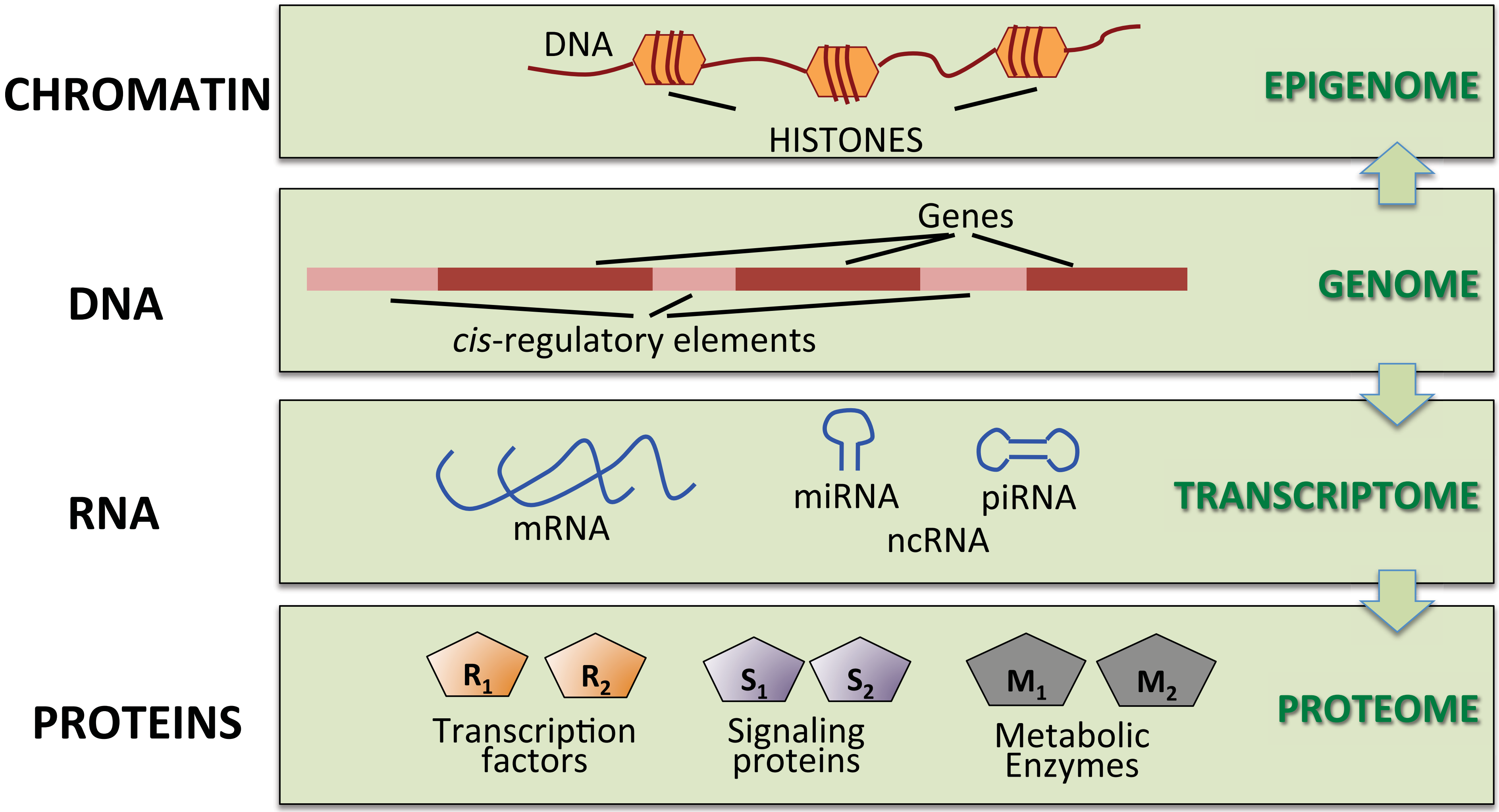
Computational Biology: Genomes, Networks, Evolution
6.047/6.878 Lecture 09

