CS 3824: Gene Function Prediction

T. M. Murali

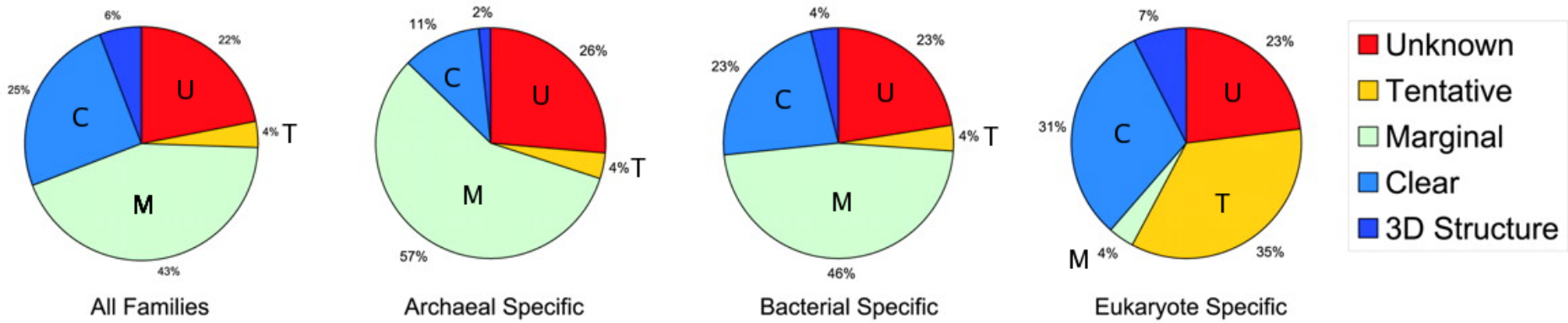
October 13, 18, 2022

Data, Data, Data

- $\geq$ 100,000+ microbial and $>$ 3,000 animal genomes sequenced.
- Computational identification of genes in sequenced genomes.
- Massive datasets measuring levels and activities of molecules.
- Molecular interaction networks, metabolic pathways.

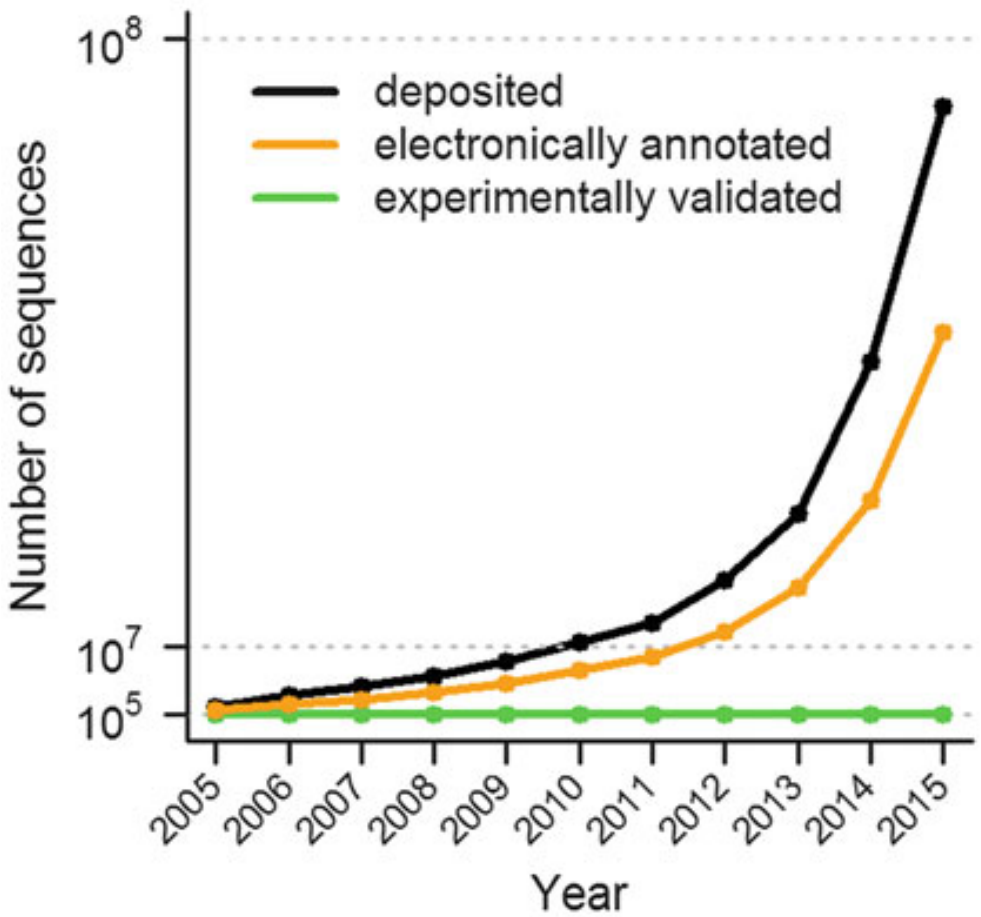
Roadblock: What Functions Do Genes Perform?

"During the last few years, we have seen enormous strides in our abilities to sequence genomes, . . . With more than 150 complete genome sequences now available and many laboratories rushing into microarray analysis, proteomic initiatives, and even systems biology, it seems an appropriate time to consider not just the opportunities those sequences present, but also their shortcomings. By far the most serious problem is the quality and degree of completeness of the annotation of those genomes." (*Identifying Protein Function—A Call for Community Action*. Roberts RJ (2004), PLoS Biol 2(3): e42.)



UniProt Annotation Coverage

Tiny fraction of genes have an experimentally validated function



Cozetti and Jones, Chapter 5, The Gene Ontology Handbook, 2017

Solution: Automated Gene Function Prediction

- Develop computational techniques that automatically integrate diverse source of data to predict function.
- Provide measures of confidence and statistical significance for each prediction.
- Present the predictions in a user-friendly manner to a biologist for designing experiments to validate prediction.

How do you Predict Function?

How do you Predict Function?

- Genes with similar sequences in different organisms are likely to have the same function.
- Use algorithms for computing sequence and structural similarity.
- Transfer the known function of a well-studied gene to a gene with a similar sequence that has no known functions.

How do you Predict Function?

- Genes with similar sequences in different organisms are likely to have the same function.
- Use algorithms for computing sequence and structural similarity.
- Transfer the known function of a well-studied gene to a gene with a similar sequence that has no known functions.

BUT

- 25% of the genes have no known sequence or structural similarity to any gene in any other organism.
- An additional 50% have poor annotations.

How do you Predict Function?

- Genes with similar sequences in different organisms are likely to have the same function.
- Use algorithms for computing sequence and structural similarity.
- Transfer the known function of a well-studied gene to a gene with a similar sequence that has no known functions.

BUT

- 25% of the genes have no known sequence or structural similarity to any gene in any other organism.
- An additional 50% have poor annotations.

We need techniques for gene function prediction that go beyond sequence similarity.

What is Gene Function?

- Not an easy question to answer!
- A gene's function has many aspects.
- Different aspects are interesting to different biologists.
- There are many ways to describe a gene's function.
- Different groups of biologists have derived different vocabularies.

The Gene Ontology (GO)

- Collaborative effort to define a controlled vocabulary to describe gene and gene product attributes in any organism. Started in 1999.

The Gene Ontology (GO)

- Collaborative effort to define a controlled vocabulary to describe gene and gene product attributes in any organism. Started in 1999.
- Visit http://www.geneontology.org
- Three GO aspects (namespaces):

The Gene Ontology (GO)

- Collaborative effort to define a controlled vocabulary to describe gene and gene product attributes in any organism. Started in 1999.
- Visit http://www.geneontology.org
- Three GO aspects (namespaces): A gene product has
  - a *molecular function*: an activity, such as catalyzing or binding, carried out by the gene product at the molecular level;

The Gene Ontology (GO)

- Collaborative effort to define a controlled vocabulary to describe gene and gene product attributes in any organism. Started in 1999.
- Visit http://www.geneontology.org
- Three GO aspects (namespaces): A gene product has
  - a *molecular function*: an activity, such as catalyzing or binding, carried out by the gene product at the molecular level;
  - is used in a *biological process*: a series of events accomplished by one or more ordered assemblies of molecular functions; and

The Gene Ontology (GO)

- Collaborative effort to define a controlled vocabulary to describe gene and gene product attributes in any organism. Started in 1999.
- Visit http://www.geneontology.org
- Three GO aspects (namespaces): A gene product has
  - a *molecular function*: an activity, such as catalyzing or binding, carried out by the gene product at the molecular level;
  - is used in a *biological process*: a series of events accomplished by one or more ordered assemblies of molecular functions; and
  - might be associated with a *cellular component*: a component of a cell that is part of some larger object, which may be an anatomical structure or a gene product group.

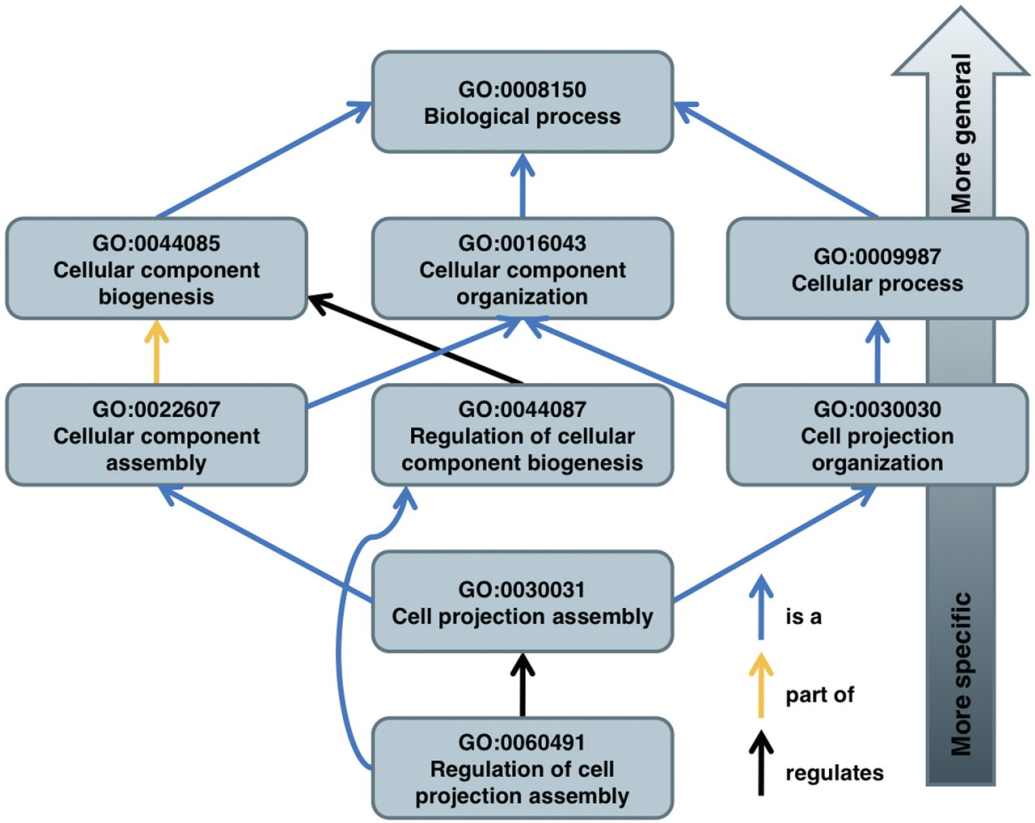
The Gene Ontology (GO)

- Collaborative effort to define a controlled vocabulary to describe gene and gene product attributes in any organism. Started in 1999.
- Visit http://www.geneontology.org
- Three GO aspects (namespaces): A gene product has
  - a *molecular function*: an activity, such as catalyzing or binding, carried out by the gene product at the molecular level;
  - is used in a *biological process*: a series of events accomplished by one or more ordered assemblies of molecular functions; and
  - might be associated with a *cellular component*: a component of a cell that is part of some larger object, which may be an anatomical structure or a gene product group.
- For example, the gene product *Angiotensin-converting enzyme 2* (ACE2) has
  - the molecular function term *virus receptor activity*,
  - the biological process terms *regulation of cytokine production* and *viral life cycle*, and
  - the cellular component term *extracellular region* and *plasma membrane*.

▸ Jump to FLNs

Features of GO: Hierarchy

- A team of experts defines GO terms.
- GO terms are described at multiple levels of detail.
- Explicit parent-child relationships between terms, forming a directed acyclic graph (DAG).



Features of GO: Evidence Codes

- Annotations typically done by individual genome databases.
- Evidence code attached to annotation in six categories: http://geneontology.org/docs/guide-go-evidence-codes/

Features of GO: Evidence Codes

- Annotations typically done by individual genome databases.
- Evidence code attached to annotation in six categories: http://geneontology.org/docs/guide-go-evidence-codes/
  - experimental evidence
  - phylogenetic evidence
  - computational evidence
  - author and curatorial statements
  - automatically generated annotations
  - not determined

Advantages of GO

- The vocabulary is controlled $\Rightarrow$ common vocabulary for all biologists.
- Designed to apply across species.
- Computed mappings from other functional catalgues to GO.
- The GO terms are constantly updated (actually a headache for gene function prediction algorithms).
- Freely available to the community.

Moving Beyond GO

- GO does not describe many aspects of a gene's function:

Moving Beyond GO

- GO does not describe many aspects of a gene's function: which cells or tissues it is expressed in, which developmental stages it is expressed in, or its involvement in disease.

Moving Beyond GO

- GO does not describe many aspects of a gene's function: which cells or tissues it is expressed in, which developmental stages it is expressed in, or its involvement in disease.
- Other ontologies are being developed to meet these needs.
  - Open Biomedical Ontologies: http://obo.sourceforge.net/
  - Ontology Working Group of the Microarray Gene Expression Data Society (MGED): http://mged.sourceforge.net/ontologies/OntologyResources.php

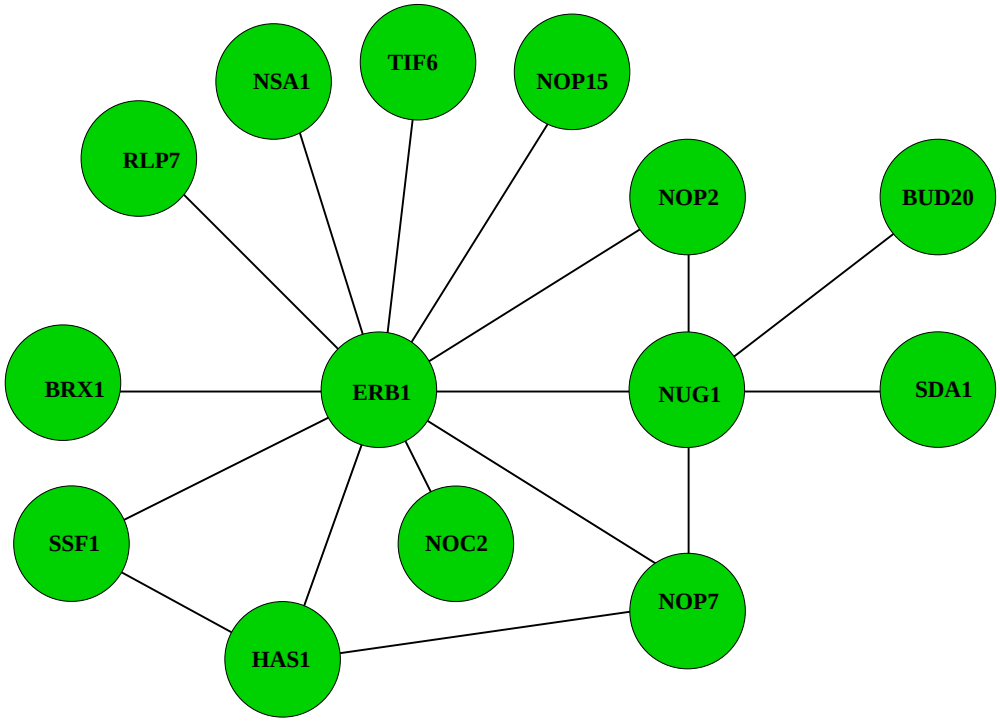
Moving Beyond GO

- GO does not describe many aspects of a gene's function: which cells or tissues it is expressed in, which developmental stages it is expressed in, or its involvement in disease.
- Other ontologies are being developed to meet these needs.
  - Open Biomedical Ontologies: http://obo.sourceforge.net/
  - Ontology Working Group of the Microarray Gene Expression Data Society (MGED): http://mged.sourceforge.net/ontologies/OntologyResources.php
- "Cross-products" of different ontologies: combine different (independent) ontologies to derive richer vocabularies.

