Large-Scale Dynamic Networks

Ananth Grama
Center for Science of Information and Computer Science
Purdue University

Thanks to the US National Science Foundation.

Overview and Motivation

- Dynamic networks are one of the most important forms of “Big Data”.

- Virtually all network data is dynamic.
  - Networks of social and economic transactions.
  - Interactions in biological, physical, and engineered systems.
  - Traffic, connectivity, and dependencies in computer/ communications systems.

- Analysing dynamic networks poses significant and diverse challenges.

Characteristics of Dynamic Networks

Instances of dynamic networks of interest are typically:

- Large – scaling to billions of nodes and interactions.
- Noisy – with high rates of false positives and negatives.
- Multiscale – incorporating interactions at vastly different levels of abstractions.
- Heterogeneous – demonstrating high variability in characteristics over space and time. Significant skews in degree distribution.
- Distributed – data is typically collected and stored at distributed locations.
- Elastic – data is typically elastic.

Analysis of Dynamic Networks

Analysis techniques for dynamic networks must:

- Rely on suitable formulations – results are typically probabilistic. Formulations must quantify (and optimize) significance (statistical).
  - Deterministic formulations on noisy data are not meaningful.
  - Distribution agnostic formulations (say, based on simple counts and frequencies) are unlikely to work.
- Provide rigorously validated solutions – garbage-in, garbage-out at scale.
- Must have efficient elastic distributed implementations (MapReduce type frameworks have considerable issues with semantics, scope, and overhead).

These issues form the focus of our current research efforts in the area.

Dynamic Network Analysis – Problems (1)

- Characterization and Modeling of Dynamic State. Study data-driven dynamic networks and characterize the evolution at micro- as well as macro-scale. This includes node-, link-, aggregate-, and network models.

- Mutual Information, Conservation. Models and methods for determining conserved information in a set of networks states and its relation to overall network dynamics.

- Discriminant Analysis. Track evolution as a sequence of discriminants across network snapshots.

- Spatio-Temporal Motifs. Define recurring patterns in both space and time.

Dynamic Network Analysis – Problems (2)

- Prediction of Network State. Predicting network state at micro- and macro-scales. This is an essential aspect of resource allocation and provisioning.
- Noise, Robustness, and Approximations. Study the impact of noise and approximation on our models and methods.
- Compression and Representation. Develop provably optimal compression and representation schemes for dynamic networks.

Models for Dynamic Networks

Models for dynamic networks provide a means for generating networks of arbitrary size and well-parametrized characteristics. Models have limited predictive capability, however:

- Models play a critical role in analyses, by providing a prior. Traditional analytics methods do a poor job here.
- Models allow analytic methods for estimating significance of results.
- Models can be used for validation.
- Models allow coarse-grain understanding of fluxes in networks.

Models for Dynamic Networks

Models for dynamic networks are in relative infancy. Generation models for static networks are often viewed as pseudo-dynamic models.

- Erdos-Renyi, Preferential Attachment, and Copying models.
- Community guided attachment and forest fire.
- Kronecker graphs.
- Microscopic models.

There have also been some true dynamic generation models, most notably the node time-series correlation model. Developing a class of true dynamic models that lend themselves to analytic methods remains an open question.

Analytics on Dynamic Networks: Conservation

Given a sequence of networks, identify sub-networks that are (statistically) significantly conserved over evolution trajectories. This poses several problems from points of view of modeling and method development:

- Model selection. Models must be true to priors, while being amenable to analytical quantification of significance.
- Ideally, significance cutoff should be an analytics parameter. Valid methods must identify all sub-networks that exceed this significance threshold. There are no known methods capable of solving this problem even at small scale (let alone large trajectories over large graphs).
- Conservation over longitudinal/ horizontally partitioned trajectories each pose challenges for distributed computations.

Analytics on Dynamic Networks: Discriminant Analysis

When does a network significantly diverge from the model? What are components responsible for this divergence? This is sometimes also called *break* analysis or change analysis.

- This poses a computational problem known to be NP-Hard.
- Approximations for different models must be developed and their performance quantified.
- Must deal with overfitting and noise.

Data Management and Graph Compression

Develop compression schemes that lend themselves to indexing and analytics in compressed form.

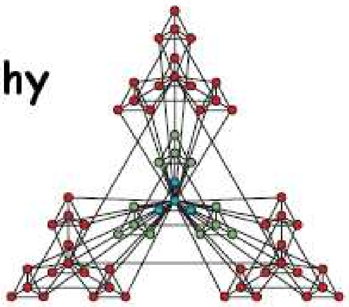
- Generalize our notion of graph entropy to temporal domain.
- Integrate graph rewriting/ grammars with compression/ indexing.
- Develop distortion measures for lossy compression to deal with inherent noise in data.
- Develop compression algorithms to operate at scale.

Part 2: Prior Results in the Area

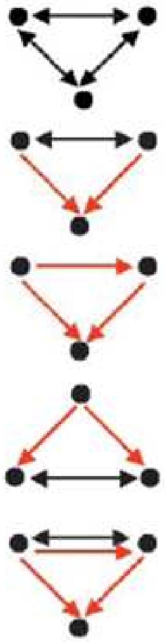
Function & Topology in Networks

How does function relate to network topology?

Hierarchy



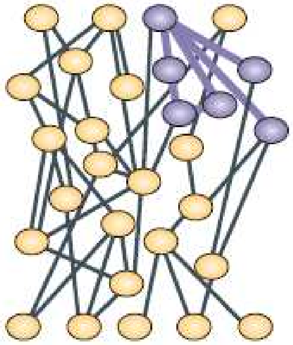
Robustness



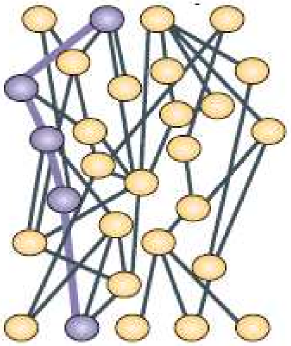
Motifs

Modularity

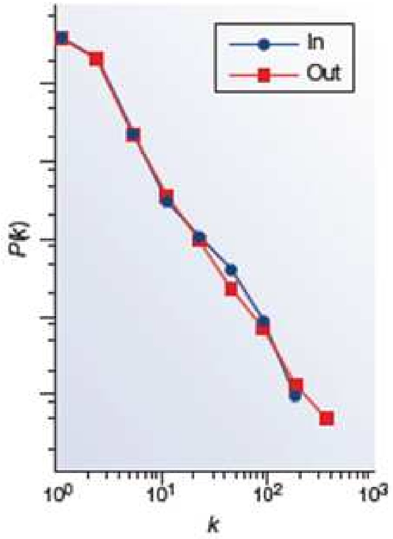
Emergence



Modules



Pathways



Prior Work on Topology and Function

- Conservation (ISMB 04/Bioinf. 04)
- Alignment (RECOMB 05/JCB 06)
- Modularity (RECOMB 06/JCB 07)
- Inference (Bioinf. 06)
- Pathway Annotation (ISMB 07/Bioinf. 07, PSB 08)
- Network Abstractions/ Annotations (ECCB 08/ Bioinf. 08)
- Modularity and Domain Interactions (APBC 10/ BMC Bioinf. 10)
- Pathway Interaction Maps (PSB 12)
- Pathway Inference (ISMB 12)

Evolution of Interactions

- "Evolution thinks modular" (Vespignani, *Nature Gen.*, 2003)
- Cooperative tasks require all participating units
  - Selective pressure on preserving interactions & interacting proteins
- Nodes organized in cohesive patterns are highly conserved (Wuchty et al., *Nature Gen.*, 2003)
  - Functional modules are likely to be consistently conserved
- Orthologs of interacting nodes are likely to interact (Wagner, *Mol. Bio. Evol.*, 2001)
  - Conservation of interactions may provide clues on conservation of function
- Interacting nodes follow similar evolutionary trajectories (Pellegrini et al., *PNAS*, 1999)

Computational Analysis of Biological Networks

- Clustering
  - Interaction network: Proteins in functional modules densely interact with each other
  - Gene expression: Genes coding cooperating proteins are likely to be co-regulated
  - Phylogenetic profiles: Interacting proteins are likely to have co-evolved
- Graph Mining
  - Common topological motifs and frequent interaction patterns reveal conserved modularity
- Graph Alignment
  - Conservation/divergence of pathways, complexes, and functional modules

Frequent Interaction Patterns: Computational Problem

- Given a set of proteins V a set of interactions E, and a many-to-many mapping from V to a set of ortholog groups $\mathcal{L} = \{l_1, l_2, ..., l_n\}$, the corresponding interaction network is a labeled graph $G = (V, E, \mathcal{L})$.
  - $v \in V(G)$ is associated with a set of ortholog groups $L(v) \subseteq \mathcal{L}$.
  - $uv \in E(G)$ represents an interaction between u and v.

