# Dynamics and context-specificity in biological networks

**Sushmita Roy**

sroy@biostat.wisc.edu

**Computational Network Biology**

Biostatistics & Medical Informatics 826

https://compnetbiocourse.discovery.wisc.edu

Oct 16th 2018

# Topics in this section

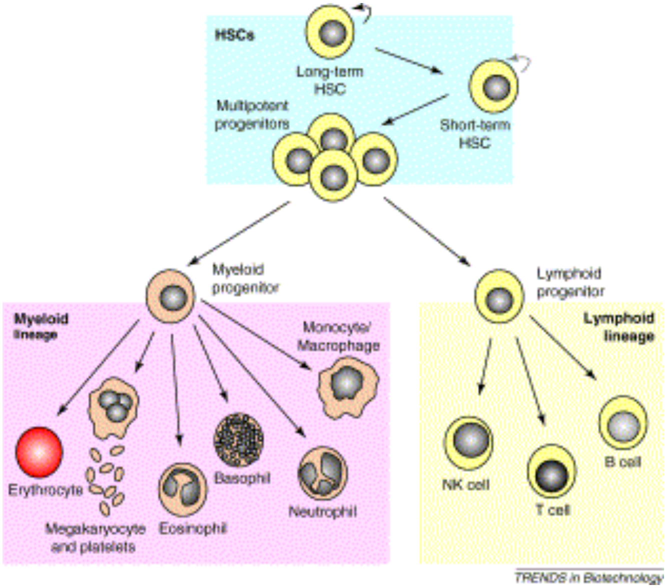
- Skeleton network-based approaches (Oct 16th, 2018)
- Input/Output Hidden Markov Models
- Multi-task learning of graphs

# Goals for today

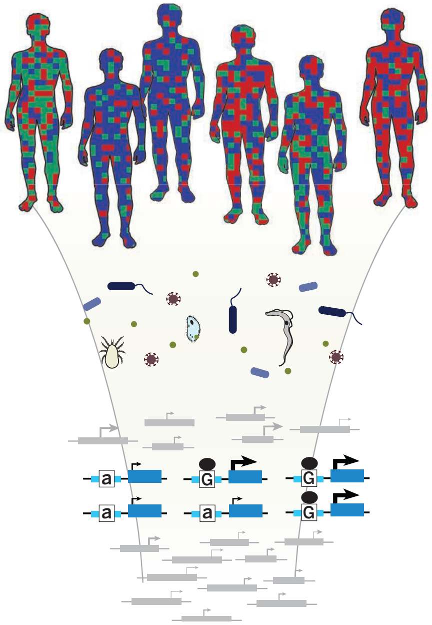
- Overview of different models to capture context-specificity/dynamics
- Skeleton network based approaches

# What do we mean by context?

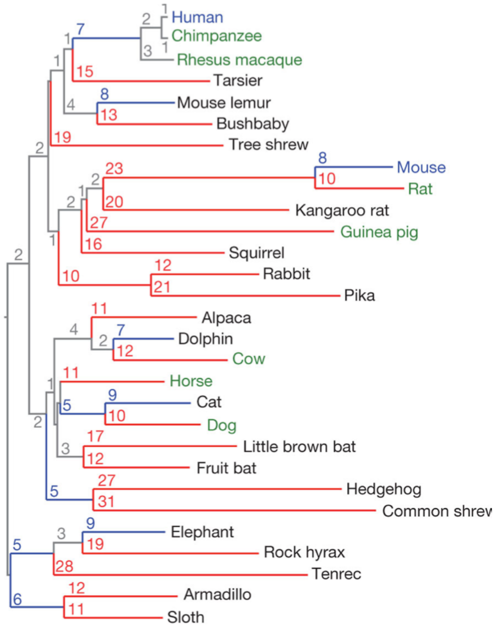
- We will define context broadly
- Context can be time, developmental stage, tissue, cell type, organ, disease, strains/individuals, species



**Different cell types**



**Different individuals**



**Different species**

# Network dynamics and context specificity

- What does modeling "dynamics" mean?
  - The activity of nodes change over time and we want to model how this happens
  - The network (structure or parameters) changes with time
    - Structure can change due to changes at the node or edge level
- What models can be used for capturing dynamics in networks?
- How do these models capture dynamics?
  - Node level
  - Edge level

# Strategies for capturing dynamics in networks

- Ordinary Differential Equation (ODE) based approaches
- Skeleton network-based approaches
- Dynamic Bayesian Networks and non-stationary extensions
- Input/Output Hidden Markov Models
- Multi-task learning approaches
- Dynamic networks with temporal transition of edges/Time varying networks

# ODE based models in networks

- Assume we are modeling a gene regulatory network
- Let $x_i$ denote the expression level of the $i^{th}$ gene
- ODE for the change in expression of $i^{th}$ gene is

Expression level of regulator $p$

$$\frac{dx_i}{dt} = -\alpha_i x_i + \sum_{p \in P_i} \beta_{i,p} x_p$$

Regulators of $i^{th}$ gene

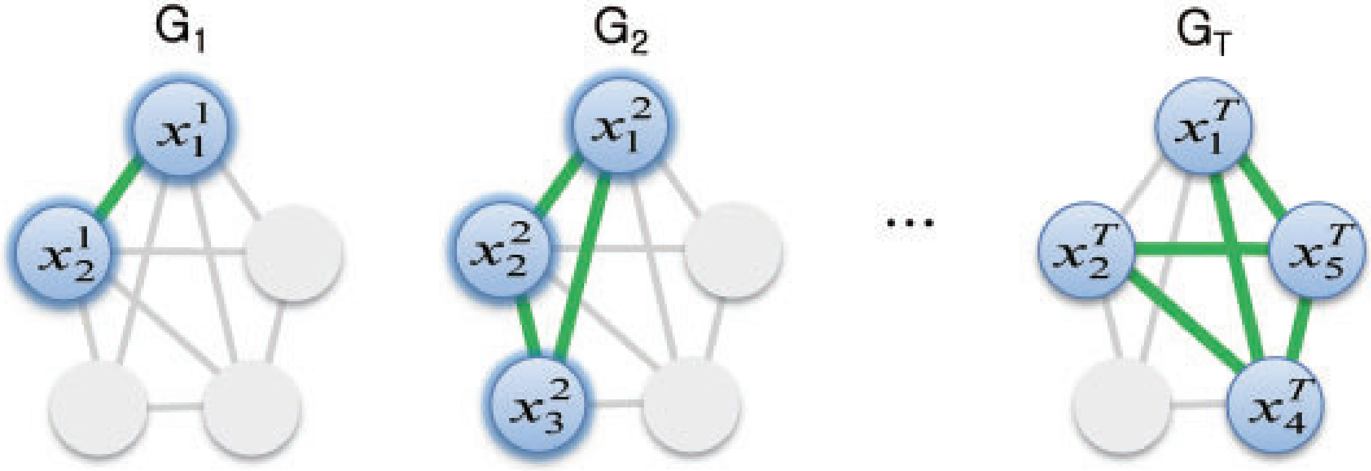
- This is often approximated by a finite difference approximation

