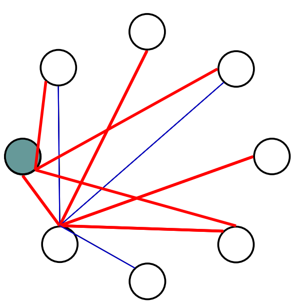
School of Computer Science
Carnegie Mellon

# Probabilistic Graphical Models

## Gaussian graphical models and Ising models: modeling networks

Eric Xing

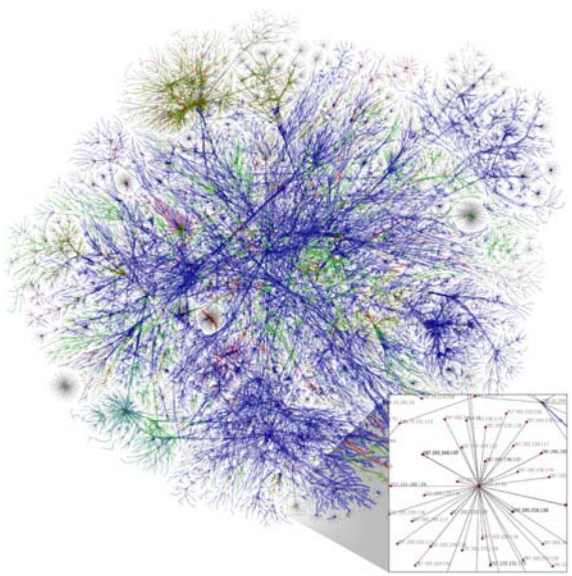
Lecture 10, February 16, 2015



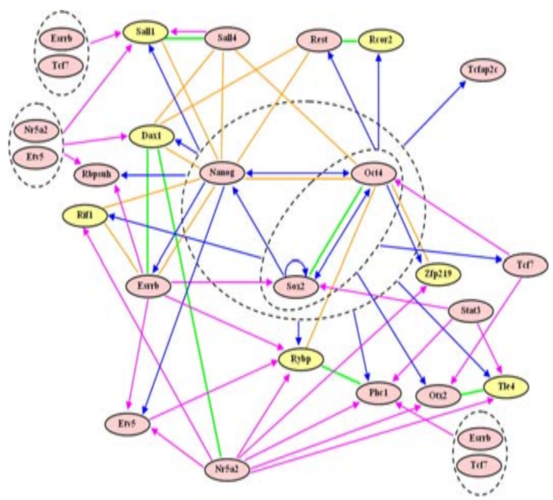
**Reading:** See class website

# Network Research

Social Network



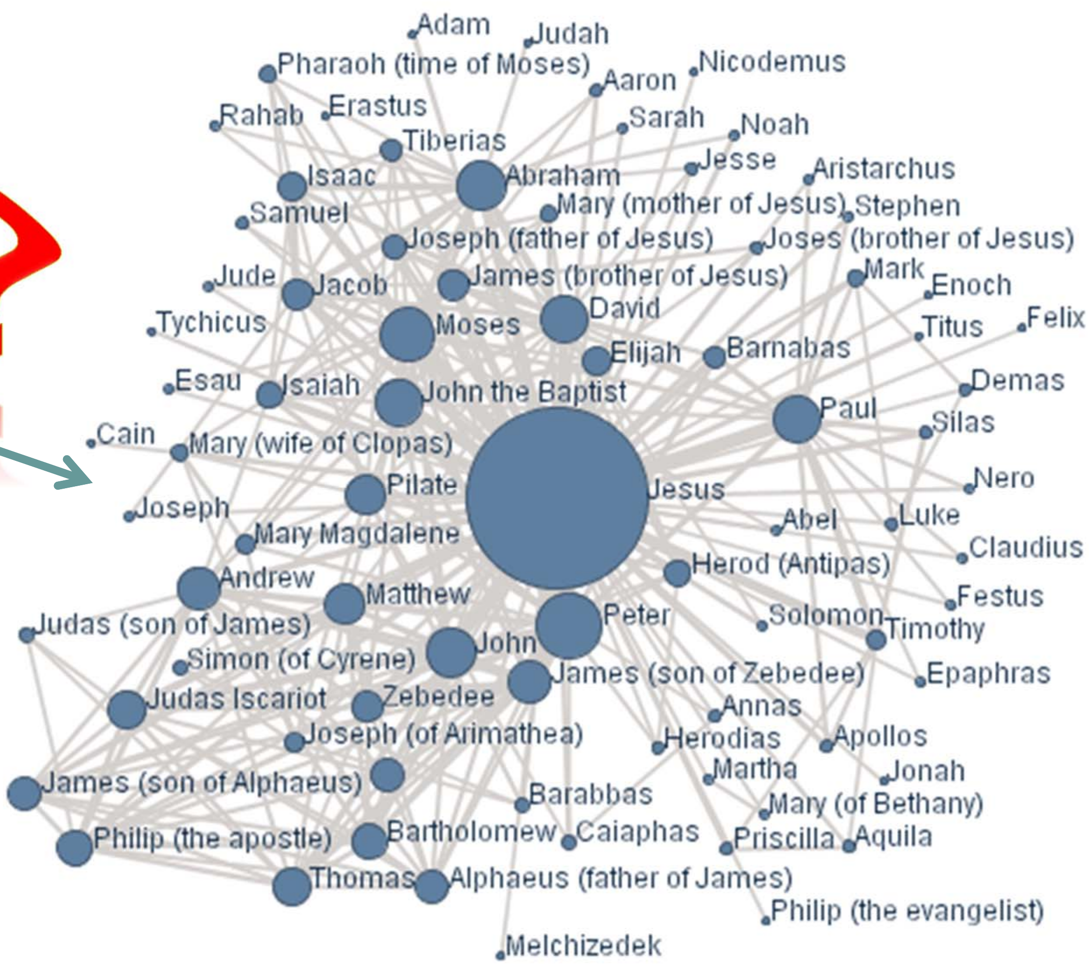
Internet



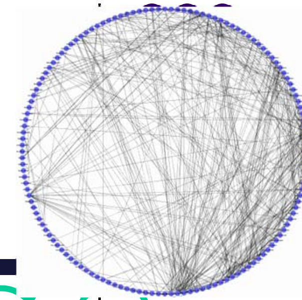
Regulatory Network

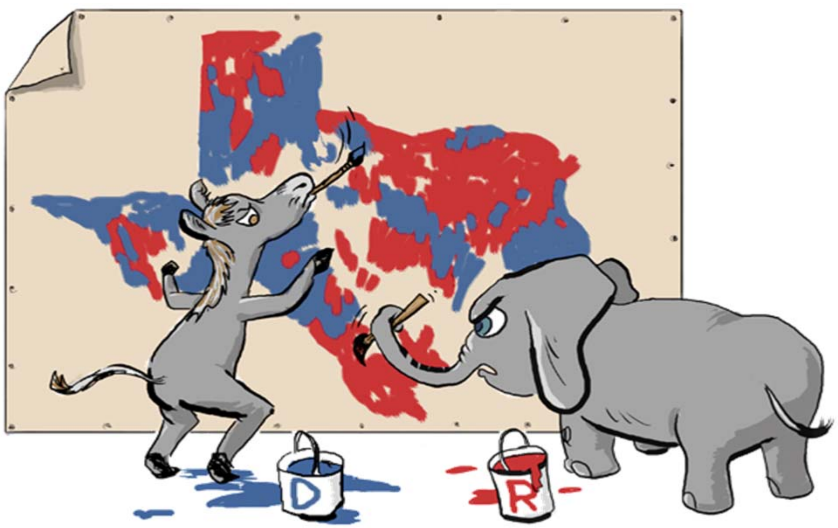
# Where do networks come from?

- **The Jesus network**



# Evolving networks






**Can I get his vote?**

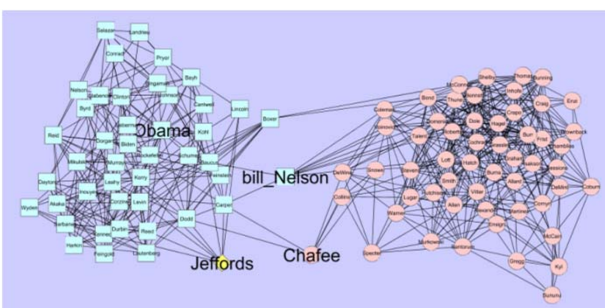
Corporativity,

Antagonism,

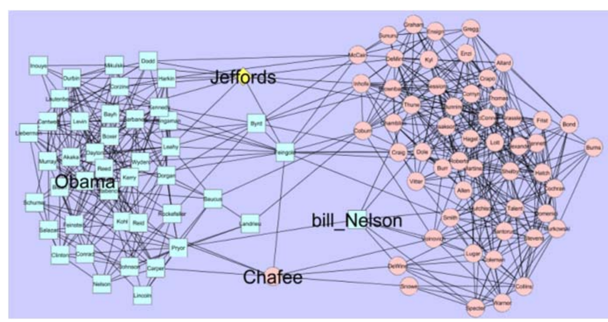
Cliques,

…

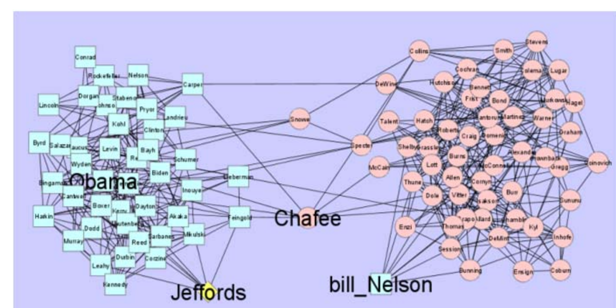
over time?




March 2005

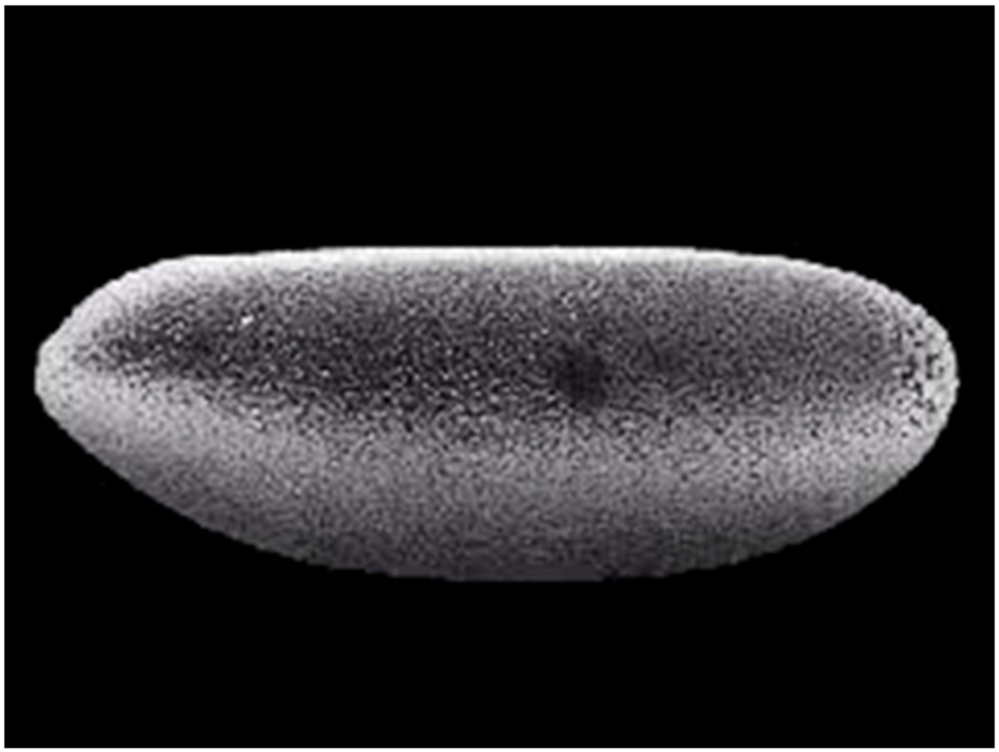


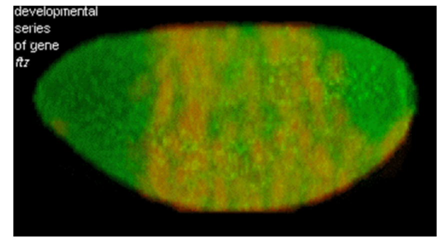

January 2006

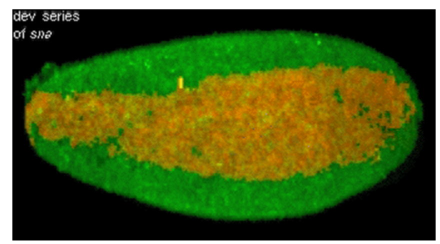


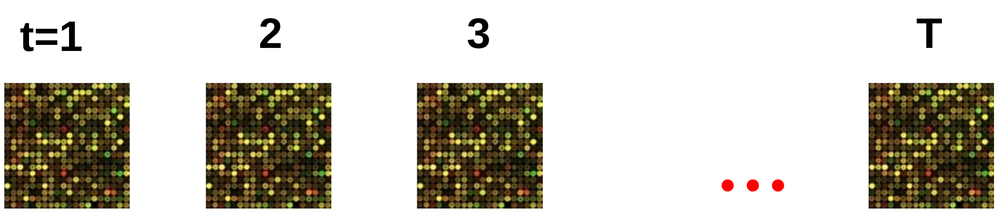

August 2006

# Evolving networks

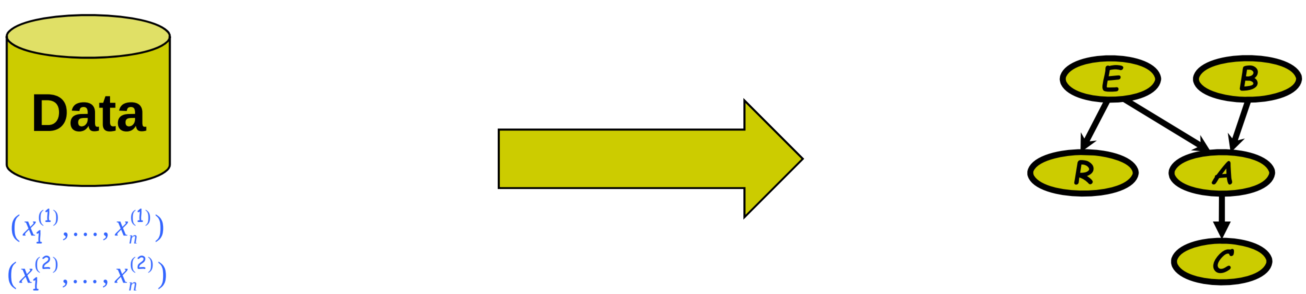




# Recall: ML Structural Learning for completely observed GMs




$$(x_1^{(1)},\dots,x_n^{(1)})$$
$$(x_1^{(2)},\dots,x_n^{(2)})$$
$$\dots$$
$$(x_1^{(M)},\dots,x_n^{(M)})$$

# Two “Optimal” approaches

- “Optimal” here means the employed algorithms guarantee to return a structure that maximizes the objectives (e.g., LogLik)
  - Many heuristics used to be popular, but they provide no guarantee on attaining optimality, interpretability, or even do not have an explicit objective
  - E.g.: structured EM, Module network, greedy structural search, etc.