- S is a sub-network of G, *i.e.*, $S \sqsubseteq G$ if there is an injective mapping $\phi : V(S) \to V(G)$ such that for all $v \in V(S)$, $L(v) \subseteq L(\phi(v))$ and for all $uv \in E(S)$, $\phi(u)\phi(v) \in E(G)$.

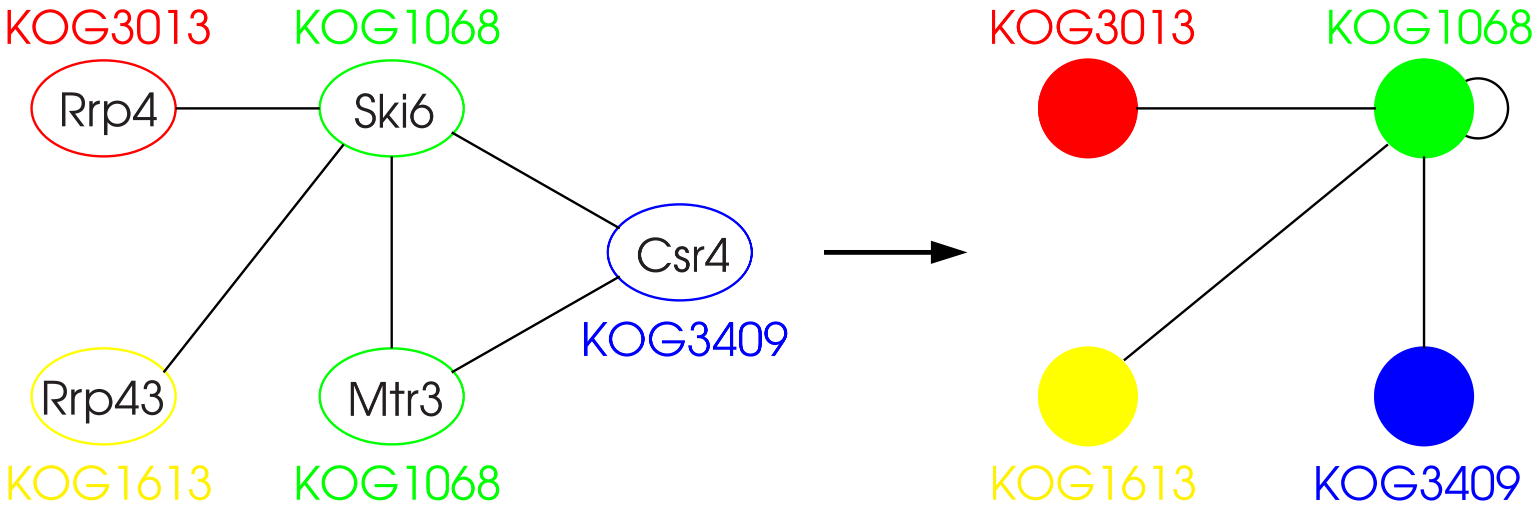
- Maximal frequent sub-network discovery
  - Instance: A set of interaction networks $\mathcal{G} = \{G_1 = (V_1, E_1, \mathcal{L}), G_2 = (V_2, E_2, \mathcal{L}), ..., G_m = (V_m, E_m, \mathcal{L})\}$, each belonging to a different organism, and a frequency threshold σ^*.
  - Problem: Let $H(S) = \{G_i : S \sqsubseteq G_i\}$ be the occurrence set of graph S. Find all connected subgraphs S such that $|H(S)| \geq \sigma^*$, *i.e.*, S is a frequent subgraph in $\mathcal{G}$ and for all $S' \sqsupset S$, $H(S) \neq H(S')$, *i.e.*, S is maximal.

Ortholog Contraction

- Contract orthologous nodes into a single node
- No subgraph isomorphism
  - Graphs are uniquely identified by their edge sets
- Frequent sub-networks are preserved ⇒ No information loss
  - Sub-networks that are frequent in general graphs are also frequent in their ortholog-contracted representation
- Discovered frequent sub-networks are still biologically interpretable!
  - Interaction between proteins becomes interaction between ortholog groups
  - Ortholog-contraction may be thought of as going back in evolutionary history (to what point?)

Ortholog Contraction in PPI Networks

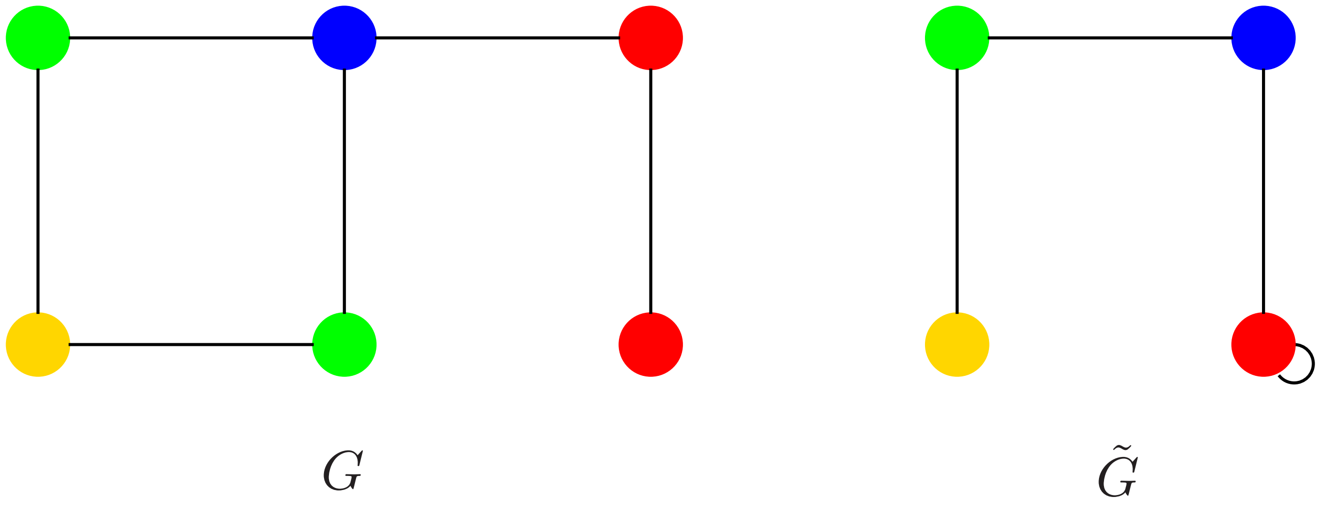
- Interaction between proteins → Interaction between ortholog groups or protein families



Preservation of Sub-networks

Theorem: Let $\tilde{G}$ be the ortholog-contracted graph obtained by contracting the orthologous nodes of network G. Then, if S is a subgraph of G, $\tilde{S}$ is a subgraph of $\tilde{G}$.

Corollary: The ortholog-contracted representation of any frequent sub-network is also frequent in the set of ortholog-contracted graphs.



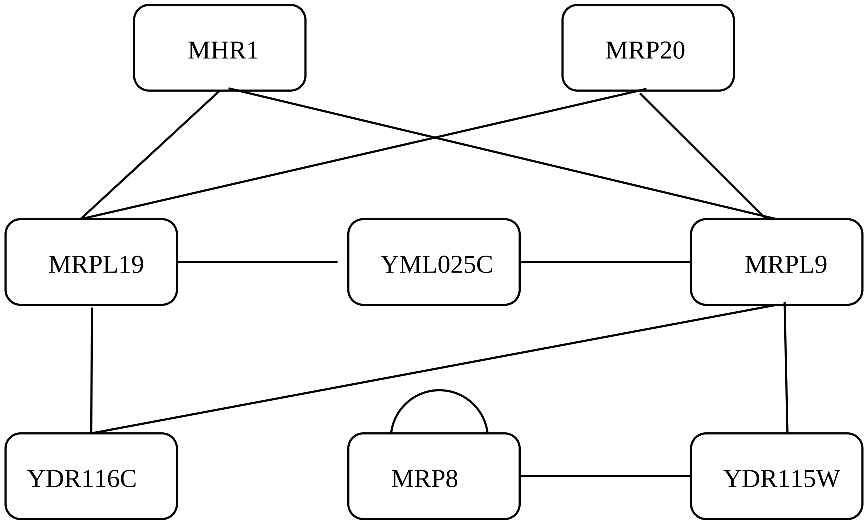
Results: Mining PPI Networks

- PPI networks for 9 eukaryotic organisms derived from BIND and DIP
  - *A. thaliania, O. sativa, S. cerevisiae, C. elegans, D. melanogaster, H. sapiens, B. taurus, M. musculus, R. norvegicus*
  - # of proteins ranges from 288 (*Arabidopsis*) to 8577 (*fruit fly*)
  - # of interactions ranges from 340 (*rice*) to 28829 (*fruit fly*)

