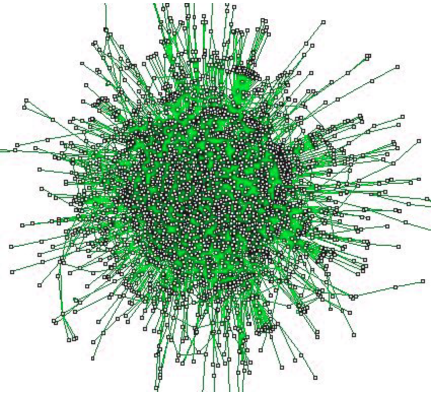
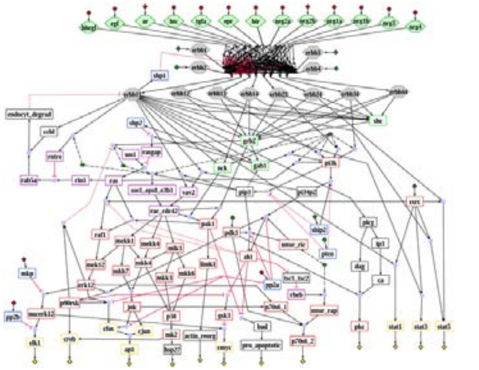
Protein networks: from topology to logic





Roded Sharan

School of Computer Science

Tel Aviv University

Motivation

- Holy grail: a working model of the cell
- More focused: model a process of interest
- Current experimental techniques yield only the global wiring of proteins
- What is missing:
  - Directionality information
  - Process specific subnetwork
  - The underlying logic

