IsoRankN: spectral methods for global alignment of multiple protein networks

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>As Published</td>
<td><a href="http://dx.doi.org/10.1093/bioinformatics/btp203">http://dx.doi.org/10.1093/bioinformatics/btp203</a></td>
</tr>
<tr>
<td>Publisher</td>
<td>Oxford University Press</td>
</tr>
<tr>
<td>Version</td>
<td>Final published version</td>
</tr>
<tr>
<td>Accessed</td>
<td>Mon Dec 03 08:32:53 EST 2018</td>
</tr>
<tr>
<td>Citable Link</td>
<td><a href="http://hdl.handle.net/1721.1/70980">http://hdl.handle.net/1721.1/70980</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>Creative Commons Attribution Non-Commercial</td>
</tr>
<tr>
<td>Detailed Terms</td>
<td><a href="http://creativecommons.org/licenses/by-nc/2.5">http://creativecommons.org/licenses/by-nc/2.5</a></td>
</tr>
</tbody>
</table>
Almost every biological process is mediated by a network of multiple PPI networks. The advent of high-throughput methods for network inference (Ito et al., 2002) 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.

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.

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., 2002; 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

© 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.
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 (Kalav et al., 2008), Grammlin 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; Koyutürk 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. Grammlin 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:

\[
R_{ij} = \sum_{v_k \in N(v_j)} \frac{1}{|N(v_i)||N(v_j)|} R_{kw},
\]

where \( N(v_j) \) is the neighborhood of \( v_j \) 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

\[
R = aAR + (1-a)E, \quad 0 \leq a \leq 1,
\]

where

\[
A_{vw} = \frac{1}{\text{number of common paths between } v \text{ and } w},
\]

\( v \in N(v), w \in N(v), \) otherwise.

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 \((v_i, v_j) \in (G_i, G_j)\). 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 \( S'_v \subseteq S_v \) 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.
2.3 Spectral partitioning

The main algorithmic challenge in obtaining functionally conserved interaction clusters 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, \( \Phi(S)=\frac{vol(S)}{\min(\{\frac{1}{2}\sum_{v \in N(v)} \sigma(v),\sum_{v \in G} \deg(v)\})} \), where \( \sigma(S)=\{\{v_1,v_2\}|v_1,v_2 \in S,v_1 \neq v_2\} \), \( vol(S)=\sum_{v \in S} \deg(v) \), 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 \( Pr(y,v) \) is the stationary distribution of the random walk on \( G \), in which at every step, with probability \( y \), the walk ‘teleports’ back to \( v \) and otherwise performs a lazy random walk with transition probabilities proportional to \( R \). Thus in this case, a personalized PageRank vector is the unique solution to:

\[
Pr(y,v) = y \cdot 1 + (1-y) Pr(y,v)W,
\]

where \( y \in (0,1), \; Pr(1,v) = \delta_v \), 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 \( \rho = Pr(y,v) \), the details of which can be found in (Anderson et al., 2006).

To compute the minimal conductance cut, we consider the sets \( T^v=\{v\;|\;\sum_{\{w\} \subseteq S^v} \sigma(w) \geq \frac{\max_{w \neq v} \{\sigma(w)\}}{2}\} \), or those vertices which contain at least as much of the mass of \( v \), normalized by \( R \). As in (Anderson et al., 2006), we then find the set \( S^v_1 \) as:

\[
S^v_1 = \min \Phi(T^v).
\]

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 \( S^v_1 \), and so \( S^v_1 \) 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 \( S^{v'}_1 \), 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 \( S^{v'}_1 \).

2.5 The IsoRankN algorithm

Given \( k \) PPI networks \( G_1, G_2, \ldots, G_k \), and a threshold \( \beta \), 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 \( S^v = \{\{v_1,v_2\}|N(v) (v_1,v_2) \geq \text{max}_{w \neq v} \{N(v),w(v_1,v_2)\}\} \), where \( N(v) \) is the neighborhood of \( v \) in the functional similarity graph.
3. Pick an arbitrary remaining PPI network \( G_i \) and order the proteins \( v \in G_i \) 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 \( S_1^v \).
4. Merge every pair of clusters \( S^v_1 \) and \( S^{v'}_1 \) in which \( \forall v_1 \in S^v_1 \setminus \{v_1\},w(v_1,v_2) \geq \text{max}_{v \neq v_1,v_2} \{w(v_1,v_2)\} \) and \( \forall v_2 \in S^{v'}_1 \setminus \{v_1\},w(v_1,v_2) \geq \text{max}_{v \neq v_1,v_2} \{w(v_1,v_2)\} \).
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 ( Xenarios 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
In the results that follow, we have aimed to evaluate Table 1. they only consider conserved modules. NetworkBLAST-M), however, these will have lower coverage as overlapping clusters, we primarily compare IsoRankN to IsoRank a cluster. As local alignment may have ambiguous, inconsistent or the reasoning that an ideal alignment should assign most proteins to have. First, we tested within-cluster consistency of GO/KEGG is that each cluster should correspond to a set of genes with the functional uniformity of genes in each cluster. The intuition here 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 Gramlin 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, Gramlin 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.