- Ortholog contraction
  - Group proteins according to existing COG ortholog clusters
  - Merge Homologene groups into COG clusters
  - Cluster remaining proteins via BLASTCLUST
  - Ortholog-contracted *fruit fly* network contains 11088 interactions between 2849 ortholog groups

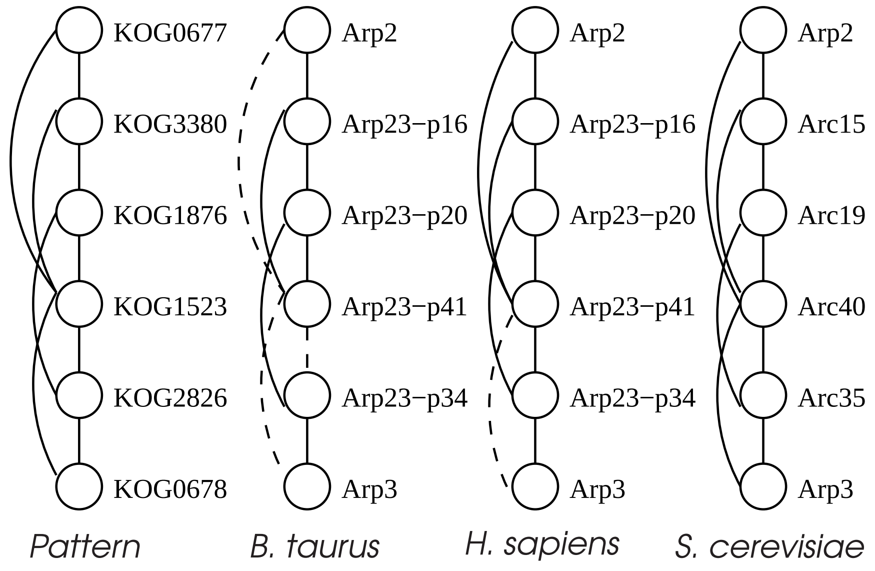
- MULE is available at `http://www.cs.purdue.edu/homes/koyuturk/mule/`

Frequent Protein Interaction Patterns



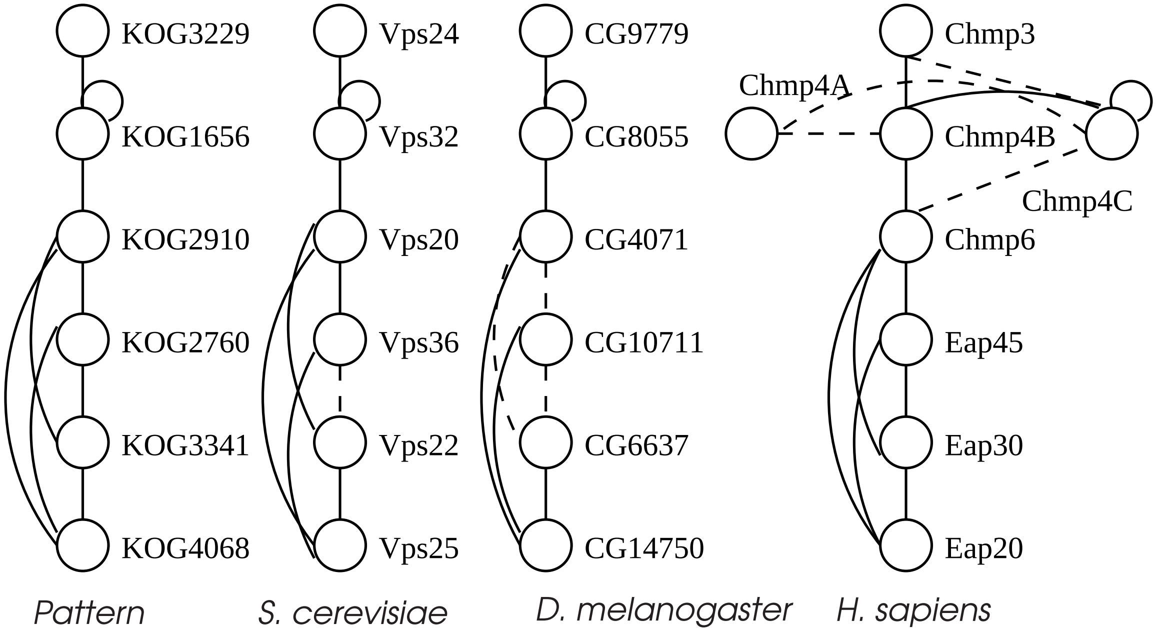
Small nuclear ribonucleoprotein complex ($p < 2e - 43$)

Frequent Protein Interaction Patterns



Actin-related protein Arp2/3 complex ($p < 9e - 11$)

Frequent Protein Interaction Patterns



Endosomal sorting ($p < 1e - 78$)

Modular Phylogenetics

- Top eight groups of three organisms that contain most frequent connected sub-networks and interactions

Organism set	# frequent sub-networks	# frequent interactions
C. elegans, D. melanogaster, H. sapiens	8	134
S. cerevisiae, D. melanogaster, H. sapiens	20	126
D. melanogaster, H. sapiens, M. musculus	17	86
S. cerevisiae, C. elegans, D. melanogaster	15	77
S. cerevisiae, C. elegans, H. sapiens	6	50
S. cerevisiae, H. sapiens, M. musculus	10	26
C. elegans, H. sapiens, M. musculus	5	23
H. sapiens, M. musculus, R. norvegicus	10	23

Runtime Characteristics

Comparison with isomorphism-based algorithms

FSG (Kuramochi & Karypis, *ICDM*, 2001), gSpan (Yan & Han, *KDD*, 2003)

		FSG			MULE		
Dataset	Minimum Support (%)	Runtime (secs.)	Largest pattern	Number of patterns	Runtime (secs.)	Largest pattern	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 patterns

Glutamate metabolism, $\sigma = 8\%$				Alanine metabolism, $\sigma = 10\%$			
Size of contracted pattern	Extraction time (secs.)		Size of extracted pattern	Size of contracted pattern	Extraction time (secs.)		Size of extracted pattern
	FSG	gSpan			FSG	gSpan	
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.			

Pairwise Alignment of PPI Networks

- Given two PPI networks that belong to two different organisms, identify sub-networks that are similar to each other
  - Biological meaning
  - Mathematical modeling

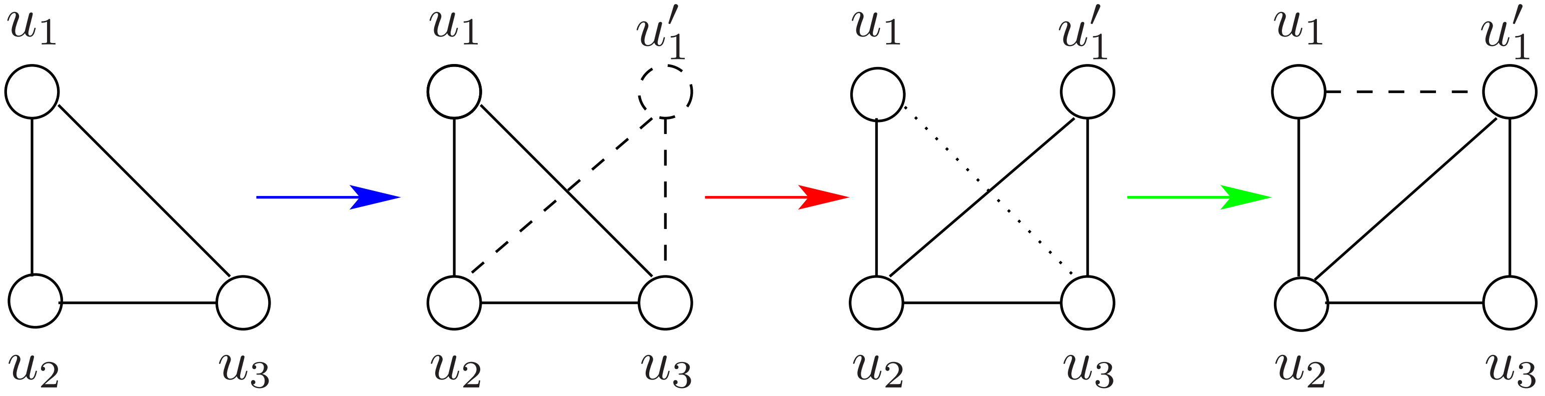
- Existing algorithms
  - PathBLAST aligns pathways (linear chains) to simplify the problem while maintaining biological meaning (Kelley et al., *PNAS*, 2004)
  - NetworkBLAST compares conserved complex model with null model to identify significantly conserved subnets (Sharan et al., *J. Comp. Biol.*, 2005)

- Our approach (Koyutürk, Kim, Topkara, Subramaniam, Szpankowski, & Grama, *J. Comp. Biol.*, 2006)
  - Guided by models of evolution
  - Scores evolutionary events
  - Identifies sets of proteins that induce high-scoring sub-network pairs

