# Building Cellular Interaction Databases: Analysis, Synthesis, and Interfaces

Mehmet Koyuturk, Yohan Kim,
Shankar Subramaniam, and Ananth Grama

March 7, 2005

# Outline

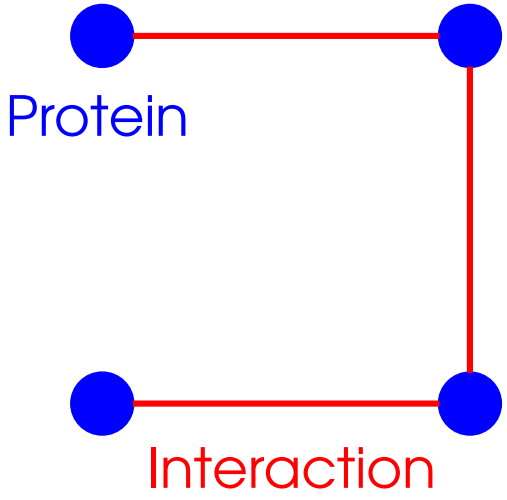
1. Biological Networks
   - Definition, problems, practical implications
2. Current Work
   - Mining biological networks for frequent molecular interaction patterns
   - Alignment of protein interaction networks based on evolutionary models
   - Module identification based on phylogenetic profiles
3. Ongoing and Future Work
   - Constructing reference module maps
   - Building a fully functional interoperable signaling database

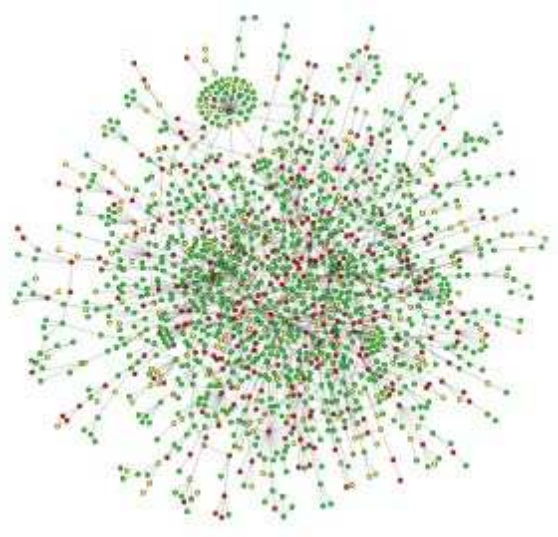
# Biological Networks

- Interactions between biomolecules that drive cellular processes
  - Genes, proteins, enzymes, chemical compounds
  - Mass & energy generation, information transfer
  - Coarser level than sequences in life's complexity pyramid
- Experimental/induced data in various forms
  - Protein-protein interaction networks
  - Gene regulatory networks
  - Metabolic & signaling pathways
- What do we gain from analysis of cellular networks?
  - Modular analysis of cellular processes
  - Understanding evolutionary relationships at a higher level
  - Assigning functions to proteins through interaction information
  - Intelligent drug design: block protein, preserve pathway

# Protein-Protein Interaction (PPI) Networks

- Interacting proteins can be discovered experimentally
  - Two-hybrid
  - Mass spectrometry
  - Tandem affinity purification (TAP)



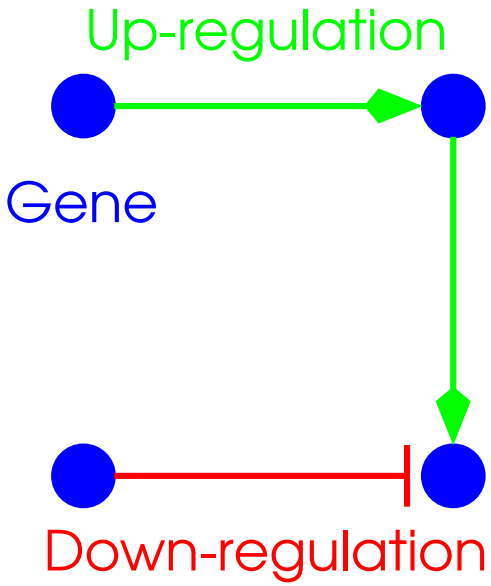



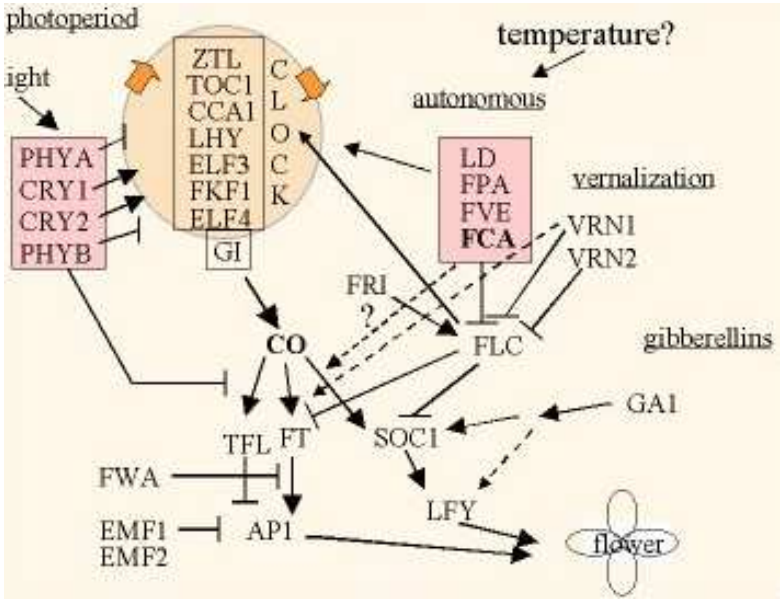
*S. Cerevisiae* protein interaction network

Source: Jeong et al. Nature 411: 41-42, 2001.

# Gene Regulatory Networks

- Genes regulate each others' expression
  - A simple model: Boolean networks
  - Can be derived from gene expression data



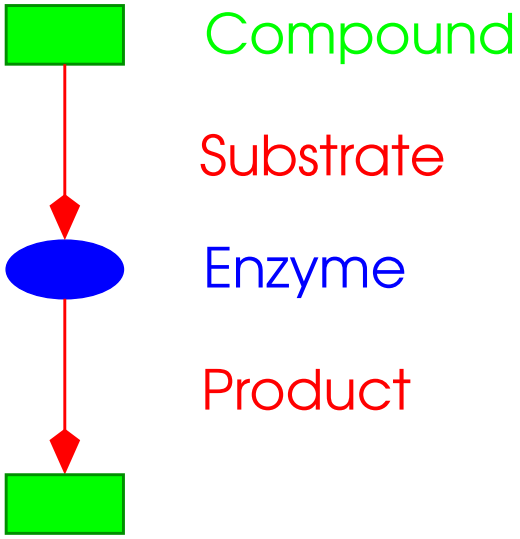



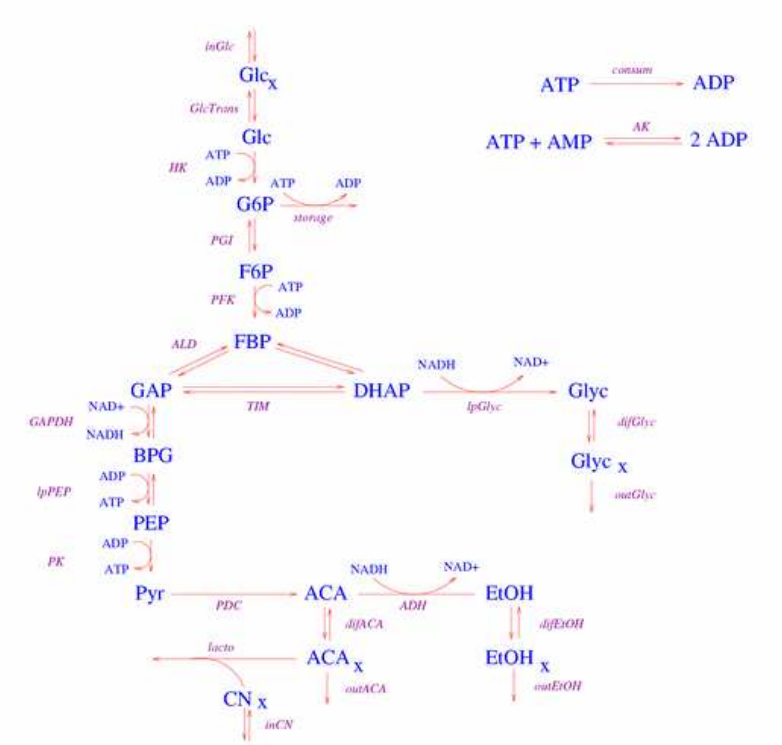

Genetic network that controls flowering time in *A. Thaliania*

Source: Blazquez et al. EMBO Reports 2: 1078-1082, 2001

# Metabolic Pathways

- Chains of reactions that perform a particular metabolic function
  - Reactions are linked to each other through substrate-product relationships
  - Directed hypergraph/ graph models




Glycolysis pathway in *S. Cerevisiae*

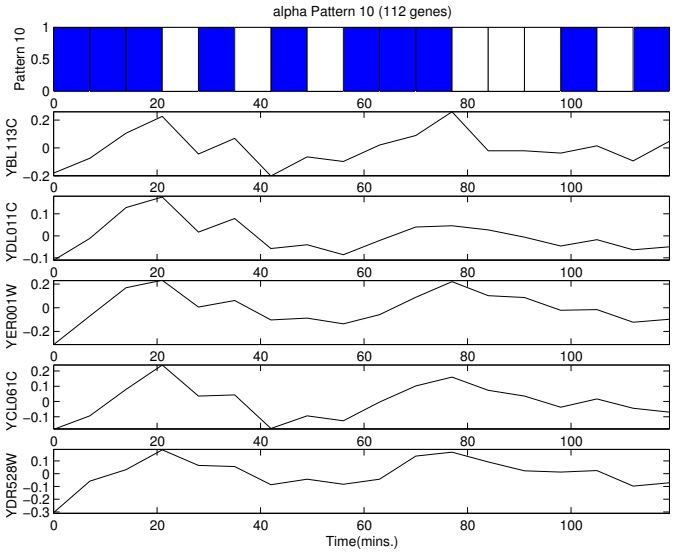
Source: Hynne et al. Biophysical Chemistry, 94, 121-163, 2001.