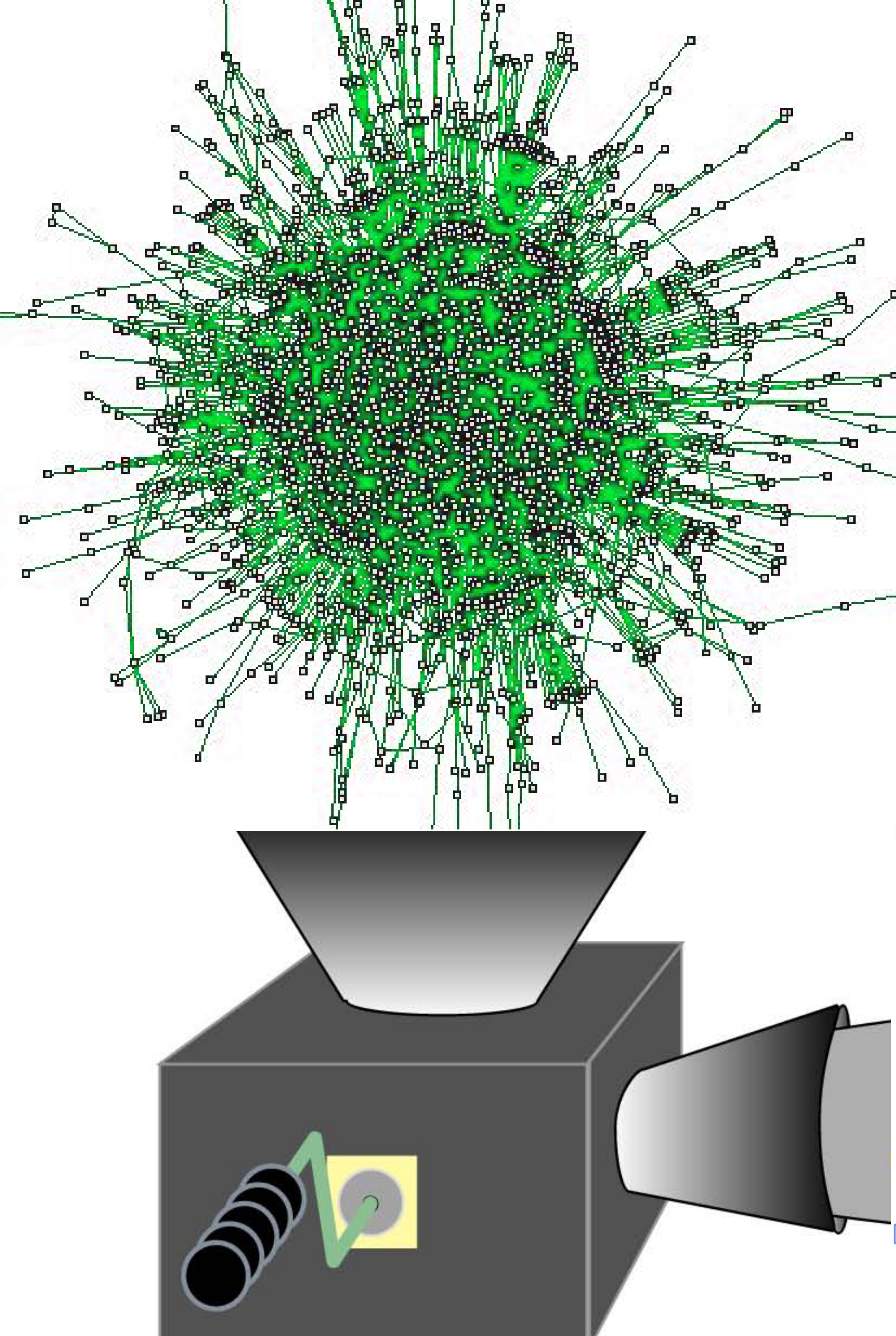
Sharan, EMBO Reports'13

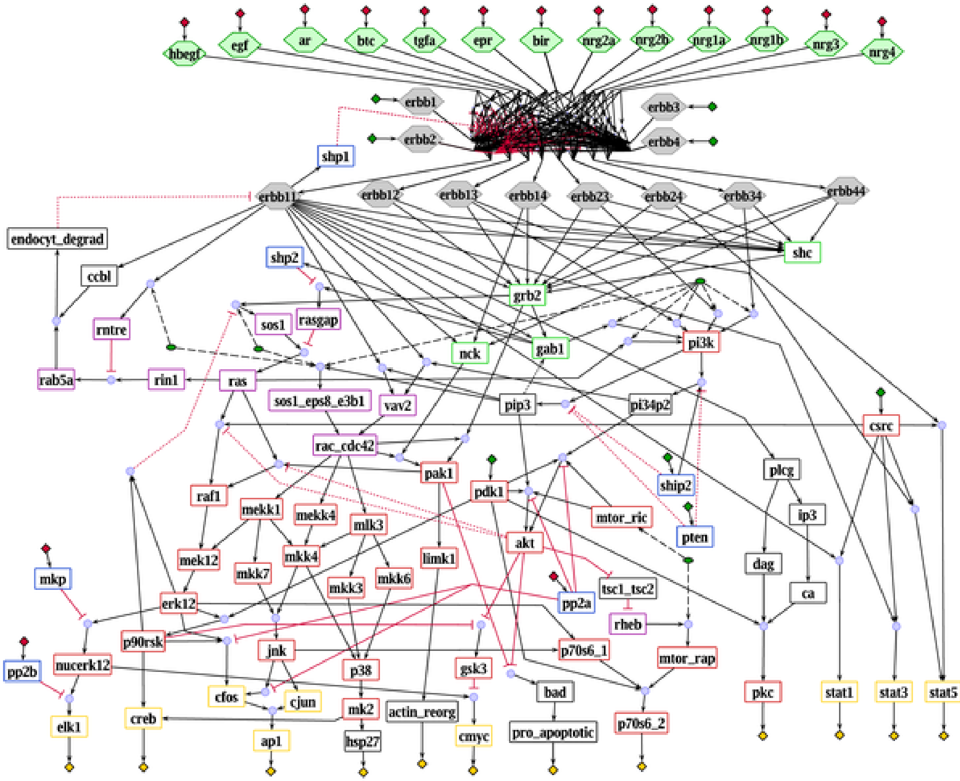
Our vision

Network Orientation

Subnetwork inference

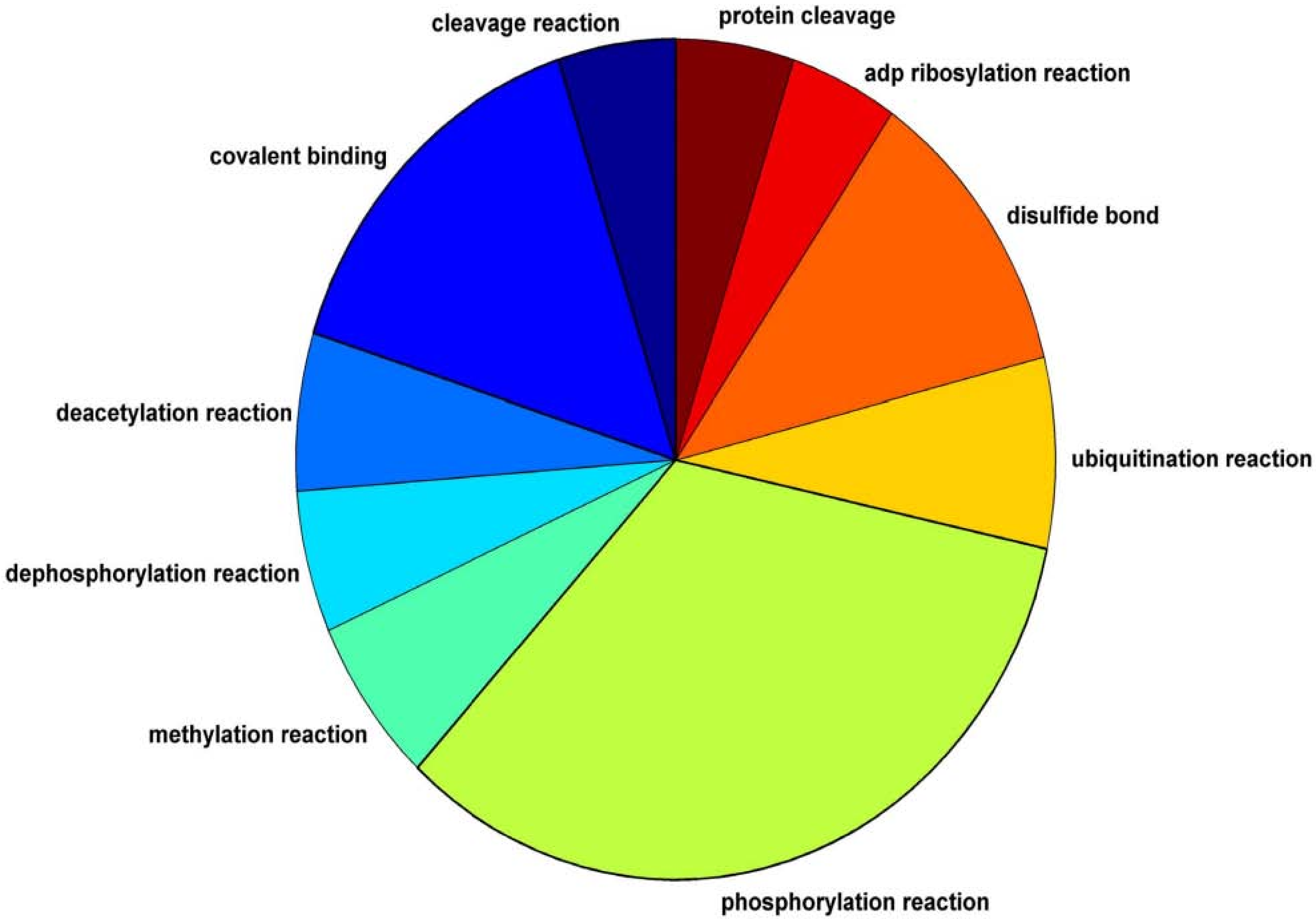
Logical model learning





Network orientation

Are protein interactions directed?



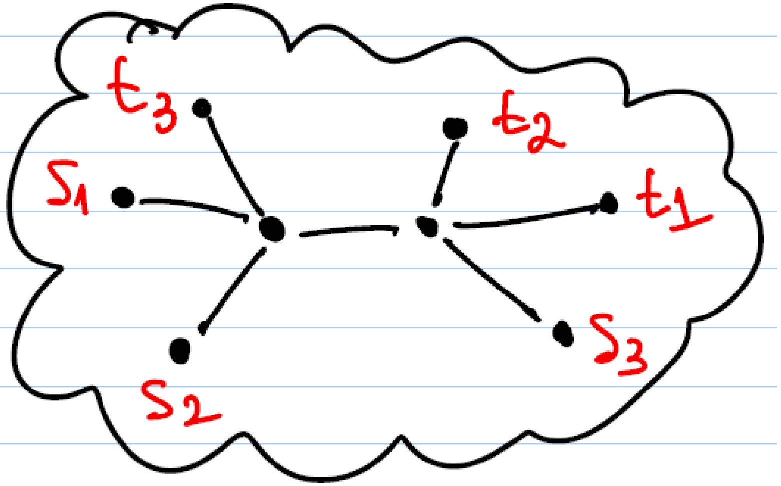
Silberberg et al., PLoS One'14

The computational problem

- Directionality is not revealed by the experiments
- Indirect information is obtained from knockout experiments:
  - Observe: knockout of protein *s* affects *t*
  - Assume: there is a directed *(s,t)* path
- <u>Goal:</u> predict directions to maximize #KO-pairs that can be "explained"

MAXIMUM GRAPH ORIENTATION

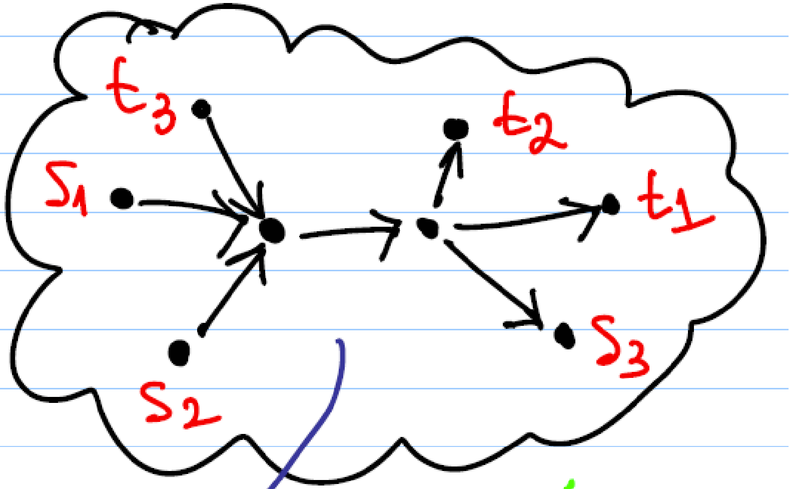
- Input: undirected graph $G=(V,E)$ with n vertices
  source-target pairs $(s_1,t_1), \dots, (s_k,t_k)$



MAXIMUM GRAPH ORIENTATION

- Input: undirected graph $G=(V,E)$ with n vertices
  source-target pairs $(s_1,t_1),\dots,(s_k,t_k)$

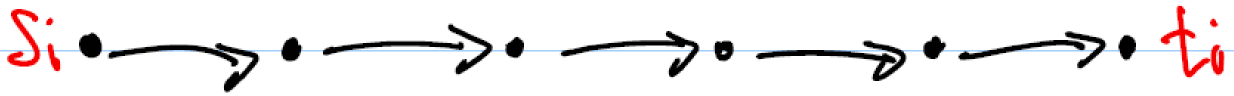
- goal: compute an orientation
  in which the number of
  connected pairs is maximized