Evolution of PPI 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

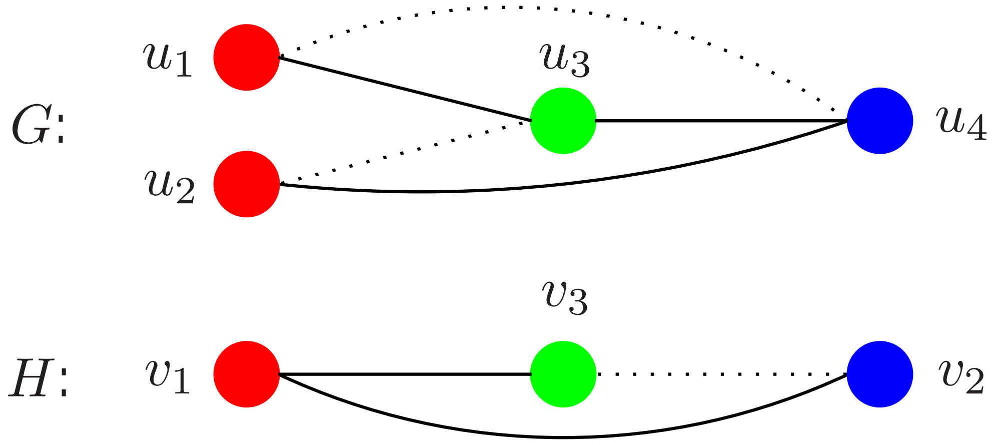
- Allows defining and scoring evolutionary events as graph-theoretical concepts



Duplication Elimination Emergence

Match, Mismatch, and Duplication

- Evolutionary events as graph-theoretic concepts
  - A match $\in \mathcal{M}$ corresponds to two pairs of homolog proteins from each organism such that both pairs interact in both PPIs. A match is associated with score μ.
  - A mismatch $\in \mathcal{N}$ corresponds to two pairs of homolog proteins from each organism such that only one pair is interacting. A mismatch is associated with penalty ν.
  - A duplication $\in D$ corresponds to a pair of homolog proteins that are in the same organism. A duplication is associated with score δ.



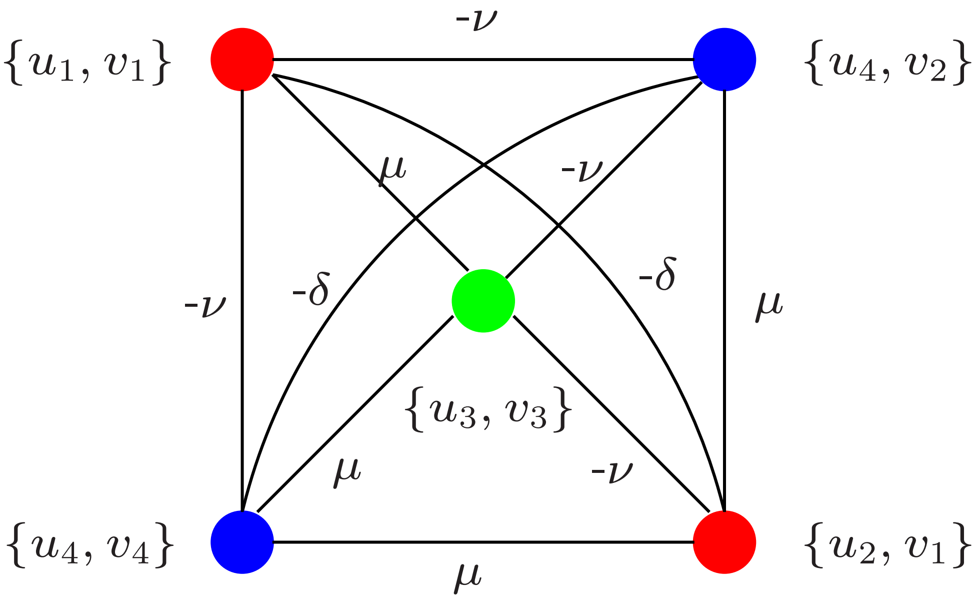
Pairwise Alignment of PPIs as an Optimization Problem

- 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 rewarded/penalized for functional conservation/differentiation after split
  - Mismatches are penalized for functional divergence (what about experimental error?)
- Scores are functions of similarity between associated proteins
- Problem: Find all protein subset pairs with significant alignment score
  - High scoring protein subsets are likely to correspond to conserved modules
- A graph equivalent to BLAST

Weighted Alignment Graph

- $\mathbf{G}(\mathbf{V},\mathbf{E})$: $\mathbf{V}$ consists of 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

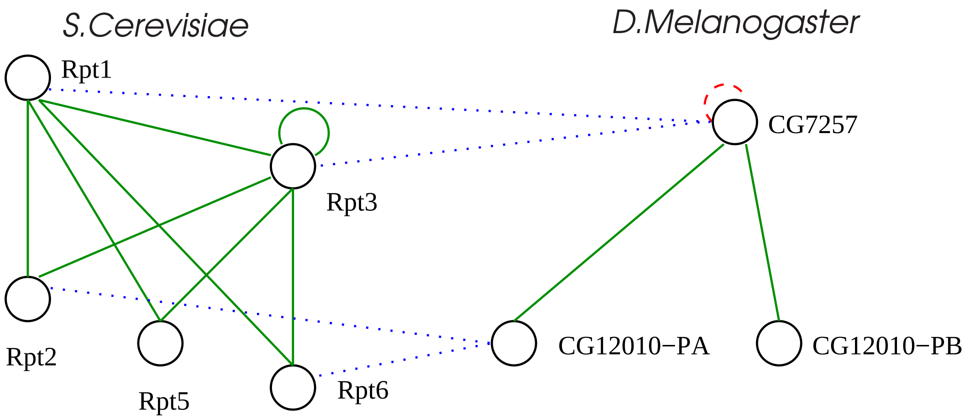
- Definition: (MAWISH)
  - Given graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and a constant ϵ, find $\tilde{\mathcal{V}} \in \mathcal{V}$ such that $\sum_{\mathbf{v},\mathbf{u} \in \tilde{\mathcal{V}}} w(\mathbf{vu}) \geq \epsilon$.
  - NP-complete
- Theorem: (MAWISH $\equiv$ Pairwise alignment)
  - If $\tilde{\mathcal{V}}$ is a solution for the MAWISH problem on $\mathcal{G}(\mathcal{V}, \mathcal{E})$, then $P = \{\tilde{U}, \tilde{V}\}$ induces an alignment $\mathcal{A}(P)$ with $\sigma(\mathcal{A}) \geq \epsilon$, where $\tilde{\mathcal{V}} = \tilde{U} \times \tilde{V}$.
- Solution: Local graph partitioning
  - Greedy graph growing + iterative refinement
  - Linear-time heuristic
- Source code available at http://www.cs.purdue.edu/homes/koyuturk/mawish/

Alignment of Yeast and Fruit Fly PPI Networks

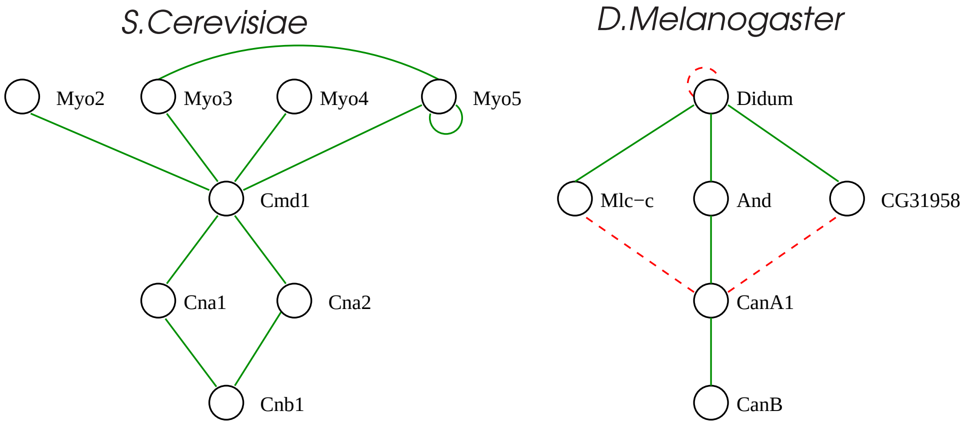
Rank	Score	z-score	# Proteins	# Matches	# Mismatches	# Dups.
1	15.97	6.6	18 (16, 5)	28	6	(4, 0)
	protein amino acid phosphorylation (69%)					
	JAK-STAT cascade (40%)					
2	13.93	3.7	13 (8, 7)	25	7	(3, 1)
	endocytosis (50%) / calcium-mediated signaling (50%)					
5	8.22	13.5	9 (5, 3)	19	11	(1, 0)
	invasive growth (sensu Saccharomyces) (100%)					
	oxygen and reactive oxygen species metabolism (33%)					
6	8.05	7.6	8 (5, 3)	12	2	(0, 1)
	ubiquitin-dependent protein catabolism (100%)					
	mitosis (67%)					
21	4.36	6.2	9 (5, 4)	18	13	(0, 5)
	cytokinesis (100%, 50%)					
30	3.76	39.6	6 (3, 5)	5	1	(0, 6)
	DNA replication initiation (100%, 80%)					