- We will learn two classes of algorithms for guaranteed structure learning, which are likely to be the only known methods enjoying such guarantee, but they only apply to certain families of graphs:
  - Trees: The Chow-Liu algorithm (this lecture)
  - Pairwise MRFs: covariance selection, neighborhood-selection (later)

# Recall Multivariate Gaussian

- Multivariate Gaussian density:

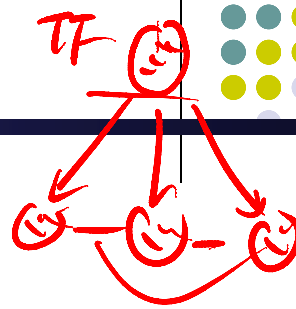
$$p(\mathbf{x} \mid \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left\{-\tfrac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)\right\}$$

- WOLG: let $\mu = 0 \quad Q = \Sigma^{-1}$

$$p(x_1, x_2, \cdots, x_p \mid \mu = 0, Q) = \frac{|Q|^{1/2}}{(2\pi)^{n/2}} \exp\left\{-\tfrac{1}{2}\sum_i q_{ii}(x_i)^2 - \sum_{i<j} q_{ij} x_i x_j\right\}$$

- We can view this as a continuous Markov Random Field with potentials defined on every node and edge:

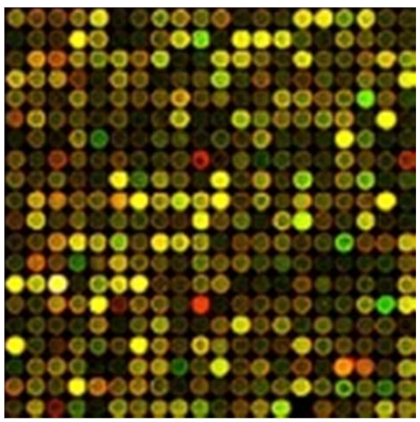
# Gaussian Graphical Model




**Cell type**

$$\boldsymbol{X}^{(n)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}^{(n)})$$

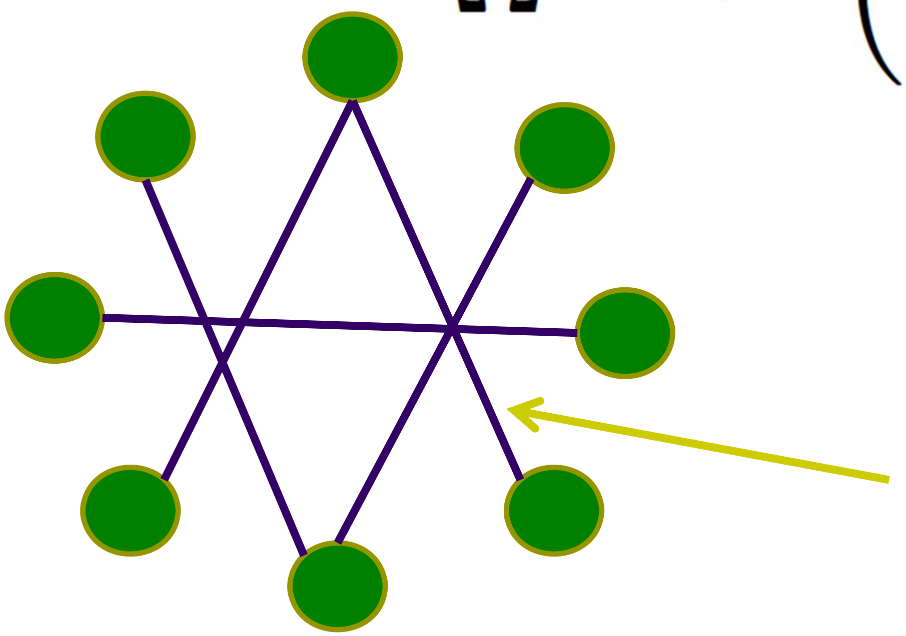
**Microarray samples**



**Encodes dependencies among genes**

# Precision Matrix Encodes Non-Zero Edges in Gaussian Graphical Modela

$$\boldsymbol{\Omega}^{(n)} = \left(\boldsymbol{\Sigma}^{(n)}\right)^{-1}$$



**Edge corresponds to non-zero precision matrix element**

# Markov versus Correlation Network

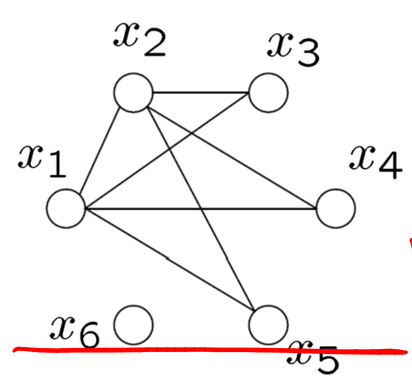
- ***Correlation network*** **is based on** ***Covariance Matrix***

$$\Sigma_{i,j} = 0 \quad \Rightarrow \quad X_i \perp X_j \quad \text{or} \quad p(X_i, X_j) = p(X_i)p(X_j)$$

- ***A GGM is a Markov Network*** **based on** ***Precision Matrix***
  - **Conditional Independence/Partial Correlation Coefficients are a more sophisticated dependence measure**

$$Q_{i,j} = 0 \quad \Rightarrow \quad X_i \perp X_j | \mathbf{X}_{-ij} \quad \text{or} \quad p(X_i, X_j | \mathbf{X}_{-ij}) = p(X_i | \mathbf{X}_{-ij}) p(X_j | \mathbf{X}_{-ij})$$

$$Q = \begin{pmatrix} * & * & * & * & * & 0 \\ * & * & * & * & * & 0 \\ * & * & * & 0 & 0 & 0 \\ * & * & 0 & * & 0 & 0 \\ * & * & 0 & 0 & * & 0 \\ 0 & 0 & 0 & 0 & 0 & * \end{pmatrix}$$




**With small sample size, empirical covariance matrix cannot be inverted**

# Sparsity

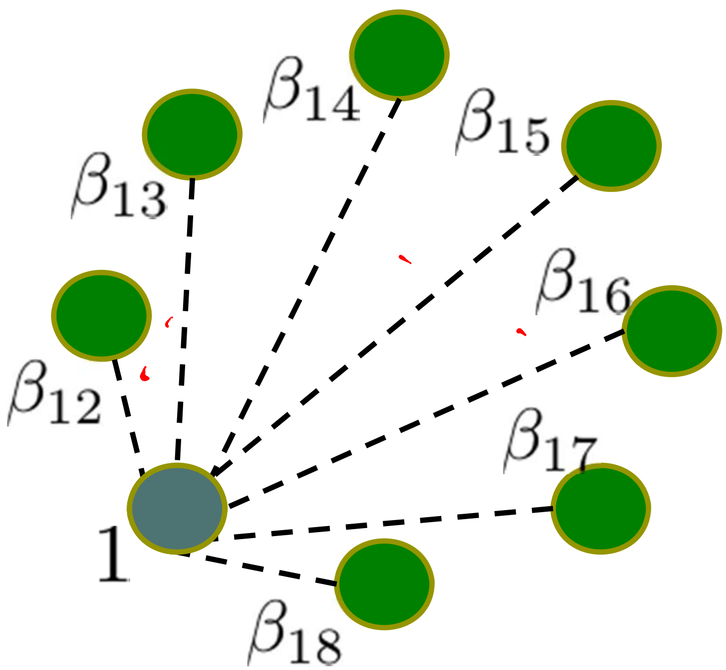
- One common assumption to make: **sparsity**
- **Makes empirical sense:** Genes are only assumed to interface with small groups of other genes.
- **Makes statistical sense:** Learning is now feasible in high dimensions with small sample size

$$\Omega^{(n)} = \left(\Sigma^{(n)}\right)^{-1}$$

**sparse**

# Network Learning with the LASSO

- Assume network is a Gaussian Graphical Model
- Perform LASSO regression of all nodes to a target node

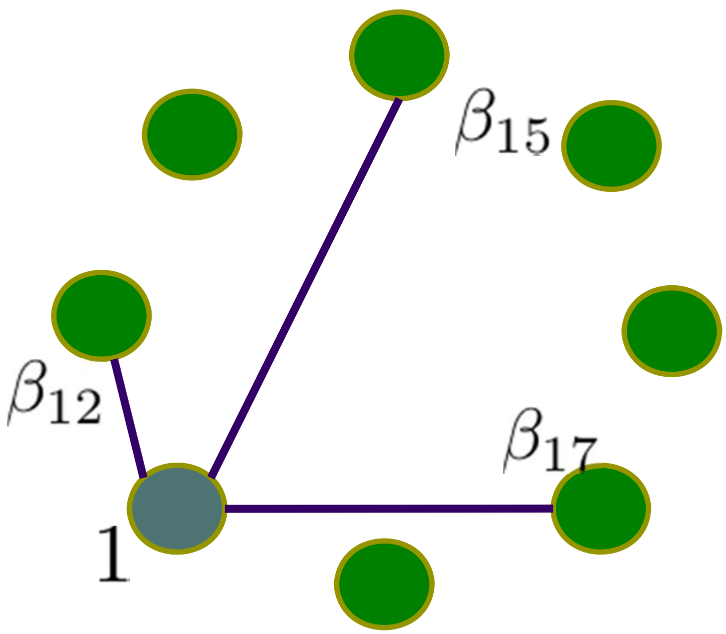



$x_1 = \beta^{J} X_{-1}$

# Network Learning with the LASSO

- LASSO can select the neighborhood of each node

$$\hat{\boldsymbol{\beta}}_{\mathbf{1}} = \mathrm{argmin}_{\beta_1} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}_{\mathbf{1}}\|^2 + \lambda \|\boldsymbol{\beta}_{\mathbf{1}}\|_1$$



# L1 Regularization (LASSO)

- A convex relaxation.

**Constrained Form**

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\beta} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2$$

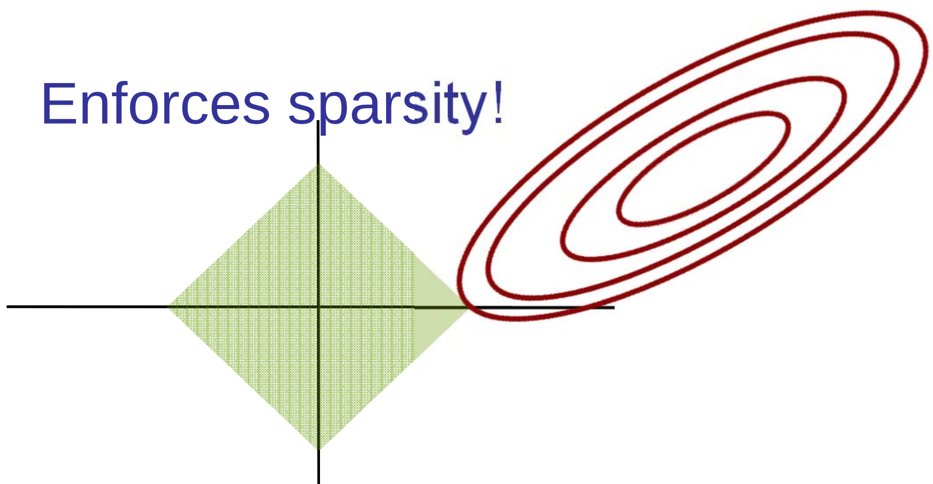
subject to:

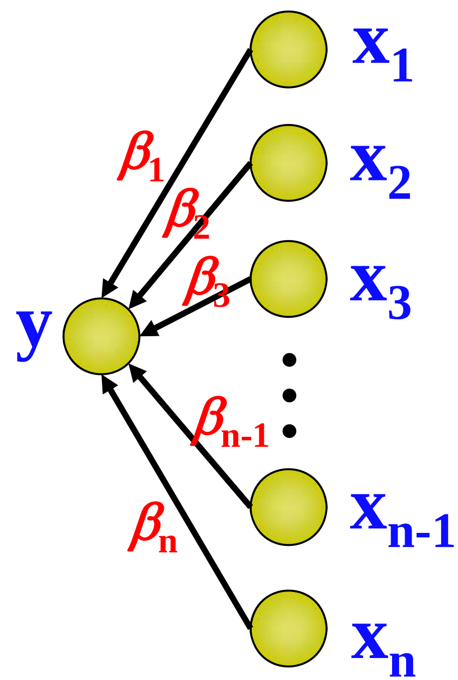
$$\sum_{j=1}^{p} |\beta_j| \leq C$$

**Lagrangian Form**

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\beta} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|_1$$

- Enforces sparsity!