$$\frac{x_i(t+1) - x_i(t)}{t_{k+1} - t_k} = -\alpha_i x_i(t_k) + \sum_{p \in P_i} \beta_{i,p} x_p(t_k)$$

Greenfield et al., Bioinformatics 2013

# Skeleton network-based approaches

- Assume that we have background/skeleton network that defines the universe of possible edges
- The network changes because node activity levels change



Network at time 1

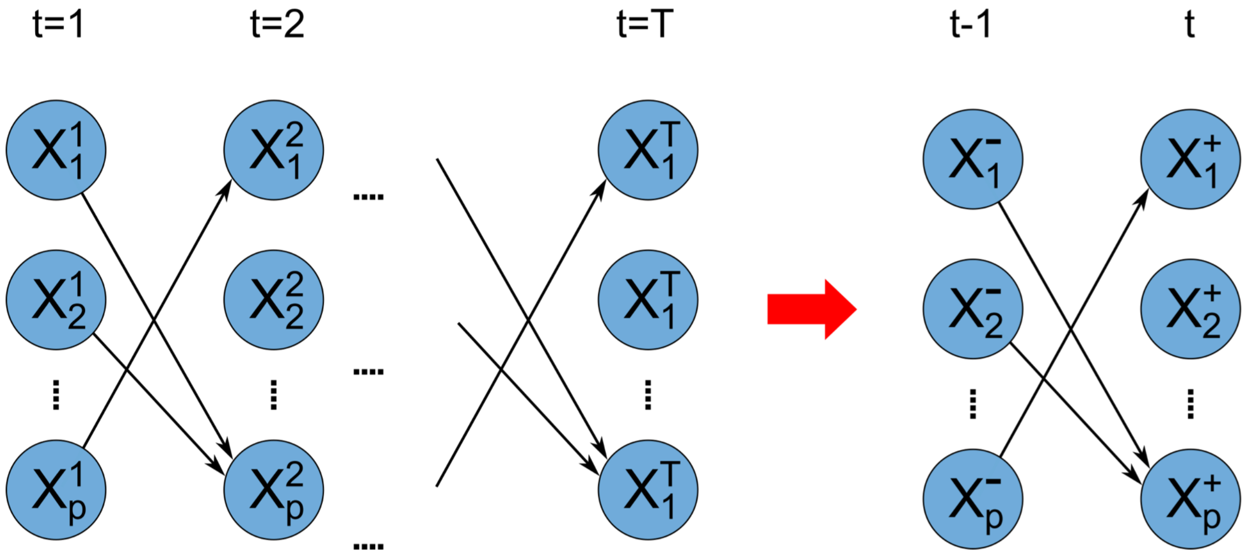
Network at time 2

Network at time T

Kim et al. 2013, Briefings in Bioinformatics

# Dynamic Bayesian networks (DBNs)

- Suppose we have a time course of node activity levels
- We assume that the activity levels at time t+1 depends upon t
  - But this does not change with t
- DBNs can be used to model how the system evolves over time
  - We may have a different network at the first time point



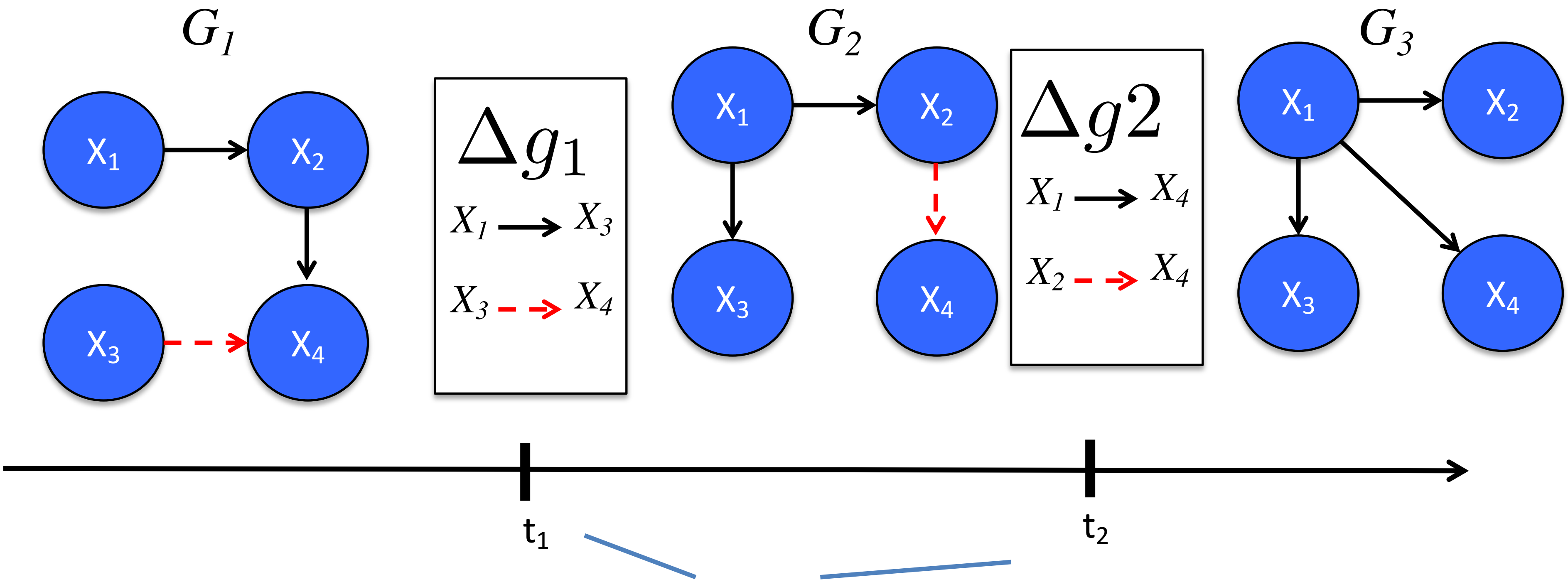
# Non-stationary Dynamic Bayesian Networks (nsDBNs)