(s_1,t_1) ✓

(s_2,t_2) ✓

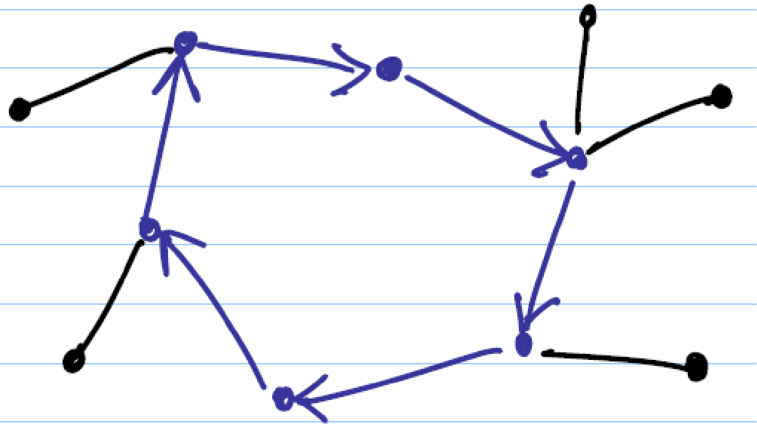
(s_3,t_3) ✗



MAXIMUM GRAPH ORIENTATION

- Input: undirected graph $G=(V,E)$ with n vertices
  source-target pairs $(s_1,t_1),\ldots,(s_k,t_k)$

- goal: compute an orientation
  in which the number of
  connected pairs is maximized



- remark: we may assume that the underlying
  graph is a tree

Maximum Tree Orientation (MTO)

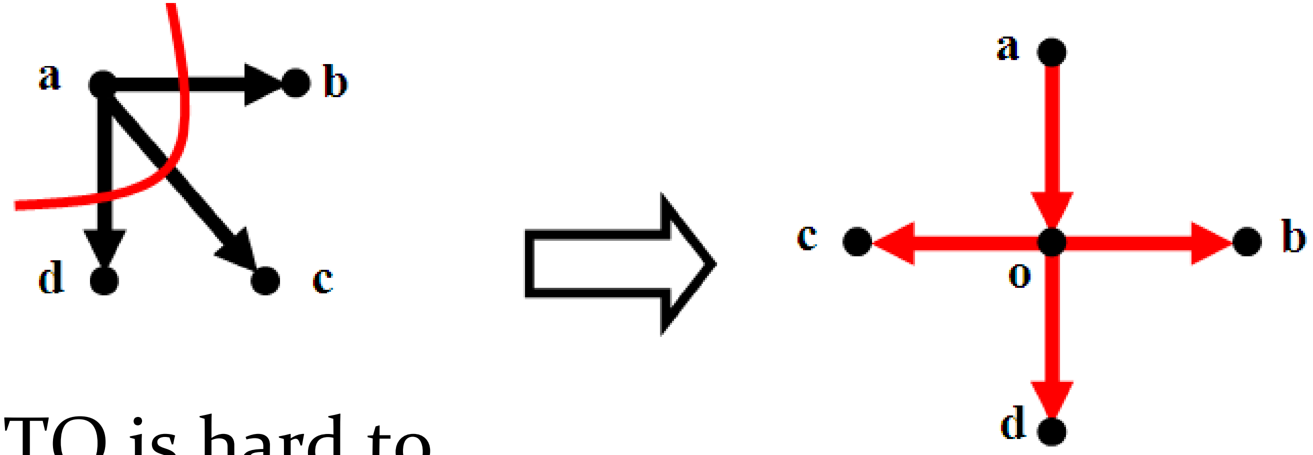
- Input:
  - An undirected tree T
  - A (multi-)set of ordered vertex pairs P
- Output:
  - An orientation of T that maximizes the number of satisfied pairs in P

Theoretical Results

Medvedovsky et al., WABI 2008
Gamzu et al., WABI 2010
Elberfeld et al., Internet Math. 2011
Elberfeld et al., TCS 2013

Complexity of MTO

- Reduction from MAX DI-CUT
- Given a directed graph *G=(V,E)*, create a star graph *G'* and a set of pairs *P*:



- MTO is hard to approximate to within 12/13

Pairs: (a,b)
(a,c)
(a,d)

A lower bound on Stars

- Choose directions uniformly at random.
- Each pair is satisfied with probability ¼
- In expectation, ¼ **of the pairs** can be satisfied.

General Trees

- *MTO(T, P):*
  - Find a node *v*, which breaks *T* into subtrees T_i of size≤*n/2*
  - *A=StarMTO(T,P,v)*
  - $B = \Sigma_i\, MTO(T_i, P)$
  - Return max*{A,B}*

Can satisfy ¼ pairs separated by v

- Thm: Fraction of satisfied pairs ≥ *1/(4* lg*n)*. This result is optimal up to a constant factor.
- Ideas can be extended to yield an *Ω(loglog n/log n)* approximation.

ILP-based solutions

Medvedovsky et al., WABI 2008
Silverbush et al., JCB 2011

An Integer Programming Formulation

- Assign a single direction for each edge

  $O(v,w) + O(w,v) = 1$

- Describe reachability relations

  $c(s,t) \leq O(x,y)$ for all edges in the path from s to t

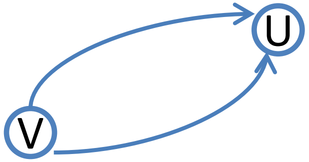
- Objective: max $\sum c(s,t)$

A biological complication

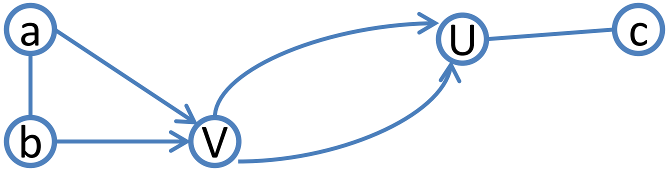
- In reality, some of the edges are pre-directed, e.g. kinase-substrate interactions.
- Can we deal with mixed graphs?
- On the theoretical side, large gap between upper (7/8) and lower ($\tilde{\Omega}(1/n^{1/\sqrt{2}})$) approximation bounds.

Mixed vs. undirected

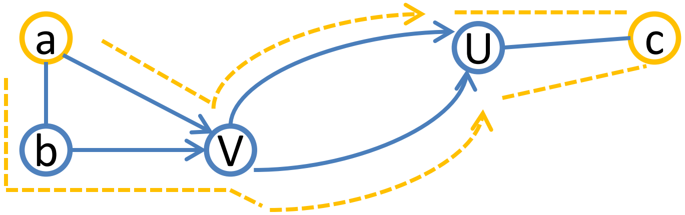
In the mixed graph there are cycles which cannot be contracted



The graph cannot be reduced to a tree

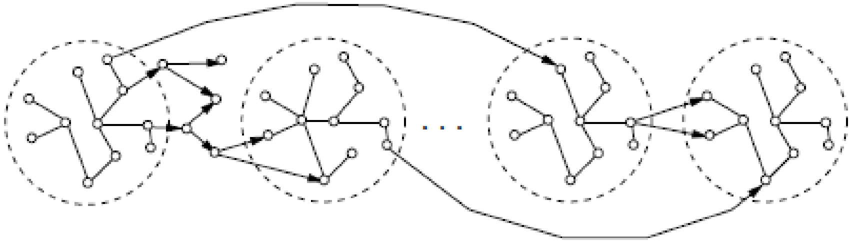


There may be multiple paths between a pair of vertices



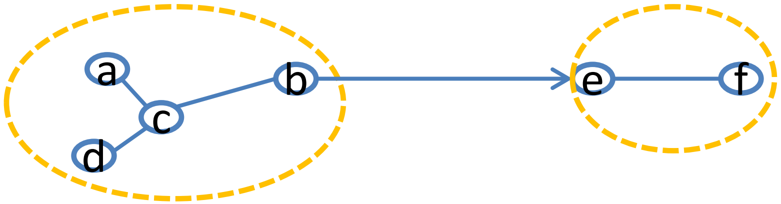
A reduction to an acyclic graph

- Contract all cycles, obtaining an acyclic graph
- Use topological sorting to create a graph of trees connected by left-to-right directed edges:



- Work recursively on pairs crossing from $G_i = T_1 \cup \ldots \cup T_i$ to T_{i+1}