Regulatory networks: Inference, Analysis and Applications

Soheil Feizi

The multi-layered organization of information in living systems



CHROMATIN

DNA

HISTONES

EPIGENOME

DNA

Genes

cis-regulatory elements

GENOME

RNA

mRNA

miRNA

piRNA

ncRNA

TRANSCRIPTOME

PROTEINS

R_1 R_2

Transcription factors

S_1 S_2

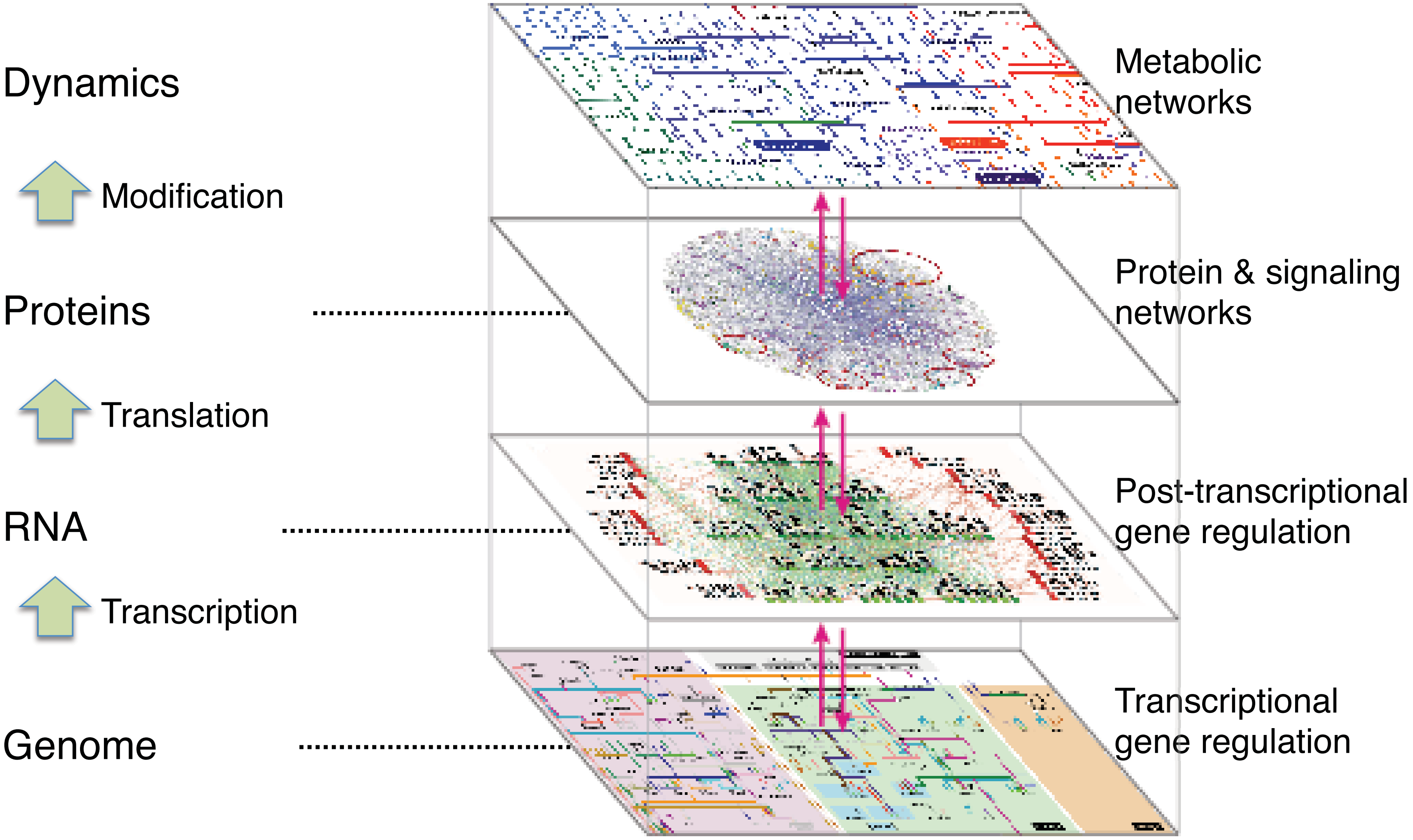
Signaling proteins

M_1 M_2

Metabolic Enzymes

PROTEOME

Biological networks at all cellular levels



Dynamics

Modification

Proteins

Translation

RNA

Transcription

Genome

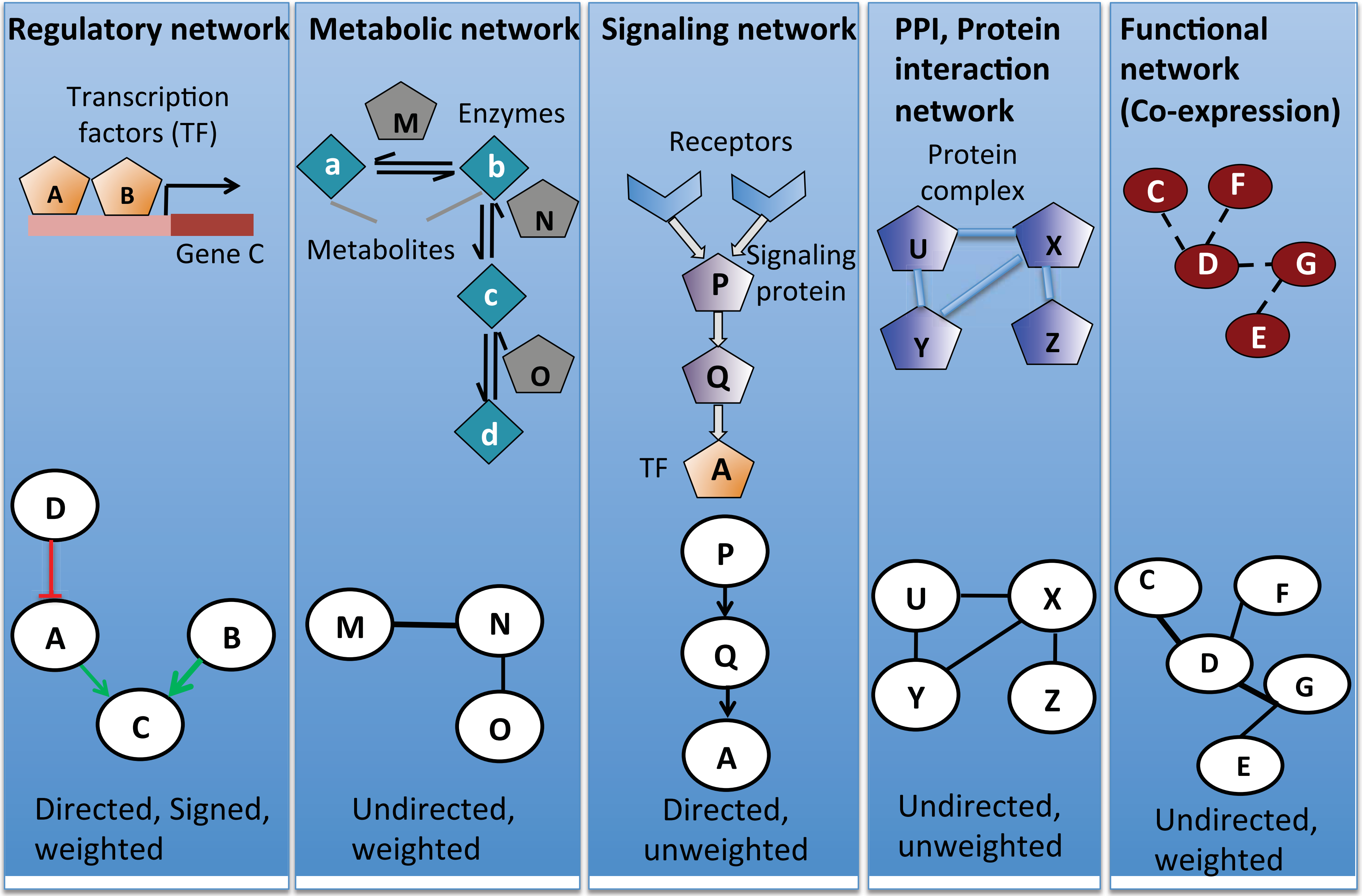
Metabolic networks

Protein & signaling networks

Post-transcriptional gene regulation

Transcriptional gene regulation

Five major types of biological networks



Regulatory network

Directed, Signed, weighted

Metabolic network

Undirected, weighted

Signaling network

Directed, unweighted

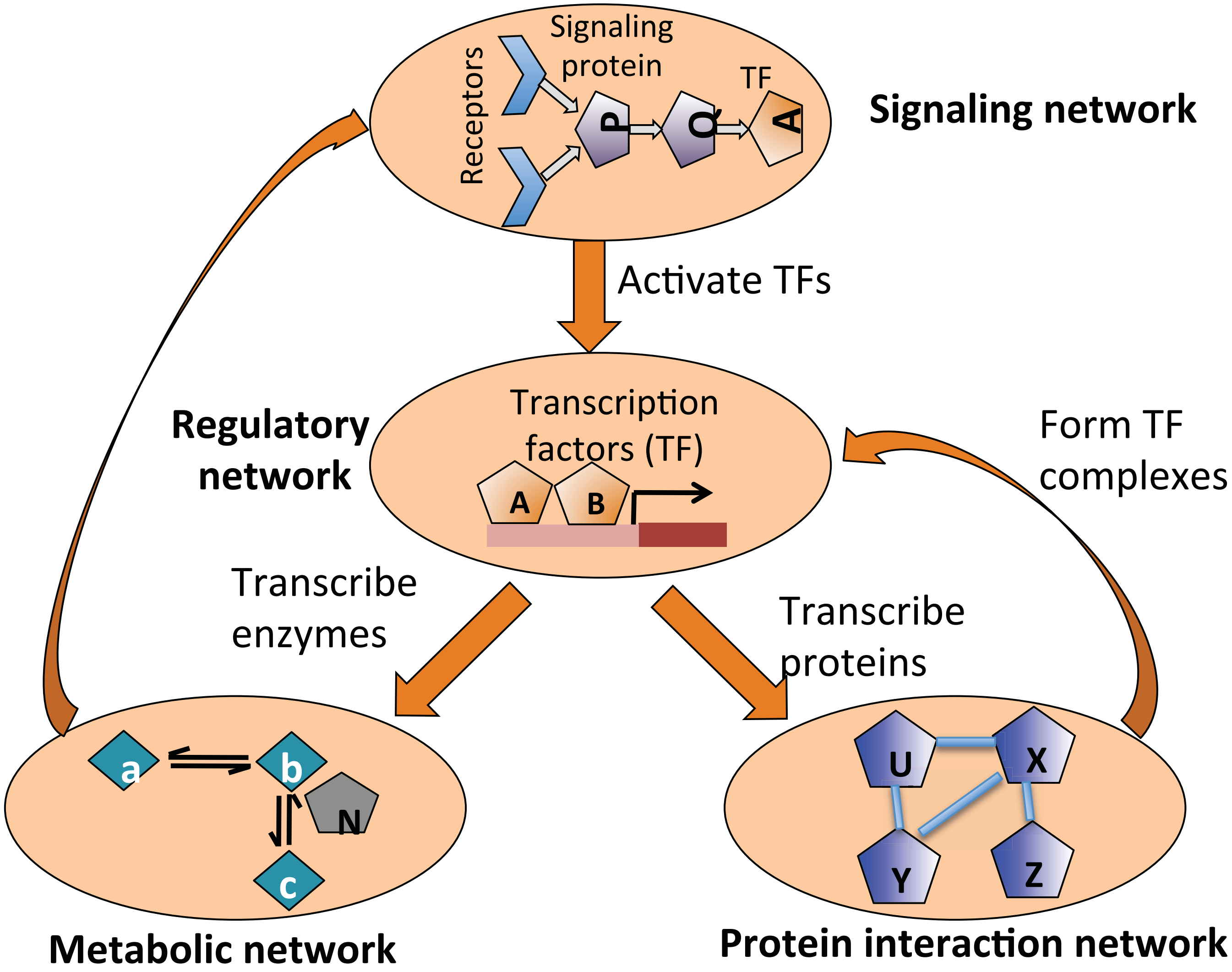
PPI, Protein interaction network

Undirected, unweighted

Functional network (Co-expression)

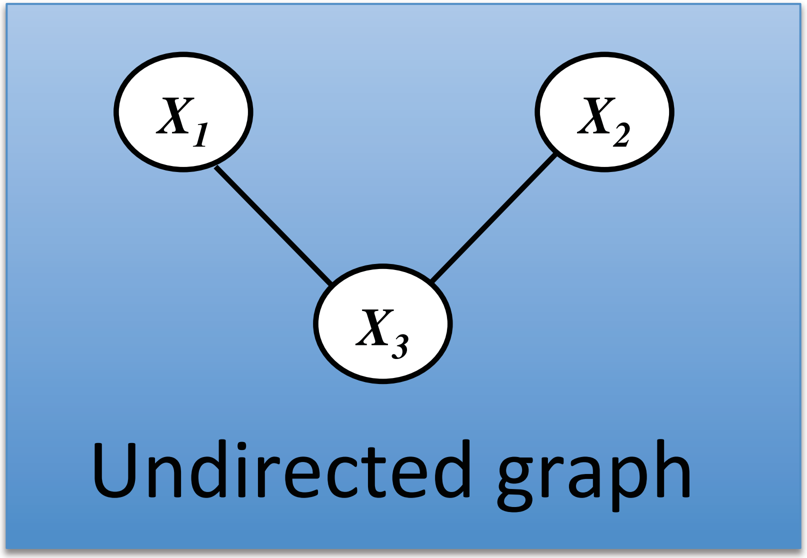
Undirected, weighted

Information exchange across networks

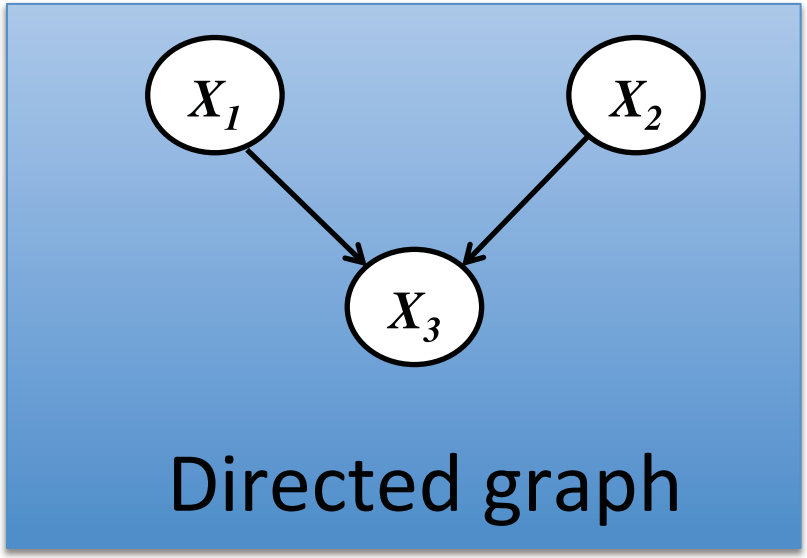


Network definitions: structural, probabilistic

- Two types of binary graphs: directed/undirected networks



Undirected graph

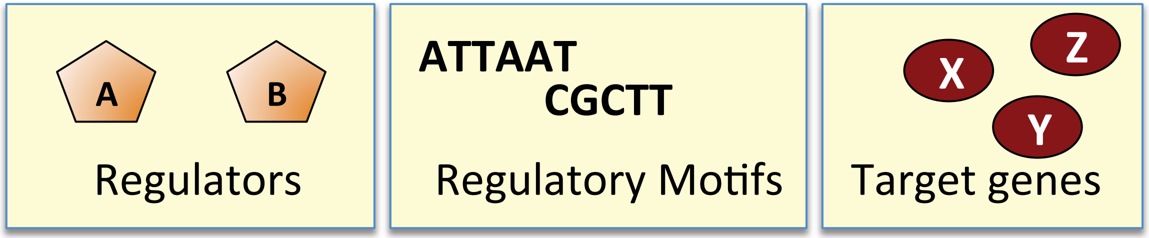


Directed graph

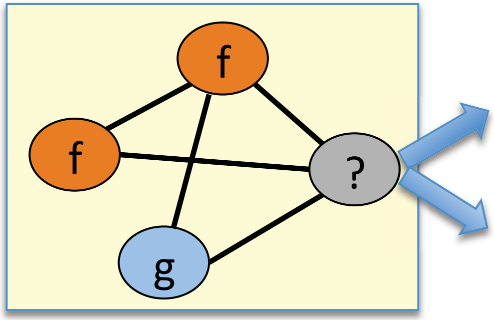
- Graph theory: Nodes, edges, weights, paths
- Probabilistically: Bayesian Networks
  - A model to represent “dependencies” among variables
  - Unconnected nodes are conditionally independent
- Linear algebra: Matrices, powers, decomposition

Network applications and challenges

1 **Element Identification (motif finding lecture)**



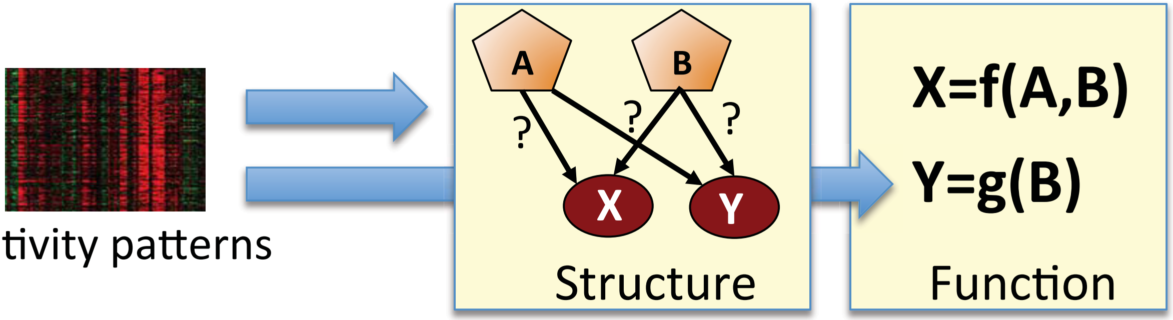
2 **Using networks to predict cellular activity**



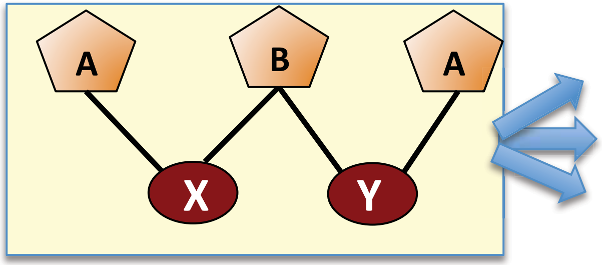
Predict expression levels

Predict gene ontology (GO) functional annotation terms

3 **Inferring networks from functional data**



4 **Network Structure Analysis**



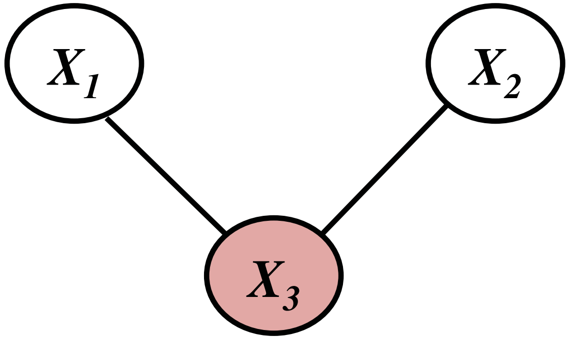
Hubs (degree-distribution)
Network motifs
Functional modules

Goals for today: Network analysis

1. Introduction to networks
2. Applications of regulatory networks
   - Predicting expression of target genes: graphical models, linear regression and regression trees
   - Predicting functions of un-annotated genes, guilt by association
3. Inferring "structure" of regulatory networks
   - Likelihood approach, challenges
   - Simplified approaches and their problems
   - Integrated approaches
4. Structural properties of regulatory networks
   - Scale free degree distribution
   - Network motifs
   - Spectral clustering and modular networks

