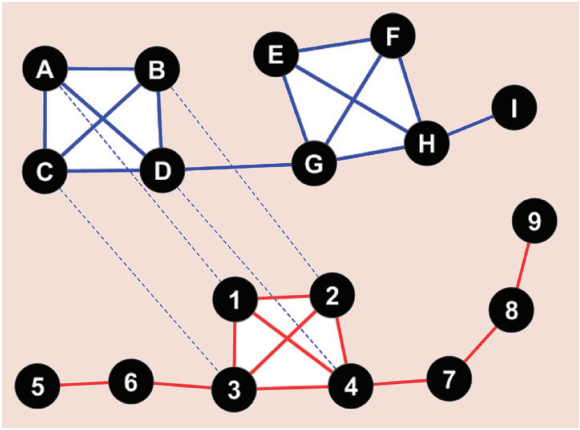
Non negative matrix factorization for Global Network Alignment

Sushmita Roy
sroy@biostat.wisc.edu

Computational Network Biology
Biostatistics & Medical Informatics 826
https://compnetbiocourse.discovery.wisc.edu

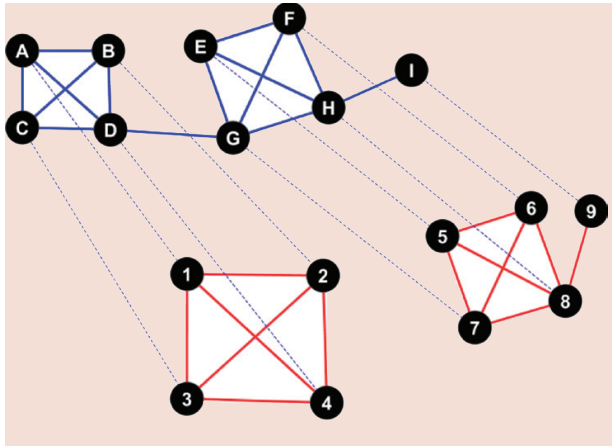
Nov 15th 2018

RECAP: Different network alignment problems



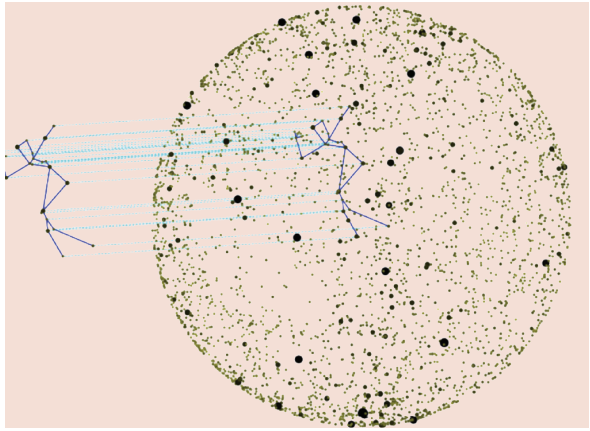
Local Network alignment: Find locally similar subnetworks

E.g. PATHBLAST, LocalAli, Sharan et al 2004



Global network alignment: Align all nodes in one network to all nodes in the second network

E.g. IsoRank, FUSE



Network query: Find instances of a small subnetwork in a larger network

Nir Atias and Roded Sharan, May 2012, ACM Communications

Algorithms for global network alignment

- IsoRank:
  - R. Singh, J. Xu, and B. Berger, "Global alignment of multiple protein interaction networks with application to functional orthology detection," *Proceedings of the National Academy of Sciences*, vol. 105, no. 35, pp. 12 763-12 768, Sep. 2008. [Online]. Available: http://dx.doi.org/10.1073/pnas.0806627105
- FUSE:
  - V. Gligorijević, N. Malod-Dognin, and N. Pržulj, "Fuse: multiple network alignment via data fusion," *Bioinformatics*, vol. 32, no. 8, pp. 1195-1203, Apr. 2016. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/btv731

IsoRank RECAP

- 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

Motivation of FUSE

- How to do multiple global network alignment?
- In existing approaches the sequence-based node mapping is local, that is one pair at a time.
- Can we improve this mapping by using protein-protein interaction networks in each species?

FUSE multiple network alignment

- Given
  - Protein-protein interaction networks for k species
  - Pairwise sequence similarities for pairs of proteins from each pair of species
- Do
  - Find a global one-to-one mapping between network nodes

Overview of FUSE

- Fuse sequence similarities and network wiring patterns over *all* proteins in *all* PPI networks being aligned

- Create a one-to-one Global Multiple Network Alignment

Fuse step

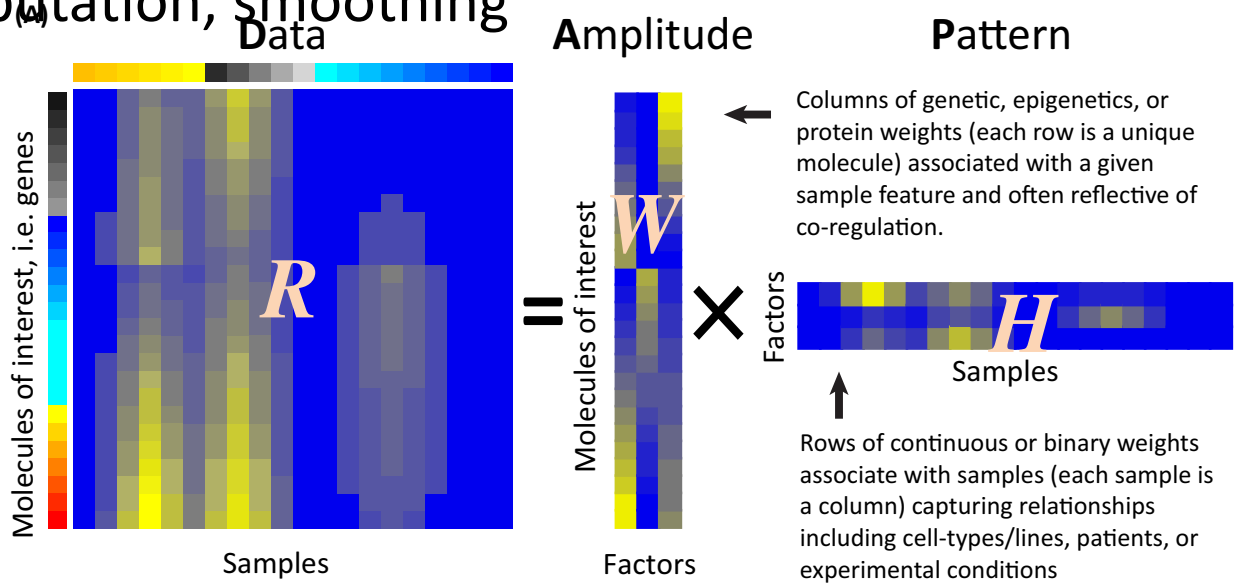
- Based on Non-negative matrix tri-factorization (NMTF)
- Derives functional scores between pairs of proteins using sequence and network information for k species
- Conceptually similar goal to IsoRank

