Spectral methods for Global Network Alignment

Sushmita Roy
sroy@biostat.wisc.edu
Computational Network Biology
Biostatistics & Medical Informatics 826
https://compnetbiocourse.discovery.wisc.edu

Nov 13th 2018
Global Network Alignment stated formally

• Finding the optimal global alignment between two or more PPI networks, aims to find a correspondence between nodes and edges of the input networks that maximizes the overall “match” between the networks
• Every node in one network must be mapped to another node in the other network or marked as a gap
• That is, we want to find a single best mapping covering all nodes in the graph
• Furthermore, for >2 species, this alignment must be transitive
  – If $a_1$ is mapped to $a_2$ in species 2, and $a_2$ is mapped to $a_3$ and $a_3'$ in species 3, $a_1$ must be mapped to $a_3$ and $a_3'$. 
Motivation of the IsoRank algorithm

• Previous approaches have used a local and pairwise alignment approaches
• Limit the possible node mappings between species and then bring in networks to do the alignment
  – “..Lacks the flexibility of producing node-pairings that diverge from sequence-only predictions.”
IsoRank overview

• An algorithm for inferring the global alignment of more than two networks
• Unlike existing algorithms which use sequence similarity first to define the mapping, IsoRank simultaneously uses both the network and the sequence similarity to define node mappings
• Key intuition: a protein in one network is a good match to a protein in another network if it is similar in sequence and its network neighborhood
• Such proteins are said to be “functionally similar” to each other across species
• The IsoRank algorithm uses eigenvalue problem to estimate the functional similarity score
Notation

- $G_k=(V_k,E_k)$ is a graph of $|V_k|$ vertices and $|E_k|$ edges for species $k$.
- $G_k$ corresponds to a protein-protein interaction (PPI) network for a species $k$.
- Edges can be weighted: $w(e)$ denotes weight of an edge $e$.
- For a node $i$ in $V_k$, $N(i)$ denotes the neighbors of $i$ in $G_k$.
- For two graphs $G_1$ and $G_2$, $R$ is a $|V_1|$ by $|V_2|$ matrix where each entry $R_{ij}$ specifies the functional similarity score of protein node $j$ in $G_1$ and node $j$ in $G_2$. 
Two Key steps IsoRank algorithm

• Estimate the functional similarity score $R$ that is based on network and sequence similarity for all pairs of networks
• Use $R$ to define node mappings and to identify subgraphs that represents the conserved parts of the network
Pairwise Global Network Alignment with IsoRank

• Let us first consider the simple case of aligning two graphs, $G_1$ and $G_2$

• IsoRank has two steps
  – Estimate the functional similarity score $R_{ij}$ that is based on network and sequence similarity of proteins $i$ in $V_1$ and $j$ in $V_2$
  – Use $R$ to define node mappings and to identify subgraphs that represents the conserved parts of the network
    • This uses a greedy approach that starts with a seed from a bi-partite graph and grows it until no more edges can be added
Defining the functional similarity $R_{ij}$

- We will first consider the simple case of estimating this from the networks alone.
- Assume the networks are unweighted.
- $R_{ij}$ is computed for every pair of nodes $i$ and $j$ where $i$ is in $V_1$ and $j$ in $V_2$.

$$R_{ij} = \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{1}{|N(u)||N(v)|} R_{uv}$$

- $R_{ij}$ should capture a similarity based on $i$ and $j$’s neighborhoods in $G_1$ and $G_2$ respectively.
Defining the functional similarity $R_{ij}$ for weighted graphs

- Let $w(i,u)$ denote the weight of edge $(i,u)$ in $G_1$, $0 \leq w(i,u) \leq 1$
- Let $w(j,v)$ denote the weight of edge $(j,v)$ in $G_1$
- Here $R_{ij}$ is defined as

$$R_{ij} = \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{w(i,u)w(j,v)}{\sum_{p \in N(u)} w(u,p) \sum_{q \in N(v)} w(v,q)} R_{uv}$$

Instead of the size of the neighborhood, we use a weighted sum over all nodes in the neighborhood of $u$
Computing $R_{ij}$ with two 5 node networks

