efficient, the star spread method has the limitation of not assigning other members of the original network to the neighborhood S_v, and so S_v^* by necessity does not contain any other proteins in the same network as v, even if it is appropriate to do so. To get around this, we introduce a procedure for merging stars, by looking at the neighbors of the neighbors of v. For two stars, $S_{v_1}^*$ and $S_{v_2}^*$, where v_1 and v_2 are in the same PPI network, if every member of $S_{v_1}^* \setminus \{v_1\}$ has v_2 as a neighbor and vice versa, we merge $S_{v_1}^*$ and $S_{v_2}^*$.

2.5 The IsoRankN algorithm

Given k PPI networks $G_1, G_2, \ldots, G_k$, and a threshold β, IsoRankN proceeds as follows:

(1) Run the original IsoRank on every pair of networks to obtain scores R_{ij} on all edges of the functional similarity graph.

(2) For every protein v, compute the star $S_v = \left\{ v_j \in N(v) | w(v, v_j) \geq \beta \max_j \left(w(v, v_j) \right) \right\}$, where $N(v)$ is the neighborhood of v in the functional similarity graph.

(3) Pick an arbitrary remaining PPI network G_ℓ and order the proteins $v \in G_\ell$ by the sum of edge weights in the induced graph on S_v. In order, excluding proteins already assigned to clusters, spectrally partition S_v to obtain S_v^*.

(4) Merge every pair of clusters $S_{v_1}^*$ and $S_{v_2}^*$ in which $\forall v_i \in S_{v_2}^* \setminus \{v_2\}, w(v_1, v_i) \geq \beta \max_j \left(w(v_1, v_j) \right)$ and $\forall v_j \in S_{v_1}^* \setminus \{v_1\}, w(v_2, v_j) \geq \beta \max_j \left(w(v_2, v_j) \right)$.

(5) Repeat steps 3 and 4 until all proteins are assigned to a cluster.

3 RESULTS

Experimental datasets: We tested IsoRankN on five eukaryotic PPI networks: *Homo sapiens* (human), *Mus musculus* (mouse), *Drosophila melanogaster* (fly), *Caenorhabditis elegans* (worm) and *Saccharomyces cerevisiae* (Yeast). IsoRankN requires two forms of data as input: PPI networks and sequence similarity scores. The PPI networks were constructed by combining data from the DIP (Xenarious *et al.*, 2002), BioGRID (Stark *et al.*, 2006) and HPRD (Mishra *et al.*, 2006) databases. In total, these five networks contained 87 737 proteins and 98 945 known interactions. The sequence similarity scores of pairs of proteins were the BLAST

Downloaded from https://academic.oup.com/bioinformatics/article/25/12/i253/189039 by guest on 06 April 2024

Table 1. Comparative consistency on the five eukaryotic networks

	IsoRankN	IsoRank	Græmlin$_{1K}$	Græmlin$_{2K}$	NetworkBLAST-M
Mean entropy	**0.274**	0.685	0.857	0.552	0.907
Mean normalized entropy	**0.179**	0.359	0.451	0.357	0.554
Exact cluster ratio[a]	**0.380** (**3079** of 8095)	0.253 (2166 of 8539)	0.306 (843 of 2754)	0.355 (1135 of 3198)	0.291 (441 of 1518)
Exact protein ratio[b]	**0.261** (**9284** of 35 604)	0.165 (6408 of 38 706)	0.159 (2393 of 15 047)	0.248 (2906 of 11 729)	0.142 (1150 of 8092)

Mean entropy and mean normalized entropy of predicted clusters. Note that the boldface numbers represent the best performance with respect to each measure.
[a]The fraction of predicted clusters which are *exact*., i.e. all contained proteins have the same KEGG or GO group ID.
[b]The fraction of proteins in exact clusters.