Subnets Conserved in Yeast and Fruit Fly

Proteosome regulatory particle subnet



Calcium-dependent stress-activated signaling pathway



Statistical Significance of Modularity

- Existing techniques
  - Mostly computational (*e.g.*, Monte-Carlo simulations)
  - Compute probability that the pattern exists rather than a pattern with the property (*e.g.*, size, density) exists
  - Overestimation of significance

- Random graph models
  - PPI networks generally exhibit power-law property (or exponential, geometric, etc.)
  - Analysis simplified through independence assumption
  - Independence assumption may cause problems for networks with arbitrary degree distribution
  - $P(uv \in E) = d_u d_v / |E|$, where d_u is expected degree of u, but generally $d^2_{\max} > |E|$ for PPI networks

- Analytical techniques based on simplified models (Koyutürk, Grama, & Szpankowski, *RECOMB*, 2006)

Significance of Dense Subgraphs

- A subnet of r proteins is said to be ρ-dense if $F(r) \geq \rho r^2$, where $F(r)$ is the number of interactions between these r proteins

- What is the expected size of the largest ρ-dense subgraph in a random graph?
  - Any ρ-dense subgraph with larger size is statistically significant!

- $G(n,p)$ model
  - n proteins, each interaction occurs with probability p
  - Simple enough to facilitate rigorous analysis
  - If we let $p = d_{\max}/n$, largest ρ-dense subgraph in $G(n,p)$ stochastically dominates that in a graph with arbitrary degree distribution

Largest Dense Subgraph

- Theorem: If G is a random graph with n nodes, where every edge exists with probability p, then

$$\lim_{n\to\infty} \frac{R_\rho}{\log n} = \frac{1}{\kappa(p,\rho)} \qquad (pr.), \tag{1}$$

where

$$\kappa(p,\rho) = \rho \log \frac{\rho}{p} + (1-\rho) \log \frac{1-\rho}{1-p}. \tag{2}$$

More precisely,

$$P(R_\rho \geq r_0) \leq O\left(\frac{\log n}{n^{1/\kappa(p,\rho)}}\right), \tag{3}$$

where

$$r_0 = \frac{\log n - \log\log n + \log \kappa(p,\rho)}{\kappa(p,\rho)} \tag{4}$$

for large n.

Generalizing Results to Complex Models

- Piecewise $G(n, p)$ model
  - Few proteins with many interacting partners, many proteins with few interacting partners
  - Captures the basic characteristics of PPI networks
  - The size of largest dense subgraph is still proportional to $\log n$

- More general models
  - Increasing the number of pieces, we approach models with characteristic degree distributions
  - Analysis of power-law graphs in progress

- Multiple networks: Conservation
  - Superpose graphs based on sequence homology

Algorithms Based on Statistical Significance

- Identification of topological modules

- Use statistical significance as a stopping criterion for graph clustering heuristics

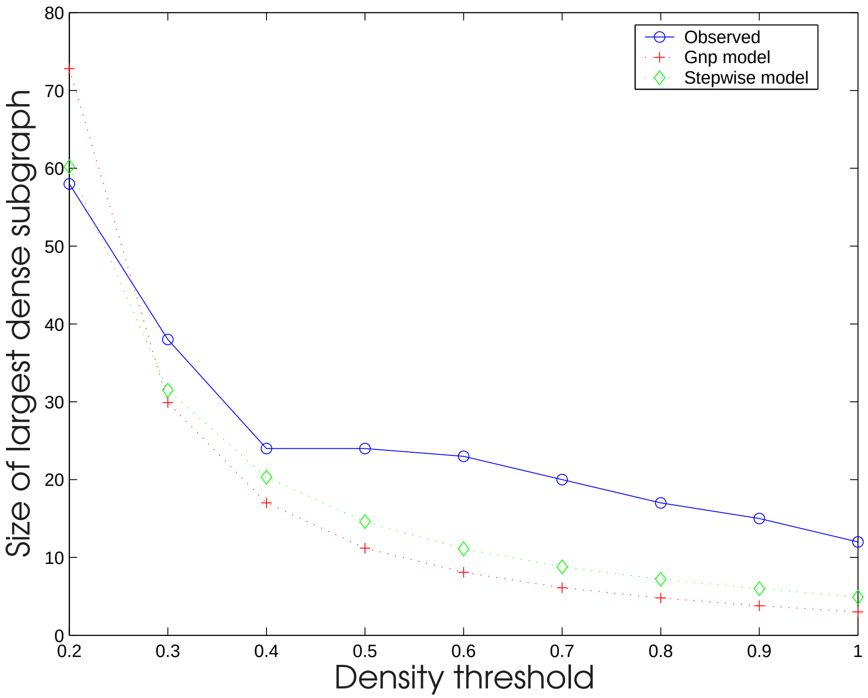
- HCS Algorithm (Hartuv & Shamir, *Inf. Proc. Let.*, 2000)

  - Find a minimum-cut bipartitioning of the network
  - If any of the parts is dense enough, record it as a dense cluster of proteins
  - Else, further partition them recursively

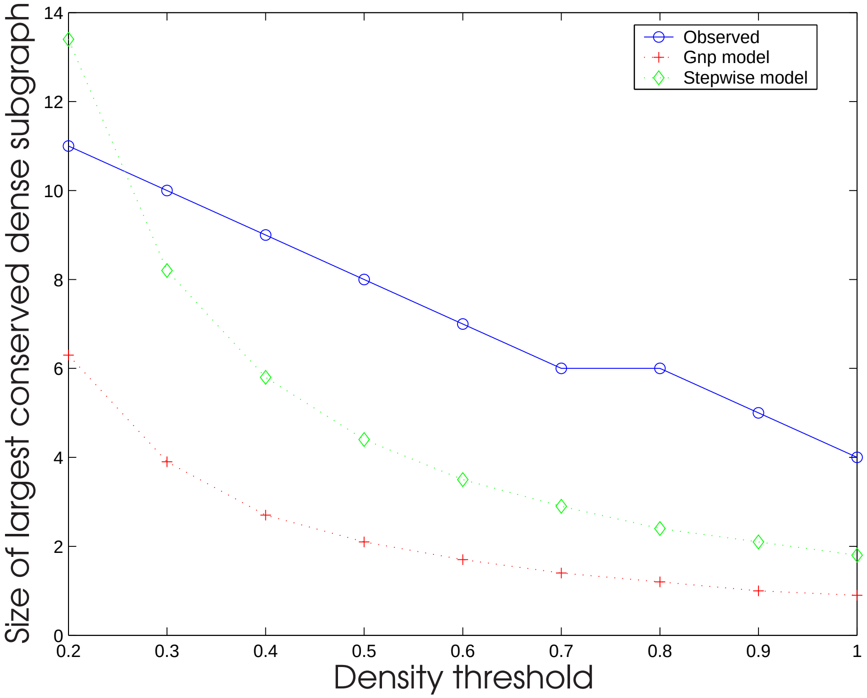
- Use statistical significance to determine whether a subgraph is sufficiently dense

  - For given number of proteins and interactions between them, we can determine whether those proteins induce a significantly dense subnet

Largest Dense Subgraph for Varying Density



Yeast PPI network



Yeast & *Fruit Fly* PPI networks

Pathway Organization: Genetic Interactome

Double mutants exhibit unexpected phenotypes, as compared to joint single mutations.

Definition 1.