- The standard DBN assumes that the dependency between the previous time point (*t-1*) and the current time point (t) is the same
    - This is called the stationarity assumption in DBNs
- Non-stationary DBNs are DBNs where this assumption is relaxed
    - Introduced by Robinson and Hartemink 2010
    - Depending upon the time-window/epoch we might have a different dependency structure

# Non-stationary Dynamic Bayesian Networks

- Suppose we have three time windows
- nsDBNs require us to define the dependency structure in each time window

Edge set change between time windows



Transition times

Adapted from Robinson & Hartemink 2010

# Non-stationary Dynamic Bayesian Networks continued

- For *m* epochs, we would like to find $G_1,.., G_m$ by optimizing their posterior distribution

$$P(G_1, .., G_m|D) \propto P(D|G_1, .., G_m)$$
$$P(G_1, .., G_m)$$

Prior over *m* graphs; can be used to incorporate our prior knowledge of how the graphs transition

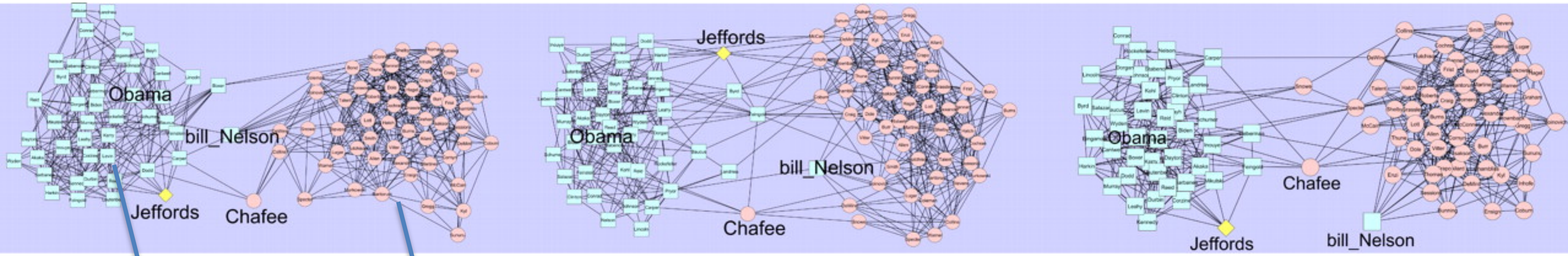
$$P(G_1, .., G_m|D) \propto P(D|G_1, .., G_m)$$
$$P(G_1, \Delta g_1, \Delta g_2, .., \Delta g_{m-1})$$

# Time-varying Networks

- The network changes between time at the node and edge levels
- Example approaches
  - TESLA (Ahmed & Xing, PNAS 2009)
  - KELLER (Song et al., Bioinformatics 2009)
- TESLA
  - Based on temporal Exponential Random Graph Models
  - Assumes binary node values
  - works by imposing a regularization term to make the graphs change smoothly over time
  - estimates the graph structure by solving a set of local regularized logistic regression problems

# Applying TESLA to US senators of the 109th congress

Epoch 2 Epoch 7 Epoch 10



Democrats

Republicans

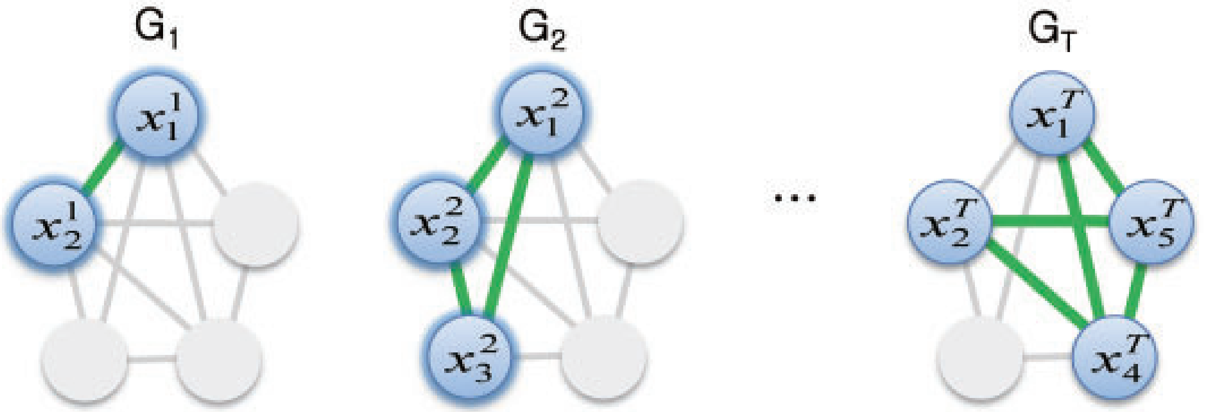
Each node is a senator. The networks were inferred using voting records of 642 bills. Votes are binary states at each time point. Voting records were grouped into 12 epochs, each epoch comprising 2 months.