Bit-values of the sequences as retrieved from Ensembl (Hubbard *et al*., 2007). We evaluated the biological relevance of our results against two gene ontology databases, GO (Ashburner *et al*., 2000) and KEGG (Kanehisa and Goto, 2000). For this article, we set $\alpha = 0.6$ and $\beta = 0.01$, and used human, mouse, fly, worm and yeast as the order of species that are at the center of the star spread. We further investigated other species permutations as discussed later.

Testing: In the results that follow, we have aimed to evaluate our method along two key dimensions: coverage and consistency. Coverage is the set of genes for which our algorithm makes non-trivial predictions. It is thus a proxy for sensitivity; a higher coverage would be desirable in that it suggests our algorithm can explain a larger amount of data. The other dimension, consistency, measures the functional uniformity of genes in each cluster. The intuition here is that each cluster should correspond to a set of genes with the same function; higher consistency is better. This measure serves as a proxy for the specificity of our method.

There currently exists no gold standard for network alignment quality, so in order to evaluate the predictions of IsoRankN we tested two properties of its predictions that we expect an optimal prediction to have. First, we tested within-cluster consistency of GO/KEGG annotation on the reasoning that predicted orthologs in an orthology should likely have similar function. Second, we tested coverage, on the reasoning that an ideal alignment should assign most proteins to a cluster. As local alignment may have ambiguous, inconsistent or overlapping clusters, we primarily compare IsoRankN to IsoRank and Græmlin 2.0. We also compare to local aligners (such as NetworkBLAST-M), however, these will have lower coverage as they only consider conserved modules.

3.1 Functional assignment

We tested IsoRankN as compared with IsoRank, Græmlin 2.0 and NetworkBLAST-M on the five available eukaryotic networks and found that it outperformed the other methods in terms of number of clusters predicted, within-cluster consistency and GO/KEGG enrichment.

Græmlin 2.0 requires a training set to learn the parameters of its scoring function. As in Flannick *et al*. (2008), we train Græmlin 2.0 on training sets of multiple sizes. The versions of Græmlin 2.0 trained on 1000 and 2000 KEGG clusters are denoted Græmlin$_{1K}$ and Græmlin$_{2K}$, respectively. We additionally attempted to train Græmlin 2.0 on 4000 clusters, but have not included the data, as it showed strong evidence of over-fitting.

Consistency: We first measured the consistency of the predicted network alignment by computing the mean entropy of the predicted clusters. The entropy of a given cluster S_v^* is:

$$H(S_v^*) = H(p_1, p_2, \ldots, p_d) = -\sum_{i=1}^{d} p_i \log p_i, \tag{3}$$

where p_i is the fraction of S_v^* with GO or KEGG group ID i. We also computed the mean entropy normalized by cluster size; i.e. $\bar{H}(S_v^*) = \frac{1}{\log d} H(S_v^*)$. Thus, a cluster has lower entropy if its GO and KEGG annotations are more within-cluster consistent. While a cluster with one element would have entropy 0, this is to be expected, as such a cluster is perfectly consistent with itself.

IsoRankN's predicted clusters have much lower entropy than IsoRank, Græmlin 2.0 and NetworkBLAST-M (Table 1). That is, the clusters obtained by IsoRankN have higher consistency of annotation. For the purpose of this measure, proteins without a GO or KEGG group ID were withheld.

We additionally measure as in Flannick *et al*. (2008) the fraction of clusters which are *exact*, i.e. those in which all proteins have the same GO or KEGG ID. For GO annotation, we restrict to the deepest categories, removing questions of multiplicity and specificity of annotations. We find that IsoRankN predicts significantly more exact clusters than existing techniques, and that a higher fraction of the predicted clusters are exact (Table 1). We note that only 60–70% of the proteins in any of the aligned networks have an assigned GO or KEGG ID, comparable to the fraction of all known proteins included in GO or KEGG. Additionally, the relative performance under either consistency measure does not change when restricted to GO or KEGG individually.