Build the ILP



- <u>Between trees:</u> use path variables for every directed edge (v',w') from G_i to T_{i+1}

$$c(v,w) \leq \sum p(v,v',w',w)$$

$$p(v,v',w',w) \leq c(v,v'),c(w,w')$$

$$c(a,f) = p\,(\overbrace{a,\ \rlap{$\underbrace{}_{\text{between trees}}$}b}^{\text{inside trees}},\ \overbrace{e,\ f}^{})$$

$$p\,(a,\ b,\ e,\ f) \leq c(a,b)$$

$$p\,(a,\ b,\ e,\ f) \leq c(e,f)$$

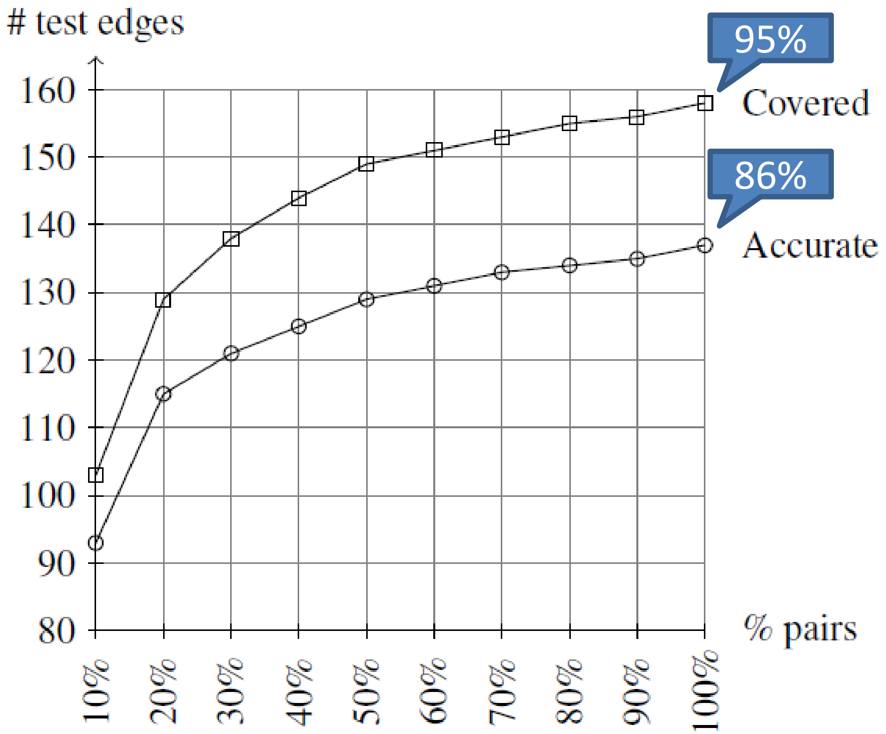
Confidence computation

- The ILP may have many optimal solutions satisfying OPT pairs.
- To evaluate our confidence in a given direction assignment u→v we rerun the ILP while forcing the opposite direction.
- Confidence(u→v) = OPT – ILP(v→u)

A taste of the results

- Applied to yeast data: ~50K pairs, ~8,000 interactions (mixed) and 1361 test edges (KPIs) whose directions are hidden from the algorithm.

- After cycle contraction:
  - ~2,000 edges
  - 166 test edges



95%

86%

- Coverage: % oriented with confidence>0
- Accuracy: % correct (confident) orientations

Increasing coverage

- Most edges (~90% in yeast) are eliminated by the cycle contraction phase, hence their directions remain ambiguous.
- One “biologically-meaningful” attack is to limit the length of the connecting paths.
- Supported by known pathways (avg. length 5)

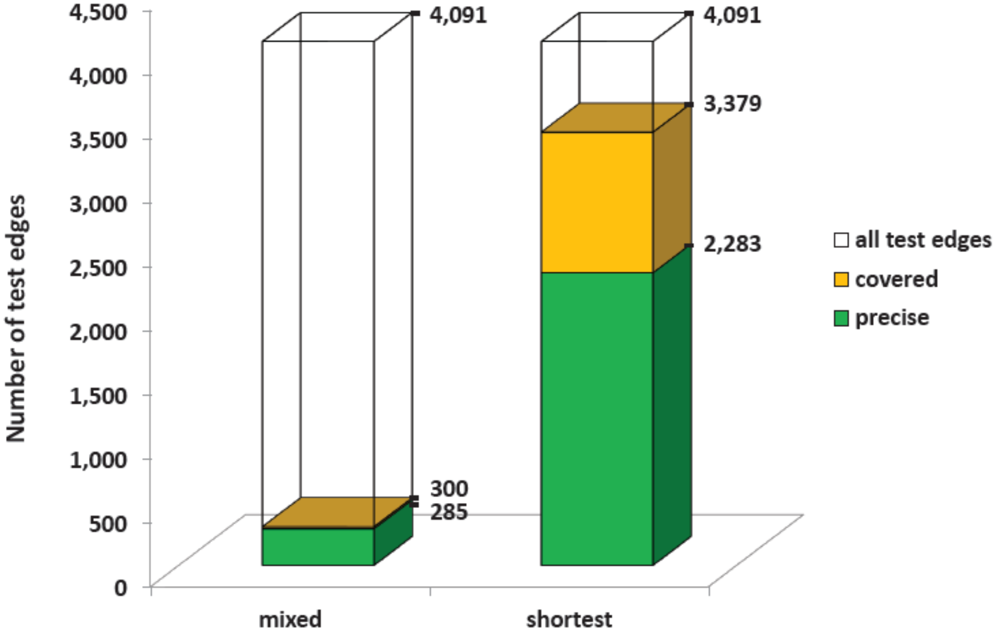
The SHORTEST approach

- A pair is satisfied iff it admits a “shortest” connecting path
- The resulting problem can be approximated to within $\Omega(1/\max\{n,k\}^{1/\sqrt{2}})$ (sublinear upper bound)
- We design an efficient ILP based on:
  - All s-t shortest-paths can be efficiently represented as a directed graph
  - Flow computations in this graph allow checking if s and t are connected (via a shortest path) under a given orientation

Blokh et al., CPM’12

Silverbush et al., Bioinformatics’14

The SHORTEST approach (application)

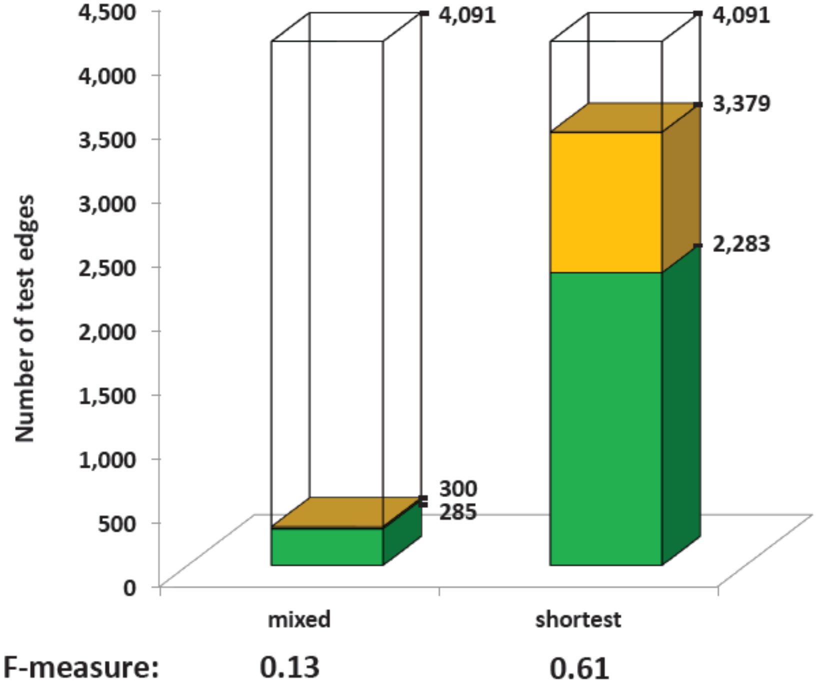


F-measure: 0.13 0.61

- Yeast: similar accuracy, 8-fold more coverage!