# Discrete Algebraic Techniques in Analysis

- Non-orthogonal decomposition of binary matrices
  - Find a compact set of vectors that represent the entire matrix
  - Recursive decomposition through rank-one approximations
  - Fast (linear-time) iterative heuristics for computing approximations

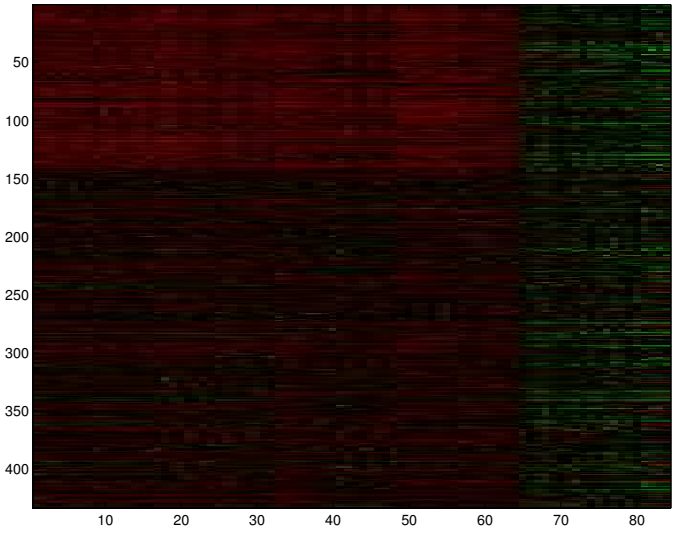
## Analysis of gene expression data

### Patterns of regulation




"Algorithms for bounded-error correlation of high dimensional data in microarray experiments" Koyutürk, Grama, Szpankowski: *CSB'03.*

### Biclustering



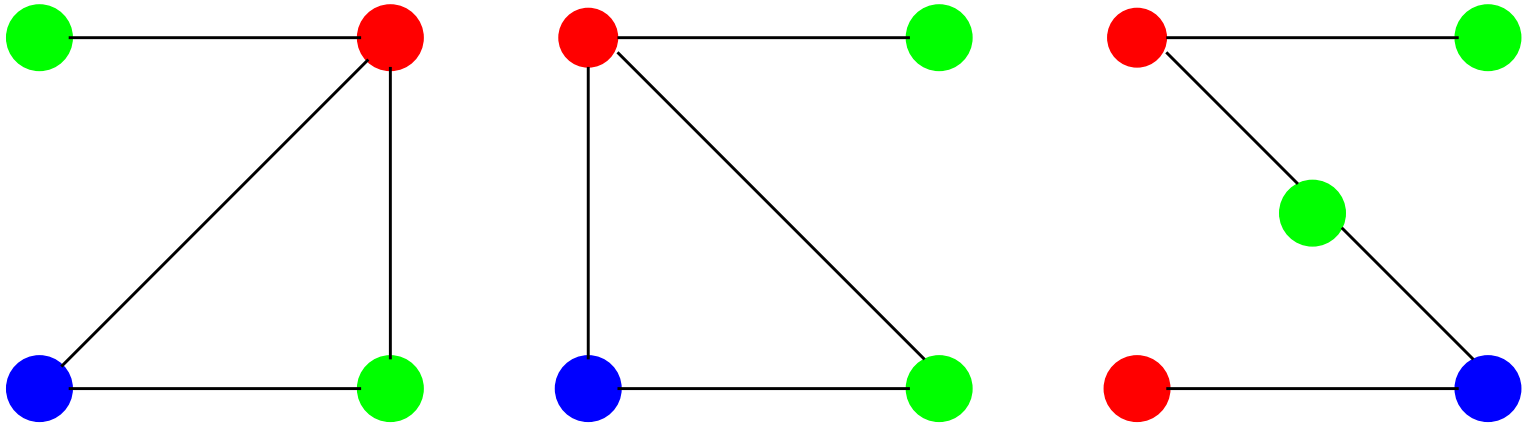
"Biclustering gene-feature matrices for statistically significant dense patterns" Koyutürk, Grama, Szpankowski: *CSB'04.*

# Analysis of Biological Networks

- Evolution thinks modularly
  - Selective pressure on preserving interactions
  - Functional modules, protein complexes are highly conserved
- Computational methods for discovery and analysis of modules and complexes
  - Graph clustering: Functionally related entities are densely connected
  - Graph mining: Common topological motifs, frequent interaction patterns reveal modularity
  - Graph alignment: Conservation/divergence of modules and pathways
  - Module maps: Canonical pathways across species
  - Phylogenetic analysis: Genes/proteins that belong to a common module are likely to have co-evolved

# How do we detect conserved subgraphs: Graph Mining

(Koyuturk, Grama, Szpankowski, ISMB04, Bioinformatics04)



Graph database

Subgraphs with frequency 3

# Extending Frequent Itemset Mining to Graph Mining

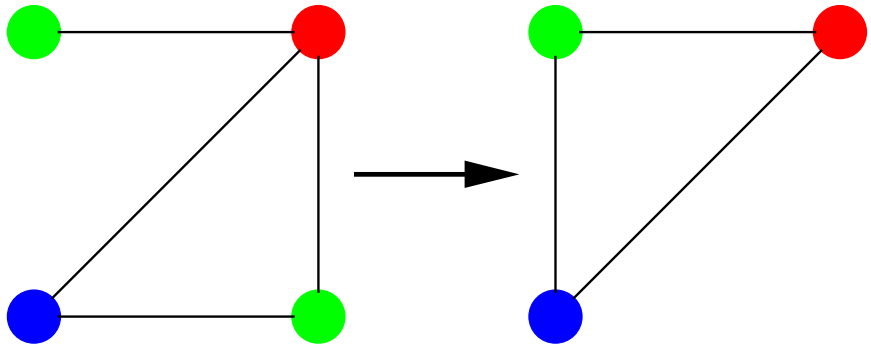
- Given a set of transactions, find sets of items that are frequent in these transactions
- Extensively studied in data mining literature
- Algorithms exploit downward closure property
  - A set is frequent only if all of its subsets are frequent
  - Generate itemsets from small to large, pruning supersets of infrequent sets
- Can be generalized to mining graphs
  - transaction $\longrightarrow$ graph
  - item $\longrightarrow$ node, edge
  - itemset $\longrightarrow$ subgraph
- However, the graph mining problem is considerably more difficult

# Graph Mining Challenges

- Subgraph Isomorphism
  - For counting frequencies, it is necessary to check whether a given graph is a subgraph of another one
  - NP-complete
- Canonical labeling
  - To avoid redundancy while generating subgraphs, canonical labeling of graphs is necessary
  - Equivalent to subgraph isomorphism
- Connectivity
  - Patterns of interest are generally connected, so it is necessary to only generate connected subgraphs
- Existing algorithms mainly focus on minimizing redundancy and mining & extending simple substructures
  - AGM, FSG, gSpan, SPIM, CLOSEGRAPH

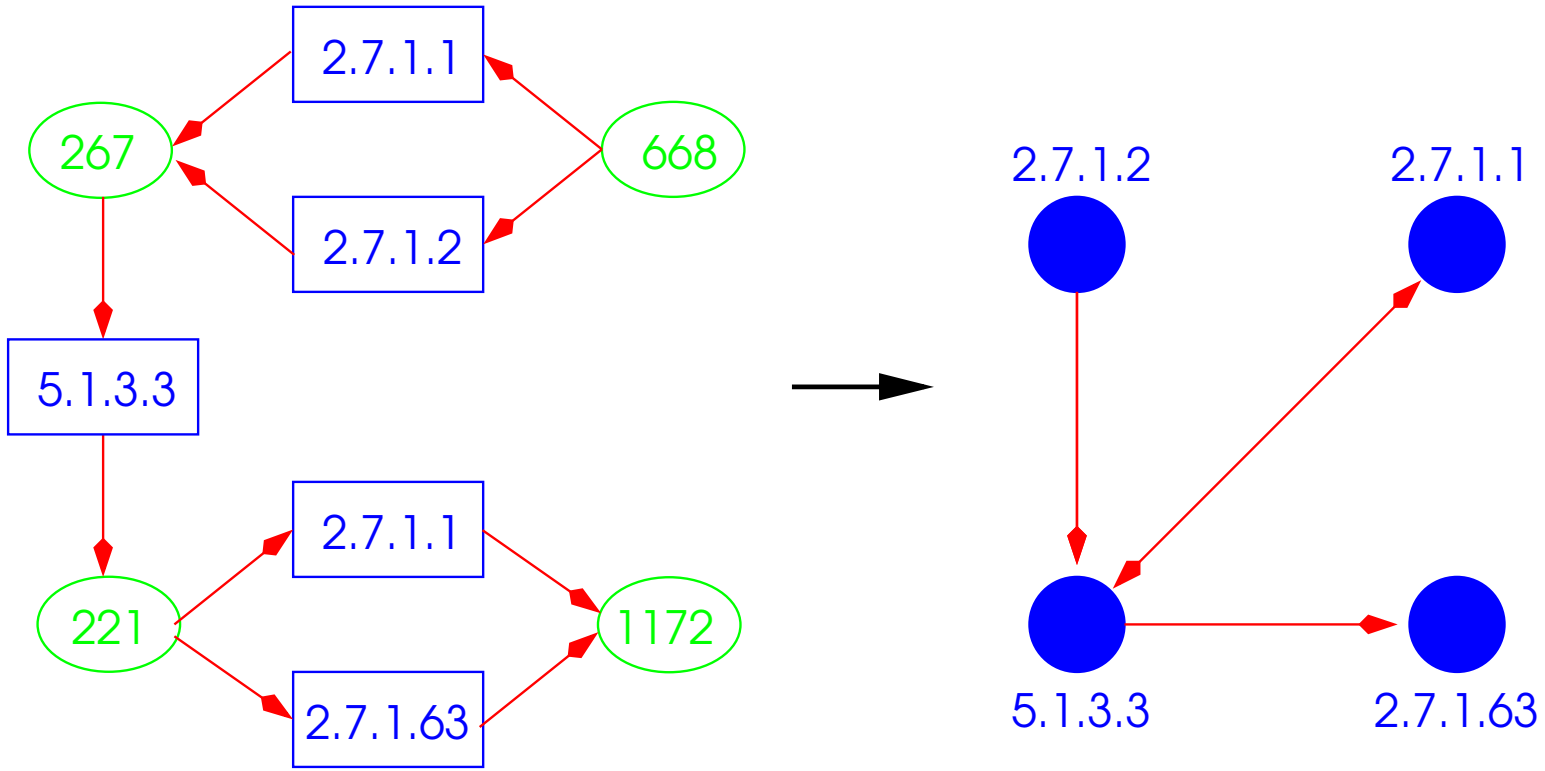
# Uniquely-Labeled Graphs

- Contract nodes with identical label into a single node
- No subgraph isomorphism
  - Graphs are uniquely identified by their edge sets
- Frequent subgraphs are preserved $\Rightarrow$ No information loss
  - Subgraphs that are frequent in general graphs are also frequent in their uniquely-labeled representation
- Discovered frequent subgraphs are still biologically interpretable!