Coverage: We first measure coverage by the number of clusters containing proteins from k species. We find that for $k \geq 3$, IsoRankN predicts more clusters with more proteins (Table 2) than other methods. Thus, as it has higher consistency, it is likely that IsoRankN is detecting more distant multiple network homology. For $k = 2$, IsoRank has greater coverage; however, this is likely due to IsoRankN having a strict threshold for edge inclusion. Note that as a result of the star spread approach, all clusters obtained by IsoRankN contain at least two species. Thus IsoRankN does not find paralogs within a species without there existing at least one homolog in another species. Of the 87 737 total proteins, IsoRankN is able to find network homologs for 48 978 (55.8%), more than any technique but IsoRank. When restricted to clusters containing at least three species, i.e. the multiple alignment case, IsoRankN predicts the most clusters.

Downloaded from https://academic.oup.com/bioinformatics/article/25/12/i253/189039 by guest on 06 April 2024

Table 2. Number of clusters/proteins predicted containing exactly k species

Number of species (k)	IsoRankN	IsoRank	Græmlin$_{1K}$	Græmlin$_{2K}$
1	–/–[a]	155/402	1418 /**4001**	**1521**/2910
2	3844/8739	**6499/20 580**	1354/ 4650	2034/5899
3	**4022/13 533**	3036/13 391	947/5414	1116/5072
4	**2926/13 991**	2446/15 422	529/5371	310/2067
5	**2056/12 715**	773/9744	58/1467	11/78
Total	12 848/48 978	**12 909/59 539**	4306/20 903	4992/16 026

The k-th row contains, for each program, the number of predicted clusters for covering exactly k species and number of constituent proteins in those clusters. Note that the boldface numbers represent the best performance with respect to each row. NetworkBLAST-M is not included, as it always outputs $k=5$ species in each cluster.
[a]All clusters obtained by IsoRankN contain at least two species.

Table 3. Comparative GO/KEGG enrichment performance

Species	IsoRankN	IsoRank	Græmlin$_{1K}$	Græmlin$_{2K}$	NB-M[a]
Total	**712/2490**	537/1760	296/772	432/1010	107/261
p-value[b]	**1.28 e-90**	1.31 e-68	5.47 e-38	6.87 e-54	2.19 e-14
Human	**632/2200**	478/1551	194/545	272/811	66/182
Mouse	**605/2124**	383/1371	191/538	268/794	65/178
Fly	**574/1787**	398/924	208/533	261/771	41/135
Worm	**552/1698**	376/901	104/257	140/389	32/124
Yeast	**368/938**	257/554	208/486	137/316	45/136

The number of GO/KEGG categories enriched by each method. Note that the boldface numbers represent the best performance w.r.t. each row.
[a]NetworkBLAST-M is denoted NB-M for convenience.
[b]As computed by GO TermFinder. We remark that this excludes those proteins tagged IEA (inferred from electronic annotation).

We further measure as in Kalaev *et al*. (2008) coverage by the enrichment of predicted groups with respect to known ontology as derived from GO and KEGG. We find that IsoRankN enriches more GO and KEGG categories in every species, with a lower overall p-value [computed by GO TermFinder Boyle *et al*. (2004)], than any other technique (Table 3).

Ordering: While we chose a particular order of genomes in the multiple alignment to report our general results, we also include results on different orderings of genomes and demonstrate that any ordering outperforms other methods (Fig. 2). The particular order of genomes used above was chosen to have the minimum mean normalized entropy.

While it may appear that yeast, as the best annotated network, should be the first network chosen in the star spread, it is sufficiently dissimilar to the other species as to cause inaccurate network alignments on such a small set of species.

Running time: Given the weighted similarity graph, the star spread component of IsoRankN (Section 2.5, steps 2–5) took under 5 min for the five eukaryotic networks above. The computation of the graph, given by the original IsoRank (Section 2.5, step 1), took ∼7 h on a single processor, though can be easily 10-way parallelized.