# Goals for today

- Overview of different models to capture context-specificity/dynamics
- Skeleton network based approaches

# Skeleton network-based approaches

- Given
  - A fixed network of nodes (genes, proteins, metabolites)
  - Activity levels of network nodes in a set of contexts (e.g. tissues, time points)
- Do
  - Find subset of edges that are active in each context



Transitional networks

Kim et al. 2013, Briefings in Bioinformatics

# Active subnetworks: A skeleton-network based approach

- Active subnetworks: A subnetwork that exhibits an unexpectedly high level of expression change
  - "Discovering regulatory and signalling circuits in molecular interaction networks". By Trey Ideker, Owen Ozier, Benno Schwikowski, Andrew Siegel. Bioinformatics. 2002;18 Suppl 1:S233-40.
- Enables us to gain a better understanding of how expression is controlled at the network level
- Enables us to link gene expression changes to higher order interactions

# Active subnetworks key idea

- Given
  - Network structure and expression data (actually P-values specifying the significance of expression of a gene)
- Find
  - Subnetwork that is differentially active
- Active network identification requires solving these tasks:
  - Define a score of subnetwork activity
    - One or multiple conditions
  - Search for high-scoring subnetworks

# Defining the score for subnetwork activity

- The score is going to be a z-score
  - The number of standard deviations from the mean.
- Need to compute a *z*-score associated with each subnetwork *A*
- Start with assessing differential expression of each gene
  - P-value for significance of expression $p_i$
- Convert P-value to gene level *z*-score $z_i$

$$z_i = \Phi^{-1}(1 - p_i)$$

Inverse normal CDF

- Aggregate *z*-scores for all genes in subnetwork *A*

$$z_A = \frac{1}{\sqrt{k}} \Sigma_{i \in A} z_i$$

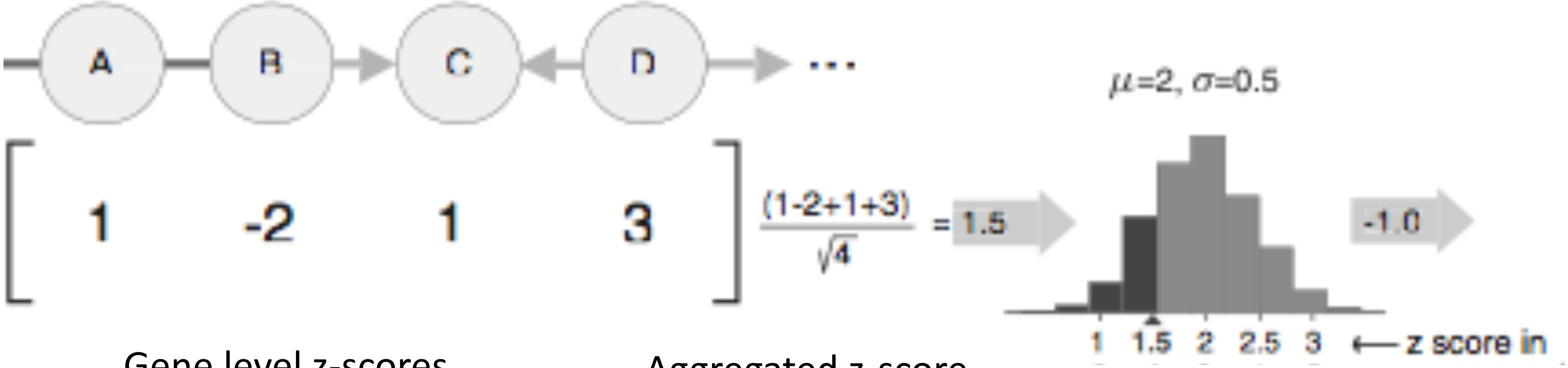
- Calibrate this against a background distribution

$$s_A = \frac{z_A - \mu_k}{\sigma_k}$$

Estimated from randomly sampling gene sets of size $k$

# Example of a score of subnetwork activity

Genes in a subnetwork



Gene level z-scores

Aggregated z-score

Calibrate against a background distribution

# Extending the scoring criteria to multiple conditions

- Suppose we had $m$ different conditions
- Given a subnetwork *A*, compute a set of *z*-scores for *A* in all $m$ conditions

$$(z_{A(1)}, z_{A(2)}, \cdots, z_{A(m)})$$

- Score of a subnetwork for multiple conditions is proportional to the probability that the subnetwork is active in at least $j$ of the $m$ conditions
- $\phi$ is the CDF of the normal distribution
- The probability of subgraph to have a score of at most $Z_{A(j),}$