Notation

- $N_i = (V_i, E_i)$ denotes the vertex and edge set for a PPI network for species i
- Let n_i denote the number of proteins in species i
- Let $\mathbf{R}_{ij}$ denote an $n_i X n_j$ sequence similarity matrix (E-values) of proteins from species i and j
- Let G_i and G_j specify the cluster assignments of proteins in i and j
- Let $\boldsymbol{S}_{ij}$ is a $k_i X k_j$ lower-dimensional approximation of $\mathbf{R}_{ij}$
  - where $k_i << n_i$ and $k_j << n_j$
  - k_i and k_j can be thought of as the number of independent groups in N_i and N_j

Matrix factorization

- A popular data analysis technique used for high-dimensional datasets
- Decomposition and factorization used interchangeably
- Many applications:
  - visualization, pattern extraction, interpretation and imputation, smoothing



Stein-O'Brien et al, Trends in Genetics 2018

Many different variants of MF

- Singular value decomposition

$$R = UDV^T$$

- Penalized matrix factorization

$$||R = UDV^T||_F^2 + \lambda_1||U||_1 + \lambda_2||V||_1$$

- Non-negative matrix factorization

$$||R - WH^T||_F^2; \text{s.t. } W \geq 0, H \geq 0$$

- …

Cluster indicator matrix

- Let k be the total number of clusters
- Let G be a cluster indicator matrix
- G is an nXk matrix which specifies the cluster ID for each entity v in V
- Example G matrix for 5 objects and two clusters

$$\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Clustering as matrix factorization

- Suppose we are given a matrix $\boldsymbol{X}$ of n objects (rows) and m attributes, that we want to cluster into k clusters

$$\mathbf{X} = [x_1, \cdots, x_n]$$

- k-means aims does this by minimizing

$$J = \sum_{c} \sum_{i, G_c(i)=1} ||x_i - f_c||^2$$

Membership vector

- This is equivalent to minimizing

$$||\mathbf{X} - \mathbf{F}\mathbf{G}^T||_2^2$$

where

$$\mathbf{F} = [f_1, \cdots, f_k], \mathbf{G} = [g_1, \cdots, g_k]$$

and each f_c is an m-dimensional vector. $\mathbf{F}$ is mXk and $\mathbf{G}$ is nXk

Semi-Supervised Clustering via Matrix Factorization, Wang et al, 2008

Clustering with guidance aka penalized MF

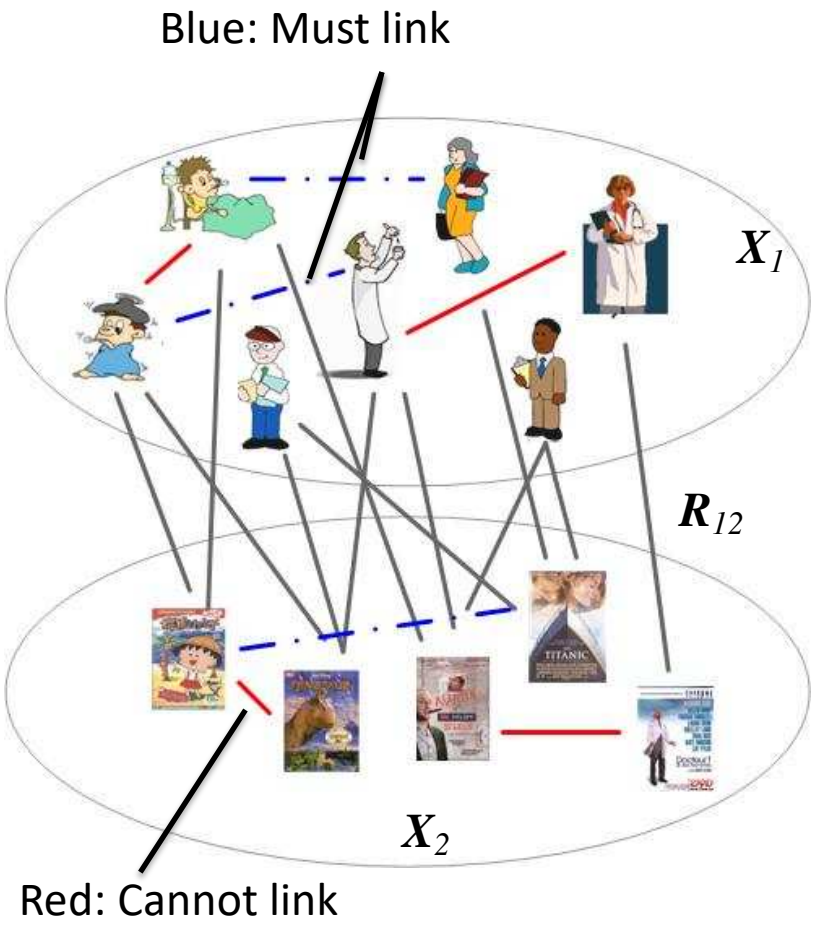
- Clustering by itself may be unreliable
- Suppose we have some additional information on the entities we wish to cluster, which allows us to say which entities tend to be together (must link)and which don't (don't link)
- These relationships could be used as constraints to guide the clustering
- Let Θ be an nXm constraint matrix encoding these link and don't link relationships
- We can use these constraints to define a new objective as follows:

$$||\mathbf{X} - \mathbf{F}\mathbf{G}^T||_2^2 + \mathrm{tr}(\mathbf{G}^T\Theta\mathbf{G}), \mathrm{s.t.}\ \mathbf{G} \geq 0$$

