Reverse Engineering of Gene Regulatory Networks

Carlo Cosentino, Ph.D.
Dipartimento di Medicina Sperimentale e Clinica
Università degli Studi Magna Græcia
Catanzaro, Italy
carlo.cosentino@unicz.it
http://bioingegneria.unicz.it/~cosentino
http://wpage.unina.it/carcosen

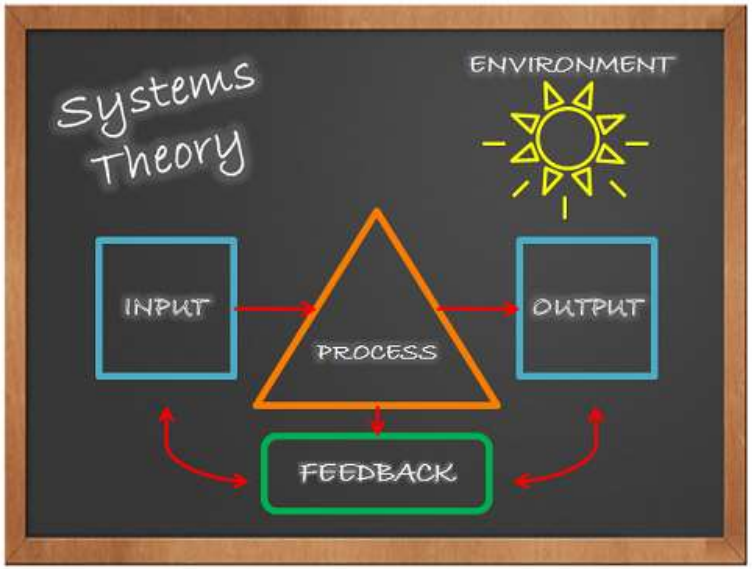
Outline

- Introduction and biological background
- Topological properties of biological networks
- Bayesian Networks and Mutual Information
- Models for network inference
- Regression for network inference: effect of noise and sampling
- Subset selection and shrinkage methods
- Convex optimization methods and prior information exploitation
- Assessment of network inference methods
- Conclusions

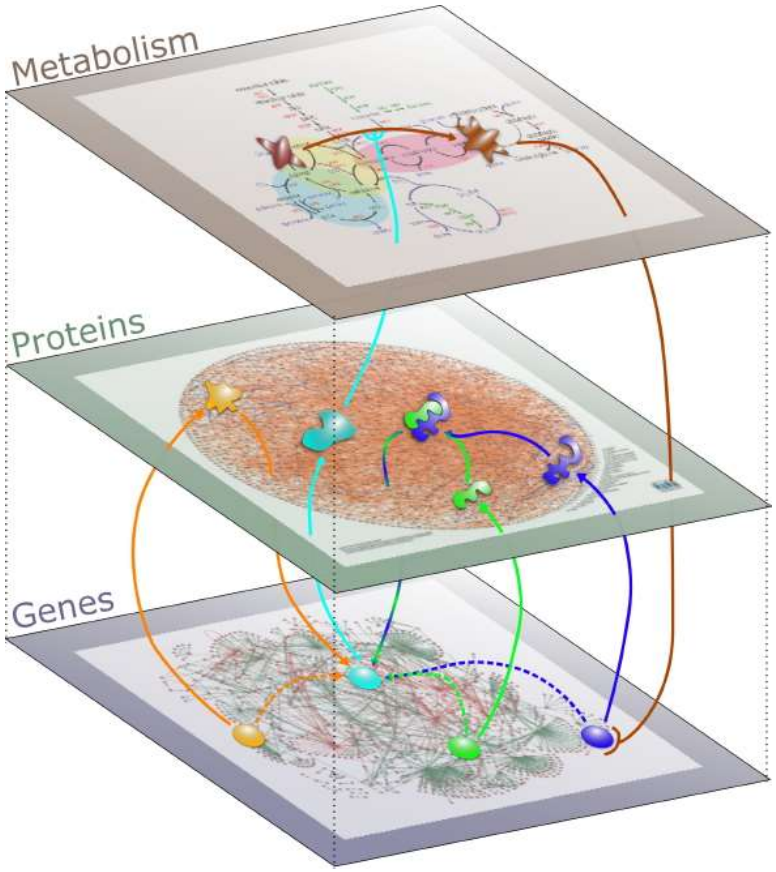
Introduction and biological background

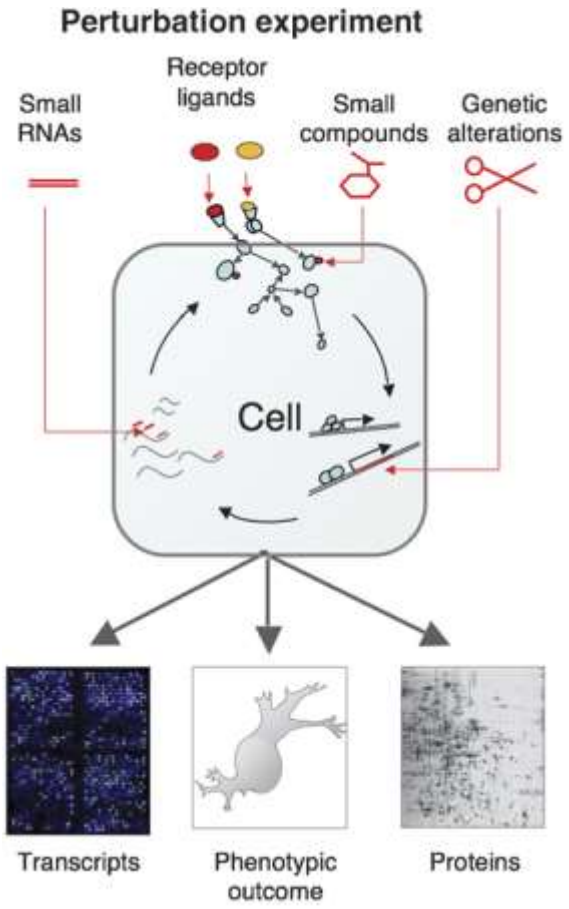
Introduction

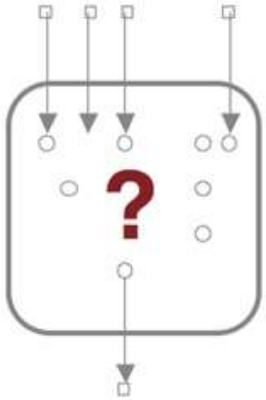
- The interpretation of the huge amounts of data provided by biotechnologies in the last years calls for novel mathematical and computational methods
- Compared to statistical approaches, dynamical models are especially useful to study the evolution over time of biological systems
- Systems and Control Theory provides us with many established tools for the identification and analysis of network models



