IsoRank
CMSC 858L

Local alignment:

1. Which nodes are dissimilar [low sim($u, v$)] but have similar neighbors / neighborhoods? (e.g. Bandyopadhyay et al.)

**functional orthologs:** proteins that play the same role, but may look very different.

2. Which edges are real and important, e.g. form a conserved pathway in the cell?

Global alignment:

Singh et al., 2007 propose:

**Maximum common subgraph:** Find the largest graph $H$ that is isomorphic to subgraphs of two given graphs $G_1$ and $G_2$. 
Maximum Common Subgraph

**Input:** weighted graphs $G_1$ and $G_2$ with weights between 0 and 1.

**Output:**
- Maximum Common Subgraph: largest subgraph $B$ that is isomorphic to a subgraph of $G_1$ and $G_2$.
- **Mapping** of nodes between $G_1$ and $G_2$ s.t. each node is mapped to $\leq 1$ other node.
Maximum Common Subgraph

**Intuition:** mapping $i \leftrightarrow j$ is good if the neighbors of $i$ can be mapped to the neighbors of $j$:

Define: $R_{ij}$ as the “quality” of mapping $i \leftrightarrow j$:

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

Over all pairings in the between the neighbors of $i$ and $j$. $R_{uv}$ has 1 unit to give, and it spreads it evenly over its $|N(u)||N(v)|$ neighbors.
Example

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

(Figure from Singh, Xu, Berger, 2007)
The Weighted Cases

**Unweighted case:**

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

**Weighted case:**

\[ R_{ij} := \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{w(i, u)w(j, v)}{W(u)W(v)} R_{uv} \]

where

\[ W(u) = \sum_{x \in N(u)} w(x, u) \quad \leftarrow \text{“weighted degree”} \]
Matrix Form

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

Want to find the \( R_{ij} \) values. Gather into matrix:

\[
R = AR
\]

where

\[
A[i, j][u, v] = \frac{1}{|N(u)||N(v)|} \quad \text{if} \quad (i, u) \in G_1 \text{ and } (j, v) \in G_2
\]

\( n_1n_2 \times n_1n_2 \) matrix.

10^8 by 10^8 for the yeast-fly alignment, but sparse.
Finding $R$:

Want an $R$ vector such that: $R = AR$

$R$ is an eigenvector of $A$. 
A Random Walk View

\[ R = AR \]

Think of A as an adjacency matrix of a graph G:

\[ V = \{ij \text{ with } i \in G_1 \text{ and } j \in G_2\} \]

\[ E = \{(ij, uv) : (i,j) \in G_1 \text{ and } (u,v) \in G_2\} \]

Then vector R is a stationary distribution for a random walk on G.
Accounting For Sequencing Similarity

$B_{ij} = \text{Sequence similarity between } i \text{ and } j$

**Normalize:** $E = B / |B|$

**New problem:** weights neighbors and similarity with parameter $\alpha$:

$$R = \alpha AR + (1 - \alpha)E$$

When $\alpha$ is 1, only network used; when $\alpha = 0$ only sequence information is used.

Convert this to the format $R' = A' R'$:

$$\begin{bmatrix} R \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha A & (1 - \alpha)E \\ 0 \cdots 0 & 1 \end{bmatrix} \begin{bmatrix} R \\ 1 \end{bmatrix}$$
Finding the Mapping, Given R

Method 1: maximum matching

Method 2: greedy

\[ F = \emptyset \]

Repeat:

Output highest weight pair \((p,q)\) such that \(p,q \notin F\)

\[ F = \{p,q\} \cup F \]
Fly vs. Yeast

Networks had > 25,000 edges each.

Largest component (35 edges) of fly-yeast alignment →

Complete alignment had 1420 edges, split into many components.

(Figures from Singh, Xu, Berger, 2007)
200 node subgraph of yeast and several randomized versions of it. Map random versions to the real one. Including even a tiny bit of sequence information improves the performance greatly.

(Figure from Singh, Xu, Berger, 2007)
Choosing $\alpha$:

Chose the $\alpha (=0.6)$ that matched the most Inparanoid database entries.
Vs. PathBLAST:

• Of IsoRank’s 701 aligned pairs, 83% were seen in at least 1 local alignment of PB.

• PB aligns the same protein to many different proteins: If aligned, a yeast protein is aligned to an average of 5.38 fly proteins.

• E.g. PathBLAST maps SNF1 to 71 different fly proteins.
Summary

• Global alignment guarantees consistent mapping of nodes.

• Values ($R_{ij}$) for each pair of nodes modeling the goodness of that mapping. (Can these $R_{ij}$ values be used for something else?)

• Via eigenvector, seek “equilibrium” values for the $R_{ij}$.

• Then select a high-weight, consistent subset of those pairs to form the mapping. (Is there a better algorithm than the greedy?)
Graemlin: General and robust alignment of multiple large interaction networks

Flannick, Novak, Srinivasan, McAdams, Batzoglou, Genome Res. 2006.
The 4 Big Ideas of Graemlin

1. Nodes scores via “likelihood” of common evolutionary history

2. Edge scores based on “edge-scoring matrices”

3. Seeded alignment based on good matches between a small number of nodes (and greedily extended)

4. Progressive alignment to align multiple sequences
Graemlin: Aligning Multiple Networks

Multiple Sequence Alignment:

Multiple Network Alignment:
Graemlin: Aligning Multiple Networks

Multiple Sequence Alignment:

Multiple Network Alignment:

← One difference: a species can have multiple nodes in each “column”

Just as with MSA, require items in the same column to be homologous
Scoring Alignments: “Column” Scores

Estimate evolutionary history:
- protein duplication
- protein divergence
- protein creation (insertion)
- protein loss (deletion)

Score(u) = \frac{\log S(M, u)}{\log S(R, u)}

Parameters for events taken from real data

• sequence similarity: sum of pairs of sequence distances

Parameters for events taken from random data
For every pair of proteins that are in the same species but different equivalent classes:

\[
\text{Score}(w) := \log \frac{\Pr_M[w - \delta < x < w + \delta]}{\Pr_R[w - \delta < x < w + \delta]}
\]

\((w-\delta = 0 \text{ and } w+\delta = \text{lowest possible edge score if there is no edge.})\)

\[w(u,v) := w\]

Equivalence classes are assigned labels from the ESM.

(Figure from Flannick et al.)
Alignments: Seeding

d-cluster: is a node u and its d-1 closest neighbors.

For each node, generate its d-cluster.

For every pair of d-clusters, compute the best alignment exhaustively.

Toss out all d-cluster alignments that score below some threshold T.

The highest-scoring pairs in the remaining d-cluster alignments become seeds around which they will attempt to grow an alignment.
**Greedy Growing**

(Figure from Flannick et al, 2006)

**Frontier**: nodes that are neighbors of nodes in the current alignment.

**Repeat**: Add the node or a pair of nodes from the frontier to the alignment that will increase the score the most.
Graemlin: Summary

- Pairwise alignment that accounts for
  - how likely a “column” is to have arisen by evolution
  - edge scores that specifies the broad topology desired

- Multiple Alignment
  - achieved via “progressive pairwise alignments”

- Sped up via
  - seeds to find good initial matches.