Moving Beyond GO

- GO does not describe many aspects of a gene's function: which cells or tissues it is expressed in, which developmental stages it is expressed in, or its involvement in disease.
- Other ontologies are being developed to meet these needs.
  - Open Biomedical Ontologies: http://obo.sourceforge.net/
  - Ontology Working Group of the Microarray Gene Expression Data Society (MGED): http://mged.sourceforge.net/ontologies/OntologyResources.php
- "Cross-products" of different ontologies: combine different (independent) ontologies to derive richer vocabularies.
- "For example, by combining the developmental terms in the GO process ontology with a second ontology that describes Drosophila anatomical structures, we could create an ontology of fly development."
- "We could create an ontology of biosynthetic pathways by combining the biosynthesis terms in the GO process ontology with a chemical ontology."

Functional Linkage Networks



- A *functional linkage network* (FLN) is a graph where each node corresponds to a gene and each edge connects two genes that may share a similar function.
- An edge may not indicate which function the connected genes share.

Constructing FLNs

- Organism specific: Example from STRING

Constructing FLNs

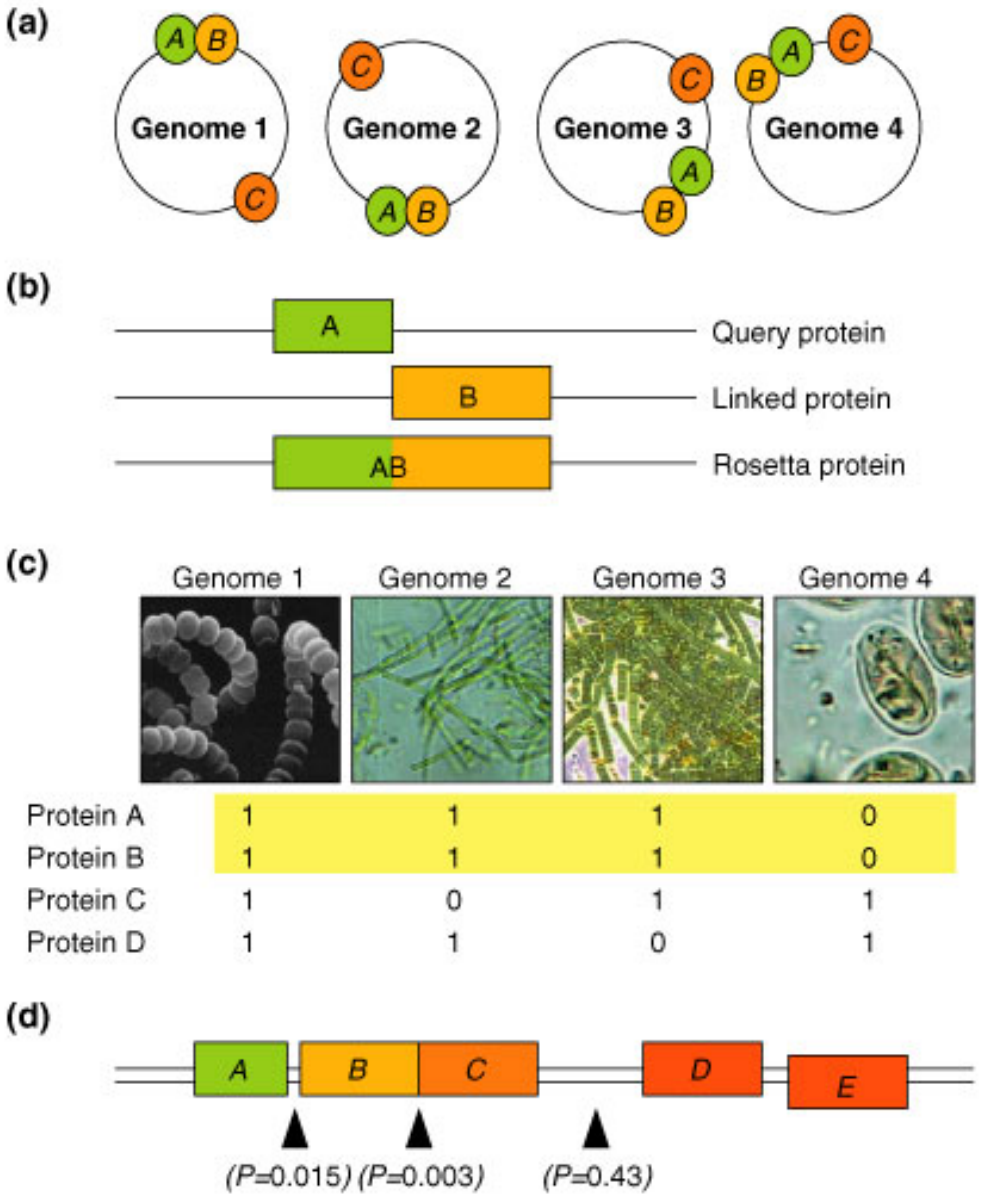
- Organism specific: Example from STRING
  - Co-expression from DNA microarray data.
  - Protein products interact.
  - Enzymes that catalyse different reactions in the same metabolic pathway.
  - Genes co-regulated by the same transcription factor.
  - Double mutants are lethal (synthetic lethality).
- Cross-organism: Information on co-evolution encoded in genomic context.

Constructing FLNs

- Organism specific: Example from STRING
  - Co-expression from DNA microarray data.
  - Protein products interact.
  - Enzymes that catalyse different reactions in the same metabolic pathway.
  - Genes co-regulated by the same transcription factor.
  - Double mutants are lethal (synthetic lethality).
- Cross-organism: Information on co-evolution encoded in genomic context.

▸ Onward to Challenges

Cross-Organism Functional Associations



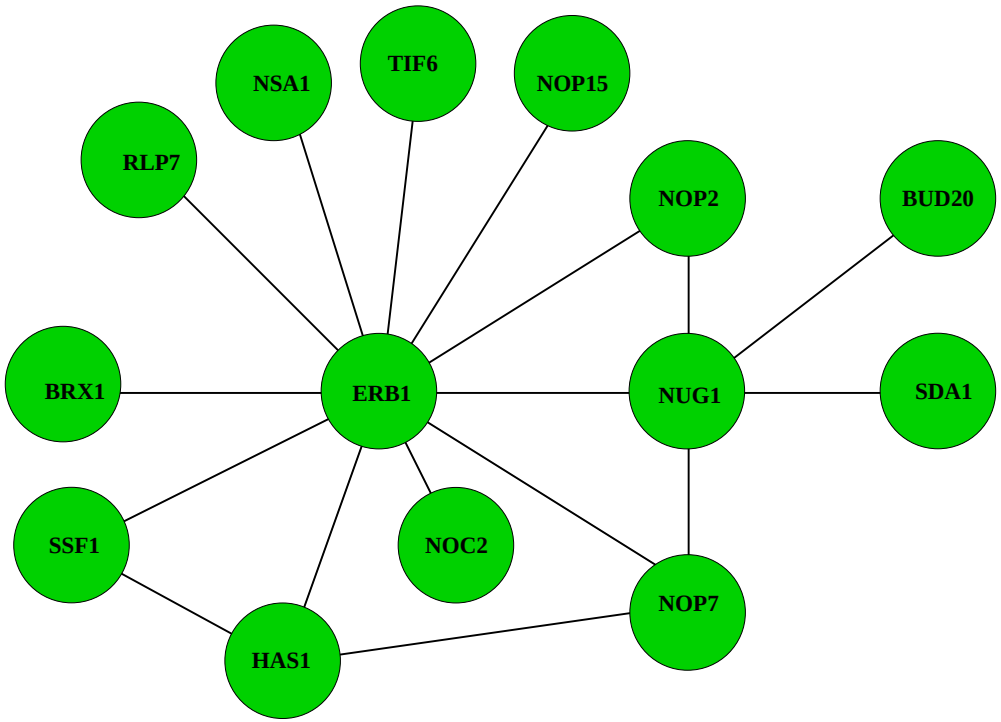
Research on Functional Links

- Databases: BIND, DIP, GRID, IDSERVE, PROLINKS, PREDICTOME, REACTOME, STRING, ...
- Techniques for predicting functional associations, e.g., protein-protein interactions (Jansen et al., *Science*, 302, 2003; Zhang et al., *BMC Bioinformatics*, 5, 2005; Park et al., *PLoS Comp. Bio.*, Nov 2010), Kovács et al., *Nature Comm.*, 2019.
- Techniques for integrating diverse pieces of evidence into a single integrated FLN (Lee et al., *Science*, 306, 2005; papers by Troyanskaya's group; Mostafavi et al., *Genome Biology*, 2008).

Research on Functional Links

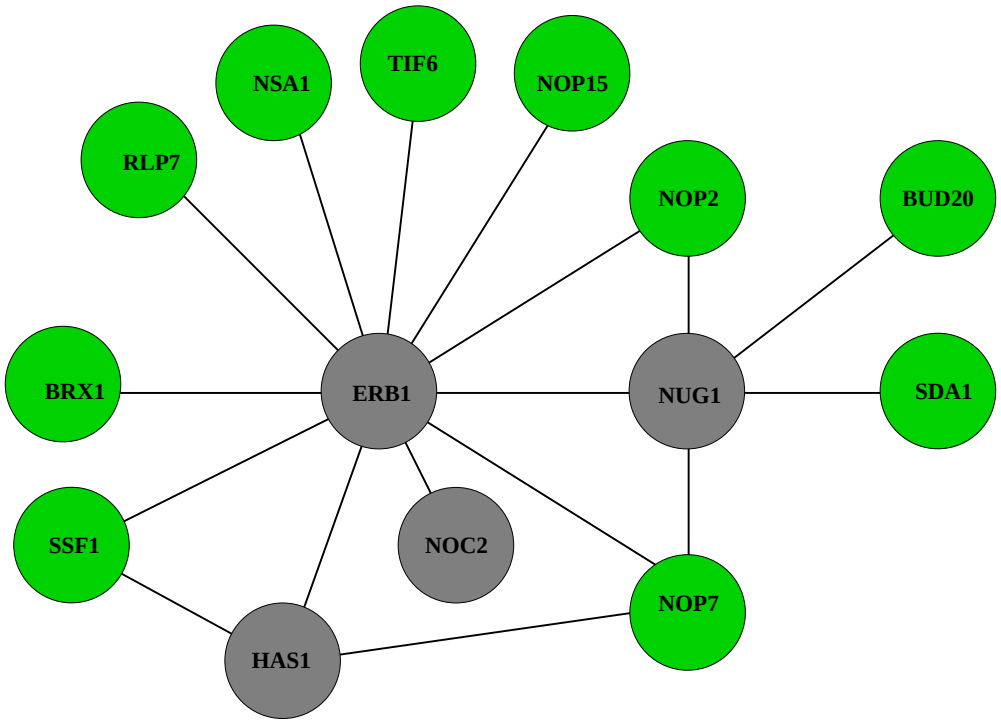
- Databases: BIND, DIP, GRID, IDSERVE, PROLINKS, PREDICTOME, REACTOME, STRING, ...
- Techniques for predicting functional associations, e.g., protein-protein interactions (Jansen et al., *Science*, 302, 2003; Zhang et al., *BMC Bioinformatics*, 5, 2005; Park et al., *PLoS Comp. Bio.*, Nov 2010), Kovács et al., *Nature Comm.*, 2019.
- Techniques for integrating diverse pieces of evidence into a single integrated FLN (Lee et al., *Science*, 306, 2005; papers by Troyanskaya's group; Mostafavi et al., *Genome Biology*, 2008).
- How do we systematically use FLNs to make robust and quantified predictions of function?

Why is Gene Function Prediction Difficult?



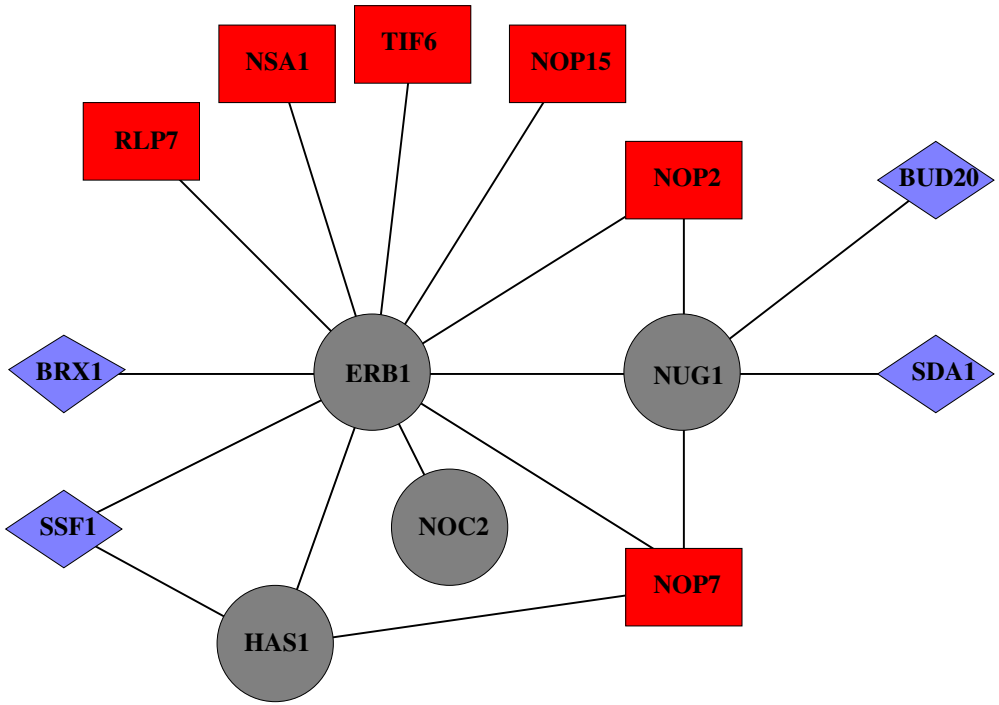
- Functional associations are not perfect indicators of shared function.

Why is Gene Function Prediction Difficult?



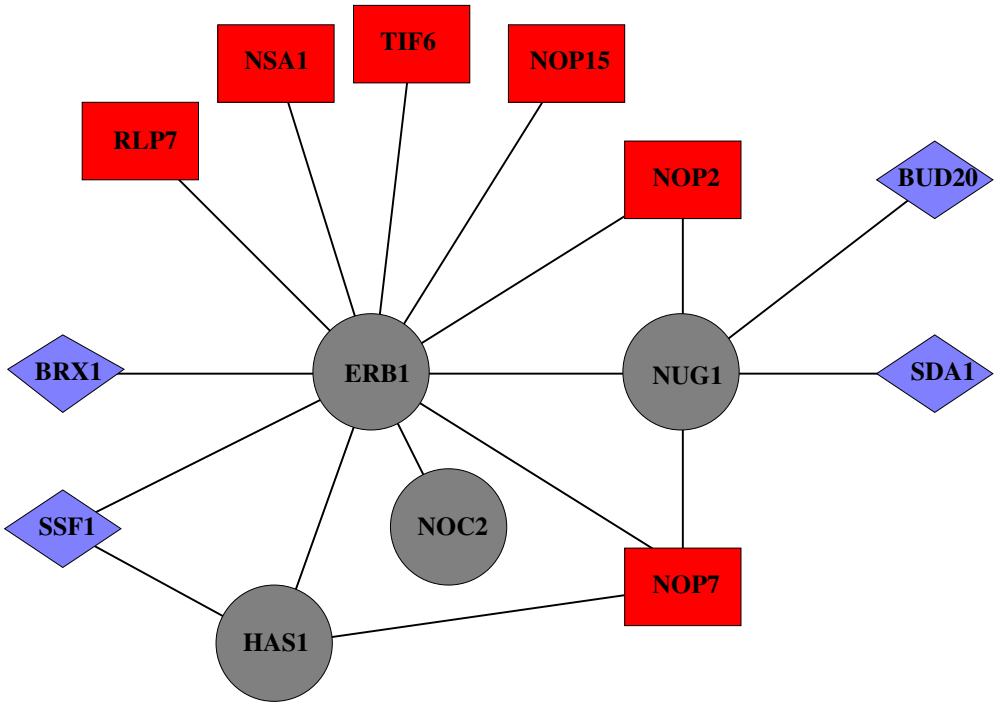
- Functional associations are not perfect indicators of shared function.
- 20–30% of genes of unknown function have only such genes as neighbours.

Why is Gene Function Prediction Difficult?



- Functional associations are not perfect indicators of shared function.
- 20–30% of genes of unknown function have only such genes as neighbours.
- Neighbourhood structure is ambiguous.