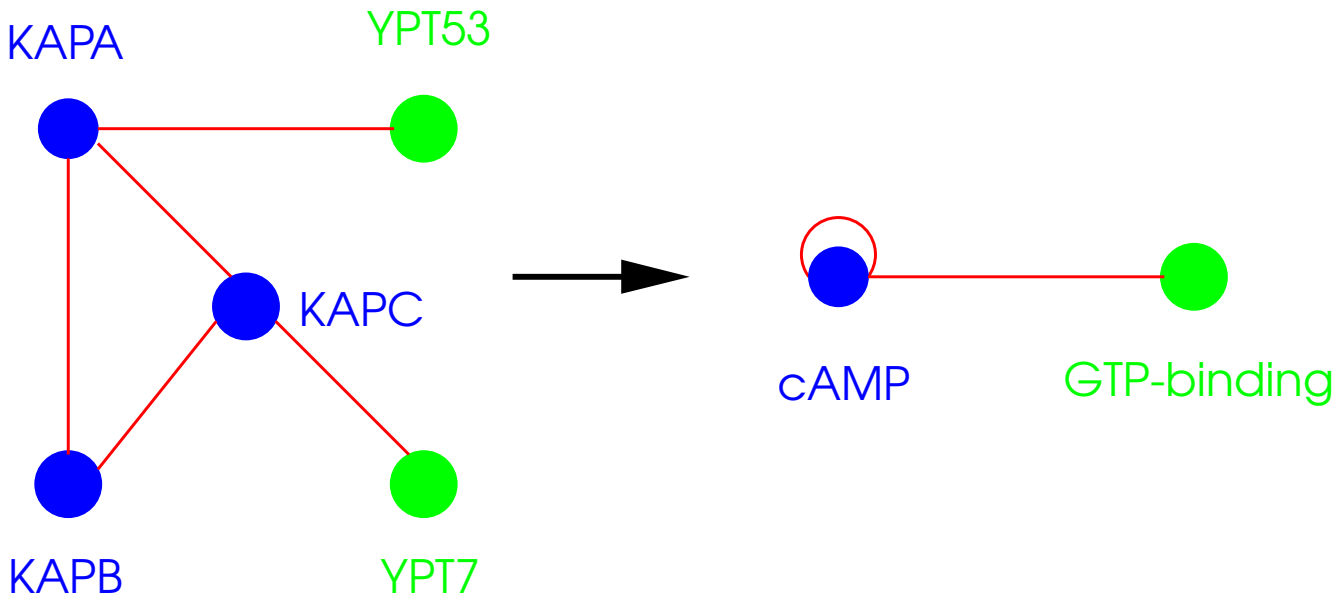
# Node Contraction in Metabolic Pathways

- Uniquely-labeled directed graph model
  - Nodes represent enzymes
  - Global labeling by enzyme nomenclature (EC numbers)
  - A directed edge from one enzyme to the other implies that the second consumes a product of the first



# Node Contraction in Protein Interaction Networks

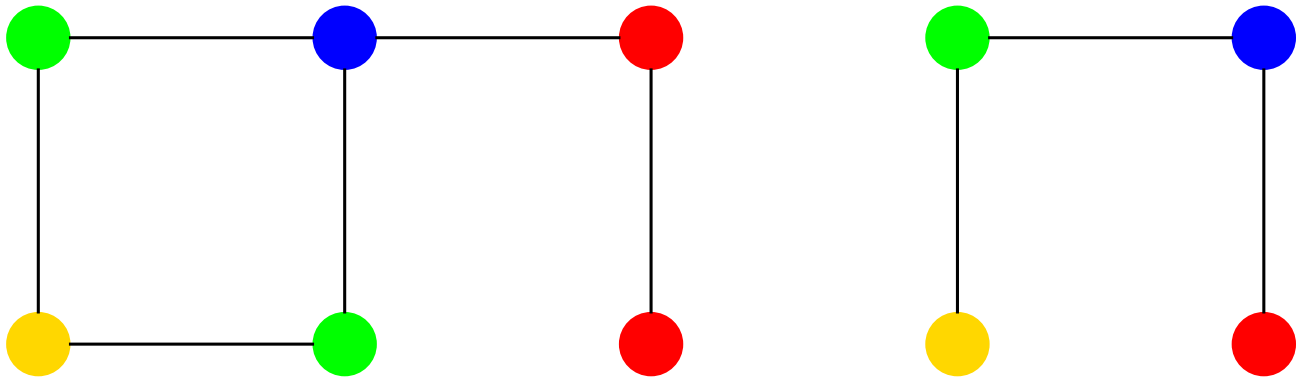
- Relating proteins in different organisms
  - Clustering: Orthologous proteins show sequence similarities
  - Phlyogenetic analysis: Allows multi-resolution analysis among distant species
  - Literature, ortholog databases
- Contraction
  - Interaction between proteins $\longrightarrow$ interaction between protein families



# Preservation of Subgraphs

Theorem: Let $\tilde{G}$ be the uniquely-labeled graph obtained by contracting the same-label nodes of graph $G$. Then, if $S$ is a subgraph of $G$, $\tilde{S}$ is a subgraph of $\tilde{G}$.

Corollary: The uniquely-labeled representation of any frequent subgraph is frequent in the set of uniquely-labeled graphs.



$G$ $\tilde{G}$

# Simplifying the Graph Mining Problem

Observation: A uniquely-labeled graph is uniquely determined by the set of its edges.

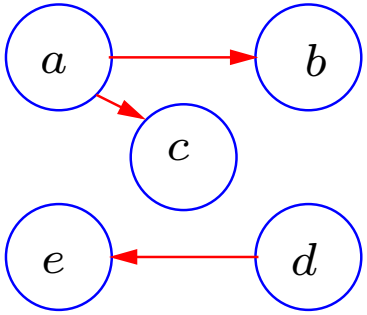
## Maximal Frequent Subgraph Mining Problem

Given a set of labeled graphs $\{G_1, G_2, ..., G_m\}$, find all connected graphs $S$ such that $S$ is a subgraph of at least $\sigma m$ of the graphs (is frequent) and no supergraph of $S$ is frequent (is maximal).

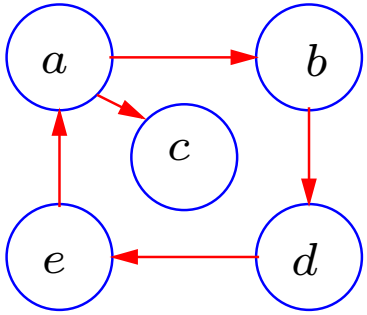
## Maximal Frequent Edgeset Mining Problem

Given a set of edge transactions $\{E_1, E_2, ..., E_m\}$, find all connected edge sets $F$ such that $F$ is a subset of at least $\sigma m$ of the edge transactions (is frequent) and no superset of $F$ is frequent (is maximal).