# Theoretical Guarantees

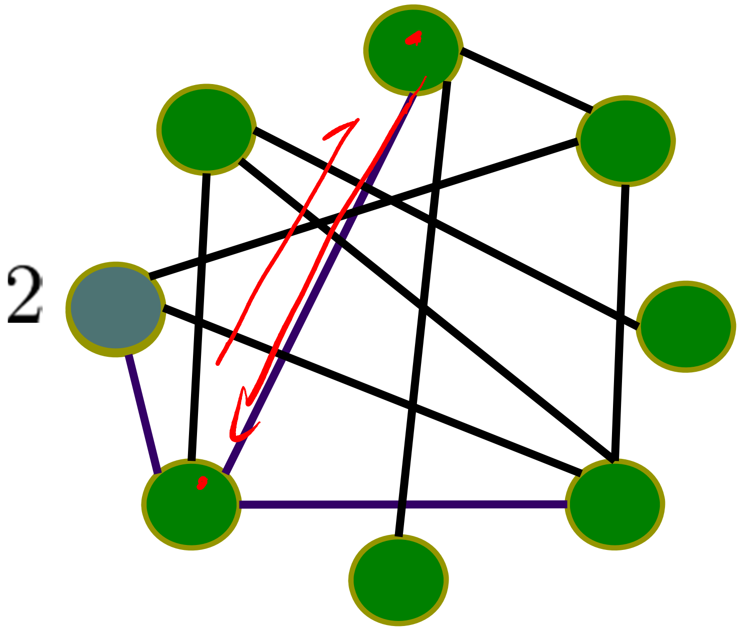
- Assumptions
  - Dependency Condition: Relevant Covariates are not overly dependent
  - Incoherence Condition: Large number of irrelevant covariates cannot be too correlated with relevant covariates
  - Strong concentration bounds: Sample quantities converge to expected values quickly

**If these are assumptions are met, LASSO will asymptotically recover correct subset of covariates that relevant.**

# Network Learning with the LASSO

- Repeat this for every node
- Form the total edge set

$$\hat{\mathcal{E}} = \{(u,v) : \max(|\hat{\beta}_{uv}|, |\hat{\beta}_{vu}|) > 0\}$$



# Consistent Structure Recovery

**[Meinshausen and Buhlmann 2006, Wainwright 2009]**

**If** $$\lambda_s > C\sqrt{\frac{\log p}{S}}$$

**Then with high probability,**

$$S(\hat{\boldsymbol{\beta}}) \rightarrow S(\boldsymbol{\beta}^*)$$

# Why this algorithm work?

- What is the intuition behind graphical regression?
  - Continuous nodal attributes
  - Discrete nodal attributes

- Are there other algorihtms?

- More general scenarios:
  non-iid sample and evolving networks

- Case study

# Multivariate Gaussian

- Multivariate Gaussian density:

$$p(\mathbf{x}\,|\,\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left\{-\tfrac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)\right\}$$

- A joint Gaussian:

$$p\left(\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix} \,\middle|\, \mu,\Sigma\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix} \,\middle|\, \begin{bmatrix}\mu_1\\ \mu_2\end{bmatrix}, \begin{bmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{bmatrix}\right)$$

- How to write down $p(\mathbf{x}_2)$, $p(\mathbf{x}_1|\mathbf{x}_2)$ or $p(\mathbf{x}_2|\mathbf{x}_1)$ using the block elements in $\mu$ and $\Sigma$?
  - Formulas to remember:

$$p(\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_2\,|\,\mathbf{m}_2^m, \mathbf{V}_2^m)$$

$$\mathbf{m}_2^m = \mu_2$$

$$\mathbf{V}_2^m = \Sigma_{22}$$

$$p(\mathbf{x}_1|\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_1\,|\,\mathbf{m}_{1|2}, \mathbf{V}_{1|2})$$

$$\mathbf{m}_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}_2 - \mu_2)$$

$$\mathbf{V}_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

# The matrix inverse lemma

- Consider a block-partitioned matrix: $M = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$
- First we diagonalize $M$

$$\begin{bmatrix} I & -FH^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} \begin{bmatrix} I & 0 \\ -H^{-1}G & I \end{bmatrix} = \begin{bmatrix} E-FH^{-1}G & 0 \\ 0 & H \end{bmatrix}$$

  - Schur complement: $M/H = E-FH^{-1}G$

- Then we inverse, using this formula: $XYZ = W \Rightarrow Y^{-1} = ZW^{-1}X$

$$M^{-1} = \begin{bmatrix} E & F \\ G & H \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ -H^{-1}G & I \end{bmatrix} \begin{bmatrix} (M/H)^{-1} & 0 \\ 0 & H^{-1} \end{bmatrix} \begin{bmatrix} I & -FH^{-1} \\ 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} (M/H)^{-1} & -(M/H)^{-1}FH^{-1} \\ -H^{-1}G(M/H)^{-1} & H^{-1} + H^{-1}G(M/H)^{-1}FH^{-1} \end{bmatrix} = \begin{bmatrix} E^{-1} + E^{-1}F(M/E)^{-1}GE^{-1} & -E^{-1}F(M/E)^{-1} \\ -(M/E)^{-1}GE^{-1} & (M/E)^{-1} \end{bmatrix}$$

- Matrix inverse lemma

$$\left(E-FH^{-1}G\right)^{-1} = E^{-1} + E^{-1}F\left(H-GE^{-1}F\right)^{-1}GE^{-1}$$

# The covariance and the precision matrices

$$\Sigma = \begin{bmatrix} \sigma_{11} & \vec{\sigma}_1^T \\ \vec{\sigma}_1 & \Sigma_{-1} \end{bmatrix}$$

$$\Downarrow$$

$$M^{-1} = \begin{bmatrix} (M/H)^{-1} & -(M/H)^{-1} F H^{-1} \\ -H^{-1} G (M/H)^{-1} & H^{-1} + H^{-1} G (M/H)^{-1} F H^{-1} \end{bmatrix}$$

$$\Downarrow$$

$$Q = \begin{bmatrix} q_{11} & -q_{11} \vec{\sigma}_1^T \Sigma_{-1}^{-1} \\ -q_{11} \Sigma_{-1}^{-1} \vec{\sigma}_1 & \Sigma_{-1}^{-1} \left( I + q_{11} \vec{\sigma}_1 \vec{\sigma}_1^T \Sigma_{-1}^{-1} \right) \end{bmatrix} = \begin{bmatrix} q_{11} & \vec{q}_1^T \\ \vec{q}_1 & Q_{-1} \end{bmatrix}$$

$$p(\mathbf{x}_1|\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_1 \mid \mathbf{m}_{1|2}, \mathbf{V}_{1|2})$$
$$\mathbf{m}_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}_2 - \mu_2)$$
$$\mathbf{V}_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

# Single-node Conditional

- The conditional dist. of a single node *i* given the rest of the nodes can be written as:

$$p(X_i|\mathbf{X}_{-i}) = \mathcal{N}\Big(\mu_i + \Sigma_{X_i\mathbf{X}_{-i}}\Sigma^{-1}_{\mathbf{X}_{-i}\mathbf{X}_{-i}}(\mathbf{X}_{-i} - \mu_{\mathbf{X}_{-i}}), \\ \Sigma_{X_iX_i} - \Sigma_{X_i\mathbf{X}_{-i}}\Sigma^{-1}_{\mathbf{X}_{-i}\mathbf{X}_{-i}}\Sigma_{\mathbf{X}_{-i}X_i}\Big)$$

- WOLG: let $\mu = 0$

$$\begin{aligned} p(X_i|\mathbf{X}_{-i}) &= \mathcal{N}\big(\Sigma_{X_i\mathbf{X}_{-i}}\Sigma^{-1}_{\mathbf{X}_{-i}\mathbf{X}_{-i}}\mathbf{X}_{-i}, \Sigma_{X_iX_i} - \Sigma_{X_i\mathbf{X}_{-i}}\Sigma^{-1}_{\mathbf{X}_{-i}\mathbf{X}_{-i}}\Sigma_{\mathbf{X}_{-i}X_i}\big) \\ &= \mathcal{N}\big(\vec{\sigma}_i^T\Sigma_{-i}^{-1}\mathbf{X}_{-i}, q_{i|-i}\big) \\ &= \mathcal{N}\Big(\frac{\vec{q}_i^T}{-q_{ii}}\mathbf{X}_{-i}, q_{i|-i}\Big) \end{aligned}$$

$$Q = \begin{bmatrix} q_{11} & -q_{11}\vec{\sigma}_1^T\Sigma_{-1}^{-1} \\ -q_{11}\Sigma_{-1}^{-1}\vec{\sigma}_1 & \Sigma_{-1}^{-1}\left(I + q_{11}\vec{\sigma}_1\vec{\sigma}_1^T\Sigma_{-1}^{-1}\right) \end{bmatrix} = \begin{bmatrix} q_{11} & \vec{q}_1^T \\ \vec{q}_1 & Q_{-1} \end{bmatrix}$$

# Conditional auto-regression

- From

$$p(X_i|\mathbf{X}_{-i}) \;=\; \mathcal{N}\Big(\frac{\vec{q}_i^{\,T}}{-q_{ii}}\mathbf{X}_{-i}, q_{i|-i}\Big)$$

- We can write the following conditional auto-regression function for each node:

- Neighborhood est. based on auto-regression coefficient

$$S_i \equiv \{j \quad : \quad j \neq i, \theta_{ij} \neq 0\}$$

# Conditional independence

- From

$$p(X_i|\mathbf{X}_{-i}) \;=\; \mathcal{N}\big(\frac{\vec{q}_i^T}{-q_{ii}}\mathbf{X}_{-i}, q_{ii}\big)$$

- Given an estimate of the neighborhood $s_i$, we have:

$$p(X_i|\mathbf{X}_{-i}) \;=\; p(X_i|\mathbf{X}_s)$$

- Thus the neighborhood $s_i$ defines the Markov blanket of node $i$

# Recent trends in GGM:

- Covariance selection (classical method)
  - Dempster [1972]:
    - Sequentially pruning smallest elements in precision matrix
  - Drton and Perlman [2008]:
    - Improved statistical tests for pruning

Serious limitations in practice: breaks down when covariance matrix is not invertible

- $L_1$-regularization based method (*hot* !)
  - Meinshausen and Bühlmann [Ann. Stat. 06]:
    - Used LASSO regression for neighborhood selection
  - Banerjee [JMLR 08]:
    - Block sub-gradient algorithm for finding precision matrix
  - Friedman et al. [Biostatistics 08]:
    - Efficient fixed-point equations based on a sub-gradient algorithm
  - …

Structure learning is possible even when # variables > # samples

# The Meinshausen-Bühlmann (MB) algorithm:

- Solving separated Lasso for every single variables:

$$x_1, x_2, \cdots, x_{k-1}, x_k, x_{k+1}, \cdots, x_p$$

Step 1: Pick up one variable

$$z = x_1, x_2, \cdots, x_{k-1}, \quad x_{k+1}, \cdots, x_p$$

Step 2: Think of it as "y", and the rest as "z"

$$y$$

Step 3: Solve Lasso regression problem between y and z

$$y = \theta^\top z$$

**The resulting coefficient does not correspond to the Q value-wise**

Step 4: Connect the *k*-th node to those having nonzero weight in ***w***

# $L_1$-regularized maximum likelihood learning

- Input: Sample covariance matrix S $\quad S_{i,j} \equiv \frac{1}{N}\sum_{n=1}^{N} x_i^{(n)} x_j^{(n)}$
  - Assumes standardized data (mean=0, variance=1)
  - S is generally rank-deficient
    - Thus the inverse does not exist
- Output: Sparse precision matrix Q
  - Originally, Q is defined as the inverse of S, but not directly invertible
  - Need to find a sparse matrix that can be thought as of as an inverse of S

$$Q^* = \arg\max_{Q} \{ \ln\det Q - \mathrm{tr}(SQ) - \rho \|Q\|_1 \}$$

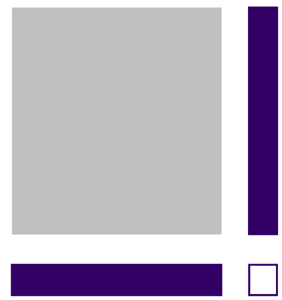
log likelihood $\ln \prod_{t=1}^{N} \mathcal{N}(\boldsymbol{x}^{(t)} | \mathbf{0}, Q^{-1})$ regularizer

- Approach: Solve an $L_1$-regularized maximum likelihood equation

# From matrix opt. to vector opt.: *coupled* Lasso for every single Var.

- Focus only on one row (column), keeping the others constant

$$Q = \begin{pmatrix} L & \boldsymbol{l} \\ \boldsymbol{l}^\top & \lambda \end{pmatrix}$$



- Optimization problem for blue vector is shown to be Lasso **($L_1$-regularized quadratic programming)**

- Difference from MB's: Resulting Lasso problems are <u>coupled</u>
  - The gray part is actually not constant; changes after solving one Lasso problem (because it is the opt of the entire Q that optimize a single loss function, whereas in MB each lasso has its own loss function..
  - This coupling is essential for stability under noise