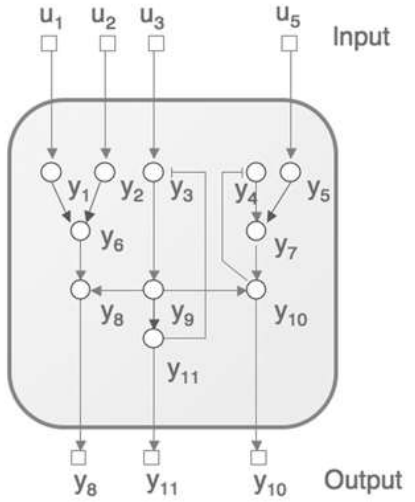
Biological networks



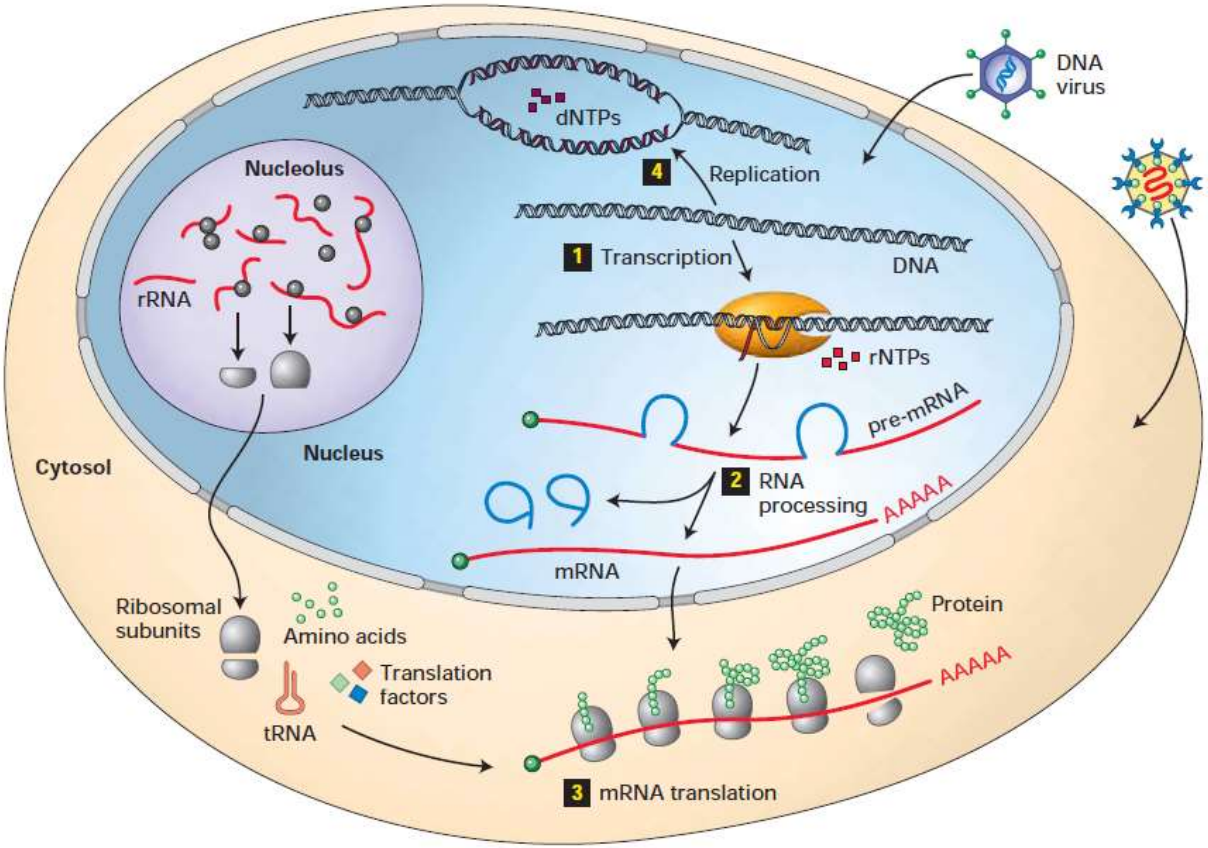


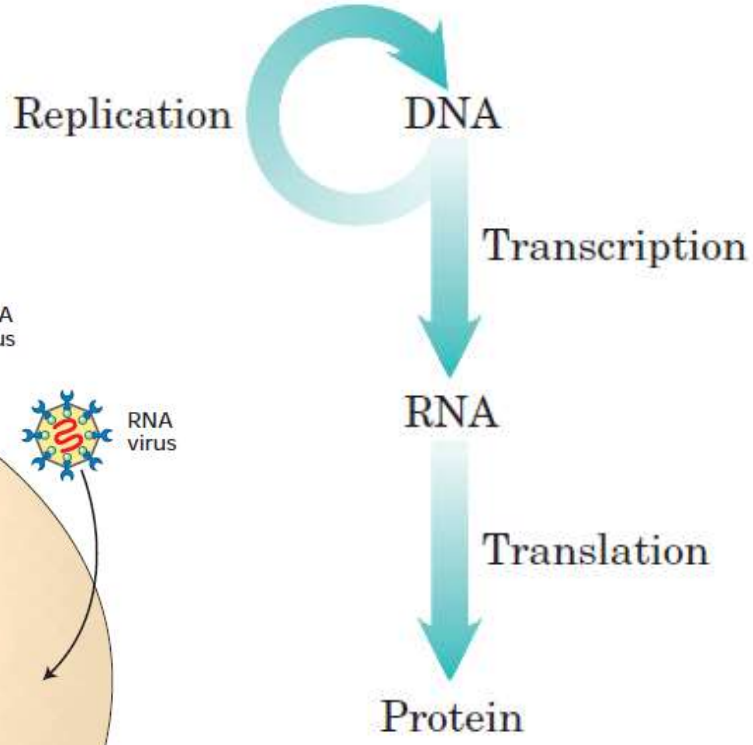


Infer functional interactions between pathway components



A closer look at gene networks

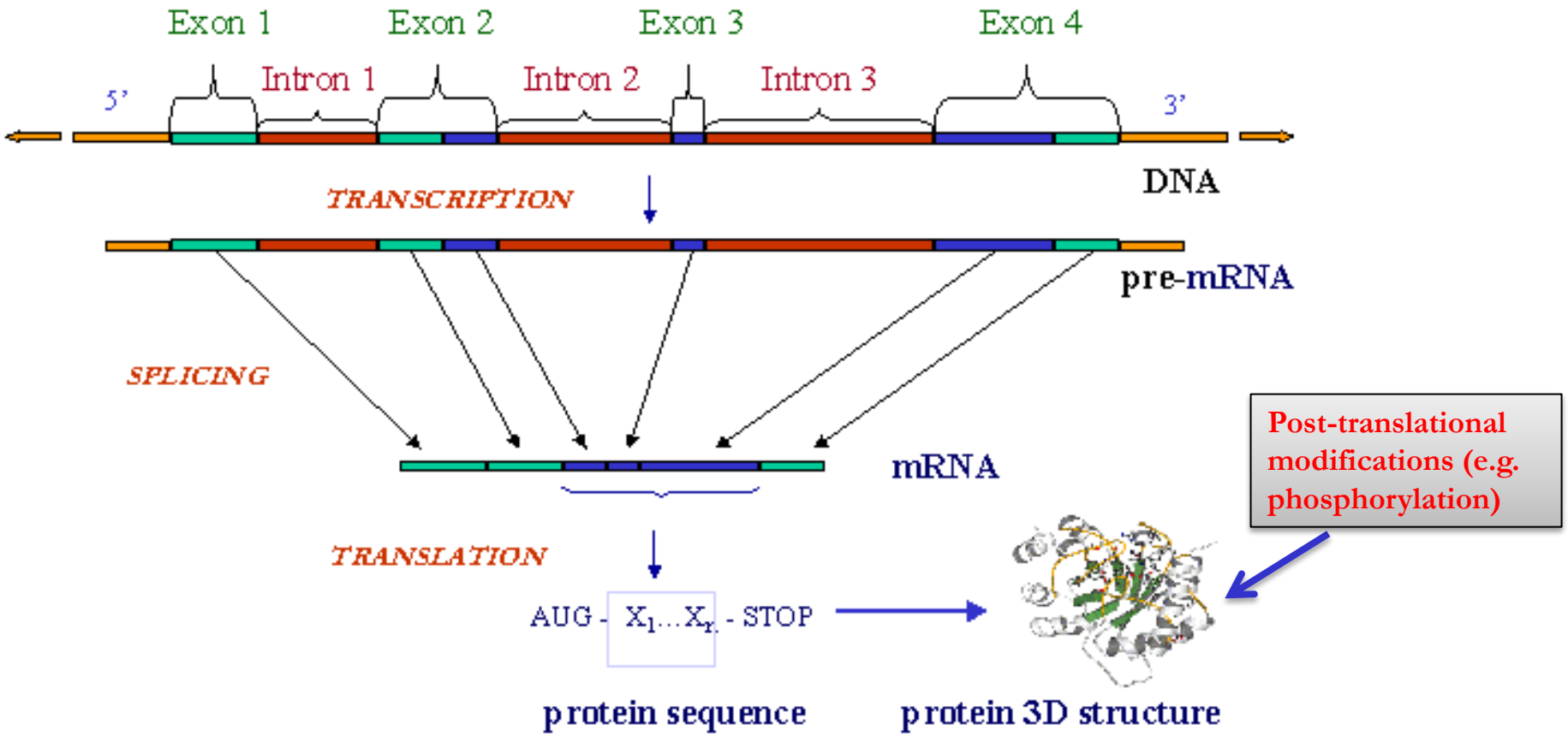




Lodish et al, *Molecular Cell Biology*

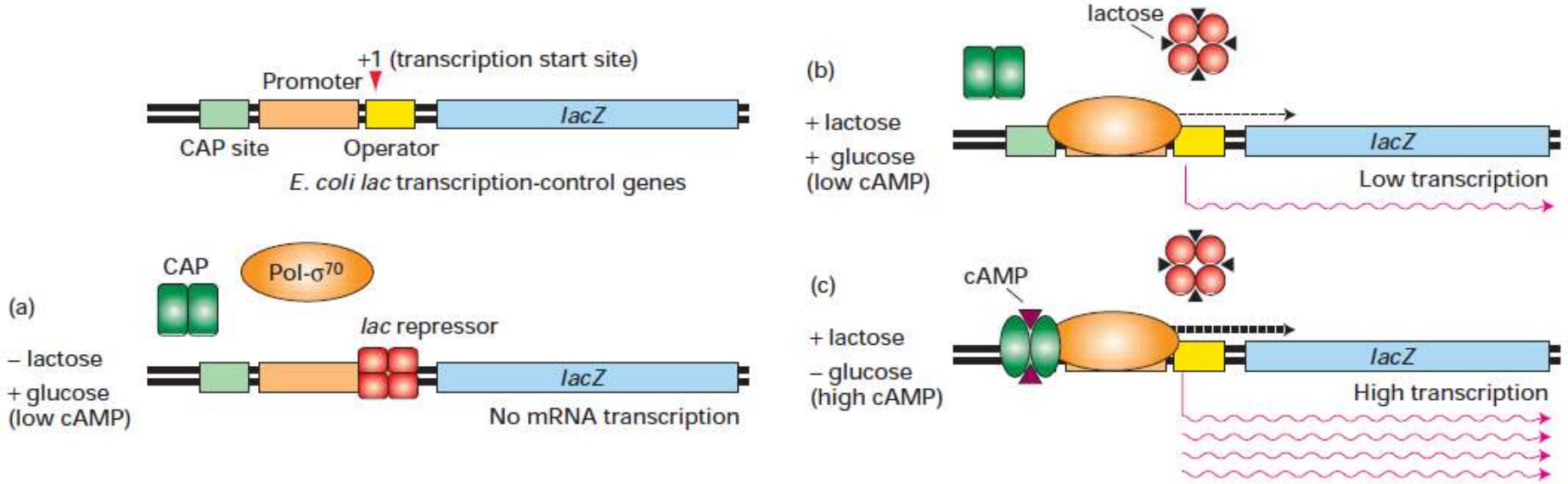