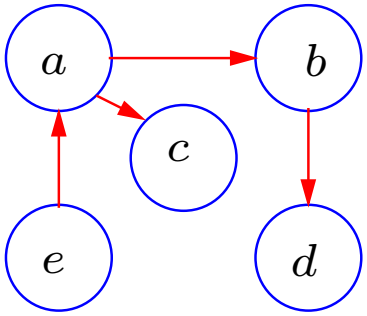
# From Graphs to Edgesets



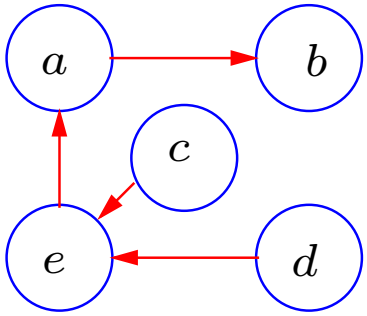

$G_1$




$G_2$




$G_3$




$G_4$

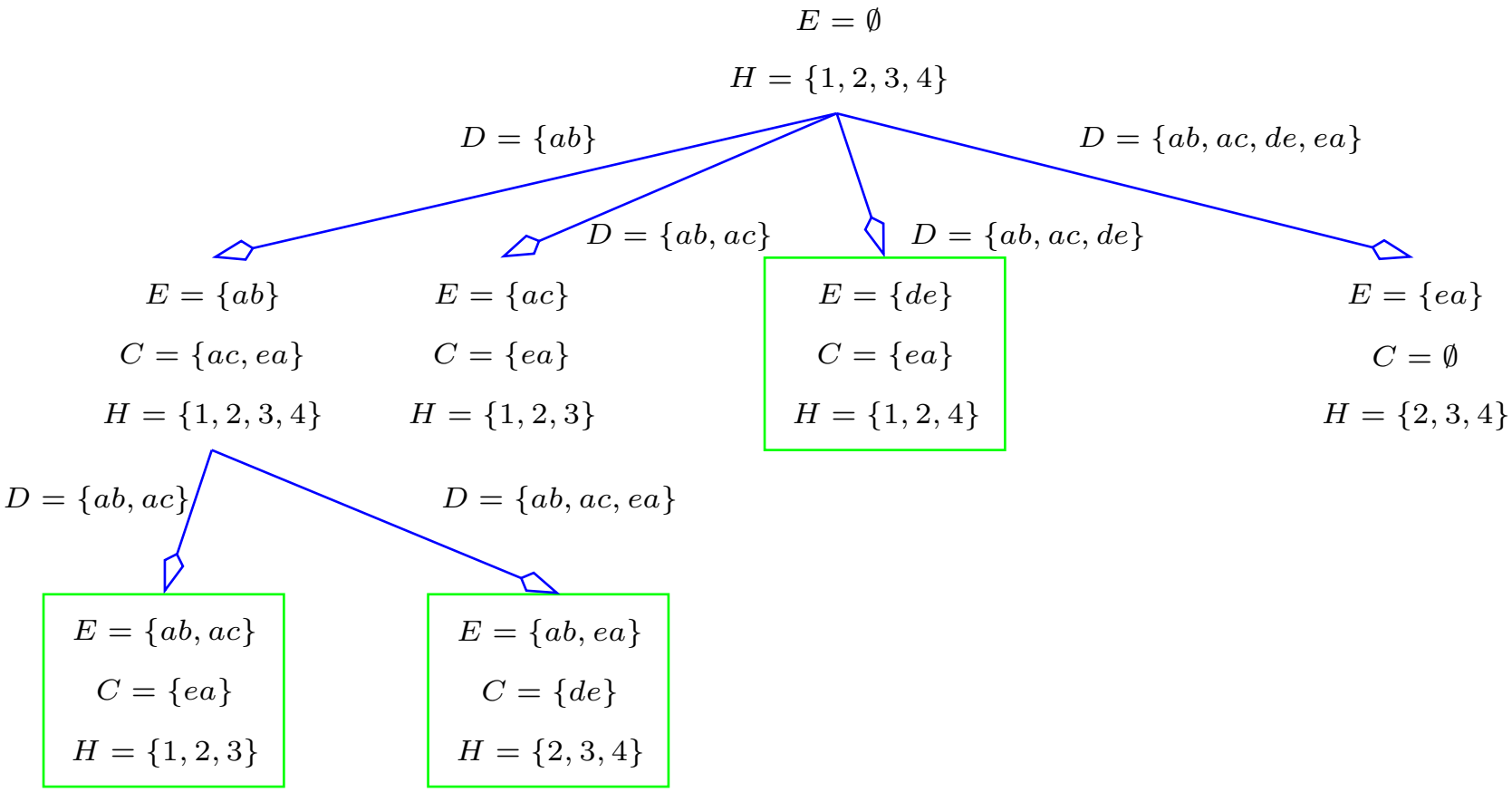
$F_1 = \{ab, ac, de\}$

$F_2 = \{ab, ac, bc, de, ea\}$

$F_3 = \{ab, ac, bc, ea\}$

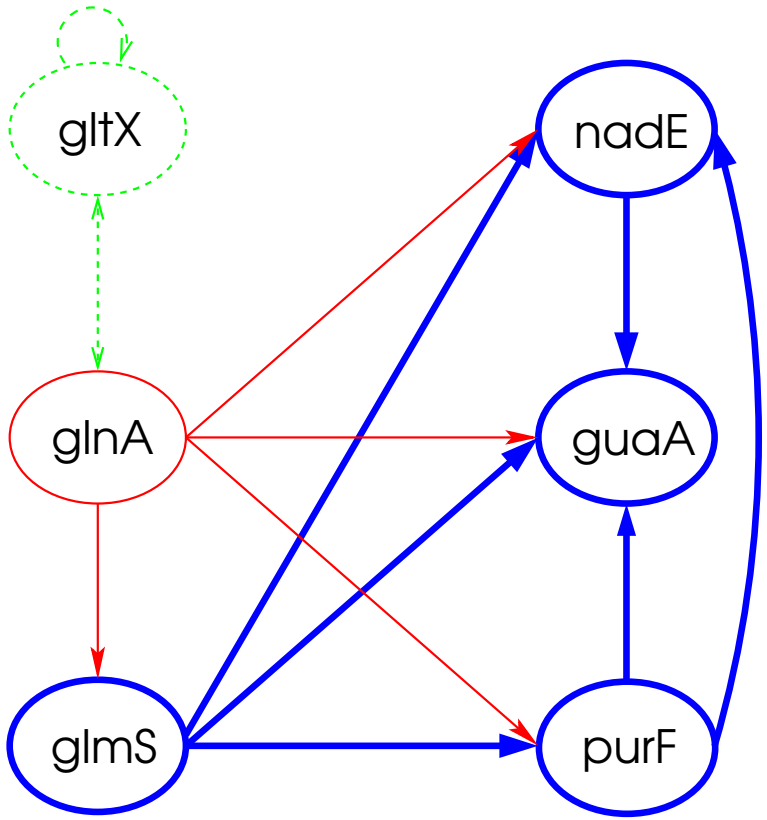
$F_4 = \{ab, ce, de, ea\}$

# MULE: Mining Uniquely Labeled Graphs



# Frequent Sub-Pathways in KEGG

Glutamate metabolism (155 organisms)



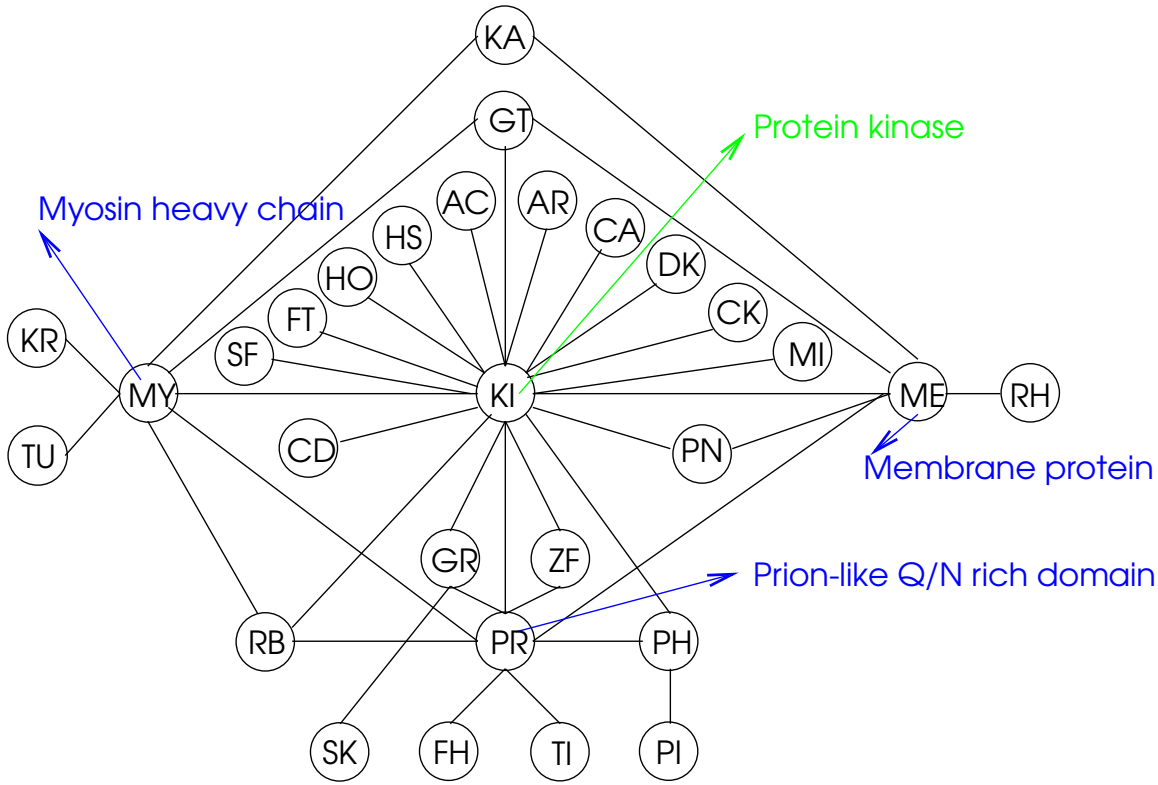

45 (29%) organisms

30 (19%) organisms

22 (14%) organisms

# Frequent Interaction Patterns in DIP

- Protein interaction networks for 7 organisms
  - Ecoli, Hpylo, Scere, Celeg, Dmela, Mmusc, Hsapi
  - 44070 interactions between 16783 proteins
- Clustering with TribeMCL & node contraction
  - 30247 interactions between 6714 protein families



# Runtime Characteristics

## Comparison with isomorphism-based algorithms

| Dataset | Minimum Support (%) | FSG Runtime (secs.) | FSG Largest pattern | FSG Number of patterns | MULE Runtime (secs.) | MULE Largest pattern | MULE Number of patterns |
|---|---|---|---|---|---|---|---|
| Glutamate | 20 | 0.2 | 9 | 12 | 0.01 | 9 | 12 |
| | 16 | 0.7 | 10 | 14 | 0.01 | 10 | 14 |
| | 12 | 5.1 | 13 | 39 | 0.10 | 13 | 39 |
| | 10 | 22.7 | 16 | 34 | 0.29 | 15 | 34 |
| | 8 | 138.9 | 16 | 56 | 0.99 | 15 | 56 |
| Alanine | 24 | 0.1 | 8 | 11 | 0.01 | 8 | 11 |
| | 20 | 1.5 | 11 | 15 | 0.02 | 11 | 15 |
| | 16 | 4.0 | 12 | 21 | 0.06 | 12 | 21 |
| | 12 | 112.7 | 17 | 25 | 1.06 | 16 | 25 |
| | 10 | 215.1 | 17 | 34 | 1.72 | 16 | 34 |

## Extraction of contracted patters

| Glutamate metabolism, $\sigma = 8\%$ | | | | Alanine metabolism, $\sigma = 10\%$ | | | |
|---|---|---|---|---|---|---|---|
| Size of contracted pattern | Extraction time (secs.) FSG | Extraction time (secs.) gSpan | Size of extracted pattern | Size of contracted pattern | Extraction time (secs.) FSG | Extraction time (secs.) gSpan | Size of extracted pattern |
| 15 | 10.8 | 1.12 | 16 | 16 | 54.1 | 10.13 | 17 |
| 14 | 12.8 | 2.42 | 16 | 16 | 24.1 | 3.92 | 16 |
| 13 | 1.7 | 0.31 | 13 | 12 | 0.9 | 0.27 | 12 |
| 12 | 0.9 | 0.30 | 12 | 11 | 0.4 | 0.13 | 11 |
| 11 | 0.5 | 0.08 | 11 | 8 | 0.1 | 0.01 | 8 |
| Total number of patterns: 56 | | | | Total number of patterns: 34 | | | |
| Total runtime of FSG alone: 138.9 secs. | | | | Total runtime of FSG alone :215.1 secs. | | | |
| Total runtime of MULE+FSG: 0.99+100.5 secs. | | | | Total runtime of MULE+FSG: 1.72+160.6 secs. | | | |
| Total runtime of MULE+gSpan: 0.99+16.8 secs. | | | | Total runtime of MULE+gSpan: 1.72+31.0 secs. | | | |

# Aligning Protein Interaction Networks

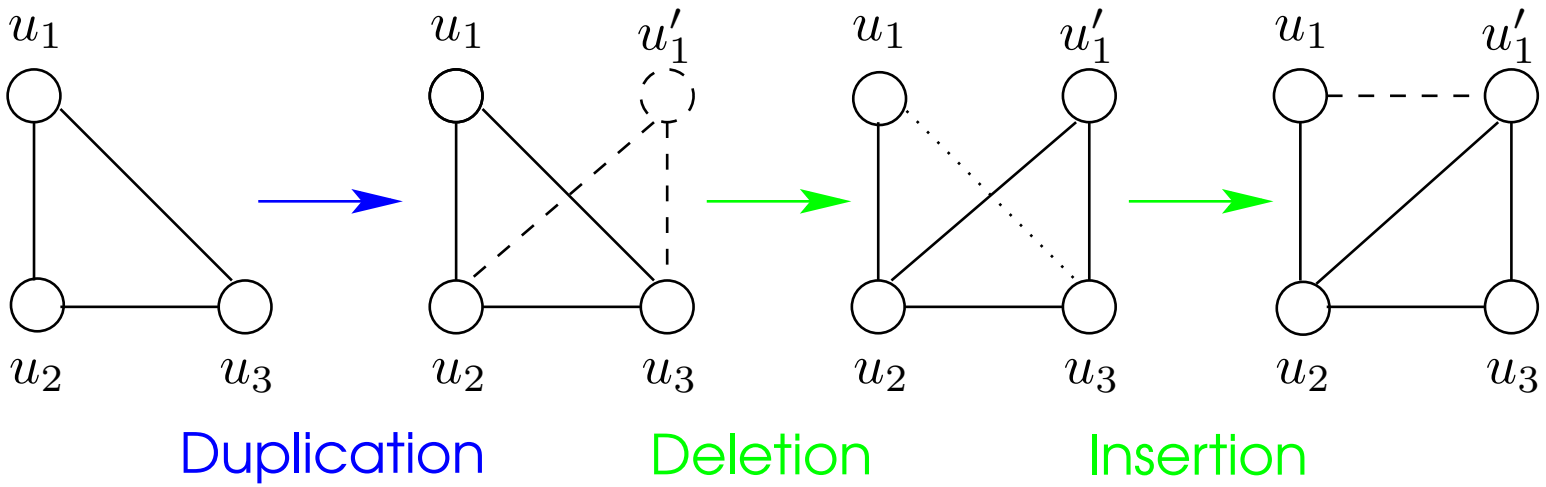
(Koyuturk, Grama, Szpankowski, RECOMB04)