# Learning Ising Model (i.e. pairwise MRF)

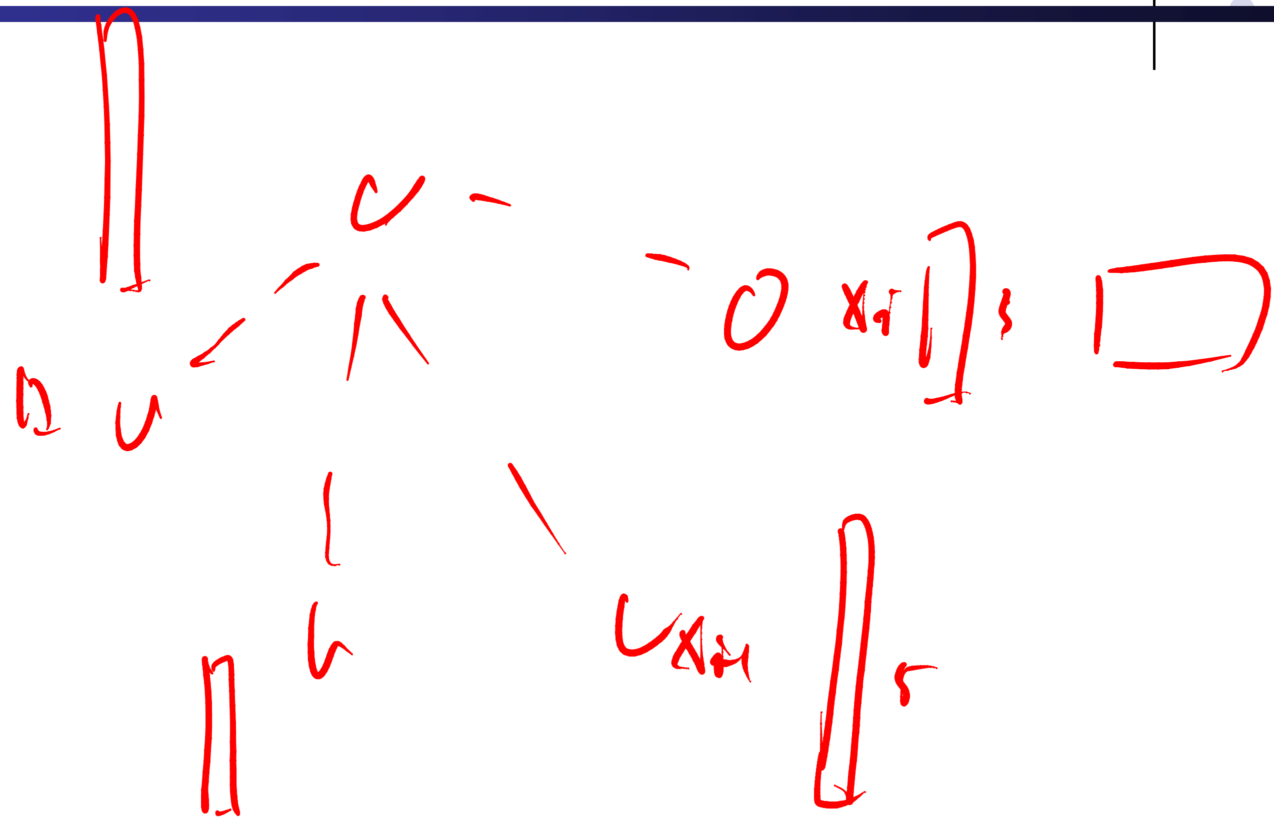
- Assuming the nodes are discrete (e.g., voting outcome of a person), and edges are weighted, then for a sample x, we have

$$P(\mathbf{x}|\Theta) = \exp\left(\sum_{i \in V} \theta_{ii}^t x_i + \sum_{(i,j) \in E} \theta_{ij} x_i x_j - A(\Theta)\right)$$

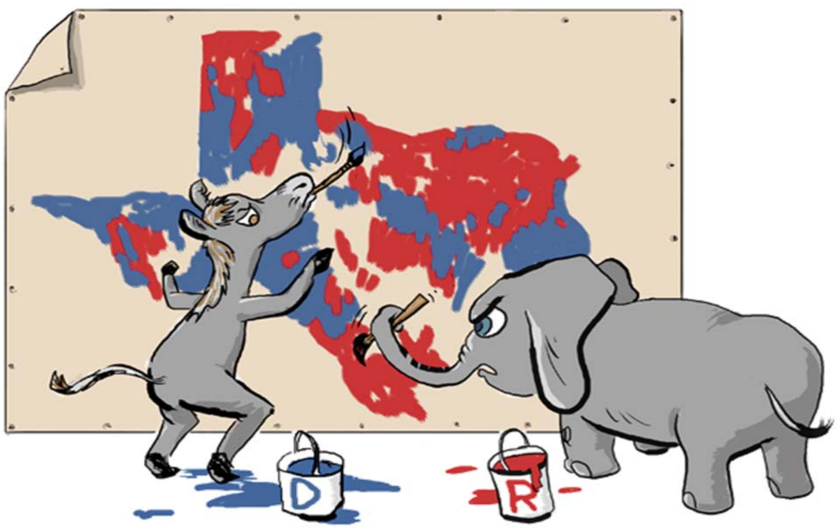
- It can be shown the pseudo-conditional likelihood for node k is

$$\mathbb{P}_\theta(x_k | x_{\setminus k}) = \text{logistic}\left(2 x_k \left\langle \theta_{\setminus k}, x_{\setminus k} \right\rangle\right)$$

# Question: vector-valued nodes



# New Problem:
# Evolving Social Networks




**Can I get his vote?**

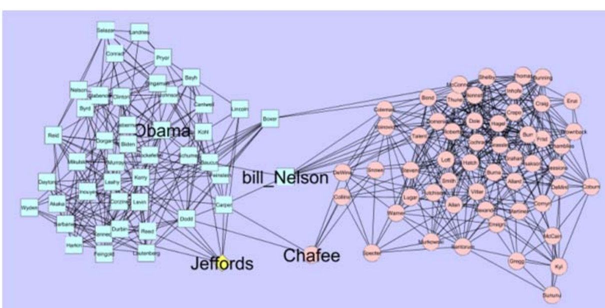
Corporativity,

Antagonism,

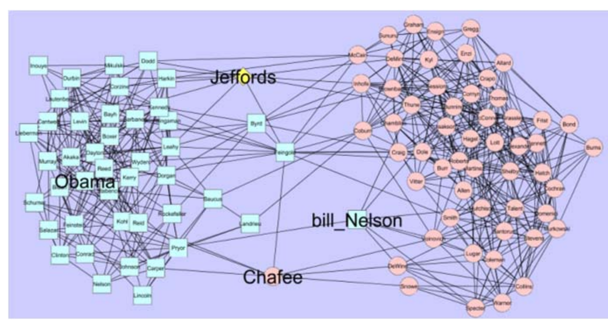
Cliques,

…

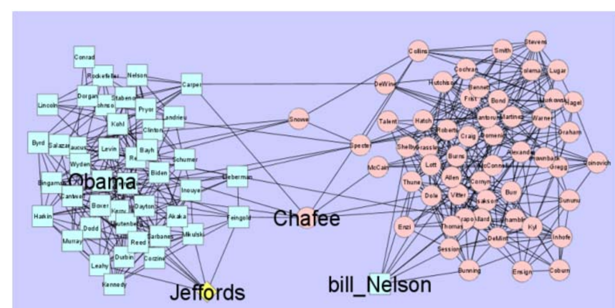
over time?




**March 2005**

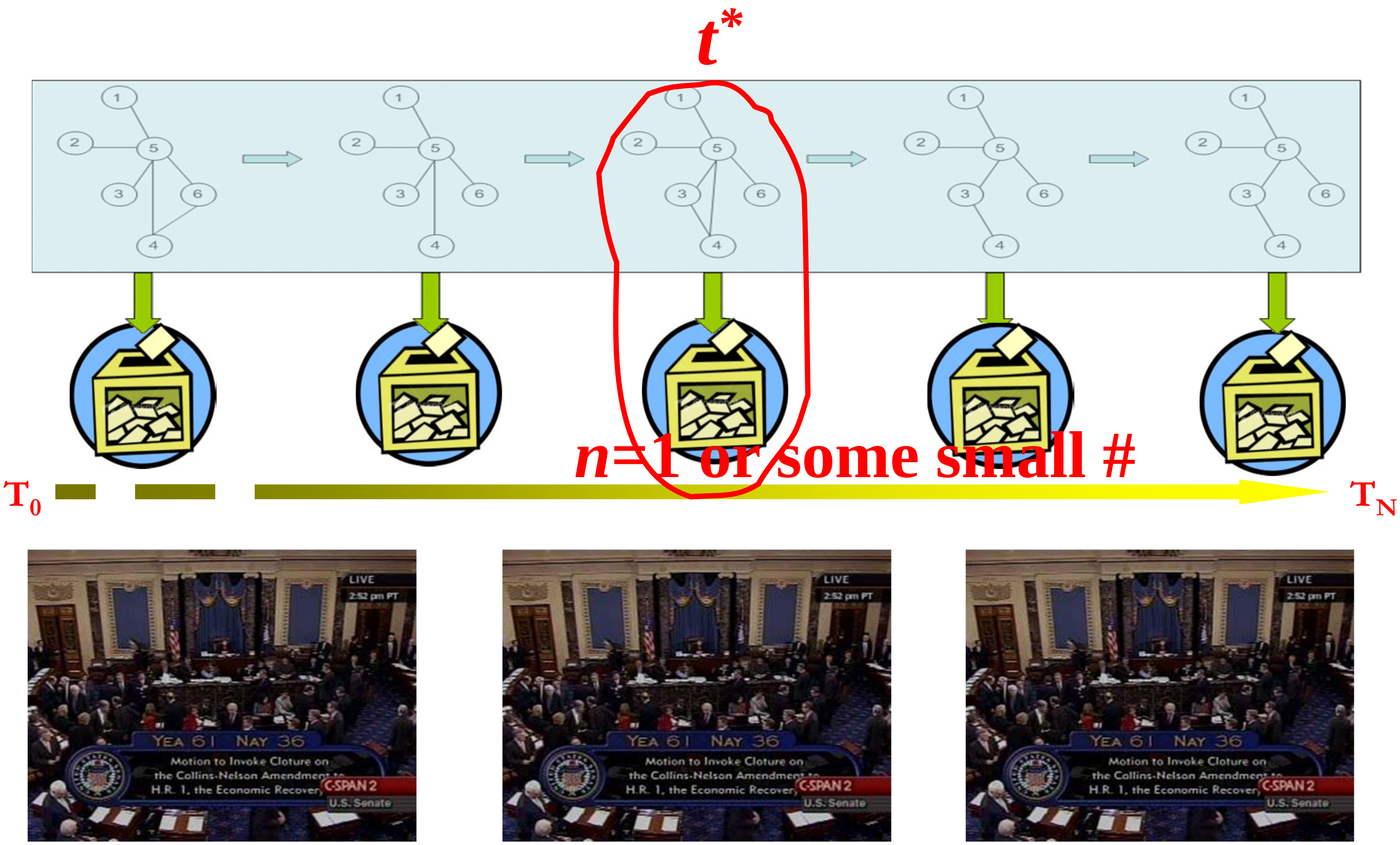



**January 2006**




**August 2006**

# Reverse engineering time-specific "rewiring" networks



# Inference I

[Song, Kolar and Xing, Bioinformatics 09]

- **KELLER**: Kernel Weighted $L_1$-regularized Logistic Regression

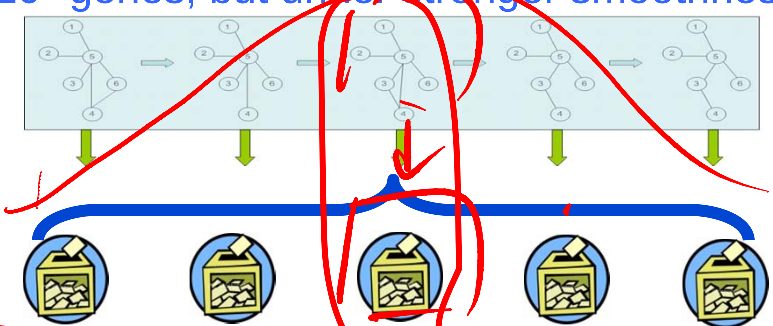
$$\hat{\theta}_i^t = \arg\min_{\theta_i^t} l_w(\theta_i^t) + \lambda_1 \| \theta_i^t \|_1 \quad \forall t$$

$$\text{where } l_w(\theta_i^t) = \sum_{t'=1}^{T} w(\mathbf{x}^{t'}; \mathbf{x}^t) \log P(x_i^{t'} | \mathbf{x}_{-i}^{t'}, \theta_i^t).$$

**Lasso:**

$$\hat{\theta} = \arg\min_{\theta} \sum_{n=1}^{N} \gamma(\mathbf{x}^{(n)}; \theta) + \lambda_1 \| \theta \|_1$$

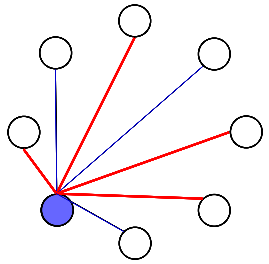
- Constrained convex optimization
  - **Estimate time-specific nets one by one, based on "virtual iid" samples**
  - Could scale to ~$10^4$ genes, but under stronger smoothness assumptions



# Algorithm – nonparametric neighborhood selection

- Conditional likelihood

$$\mathbb{P}_{\theta^t}(x_i^t | x_{\setminus i}^t) = \text{logistic}\left(2x_i^t \langle \theta_{\setminus i}^t, x_{\setminus i}^t \rangle\right)$$



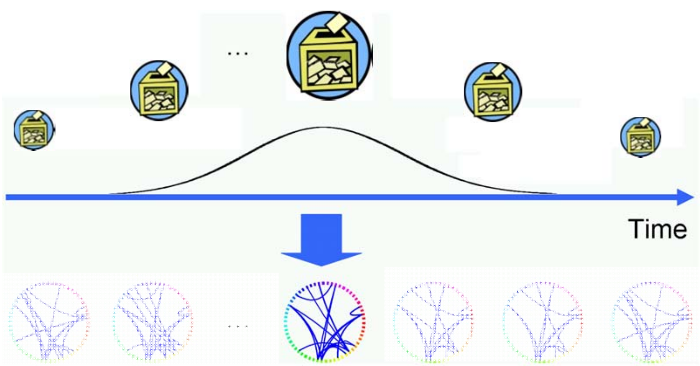
  - Neighborhood Selection: $S(x_i) = \{j \mid \theta_{i,j}^t \neq 0\}$