Alternative splicing

- Through alternative splicing, the same coding region can produce more than one protein



Transcriptional gene regulation

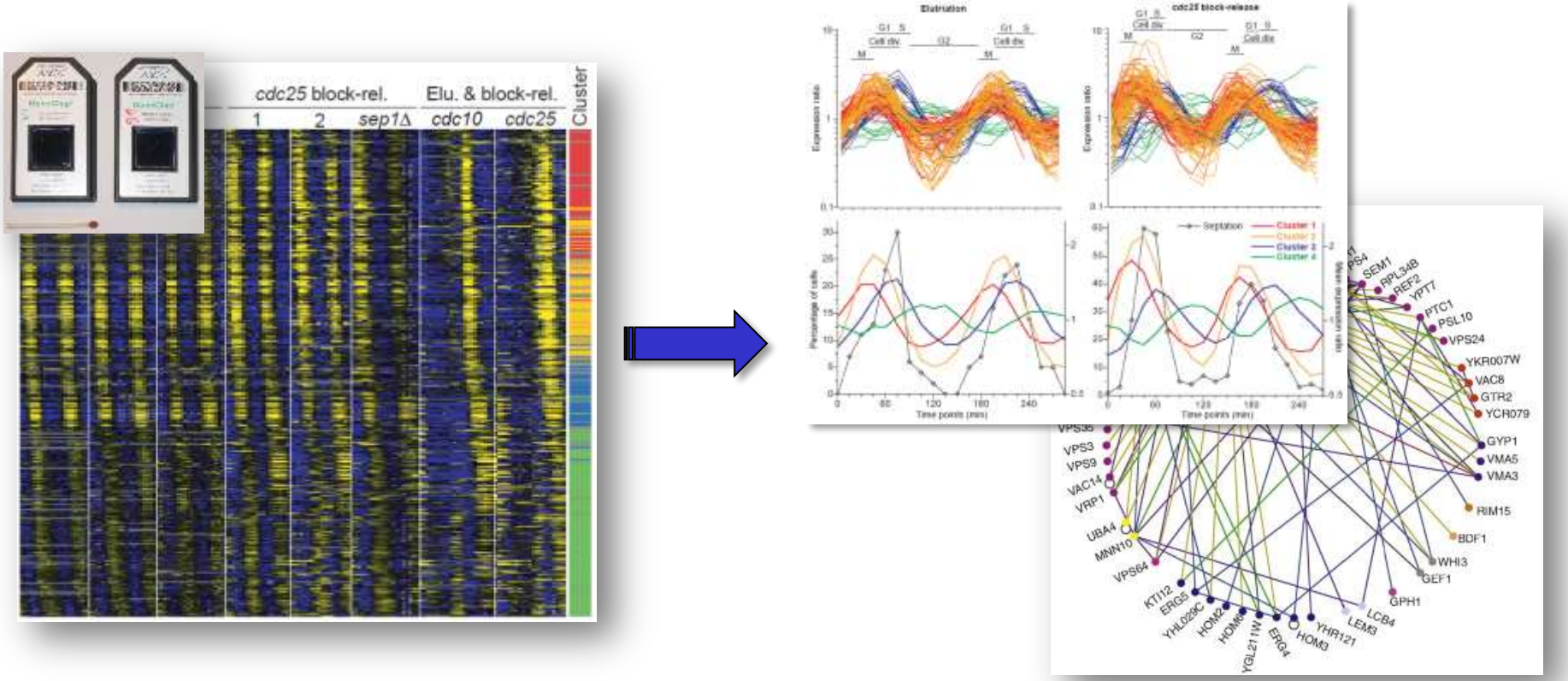
- Genes expression is regulated via adjacent transcription-control regions in a combinatorial way
- Only a subset of the whole genome is expressed at a particular time or in a specific cell type