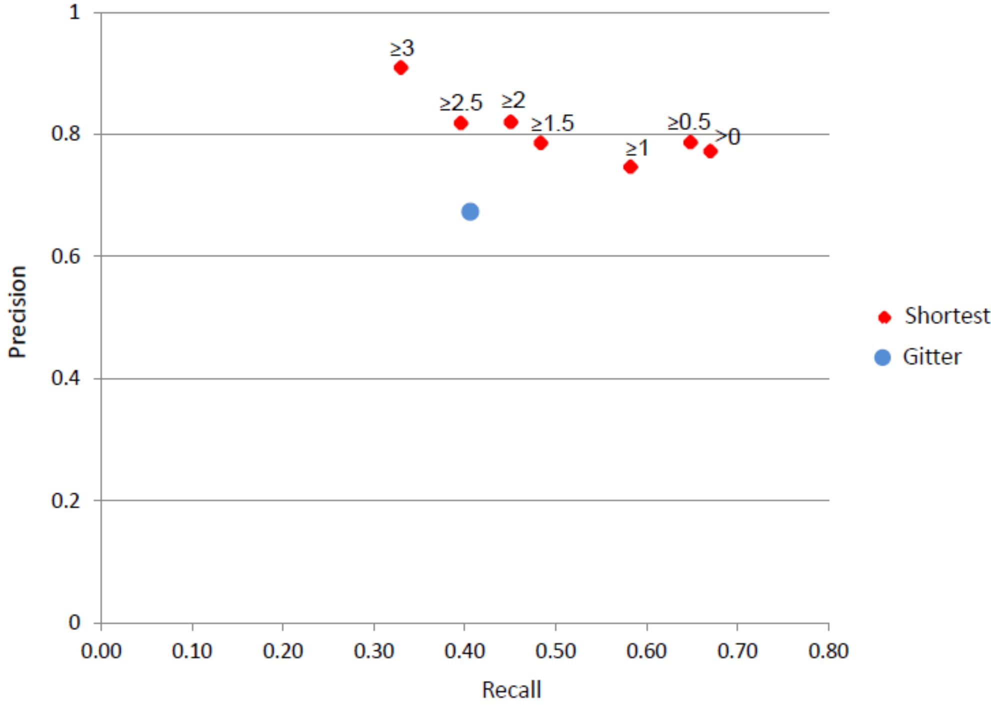
Silverbush et al., Bioinformatics'14

The SHORTEST approach (application)



- Yeast: similar accuracy, 8-fold more coverage!

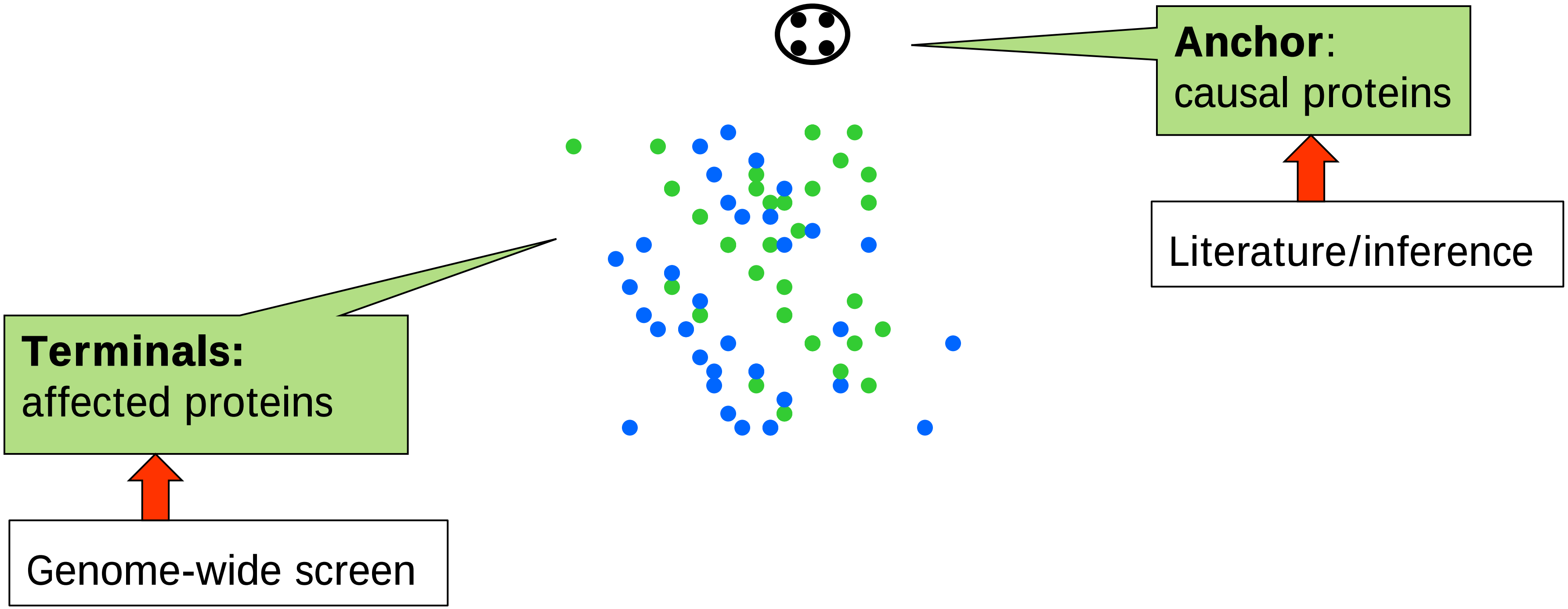
- Human: outperforms a previous method by Gitter et al.