- ***Negative Interactions***: *more severe phenotype than expected*
  - *Also known as aggravating or synergistic*
- ***Positive Interactions***: *Less severe phenotype than expected*
  - *Also known as alleviating or epistatic*

Most commonly used:

Phenotype : Growth rate

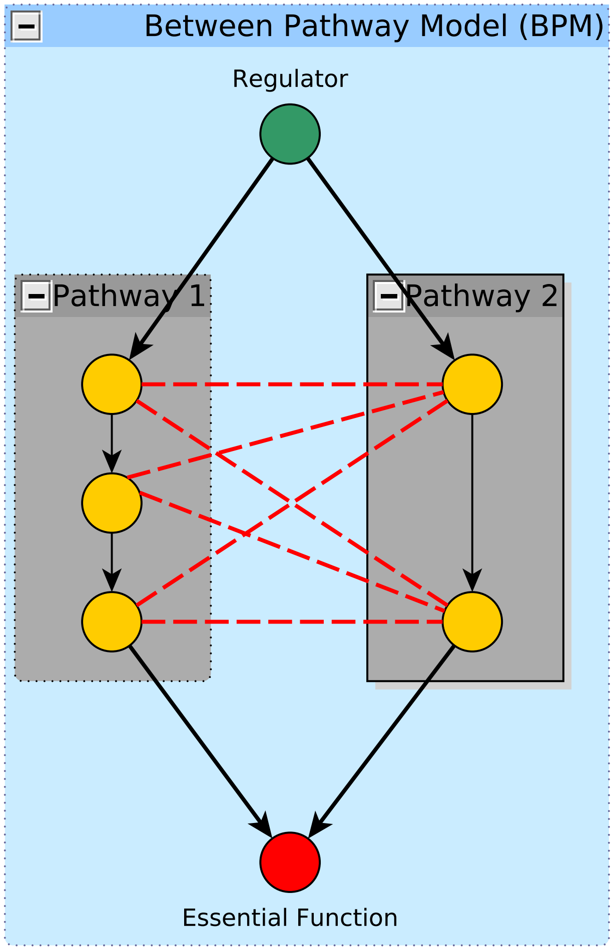
Model : Multiplicative null model

Organization of Genetic Interactions

Definition 2.

- *Between-Pathway Model*
  - *Among genes participating in redundant functions*
- *Within-Pathway Model*
  - *Among genes with additive effect*
- *Indirect Effect*
  - *Among genes with distant functions that are not directly related*

Between-Pathway Model (BPM)

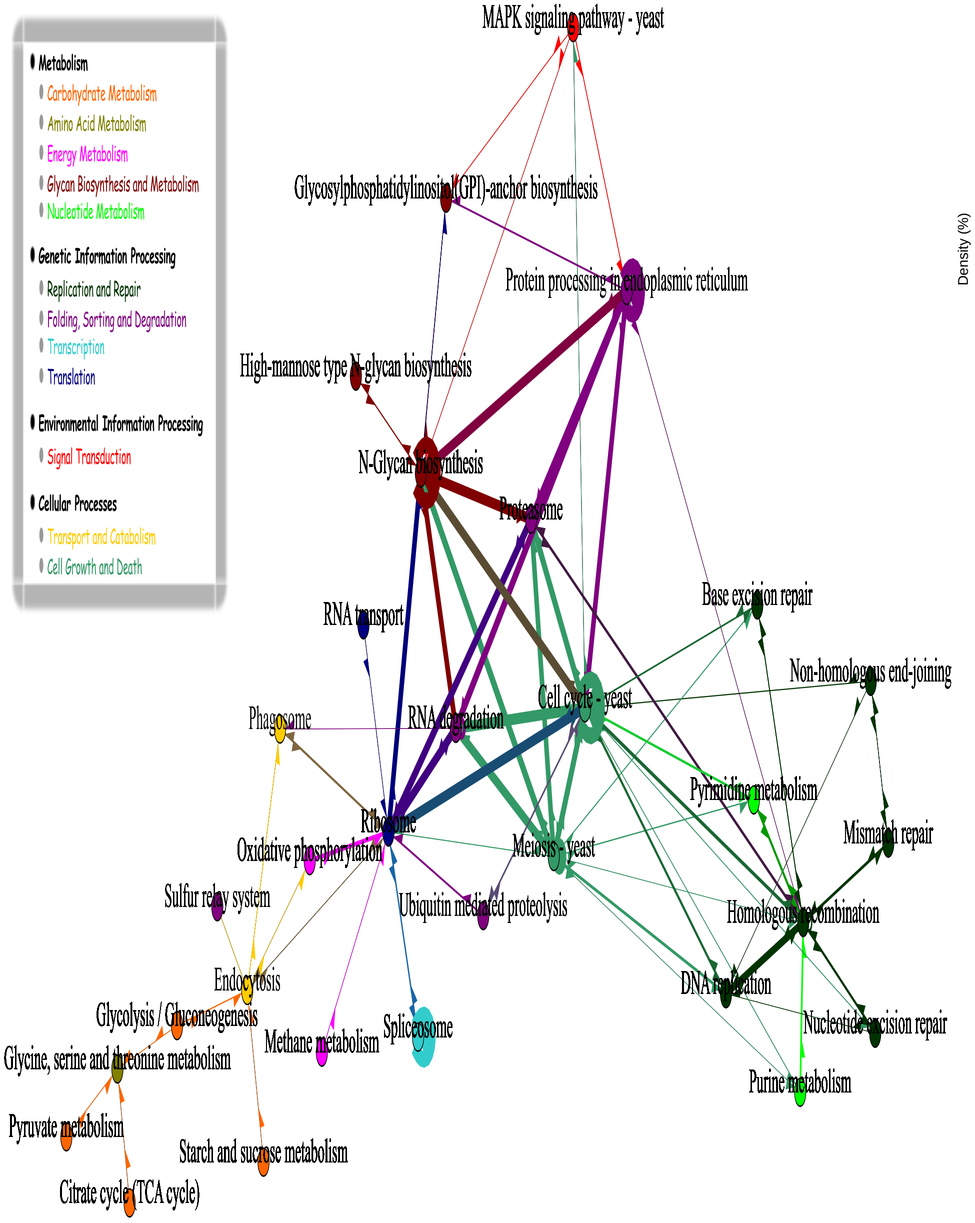


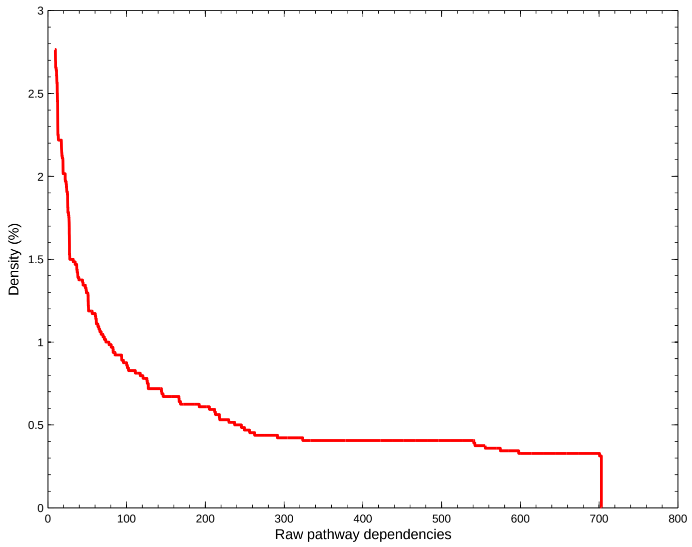
- Bi-cliquish structure

- Have been used to:

  1. Predict co-pathway membership of gene pairs
  2. Extract redundant pathways

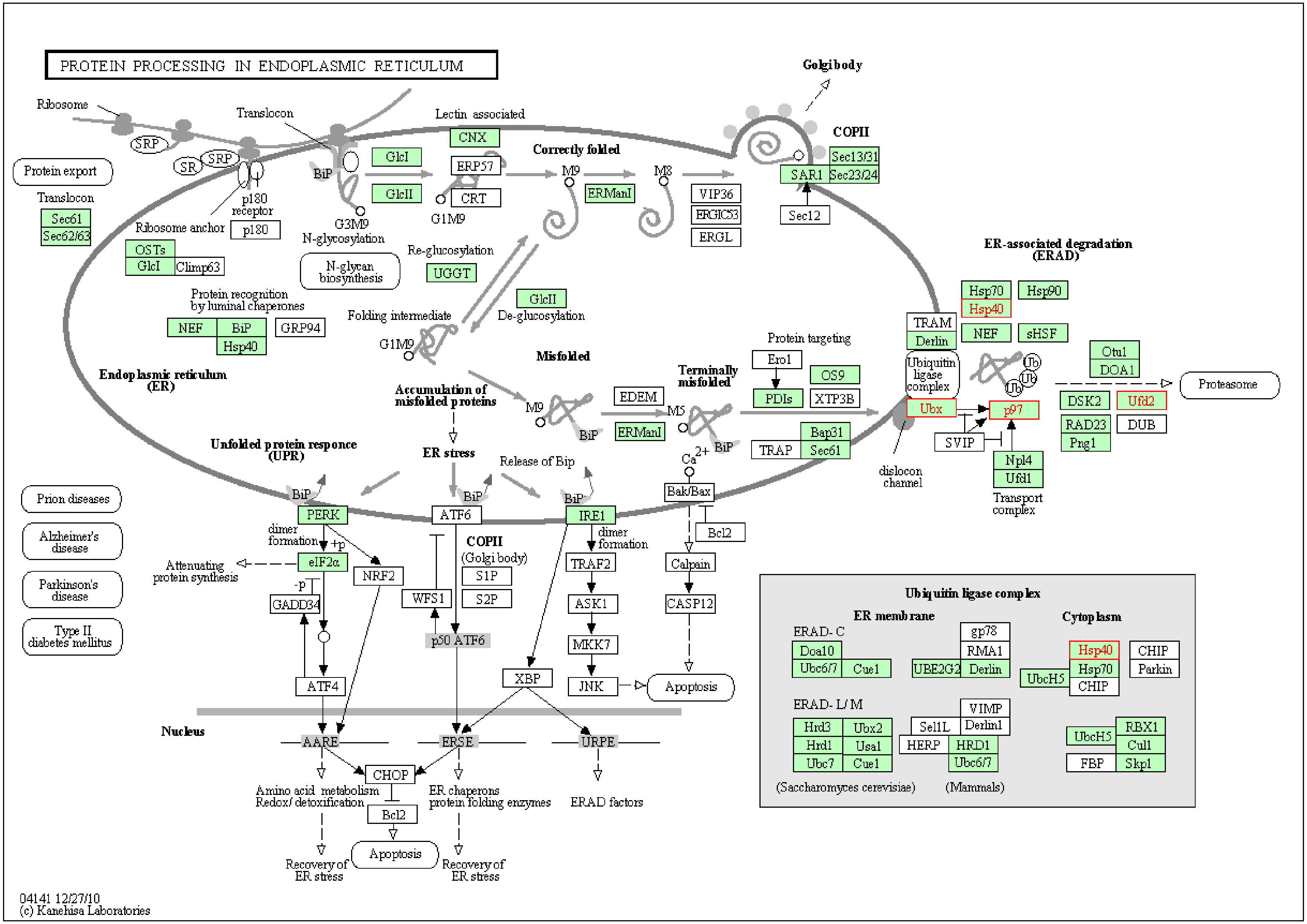
KEGG Crosstalk Map


Metabolism
Carbohydrate Metabolism
Amino Acid Metabolism
Energy Metabolism
Glycan Biosynthesis and Metabolism
Nucleotide Metabolism
Genetic Information Processing
Replication and Repair
Folding, Sorting and Degradation
Transcription
Translation
Environmental Information Processing
Signal Transduction
Cellular Processes
Transport and Catabolism
Cell Growth and Death
MAPK signaling pathway - yeast
Glycosylphosphatidylinositol(GPI)-anchor biosynthesis
Protein processing in endoplasmic reticulum
High-mannose type N-glycan biosynthesis
N-Glycan biosynthesis
Proteasome
Base excision repair
RNA transport
Non-homologous end-joining
Cell cycle - yeast
Phagosome
RNA degradation
Pyrimidine metabolism
Mismatch repair
Ribosome
Oxidative phosphorylation
Meiosis - yeast
Sulfur relay system
Ubiquitin mediated proteolysis
Homologous recombination
Endocytosis
Glycolysis / Gluconeogenesis
DNA replication
Methane metabolism
Spliceosome
Nucleotide excision repair
Glycine, serine and threonine metabolism
Purine metabolism
Pyruvate metabolism
Starch and sucrose metabolism
Citrate cycle (TCA cycle)

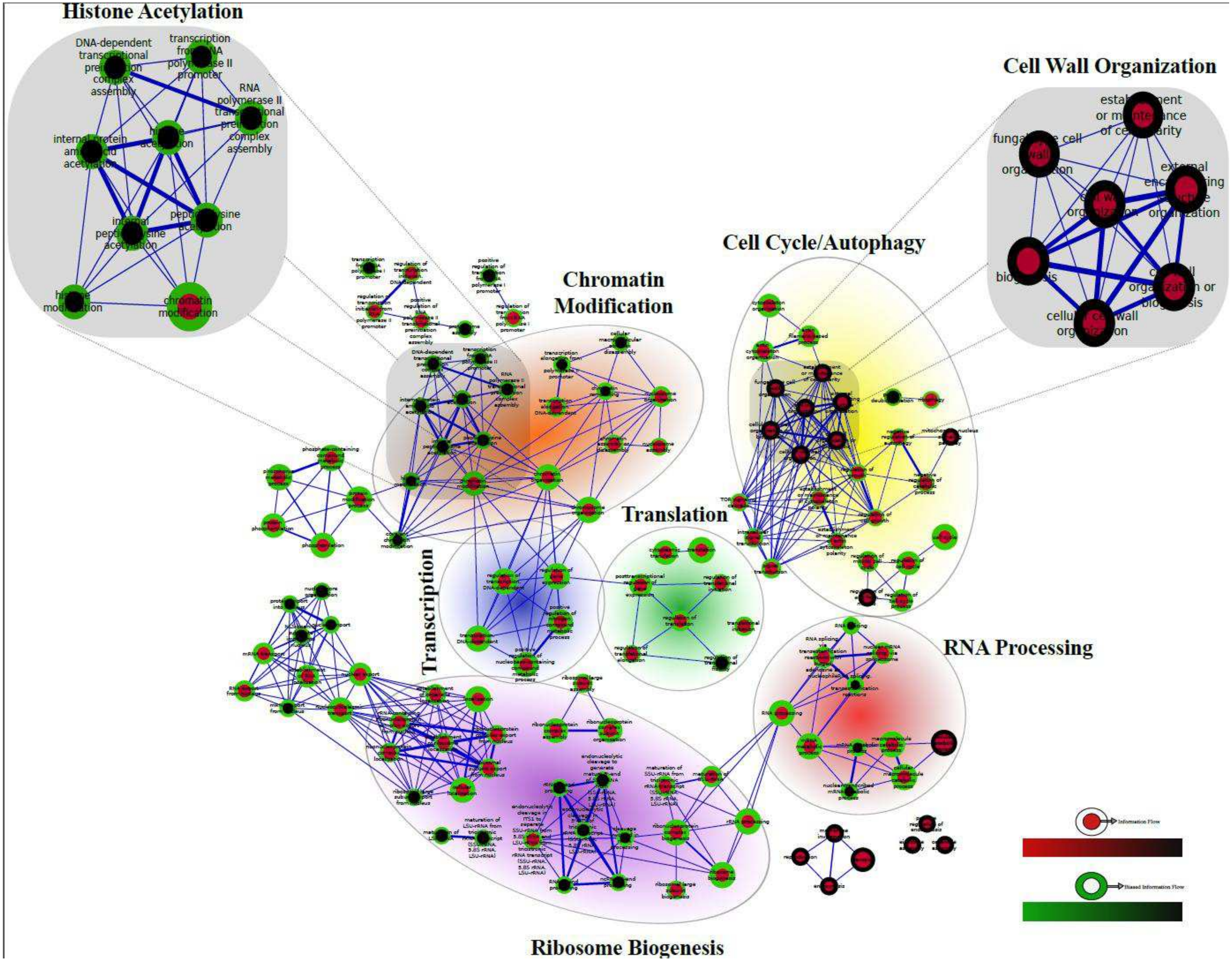


Interaction Port Case Study

Crosstalk Between Protein Processing in ER and Proteasome



The Interaction Map of Aging


Histone Acetylation
DNA-dependent transcriptional preinitiation complex assembly
transcription from RNA polymerase II promoter
RNA polymerase II transcriptional preinitiation complex assembly
internal protein amino acid acetylation
histone acetylation
peptidyl-lysine acetylation
internal peptidyl-lysine acetylation
histone modification
chromatin modification
Cell Wall Organization
establishment or maintenance of cell polarity
fungal-type cell wall organization
external encapsulating structure organization
cell wall organization
biogenesis
cell wall organization or biogenesis
cellular cell wall organization
Chromatin Modification
Cell Cycle/Autophagy
Translation
Transcription
RNA Processing
Ribosome Biogenesis
Information Flow
Biased Information Flow

Functional PageRank (PR)

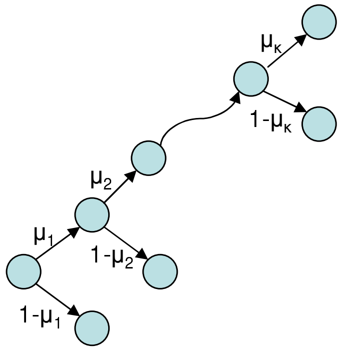
Computing PageRank (PR)

- PageRank as a *random surfer process*: Start surfing from a random node and keep following links with probability μ restarting with probability $1 - \mu$; the node for restarting will be selected based on a personalization vector v. The ranking value x_i of a node i is the probability of visiting this node during surfing.
- PR can also be cast in power series representation as $x = (1 - \mu) \sum_{j=0}^{k} \mu^j S^j v$; S encodes column-stochastic adjacencies.

Functional rankings

- A general method to assign ranking values to graph nodes as $x = \sum_{j=0}^{k} \zeta_j S^j v$. PR is a functional ranking, $\zeta_j = (1 - \mu)\mu^j$.
- Terms attenuated by outdegrees in S *and* damping coefficients ζ_j.