Lodish et al, *Molecular Cell Biology*

High-throughput experiments

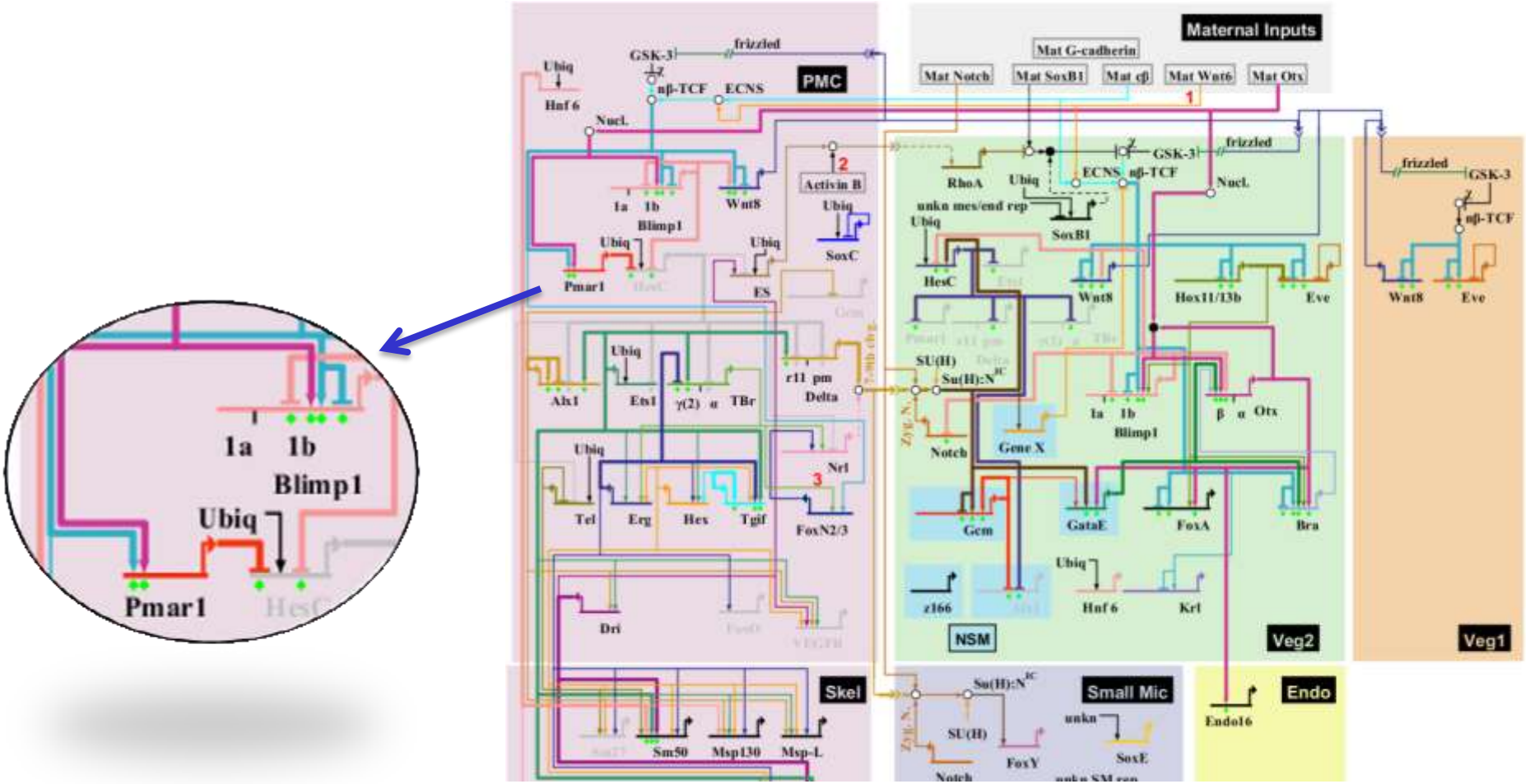
- Modern biotechnologies, like cDNA and Protein arrays, RNA-seq, ChIP-Chip, enable to monitor the activity of thousands of species, resulting in a *systemic* snapshot of cellular activity at a certain time instant
- Is it possible to infer interaction networks from these large datasets?