- Defining graph alignment is difficult in general
  - Biological meaning
  - Mathematical modeling
- Existing algorithms are based on simplified formulations
  - PathBLAST aligns pathways (linear chains) to render problem computationally tractable
  - Motif search algorithms look for small topological motifs, do not take into account conservation of proteins
- Our approach
  - Aligns subsets of proteins based on the observation that modules and complexes are conserved
  - Guided by models of evolution

# Evolution of Protein Interaction Networks

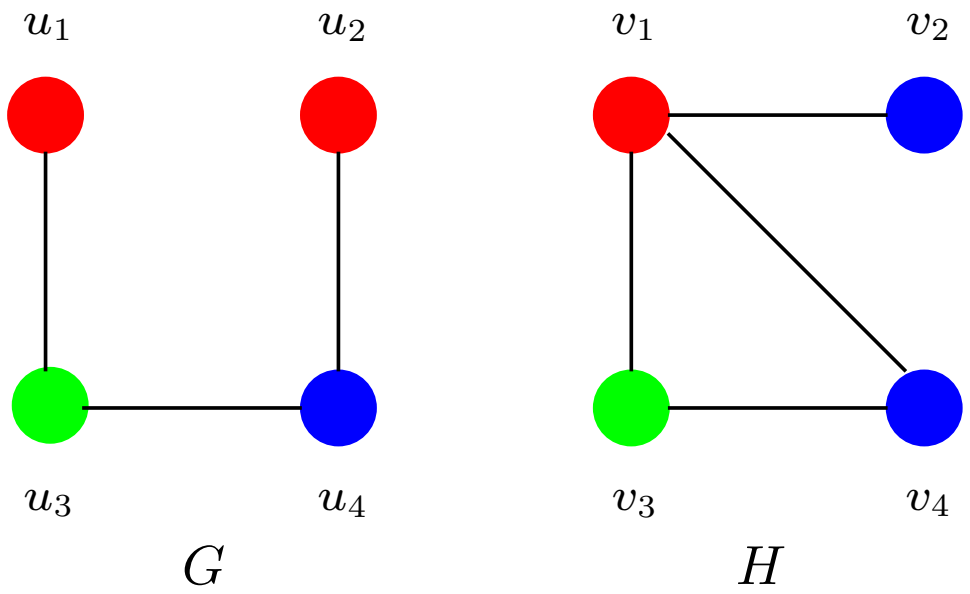
- Duplication/divergence models for the evolution of protein interaction networks
  - Interactions of duplicated proteins are also duplicated
  - Duplicated proteins rapidly lose interactions through mutations

- This provides us with a simplified basis for solving a very hard problem



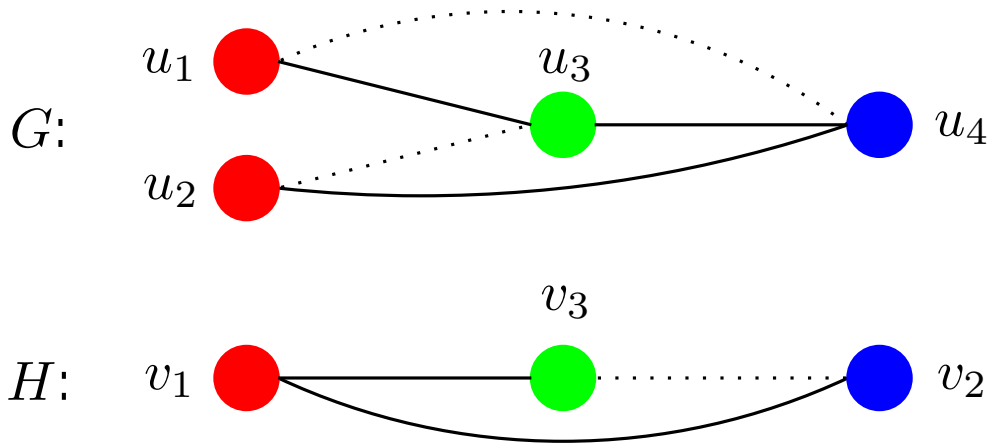
# Aligning Protein Interaction Networks: Input

- PPI networks $G(U, E)$ and $H(V, F)$
- Sparse similarity function $S(u, v)$ for all $u, v \in U \cup V$
  - If $S(u, v) > 0$, $u$ and $v$ are homologous



# Local Alignment Induced by Subsets of Proteins

- Alignment induced by protein subset pair $P = \{\tilde{U} \in U, \tilde{V} \in V\}$: $\mathcal{A}(\mathcal{P}) = \{\mathcal{M}, \mathcal{N}, \mathcal{D}\}$
  - A match $\in \mathcal{M}$ corresponds to two pairs of homolog proteins from each protein subset such that both pairs interact in both PPI networks. A match is associated with score $\mu$.
  - A mismatch $\in \mathcal{N}$ corresponds to two pairs of homolog proteins from each PPI network such that only one pair is interacting. A mismatch is associated with penalty $\nu$.
  - A duplication $\in D$ corresponds to a pair of homolog proteins that are in the same protein subset. A duplication is associated with penalty $\delta$.




Alignment induced by protein subset pair

$\{\{u_1, u_2, u_3, u_4\}, \{v_1, v_2, v_3\}\}$

# Pairwise Local Alignment of PPI networks

- Alignment score:
  $\sigma(\mathcal{A}(P)) = \sum_{M \in \mathcal{M}} \mu(M) - \sum_{N \in \mathcal{N}} \nu(N) - \sum_{D \in \mathcal{D}} \delta(D)$
  - Matches are rewarded for conservation of interactions
  - Duplications are penalized for differentiation after split
  - Mismatches are penalized for divergence and experimental error

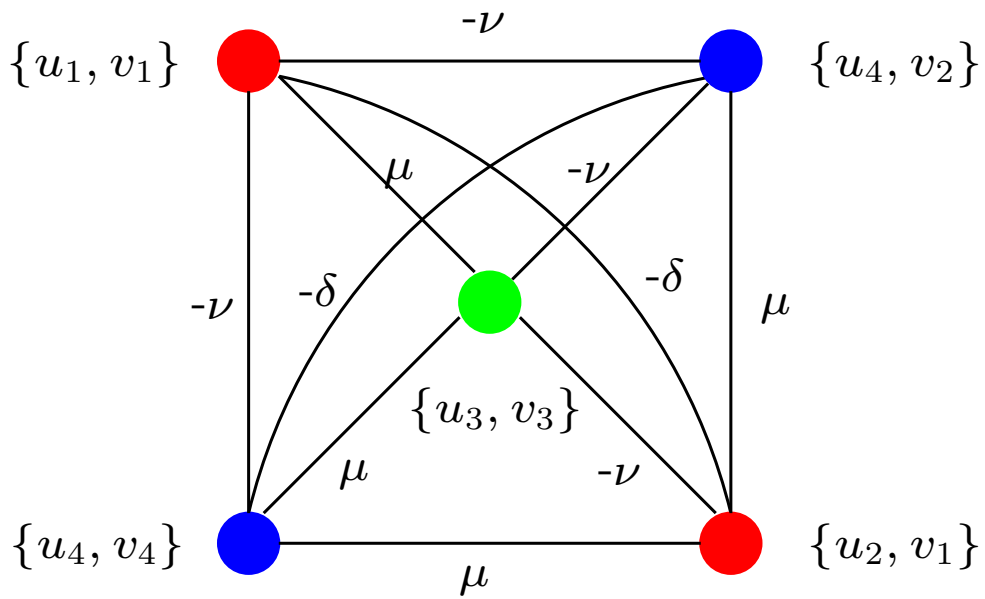
- All scores and penalties are functions of similarity between associated proteins

- Problem: Find all protein subset pairs with alignment score larger than a certain threshold.
  - High scoring protein subsets are likely to correspond to conserved modules or complexes

- A graph equivalent to BLAST

# Weighted Alignment Graph $\mathbf{G(V, E)}$

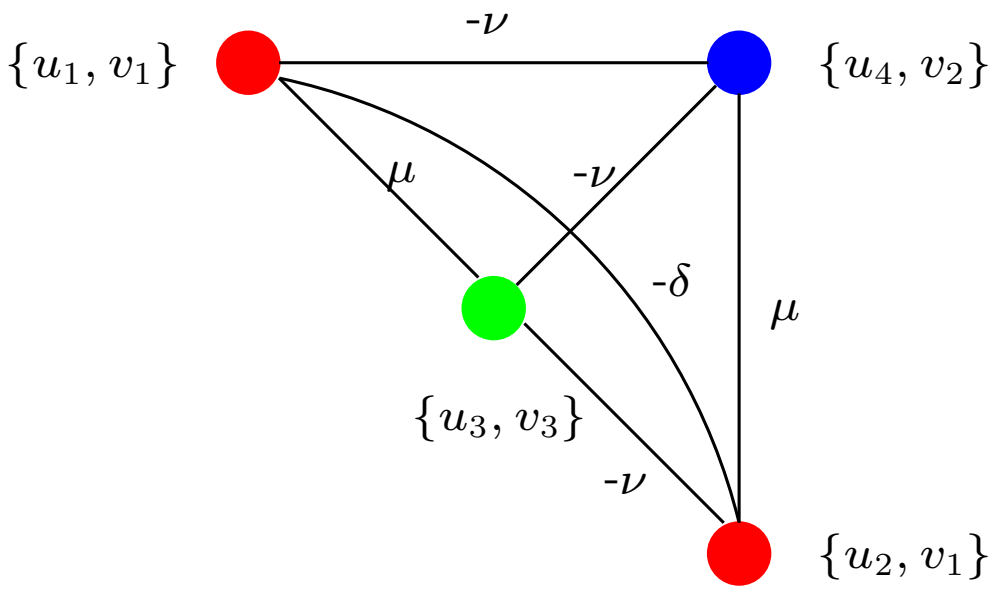
- $\mathbf{V}$ consists all pairs of homolog proteins $\mathbf{v} = \{u \in U, v \in V\}$
- An edge $\mathbf{vv'} = \{uv\}\{u'v'\}$ in $\mathbf{E}$ is a
  - match edge if $uu' \in E$ and $vv' \in V$, with weight $w(\mathbf{vv'}) = \mu(uv, u'v')$
  - mismatch edge if $uu' \in E$ and $vv' \notin V$ or vice versa, with weight $w(\mathbf{vv'}) = -\nu(uv, u'v')$
  - duplication edge if $S(u, u') > 0$ or $S(v, v') > 0$, with weight $w(\mathbf{vv'}) = -\delta(u, u')$ or $w(\mathbf{vv'}) = -\delta(v, v')$



# Maximum Weight Induced Subgraph Problem

- Definition: (MAWISH)
  - Given graph $\mathbf{G}(\mathbf{V}, \mathbf{E})$ and a constant $\epsilon$, find $\tilde{\mathbf{V}} \in \mathbf{V}$ such that $\sum_{\mathbf{v},\mathbf{u} \in \tilde{\mathbf{V}}} w(\mathbf{vu}) \geq \epsilon$.
  - NP-complete

- Theorem: (MAWISH $\equiv$ Pairwise alignment)
  - If $\tilde{\mathbf{V}}$ is a solution for the MAWISH problem on $\mathbf{G}(\mathbf{V}, \mathbf{E})$, then $P = \{\tilde{U}, \tilde{V}\}$ induces an alignment $\mathcal{A}(P)$ with $\sigma(\mathcal{A}) \geq \epsilon$ , where $\tilde{\mathbf{V}} = \tilde{U} \times \tilde{V}$.