Non-negative matrix tri-factorization (NMTF)

- Extends the constrained matrix factorization from one type of entity to two types of entities represented by X_1 and X_2
- A co-clustering (simultaneously clustering) of different types of entities based on the relationship of within and between entity types
  - Cluster X_1 into G_1 and X_2 into G_2
- The intra-type relationships provide constraints into what objects can (must-link) and cannot be grouped together

Example of NMTF



Two types of objects: people and movies

Movies can be grouped based on actors, characters, titles

People can be grouped based on their hobbies and jobs

Semi-Supervised Clustering via Matrix Factorization, Wang et al, 2008

NMTF for two entity types

- Let J denote the objective

Matrix tri-factorization

$$\min_{G_i \geq 0, G_j \geq 0} J = ||\mathbf{R}_{ij} - G_i \mathbf{S}_{ij} G_j{}^T||_2^2 + P(V_i) + P(V_j)$$

Non-negativity

Constraints based on the intra-type graphs

- Here $P(V_i)$ and $P(V_j)$ are penalties one pays, if the clustering of the objects do not obey the intra-type constraints
- How to define this?
  - We will use the Graph Laplacian for this

Defining the penalty function with the graph Laplacian

- Recall f'the Laplacian L can be defined as

$$L = D - A$$

- Where D is the degree matrix and A is the adjacency matrix
- Furthermore, for every vector f in R^n,

$$f'Lf = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

Edge weight

- If f is a cluster assignment to nodes in the graph, the above function measures how consistent is f wrt to the graph
- The more the cluster assignment obeys the connectivity the smaller this quantity

Defining the penalty function with the graph Laplacian

- Let G_i be a cluster indicator matrix
- We can assess the goodness of G_i with respect to the graph N_i as

$$P(V_i) = Tr(G_i^T L_i G_i)$$

Tr: Trace: sum of diagonal elements

NMTF for two entity types

- For two entity types i and j

$$\min_{G_i \geq 0, G_j \geq 0} J = ||\mathbf{R}_{ij} - G_i \mathbf{S}_{ij} G_j^T||_F^2 + \gamma(\mathrm{Tr}(G_i^T L_i G_i) + \mathrm{Tr}(G_j^T L_j G_j))$$

Trade-off between maintaining intra-type constraints and estimating $\boldsymbol{R}_{ij}$

Rewriting the objective

- Let $\mathbf{L}$, $\mathbf{R}$, $\mathbf{G}$ be defined as follows matrices

$$\mathbf{R} = \begin{bmatrix} 0 & \mathbf{R}_{12} \\ \mathbf{R}_{21} & 0 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix}, \mathbf{S} = \begin{bmatrix} 0 & \mathbf{S}_{12} \\ \mathbf{S}_{21} & 0 \end{bmatrix}, \mathbf{L} = \begin{bmatrix} \mathbf{L}_1 & 0 \\ 0 & \mathbf{L}_2 \end{bmatrix}$$

- We can re-write the objective as follows

$$\left\| \begin{bmatrix} 0 & R_{12} \\ R_{21} & 0 \end{bmatrix} - \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix} \begin{bmatrix} 0 & \mathbf{S}_{12} \\ \mathbf{S}_{21} & 0 \end{bmatrix} \begin{bmatrix} G_1^T & 0 \\ 0 & G_2^T \end{bmatrix} \right\|_F^2$$
$$+ \gamma \mathrm{Tr} \left(\begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix} \begin{bmatrix} L_1 & 0 \\ 0 & L_2 \end{bmatrix} \begin{bmatrix} G_1^T & 0 \\ 0 & G_2^T \end{bmatrix} \right)$$

- Which is compactly written as

$$||\boldsymbol{R} - \boldsymbol{G}\boldsymbol{S}\boldsymbol{G}^T||_2^2 + \gamma \mathrm{Tr}(\boldsymbol{G}\boldsymbol{L}\boldsymbol{G}^T)$$

Extending to k entity types

- For entities of k different types, we have k different constraint graphs
- We can write the objective as

$$\min_{G_1 \geq 0, \cdots G_k \geq 0} J$$

$$= \sum_{ij} ||\mathbf{R}_{ij} - G_i \mathbf{S}_{ij} G_j^T||_2^2 + \gamma \left(\sum_i \mathrm{Tr}(G_i^T L_i G_i) \right)$$

This objective can be solved using an iterative multiplicative update algorithm from Wang 2008

NMTF for the GMNA problem

- Each entity type is a species
- Each entity is a protein from a species
- Constraints are specified by the protein-protein interaction networks N_i
- $\mathbf{R}_{ij}$ is the pairwise functional similarity between proteins of species i and j

NMTF for k=5 species

Sequence similarity for species 1 and 2

$$\mathbf{R} = \begin{bmatrix} 0 & \mathbf{R}_{12} & \dots & \mathbf{R}_{15} \\ \mathbf{R}_{12}^T & 0 & \dots & \mathbf{R}_{25} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{15}^T & \mathbf{R}_{25}^T & \dots & 0 \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} \mathbf{L}_1 & 0 & \dots & 0 \\ 0 & \mathbf{L}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{L}_5 \end{bmatrix}$$