The GAIN System

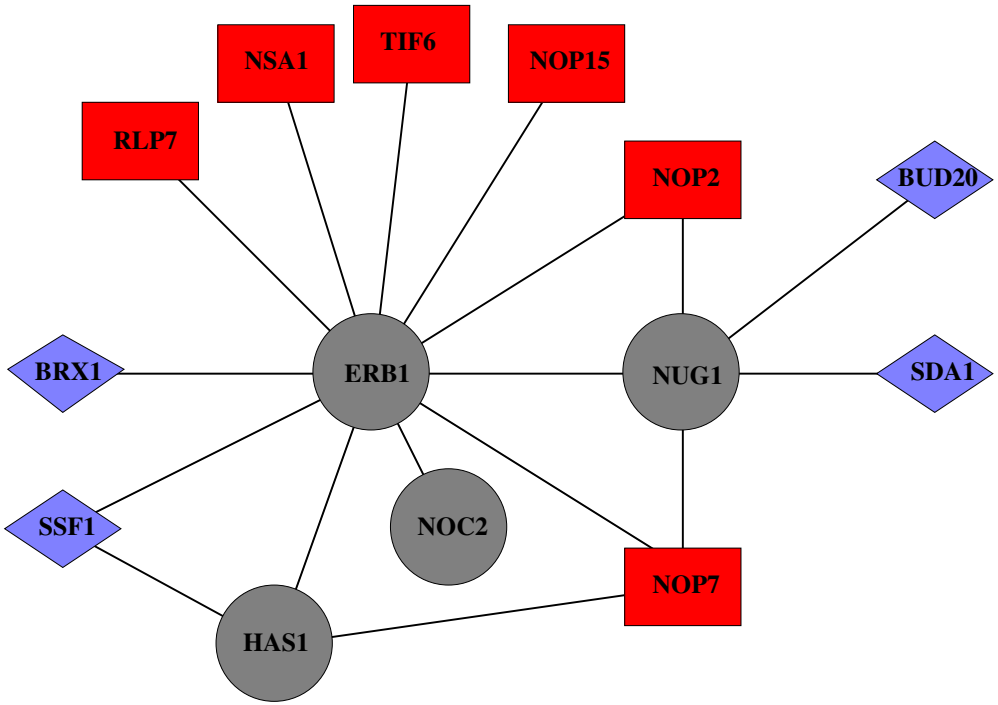


Gene Annotation Using Integrated Networks (GAIN):

- Propagate evidence systematically across the entire FLN.
- Integrate information from different sources to improve robustness.

(Karaoz, Murali, Letovsky, Zheng, Ding, Cantor and Kasif, *PNAS*, 2004, 101, 2888–2893.)

The GAIN System



Gene Annotation Using Integrated Networks (GAIN):

- Propagate evidence systematically across the entire FLN.
- Integrate information from different sources to improve robustness: protein-protein interactions and gene expression data.

(Karaoz, Murali, Letovsky, Zheng, Ding, Cantor and Kasif, *PNAS*, 2004, 101, 2888–2893.)

Overview of the GAIN Pipeline

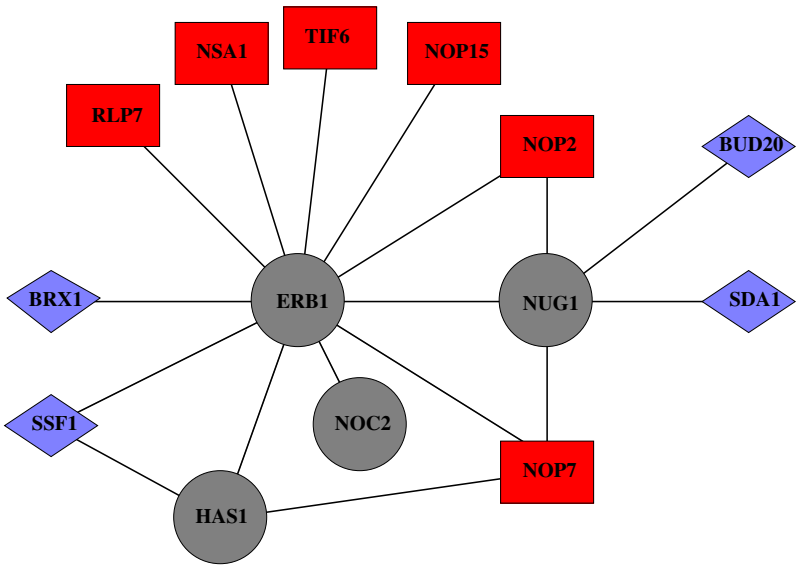
- Inputs: Functional genomic data sets, GO functional annotations.
- Outputs: For each function in GO, a set of genes predicted to have that function.

1. Construct FLN G from functional genomic data sets.
2. For each function f in GO
   1. Construct a labelled FLN G_f for f.
   2. Propagate the label f or $\mathrm{not}f$ across G_f.
   3. Output set of genes that have been assigned the function f.

- Can predict multiple functions for a gene.

Labelled FLNs

- *Labelled FLN* G_f for a function $f \equiv$ the FLN G with states (labels) attached to nodes.
- FLN $\rightarrow$ discrete Hopfield network.
  - Gene $\equiv$ node.
  - Interaction $\equiv$ edge.



- Each node v has an associated state s_v:
  - $s_v = 1$: gene v is annotated with f.
  - $s_v = -1$: gene v is annotated with another function f'.
  - $s_v = 0$: otherwise.
- An edge between nodes u and v has a weight w_{uv}.

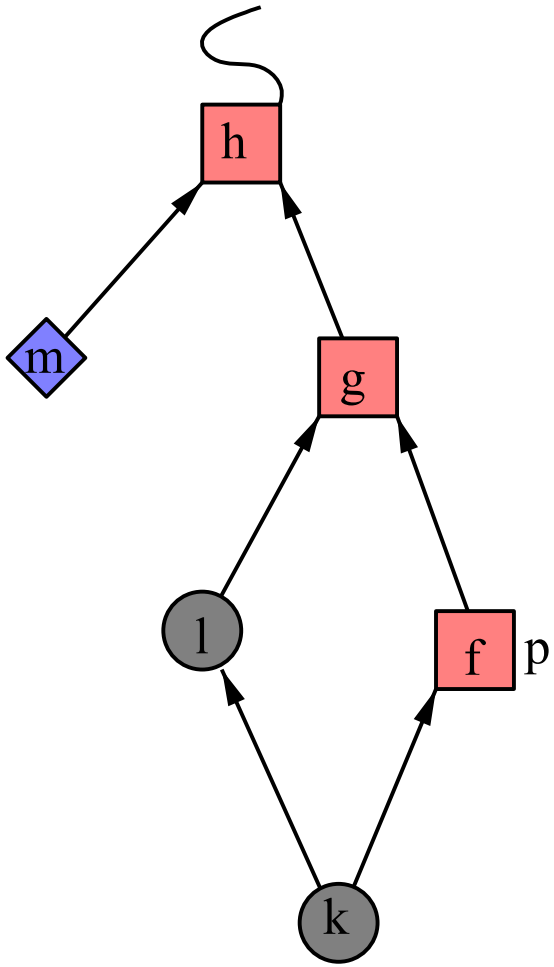
▸ Skip Node States

Assigning Node States

- Assigning node states correctly is not a trivial manner.
- We must respect/exploit GO's hierarchical structure .

Assigning Node States

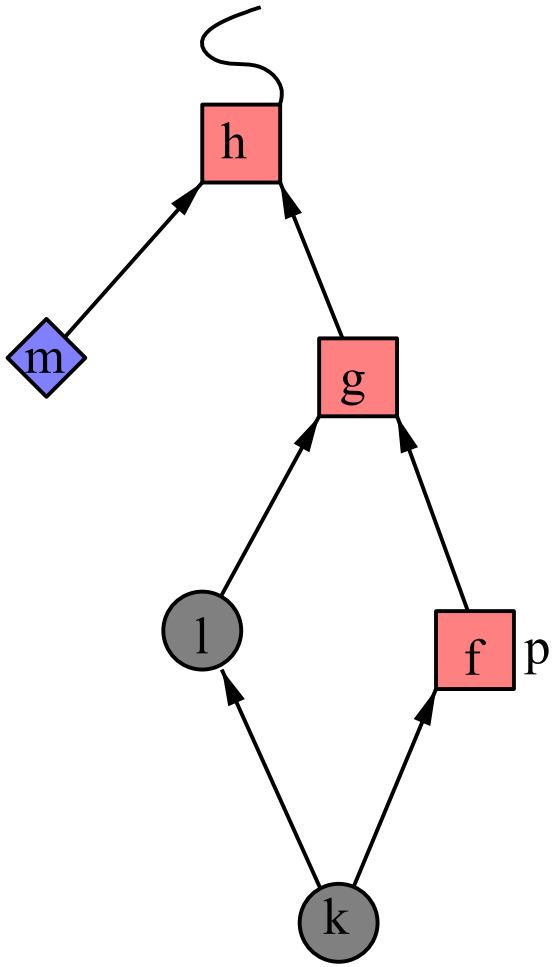
- Assigning node states correctly is not a trivial manner.
- We must respect/exploit GO's hierarchical structure .



- What is state of gene p with respect to function
  - f:
  - g:
  - h:
  - m:
  - k:
  - l:

Assigning Node States

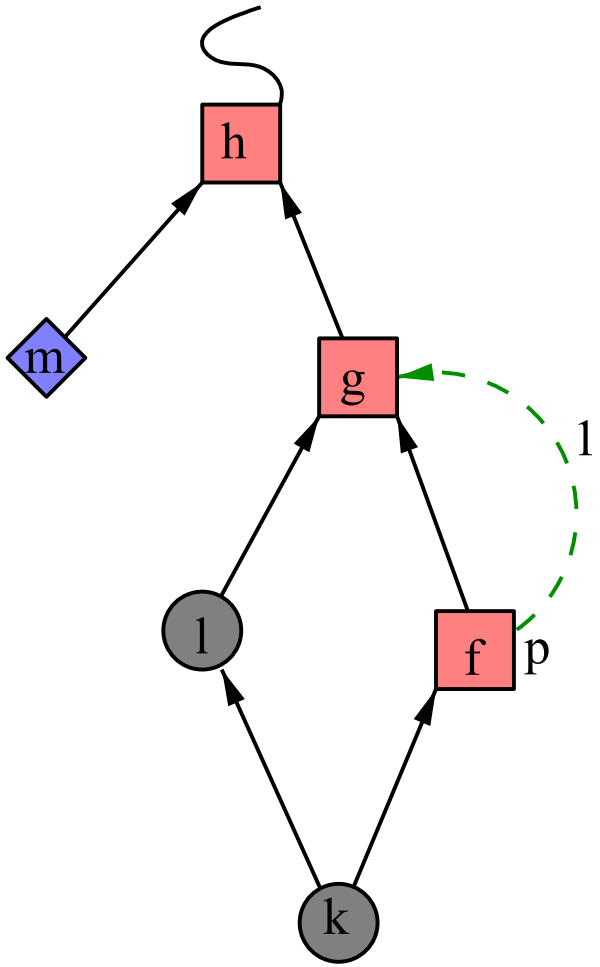
- Assigning node states correctly is not a trivial manner.
- We must respect/exploit GO's hierarchical structure .



- What is state of gene p with respect to function
  - f: 1
  - g:
  - h:
  - m:
  - k:
  - l:

Assigning Node States

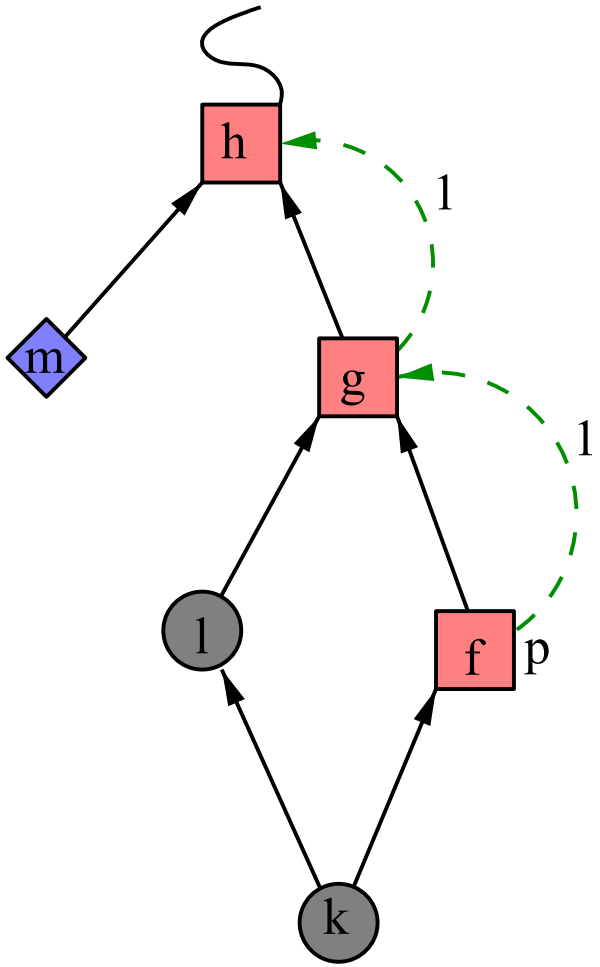
- Assigning node states correctly is not a trivial manner.
- We must respect/exploit GO's hierarchical structure .



- What is state of gene p with respect to function
  - f: 1
  - g: 1
  - h:
  - m:
  - k:
  - l:

Assigning Node States

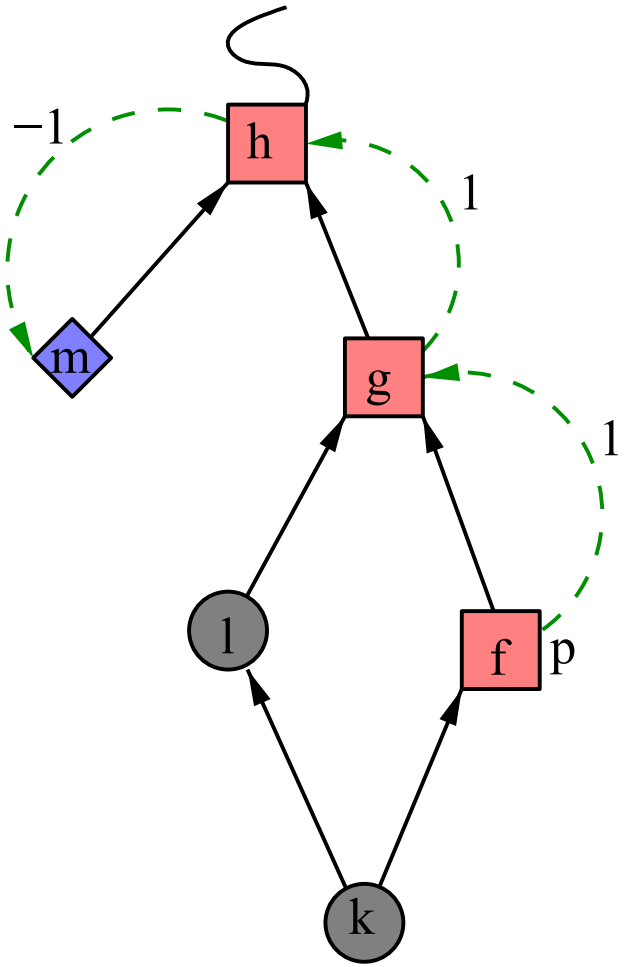
- Assigning node states correctly is not a trivial manner.
- We must respect/exploit GO's hierarchical structure .



- What is state of gene p with respect to function
  - f: 1
  - g: 1
  - h: 1
  - m:
  - k:
  - l:

Assigning Node States

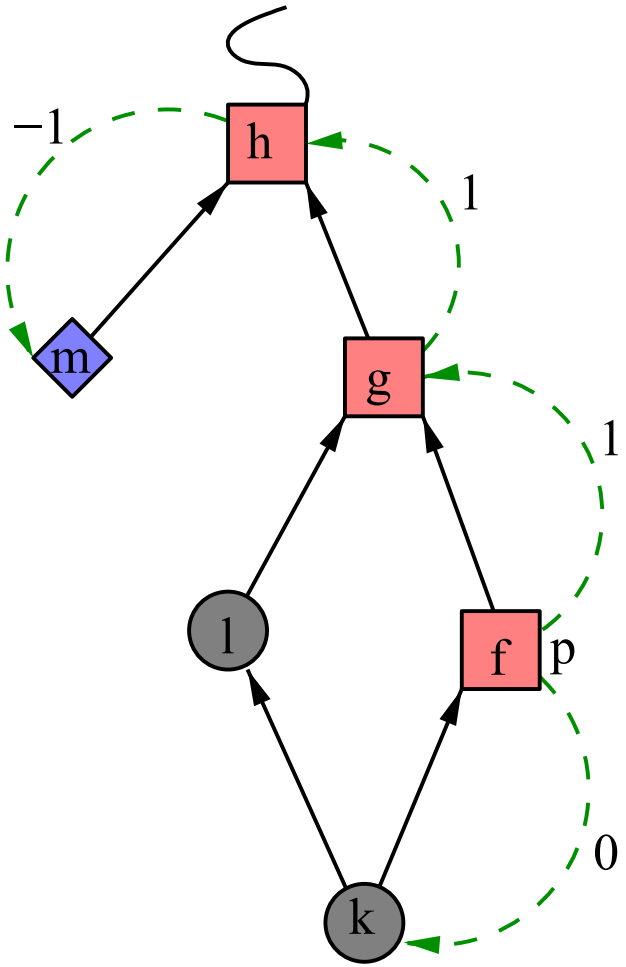
- Assigning node states correctly is not a trivial manner.
- We must respect/exploit GO's hierarchical structure .