$N(a) = \{b\}$
$N(a') = \{b'\}$

$$R_{aa'} = \frac{1}{|N(b)||N(b')|} R_{bb'}$$

$$R_{aa'} = \frac{1}{4} R_{bb'}$$
Computing $R_{ij}$ with two 5 node networks

$N(b)=\{a,c\}$
$N(b')=\{a',c'\}$

$$R_{bb'} = \sum_{u \in N(b)} \sum_{v \in N(b')} \frac{1}{|N(u)||N(v)|} R_{uv}$$

$$R_{bb'} = \frac{1}{|N(a)||N(a')|} R_{aa'} + \frac{1}{|N(a)||N(c')|} R_{ac'} + \frac{1}{|N(c)||N(a')|} R_{ca'} + \frac{1}{|N(c)||N(c')|} R_{cc'}$$

$$R_{bb'} = \frac{1}{1} R_{aa'} + \frac{1}{3} R_{ac'} + \frac{1}{3} R_{ca'} + \frac{1}{9} R_{cc'}$$
Similarly for the other node pairs

\[ R_{a'a'} = \frac{1}{4} R_{bb'} \]
\[ R_{bb'} = \frac{1}{3} R_{ac'} + \frac{1}{3} R_{a'c} + R_{aa'} + \frac{1}{9} R_{cc'} \]
\[ R_{cc'} = \frac{1}{4} R_{bb'} + \frac{1}{2} R_{be'} + \frac{1}{2} R_{bd'} + \frac{1}{2} R_{eb'} + \frac{1}{2} R_{db'} + R_{ee'} + R_{ed'} + R_{de'} + R_{dd'} \]

Only a partial set of scores are shown

<table>
<thead>
<tr>
<th></th>
<th>a'</th>
<th>b'</th>
<th>c'</th>
<th>d'</th>
<th>e'</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0.0312</td>
<td></td>
<td>0.0937</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>0.1250</td>
<td>0.0625</td>
<td>0.062</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>0.0937</td>
<td></td>
<td>0.2812</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>0.0625</td>
<td>0.0312</td>
<td>0.031</td>
<td></td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>0.0625</td>
<td>0.0312</td>
<td>0.031</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

G₁

G₂

R
Rewriting $R$ in matrix form

- Let $A$ be a matrix with $|V_1| \times |V_2|$ rows and $|V_1| \times |V_2|$ columns.
- Each row of $A[i,j]$ corresponds to a pair of nodes $(i,j)$ where $i$ is from $V_1$ and $j$ is from $V_2$.
- Each column of $A$ corresponds to a pair of nodes $(u,v)$ where $u$ is from $V_1$ and $v$ is from $V_2$.

\[
A[i, j][u, v] = \begin{cases} 
\frac{1}{|N(u)||N(v)|}, & \text{if } (i, u) \in E_1, (j, v) \in E_2 \\
0 & \text{otherwise}
\end{cases}
\]

- In matrix form we have

\[
R = AR
\]

- Thus $R$ is the eigen vector of $A$, with eigen value 1.
$R$ is the eigen vector of a specific matrix

\[
A[i, j][u, v] = \begin{cases} 
\frac{1}{|N(u)||N(v)|}, & \text{if } (i, u) \in E_1, (j, v) \in E_2 \\
0 & \text{otherwise}
\end{cases}
\]

$$A = \begin{bmatrix}
0 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.5 & 0 & 0.5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0 \\
0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 \\
0 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.5 & 0 & 0.5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0
\end{bmatrix}$$
$R$ is the eigen vector of $A$

\[ R_{bb'} = \frac{1}{|N(a)||N(a')|} R_{aa'} + \frac{1}{|N(a)||N(c')|} R_{ac'} + \frac{1}{|N(c)||N(a')|} R_{ca'} + \frac{1}{|N(c)||N(c')|} R_{cc'} \]

\[ R_{bb'} = R_{aa'} + R_{ac'} + R_{ca'} + R_{cc'} \]
$R_{bb'}$ from the matrix form

- Let’s check if we get the same answers

<table>
<thead>
<tr>
<th></th>
<th>a a'</th>
<th>a b'</th>
<th>a c'</th>
<th>b a'</th>
<th>b b'</th>
<th>b c'</th>
<th>c a'</th>
<th>c b'</th>
<th>c c'</th>
</tr>
</thead>
<tbody>
<tr>
<td>a a'</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>a b'</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>a c'</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>b a'</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>b b'</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>b c'</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>c a'</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c b'</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c c'</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

$R_{new} = R_{old}^{aa'} + R_{old}^{ac'} + R_{old}^{ca'} + R_{old}^{cc'}$

$R_{bb'}^{new} = R_{aa'}^{old} + R_{ab'}^{old} + R_{ac'}^{old} + R_{bb'}^{old}$
Estimating $R$