- Time-specific graph regression:
  - Estimate at $t^* \in [0,1]$

$$\min_{\theta \in \mathbb{R}^{p_n - 1}} \left\{ -\sum_{t \in \mathcal{T}^n} w_t(t^*) \gamma(\theta_i; x^t) + \lambda_1 \|\theta_i\|_1 \right\}$$

Where $\gamma(\theta_i^t; x^t) = \log \mathbb{P}_{\theta_i^t}(x_i^t | x_{\setminus i}^t)$

and $w_t(t^*) = \dfrac{K_{h_n}(t - t^*)}{\sum_{t' \in \mathcal{T}^n} K_{h_n}(t' - t^*)}$



# Structural consistency of KELLER

## Assumptions

- Define: $Q_u^t := \mathbb{E}\left[\nabla^2 \log \mathbb{P}_{\theta^t}[X_u | X_{\setminus u}]\right], \quad \forall u \in V \qquad \Sigma_u^t := \mathbb{E}\left[X_{\setminus u}^t {X_{\setminus u}^t}^T\right], \quad \forall u \in V$

  $s = \max_u \max_t |S_u^t|, \quad \theta_{\min} = \min_{e \in E} \max |\theta_e^t|$
- A1: Dependency Condition

$$\Lambda_{\min}(Q_{SS}^{t^*}) \geq C_{\min}, \quad \forall t \in [0,1]$$

$$\Lambda_{\max}\left(\Sigma^{t^*}\right) \leq D_{\max}, \quad \forall t \in [0,1]$$

- A2: Incoherence Condition $\exists \alpha \in (0,1]$ such that

$$\left\| Q_{S^c S}^{t^*} (Q_{SS}^{t^*})^{-1} \right\|_\infty \leq 1 - \alpha, \quad \forall t^* \in [0,1]$$

- A3: Smoothness Condition

$$\max_{u,v} \sup_{t^*} |\sigma'_{uv}(t^*)| \leq A_0, \quad \max_{u,v} \sup_{t^*} |\sigma''_{uv}(t^*)| \leq A$$

$$\max_{u,v} \sup_{t^*} |\theta'_{uv}(t^*)| \leq B_0, \quad \max_{u,v} \sup_{t^*} |\theta''_{uv}(t^*)| \leq B$$

- A4: Bounded Kernel

$$\exists M_k \geq 1 \qquad \max_{z \in \mathbb{R}} |K(z)| \leq M_k \quad \max_{z \in \mathbb{R}} K(z)^2 \leq M_k$$

# Theorem

[Kolar and Xing, 09]

Assume that A1, A2, A3, A4 hold. Furthermore, assume that the following conditions hold:

1. $h_n = \mathcal{O}(n^{-\frac{1}{3}})$
2. $s_n h_n = o(1)$,
3. $\frac{s_n^3 \log p_n}{n h_n} = o(1)$
4. $\lambda_1 = \mathcal{O}(\sqrt{\frac{\log p}{n h_n}})$
5. $\theta^*_{\min} = \Omega(\sqrt{\frac{s_n \log p_n}{n h_n}})$

then

$$\mathbb{P}\left[\hat{G}(\lambda_1, h_n, t^*) \neq G^{t^*}\right] = \mathcal{O}\left(\exp\left(-C\frac{n h_n}{s_n^3} + C' \log p\right)\right) \to 0$$

# Inference II

[Amr and Xing, PNAS 2009, AOAS 2009]

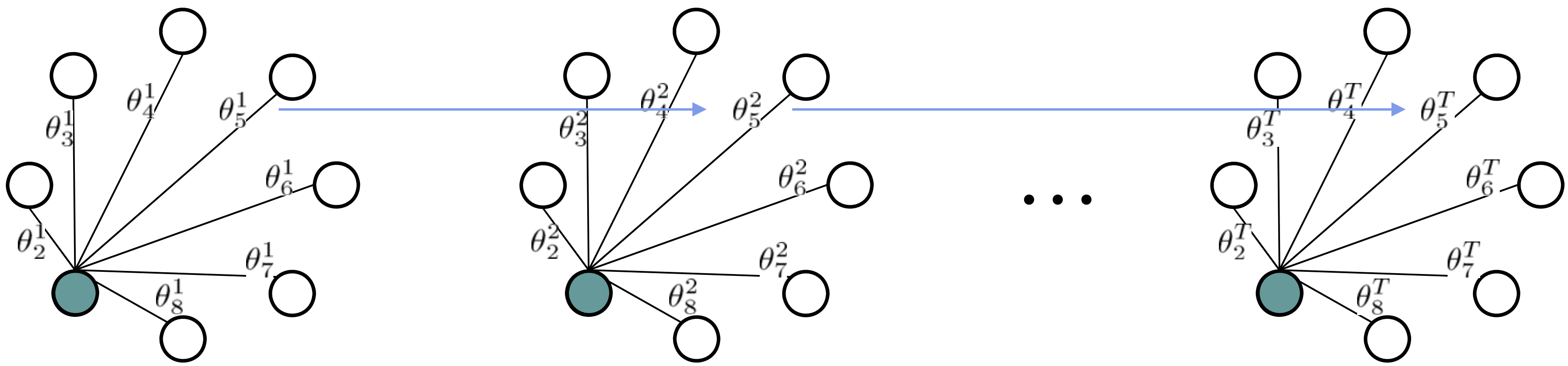
- **TESLA**: Temporally Smoothed $L_1$-regularized logistic regression

$$\hat{\theta}_i^1, \ldots, \hat{\theta}_i^T = \arg\min_{\theta_i^1,\ldots,\theta_i^T} \sum_{t=1}^{T} l_{avg}(\theta_i^t) + \lambda_1 \sum_{t=1}^{T} \| \theta_{-i}^t \|_1 + \lambda_2 \sum_{t=2}^{T} \| \theta_i^t - \theta_i^{t-1} \|_q^q,$$

$$\text{where} \quad l_{avg}(\theta_{\mathbf{i}}^{\mathbf{t}}) = \frac{1}{N^t} \sum_{d=1}^{N^t} \log P\left(x_{d,i}^t | \mathbf{x}_{\mathbf{d},-\mathbf{i}}^{\mathbf{t}}, \theta_{\mathbf{i}}^{\mathbf{t}}\right).$$

- Constrained convex optimization
  - Scale to ~5000 nodes, does not need smoothness assumption, can accommodate abrupt changes.

# Temporally Smoothed Graph Regression




**TESLA:**

$$\min_{\substack{\theta_i^1,\ldots,\theta_i^T \\ \mathbf{u}_i^1,\ldots,\mathbf{u}_i^T;\mathbf{v}_i^2,\ldots,\mathbf{v}_i^T}} \sum_{t=1}^{T} \ell(\mathbf{x}^t;\theta_i^t) + \lambda_1 \sum_{t=1}^{T} \mathbf{1}'\mathbf{u}_i^t + \lambda_2 \sum_{t=2}^{T} \mathbf{1}'\mathbf{v}_i^t$$

$$\text{s.t.} \quad -u_{i,j}^t \leq \theta_{i,j}^t \leq u_{i,j}^t,\ t = 1,\ldots,T,\ \forall j \in V \setminus i,$$

$$\text{s.t.} \quad -v_{i,j}^t \leq \theta_{i,j}^t - \theta_{i,j}^{t-1} \leq v_{i,j}^t,\ t = 2,\ldots,T,\ \forall j \in V \setminus i,$$

# Modified estimation procedure

- estimate block partition on which the coefficient functions are constant

$$\min_{\beta} \sum_{i=1}^{n} (Y_i - \mathbf{X}_i \beta(t_i))^2 + 2\lambda_2 \sum_{k=1}^{p} ||\beta_k||_{\mathrm{TV}} \qquad (*)$$

- estimate the coefficient functions on each block of the partition

$$\min_{\gamma \in \mathbb{R}^p} \sum_{t_i \in \hat{\mathcal{B}}_j} (Y_i - \mathbf{X}_i \gamma)^2 + 2\lambda_1 ||\gamma||_1 \qquad (**)$$

# Structural Consistency of TESLA

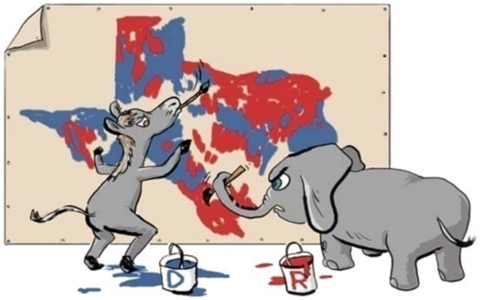
[Kolar, and Xing, 2009]

I. It can be shown that, by applying the results for model selection of the Lasso on a *temporal difference transformation* of (*), **the block are estimated consistently**

II. Then it can be further shown that, by applying Lasso on (**), **the neighborhood of each node on each of the estimated blocks consistently**

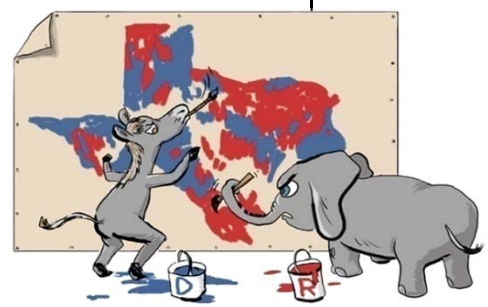
- Further advantages of the two step procedure
  - choosing parameters easier
  - faster optimization procedure

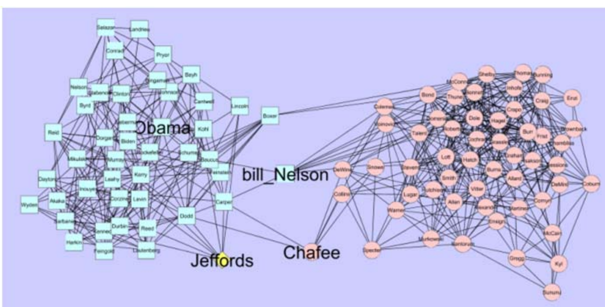
# Senate network – 109$^{th}$ congress



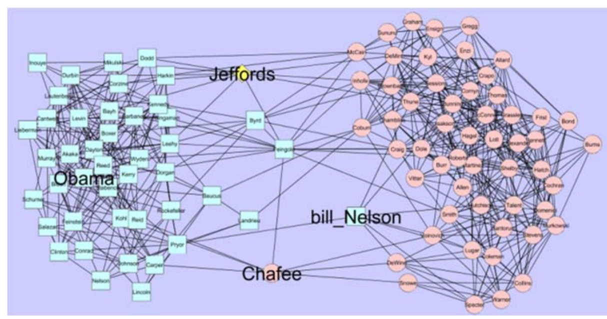

- Voting records from 109th congress (2005 - 2006)
- There are 100 senators whose votes were recorded on the 542 bills, each vote is a binary outcome