Silverbush et al., Bioinformatics’14

Subnetwork inference

Identifying process-specific proteins



Anchor:
causal proteins

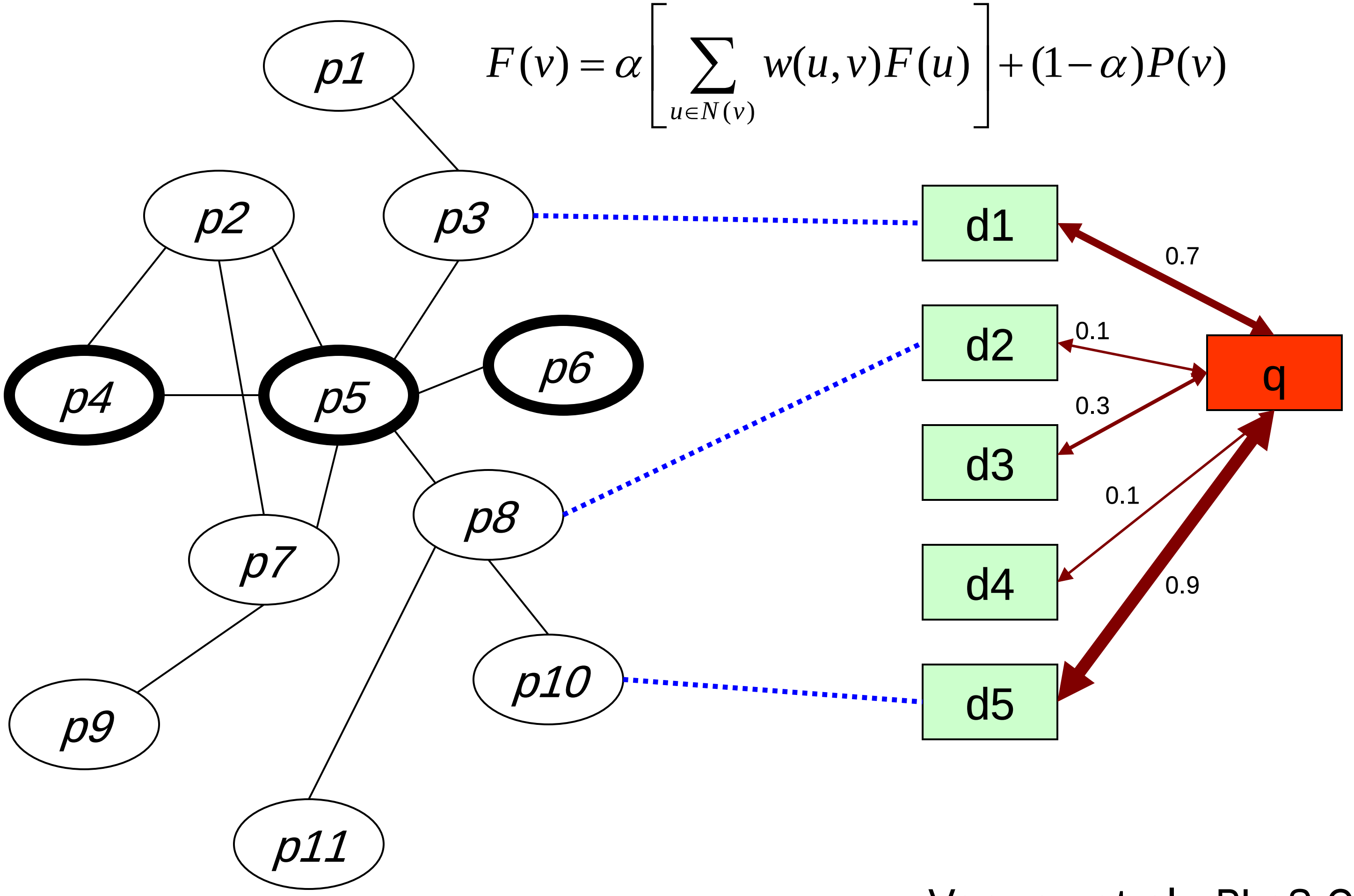
Literature/inference

Terminals:
affected proteins

Genome-wide screen

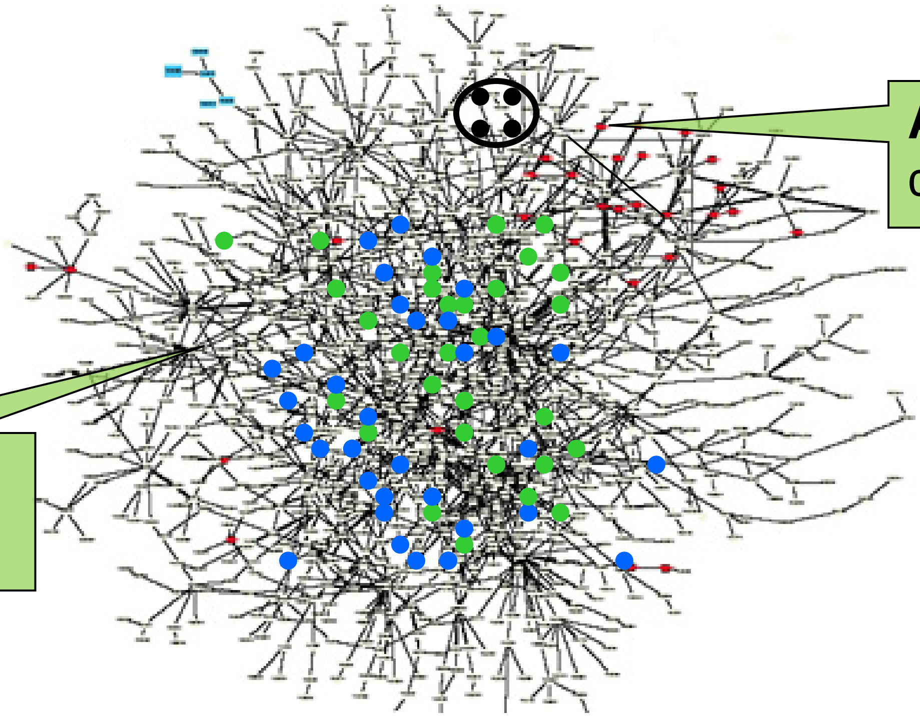
PRINCE: anchor prediction via network propagation

$$F(v) = \alpha \left[\sum_{u \in N(v)} w(u,v) F(u) \right] + (1-\alpha) P(v)$$



Vanunu et al., PLoS CB’10
Magger et al., PLoS CB’12

From components to a map



Anchor:
causal proteins

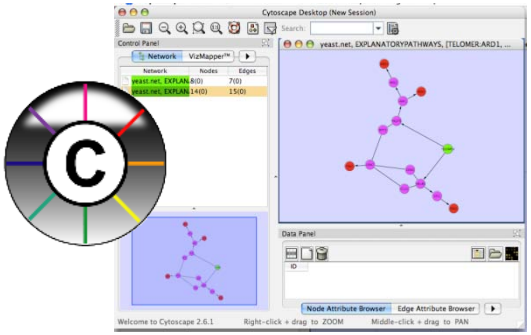
Terminals:
affected proteins

Goal: Infer the underlying subnetwork

Shachar et al., MSB 2008
Yosef et al., MSB 2009
Atias et al., MBS 2013

From components to a map (cont.)

- Unique approach to simultaneously optimize subnetwork size and length of anchor-terminal paths.
- Shown to outperform existing tools on yeast and human data
- Implemented as a cytoscape plugin called **ANAT**

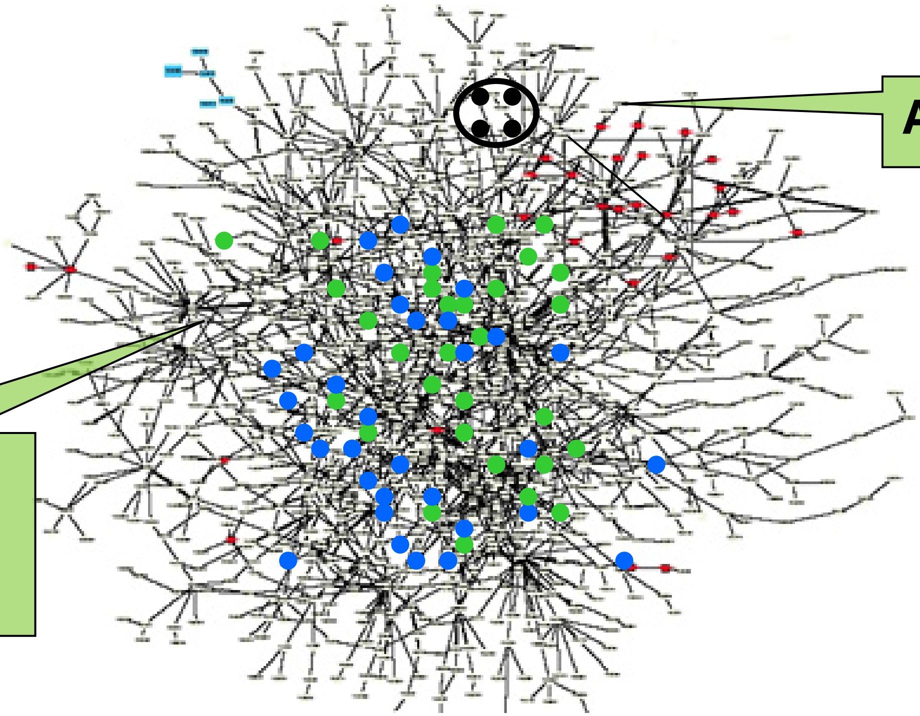


Network	Nodes	Edges
yeast.net, EXPLAN.	8(0)	7(0)
yeast.net, EXPLAN.	14(0)	15(0)

Welcome to Cytoscape 2.6.1 Right-click + drag to ZOOM Middle-click + drag to PAN

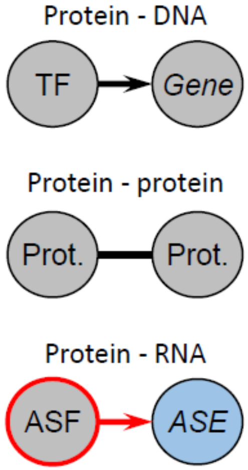
Yosef et al., Science Signaling’11
Atias et al., MBS’13