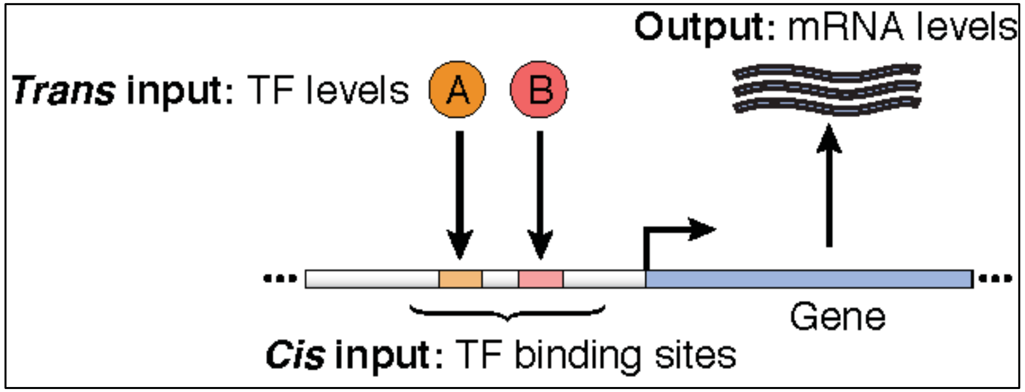
Applications of regulatory networks

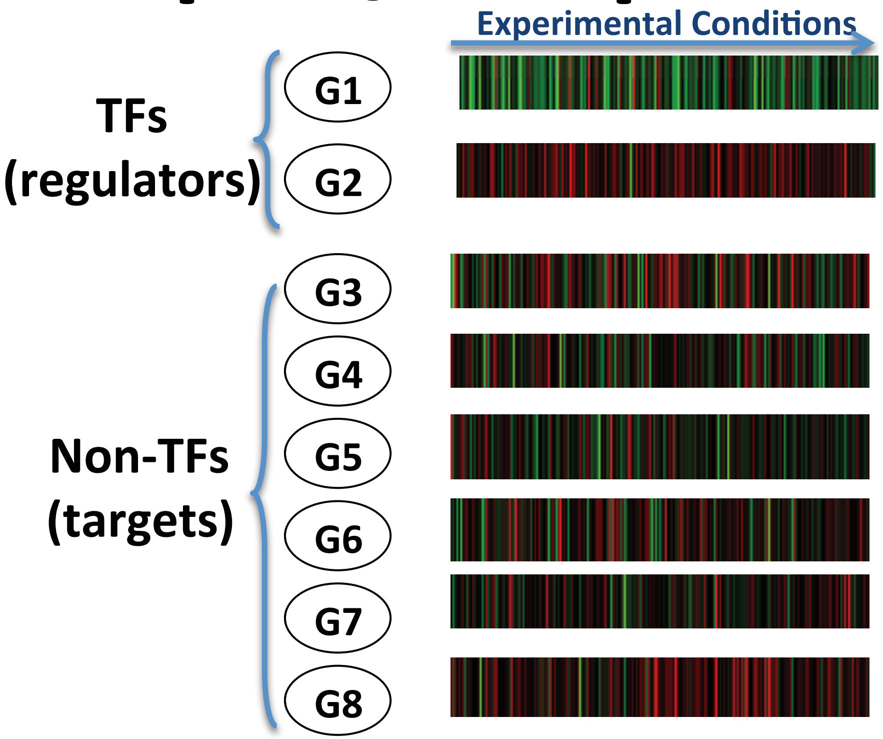
- Predicting expression of targets from expression of regulators
- Predicting function of un-annotated genes based on co-expression and co-regulation



Gene expression prediction

Regulatory network: Input / output

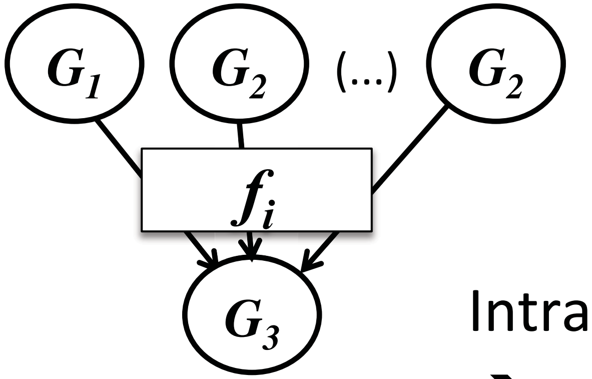




- Gene expression prediction:

$$G_i = f_i(G_j)$$

$$j \in \{1, \ldots, n\} - i$$

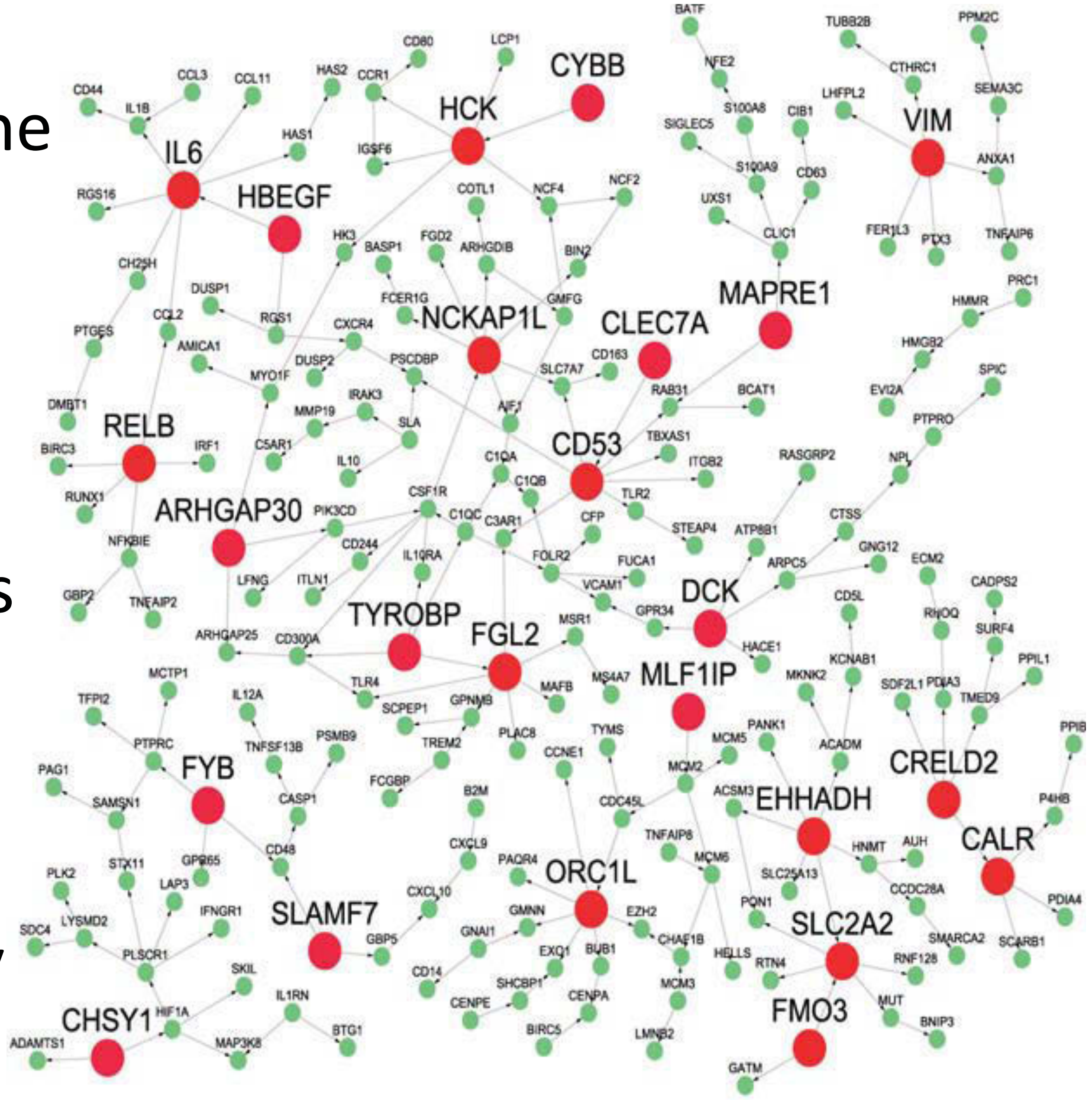


Intractable to compute joint distribution

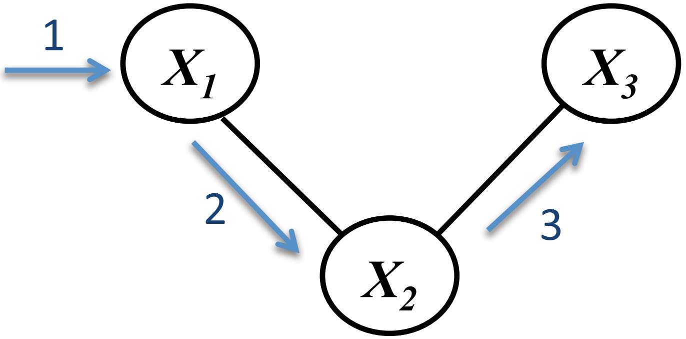
➔ Focus on marginal distributions.

Very large number of regulators / targets

- Regulatory network limits the number of possible hypotheses
- Only directly related elements are connected
- Assume other pairs of nodes are conditionally independent



Graphs represent variable dependencies



- X_1 and X_2 are dependent.
- X_2 and X_3 are dependent.

- X_1 and X_3 are ***conditionally*** independent
  - ***If*** we know the value of X_2, they are independent
  - But if the value of X_2 is ***not*** known, then:

    1. Observing (or estimating) value of X1 ….
    2. … can influence our estimate of the value of X2…
    3. … which in turn can influence our estimate of value of X3

→ **Some information does flow X_1→X_3 through X_2: Dependent!**

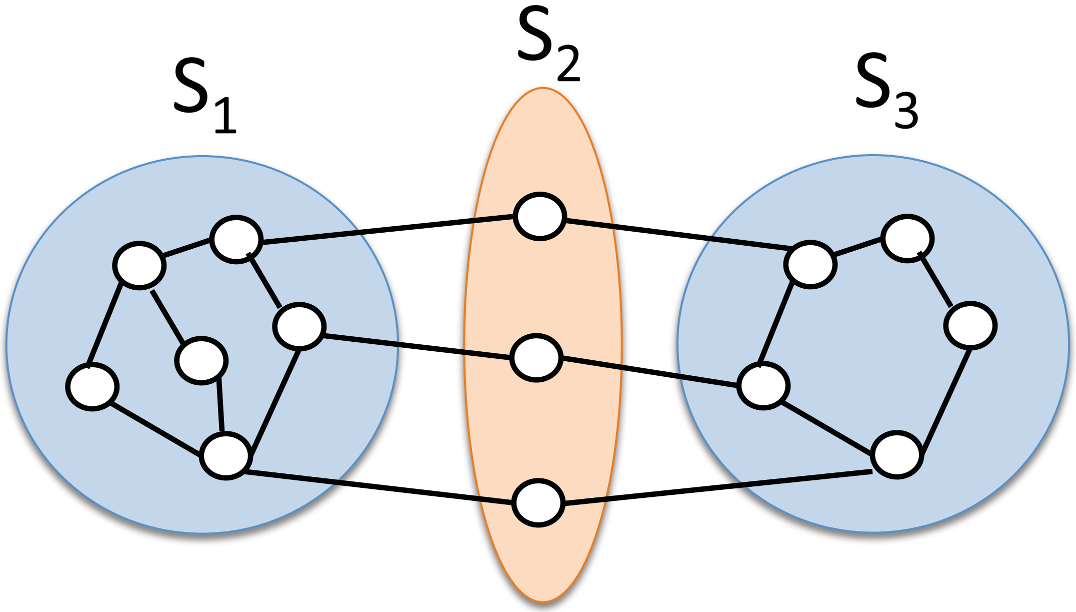
X_1 and X_3 are independent ***given*** X_2 : $X_1 \perp\!\!\!\perp X_3 | X_2$