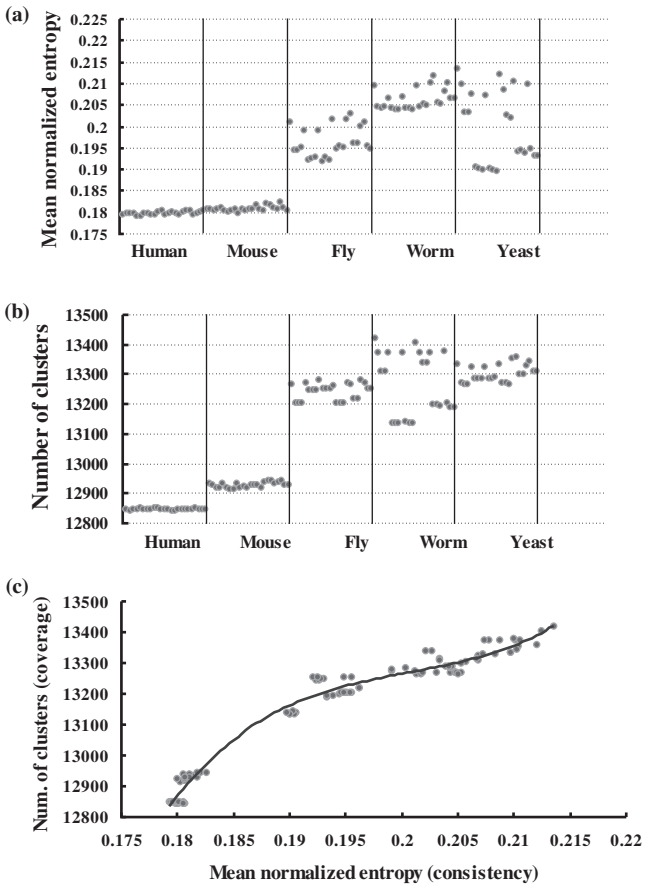

Fig. 2. The consistency and coverage performance of IsoRankN under species permutations in the star spread. Each dot represents one of the 120 possible permutations of the five species. (**a**) and (**b**) Report the consistency and coverage of the network fit as a function of the species first at the center of the star spread. (**c**) The relationship between mean normalized entropy and number of clusters.

All computations were run on a 64 bit 2.4 GHz Linux system with 2GB RAM.

4 CONCLUSION

In this article, we present an efficient method for computing multiple PPI network alignments. Based on spectral clustering on the induced graph of pairwise alignment scores, our program IsoRankN automatically handles noisy and incomplete input data. Our method differs from others in that it does not require training or phylogeny data and seeks vertex-specific rankings in the spectral clustering.

We demonstrate the effectiveness of this technique on the five available eukaryotic PPI networks. Our results suggest that IsoRankN has higher coverage and consistency compared to existing approaches, which should lead to improved functional ortholog prediction.

In future work, we plan to more fully explore and evaluate the database of functional orthologs as predicted by IsoRankN. Additionally, it may be possible to modify the star spread to account for existent gold standard network homology data, yielding even higher fidelity multiple network alignments.

Downloaded from https://academic.oup.com/bioinformatics/article/25/12/i253/189039 by guest on 06 April 2024

ACKNOWLEDGEMENTS

We thank Leonid Chindelevitch, Jon Kelner, and Michael Schnall-Levin for helpful comments.

Funding: National Science Council (Taiwan) (NSC-096-2917-I-002-114 and NSC-095-2221-E-001-016-MY3 to C.-S. L.). Fannie and John Hertz Foundation (to M.B.).

Conflict of Interest: none declared.

REFERENCES

Anderson,R. *et al*. (2006) Local graph partitioning using pagerank vectors. *Foundations of Computer Science*, IEEE Computer Society, Los Alamitos, CA, USA, pp. 475–486.

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

Berg,J. and Lässig,M. (2006) Cross-species analysis of biological networks by Bayesian alignment. *Proc. Natl Acad. Sci. USA*, **103**, 10967–10972.