# Senate network – 109th congress

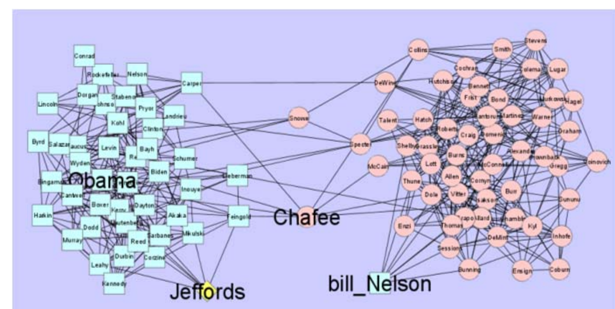



**March 2005**

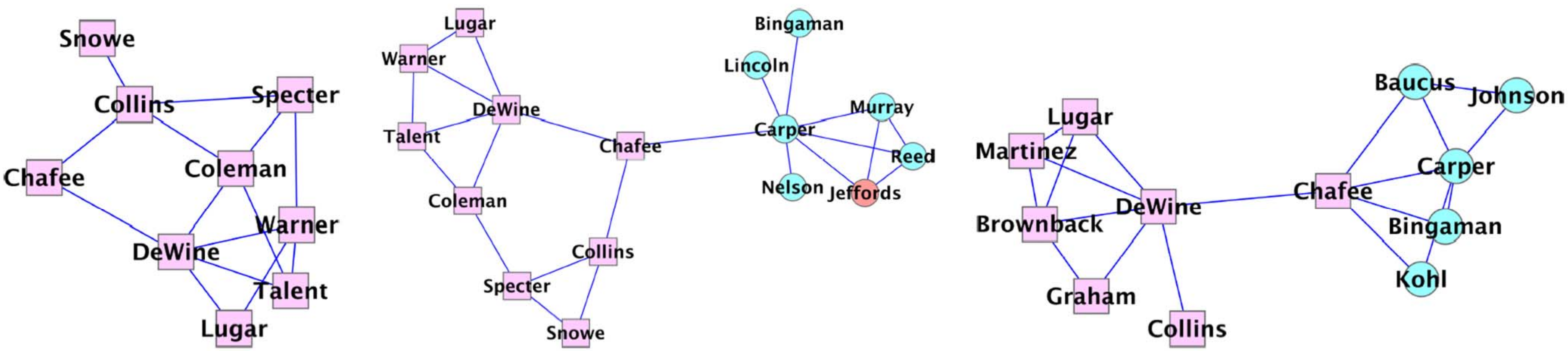



**January 2006**



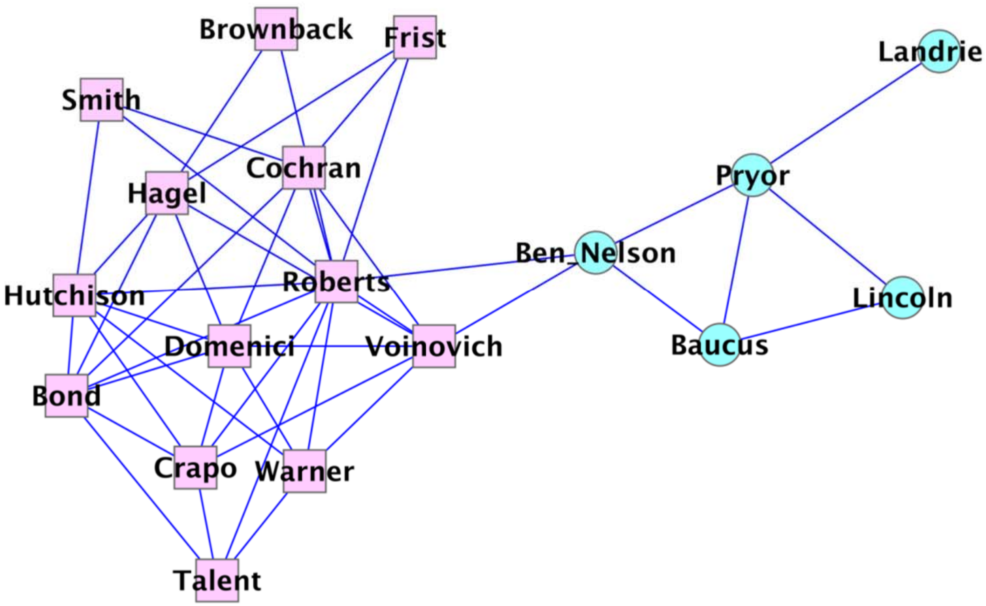

**August 2006**

# Senator Chafee

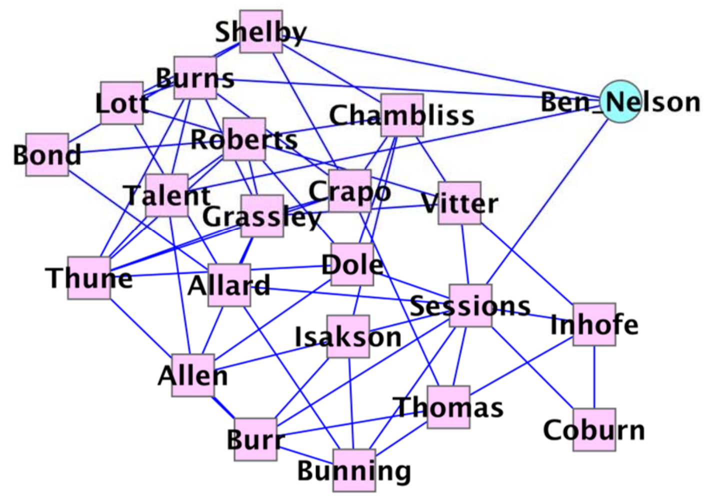



(a) $t = 0.1$ (b) $t = 0.4$ (c) $t = 0.8$

# Senator Ben Nelson

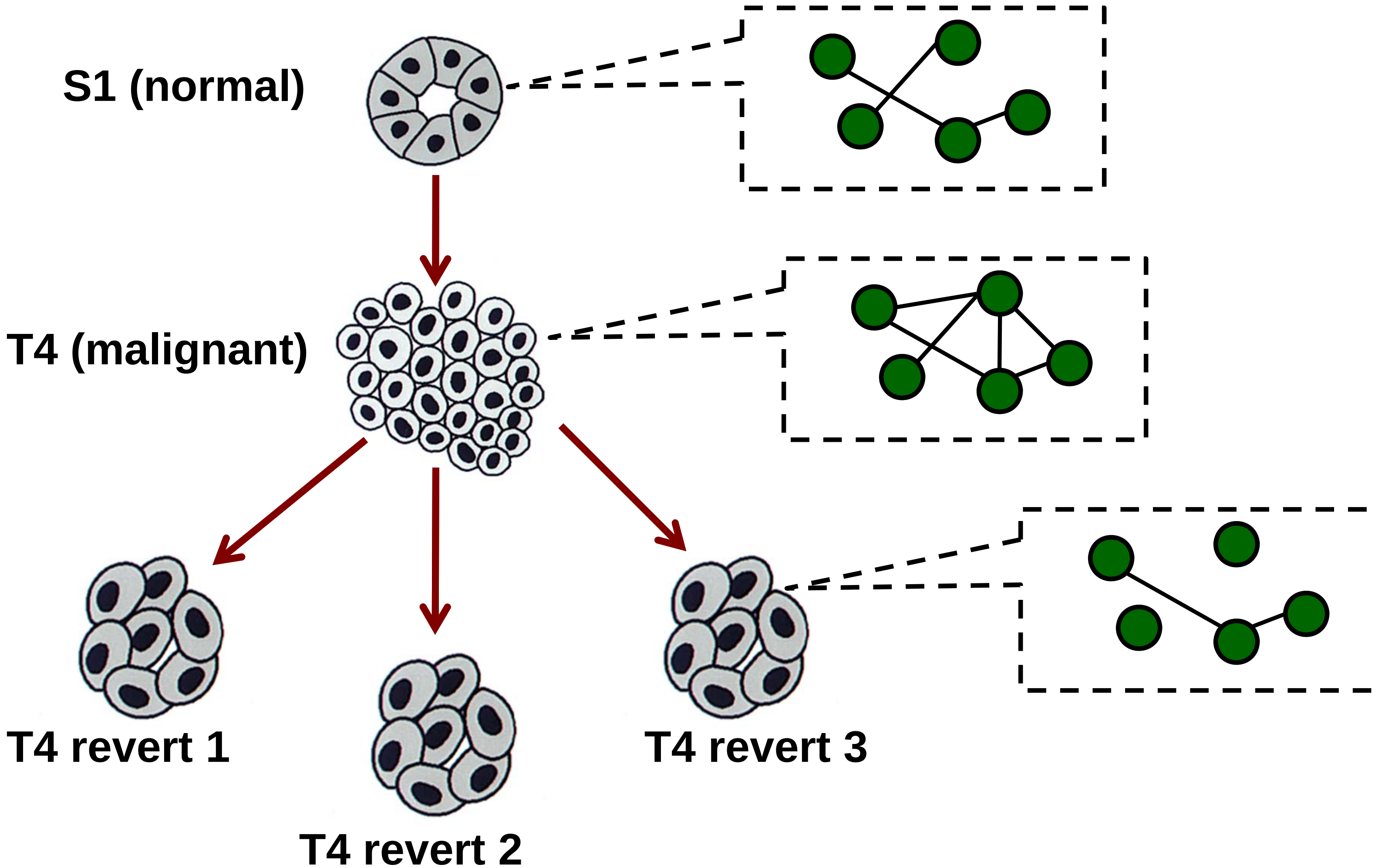



T=0.2

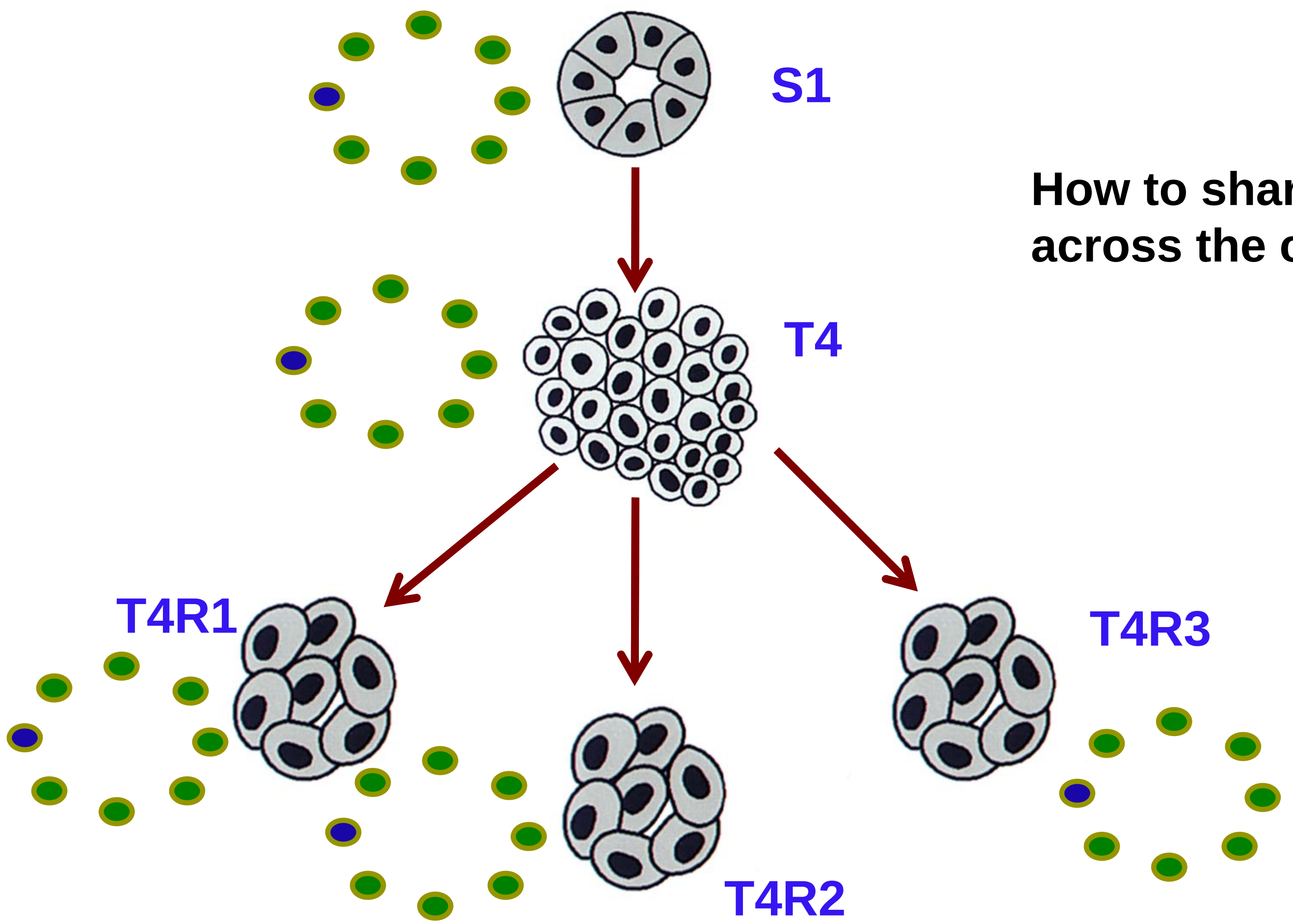



T=0.8

# Progression and Reversion of Breast Cancer cells



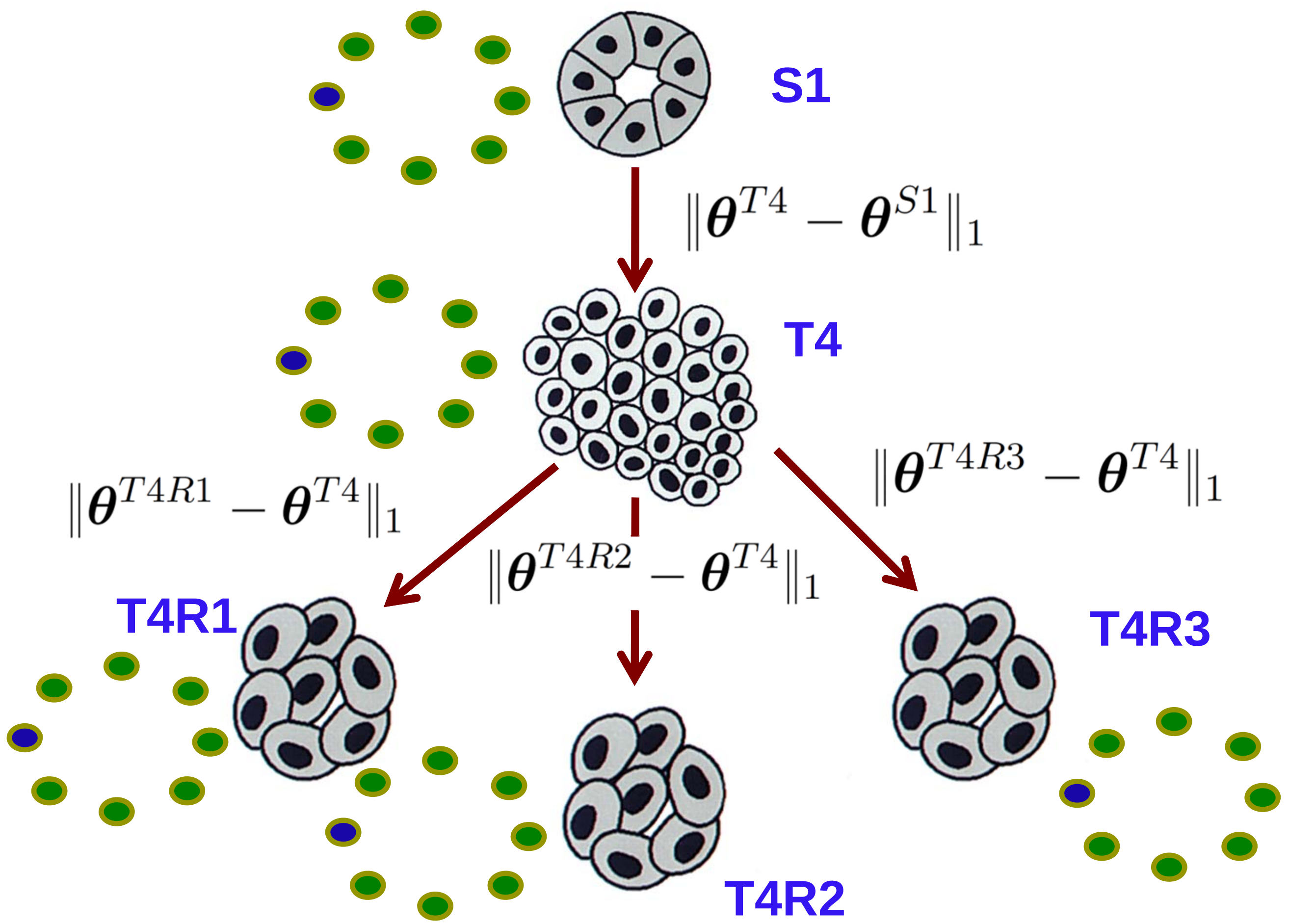
# Estimate Neighborhoods Jointly Across All Cell Types