- What is state of gene p with respect to function
  - f: 1
  - g: 1
  - h: 1
  - m: -1
  - k:
  - l:

Assigning Node States

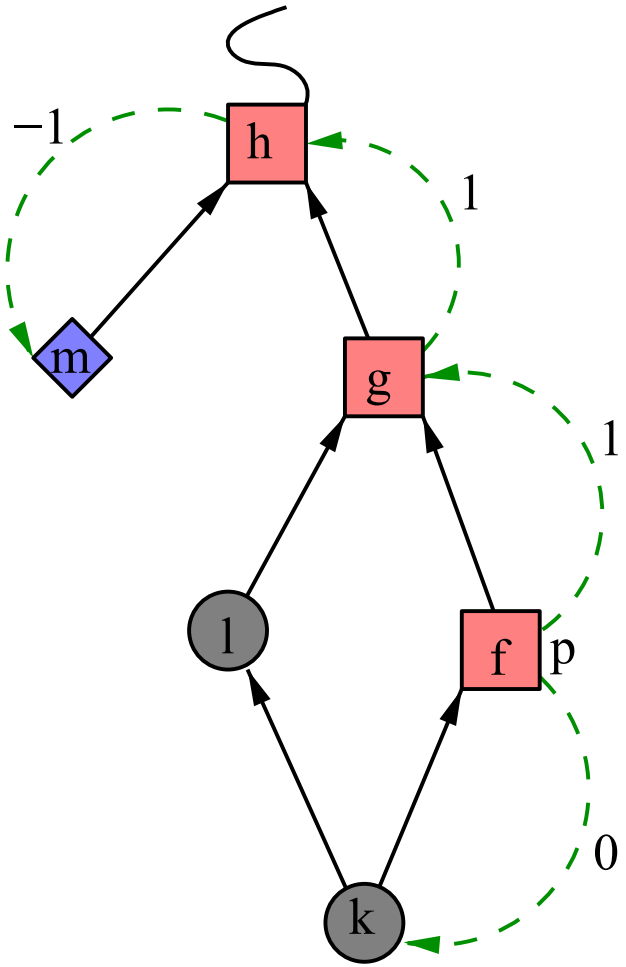
- Assigning node states correctly is not a trivial manner.
- We must respect/exploit GO's hierarchical structure .



- What is state of gene p with respect to function
  - f: 1
  - g: 1
  - h: 1
  - m: -1
  - k: 0
  - l:

Assigning Node States

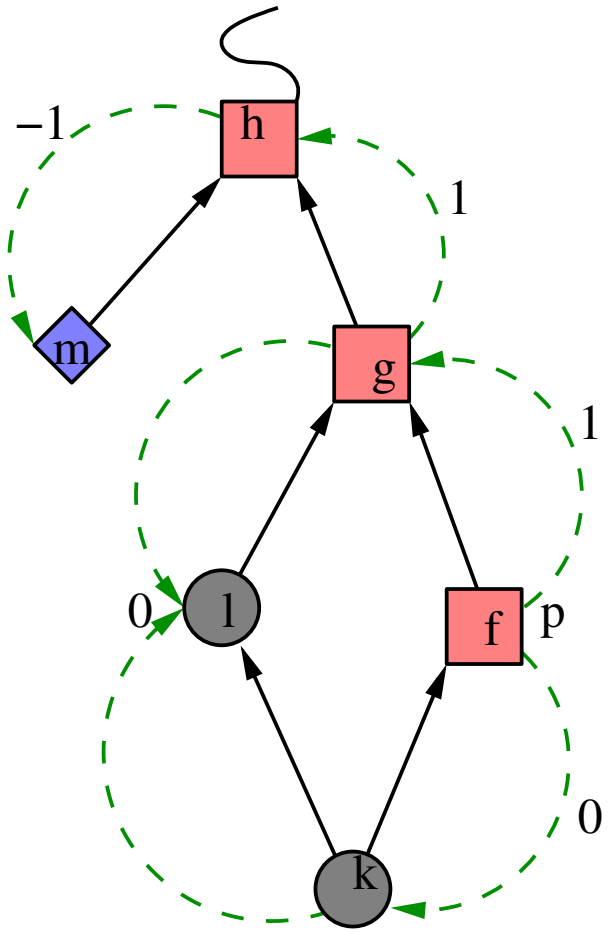
- Assigning node states correctly is not a trivial manner.
- We must respect/exploit GO's hierarchical structure .



- What is state of gene p with respect to function
  - f: 1
  - g: 1
  - h: 1
  - m: -1
  - k: 0
  - l: -1 or 0?

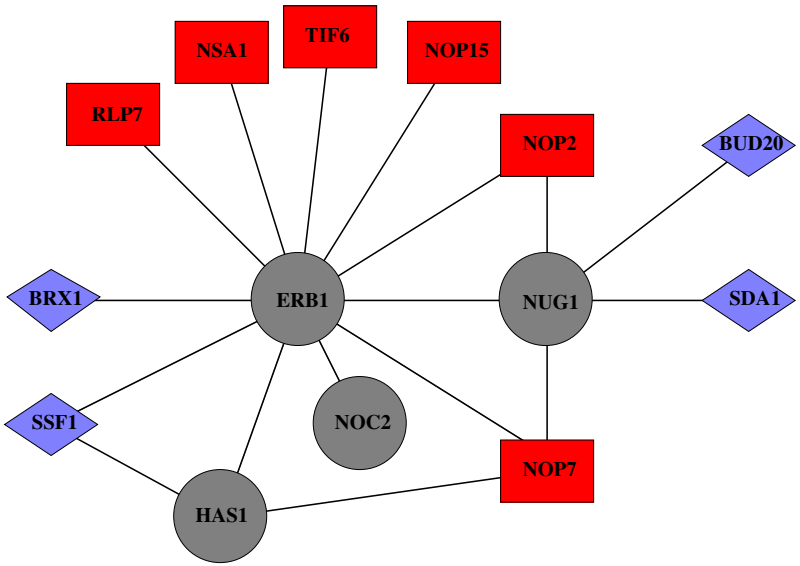
Assigning Node States

- Assigning node states correctly is not a trivial manner.
- We must respect/exploit GO's hierarchical structure .



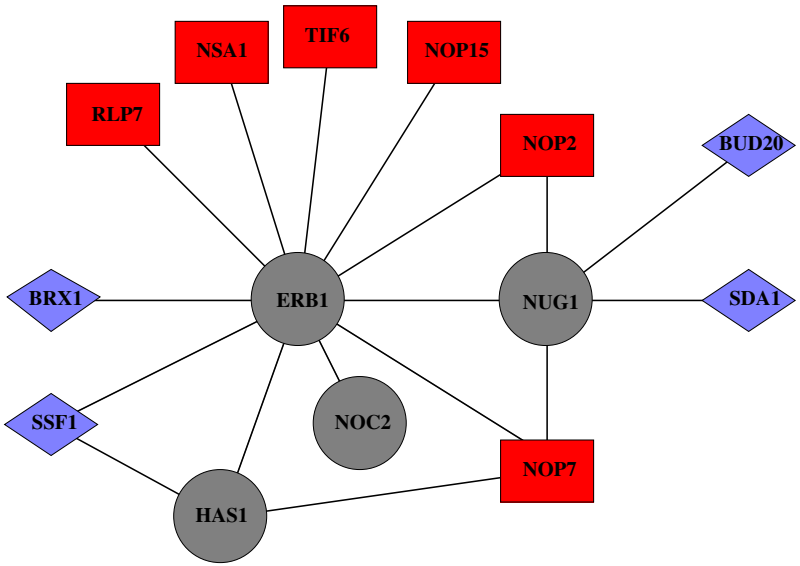
- What is state of gene p with respect to function
  - f: 1
  - g: 1
  - h: 1
  - m: -1
  - k: 0
  - l: -1 or 0? Correct state is 0.

Goal: Maximally-Consistent Assignments



- An edge is *consistent* if it is incident on nodes with the same state.
- *Maximally-consistent assignment*: number of consistent edges is maximised.

Goal: Maximally-Consistent Assignments



- An edge is *consistent* if it is incident on nodes with the same state.
- *Maximally-consistent assignment*: number of consistent edges is maximised.

Computational goal: Assign state of -1 or $+1$ to nodes with initial state 0 to achieve maximal consistency by minimising

$$E = \sum_{(u,v) \text{ is an edge}} -w_{uv} s_u s_v$$

Predict nodes in state 1 as being annotated with the function.

Minimising E

- Finding state assignments to all nodes with initial $s_u = 0$ to minimise E is NP-complete if some edge weights are negative.

Minimising E

- Finding state assignments to all nodes with initial $s_u = 0$ to minimise E is NP-complete if some edge weights are negative.
- Vasquez et al., *Nature Biotech. 2003* use a simulated annealing-based approach.

Minimising E

- Finding state assignments to all nodes with initial $s_u = 0$ to minimise E is NP-complete if some edge weights are negative.
- Vasquez et al., *Nature Biotech. 2003* use a simulated annealing-based approach.
- Our approach is based on the idea of *local updates*: each node looks at its neighbours and decides what its state should be.

Minimising E

- Finding state assignments to all nodes with initial $s_u = 0$ to minimise E is NP-complete if some edge weights are negative.
- Vasquez et al., *Nature Biotech. 2003* use a simulated annealing-based approach.
- Our approach is based on the idea of *local updates*: each node looks at its neighbours and decides what its state should be.
- Both approaches are well-known and well-studied.

Minimising E

- Finding state assignments to all nodes with initial $s_u = 0$ to minimise E is NP-complete if some edge weights are negative.
- Vasquez et al., *Nature Biotech. 2003* use a simulated annealing-based approach.
- Our approach is based on the idea of *local updates*: each node looks at its neighbours and decides what its state should be.
- Both approaches are well-known and well-studied.
- Can use minimum cuts and integer programming (Nabieva et al., Proc. ISMB 2005; Murali, Wu, and Kasif, *Nature Biotech.*, 2006).

Local Update Rule

- Activation rule is

$$s_u = \mathrm{sgn}\left(\sum_{v \in N_u} w_{uv} s_v \right),$$

where N_v = neighbours of node u.

Local Update Rule

- Activation rule is

$$s_u = \mathrm{sgn}\left(\sum_{v \in N_u} w_{uv} s_v \right),$$

  where N_v = neighbours of node u.
- Applying this rule:

Local Update Rule

- Activation rule is

$$s_u = \mathrm{sgn}\left(\sum_{v \in N_u} w_{uv} s_v \right),$$

  where N_v = neighbours of node u.
- Applying this rule:
  - Parallel update: each node updates itself in parallel with the other nodes.

Local Update Rule

- Activation rule is

$$s_u = \mathrm{sgn}\left(\sum_{v \in N_u} w_{uv} s_v \right),$$

  where N_v = neighbours of node u.
- Applying this rule:
  - Parallel update: each node updates itself in parallel with the other nodes.
  - Serial update: go through each node in sequence.

Local Update Rule

- Activation rule is

$$s_u = \operatorname{sgn}\left(\sum_{v \in N_u} w_{uv} s_v \right),$$

  where N_v = neighbours of node u.
- Applying this rule:
  - Parallel update: each node updates itself in parallel with the other nodes.
  - Serial update: go through each node in sequence.
- Stopping criterion: converge when no node's state changes.

Why does the Local Update Algorithm Converge?

- Every time a node x's state changes, E

Why does the Local Update Algorithm Converge?

- Every time a node x's state changes, E decreases.
- Let E^o and E^n be the old and values of energy, respectively.
- Let s_x^o and s_x^n be the old and new states of x, respectively.

Why does the Local Update Algorithm Converge?

- Every time a node x's state changes, E decreases.
- Let E^o and E^n be the old and values of energy, respectively.
- Let s_x^o and s_x^n be the old and new states of x, respectively.

$$E^n - E^o = \sum_{(u,v)} -w_{uv} s_u^o s_v^o - \sum_{(u,v)} -w_{uv} s_u^n s_v^n$$

- What are the maximum and minimum values of E?

Why does the Local Update Algorithm Converge?

- Every time a node x's state changes, E decreases.
- Let E^o and E^n be the old and values of energy, respectively.
- Let s_x^o and s_x^n be the old and new states of x, respectively.

$$E^n - E^o = \sum_{(u,v)} -w_{uv} s_u^o s_v^o - \sum_{(u,v)} -w_{uv} s_u^n s_v^n$$

$$= \sum_{u \in N_x} -w_{ux}(s_u^o s_x^o - s_u^n s_x^n)$$

- What are the maximum and minimum values of E?

Why does the Local Update Algorithm Converge?

- Every time a node x's state changes, E decreases.
- Let E^o and E^n be the old and values of energy, respectively.
- Let s_x^o and s_x^n be the old and new states of x, respectively.

$$\begin{aligned} E^n - E^o &= \sum_{(u,v)} -w_{uv} s_u^o s_v^o - \sum_{(u,v)} -w_{uv} s_u^n s_v^n \\ &= \sum_{u \in N_x} -w_{ux}(s_u^o s_x^o - s_u^n s_x^n) \\ &= \sum_{u \in N_x} -w_{ux} s_u^o (s_x^o - s_x^n) \end{aligned}$$

- What are the maximum and minimum values of E?

Why does the Local Update Algorithm Converge?

- Every time a node x's state changes, E decreases.
- Let E^o and E^n be the old and values of energy, respectively.
- Let s_x^o and s_x^n be the old and new states of x, respectively.

$$
\begin{aligned}
E^n - E^o &= \sum_{(u,v)} -w_{uv} s_u^o s_v^o - \sum_{(u,v)} -w_{uv} s_u^n s_v^n \\
&= \sum_{u \in N_x} -w_{ux} (s_u^o s_x^o - s_u^n s_x^n) \\
&= \sum_{u \in N_x} -w_{ux} s_u^o (s_x^o - s_x^n) \\
&= -(s_x^o - s_x^n) \sum_{u \in N_x} w_{ux} s_u^o
\end{aligned}
$$

- What are the maximum and minimum values of E?

Why does the Local Update Algorithm Converge?

- Every time a node x's state changes, E decreases.
- Let E^o and E^n be the old and values of energy, respectively.
- Let s_x^o and s_x^n be the old and new states of x, respectively.

$$\begin{aligned} E^n - E^o &= \sum_{(u,v)} -w_{uv} s_u^o s_v^o - \sum_{(u,v)} -w_{uv} s_u^n s_v^n \\ &= \sum_{u \in N_x} -w_{ux}(s_u^o s_x^o - s_u^n s_x^n) \\ &= \sum_{u \in N_x} -w_{ux} s_u^o (s_x^o - s_x^n) \\ &= -(s_x^o - s_x^n) \sum_{u \in N_x} w_{ux} s_u^o \end{aligned}$$

$$\operatorname{sgn}(E^n - E^o) = \operatorname{sgn}(s_x^n - s_x^o) \operatorname{sgn}\left(\sum_{u \in N_x} w_{ux} s_u^o \right)$$

- What are the maximum and minimum values of E?

Why does the Local Update Algorithm Converge?