Application to alternative splicing events in cancer



Anchor: TF

Terminals: Differentially spliced events

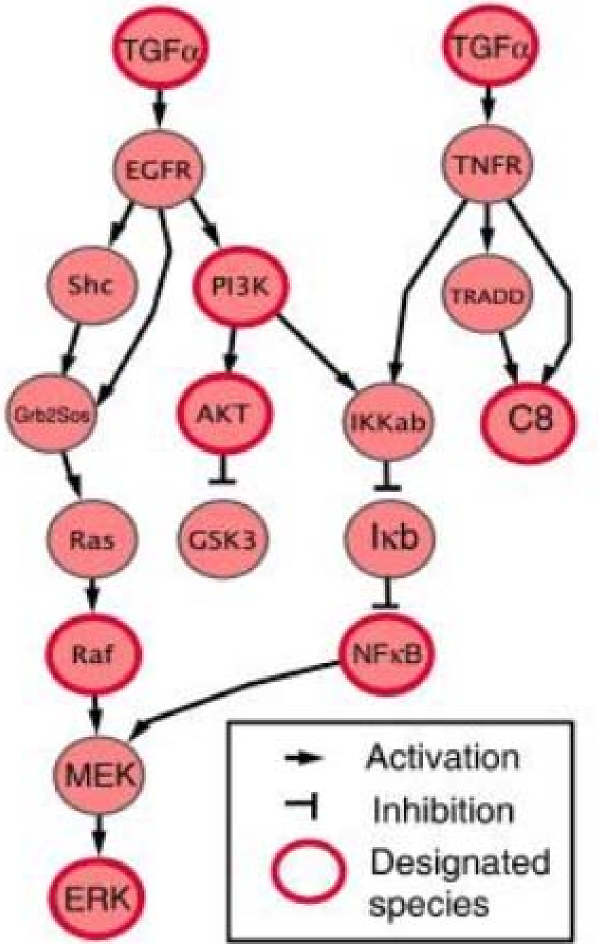


Dror Hollander, Gil Ast

Logical model learning

The Boolean model

- Each node=protein/ligand can be active (1) or inactive (0).
- The activity of a node is a *Boolean function* of the activities of its predecessors in the network.

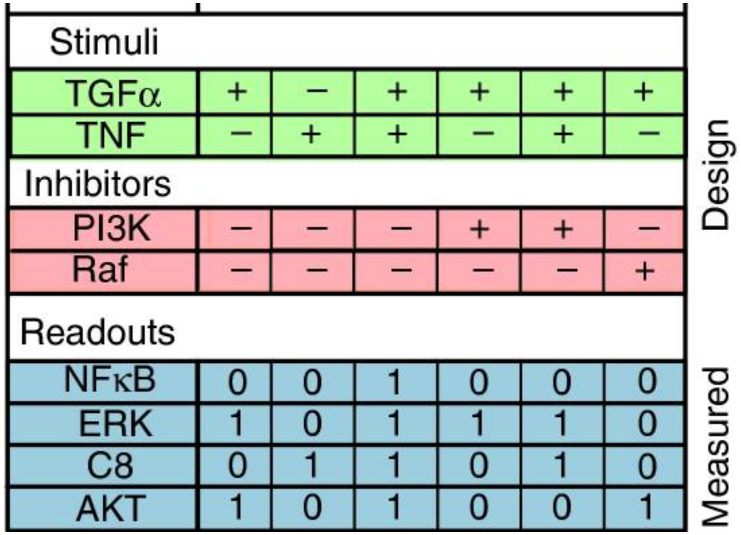


The computational problem

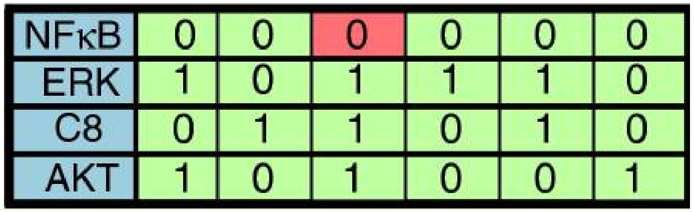
Input: (i) Directed network

(ii) Protein activity readouts following different perturbations

Goal: learn the Boolean functions so as to minimize disagreements with experimental data



Stimuli							Design
TGFα	+	−	+	+	+	+	
TNF	−	+	+	−	+	−	
Inhibitors							
PI3K	−	−	−	+	+	−	
Raf	−	−	−	−	−	+	
Readouts							Measured
NFκB	0	0	1	0	0	0	
ERK	1	0	1	1	1	0	
C8	0	1	1	0	1	0	
AKT	1	0	1	0	0	1	



NFκB	0	0	0	0	0	0
ERK	1	0	1	1	1	0
C8	0	1	1	0	1	0
AKT	1	0	1	0	0	1

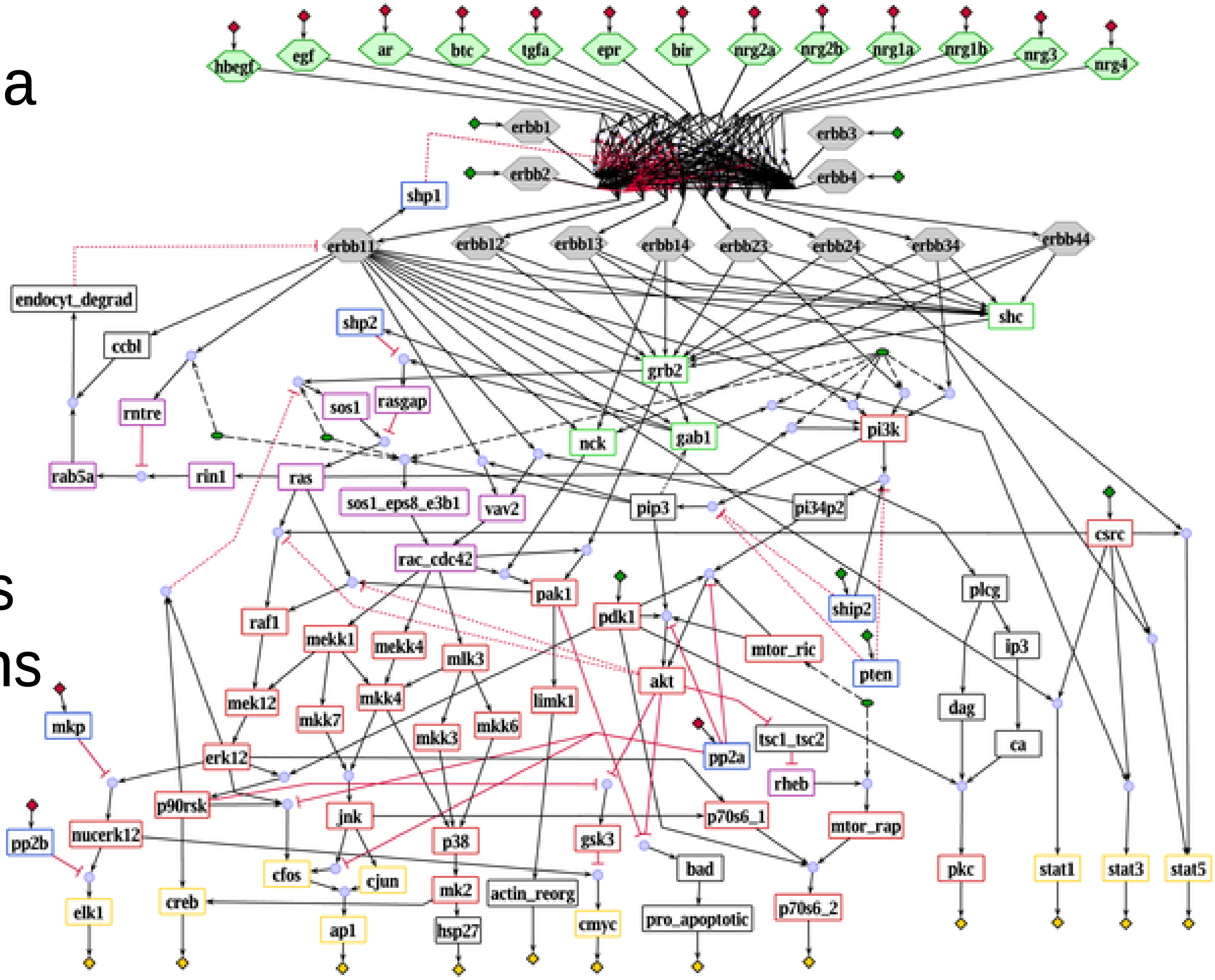
Algorithmic results

- *ILP* formulation, solved to *optimality*
- *Activation/repression effects* are automatically learned as part of the logic
- Particularly efficient solution for *threshold* functions (generalize AND & OR)