Probability tables vs. graphical models

- Equations

- Network

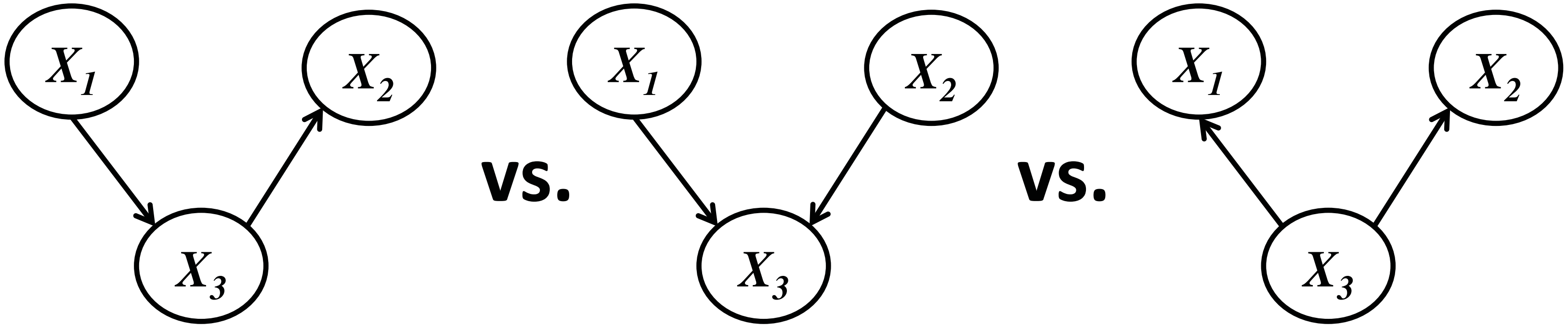
Network structure →sets of ind. variables



$$X_{S_1} \perp\!\!\!\perp X_{S_3} | X_{S_2}$$

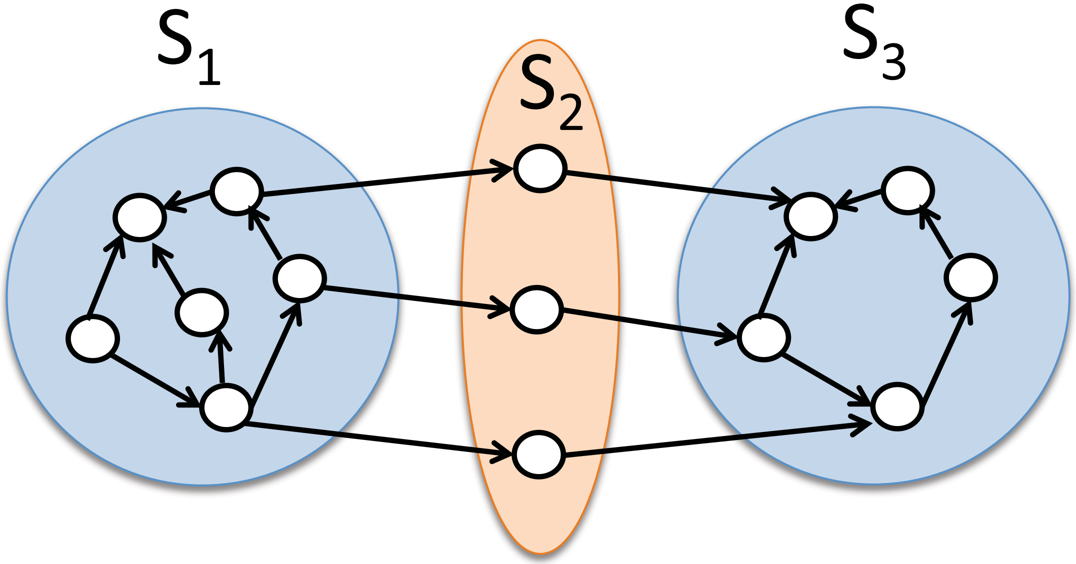
- Variables S_1={…} and S_3={…} are ***conditionally independent*** given S_2, if they become disconnected by removing S_2
- Graphical models represent "structure" of joint probability distribution: reason about **graph**, instead of reasoning about **probability tables**

Directed graphs ➔ Asymetry of conditional ind



$X_1 \perp\!\!\!\perp X_2 | X_3$ $X_1 \perp\!\!\!\perp X_2 | X_3$ $X_1 \perp\!\!\!\perp X_2 | X_3$

- Parent nodes vs. children nodes [EXPLAIN]



$$X_{S_1} \perp\!\!\!\perp X_{S_3} | X_{S_2}$$

- Given parents: children nodes independent from

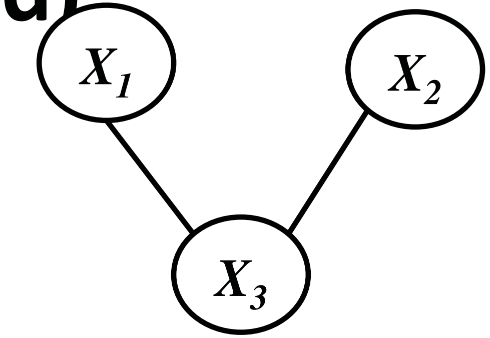
Rules for conditional independence

Joint distribution → node/edge potentials (Markov Random Field)

Bayes' rule

$$P(X_1, X_2, X_3) = P(X_1, X_2 | X_3) P(X_3)$$

$$= P(X_1 | X_3) P(X_2 | X_3) P(X_3)$$



- **Conditionally independent variables appear in separate terms**

Node potential **Edge potential**

$$P(X_1, X_2, \dots, X_n) = \frac{1}{Z} \prod_{i \in V} \phi(X_i) \prod_{(i,j) \in E} \Psi(X_i, X_j)$$

partition function (typically cancels out)

For every network edge

X1	X2	X3	X4
F(.)	F(.)	F(.)	F(.)

	X1	X2	X3	X4
X1	F(.)	F(.)	F(.)	F(.)
X2	F(.)	F(.)	F(.)	F(.)
X3	F(.)	F(.)	F(.)	F(.)
X4	F(.)	F(.)	F(.)	F(.)

What about the function F(.)?

Predicting gene expression

Edge potential functions

Gaussian functions

Linear regression

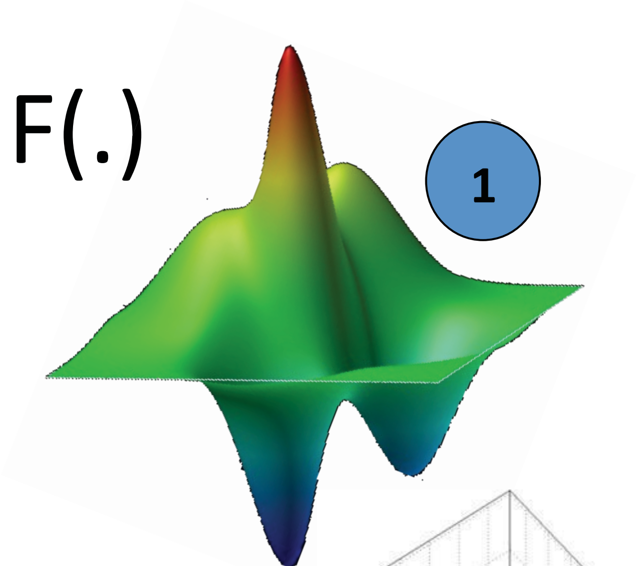
Regression trees

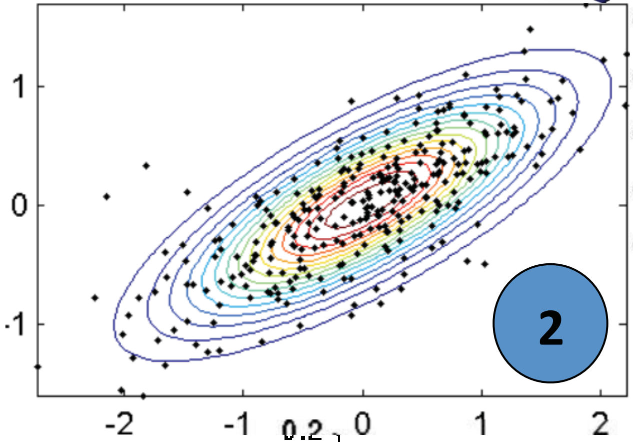
Types of potential functions F(.)

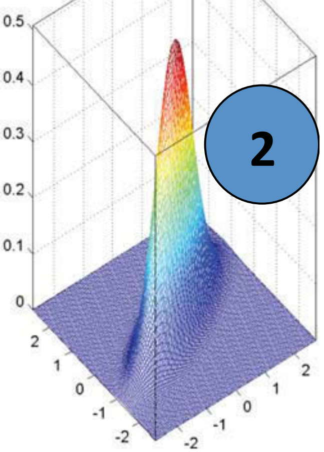
- General

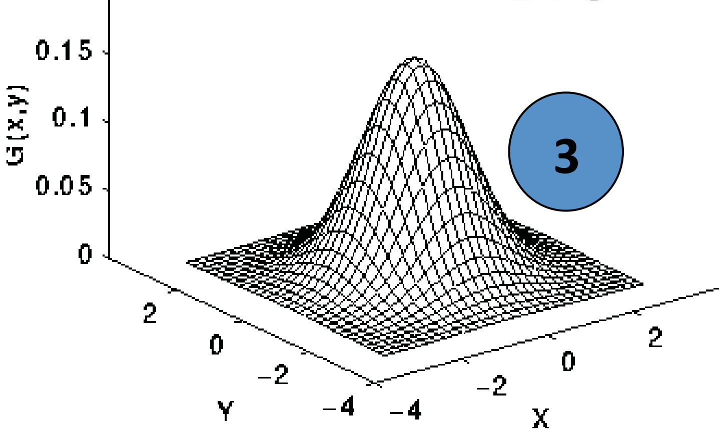
- Exponential functions

- **Gaussian functions**
  - General covariance
  - Unit variance,
    only correlations ρ
  - No covariance (indpent)





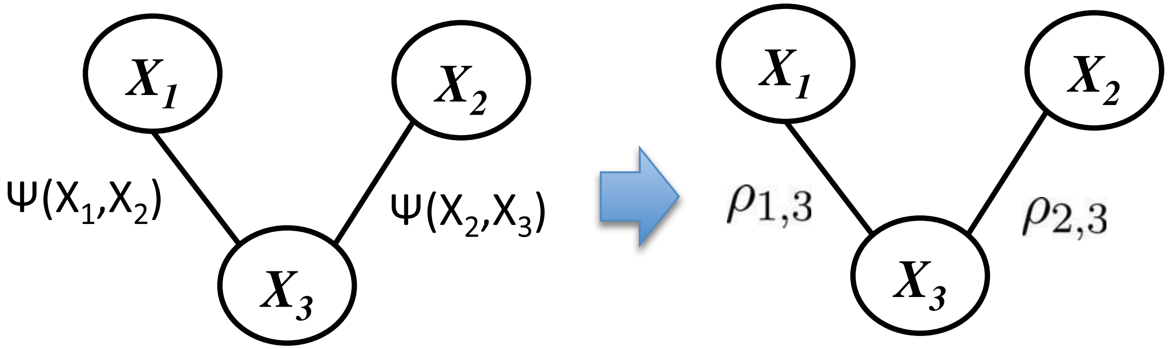




Gaussian edge potential functions
(Gaussian graphical models)

- If X_1, X_2 and X_3 are jointly Gaussian with $\mu=0$ and $\sigma=1$

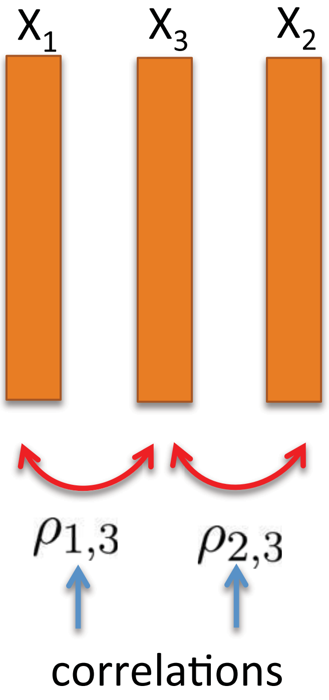
➔ edge potential functions simplify to correlations $\rho_{i,j}$



$$P(X_1, X_2, X_3)\alpha$$

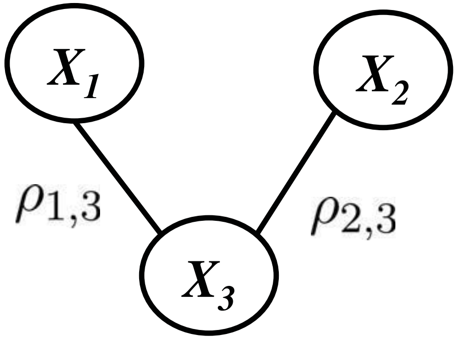
$$\phi(X_1)\phi(X_2)\phi(X_3)\Psi(X_1, X_3)\Psi(X_2, X_3)$$

$$e^{\frac{-1}{2}x_1^2}\ e^{\frac{-1}{2}x_2^2}\ e^{\frac{-1}{2}x_3^2}\quad e^{\rho_{1,3}x_1x_3}\quad e^{\rho_{2,3}x_2x_3}$$



Prediction problem ➔ calculate marginals

$$P(X_1|X_3) = \frac{P(X_1, X_3)}{P(X_3)} = \frac{\int P(x_1, x_2, x_3)dx_2}{\int P(x_1, x_2, x_3)dx_2dx_3}$$



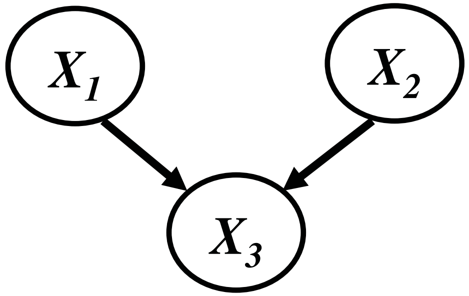
One more expansion, showing Z

- Normalization term (Z) will be canceled out!

$$P(X_1, X_2, \ldots, X_n) = \frac{1}{Z}\prod_{i\in V}\phi(X_i) \prod_{(i,j)\in E} \Psi(X_i, X_j)$$

Assume linear function from regulators to target (Linear regression)

- Goal: $X_3 = f(X_1, X_2)$
- Probabilistic approach: $P(X_3|X_1, X_2)$



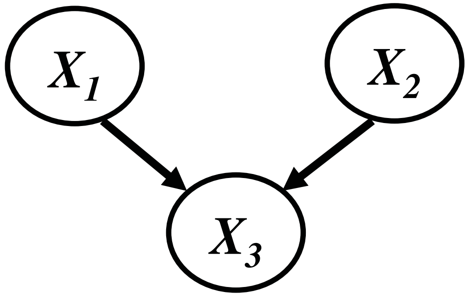
- Assume expression of a target is Gaussian whose mean is a linear combination of the expression level of regulators

$$P(X_3|X_1, X_2) \sim N(\underset{\uparrow}{\alpha_1} X_1 + \underset{\uparrow}{\alpha_2} X_2 + \underset{\uparrow}{\alpha_0}, 1)$$

- Use maximum likelihood to find parameters.