# A Greedy Algorithm for MAWISH

- Greedy graph growing
  - Start with a heavily connected node, put it in $\tilde{\mathbf{V}}$
  - Choose $\mathbf{v}$ that is most heavily connected to $\tilde{\mathbf{V}}$ and put it in $\tilde{\mathbf{V}}$ until no $\mathbf{v}$ is positively connected to $\tilde{\mathbf{V}}$.
  - If total weight of the subgraph induced by $\tilde{\mathbf{V}}$ is greater than a threshold, return $\tilde{\mathbf{V}}$
  - Works in linear time.

- As modules and complexes are densely connected within the module and loosely connected to the rest of the network, this algorithm is expected to be effective.

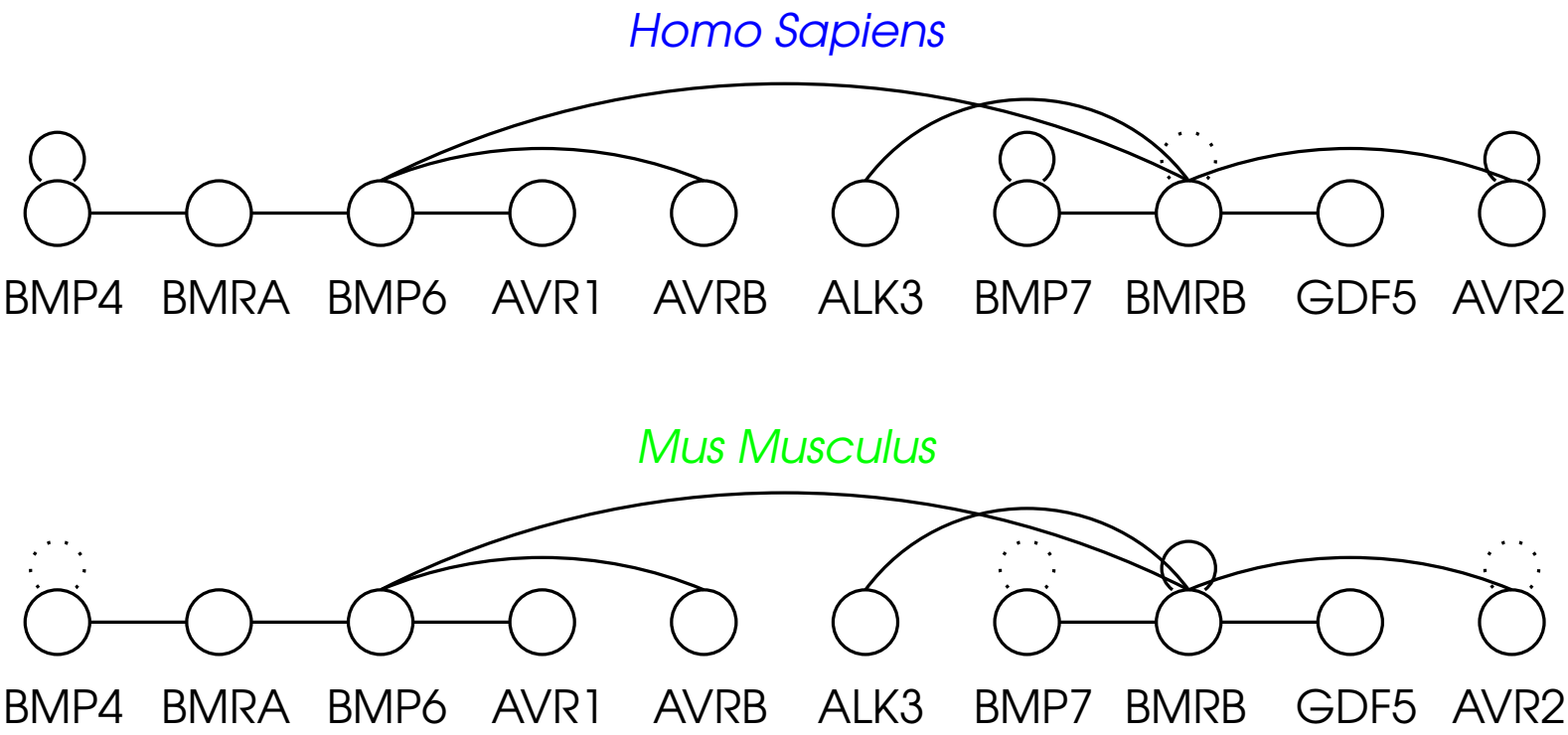
- For all local alignments, remove discovered subgraph and run the greedy algorithm again.

- If the number of homologs for each protein is constant, construction of alignment graph and solution of the MAWISH takes $O(|E| + |F|)$ time.

# Scoring Matches, Mismatches and Duplications

- Quantizing similarity between two proteins
  - Confidence in two proteins being orthologous (paralogous)
  - BLAST E-value: $S(u,v) = log_{10}\frac{p(u,v)}{p_{random}}$
  - Ortholog clustering: $S(u,v) = c(u)c(v)$
- Match score
  - $\mu(uu', vv') = \bar{\mu}\min\{S(u,v), S(u',v')\}$
- Mismatch penalty
  - $\nu(uu', vv') = \bar{\nu}\min\{S(u,v), S(u',v')\}$
- Duplication penalty
  - $\delta(u,u') = \bar{\delta}(d - S(u,u'))$

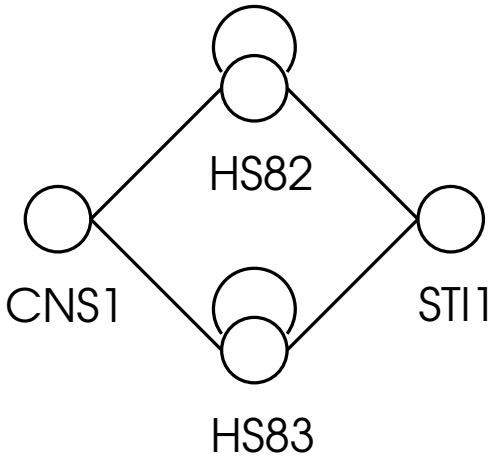
# Alignment of Human and Mouse PPI Networks



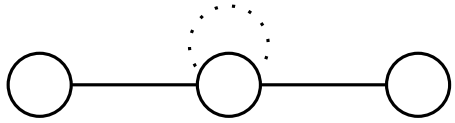

A conserved subnet that is part of
transforming growth factor beta receptor signaling pathway

# Alignment of Yeast and Fly PPI Networks

*Saccharomyces Cerevisiae*




*Drosophila Melanogaster*




A conserved subnet that is part of response to stress

# Ongoing Work on PPI Network Alignment

- Assessing statistical significance
  - Constructing a refence model based on models of evolution
- BLAST-like search queries for network alignment
  - Given a query graph, find all high-scoring local alignments in a database of PPI networks
- Multiple Graph Alignment (CLUSTAL, BLASTCLUST)
  - How to combine graph mining and pairwise alignment

# Inferring Functional Modules from Phylogenetic Information

(Kim, Koyuturk, Topkara, Grama, Subramaniam, ECCB05 (submitted))

- Functionally related proteins are likely to have co-evolved
  - Construct phylogenetic profile for each genome: Vector of E-values signifying existence of an orthologous protein in each organism
  - Identify pairwise functional associations based on mutual information between phylogenetic profiles (Pellegrini et al. (1999))
  - Mutual information:
    $I(X,Y) = H(X) - H(X|Y) = \sum_x \sum_y p(x,y) \log(p(x,y)/p(x)p(y))$
  - Shown to identify functionally associated protein pairs at a coarser level than high-throughput methods

- However, domains, not proteins, co-evolve
  - How can we incorporate domain information to enhance performance of phylogeny-based interaction prediction?

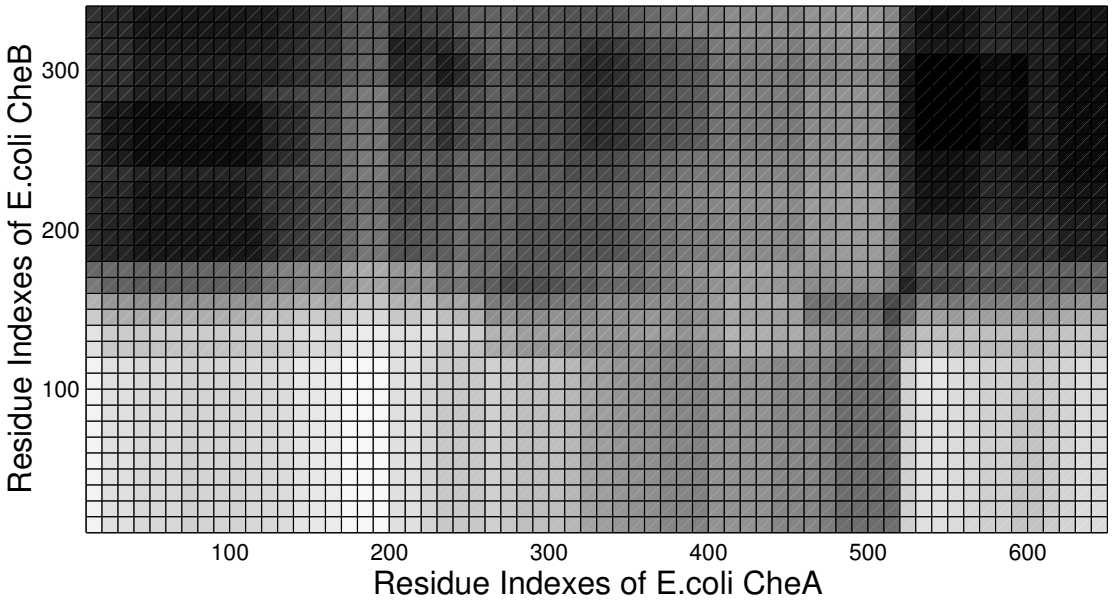
# Identification of Co-evolved Domains

- While sequence information is widely available, domain information is not generally comprehensive
- Approximating domains between fixed-size segments (Kim & Subramaniam (2004))
  - Chop proteins into overlapping (*e.g.*, 30 b.p.) fixed-size (*e.g.*, 120 b.p.) segments
  - Construct phylogenetic profile for each segment, find maximum-mutual-information segment pair for each protein pair
  - Improves single-profile based approach
  - However, there is no fixed domain size
- Can we identify domains from phylogenetic information as well?
  - Residue phylogenetic profiles!