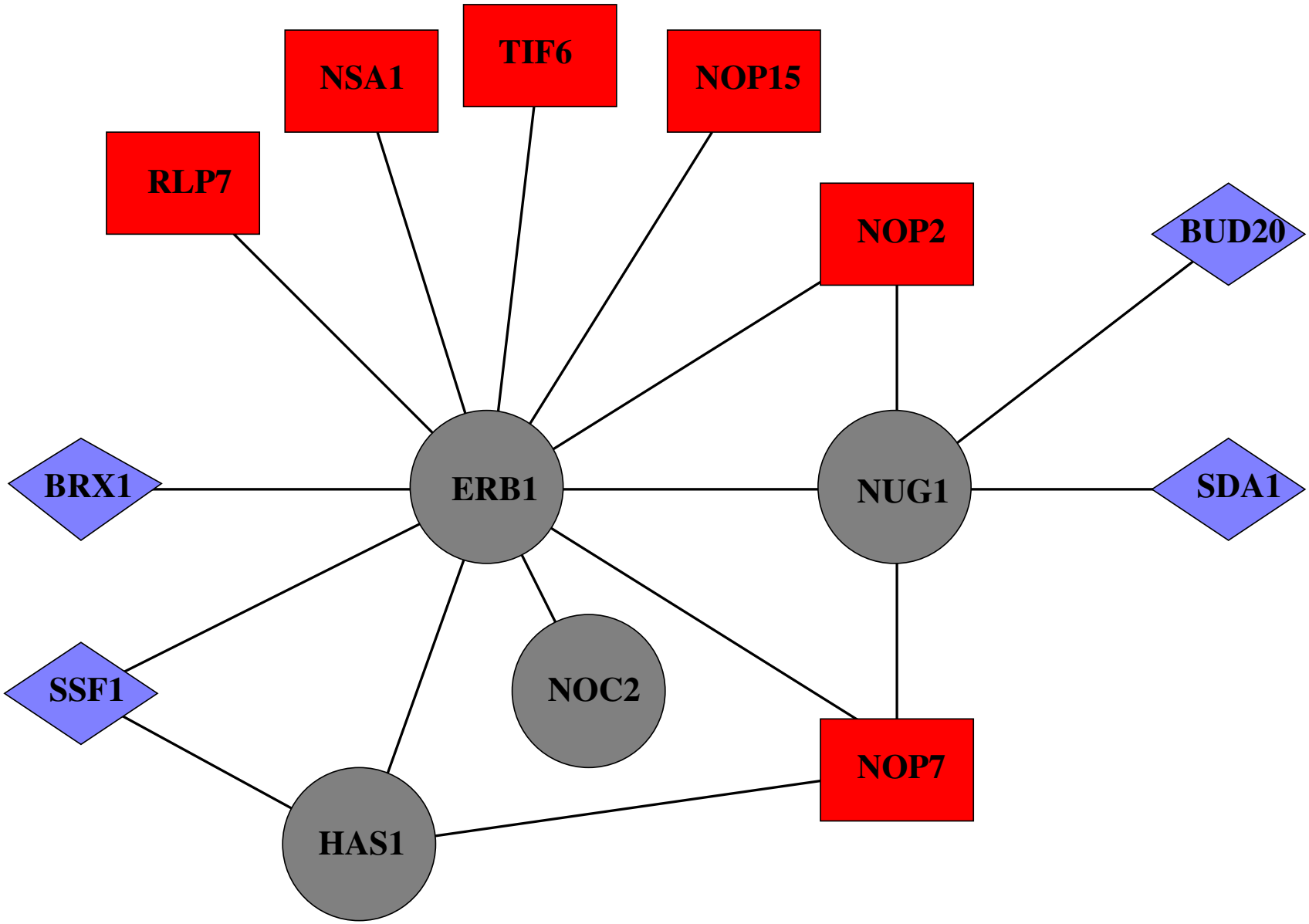
- Every time a node x's state changes, E decreases.
- Let E^o and E^n be the old and values of energy, respectively.
- Let s_x^o and s_x^n be the old and new states of x, respectively.

$$\begin{aligned} E^n - E^o &= \sum_{(u,v)} -w_{uv} s_u^o s_v^o - \sum_{(u,v)} -w_{uv} s_u^n s_v^n \\ &= \sum_{u \in N_x} -w_{ux}(s_u^o s_x^o - s_u^n s_x^n) \\ &= \sum_{u \in N_x} -w_{ux} s_u^o (s_x^o - s_x^n) \\ &= -(s_x^o - s_x^n) \sum_{u \in N_x} w_{ux} s_u^o \end{aligned}$$

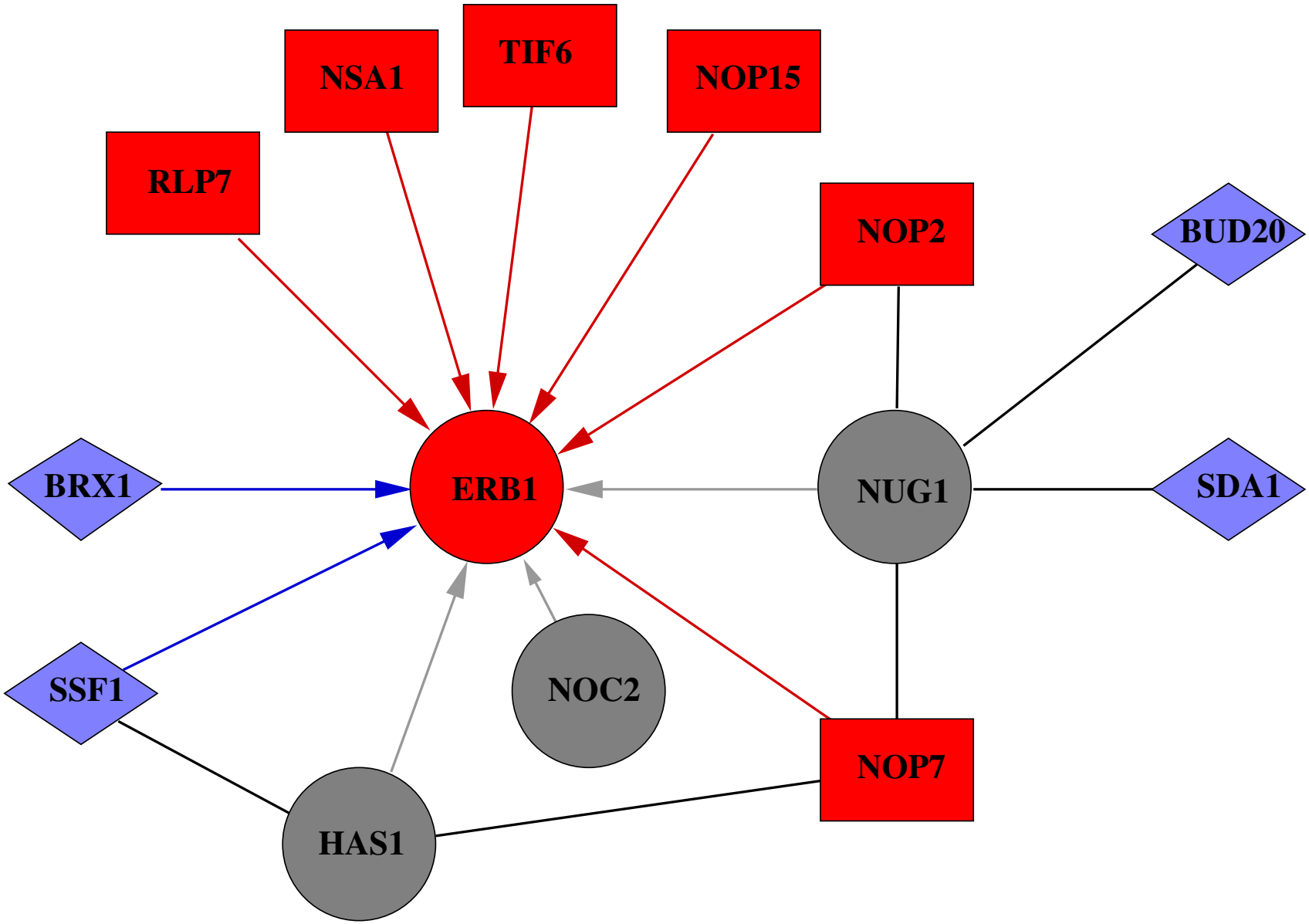
$$\mathrm{sgn}(E^n - E^o) = \mathrm{sgn}(s_x^n - s_x^o)\mathrm{sgn}\left(\sum_{u \in N_x} w_{ux} s_u^o\right) \qquad = \mathrm{sgn}(s_x^n - s_x^o)s_x^n = 1$$

- What are the maximum and minimum values of E?