Predicting gene expressions using linear regression (combine with prev)

$$P(X_3|X_1, X_2) \sim N(\alpha_1 X_1 + \alpha_2 X_2 + \alpha_0, 1)$$



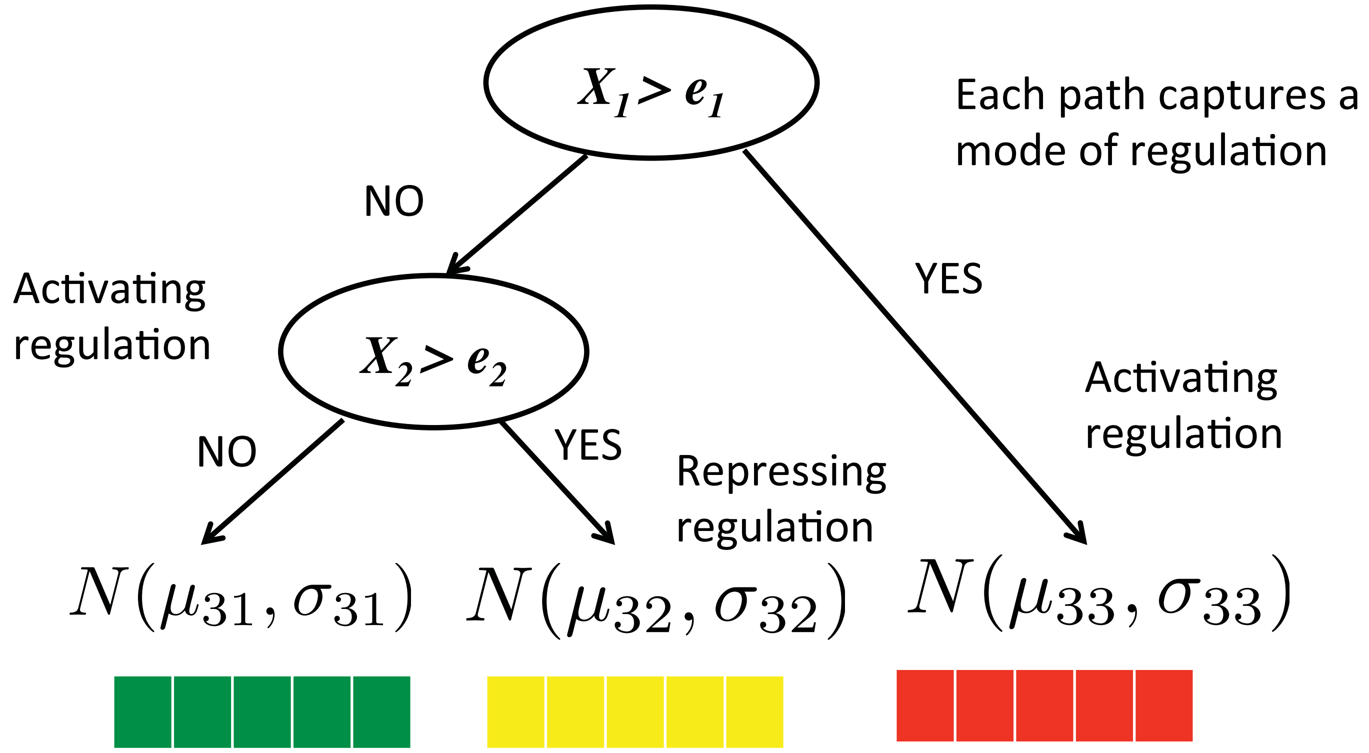
Data: D

Model: M

$$P(M|D) = \frac{P(M)P(D|M)}{P(D)}$$

- Take derivatives to find optimal model parameters
- Problem of over-fitting => regularization (DETAILS on regularization functions)

Predicting expression using regression trees



$X_1 > e_1$
Each path captures a mode of regulation
NO
YES
Activating regulation
$X_2 > e_2$
NO
YES
Repressing regulation
Activating regulation
$N(\mu_{31}, \sigma_{31})$
$N(\mu_{32}, \sigma_{32})$
$N(\mu_{33}, \sigma_{33})$

Expression of target modeled using Gaussians at each leaf node

- Assumes variables are continuous. Arranges regulators in a tree
- Expression prediction follows a set of decision rules
  → Can model combinatorics
- Allows non-linear dependencies between regulators and target
- Targets can share regulatory programs

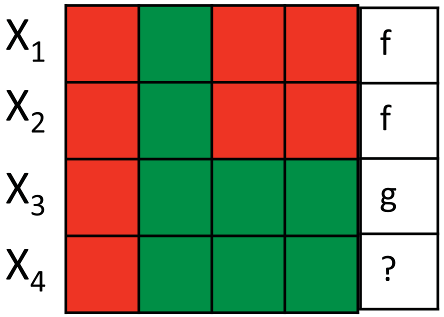
Predicting gene function

Guilt by association

Predicting functions of un-annotated genes

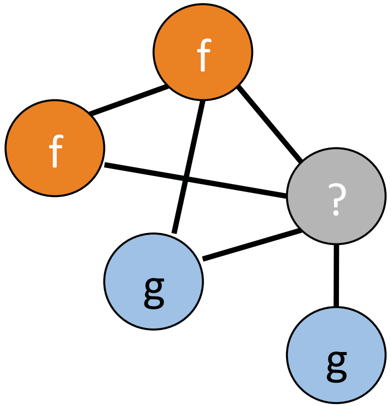
- Goal: Predict function of unannotated genes based on "guilt by association"
- Different types of "association"

Co-expression

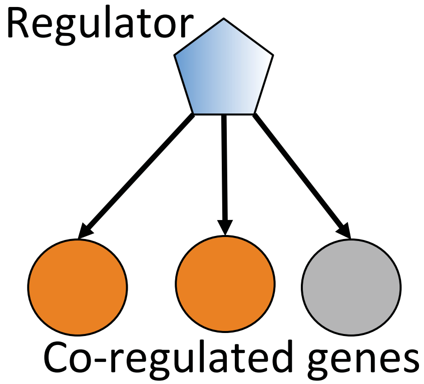


X_1	🟥	🟩	🟥	🟥	f
X_2	🟥	🟩	🟥	🟥	f
X_3	🟥	🟩	🟩	🟩	g
X_4	🟥	🟩	🟩	🟩	?

Protein-interactions



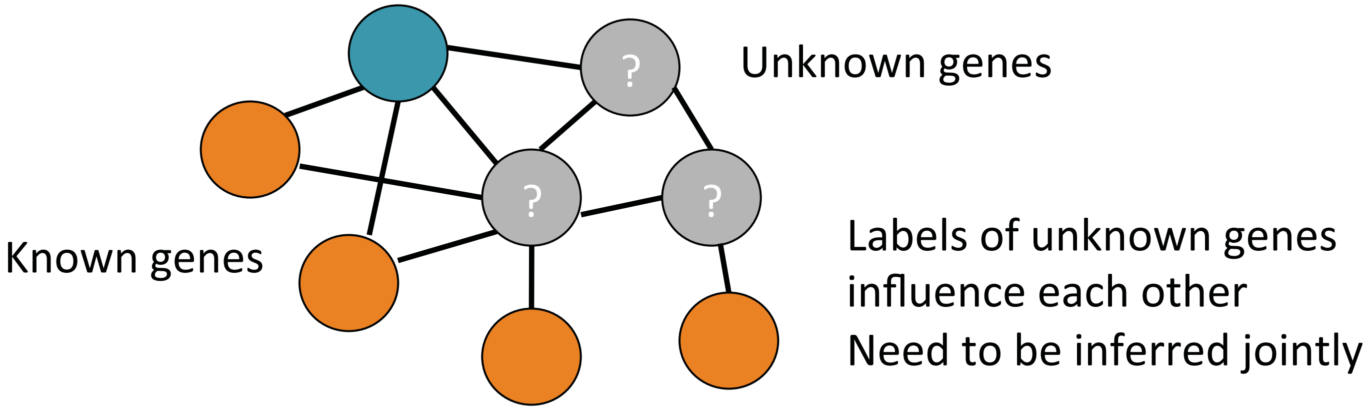
Co-regulation



However most approaches work with "functional networks"

Deng et al 03, Sharan et al 07

Iterative classification algorithm



- Start with an initial assignment of labels
- Repeat iteratively
  - Update relational attributes
  - Re-infer the labels

Neville 03, Getoor 05

Approaches for "network-based" function prediction

- Neighborhood counting
  - Add sentence
- Markov Random Field Structure
  - Add sentence
- Relaxation Labeling
  - Add sentence
- Collective classification
  - Add sentence
- Most approaches work with functional networks
  - Add sentence

Take away messages so far …
(combine with outline slide)

- Use graphical models to represent "dependencies" among variables
- Gene expression predictions are equivalent to finding marginal distributions
  - Linear regression, regression trees
- Use network structure to predict functions of un-annotated genes

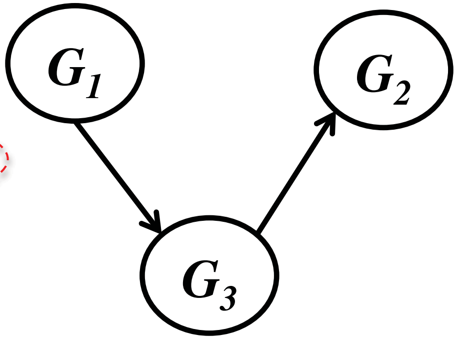
Goals for today: Network analysis

1. Introduction to networks
2. Applications of regulatory networks
   - Predicting expression of target genes: graphical models, linear regression and regression trees
   - Predicting functions of un-annotated genes, guilt by association
3. Inferring "structure" of regulatory networks
   - Likelihood approach, challenges
   - Simplified approaches and their problems
   - Integrated approaches
4. Structural properties of regulatory networks
   - Scale free degree distribution
   - Network motifs
   - Spectral clustering and modular networks

Likelihood approach to infer “network structure”

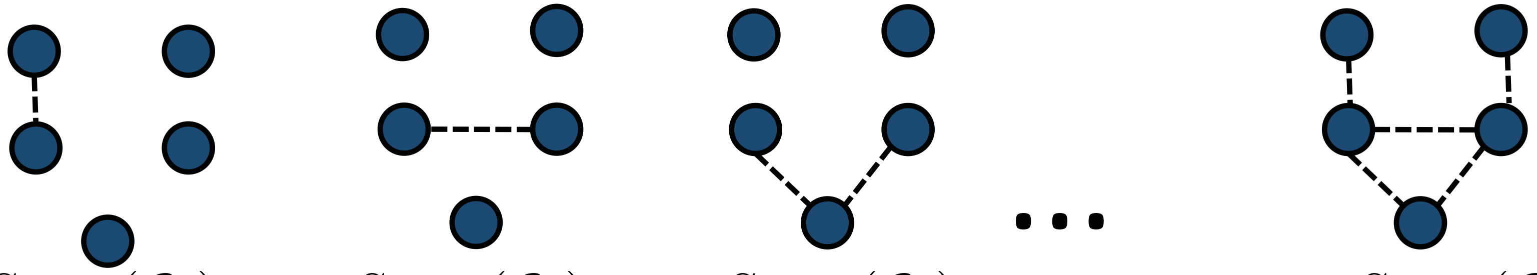
- Likelihood approach:
  - Assign a likelihood score to each structure
  - Pick the best one!

$$P(\text{structure}|\text{data}) = \frac{P(\text{structure})P(\text{data}|\text{structure})}{P(\text{data})}$$



Structure Learning needs search

$$\text{Score}(\mathcal{G}) = \text{Likelihood}(\mathbf{X}; \mathcal{G}, \theta) = P(\mathbf{X}|\theta, \mathcal{G})$$



$\text{Score}(\mathcal{G}_1)$ $\text{Score}(\mathcal{G}_2)$ $\text{Score}(\mathcal{G}_3)$ $\text{Score}(\mathcal{G}_n)$

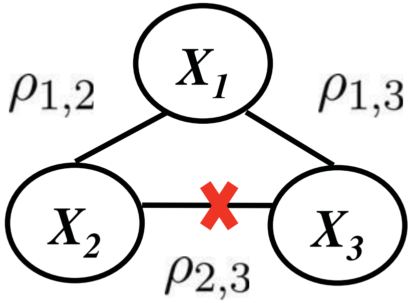
$$\underset{\substack{\uparrow \\ \text{Best graph}}}{\widehat{\mathcal{G}}} = \arg\max_{\mathcal{G}} \underbrace{\max_{\theta} P(\mathbf{X}|\theta, \mathcal{G})}_{\text{Maximum likelihood}}$$

Likelihood approach to infer network structure: challenges

- Problems:
  - Exponentially many structures!
  - Unable to discriminate between direct vs indirect links (Undistinguishable structures!)