Laplacian for PPI network 2

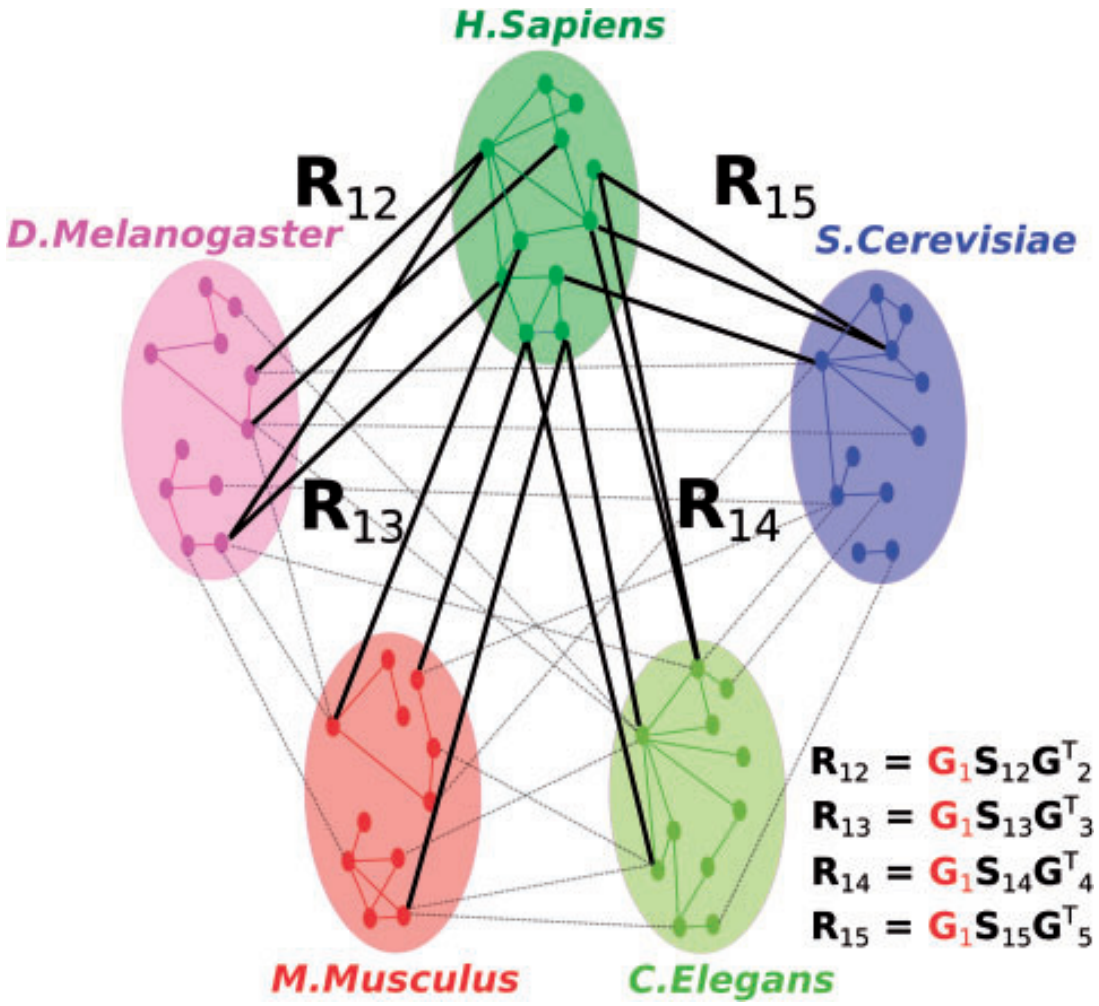
Low-dimensional representation of R_{12}

$$\mathbf{S} = \begin{bmatrix} 0 & \mathbf{S}_{12} & \dots & \mathbf{S}_{15} \\ \mathbf{S}_{12}^T & 0 & \dots & \mathbf{S}_{25} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{15}^T & \mathbf{S}_{25}^T & \dots & 0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{G}_1 & 0 & \dots & 0 \\ 0 & \mathbf{G}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{G}_5 \end{bmatrix}$$

Cluster assignment for proteins in species 2

$$\min_{G>=0} ||\boldsymbol{R} - \boldsymbol{GSG}^T||_2^2 + \gamma \mathrm{Tr}(\boldsymbol{GLG}^T)$$

Pictorial illustration for five species



Minimizing the objective NMTF

- We need to find S_{ij}, and G_i for all $1 \leq i,j \leq k$ entity types
- An iterative algorithm is used that updates each entity one at a time
- Updates are obtained by deriving the J (while accounting for the non-negativity constraints) with respect to S_{ij} and G_i respectively

Overview of FUSE

- Fuse sequence similarities and network wiring patterns over *all* proteins in *all* PPI networks being aligned
- Create a one-to-one Global Multiple Network Alignment

Global network alignment step

- Find a one-to-one Global Multiple Network Alignment
  - Create a k-partite graph, where k is the number of species
  - Finding approximately maximum weight k-partite **matching**

Create a k-partite weighted graph

- Recompute the new similarity based on sequence and network

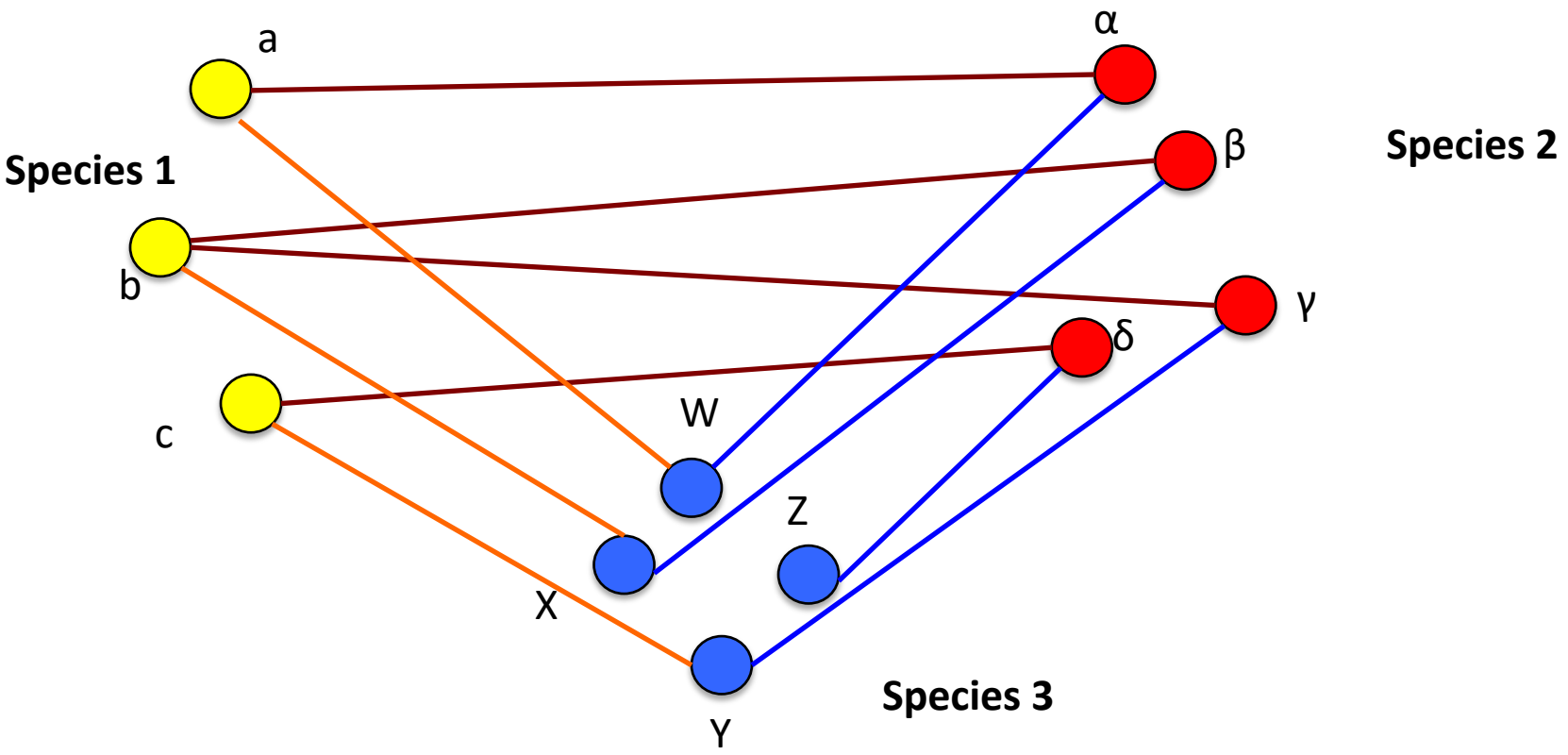
$$\widehat{\mathbf{R}}_{ij} = G_i \mathbf{S}_{ij} G_j^T$$