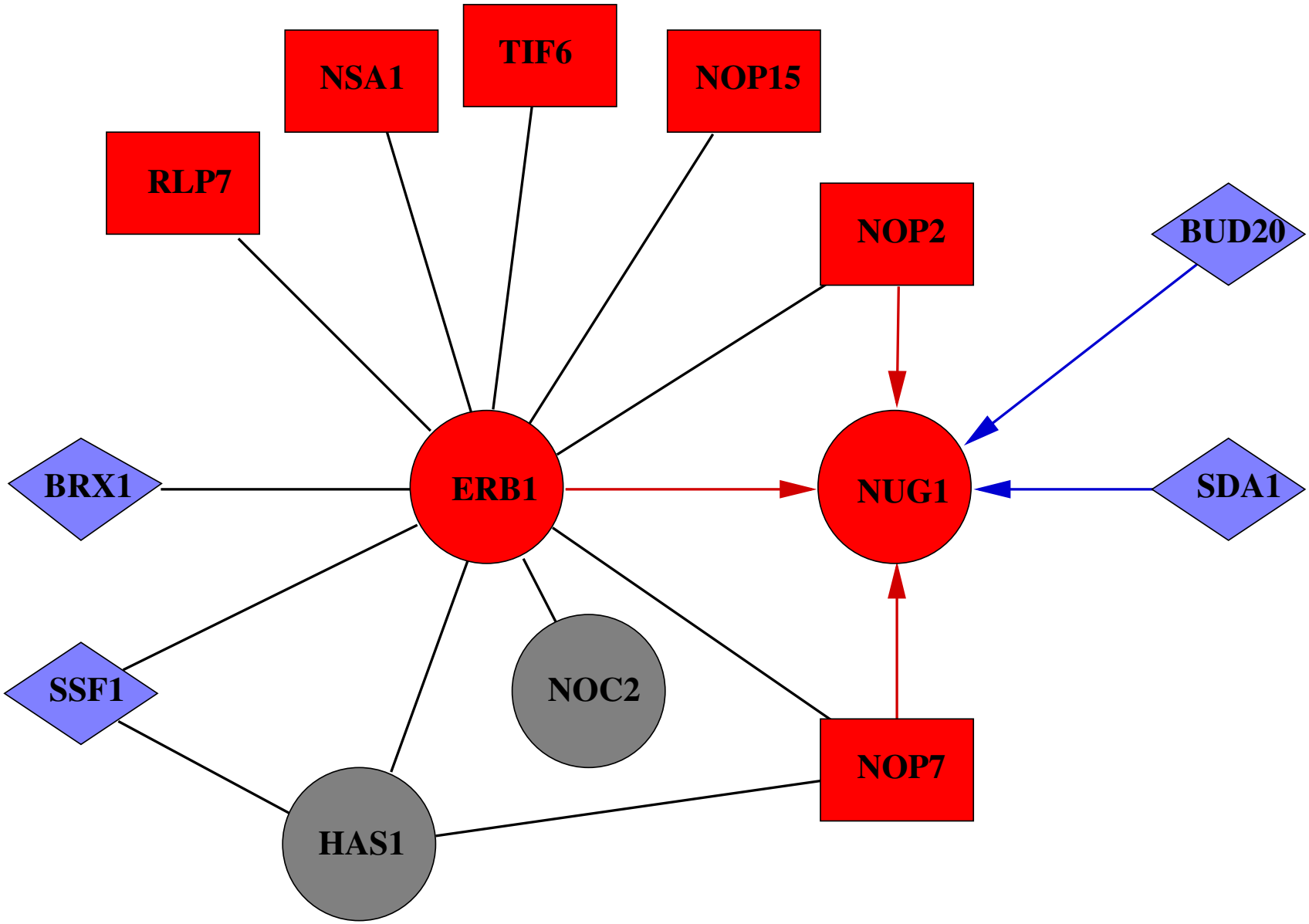
Example of Local Updates



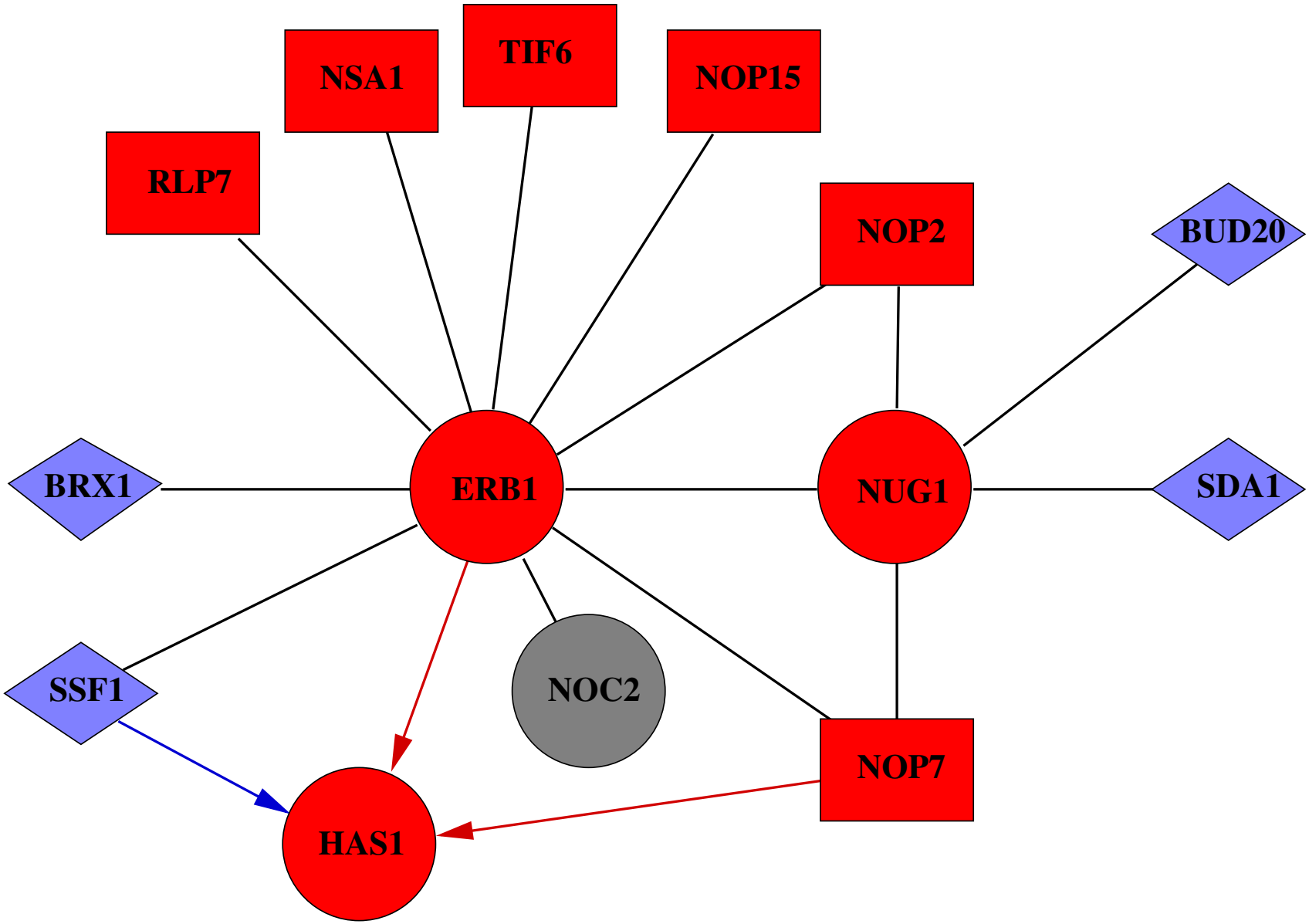
Example of Local Updates



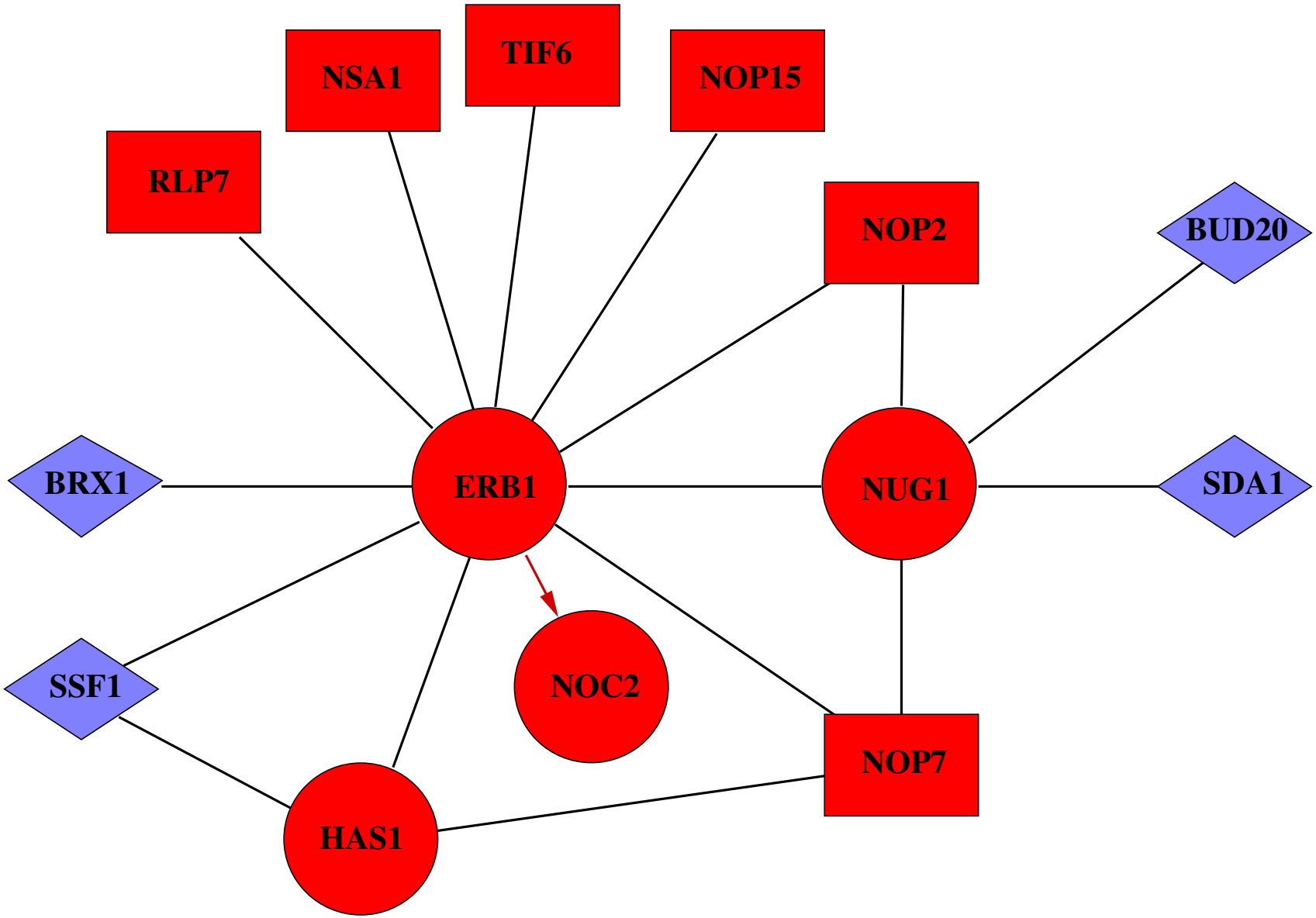
Example of Local Updates



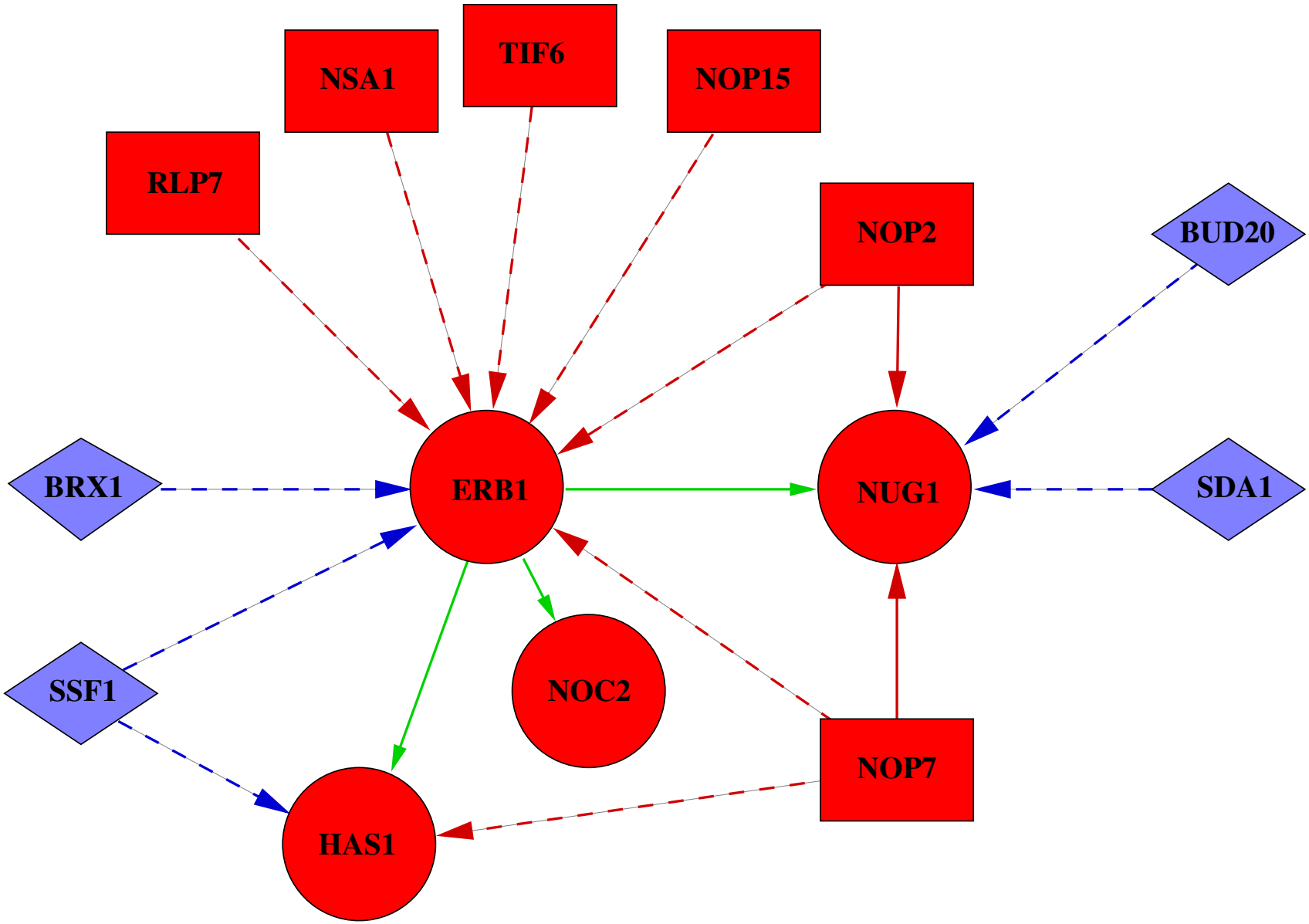
Example of Local Updates



Example of Local Updates



Example of Local Updates



Data Sets

- Interactions: General Repository of Interaction Datasets (GRID).
- Microarray: *Functional discovery via a compendium of expression profiles*. Hughes TR et al. *Cell*. 2000 102: 109–26.
- Functional Annotations: Gene Ontology, three categories are biological process, molecular function, and cellular component.

Cleaning Up PPI Network

- GRID data set has 4711 genes and 13607 interactions.
- GRID data set has information on publications.

ORF_A	ORF_B	EXPERIMENTAL_SYSTEM	SOURCE	PUBMED_ID
YER006W	YPL211W	Affinity Precipitation	Bassler et al.	;11583615;
YDL140C	YBR154C	Two Hybrid	BIND	;2496296;9207794;10393904;

- We only consider interactions reported by at least two different experiments to obtain 997 interactions between 1004 genes.

Data Integration

- Unweighted: $w_{uv} = 1$.
- Integrated: w_{uv} is the absolute value of correlation coefficient of the expression profiles of gene u and gene v in the "Compendium" data set.

Leave One-Out Cross Validation

- For each function f,
  1. for each gene u annotated with f, set initial value of $s_u = 0$ and compute state assigned to u by the Hopfield network.
  2. Perform a similar operation for each gene not annotated with f.

Leave One-Out Cross Validation

- For each function f,
  1. for each gene u annotated with f, set initial value of $s_u = 0$ and compute state assigned to u by the Hopfield network.
  2. Perform a similar operation for each gene not annotated with f.
- Measurement of performance:

- True positive: $s_u : 1 \rightarrow 0 \rightarrow 1$
- False positive: $s_u : -1 \rightarrow 0 \rightarrow 1$
- True negative: $s_u : -1 \rightarrow 0 \rightarrow -1$
- False negative: $s_u : 1 \rightarrow 0 \rightarrow -1$

- Precision = TP/(TP + FP)
- Sensitivity = Recall = TP/(TP + FN)
- F-measure = Harmonic mean of precision and recall.

k-fold cross validation

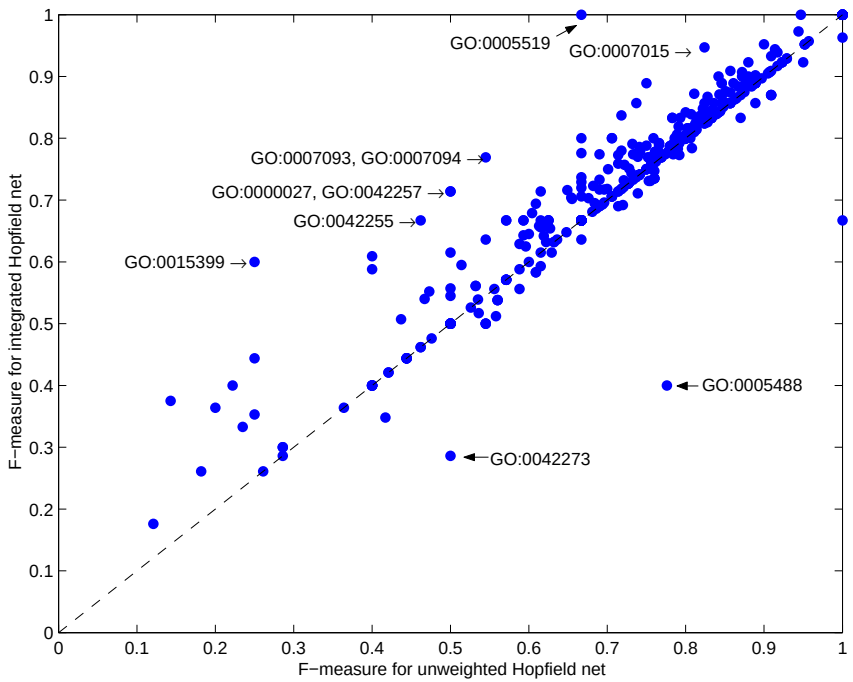
1. Partition union of positive and negative examples into k groups, uniformly at random.
2. For each group, use algorithm to predict the state of each positive/negative example in that group using all other examples.
3. Sort all positive and negative examples in decreasing order of prediction confidence.
4. For each threshold on prediction confidence, compute the number of true positives (tp), false positives (fp), true negatives (tn), and false negatives (fn).
5. For each threshold on prediction confidence, compute precision ($tp/(tp + fp)$), recall ($tp/(tp + fn)$), and false positive rate ($fp/(fp + tn)$).
6. As prediction confidence varies, plot precision against recall.

Results for Both Variants

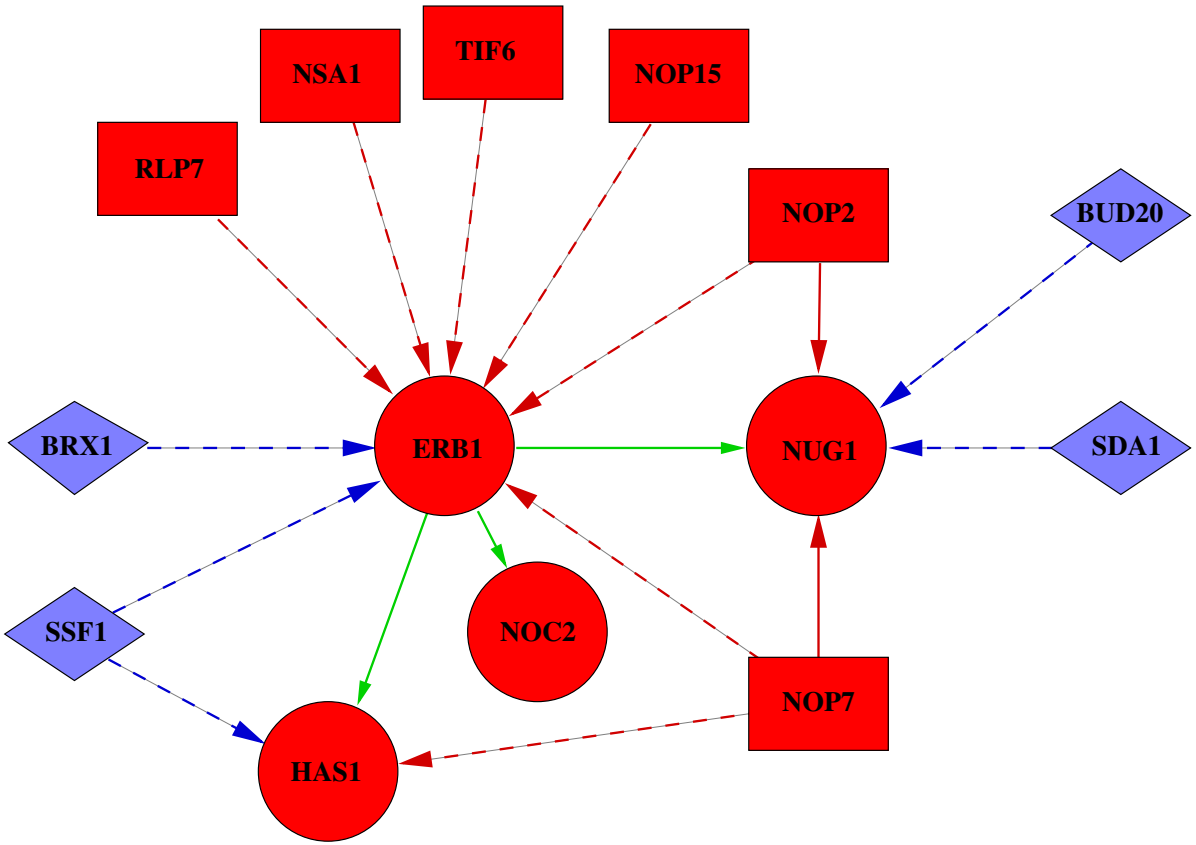
1. Overall comparison of cross-validation.
2. Specific examples of genes that perform better on cross-validation (see paper).
3. Novel functional annotations.

Overall Cross-Validation Results

- Restricted to 828 functions for which F-score > 0.
- Unweighted network: Precision $= 94\%$, Recall $= 64\%$.
- Integrated network: Among 440 functions for which we make at least one novel prediction,
  - 168 function had better F-measures, 227 the same, and 45 smaller F-measures in the integrated network.



Novel Functional Annotations



- ERB1, HAS1, and NUG1: validated to have the function "rRNA processing."
- NOC2: validated to have the function "ribosome assembly and ribosome-nuclear export."

Novel Functional Annotations

- NHP10
  - biological process *chromatin modeling* and cellular component *chromatin remodeling complex*.
  - HMG1 proteins are involved in chromatin structure.
- UFO1
  - cellular component *nuclear ubiquitin ligase complex*
  - molecular function *ubiquitin-protein ligase activity* and biological processes *ubiquitin-dependent protein catabolism*.
- PKC1
  - cellular component *1,3 beta-glucan synthase complex*.
  - known: cellular component *intracellular* and biological processes *cell wall organization and biogenesis*.

More Novel Functional Annotations

- YKL067W
  - biological process *signal transduction* and cellular component *spindle pole body*.
  - molecular function *nucleoside-diphosphate kinase (NDK) activity*; NDK interferes with the mating pheromone signal transduction in *S. pombe*.
- YCR099C and YBL059W
  - biological process *ER to Golgi transport* and cellular component *COPII vesicle coat*.
  - Vesicles with COPII coats are found associated with ER membranes at steady state.

Overall Correctness of Predictions

- 207 predictions for functions with F-score $> 75\%$.
- 15 predictions are correct.
- 11 predictions at distance 1 from true function.
- 49 predictions at distance 2 from true function.
- Remaining predictions not validated.
- Validated functions include nucleolus, chromatin remodeling complex, snoRNA binding, RNA binding, vesicle-mediated transport.

Features of the GAIN System

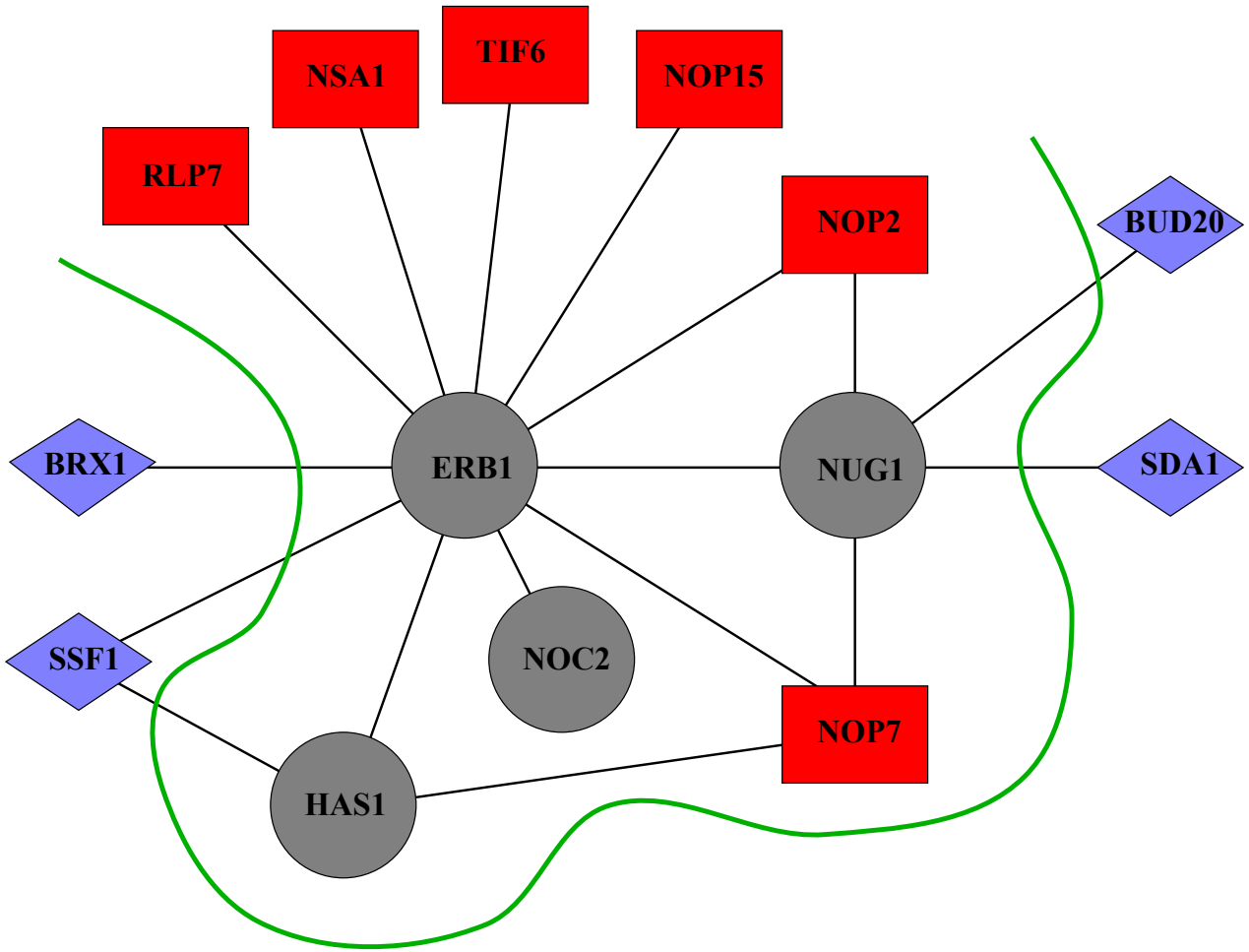
- Systematic algorithm for propagating evidence in an FLN.
- Clean separation between construction of functional links and prediction of function.
- For each function, predictions are maximally consistent.
- Each prediction associated with measures of confidence.
- Propagation diagrams provide intuitive visualisation of evidence flow.
- VIRGO webserver for invoking GAIN and querying and browsing its predictions.