Solution 1: Correlation-based inference methods

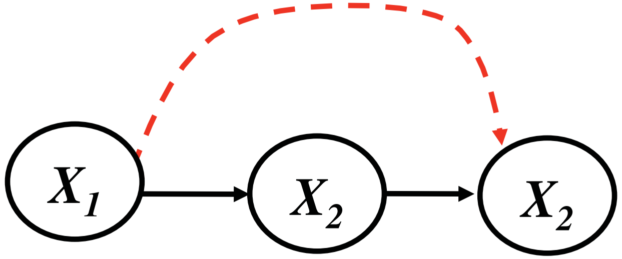
- Only consider structures whose "observed" edge weights are high
- Perform maximum likelihood test among fewer structures



$$\rho_{2,3} < \min(\rho_{1,2}, \rho_{1,3})$$

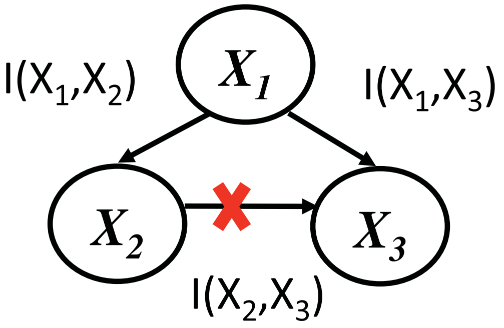
Issues of correlation-based inference methods

- Problems:
  - Many false positive and true negative edges
  - Observed edge weights may be different than true edge weights.
  - Indirect effects and transitive edges:



Indirect information flows cause transitive edges

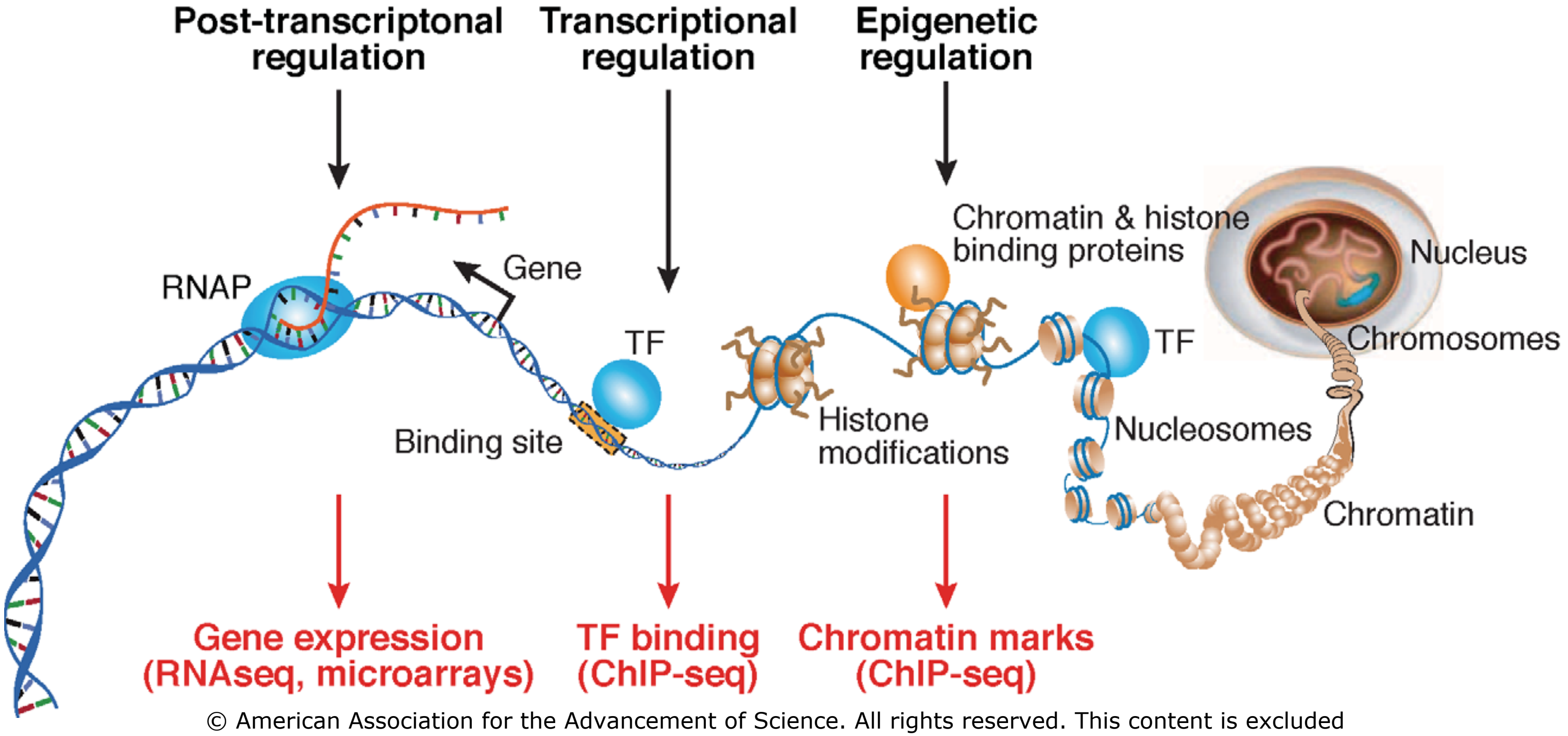
- Transitive edges are due to information flows over indirect paths
- ARACNE solution: Exclude edges with lowest Information in a triplet => information inequality



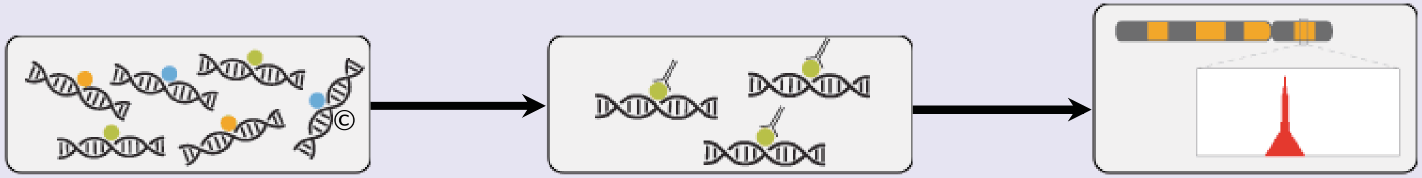
$I(X_2,X_3) < \min(I(X_1,X_2),I(X_1,X_3))$

- Network deconvolution!

Solution 2: Use many data types to infer regulatory networks (chip, motif, chromatin)


Post-transcriptonal regulation
Transcriptional regulation
Epigenetic regulation
RNAP
Gene
TF
Binding site
Chromatin & histone binding proteins
Nucleus
TF
Chromosomes
Histone modifications
Nucleosomes
Chromatin
Gene expression (RNAseq, microarrays)
TF binding (ChIP-seq)
Chromatin marks (ChIP-seq)

Source: Roy, Sushmita et al. "Identification of functional elements and regulatory circuits by Drosophila modENCODE." Science 330, no. 6012 (2010): 1787-1797.