- Select top 5% of the associations of each protein in a species
- Add back entries that were set to 0 by NMTF but have sequence similarity using a weighted sum of sequence and NMTF-based similarity

$$w_{u,v} = \alpha \times \mathrm{seq}(u, v) + (1 - \alpha) \times \widehat{\mathbf{R}}_{ij}(u, v)$$

- This produces a weighted k-partite graph

A 3-partite weighted graph



Algorithm to find the best matching

- Matching between two node sets is defined as a one-to-one mapping
- Weighted k-partite matching for k>2 is NP-hard
- Need a heuristic approach

Heuristic algorithm to find a maximum k-partite matching

K-partite graph

Input $G = (\cup_{i=1}^{k} V_i, E, W)$

for $i = \{2, \ldots, k\}$ **do**

Find maximum weight bipartite matching $F_{1,i}$ of $G[V_1, V_i]$

Construct G_{1i}, the merge of V_1 and V_i from G along $F_{1,i}$

Set $G = G_{1i}$ and relabel V_{1i} as V_1

$C = \{\emptyset\}$

for each merged node u in V_1 **do**

Cluster C_u is the set of nodes that are merged into u

Add C_u to C

Output C

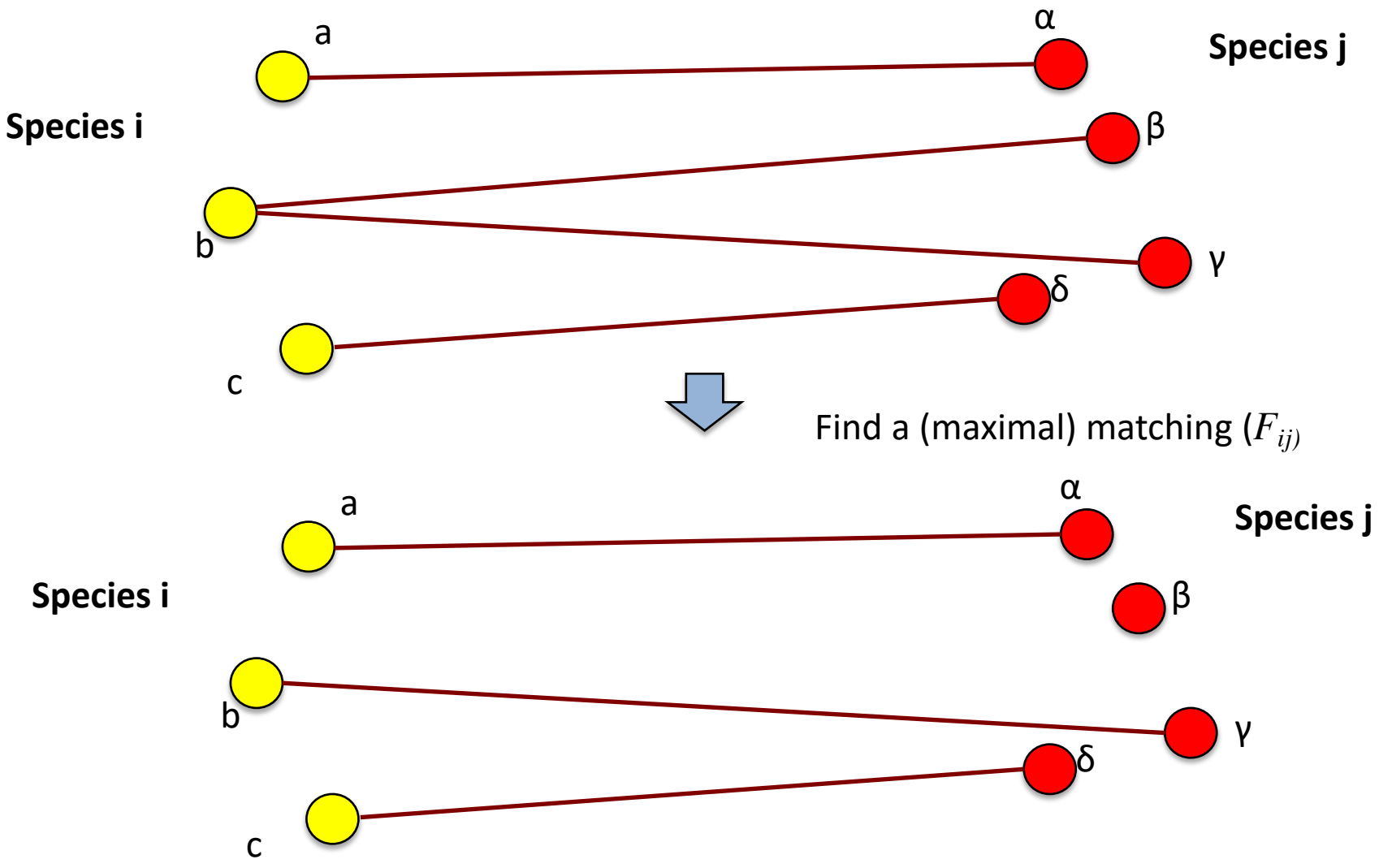