$$\phi(Z_{A(j)}) = P(s_A \leq Z_{A(j)})$$

# Extending the scoring criteria to multiple conditions

- Hence, the probability that any single condition has score at least $Z_{A(j)}$

$$P_z = 1 - \phi(Z_{A(j)})$$

- Probability that at least *j* of $m$ conditions has score at least $Z_{A(j)}$

$$p_{A(j)} = \sum_{h=j}^{m} \binom{m}{h} P_z^h (1 - P_z)^{m-h}$$

- This is equivalent to a p-value for $z_{A(j)}$

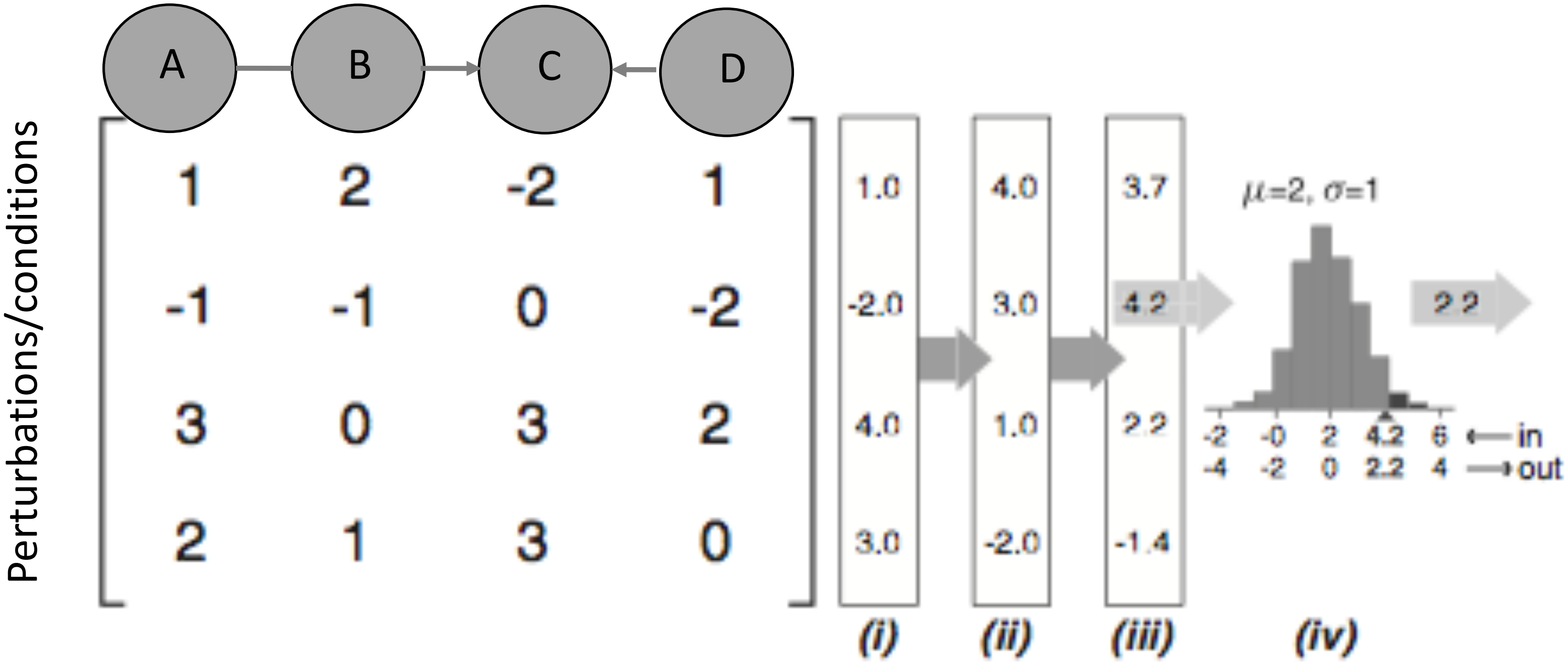
$$r_{A(j)} = \phi^{-1}(1 - p_{A(j)})$$

Score associated with the $j^{th}$ subset of conditions

$$r_A^{\max} = \max_j (r_{A(j)})$$

Score associated with best subset

# Overview of scoring criteria for multiple conditions



$$\begin{array}{c|cccc} & A & B & C & D \\ \hline & 1 & 2 & -2 & 1 \\ \text{Perturbations/conditions} & -1 & -1 & 0 & -2 \\ & 3 & 0 & 3 & 2 \\ & 2 & 1 & 3 & 0 \end{array}$$

(i) Score for each condition

(ii) Sorted scores

(iii) Z-score calculation for the $j^{th}$ subset

# Active subnetwork finding algorithm

- Given a subnetwork, we now have a way to score its activity in one or more conditions
- How do we find such a subnetwork?
- We need to search in the space of subnetworks, score each subnetwork and select the most active ones

# Simulated annealing to find active subnetworks

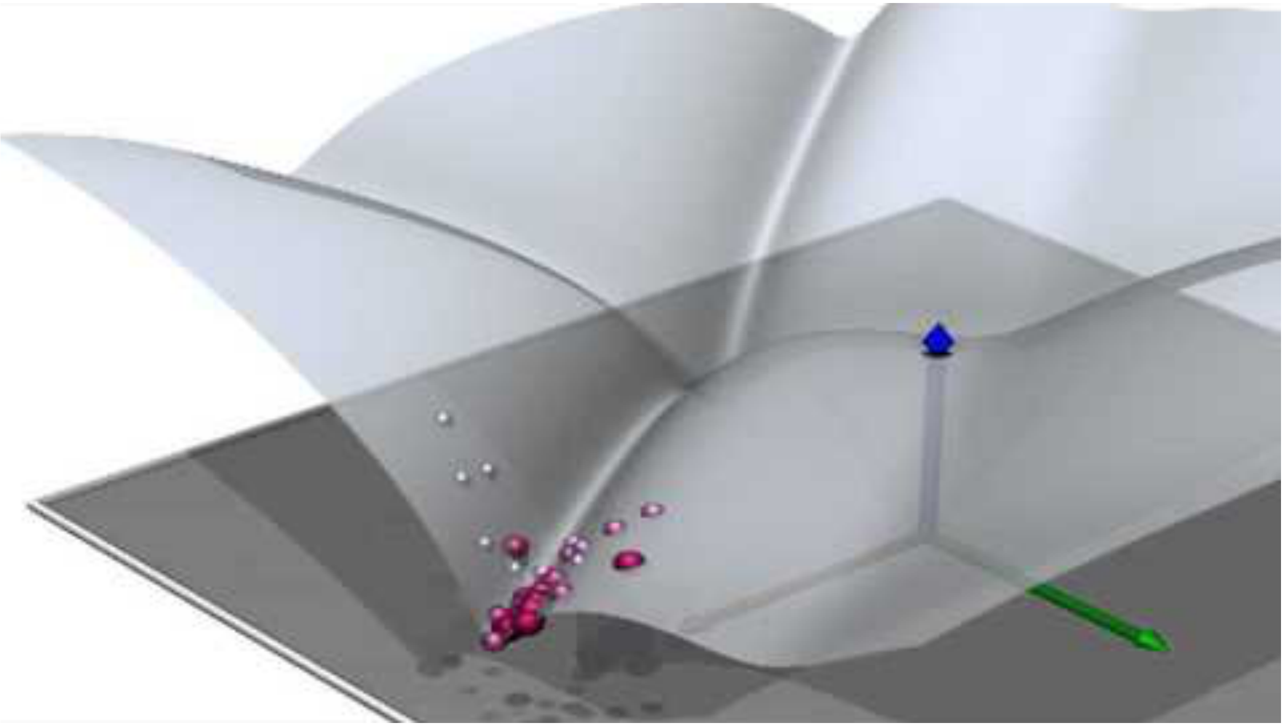
- Finding the maximal scoring subnetwork is computationally intractable
- Apply simulated annealing to find a close approximation to the global optimum
- Simulated annealing
  - A heuristic search approach to find  a good approximation to the global optimum
  - Enables one to take non-optimal steps to avoid being too greedy