Topological properties of biological networks

Graphical models of GRNs

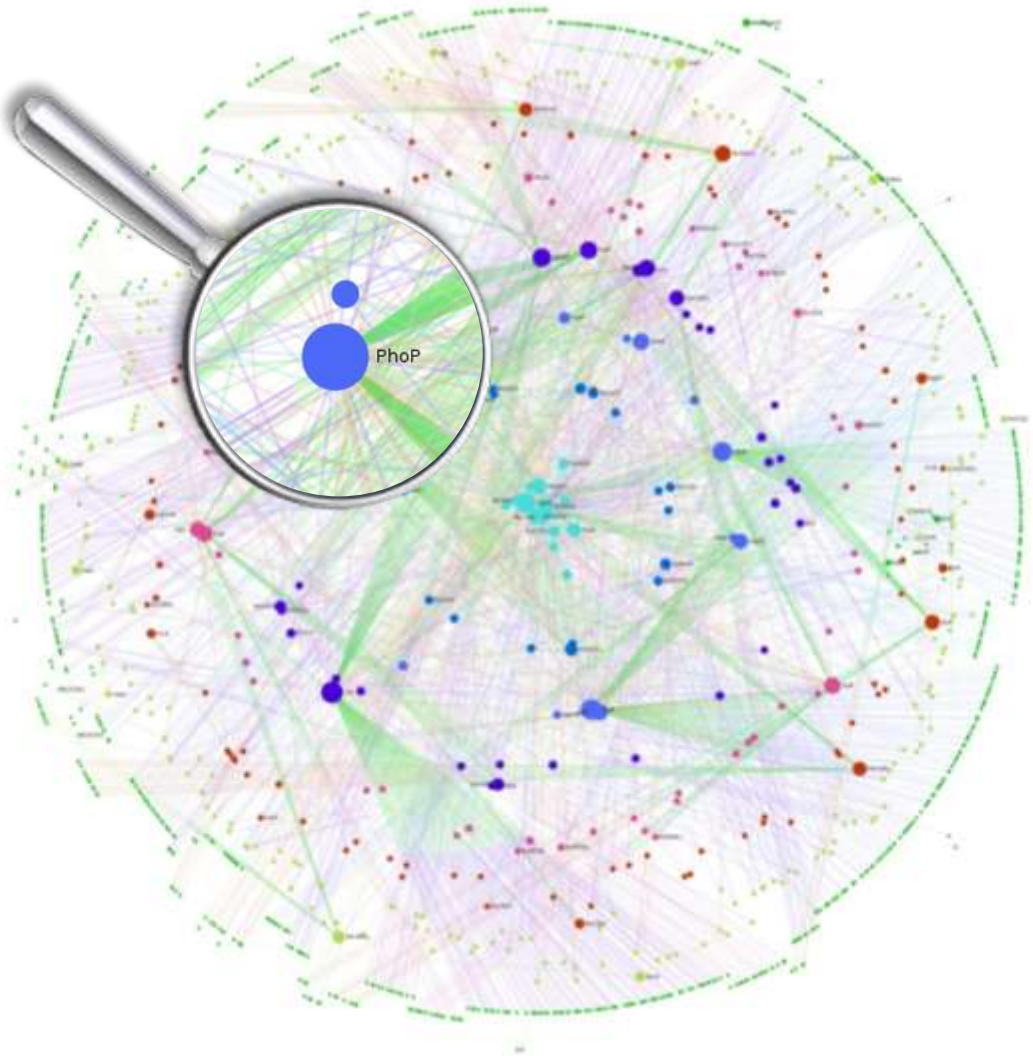
Transcriptional regulatory network of *S. purpuratus* endomesoderm development (6-18 h)



BioTapestry, Davidson Lab, California Institute of Technology

Large-scale networks

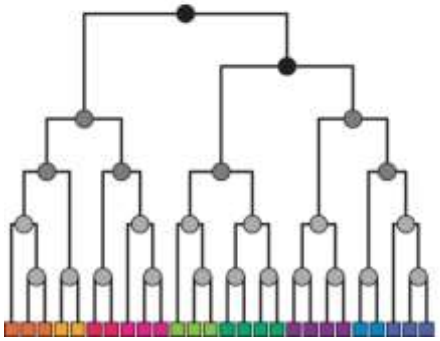
Transcriptional regulatory network of *Mycobacterium Tubercolosis*



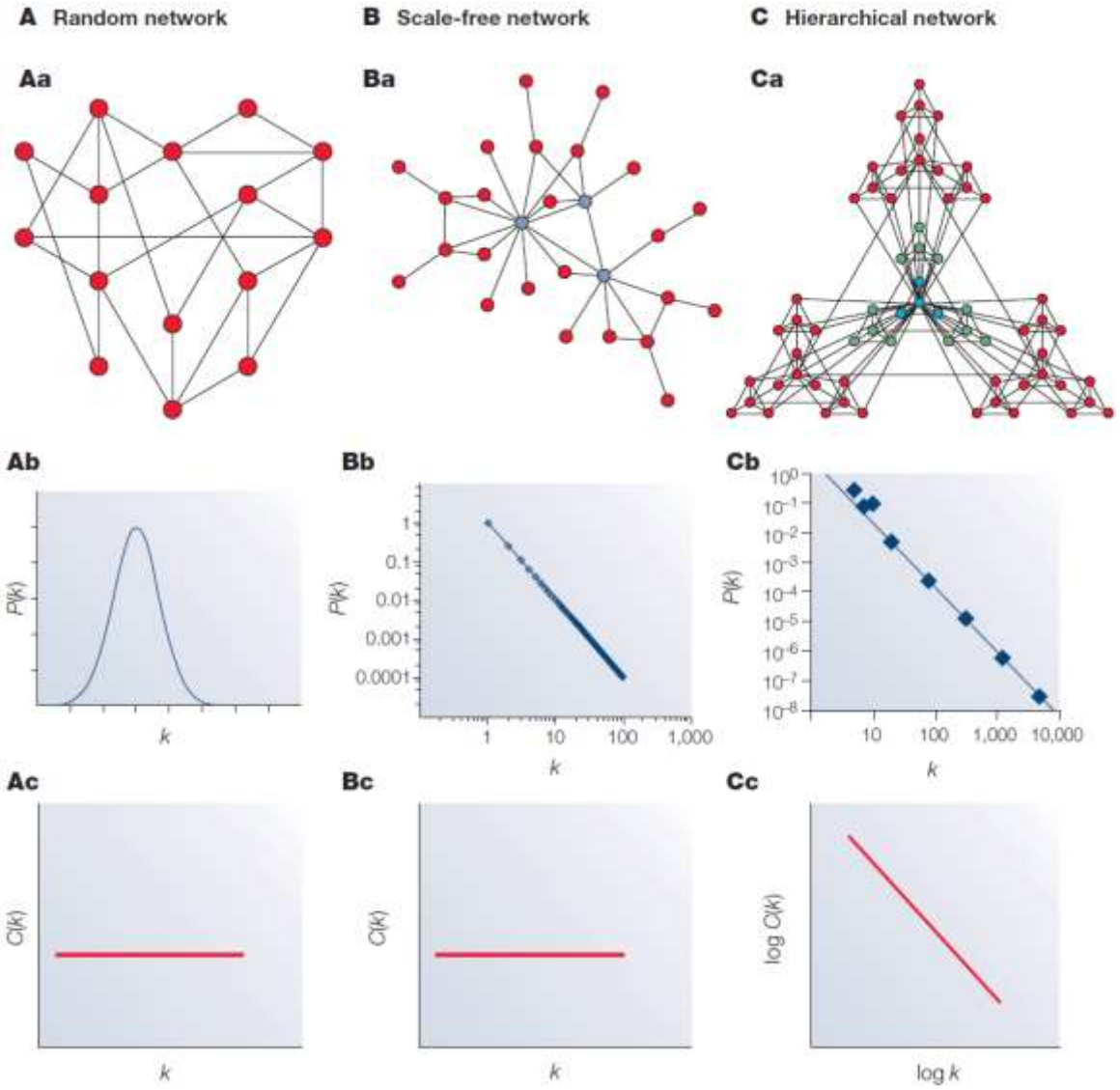
Topological properties of networks

- *Degree*: number of edges starting from (*out-degree*) or pointing at (in-degree) a node
- *Local Clustering* : measures the connectivity between the neighbors of a node
- *Network Average Clustering*: average of local clustering coefficients
- *Modularity*: measure of the division of nodes into highly interconnected subgroups
- *Network indexes*:
  - *Radius, mean path length, …*
- Node Centrality indexes:
  - *Closeness, Betweenness, …*