Graph merge step

- Let our k-partite graph be

$$M = (\cup_{i=1}^{k} V_i, E, W)$$

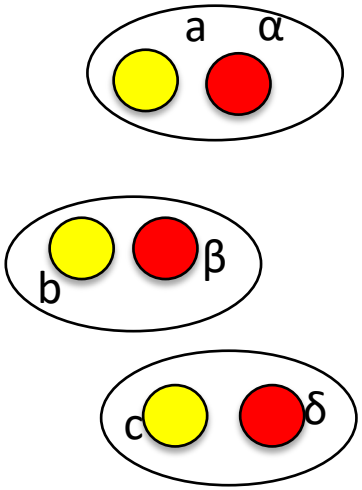
- Let F_{ij} be a matching between nodes in V_i and V_j, where u_i in V_i is mapped to v_j in V_j
- Create new vertices V_{ij} from the matching, each vertex represented by a pair of nodes one from each graph
- This step is like creating an alignment graph!

Bi-partite matching example

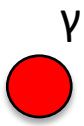


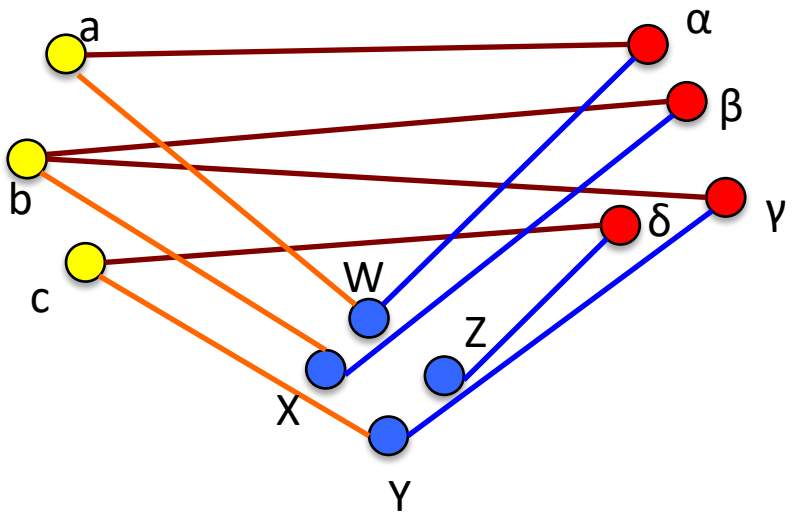
Graph merge from matching

Merged graph of species 1, 2

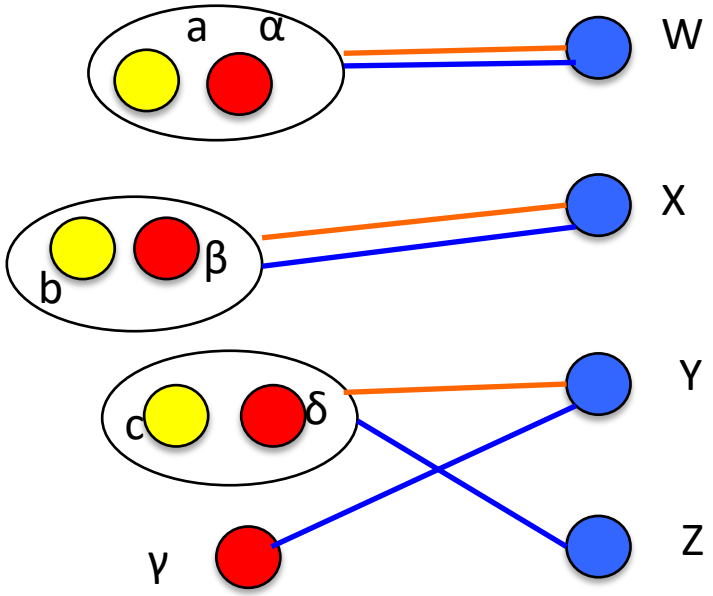


The merged nodes inherit the edges of the constituent nodes





New graph with all species



Note, this is not a matching

Results

- Dataset: Protein-protein interaction networks for five species
  - Human (H. sapiens), mouse (M. musculus), fly (D. melanogaster), worm (C. elegans), yeast (S. cerevisiae)
- Experiments
  - Assess the inferred functional orthologies based on similarity in annotation
  - Compare against other methods