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

Dutkowski,J. and Tiuryn,J. (2007) Identification of functional modules from conserved ancestral protein-protein interactions. *Bioinformatics*, **23**, 149–158.

Flannick,J. *et al*. (2008) Automatic parameter learning for multiple network alignment. In *Research in Computational Molecular Biology*. Vol. 4955, *Lecture Notes in Computer Science*, Springer, Berlin/Heidelberg, pp. 214–231.

Han,J.-D. *et al*. (2005) Effect of sampling on topology predictions of protein-protein interaction networks. *Nat. Biotech.*, **23**, 839–844.

Huang,H. *et al*. (2007) Where have all the ineractions gone? Estimating the coverage of two-hybrid protein interaction maps. *PLoS Comput. Biol.*, **3**, 2155–2174.

Hubbard,T.J.P. *et al*. (2007) Ensembl 2007. *Nucleic Acids Res.*, **35**, 610–617.

Ito,T. *et al*. (2001) A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl Acad. Sci. USA*, **98**, 4569–4574.

Kelley,B.P. *et al*. (2003) Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proc. Natl Acad. Sci. USA*, **100**, 11394–11399.

Kelley,B.P. *et al*. (2004) Pathblast: a tool for alignment of protein interaction networks. *Nucleic Acids Res.*, **32**, 83–88.

Krogan,N.J. *et al*. (2006) Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. *Nature*, **440**, 4412–4415.

Koyuturk,M. *et al*. (2005) Pairwise local alignment of protein interaction networks guided by models of evolution. In *Research in Computational Molecular Biology*. Vol. 3500, *Lecture Notes in Computer Science*, Springer, Berlin/Heidelberg, pp. 48–65.

Lipman,D.J. *et al*. (1989) A tool for multiple sequence alignment. *Proc. Natl Acad. Sci. USA*, **86**, 4412–4415.

Kalaev,M. *et al*. (2008) Fast and accurate alignment of multiple protein networks. In *Research in Computational Molecular Biology*. Vol. 4955, *Lecture Notes in Computer Science*, Springer, Berlin/Heidelberg, pp. 246–256.

Kanehisa,M. and Goto,S. (2000) KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, **28**, 27–30.

Matthews,L.R. *et al*. (2001) Identification of potential interaction networks using sequence-based searches for conserved protein-protein interactions or interologs. *Genome Res.*, **11**, 2120–2126.

Mishra,G.R. *et al*. (2006) Human protein reference database–2006 update. *Nucleic Acids Res.*, **34**, 411–414.

Schlicker,A. *et al*. (2006) A new measure for functional similarity of gene products based on Gene Ontology. *BMC Bioinformatics*, **7**, 1–16.

Segal,E. *et al*. (2004) GeneXPress: a visualization and statistical analysis tool for gene expression and sequence data. In *Proceedings of the 11th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pp. 1–3.

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

Singh,R. *et al*. (2007) Pairwise global alignment of protein interaction networks by matching neighborhood topology. In *Research in Computational Molecular Biology*. Vol. 4453, *Lecture Notes in Computer Science*, Springer, 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.

Srinivasan,B.S. *et al*. (2006) Integrated protein interaction networks for 11 microbes. In *Research in Computational Molecular Biology*. Vol. 3909, *Lecture Notes in Computer Science*, Springer, Berlin/Heidelberg, pp. 1–14.

Stark,C. *et al*. (2006) BioGRID: a general repository for interaction datasets. *Nucleic Acids Res.*, **34**, 535–539.

Thompson,J.D. *et al*. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673–4680.

Uetz,P. *et al*. (2000) A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, **403**, 623–627.

Xenarious,I. *et al*. (2002) DIP, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res.*, **30**, 303–305.

Yu,H. *et al*. (2004) Annotation transfer between genomes: protein protein interologs and protein DNA regulogs. *Genome Res.*, **14**, 1107–1118.

Downloaded from https://academic.oup.com/bioinformatics/article/25/12/i253/189039 by guest on 06 April 2024