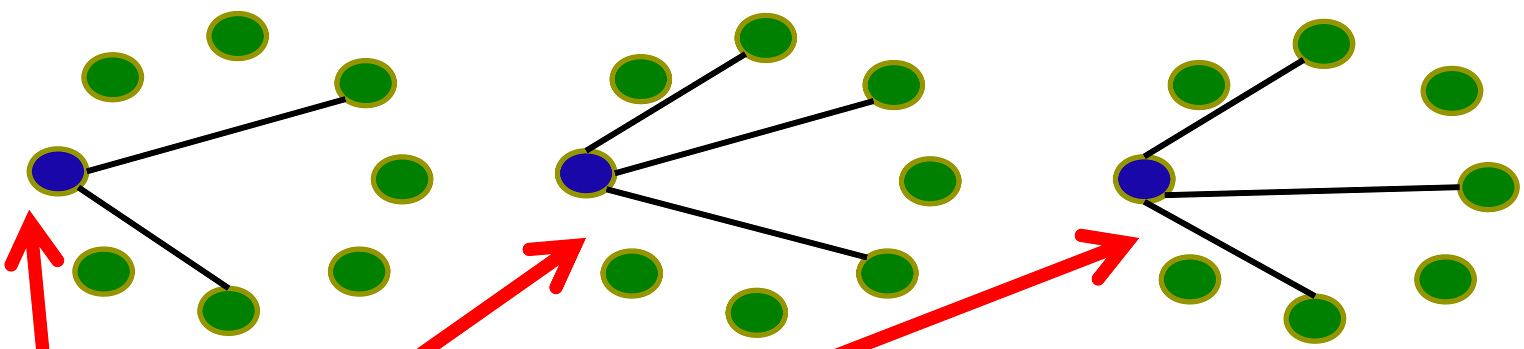

**How to share information across the cell types?**

# Sparsity of Difference

**Penalize differences between networks of adjacent cell types**



# Tree-Guided Graphical Lasso (Treegl)



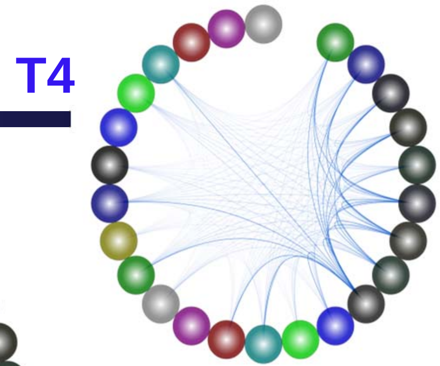
RSS for all cell types

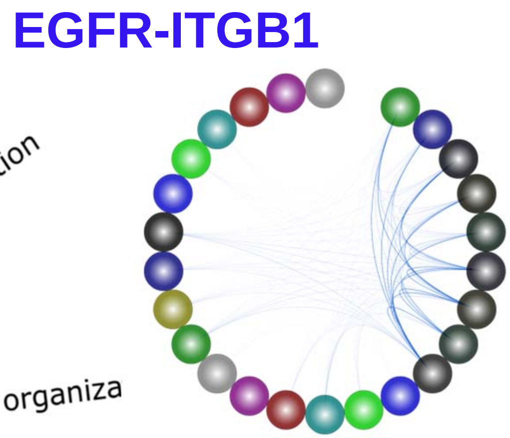
$$\hat{\boldsymbol\theta}^{(1)}_{\setminus u},\ldots,\hat{\boldsymbol\theta}^{(n)}_{\setminus u} = \operatorname*{argmin}_{\boldsymbol\theta^{(1)}_{\setminus u},\ldots,\boldsymbol\theta^{(n)}_{\setminus u}} \left( \sum_{n=1}^{N}\sum_{s=1}^{S_n}\left(x_u^{(n,s)} - \boldsymbol\theta^{(n)}_{\setminus u}\boldsymbol{x}^{(n,s)}_{\setminus u}\right)^2 + \lambda_1\sum_{n=1}^{N} \|\boldsymbol\theta^{(n)}_{\setminus u}\|_1 + \lambda_2 \sum_{n=2}^{N} \|\boldsymbol\theta^{(n)}_{\setminus u} - \boldsymbol\theta^{(\pi(n))}_{\setminus u}\|_1\right)$$