# Simulated annealing

- Requires a “Temperature” schedule that starts from a high value and gradually “cools” down
  - At high temperatures (beginning), it will accept more non-optimal moves
  - At low temperatures, the probability of accepting non-optimal moves is smaller
- Let e and e’ be the scores of the current and new moves
- Probability of accepting e’ is

$$P(e, e', T_i) = \begin{cases} 1, \text{if } e < e' \\ \exp(\frac{e-e'}{T_i}) \end{cases}$$

# Simulated Annealing example



https://www.youtube.com/watch?v=iaq_Fpr4KZc

# Algorithm sketch to find a high scoring subnetwork

- **Input**: Graph $G=(V,E)$, Number of iterations $N$, Temperature function $T_i$ which decreases geometrically from $T_{start}$ to $T_{end}$
- **Output**: A subgraph $G_w$ of $G$

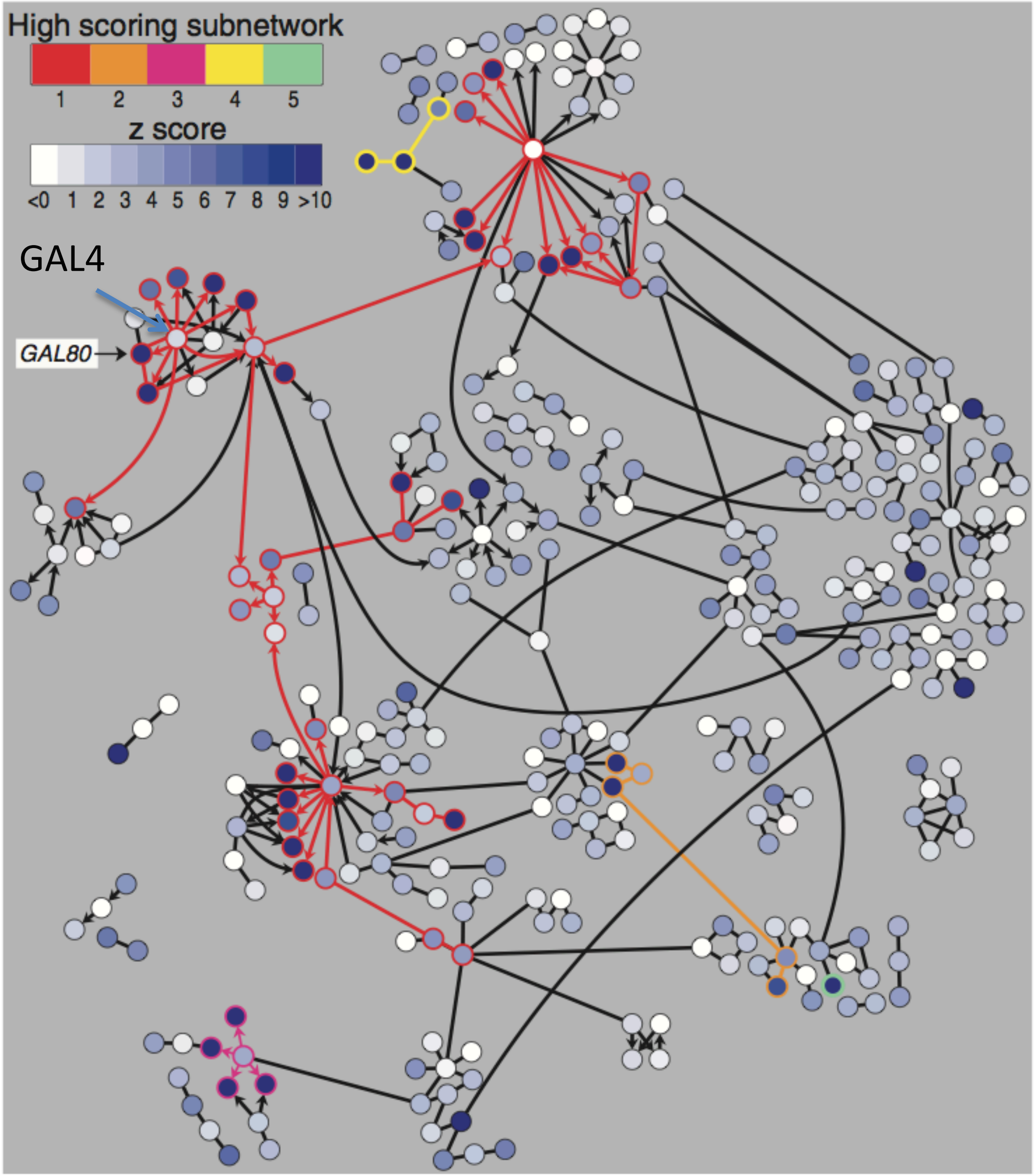
- Initialize $G_w$ (working subgraph) by randomly setting each $v$ to active/inactive and obtaining the induced subgraph
- Loop $i=1$ $to$ $N$
  - Randomly pick a node $v$ in V and toggle its state (new $G_w$)
  - Compute score for new subgraph $s_i$
  - If $s_i>s_{i-1}$ keep $v$ toggled
  - Else keep $v$ toggled with probability $p = e^{(s_i - s_{i-1})/T_i}$
  - Output $G_w$

# Additional heuristics

- Additional heuristics: Search for multiple (M) subnetworks simultaneously
- $d_{min}$ (degree): remove low scoring neighbors for nodes with degree greater than $d_{min}$

# Results

- Application to two networks
  - 362 protein-protein and protein-dna interactions in a yeast galactose utilization study
  - 7145 protein-protein and 317 protein-dna interactions
- Each condition was a single or double knockout of a gene in the galactose pathway
- Examine subnetworks in one condition (GAL80) (Figure 1)
- Examine subnetworks in multiple conditions (Figure 5)