Hierarchical modularity



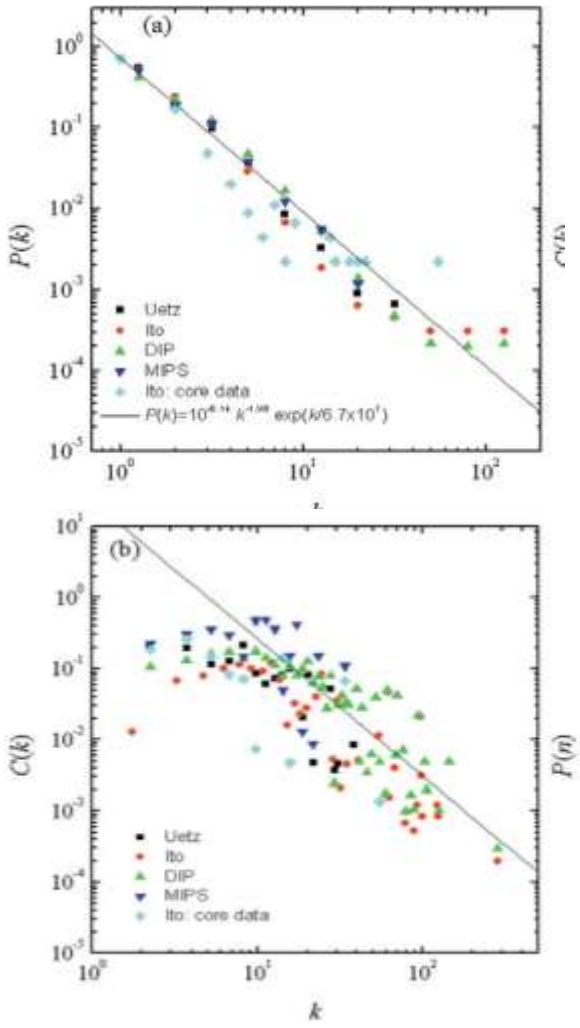
Erdös-Rényi, Scale-free, and Hierarchical networks



Barabasi et al, Nature Review Genetics 101(5), 101–114 , 2004

Topology of biological networks

- Albert has reviewed the topology of different kinds of biomolecular interaction networks
- Several of these networks seem to exhibit a scale-free topology
- For instance, transcriptional regulation networks exhibit a scale–free out–degree distribution, signifying the potential of transcription factors to regulate multiple targets
- On the other hand, their in–degree is a more restricted exponential function, suggesting that combinatorial regulation by several TFs is less frequent



Albert, Scale–free networks in cell biology, *Journal of Cell Science* 118(21), 4947–4957, 2005

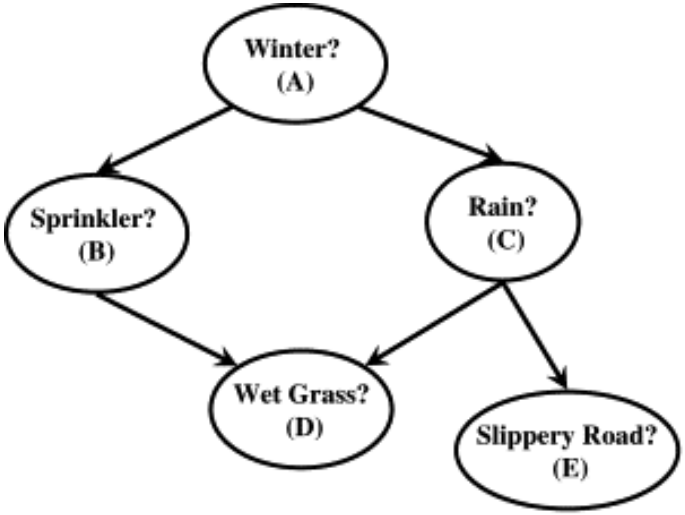
Main approaches to GRNs inference

- A plethora of reverse-engineering approaches have been proposed, mostly applied to gene regulatory networks
- Most of them fit into one of these three frameworks
  - Bayesian Networks
  - Information Theory
  - Dynamical Systems
- The first two are very good to capture the stochastic nature of biomolecular systems
- However, they are not suitable to describe dynamical phenomena, such as those occurring in Gene Regulatory Networks (GRNs)

Approaches based on Bayesian Networks and Mutual Information

Bayesian Networks

- A Bayesian Network is a graphical model of probabilistic relationships among a set of random variables
- The nodes of the network represent genes expression levels and correspond to random variables X_i.
- The graph G and the set of conditional distributions uniquely specify a joint probability distribution $p(\boldsymbol{X})$



A	Θ_A
true	0.6
false	0.4

A	B	$\Theta_{B\|A}$
true	true	0.2
true	false	0.8
false	true	0.75
false	false	0.25

A	C	$\Theta_{C\|A}$
true	true	0.8
true	false	0.2
false	true	0.1
false	false	0.9

B	C	D	$\Theta_{D\|B,C}$
true	true	true	0.95
true	true	false	0.05
true	false	true	0.9
true	false	false	0.1
false	true	true	0.8
false	true	false	0.2
false	false	true	0
false	false	false	1

C	E	$\Theta_{E\|C}$
true	true	0.7
true	false	0.3
false	true	0
false	false	1

Inferring Bayesian Networks

- In order to reverse-engineer a Bayesian network model of a gene network, we must find the directed acyclic graph that best describes the data
- To do this, a scoring function is chosen, in order to evaluate the candidate graphs G with respect to the data set D
- The score can be defined using Bayes rule

$$P(G \mid D) = \frac{P(D \mid G)P(G)}{P(D)}$$

- If the topology of the network is partially known, the *a priori* knowledge can be included in P(G)
- The most popular scores are the Bayesian Information Criterion (BIC) or Bayesian Dirichlet equivalence (BDe)
- They incorporate a penalty for complexity to cope with overfitting

Limitations of Bayesian Networks

- An important limitation of BNs is that they cannot take into account feedback loops
- The evaluation of all possible networks involves checking all possible combinations of interactions among the nodes
- This problem is NP-hard, therefore heuristic methods are used, like the greedy–hill climbing approach, the Markov–Chain Monte Carlo method, or Simulated Annealing
- BNs are static models, thus they cannot capture the dynamics of the biological system
- A software tool for inferring BNs is Banjo, developed by the group of Hartemink (http://www.cs.duke.edu/~amink/software/banjo)

Information–Theoretic Approaches

- Information – theoretic approaches use a generalization of the Pearson correlation coefficient

$$r_{ij} = \frac{\sum_{k=1}^{M} (x_i(k) x_j(k))}{\sqrt{(\sum_{k=1}^{M} x_i^2(k) \sum_{k=1}^{M} x_j^2(k))}}$$

used in hierarchical clustering, namely the Mutual Information (MI)