# Residue-Level Phylogenetic Analysis

- Residue phylogenetic profile
  - For each residue $r_{ij}$ on protein $P_i$, the existence of $r_{ij}$ in genome $G_k$ is signified by the minimum e-value of alignments between $P_i$ and $G_k$ that contain $r_{ij}$
- Mutual information matrix
  - Matrix of mutual information between any pair of residues each from one protein
  - $M(P_i, P_j) = [m_{kl}]$,
    where $m_{kl} = I(profile(r_{ik}), profile(r_{jl}))$

# Mutual Information Matrix




Mutual information matrix for proteins CheA and CheB in E-coli.
Darker pixels indicate higher mutual information.

Co-evolved domains identified by dark rectangles!

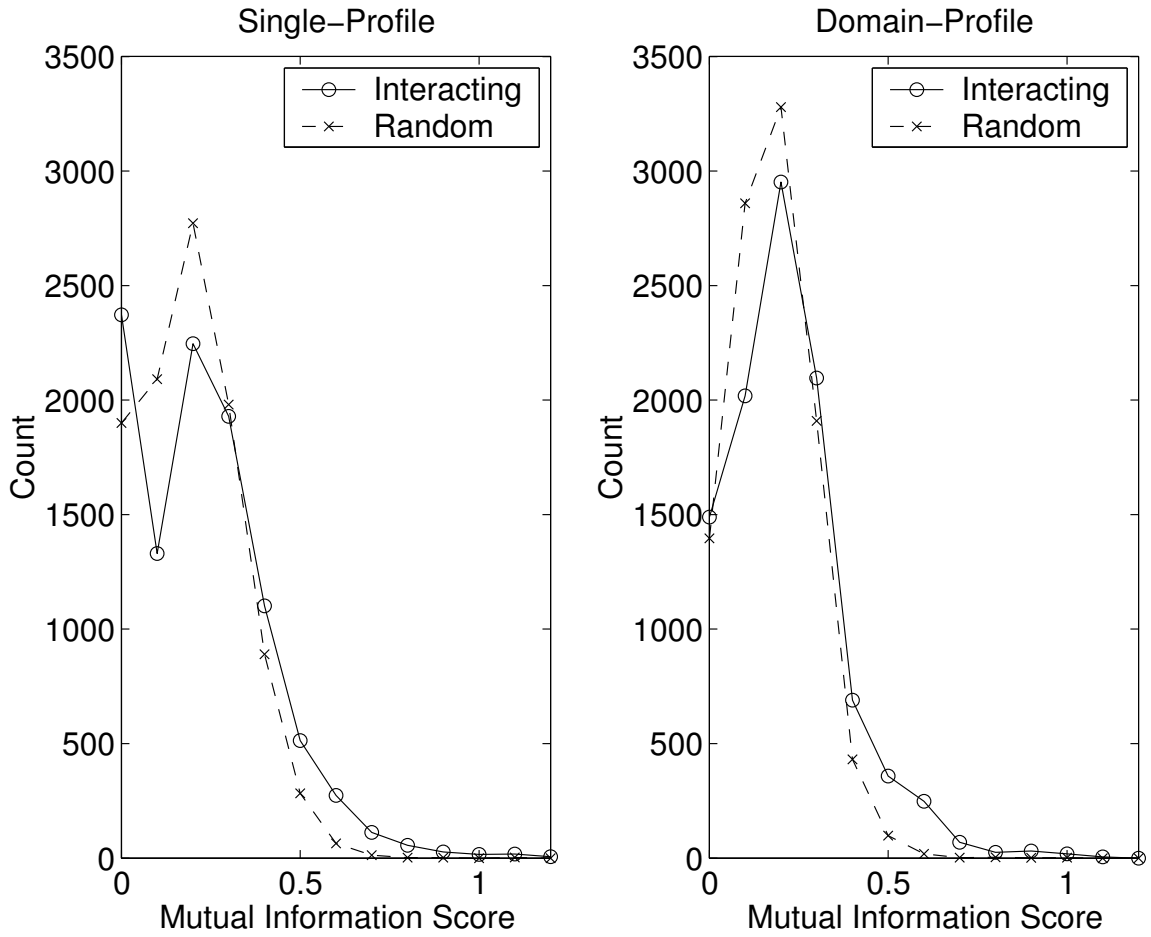
# Clustering Residue Phylogenetic Profiles

- Cluster residues to identify co-evolved domains (Kim et al., 2005)

- For each protein pair
  - Downsample residues of each protein (for computational efficiency)
  - Construct residue phylogenetic profiles
  - Compute mutual information matrix
  - Identify sufficiently large contiguous rectangles on mutual information matrix with consistently high mutual-information
  - Set phylogenetic association score of the two proteins to the maximum of mutual information of such rectangles

- Can be used for domain identification as well!

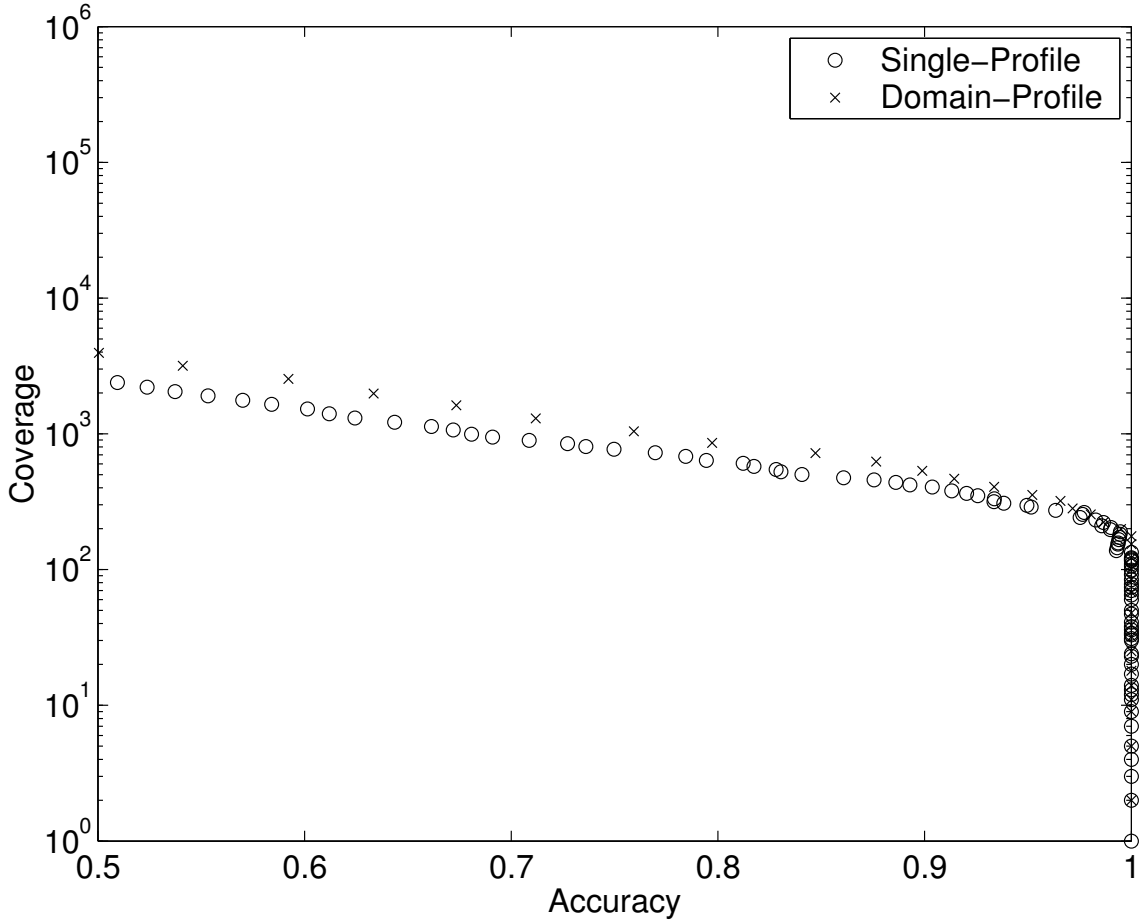
# Comparison of Domain-Profile and Single-Profile Methods




Distribution of Mutual Information

Two proteins are considered functionally associated
if they co-occur in a metabolic pathway derived from KEGG

# Comparison of Domain-Profile and Single-Profile Methods




Accuracy vs Coverage

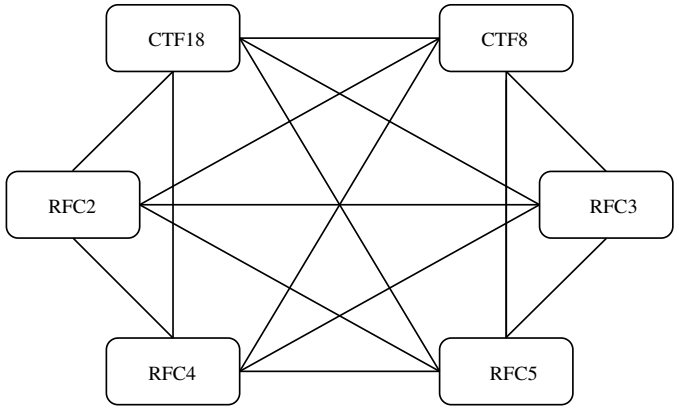
Accuracy: Fraction of true-positives among all predicted functional associations