Functional Rankings Through Multidamping (Kollias, Gallopoulos, AG, TKDE'13)



Computing μ_j in multidamping
Simulate a functional ranking by random surfers following emanating links with probability μ_j at step j given by :

$$\mu_j = 1 - \frac{1}{1+\frac{\rho_{k-j+1}}{1-\mu_{j-1}}}, j = 1, \ldots, k,$$

where $\mu_0 = 0$ and $\rho_{k-j+1} = \frac{\zeta_{k-j+1}}{\zeta_{k-j}}$

Examples

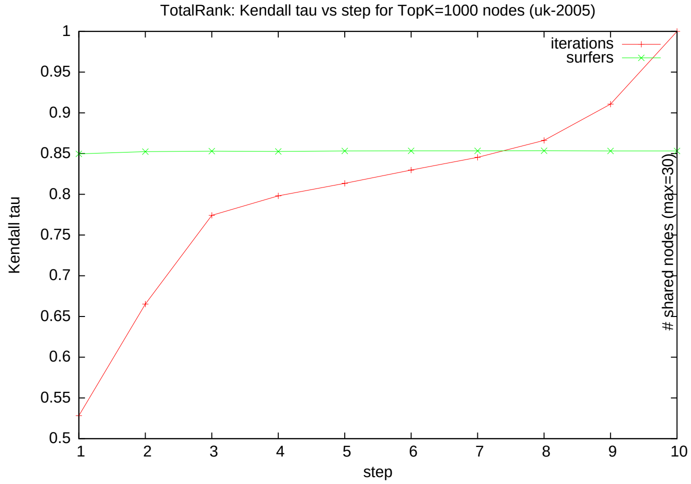
LinearRank (LR) $x^{\mathrm{LR}} = \sum_{j=0}^{k} \frac{2(k+1-j)}{(k+1)(k+2)} S^j v : \mu_j = \frac{j}{j+2}, j = 1, \ldots, k.$

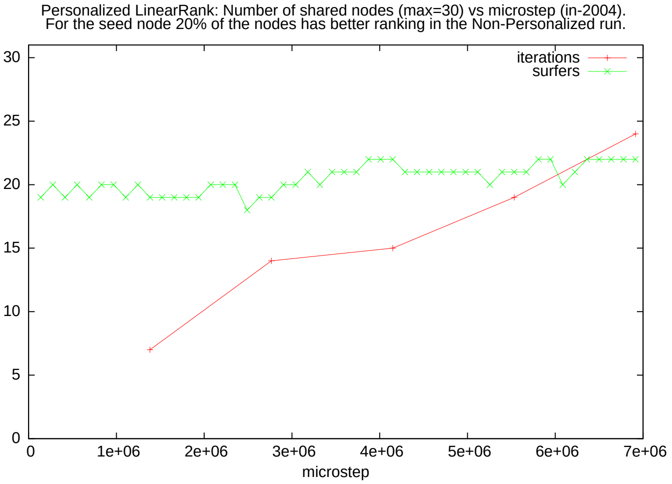
TotalRank (TR) $x^{\mathrm{TR}} = \sum_{j=0}^{\infty} \frac{1}{(j+1)(j+2)} S^j v : \mu_j = \frac{k-j+1}{k-j+2}, j = 1, \ldots, k.$

Advantages of multidamping

- Interpretability and Design!
- Reduced computational cost in *approximating* functional rankings using the Monte Carlo approach. A random surfer terminates with probability $1 - \mu_j$ at step j.
- Inherently parallel and synchronization free computation.

Multidamping Performance

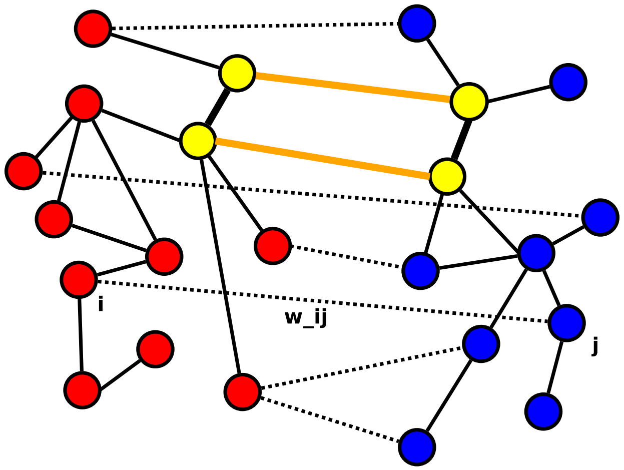




Approximate ranking: Run n surfers to completion for graph size n. How well does the computed ranking capture the “reference” ordering for `top-`k nodes, compared to standard iterations of equivalent computational cost/number of operations? *(Left)*

Approximate personalized ranking: Run less than n surfers to completion (each called a microstep, x-axis), from a selected node (personalized). How well can we capture the “reference” `top-`k nodes, i.e., how many of them are shared (y-axis), compared to the simple approach? *(Right)*

Network Alignment



- **Node similarity:** Two nodes are similar if they are linked by other similar node pairs. By pairing similar nodes, the two graphs become *aligned*.
- Let $\tilde{A}$ and $\tilde{B}$ be the normalized adjacency matrices of the graphs (normalized by columns), H_{ij} be the independently known similarity scores (preferences matrix) of nodes $i \in V_B$ and $j \in V_A$, and μ be the fractional contribution of topological similarity.
- To compute X, IsoRank iterates:

$$X \leftarrow \mu \tilde{B} X \tilde{A}^T + (1-\mu) H$$

Network Similarity Decomposition (NSD) (Kollias, Mohammadi, AG, TKDE'12)

Network Similarity Decomposition (NSD)

- In n steps of we reach $X^{(n)} = (1-\mu)\sum_{k=0}^{n-1} \mu^k \tilde{B}^k H (\tilde{A}^T)^k + \mu^n \tilde{B}^n H (\tilde{A}^T)^n$
- Assume that $H = uv^T$ $(1$ component$)$. Two phases for X:
  1. $u^{(k)} = \tilde{B}^k u$ and $v^{(k)} = \tilde{A}^k v$ *(preprocess/compute iterates)*
  2. $X^{(n)} = (1-\mu)\sum_{k=0}^{n-1} \mu^k u^{(k)} {v^{(k)}}^T + \mu^n u^{(n)} {v^{(n)}}^T$ *(construct X)*

  This idea extends to s components, $H \sim \sum_{i=1}^{s} w_i z_i^T$.
- NSD computes matrix-vector iterates and builds X as a sum of outer products; these are much cheaper than triple matrix products.

We can then apply Primal-Dual or Greedy Matching (1/2 approximation) to extract the actual node pairs.

NSD: Performance (Kollias, Madan, Mohammadi, AG, BMC RN'12)

Species	Nodes	Edges
celeg (worm)	2805	4572
dmela (fly)	7518	25830
ecoli (bacterium)	1821	6849
hpylo (bacterium)	706	1414
hsapi (human)	9633	36386
mmusc (mouse)	290	254
scere (yeast)	5499	31898

Species pair	NSD (secs)	PDM (secs)	GM (secs)	IsoRank (secs)
celeg-dmela	**3.15**	152.12	7.29	783.48
celeg-hsapi	**3.28**	163.05	9.54	1209.28
celeg-scere	**1.97**	127.70	4.16	949.58
dmela-ecoli	**1.86**	86.80	4.78	807.93
dmela-hsapi	**8.61**	590.16	28.10	7840.00
dmela-scere	**4.79**	182.91	12.97	4905.00
ecoli-hsapi	**2.41**	79.23	4.76	2029.56
ecoli-scere	**1.49**	69.88	2.60	1264.24
hsapi-scere	**6.09**	181.17	15.56	6714.00

- We compute similarity matrices X for various pairs of species using Protein-Protein Interaction (PPI) networks. $\mu = 0.80$, uniform initial conditions (outer product of suitably normalized 1's for each pair), 20 iterations, one component.
- We then extract node matches using PDM and GM.
- *Three orders of magnitude speedup* from NSD-based approaches compared to IsoRank.

NSD: Parallelization (KKG JPDC'13, Submitted, KMSAG ParCo'13 Submitted)

Parallelization: NSD has been ported to parallel and distributed platforms.

- We have aligned up to million-node graph instances using over 3K cores.
- We process graph pairs of over a billion nodes and twenty billion edges each (!), on MapReduce-based distributed platforms.

Part 3: Systems Development

In support of graph analytics, we have build extensive systems infrastructure for programming at scale.

- TransMR – Transactional MapReduce that enables maps to operate on persistent key-value stores while supporting well-defined semantics.
- TransDF – a Transactional Dynamic Dataflow environment that enables distributed computations without heavy (three-copy) overheads, while supporting fault tolerance and speculation.
- Concurrency management schemes for TransMR and TransDF.
- Distributed graph kernel library on TransMR.

Acknowledgements

- To the US National Science Foundation for their funding.