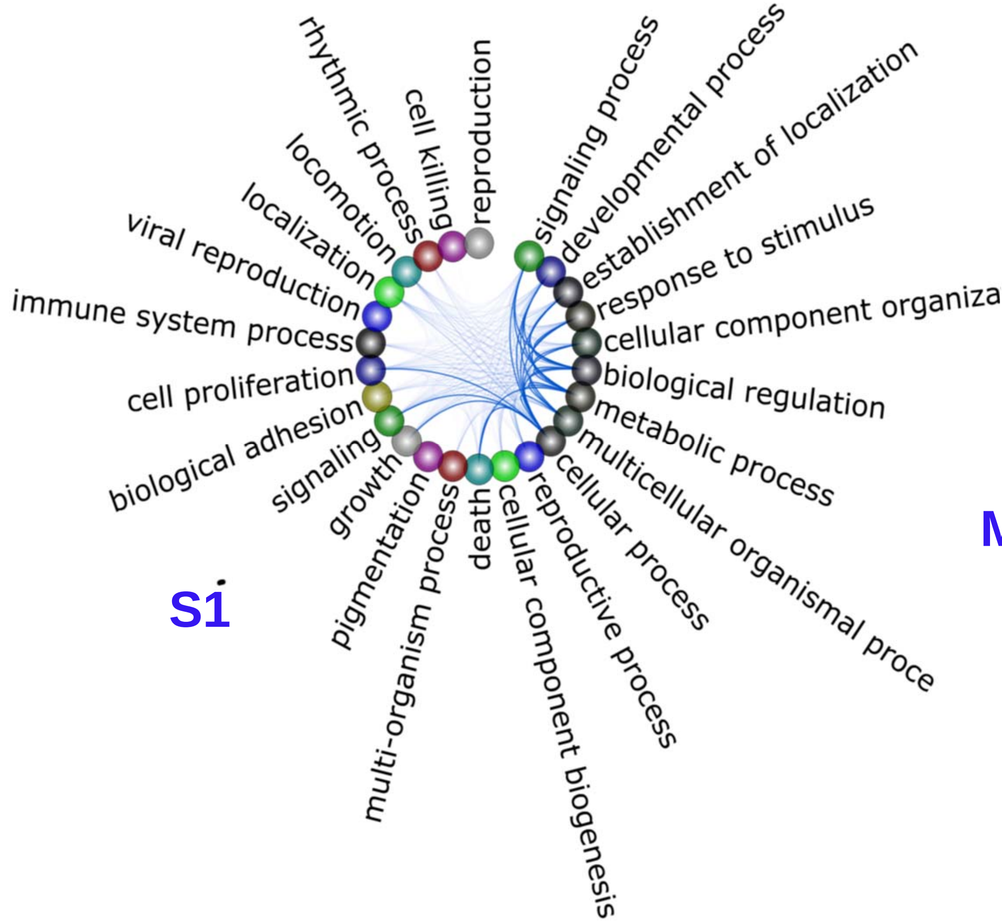
sparsity

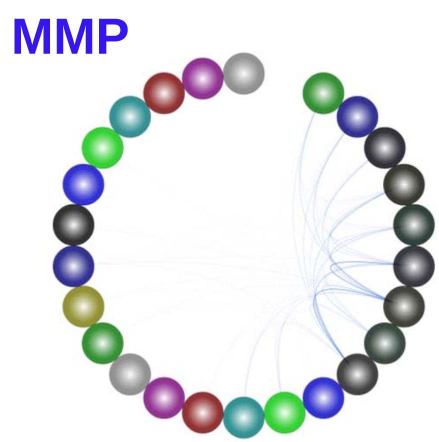
Sparsity of difference

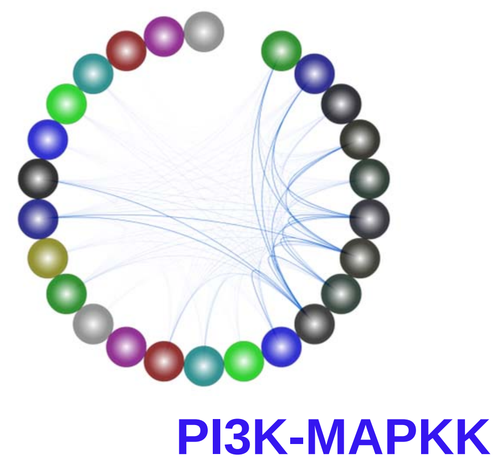
# Network Overview



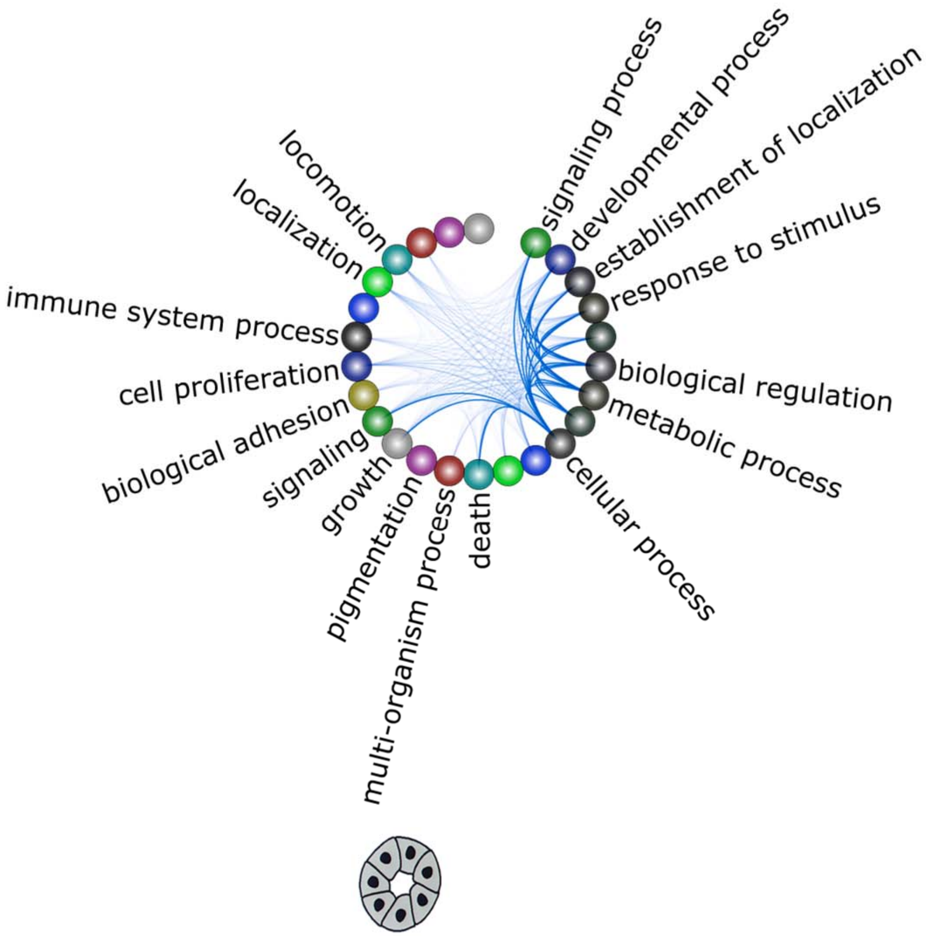



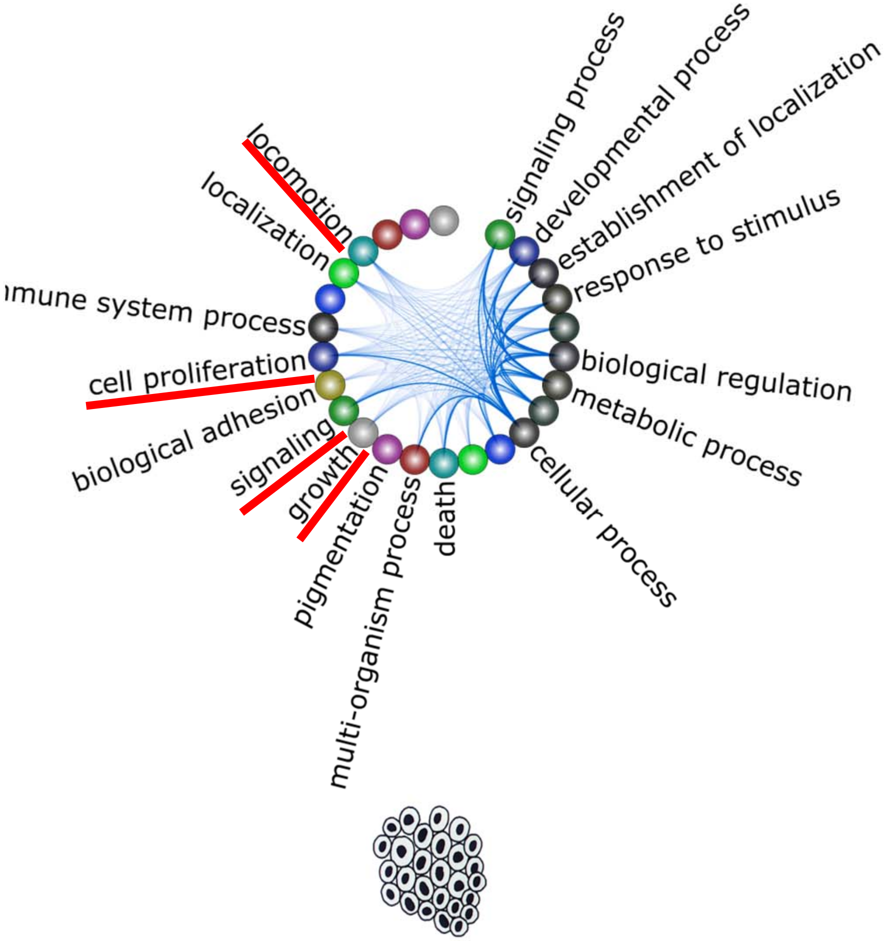



# Interactions – Biological Processes

**S1 cells**

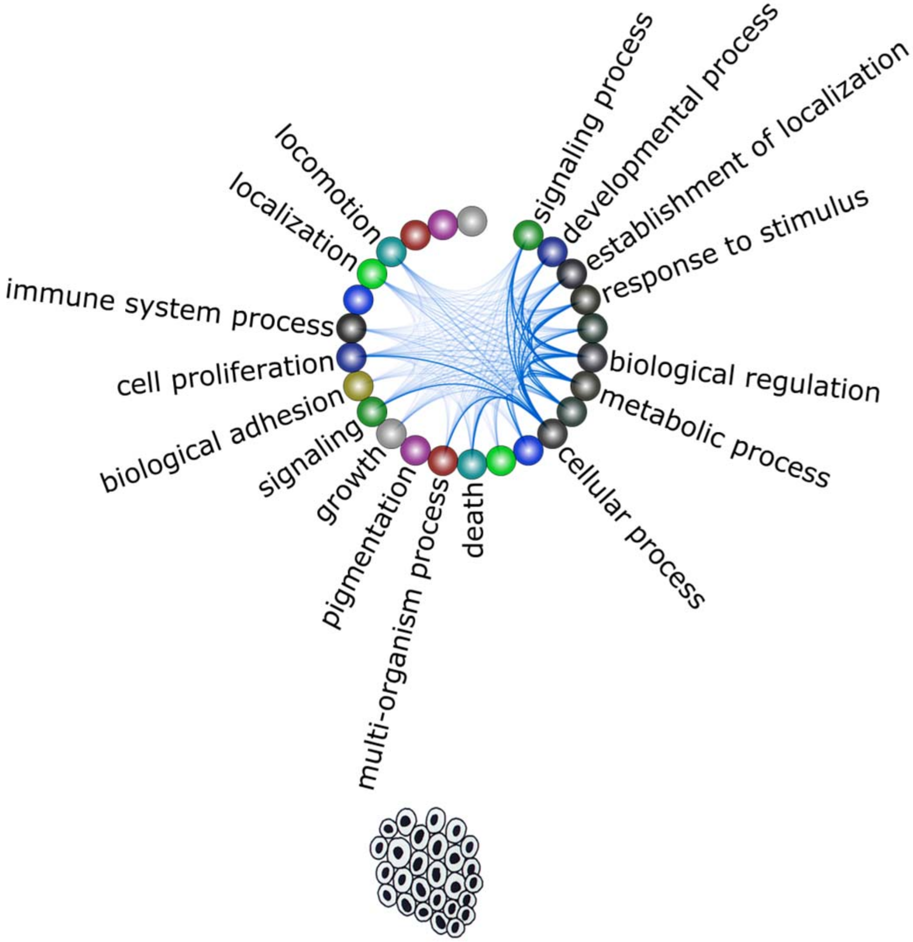



**T4 cells: Increased Cell Proliferation, Growth, Signaling, Locomotion**

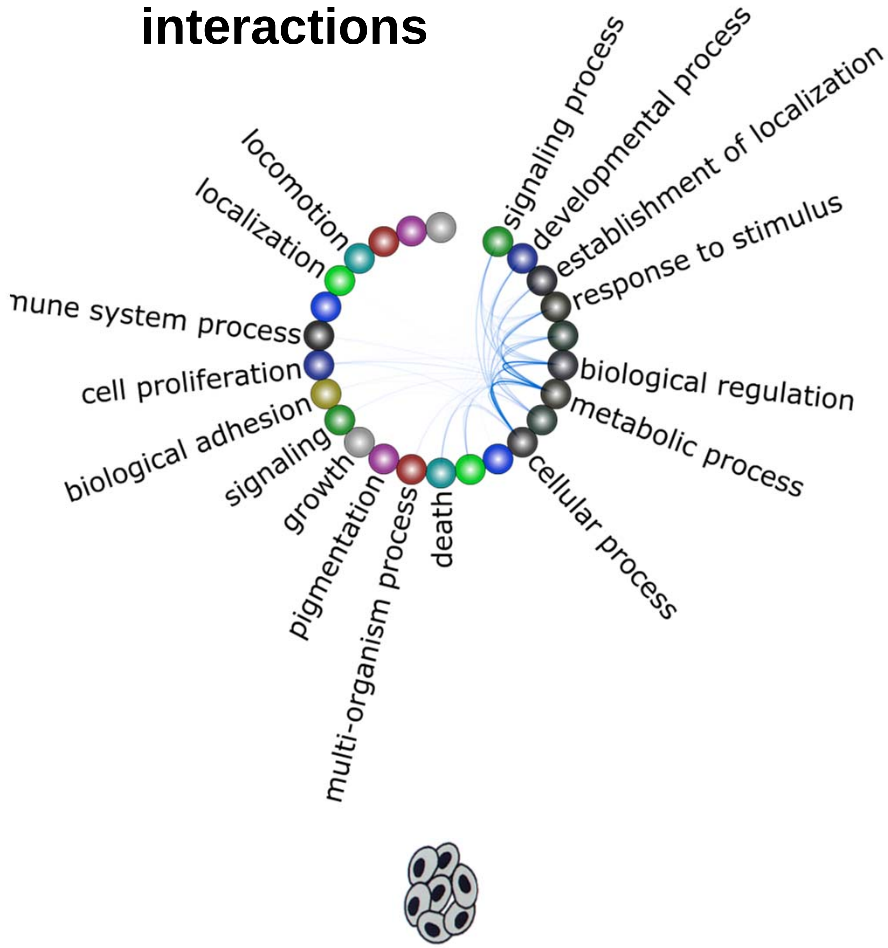


# Interactions – Biological Processes

**T4 cells**

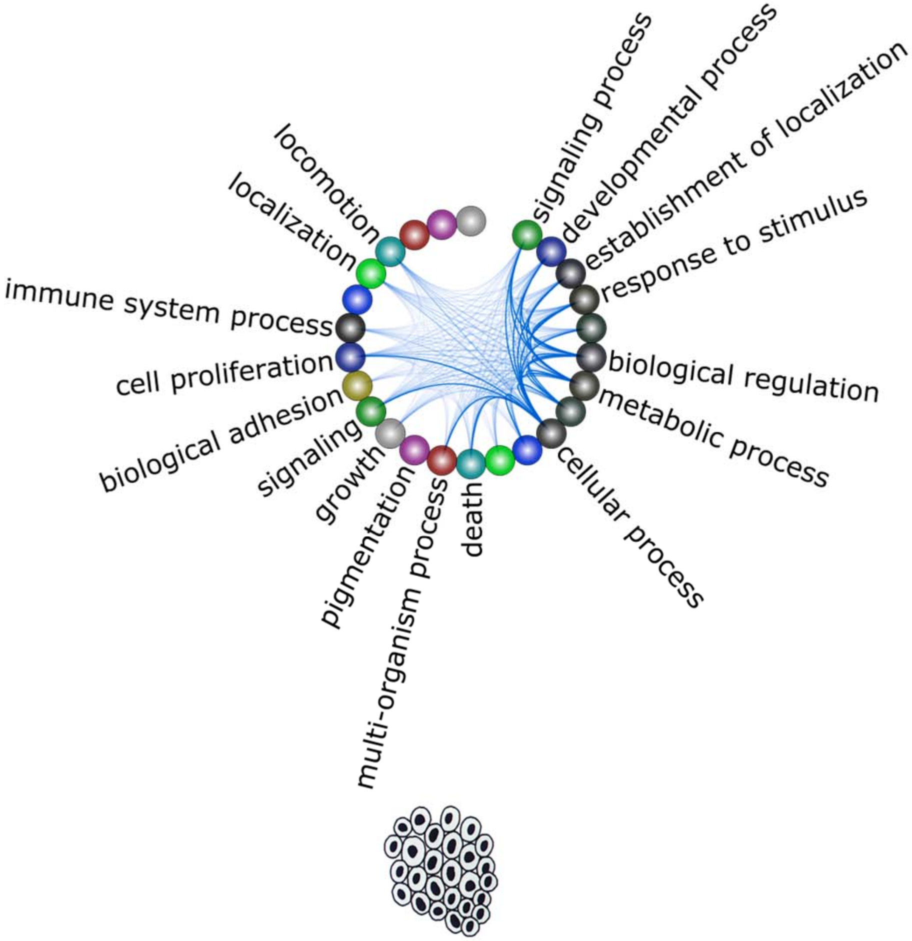



**MMP-T4R cells: Significantly reduced interactions**

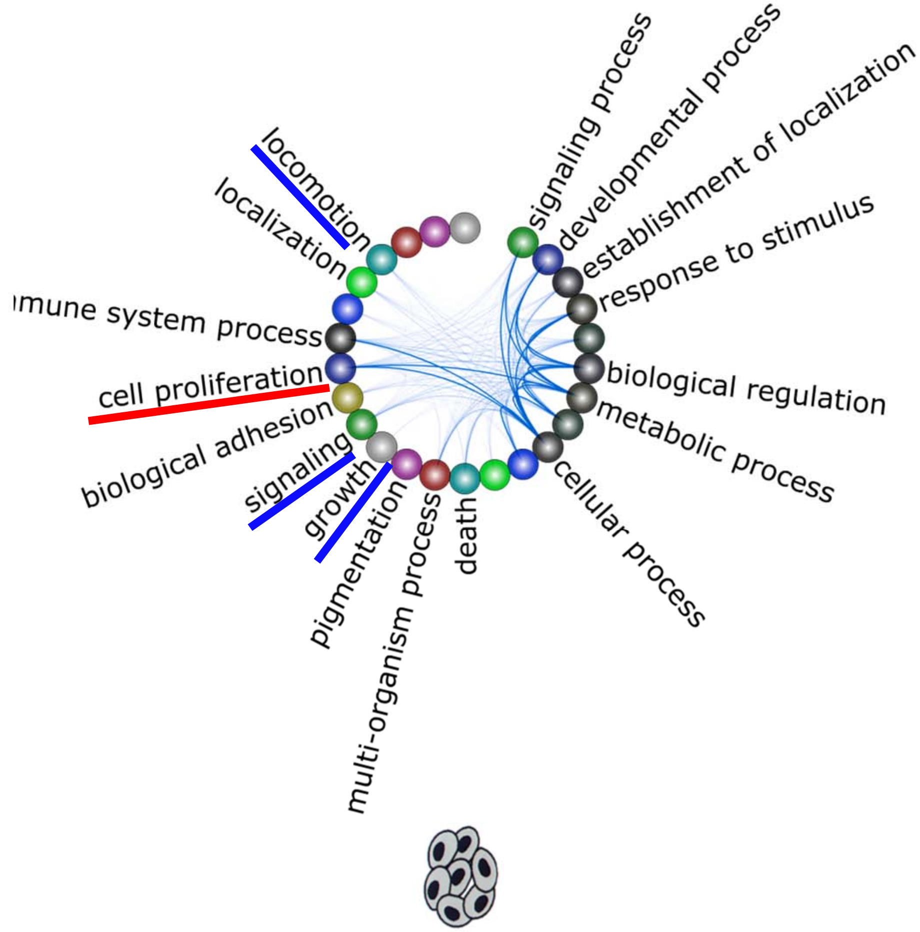


# Interactions – Biological Processes

**T4 cells**



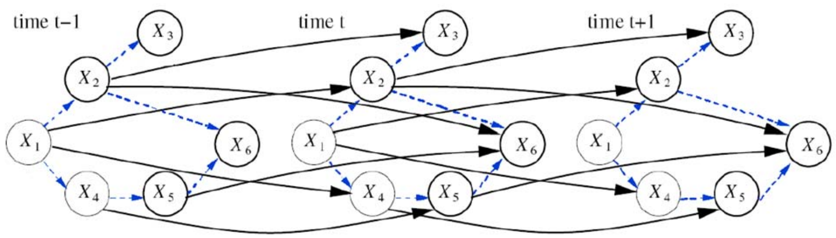

**PI3K-MAPKK-T4R: Reduced Growth, Locomotion and Signaling**



# Fancier network est. scenarios

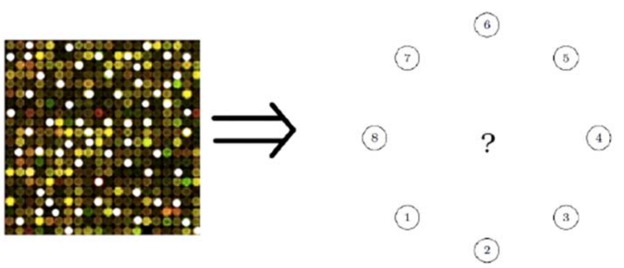
- Dynamic Directed (auto-regressive) Networks

  **[Song, Kolar and Xing, NIPS 2009]**



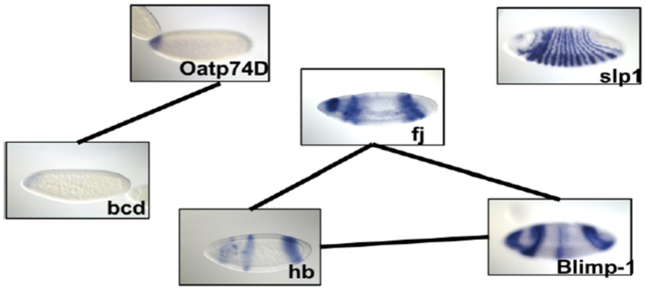

- Missing Data

  **[Kolar and Xing, ICML 2012]**



- Multi-attribute Data

  **[Kolar, Liu and Xing, JMLR 2013]**



# Summary

- Graphical Gaussian Model
  - The precision matrix encode structure
  - Not estimatable when p >> n

- Neighborhood selection:
  - Conditional dist under GGM/MRF
  - Graphical lasso
  - Sparsistency

- Time-varying Markov networks
  - Kernel reweighting est.
  - Total variation est.