Gramlin 2.0 requires a training set to learn the parameters of its scoring function. As in Flannick et al. (2008), we train Gramlin 2.0 on training sets of multiple sizes. The versions of Gramlin 2.0 trained on 1000 and 2000 KEGG clusters are denoted Gramlin1K and Gramlin2K, respectively. We additionally attempted to train Gramlin 2.0 on 4000 clusters, but have not included the data, as it showed strong evidence of over-fitting.

<table>
<thead>
<tr>
<th></th>
<th>IsoRankN</th>
<th>IsoRank</th>
<th>Gramlin1K</th>
<th>Gramlin2K</th>
<th>NetworkBLAST-M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean entropy</td>
<td>0.274</td>
<td>0.685</td>
<td>0.857</td>
<td>0.552</td>
<td>0.907</td>
</tr>
<tr>
<td>Mean normalized entropy</td>
<td>0.179</td>
<td>0.359</td>
<td>0.451</td>
<td>0.357</td>
<td>0.544</td>
</tr>
<tr>
<td>Exact cluster ratio$^a$</td>
<td>0.380 (3079 of 8095)</td>
<td>0.253 (2166 of 8539)</td>
<td>0.306 (843 of 2754)</td>
<td>0.355 (1115 of 3198)</td>
<td>0.291 (441 of 1518)</td>
</tr>
<tr>
<td>Exact protein ratio$^b$</td>
<td>0.261 (9284 of 35 604)</td>
<td>0.165 (6408 of 38 706)</td>
<td>0.159 (2393 of 15 047)</td>
<td>0.248 (2906 of 11 729)</td>
<td>0.142 (1150 of 8092)</td>
</tr>
</tbody>
</table>

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.

**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:

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

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. $\frac{1}{S_v^*} H(S_v^*)$. 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, Gramlin 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.
Table 2. Number of clusters/proteins predicted containing exactly \( k \) species

<table>
<thead>
<tr>
<th>Species</th>
<th>IsoRankN</th>
<th>IsoRank</th>
<th>Græmlin1 ( K )</th>
<th>Græmlin2 ( K )</th>
<th>NB-M ( a )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>632/2200</td>
<td>478/1551</td>
<td>194/545</td>
<td>572/881</td>
<td>66/182</td>
</tr>
<tr>
<td>Mice</td>
<td>605/592</td>
<td>383/1371</td>
<td>191/538</td>
<td>268/794</td>
<td>65/178</td>
</tr>
<tr>
<td>Fly</td>
<td>574/1787</td>
<td>398/924</td>
<td>208/533</td>
<td>261/771</td>
<td>41/135</td>
</tr>
<tr>
<td>Worm</td>
<td>552/1698</td>
<td>376/901</td>
<td>104/257</td>
<td>140/389</td>
<td>32/124</td>
</tr>
<tr>
<td>Yeast</td>
<td>368/938</td>
<td>257/554</td>
<td>208/486</td>
<td>137/316</td>
<td>45/136</td>
</tr>
</tbody>
</table>

The number of GO/KEGG categories enriched by each method. Note that the boldface numbers represent the best performance with respect to each row.

\( a \)NetworkBLAST-M is not included, as it always outputs \( k = 5 \) species in each cluster.

\( b \)All clusters obtained by IsoRankN contain at least two species.

Table 3. Comparative GO/KEGG enrichment performance

<table>
<thead>
<tr>
<th>Species</th>
<th>IsoRankN</th>
<th>IsoRank</th>
<th>Græmlin1 ( K )</th>
<th>Græmlin2 ( K )</th>
<th>NB-M ( a )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>537/1760</td>
<td>296/772</td>
<td>432/1010</td>
<td>107/261</td>
<td></td>
</tr>
<tr>
<td>Mice</td>
<td>383/1371</td>
<td>191/538</td>
<td>268/794</td>
<td>65/178</td>
<td></td>
</tr>
<tr>
<td>Fly</td>
<td>398/924</td>
<td>208/533</td>
<td>261/771</td>
<td>41/135</td>
<td></td>
</tr>
<tr>
<td>Worm</td>
<td>376/901</td>
<td>104/257</td>
<td>140/389</td>
<td>32/124</td>
<td></td>
</tr>
<tr>
<td>Yeast</td>
<td>257/554</td>
<td>208/486</td>
<td>137/316</td>
<td>45/136</td>
<td></td>
</tr>
</tbody>
</table>

The \( p \)-value \( b \) computed by GO TermFinder. We remark that this excludes those proteins tagged IEA (refer to 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 multiplet 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.
Acknowledgements

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


Conflicts of Interest: none declared.

References