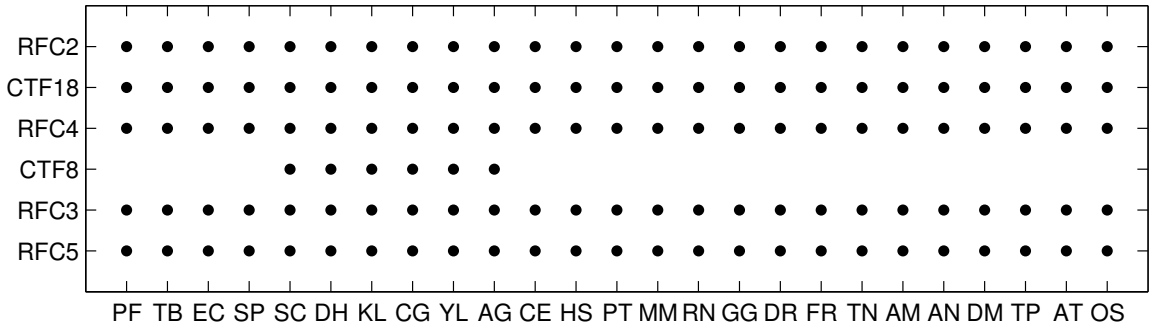
Coverage: Number of functionally associated protein pairs that are identified by the algorithm

# Augmenting PPI Networks with Phylogenetic Information to Enhance Module Identification

- Density-based clustering of PPI networks is commonly used to identify modules
  - MCODE: Greedy graph growing based on local neighborhood density of each protein
  - MCL: Markovian clustering based on random walks on PPI network
  - MCS: Recursive min-cut partitioning
- Information derived from PPI networks is not comprehensive
  - High-throughput methods are prune to false-negatives and even false-positives
  - Available data is generated for target functional units in cell (*e.g.*, fly network mostly contains signaling information)

# Functional Modules and Phylogeny

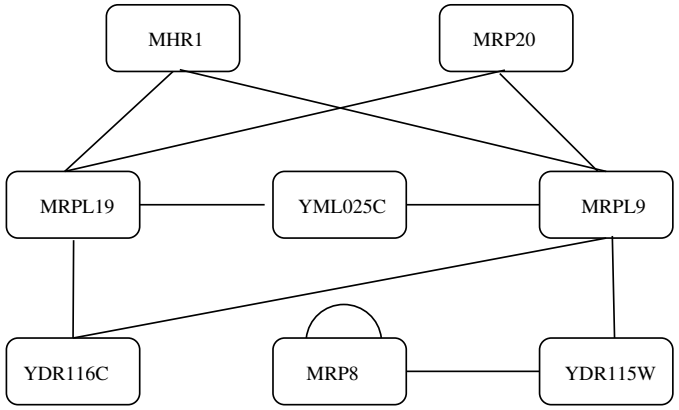


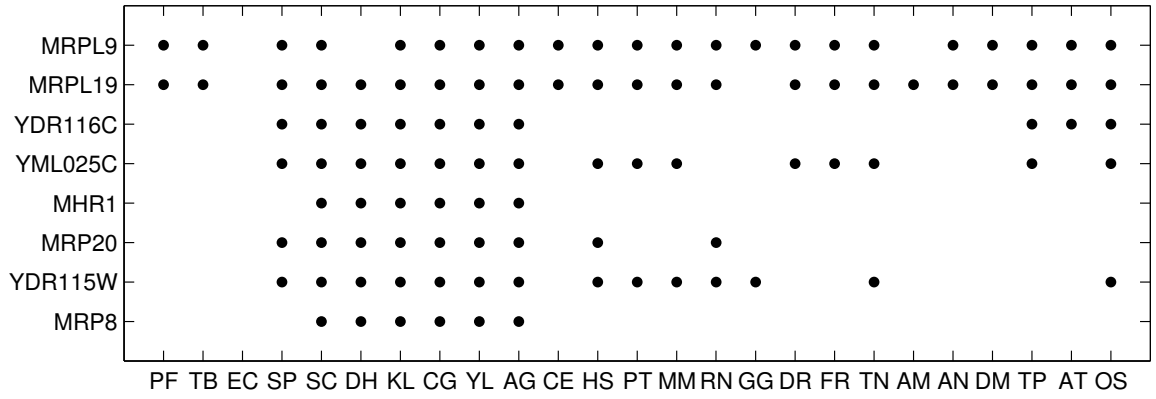




Replication Factor C complex identified on yeast PPI network by MCODE algorithm and the phylogenetic profiles of its proteins on 25 eukaryotic genomes

Conserved in all eukaryotic species!

# Functional Modules and Phylogeny

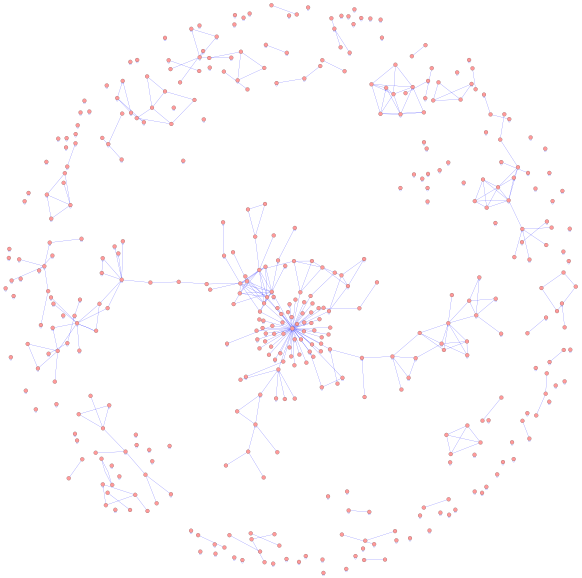


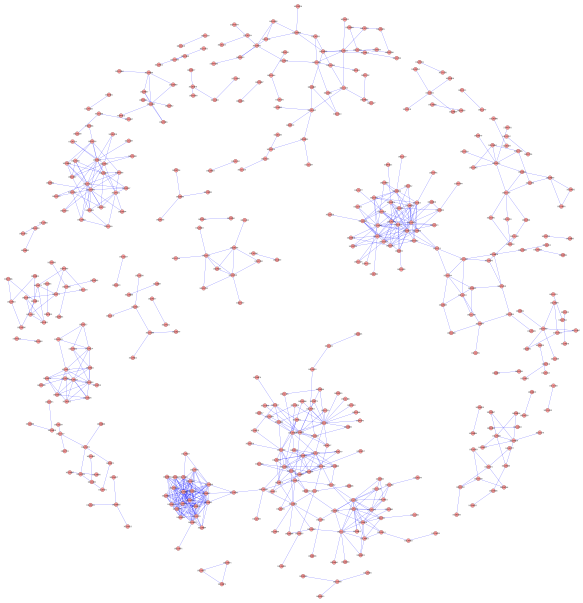

A component of mitochondrial ribosome identified on yeast PPI network by MCODE algorithm and the phylogenetic profiles of its proteins on 25 eukaryotic genomes

Conserved in only yeast species!

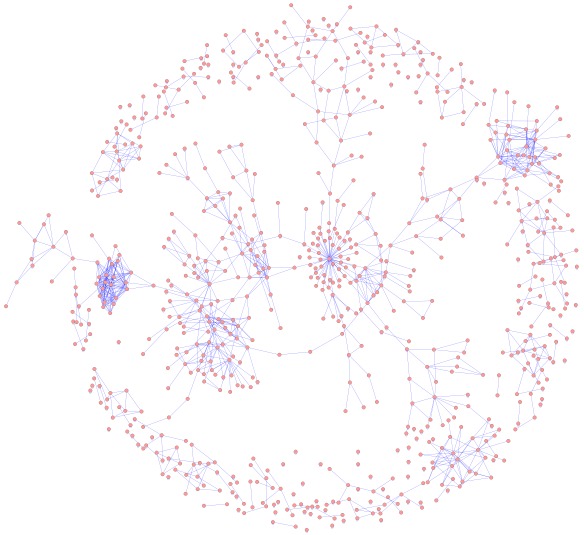
# Superposing PPI Networks with Phylogenically Predicted Networks



Protein Interaction (PPI) network



Phylogenetic Association (PGA) network



Superposition of PPI and PGA networks

Cluster the superposed network!

# Clustering The Superposed Network

- MCODE discovers
  - 15 modules on E-coli PPI network
  - 11 modules on E-coli PGA network constructed by domain-profile method
  - 26 modules on PPI ∪ PGA

| PPI | PGA | PPI ∪ PGA |
|---|---|---|
| 4 proteins<br>9 interactions | 4 proteins<br>6 phylogenetic associations | 9 proteins<br>21 functional associations |
| molybdopterin biosynthesis<br>molybdochetalase<br>MGD biosynthesis B<br>molybdopterin → MGD | molybdopterin biosynthesis<br>molybdopterin biosynthesis A<br>molybdopterin biosynthesis C<br>DMSO reductase | molybdopterin biosynthesis<br>molybdochetalase<br>MGD biosynthesis B<br>molybdopterin → MGD<br>molybdopterin biosynthesis A<br>molybdopterin biosynthesis C<br>DMSO reductase<br>molybdopterin biosynthesis B<br>anaerobic DMSO reductase |

A module of the molybdopterin biosynthesis pathway in Ecoli that can only be partially discovered on PPI and PGA networks is comprehensively identified on the augmented network

# Building a Comprehensive Signaling Database

There are four major components to any such effort:

- The availability of up-to-date, curated/annotated signaling data (The Biology WorkBench provides us with an excellent starting point. ITaP is in the process of mirroring the WorkBench at Purdue.

- Developing commonly accepted (flexible, extensible) data standards. These do not exist in the signaling community at this point, although, SBML addresses a closely related community.

- Developing analysis techniques – we are one of the leading groups in this area.

- Developing interfaces and middleware – this presents a significant opportunity for development.

# Building a Comprehensive Signaling Database

Any such effort must:

- Interoperate with other existing signaling databases in terms of data formats, APIs for services, and standardized output formats.
- Interoperate with existing genotype and phenotype tools and databases.
- Provide support for building complex tools that build on a variety of existing data sources and APIs.

# Using a Signaling Database: An Example (1)

Consider the problem of predicting protein interactions using phylogeny data. The algorithm builds on the following sources:

- Sequence data for generating phylogenetic profiles.
- BLAST for finding matches (populating the phylogenetic profiles).
- In-house analyses on generating phylogeny vectors and inferring interactions.
- DIP for validating results.

Our current study downloads all the data and analysis tools (including BLAST) and performs analyses locally. This is extremely cumbersome and inaccurate, since databases are always in a state of flux.

# Using a Signaling Database: An Example (2)

Consider the problem of detecting modules in protein interaction networks using interaction and phylogeny data. The algorithm builds on the following sources:

- Sequence data and BLAST for generating profile based hyper-edges.
- DIP for protein interactions.
- In-house code for analysis and augmentation of interaction networks.
- MCODE/MCL/MCS for graph clustering.
- A variety of online sources for validation.

As before, even simple analyses tasks can be extremely data, effort, and time intensive.

# Building a Comprehensive Signaling Database: Challenges

- Interfaces for data and API conversion.
- Analysis modules for interaction data (harden existing analyses tools to production quality, develop new tools).
- Tools to enable submission of other tools to the infrastructure (controlled vocabularies, ontologies, etc.).
- Service discovery service.
- Support for building applications (composing applications, type-checking, consistency).
- Runtime system for RPC, debugging, and visualization.