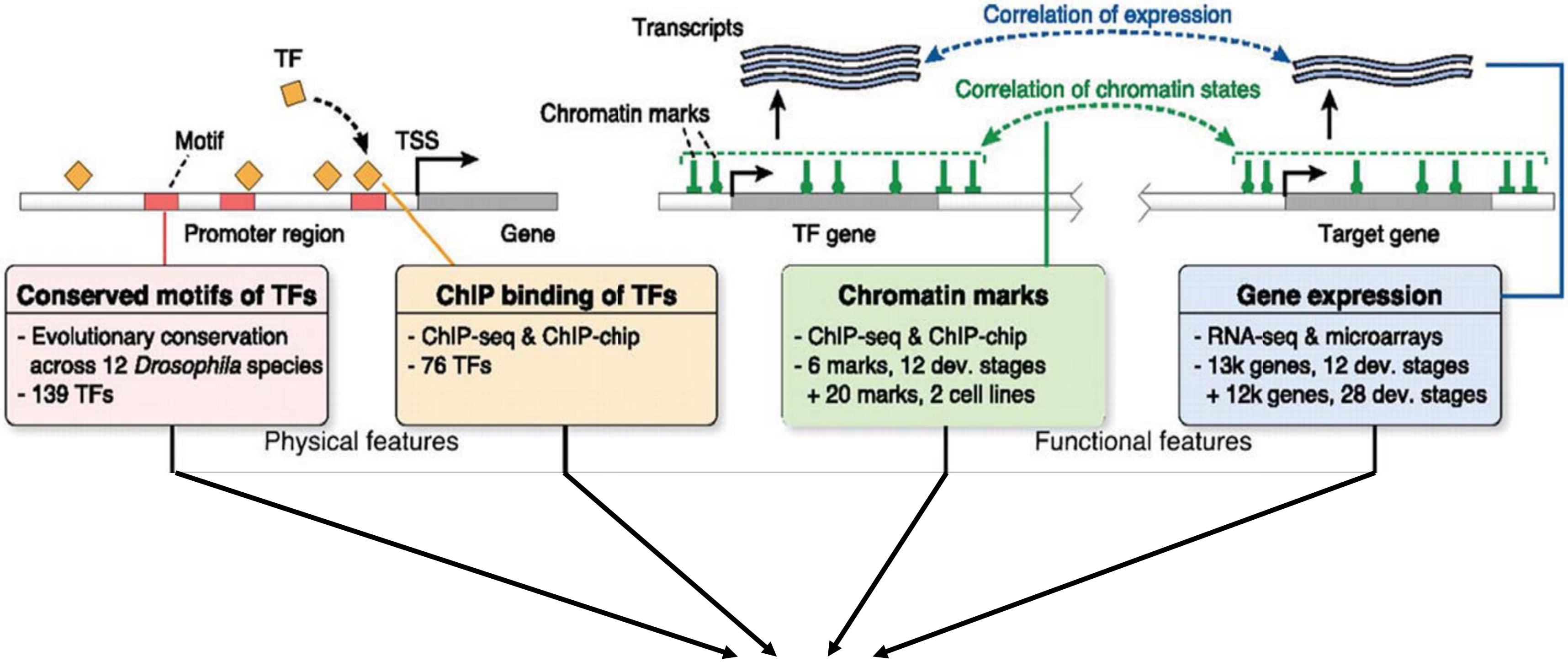
Glue proteins to DNA, cut into pieces

Use antibody to filter for a specific protein

Sequence the pieces, map back to genome

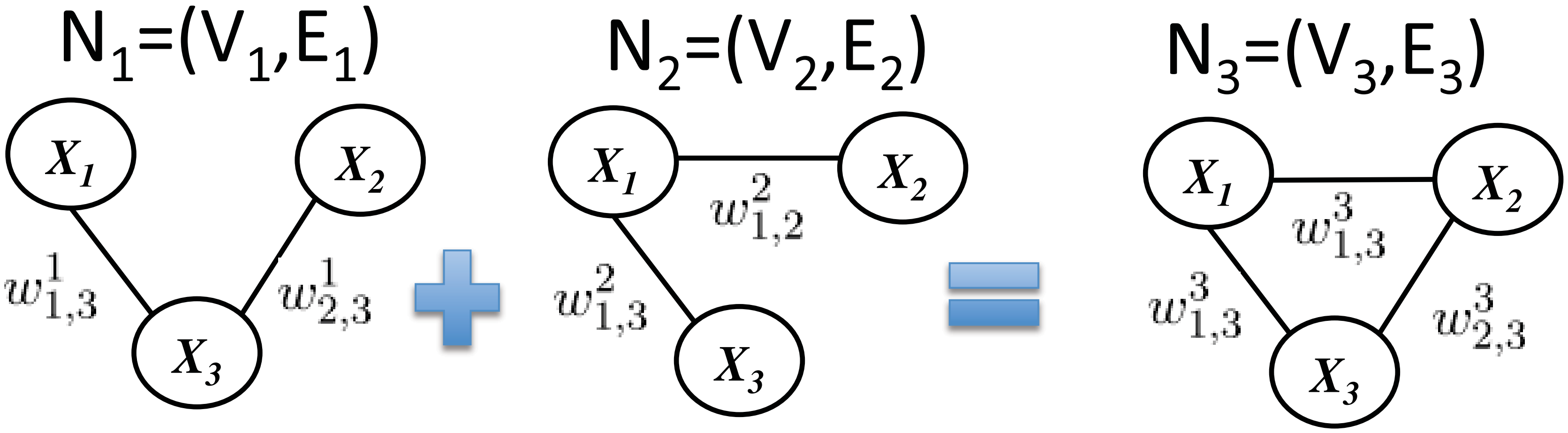
Integrated approach to infer regulatory networks Solution 3: Solution 1+Solution 2

- Combine inferred regulatory networks from many data types



Source: Marbach, Daniel et al. "Predictive regulatory models in Drosophila melanogaster by integrative inference of transcriptional networks." Genome Research 22, no. 7 (2012): 1334-1349.

Network integration: problem setup



- Can we simply add weights?
- Assumptions:
  - Input networks are "independent"
  - Weights represent log-likelihoods

Likelihood approach to integrate weighted networks

$$w^3_{1,3} = \log \frac{P\big((1,3) \in E_3 \big| w^1_{1,3}, w^2_{1,3}\big)}{P\big((1,3) \notin E_3 \big| w^1_{1,3}, w^2_{1,3}\big)}$$

$$= \log \frac{\frac{P\big((1,3) \in E_3\big) P\big(w^1_{1,3}, w^2_{1,3})|(1,3) \in E_3\big)}{P(w^1_{1,3}, w^2_{1,3})}}{\frac{P\big((1,3) \in E_3\big) P\big(w^1_{1,3}, w^2_{1,3})|(1,3) \notin E_3\big)}{P(w^1_{1,3}, w^2_{1,3})}}$$

Bayes' rule

$$= \log \frac{P\big(w^1_{1,3}, w^2_{1,3})|(1,3) \in E_3\big)}{P\big(w^1_{1,3}, w^2_{1,3})|(1,3) \notin E_3\big)}$$

$$= \log \frac{P\big(w^1_{1,3}|(1,3) \in E_3\big) P\big(w^2_{1,3})|(1,3) \in E_3\big)}{P\big(w^1_{1,3}|(1,3) \notin E_3\big) P\big(w^2_{1,3})|(1,3) \notin E_3\big)}$$

Independence assumption

$$= w^1_{1,3} + w^2_{1,3}$$

Take away messages so far …
(combine with outline slide)

- Maximum likelihood approach: inferring the regulatory network structure by using gene expressions is difficult => exponentially many cases to score, some undistinguishable cases)
- Limit search space => relevance networks
- Use many data types => binding, motif, chromatin, etc.
- Integrated approaches work the best!

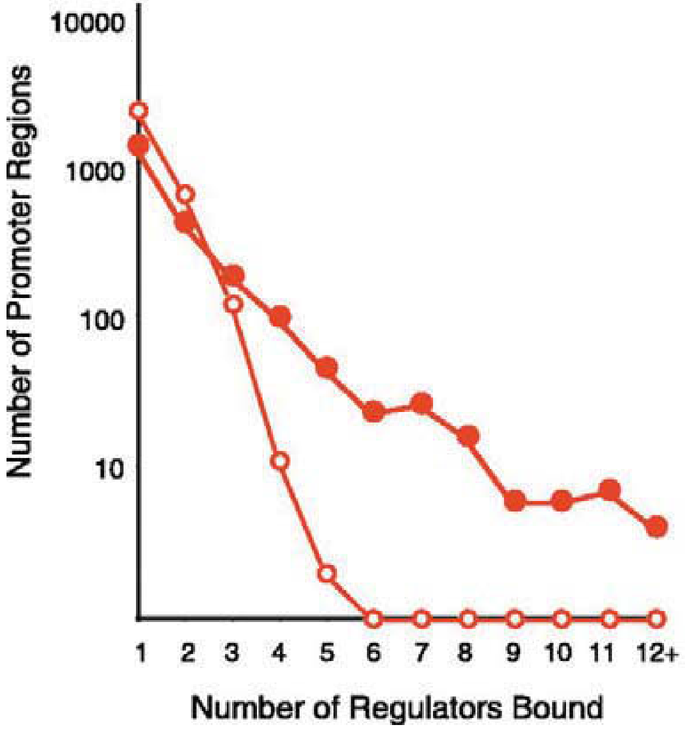
Goals for today: Network analysis

1. Introduction to networks
2. Applications of regulatory networks
   - Predicting expression of target genes: graphical models, linear regression and regression trees
   - Predicting functions of un-annotated genes, guilt by association
3. Inferring "structure" of regulatory networks
   - Likelihood approach, challenges
   - Simplified approaches and their problems
   - Integrated approaches
4. Structural properties of regulatory networks
   - Scale free degree distribution
   - Network motifs
   - Spectral clustering and modular networks

Structural Properties of Regulatory networks

- "Scale-free": Graph is self-similar at all scales
- Degree distribution follows a power law
  - P(d) ~ d^{γ}
- Implies the presence of hubs
- Hub perturbations are often lethal

Regulatory networks have scale-free distribution



In degree distribution of E. coli regulatory network

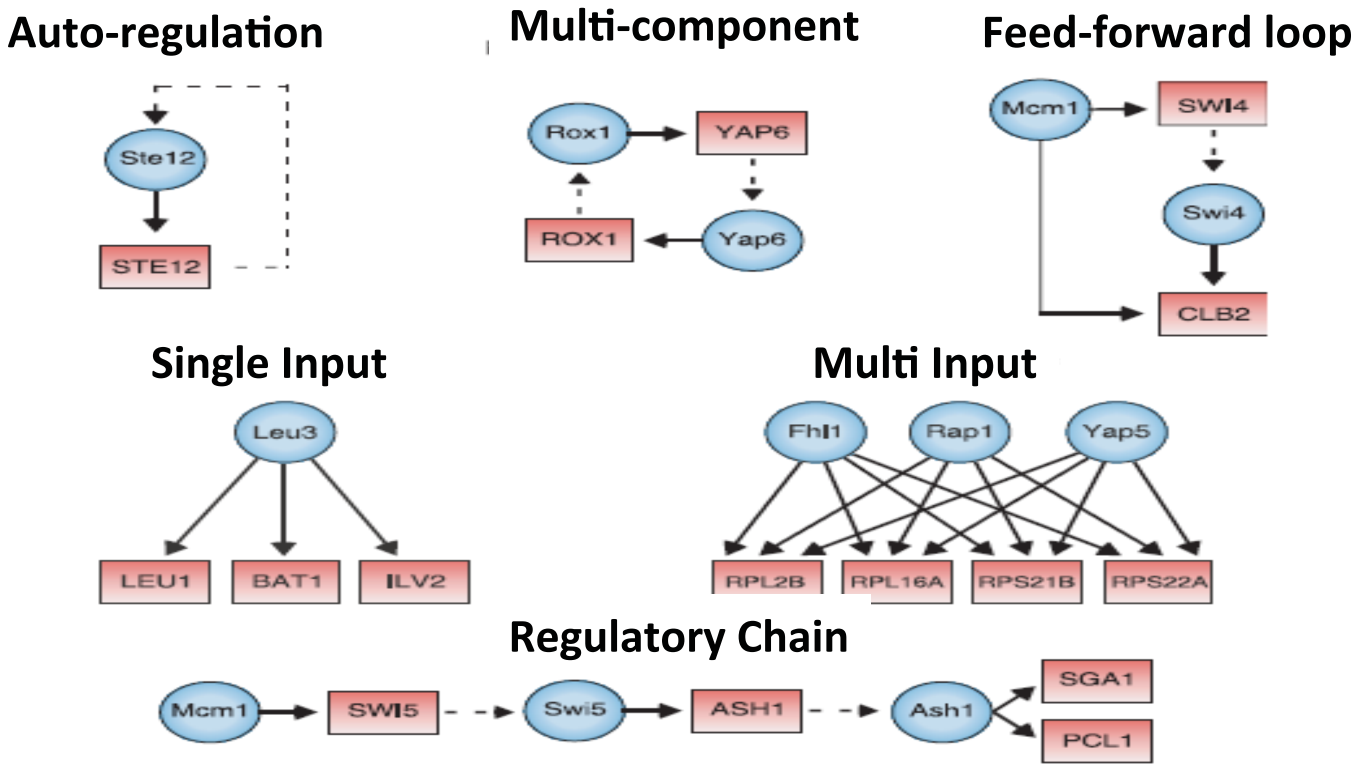
Source: Lee, Tong Ihn et al. "Transcriptional regulatory networks in Saccharomyces cerevisiae." Science 298, no. 5594 (2002): 799-804.

Adapted from Albert 05,

Why are scale free distributions important

- Presence of hubs
- Make the network robust to perturbations
- Preserve overall connectivity
- Perturbations to hubs is often lethal for an organism

Structural network motifs



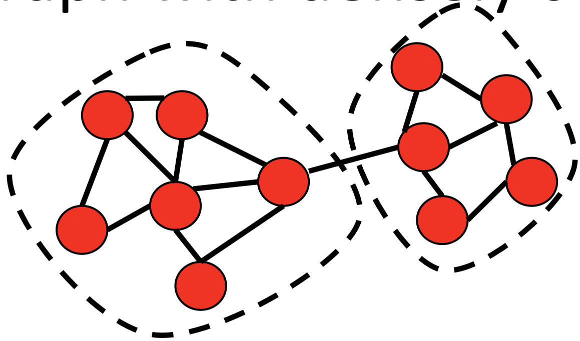
Source: Lee, Tong Ihn et al. "Transcriptional regulatory networks in Saccharomyces cerevisiae." Science 298, no. 5594 (2002): 799-804.