- This is an eigen value problem
- $A$ is a stochastic matrix (columns of $A$ sum to 1)
- $R$ can be found using the *power method*
- The power method repeatedly updates $R$ at iteration $k+1$ as follows

$$R(k+1) = AR(k) / |AR(k)|$$
Adding sequence similarity to $R$

- We can integrate other types of information in the same framework
- Let $B$ be a $|V_1| \times |V_2|$ matrix where each entry $B_{ij}$ is the sequence similarity of $i$ and $j$
- Let $E$ be the normalized version of $B$, $E=B/B\|B\|$
- $R$ can be redefined as
  \[ R = \alpha AR + (1 - \alpha)E \]
- This too can be solved with an eigen value problem
  \[ R = \alpha AR + (1 - \alpha)E1^T R \]
  $1^T$ is a vector of all ones
  \[ R = (\alpha A + (1 - \alpha)E1^T) R \]
Multiple GNA

- Extension to more than two networks is straightforward
- For every pair of graphs $G_m, G_n$, estimate $R^{mn}$
Two Key steps IsoRank algorithm

• Estimate the functional similarity score $R$ that is based on network and sequence similarity for all pairs of networks

• Use $R$ to define node mappings and to identify subgraphs that represent the aligned parts of the network
Extracting the aligned parts

• Once $R$ is estimated for all pairs of networks, we need to extract out node pairs with the highest values
  – 99% of the entries of $R$ will be zero

• Two ways to do this
  – One to One mapping
    • For each node, map it to at most one other node
    • Computationally efficient but ignores gene duplications
  – Many to Many mapping
Many to many mapping

- The goal here is to extract groups with multiple genes from the same species
- Each group represents an functional similarity between genes of one species to genes of another species
- Each set has the following property
  - Each gene in the set has high pairwise $R$ scores with most other genes in the set
  - there are no genes outside each set with this property
  - there are a limited number of genes from each species
- Identified via greedy algorithm
Greedy algorithm for finding aligned parts

- Construct a $k$-partite graph.
  - Each part $k$ has nodes from each species
  - Allow nodes to interact between different parts
- Extract a high confidence edge expand to connected neighbors
Greedy algorithm to find many-many node mappings

- Input k-partite graph $H, b_1, b_2, r$
- Repeat until no more edges are in $H$
  - Select an edge with the highest score $(i, j)$, where $i$ is $G_1$ and $j$ is in $G_2$ to initialize a match-set
  - Grow $(i, j)$ to create the primary match-set. This is the max k-partite matching
  - Primary match set is a set of nodes with at most 1 node from a species using $b_1$ to control the similarity
    - For all other graphs $G_3..G_k$, add a node $l$ if two conditions hold
      1. $R_{il}$ and $R_{jl}$ are the highest scores between $l$ and any node in $G_1$ and $G_2$, respectively and,
      2. the scores $R_{il} \geq b_1 R_{ij}$, and $R_{jl} \geq b_1 R_{ij}$,
  - Add upto $(r-1)$ nodes $v$ based on $b_2$, such that there exists $u, w$ in the primary set and $R_{vw} \geq b_2 R_{uw}$
  - Remove this set from $H$
Results

• Global alignment of multiple protein-protein interaction networks
  – Yeast, human, fly, mouse, worm
• Assess functional coherence of predicted functional orthologs
Alignment results of five PPI networks

- Common subgraph has 1,663 edges supported by at least 2 networks and 157 edges by at least 3
- Very few edges from all species
  - It is possible that the networks are too noisy and incomplete
- But this is much better than a pure sequence only mapping
  - 509 edges would be identified in two or more species with 40 in three species
IsoRank framework is robust to noisy data

Experiments done on a PPI network of 200 nodes. Randomized graphs obtained by swapping pairs of edges.
Yeast-fly GNA exhibit subgraphs of different topologies
IsoRank predictions of functional orthology

• The output of IsoRank can be used to define “functional orthologs” (FO)
• Of the 86,932 proteins from the five species, 59,539 (68.5%) of the proteins were matched to at least one protein in another species (i.e., had at least one FO).
• In contrast, sequence orthology maps only 38.5% of the proteins.
How good are functional orthologs?

• Use functional coherence measure
  – Obtain sets of orthologous proteins (each set is made up of proteins from different species) and select sets with the majority (80%) of the proteins with a GO annotation
  – For each such set P,
    • Collect all GO terms associated with the proteins in P.
    • Compute a similarity between each pair of GO terms based on the similarity of the gene content of each term (this is the Jaccard coefficient of the annotated proteins)
    • Take a median of all pairs of similarity
  – Functional coherence for the input ortholog list is the mean of the coherence per set
Functional orthologs from IsoRank are comparable to sequence based orthology

- Functional coherence for IsoRank: 0.22
- Functional coherence for Homologene: 0.223
- Functional coherence for InParanoid: 0.206
Summary

• IsoRank is a global alignment algorithm
• How does it differ from PATHBLAST?
  – Identifies different types of subnetworks
  – Uses a global alignment
  – Applicable to multiple networks
• How is it similar Sharan 2004?
  – Search and score of subnetworks is done similarly