Algorithms: Local and Local+

Local	Local+
$s_u = \frac{\sum_{v \in N_u} w_{uv} s_v}{\sum_{v \in N_u} w_{uv}}$	$s_u = \frac{\sum_{v \in N_u} w_{uv} s_v}{\sum_{v \in N_u} w_{uv}}$

- N_u is the set of neighbours of gene u.
- Local+ does not use negative examples, i.e., s_v is initially 0 for negative examples.

Graph cuts



- Transform the problem to computing minimum cuts in a flow network (Nabieva et al., Proc. ISMB 2005; Murali, Wu, and Kasif, *Nature Biotech.*, 2006).

Algorithm: FunctionalFlow

(Nabieva et al., ISMB 2005.)

- No negative examples.
- Each node sends flow to or receives flow from each neighbour.
- $s(v)$ is the total inflow into node over multiple phases.
- Number of phases is input to the algorithm (half the diameter of the network suggested.)

Algorithm: FunctionalFlow

(Nabieva et al., ISMB 2005.)

- No negative examples.
- Each node sends flow to or receives flow from each neighbour.
- $s(v)$ is the total inflow into node over multiple phases.
- Number of phases is input to the algorithm (half the diameter of the network suggested.)

$$g_0(u, v) = 0$$

$$s_0(u) = \begin{cases} \infty & \text{if } u \text{ is a positive example} \\ 0 & \text{otherwise} \end{cases}$$

Algorithm: FunctionalFlow

(Nabieva et al., ISMB 2005.)

- No negative examples.
- Each node sends flow to or receives flow from each neighbour.
- $s(v)$ is the total inflow into node over multiple phases.
- Number of phases is input to the algorithm (half the diameter of the network suggested.)

$$g_0(u,v) = 0$$

$$s_0(u) = \begin{cases} \infty & \text{if } u \text{ is a positive example} \\ 0 & \text{otherwise} \end{cases}$$

$$g_t(u,v) = \begin{cases} 0 & \text{if } s_{t-1}(u) < s_{t-1}(v) \\ \min\left(w_{uv}, s_{t-1}(u)\frac{w_{uv}}{\sum_{y \in N_u} w_{uy}}\right) & \text{otherwise} \end{cases}$$

Algorithm: FunctionalFlow

(Nabieva et al., ISMB 2005.)

- No negative examples.
- Each node sends flow to or receives flow from each neighbour.
- $s(v)$ is the total inflow into node over multiple phases.
- Number of phases is input to the algorithm (half the diameter of the network suggested.)

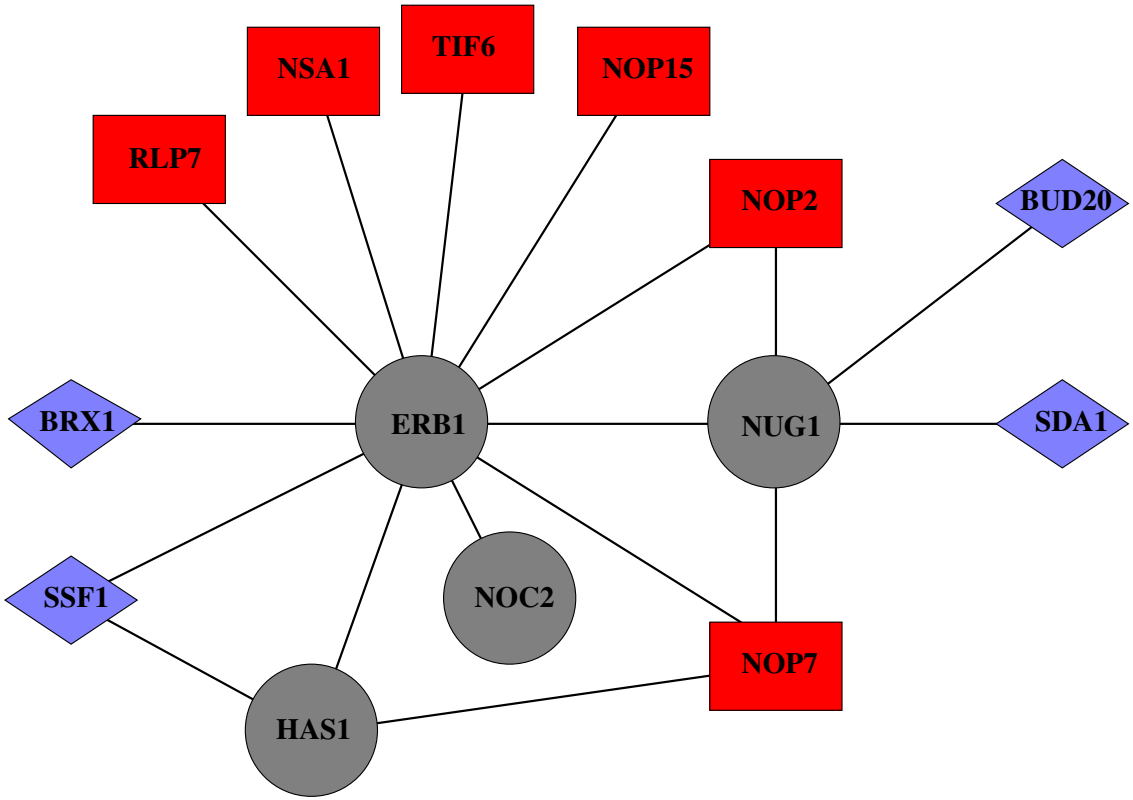
$$g_0(u,v) = 0$$

$$s_0(u) = \begin{cases} \infty & \text{if } u \text{ is a positive example} \\ 0 & \text{otherwise} \end{cases}$$

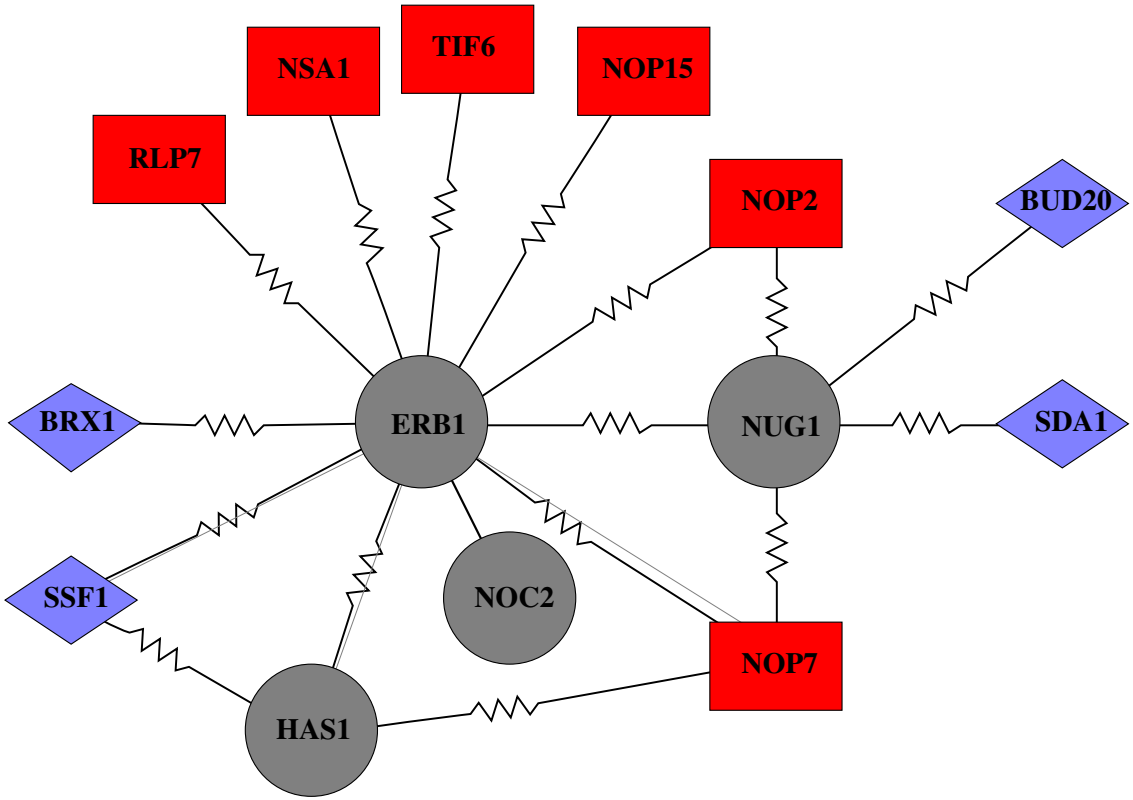
$$g_t(u,v) = \begin{cases} 0 & \text{if } s_{t-1}(u) < s_{t-1}(v) \\ \min\left(w_{uv}, s_{t-1}(u)\frac{w_{uv}}{\sum_{y \in N_u} w_{uy}}\right) & \text{otherwise} \end{cases}$$

$$s_t(u) = s_{t-1}(u) + \sum_{v \in N_u} \left(g_t(v,u) - g_t(u,v)\right)$$

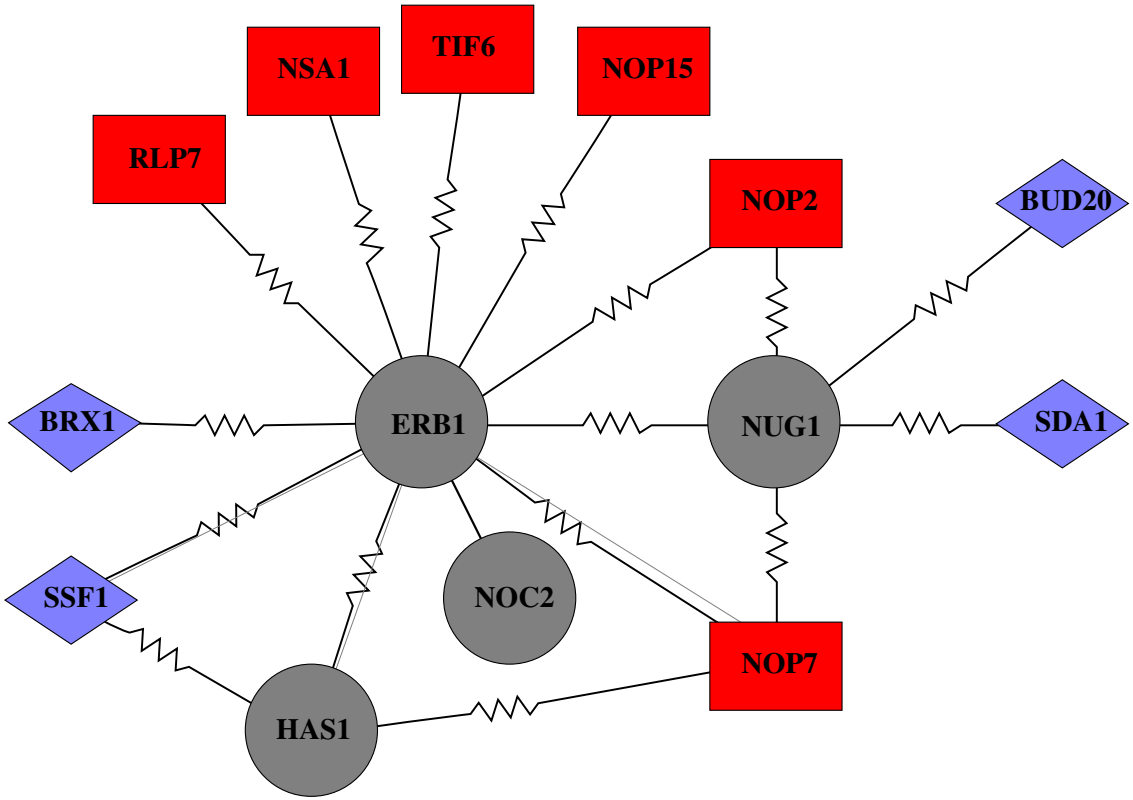
Algorithm: SinkSource



Algorithm: SinkSource



Algorithm: SinkSource



- Compute voltage at each unknown example by minimising

$$\sum_{(u,v)} w_{uv}(s_u - s_v)^2$$

- Solve linear system of equations:

$$s_v = \frac{\sum_u w_{uv} s_u}{\sum_u w_{uv}}$$

Matrix Formulation of SinkSource

$$s_v = \frac{\sum_u w_{uv} s_u}{\sum_u w_{uv}}$$

$$\left(\sum_u w_{uv} \right) s_v = \sum_u w_{uv} s_u$$

Define $y_u = s_u$ only for positive and negative examples and split RHS,

$$\left(\sum_u w_{uv} \right) s_v = \sum_u w_{uv} s_u + \sum_u w_{uv} y_u$$

Matrix Formulation of SinkSource

$$s_v = \frac{\sum_u w_{uv} s_u}{\sum_u w_{uv}}$$

$$\left(\sum_u w_{uv}\right) s_v = \sum_u w_{uv} s_u$$

Define $y_u = s_u$ only for positive and negative examples and split RHS,

$$\left(\sum_u w_{uv}\right) s_v = \sum_u w_{uv} s_u + \sum_u w_{uv} y_u$$

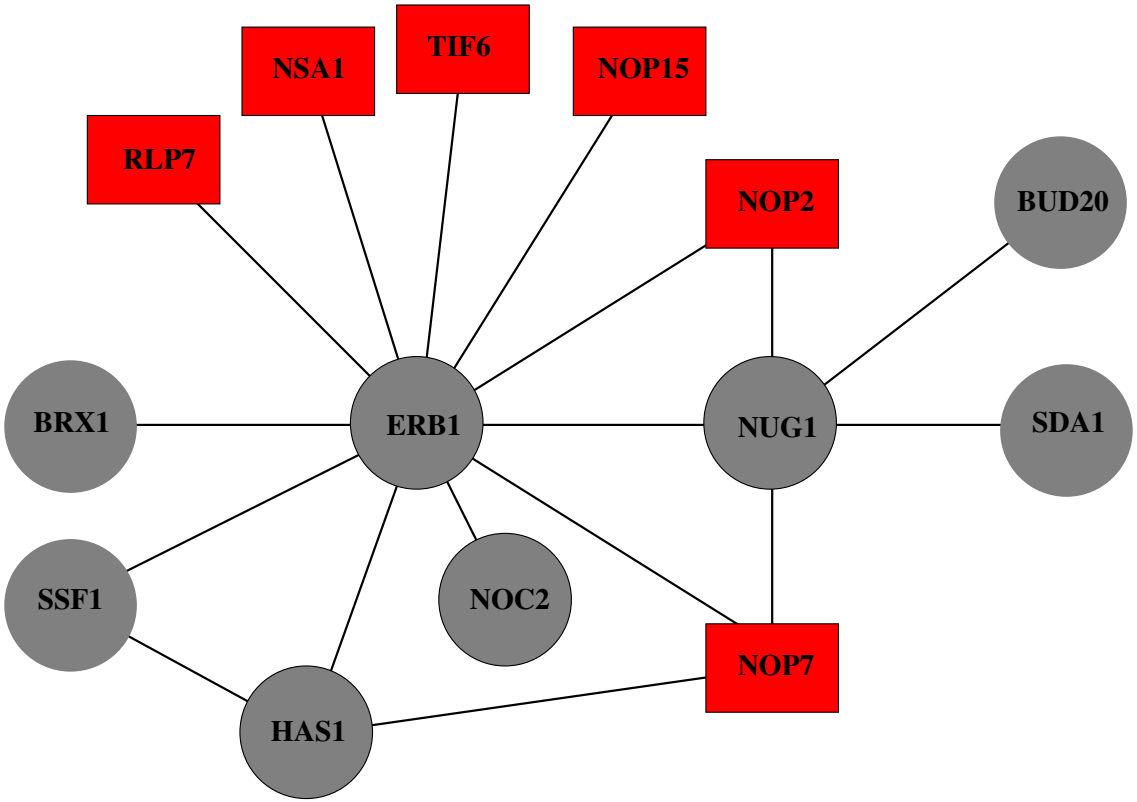
Define $W = [w_{uv}]$, $D = [\sum_u w_{uv}]$, $L = D - W$, $s = [s_u]$ and $y = [\sum_u w_{uv} y_u]$.