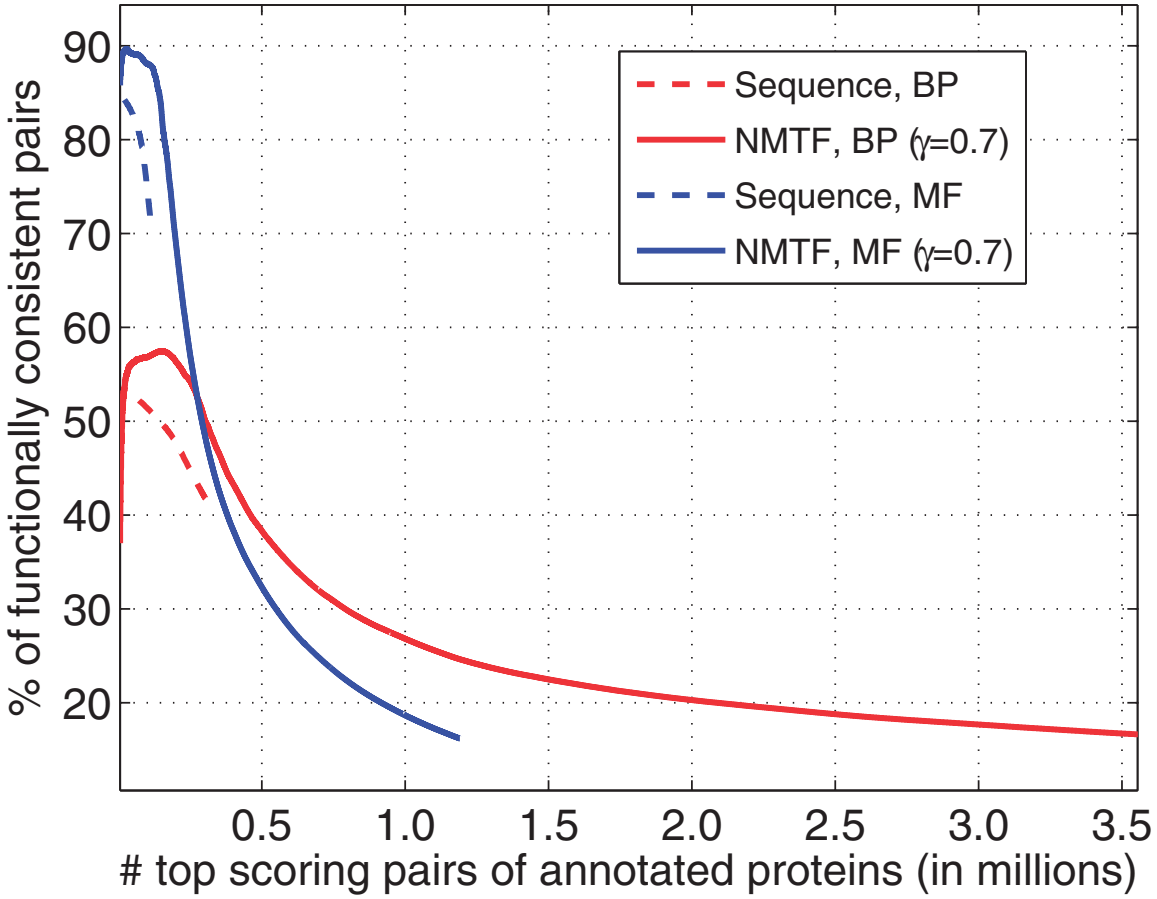
Statistics of PPI networks used

Id Species	No. nodes	BP Ann. (%)	MF Ann. (%)	CC Ann. (%)	No. edges
Homo sapiens	14 164	37.2	23.2	9.6	127 907
Saccharomyces cerevisiae	6004	65.0	41.7	17.4	223 008
Drosophila melanogaster	8125	36.1	13.4	6.3	38 892
Mus musculus	5100	53.3	23.9	10.6	11 061
Caenorhabditis elegans	3841	35.0	7.3	4.2	7726

NMTF induces new and reconstructs existing associations between proteins

- Apply PCA to estimate k_i, the number of clusters/factors for each species
  - k_1= 80 (human); k_2 = 90 (yeast); k_3= 80 (fly); k_4= 70 (mouse) and k_5=50 (worm)
- 1,477, 372 interactions based on sequence
- 5% edges inferred corresponds to 19, 175, 378
  - Covers 60% of the sequence-only edges
  - What happens to the 40% edges?
- Compare the reconstructed (60%), predicted and non-reconstructed (40%) pairs
- Count the number of sequence-similar neighbors in each network
  - Pairs with reconstructed similarities or new similarities are connected to many more similar neighbors (20.4 on average)
  - Pairs with new similarities are also connected to neighbors with high sequence similarity (12.1)
  - Pairs that are not reconstructed have much lower sequence similarity in their neighborhood.

Do the new similarities make sense?



Compute the cumulative number of associations between annotated proteins and the percentage of them sharing GO term (Biological Process and Molecular Function annotations separately).