Feed-forward loops involved in speeding up in response of target gene

Lee *et.al. 2002,* Mangan & Alon, 2003

Modularity of regulatory networks

- Modular: Graph with densely connected subgraphs

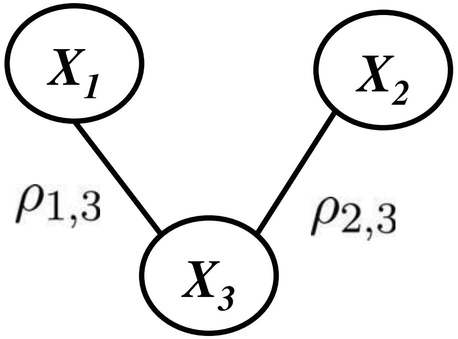


- Genes in modules involved in similar functions and co-regulated
- Modules can be identified using graph partitioning algorithms
  - Markov Clustering Algorithm
  - Girvan-Newman Algorithm
  - ***Spectral partitioning***

Newman PNAS 2007

An algebraic view to networks

- A matrix representation of a network:
  - Unweighted network => binary adjacency matrix
  - Weighted network => real-valued matrix

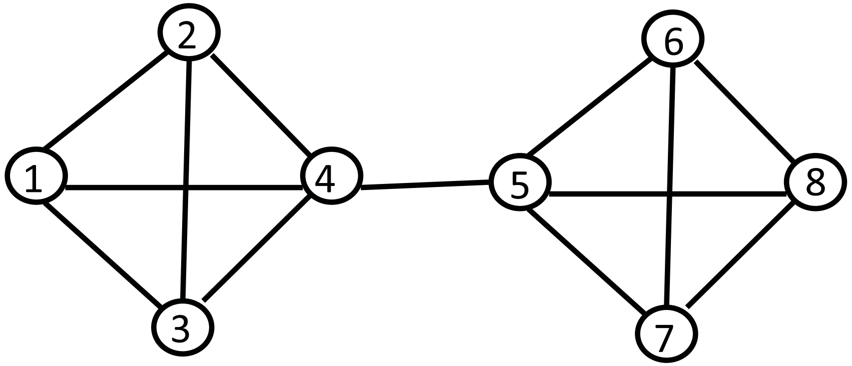


$$A=\begin{bmatrix} 0 & 0 & \rho_{1,3} \\ 0 & 0 & \rho_{2,3} \\ \rho_{1,3} & \rho_{2,3} & 0 \end{bmatrix}$$

- Laplacian Matrix

$$L=\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

An algebraic view to networks-example



A =

$$\begin{array}{cccccccc}
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0
\end{array}$$

Adjacency Matrix

L =

$$\begin{array}{cccccccc}
3 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\
-1 & 3 & -1 & -1 & 0 & 0 & 0 & 0 \\
-1 & -1 & 3 & -1 & 0 & 0 & 0 & 0 \\
-1 & -1 & -1 & 4 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 4 & -1 & -1 & -1 \\
0 & 0 & 0 & 0 & -1 & 3 & -1 & -1 \\
0 & 0 & 0 & 0 & -1 & -1 & 3 & -1 \\
0 & 0 & 0 & 0 & -1 & -1 & -1 & 3
\end{array}$$

Laplacian Matrix

Eigen decomposition principle-introduction

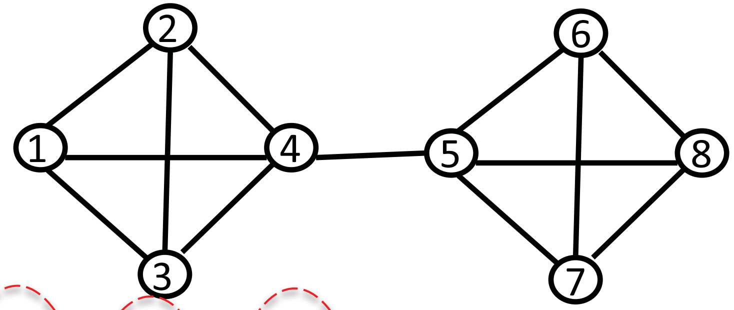
- Suppose *L* is a square matrix:

$$L = U\Sigma U^{-1}$$

$$\mathbf{\Sigma} = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & & \\ \vdots & & \ddots & \\ 0 & & & \lambda_n \end{pmatrix}$$

- $\mathbf{U}$ contains eigenvectors.
- $\mathbf{\Sigma}$ is a diagonal matrix of eigenvalues.
- For symmetric matrices, eigenvalues are real
- Why is it useful?

Eigen decomposition-example



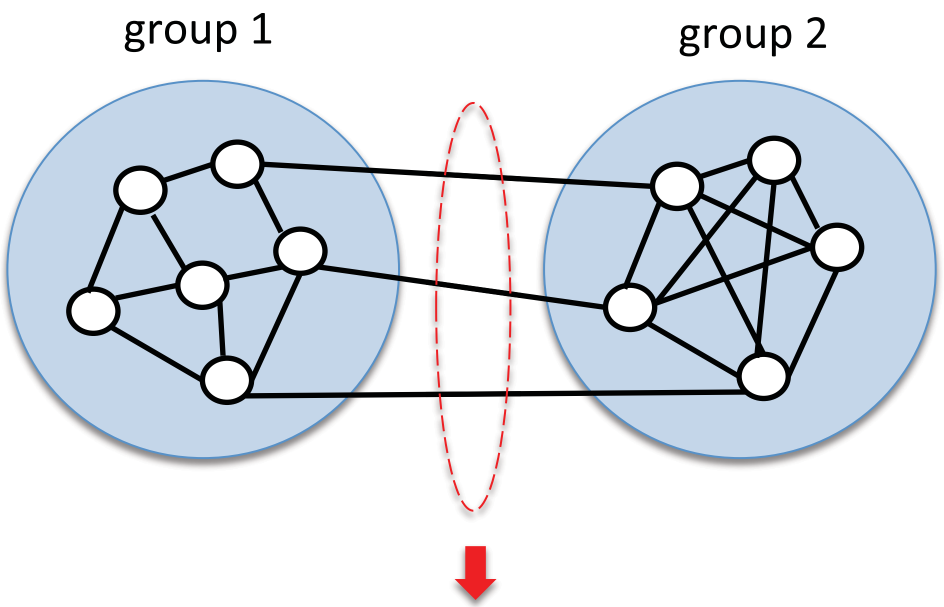
$$L = U\Sigma U^{-1}$$

$$U = \begin{matrix} 0.3536 & -0.3825 & 0.2714 & -0.1628 & -0.7783 & 0.0495 & -0.0064 & -0.1426 \\ 0.3536 & -0.3825 & 0.5580 & -0.1628 & 0.6066 & 0.0495 & -0.0064 & -0.1426 \\ 0.3536 & -0.3825 & -0.4495 & 0.6251 & 0.0930 & 0.0495 & -0.3231 & -0.1426 \\ 0.3536 & -0.2470 & -0.3799 & -0.2995 & 0.0786 & -0.1485 & 0.3358 & 0.6626 \\ 0.3536 & 0.2470 & -0.3799 & -0.2995 & 0.0786 & -0.1485 & 0.3358 & -0.6626 \\ 0.3536 & 0.3825 & 0.3514 & 0.5572 & -0.0727 & -0.3466 & 0.3860 & 0.1426 \\ 0.3536 & 0.3825 & 0.0284 & -0.2577 & -0.0059 & -0.3466 & -0.7218 & 0.1426 \\ 0.3536 & 0.3825 & 0.0000 & 0.0000 & 0.0000 & 0.8416 & -0.0000 & 0.1426 \end{matrix}$$

$$\Sigma = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.3542 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.0000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4.0000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4.0000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5.6458 \end{matrix}$$

- First eigenvalue of Laplacian matrix is always zero.
- What does second eigenvector of Laplacian matrix represent?

Spectral Partitioning- problem setup



minimize # of edges between groups

# of edges between groups=(total # of edges)-(# edges within groups)

nodes *i* and *j* are connected => A_{ij}=1

node *i* in group 1 => s_i=1
node *i* in group 2 => s_i=-1

→

nodes *i* and *j* in the same group => $(s_i s_j+1)/2=1$
nodes *i* and *j* in different groups => $(s_i s_j+1)/2=0$

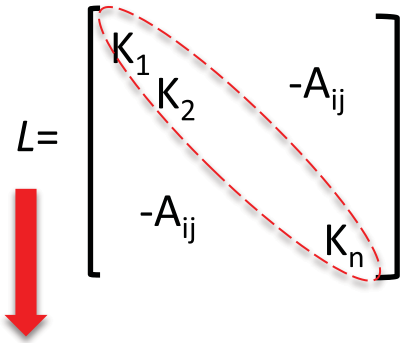
Laplacian matrix plays a major role in network modularization

# of edges between groups=(total # of edges)-(# edges within groups)

$$
\begin{aligned}
&= \left(\frac{1}{2}\sum_{i,j} A_{i,j}\right) - \left(\frac{1}{2}\sum_{i,j}\left(\frac{1}{2}(1+s_is_j)A_{i,j}\right)\right)\\
&= \frac{1}{4}\left(\sum_{i,j} A_{i,j}\right) - \frac{1}{4}\sum_{i,j}\left(s_is_jA_{i,j}\right)\\
&= \frac{1}{4}\left(\sum_{i} K_i\right) - \frac{1}{4}\sum_{i,j}\left(s_is_jA_{i,j}\right) = \frac{1}{4}\mathbf{s}^tL\mathbf{s}
\end{aligned}
$$

node *i* in group 1 => s_i=1

node *i* in group 2 => s_i=-1



$$L=\begin{bmatrix} K_1 & & -A_{ij} \\ & K_2 & \\ -A_{ij} & & K_n \end{bmatrix}$$

Laplacian Matrix

Network modularization by using decomposition of Laplacian matrix

$$\min_{\mathbf{s}} \mathbf{s}^t L \mathbf{s}$$

- Use eigen decomposition principles:

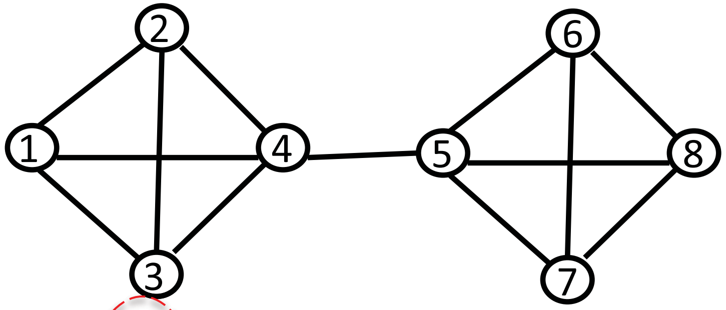
$$L \rightarrow (\mathbf{v}_i, \lambda_i) \qquad L = \sum_i \lambda_i \mathbf{v}_i^t \mathbf{v}_i$$

- Project *s* over eigenvectors of *L*: $\mathbf{s} = \sum_i a_i \mathbf{v}_i$

$$\mathbf{s}^t L \mathbf{s} = \sum_i a_i^2 \lambda_i$$

- Challenges in finding optimal a_i's:
  - – Without other conditions, a trivial solution exists
  - – Second eigenvector characterizes partitioning
  - – Vector s should be integer-valued => projection

Network modularization -revisit to example



$$L = U \Sigma U^{-1}$$

$$U = \begin{matrix}
0.3536 & -0.3825 & 0.2714 & -0.1628 & -0.7783 & 0.0495 & -0.0064 & -0.1426 \\
0.3536 & -0.3825 & 0.5580 & -0.1628 & 0.6066 & 0.0495 & -0.0064 & -0.1426 \\
0.3536 & -0.3825 & -0.4495 & 0.6251 & 0.0930 & 0.0495 & -0.3231 & -0.1426 \\
0.3536 & -0.2470 & -0.3799 & -0.2995 & 0.0786 & -0.1485 & 0.3358 & 0.6626 \\
0.3536 & 0.2470 & -0.3799 & -0.2995 & 0.0786 & -0.1485 & 0.3358 & -0.6626 \\
0.3536 & 0.3825 & 0.3514 & 0.5572 & -0.0727 & -0.3466 & 0.3860 & 0.1426 \\
0.3536 & 0.3825 & 0.0284 & -0.2577 & -0.0059 & -0.3466 & -0.7218 & 0.1426 \\
0.3536 & 0.3825 & 0.0000 & 0.0000 & 0.0000 & 0.8416 & -0.0000 & 0.1426
\end{matrix}$$

$$\Sigma = \begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.3542 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 4.0000 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 4.0000 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 4.0000 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 4.0000 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 4.0000 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 5.6458
\end{matrix}$$

Goals for today: Network analysis

1. Introduction to networks
2. Applications of regulatory networks
   - Predicting expression of target genes: graphical models, linear regression and regression trees
   - Predicting functions of un-annotated genes, guilt by association
3. Inferring "structure" of regulatory networks
   - Likelihood approach, challenges
   - Simplified approaches and their problems
   - Integrated approaches
4. Structural properties of regulatory networks
   - Scale free degree distribution
   - Network motifs
   - Spectral clustering and modular networks

Conclusions

- Regulatory networks are central to gaining a systems-level understanding of living systems
- Structure and functional aspects of the network is unknown
- Probabilistic models provide a mathematical framework of representing and learning regulatory networks

Open issues

- Validation
  - How do we know the network structure is right?
- How do we know if the network function is right?
- Measuring and modeling protein expression
- Understanding the evolution of regulatory networks

Further reading

- Probabilistic graphical models
- Network structure analysis
- Function Prediction

Predicting expression

- Goal: Learn a parametric relationship between regulators and a target gene
- Use the “regulation function” of every target gene as a predictive model
- Predicting expression of multiple genes is essentially equivalent to solving a bunch of regression problems

Modeling the regulatory functions

- Conditional Gaussian models
  - Linear regression model
- Regression Trees
  - Non-linear regression

Hierarchy of more complex models

Figure removed due to copyright restrictions.

From Lei Zhang, RPI