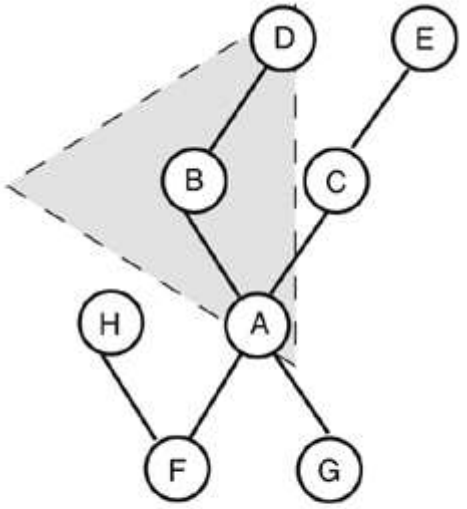
- Mutual information is a metrics of dependency between two random variables

$$I(X,Y) = \sum_{\substack{x \in \mathcal{X} \\ y \in \mathcal{Y}}} p(x,y) \frac{\log p(x,y)}{p(x)p(y)}$$

where *p(x,y)* is joint probability distribution of *X* and *Y*, and *p(x)*, *p(y)* are the marginal probabilities.

Information–Theoretic Approaches

- From the definitions above it follows that
  - MI becomes zero if the two variables are statistically independent
  - A high value of MI indicates that the variables are non–randomly associated to each other
  - $\mathrm{MI}_{ij}=\mathrm{MI}_{ji}$ therefore the resulting reconstructed graph is undirected
- The network is pruned based on the Data Processing Inequality (see figure)
- Well assessed software tools based on Information Theory: ARACNe and CLR
- Drawbacks: no causality, not possible to exploit prior knowledge



MI(A,H)=0
MI(A,B)>0
0<MI(A,D)≤min{MI(A,B), MI(B,D)}

Models for network reverse engineering

Dynamical model of a GRN

- A basic model of transcriptional regulation is composed of two types of species: genes (x_i) and proteins (y_i)

$$\dot{x}_i = m_i \cdot f_i(\mathbf{y}) - \lambda_i^{\text{RNA}} \cdot x_i$$

$$\dot{y}_i = r_i \cdot x_i - \lambda_i^{\text{Prot}} \cdot y_i$$

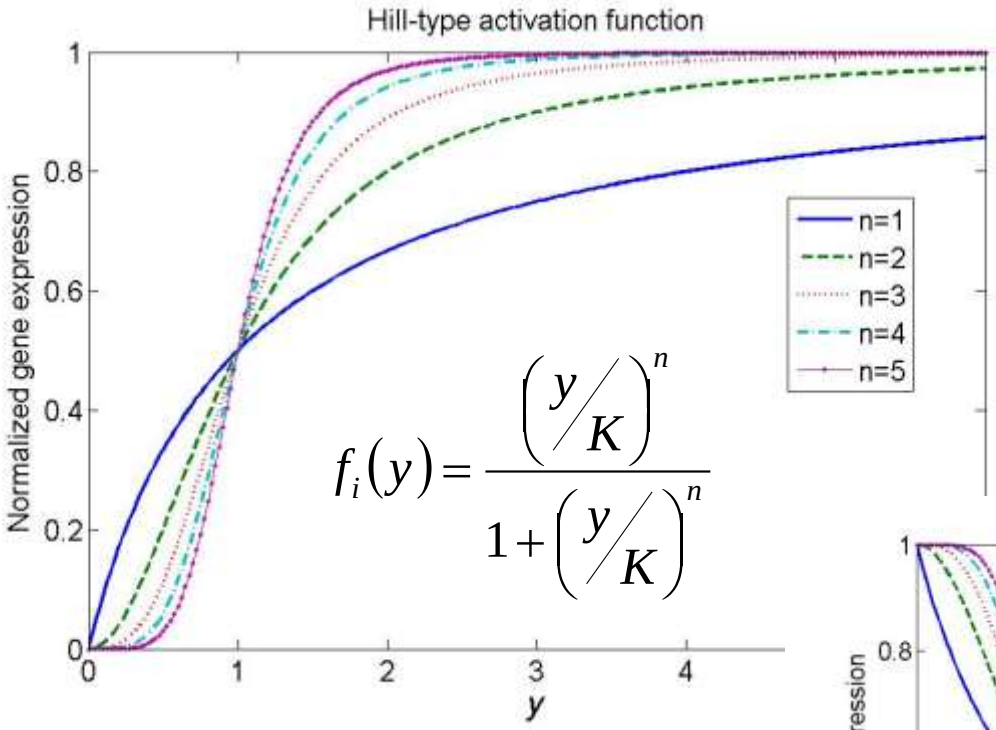
Parameter	Description
m_i	Max transcription rate
λ_i^{RNA}	mRNA degradation rate
r_i	Translation rate
λ_i^{Prot}	Protein degradation rate
k_{ij}	Dissociation constant
n_{ij}	Hill coefficient

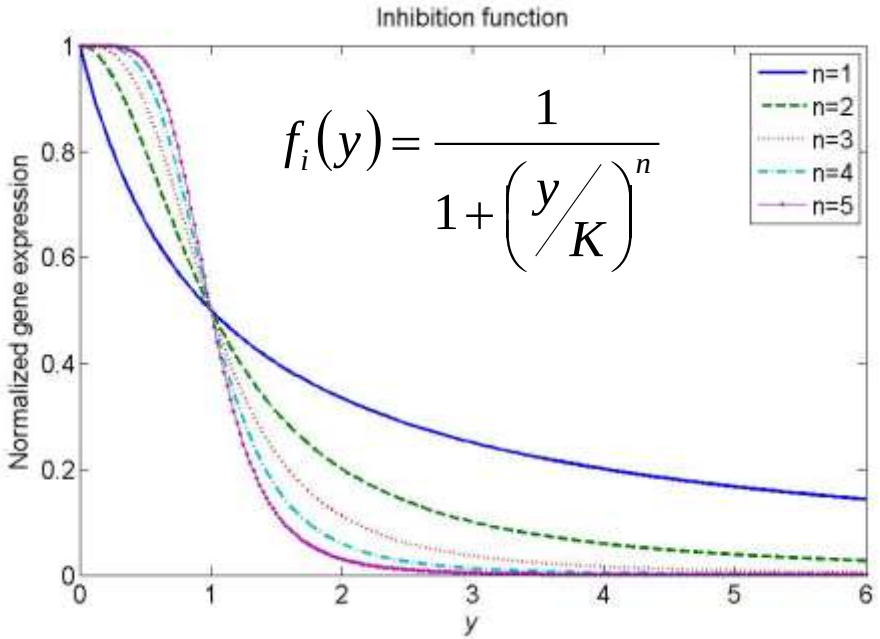
- Simplifying assumption: one protein for each gene!
- The input-function $f_i(\mathbf{y})$ computes the relative activation of gene i as a function of the transcription factor proteins
- In a typical transcriptomic experiment, only the (steady-state) values x_i are measured