$$Ds = Ws + y$$

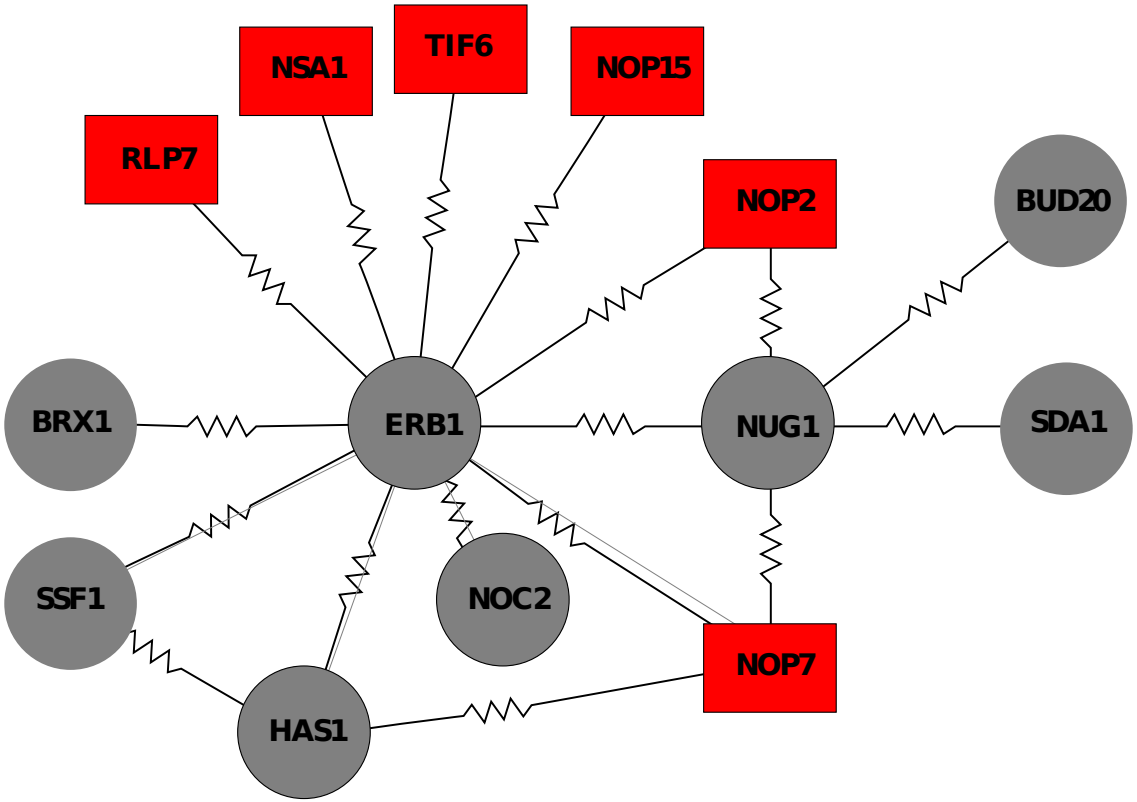
$$(D - W)s = Ls = y$$

$$s = L^{-1} y$$

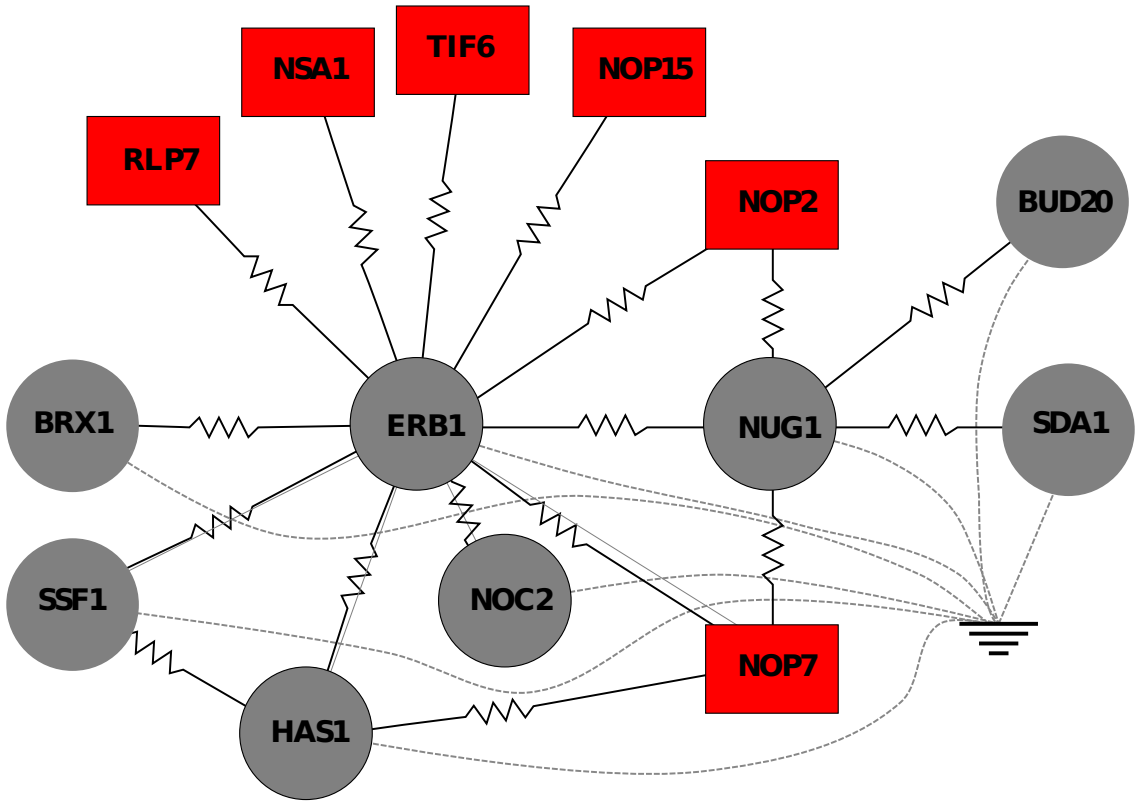
Algorithm: SinkSource+



Algorithm: SinkSource+

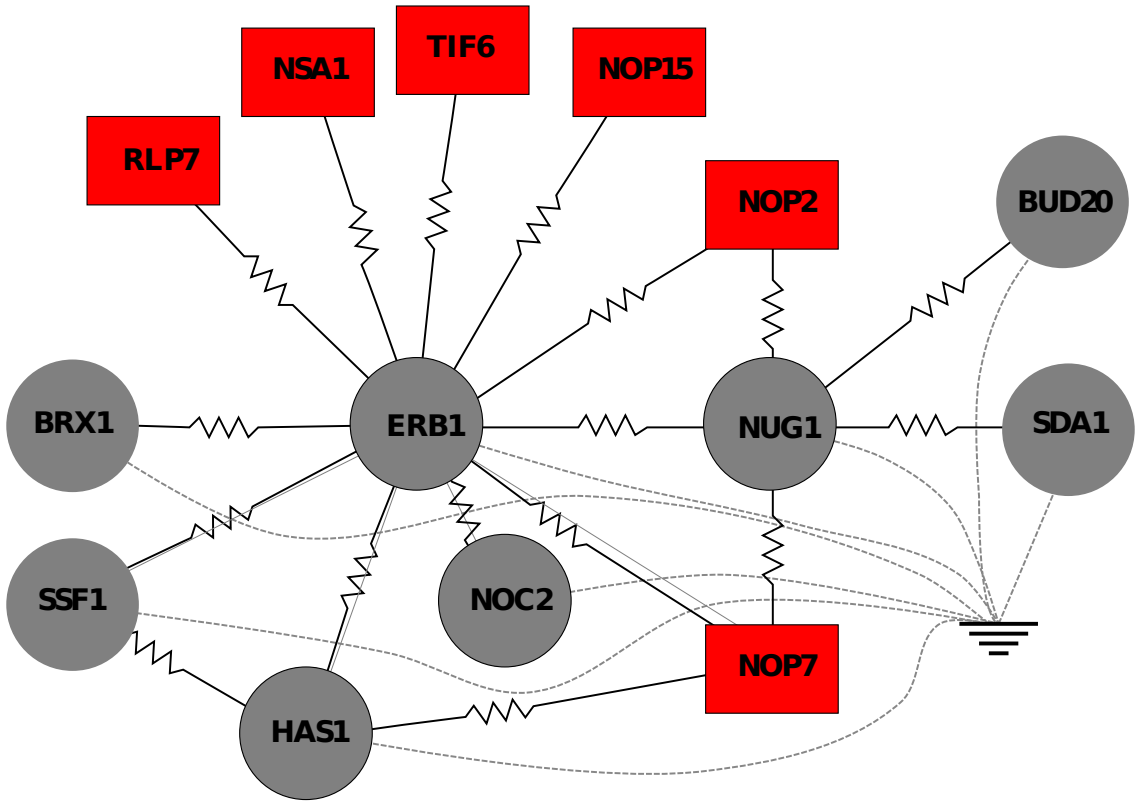


Algorithm: SinkSource+



- Compute voltage at each unknown example by minimising $\sum_{(u,v)} w_{uv}(s_u - s_v)^2$
- Solve linear system of equations: $s_v = \frac{\sum_u w_{uv} s_u}{\sum_u w_{uv}}$

Algorithm: SinkSource+



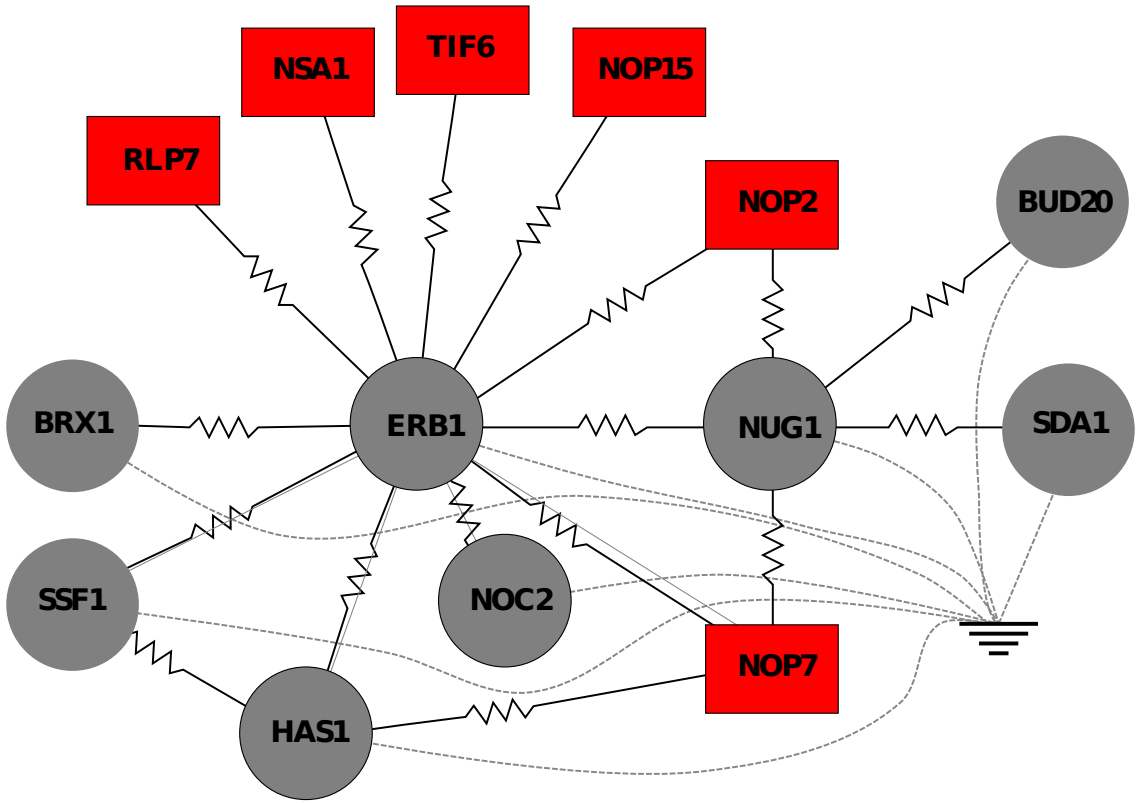
- Compute voltage at each unknown example by minimising

$$\sum_{(u,v)} w_{uv}(s_u - s_v)^2 + \lambda \sum_v s_v^2$$

- Solve linear system of equations:

$$s_v = \frac{\sum_u w_{uv} s_u}{\sum_u w_{uv}}$$

Algorithm: SinkSource+



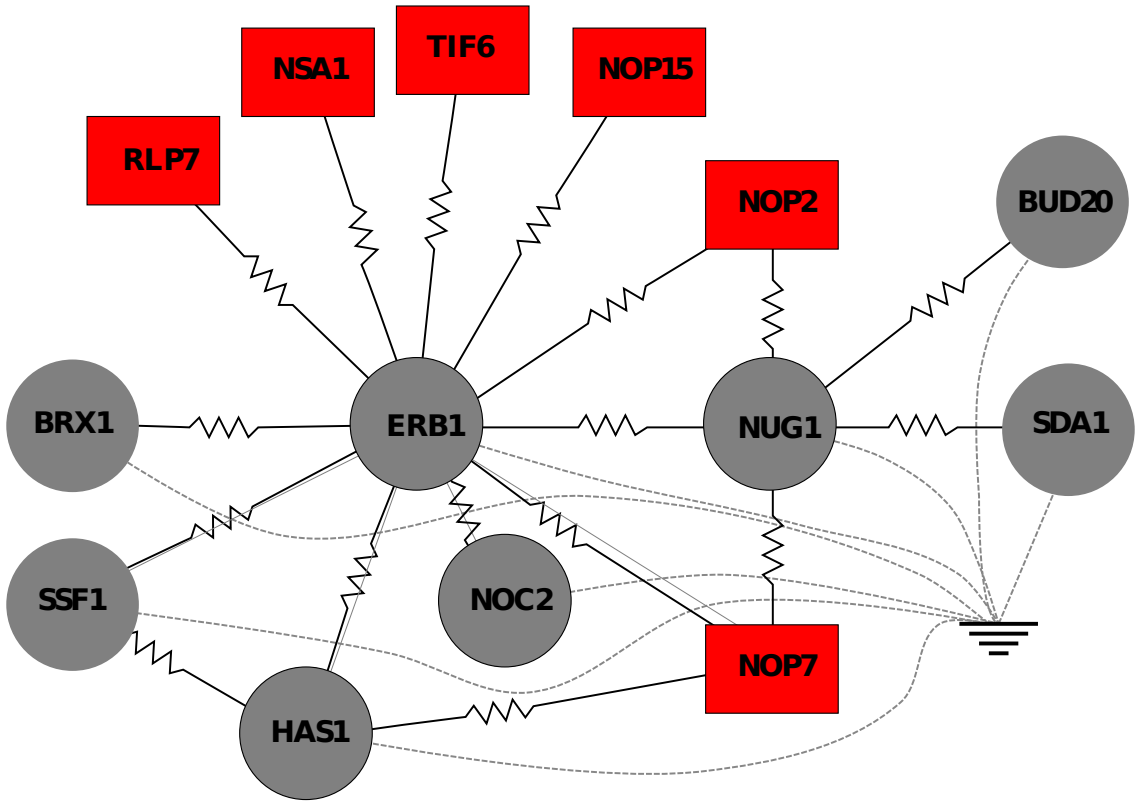
- Compute voltage at each unknown example by minimising

$$\sum_{(u,v)} w_{uv}(s_u - s_v)^2 + \lambda \sum_v s_v^2$$

- Solve linear system of equations:

$$s_v = \frac{\sum_u w_{uv} s_u}{\lambda + \sum_u w_{uv}}$$

Algorithm: SinkSource+



- Compute voltage at each unknown example by minimising

$$\sum_{(u,v)} w_{uv}(s_u - s_v)^2 + \lambda \sum_v s_v^2$$

- Solve linear system of equations:

$$s_v = \frac{\sum_u w_{uv} s_u}{\lambda + \sum_u w_{uv}}$$

- Matrix form is $s = (\lambda I + L)^{-1} y$.