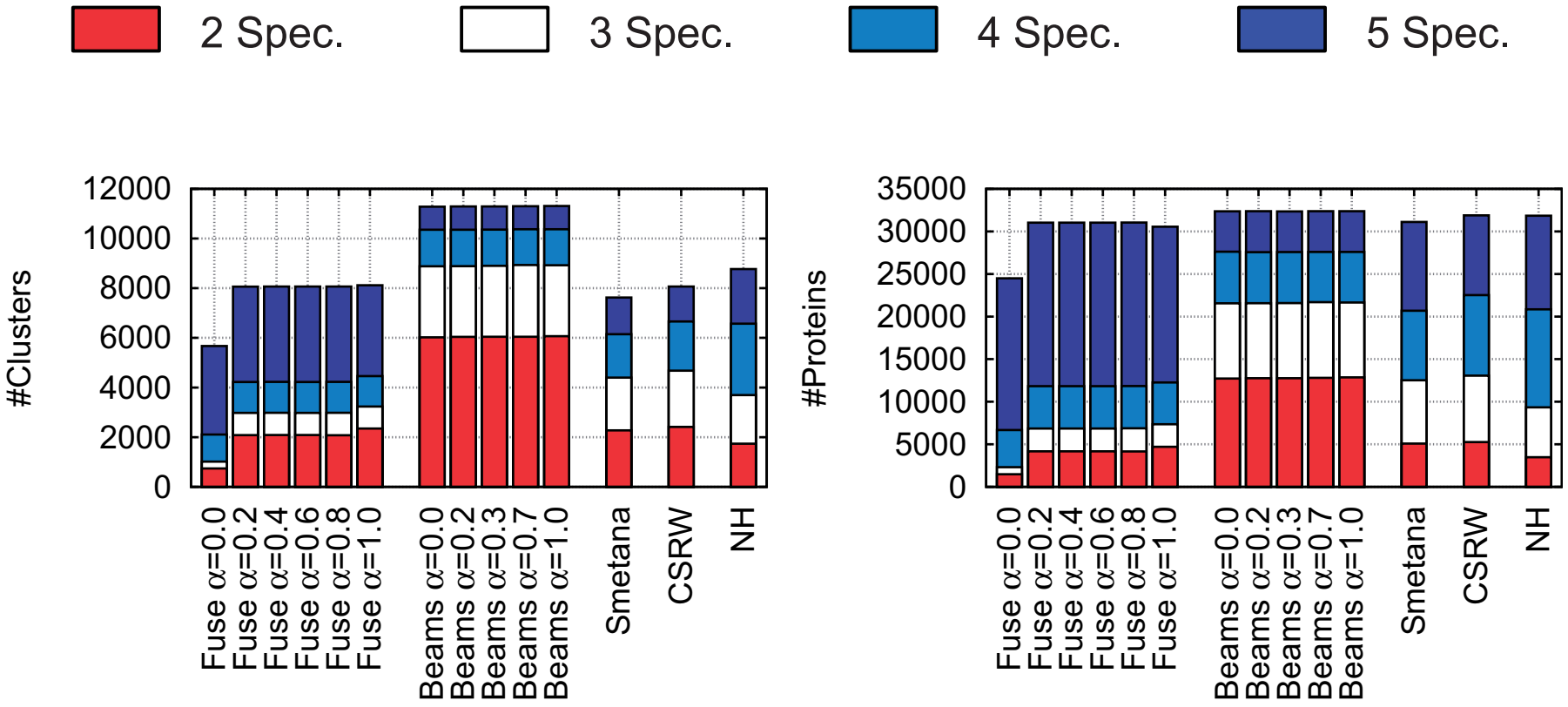
Comparison against other algorithms

- Algorithms compared
  - Beams
  - Smetana
  - CSRW
  - NH
  - IsoRank
  - NetCoffee

Did not finish in time

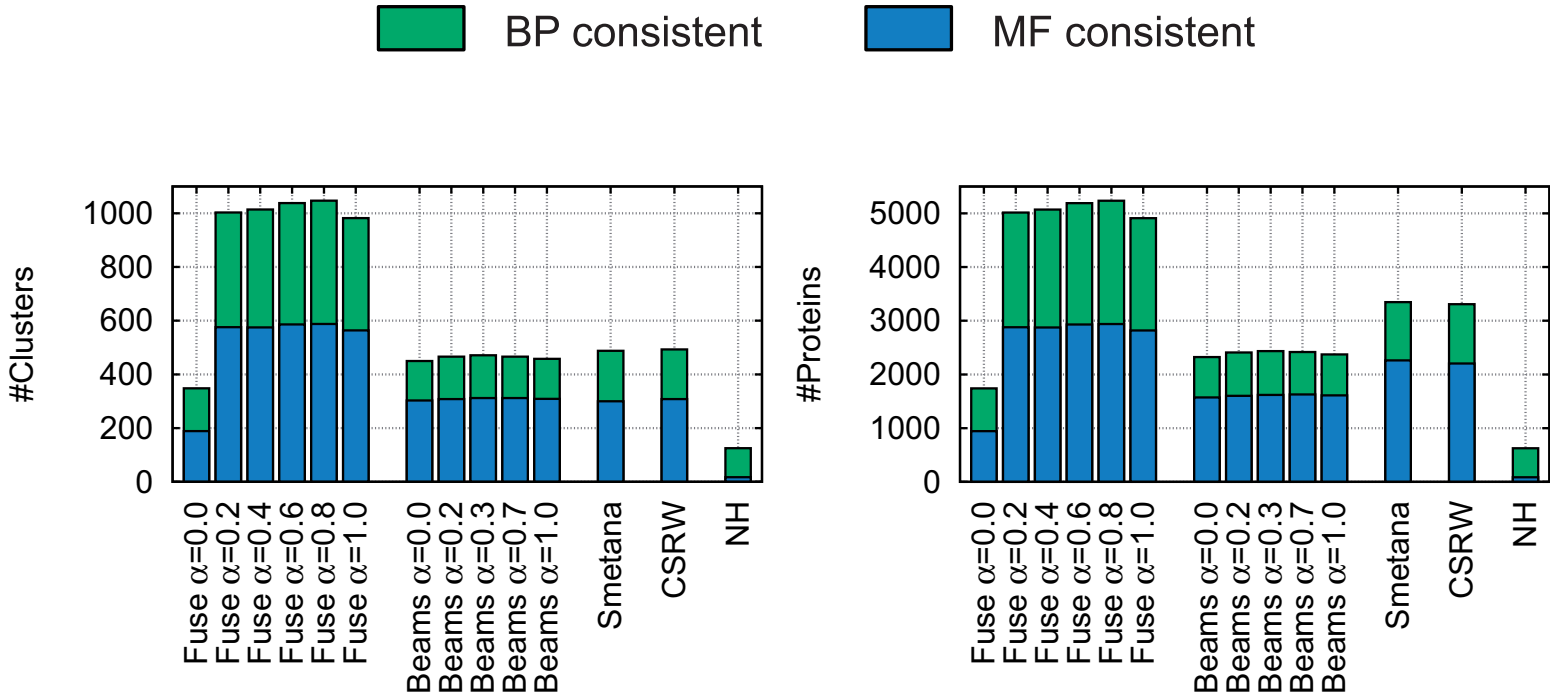
- Evaluation metrics
  - Coverage
    - Good clusters: cover all five PPI networks
    - Bad clusters: cover less than five PPIs
    - Computed at the cluster and protein level

FUSE produces the largest number of good clusters



Fraction of blue is highest for clusters and proteins.

FUSE produces functionally consistent clusters



A cluster is said to be functionally consistent, if all its annotated proteins have at least one GO term in common.

Summary

- FUSE is a multiple network alignment algorithm
- It uses multiple graphs simultaneously to redefine the functional similarity among proteins
- Strengths: Compared to existing algorithms it is able to infer higher coverage and functionally consistent protein clusters (orthologous groups)
- Weaknesses:
  - One-to-one mapping misses out on gene duplications
  - The hyper-parameters might influence the results, and it is not clear how to set them.

Concluding remarks

- Network alignment seeks to find similarities and differences between molecular networks of different species
- We have seen algorithms for
  - Local Alignment
    - PathBLAST, Sharan et al 2004
    - Used a probabilistic, per edge score but was trying to find paths and modules
  - Global pairwise and multiple network alignment
    - IsoRank (many-to-many node mappings)
    - FUSE (one-to-one mappings)