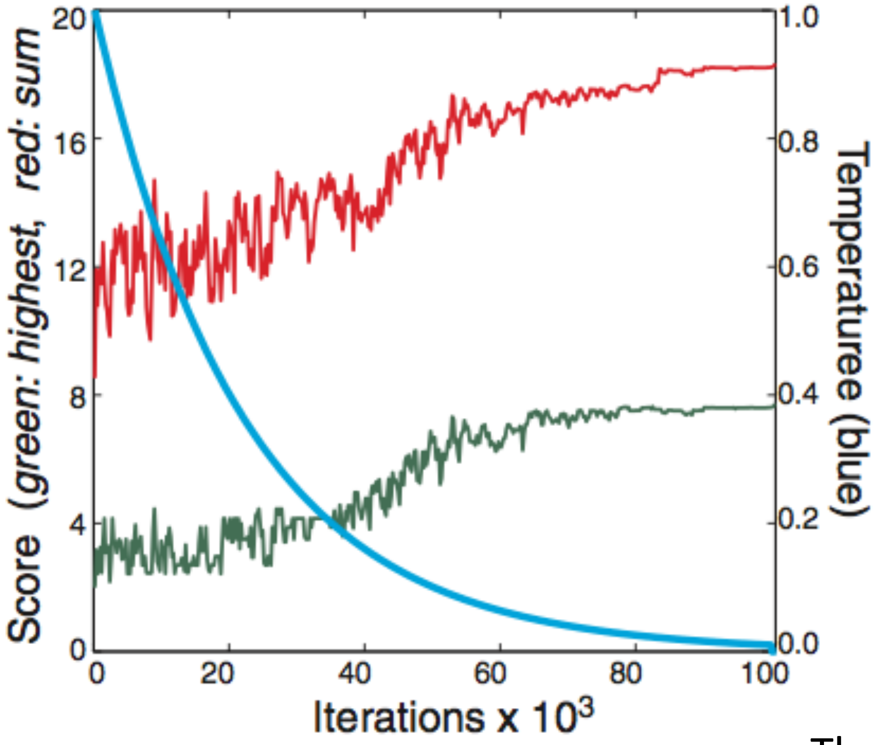


# Fig 1. Subnetworks identified in the 362 PPI, PDI using GAL80 knock out based z-score

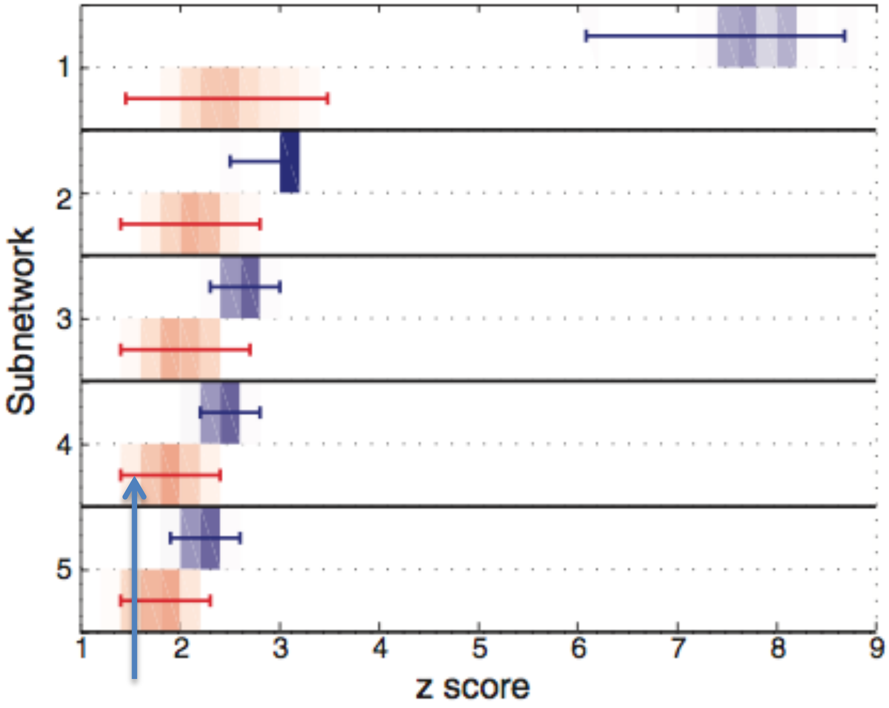
Transcription factors are often not very differentially expressed