Sharan & Karp, JCB 2013

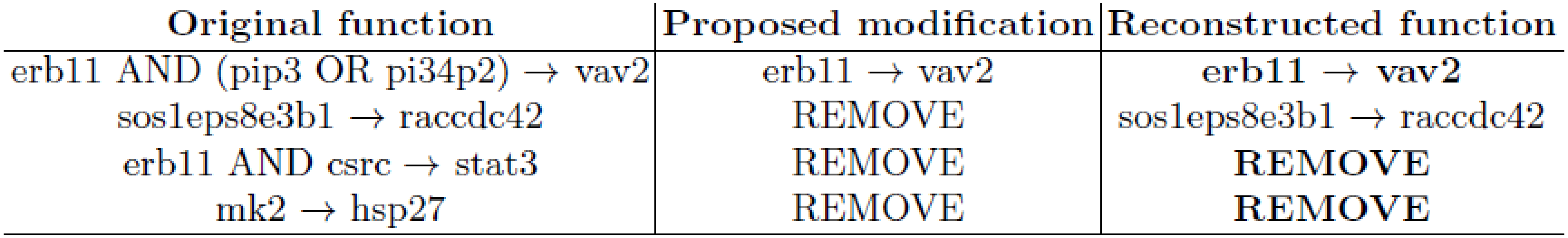
Application to EGFR signaling

- Detailed model by Oda et al. and Samaga et al. contains:
  - 112 nodes
  - 157 non-I/O reactions
- Readouts: 11 proteins under 34 perturbations
- **76%** fit to data



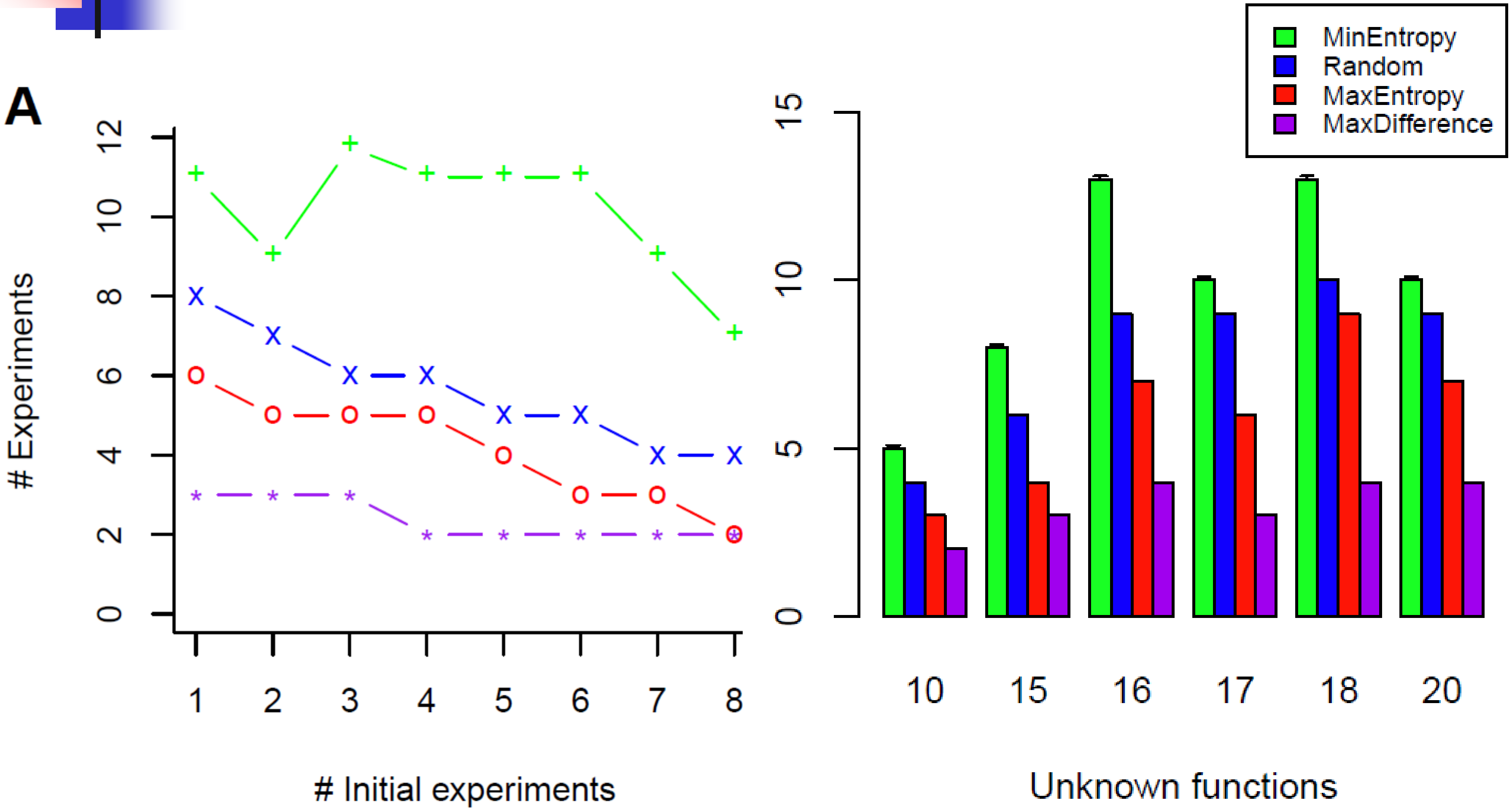
Improving the fit

- Focus on 16 uncertain gates (2^33 possible models), for 4 of which modifications were manually proposed
- 11 of 12 reconstructed functions matched the curated description
- 3 of 4 proposed changes were predicted correctly, the fourth rejected.
- The learned model achieved the same **90%** fit as the manual model!



Original function	Proposed modification	Reconstructed function
erb11 AND (pip3 OR pi34p2) → vav2	erb11 → vav2	**erb11 → vav2**
sos1eps8e3b1 → raccdc42	REMOVE	sos1eps8e3b1 → raccdc42
erb11 AND csrc → stat3	REMOVE	**REMOVE**
mk2 → hsp27	REMOVE	**REMOVE**

How many experiments are needed?



Atias et al., Bioinformatics’14 (ECCB)

Conclusions

- A framework for logic learning:

  orientation => inference => logic
- ILP-based formulations allow optimal and efficient solutions for all 3 problems
- Inference tools are available as cytoscape plugins:
  - PRINCE: www.cs.tau.ac.il/~bnet/software/PrincePlugin/
  - Propagate on the cytoscape app store
  - ANAT: www.cs.tau.ac.il/~bnet/anat/

Acknowledgments

Orientation
Dana Silverbush
Michael Elberfeld
Danny Segev…

Inference
Nir Yosef
Nir Atias
Assaf Gottlieb
Gil Ast
Dror Hollander
Martin Kupiec
Eytan Ruppin…

Logic
Richard Karp
Nir Atias…