# Analysis on small network with one perturbation (one condition)

Simulated annealing score convergence

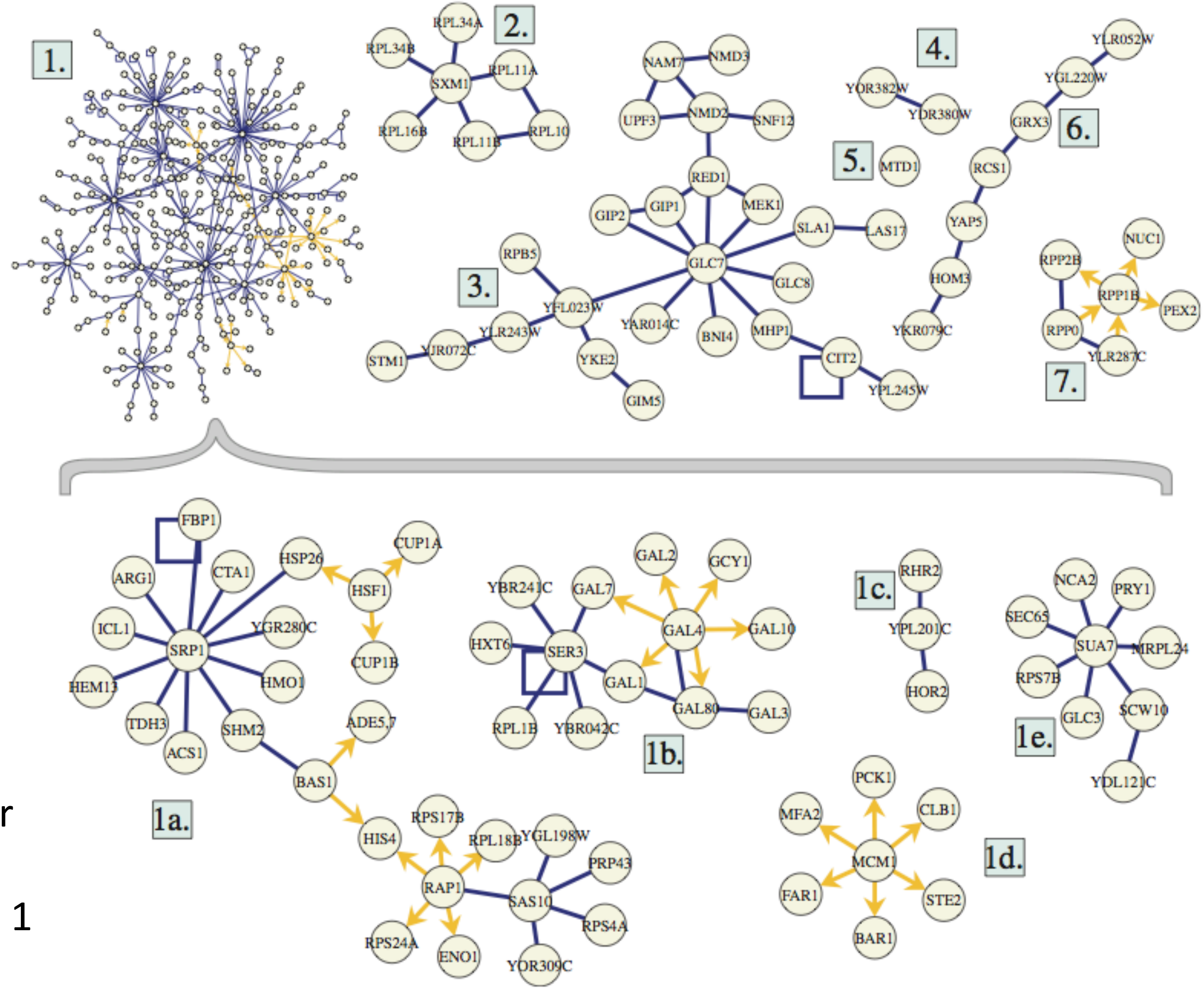


Distribution of true and random subnetwork scores



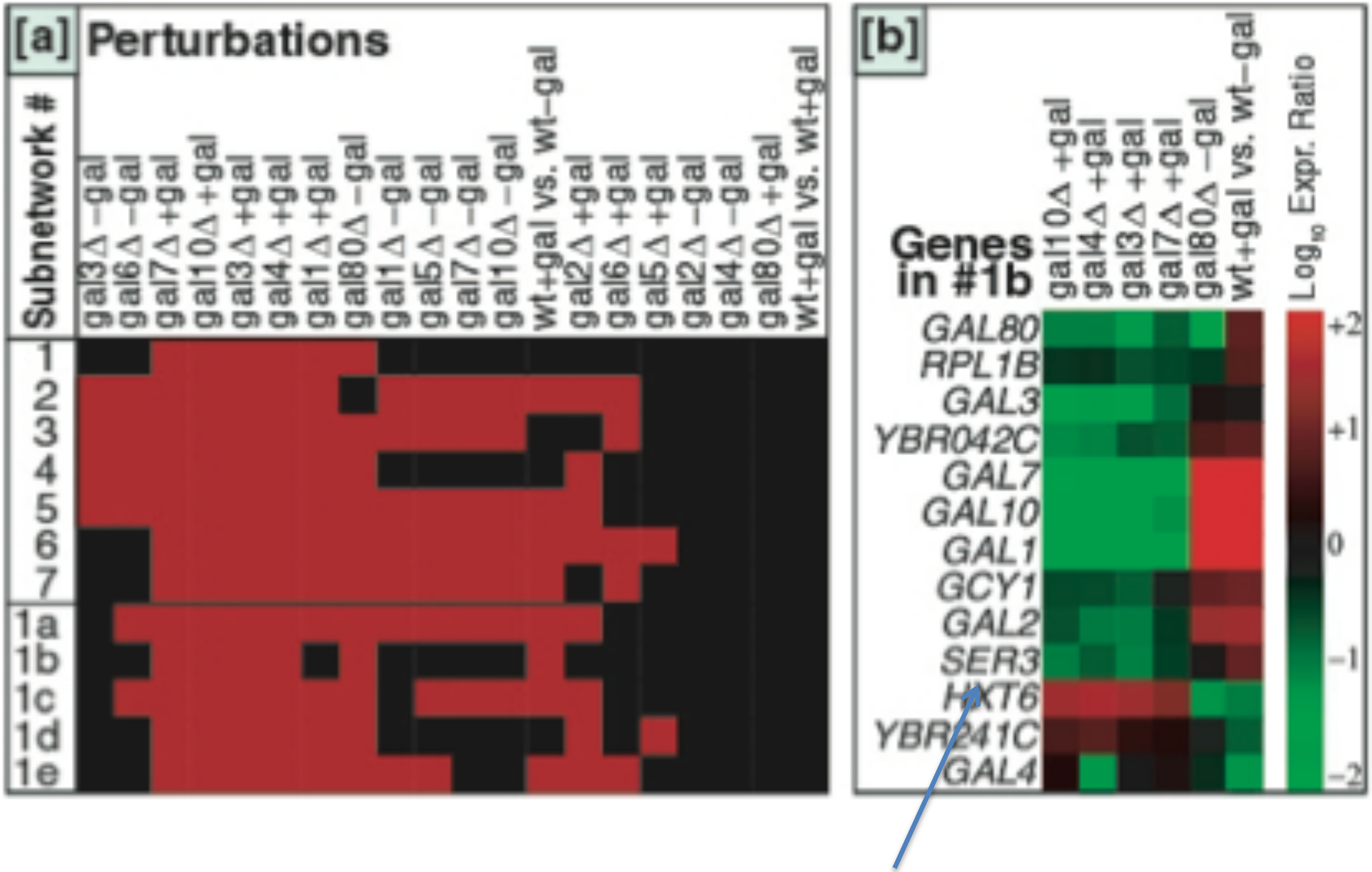
These red scores correspond to randomized expression data

# Fig 5: Analysis on a larger network with multiple conditions



Result of applying another round of Simulated Annealing on subnetwork 1

# Fig 6: Conditions in which specific subnetworks were active



Genes in the same subnetwork may be inversely correlated to each other

# Take-away points

- A skeleton network based approach
  - Fixed network, but the node activity can change across conditions
- Key advantages over simple expression-based clustering
  - Find weakly expressed genes that are connected to genes that are highly expressed
  - More interpretative
  - Clustering may not connect two genes of opposite expression profiles
  - Can find subnetworks active for a subset of conditions
- Future extensions
  - Correcting for network topology
  - Inferring more attributes on the subnetworks
  - Better/different ways to find a subnetwork