A closer look at the input function

- The network topology is implicitly defined by the input function
- The protein concentrations appearing in $f_i(\mathbf{y})$ define the regulatory relationships for the i-th gene
- A rational form is typically assigned to $f_i(\mathbf{y})$
  - One transcription factor $\Rightarrow$ $f_i(y_j) = \dfrac{\alpha_0 + \alpha_1 \chi_j}{1 + \chi_j}$ $\qquad \chi_j = \left(\dfrac{y_j}{k_{ij}} \right)^{n_{ij}}$
  - Two transcription factors $\Rightarrow$ $f_i(y_j, y_k) = \dfrac{\alpha_0 + \alpha_1 \chi_j + \alpha_2 \chi_k + \alpha_3 \rho_3 \chi_j \chi_k}{1 + \chi_j + \chi_k + \rho_3 \chi_j \chi_k}$

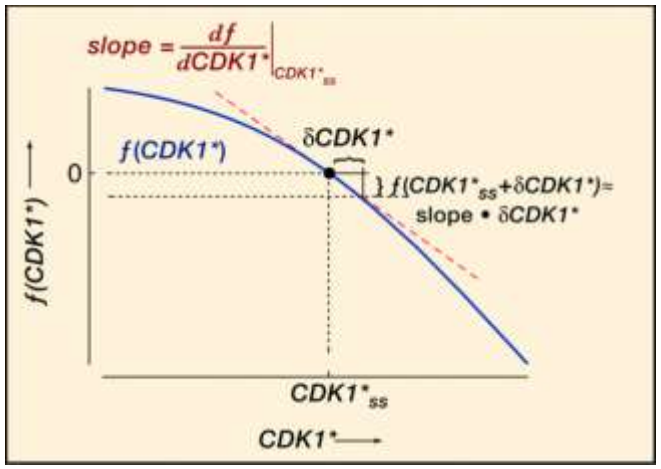
A closer look at the input function

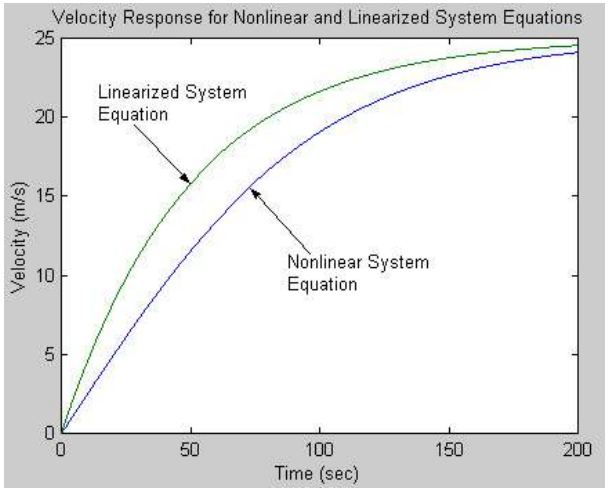




Linearized models

- The identification of high-order nonlinear ODE models is a daunting task, both from a theoretical point of view and in terms of computational requirements
- Linearized models yield good results when applied to data from perturbation experiments





- Most methods are based on linearized models, e.g. those by Gardner and di Bernardo, dealing both with steady–state (*NIR*) and time–series data (*TSNI*), or the *Inferelator* by Bonneau et al.

Modular Response Analysis (MRA)

- Assume that the steady-state value of the j-th species is given by a function of the other species x and of the parameters p

$$f_i(x,p)=0 \quad \Longrightarrow \quad \frac{\partial f_i}{\partial p_j}=\sum_k \frac{\partial f_i}{\partial x_k}\frac{\partial x_k}{\partial p_j}=-\frac{\partial f_i}{\partial x_i}\sum_k r_{ij}R_{kj}=0$$

- The influence on the i-th species of the other species and of external perturbations are given by the coefficients

connection coeffients (unknown)	perturbation coefficients (measured)
$r_{ij} \equiv -\left(\frac{\partial f_i}{\partial x_j}\right) \Big/ \left(\frac{\partial f_i}{\partial x_i}\right)$	$R_{ij} \equiv \frac{\partial x_i}{\partial p_j}$

- If the j-th perturbation does not directly affect the i-th species, then

$$\frac{\partial f_i}{\partial p_j}=0 \quad \Longrightarrow \quad \sum_{k\neq j} r_{jk}\,R_{kj}=R_{ij} \qquad j=1,\ldots,n, \quad j\neq i$$

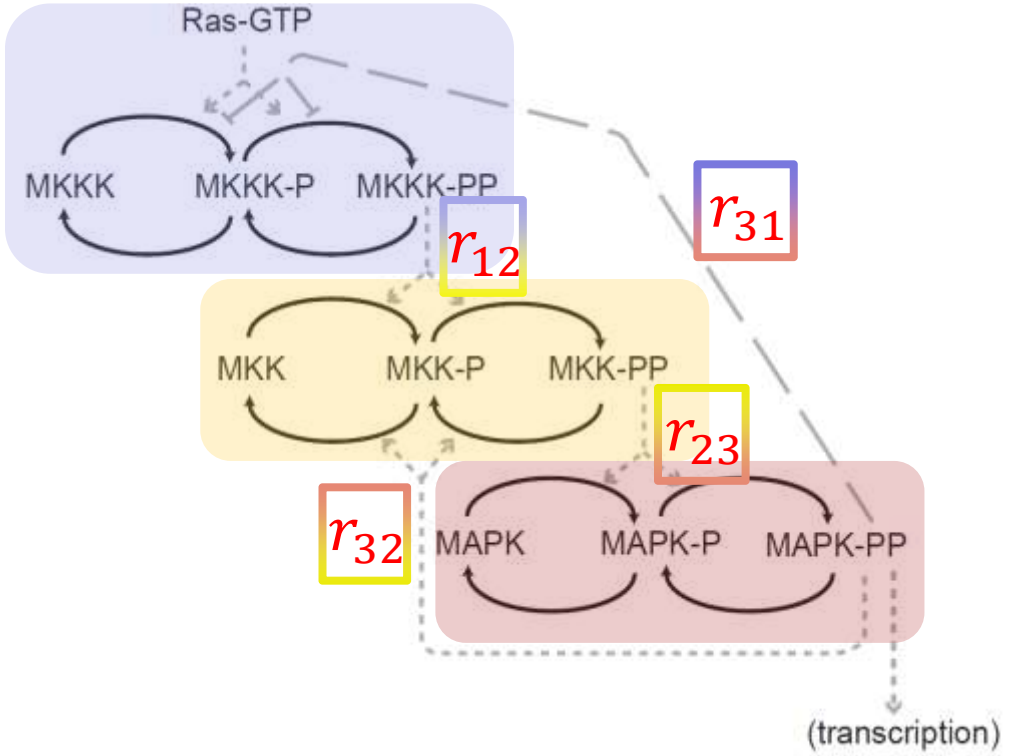
Kholodenko, …, Sontag, et al, *PNAS* (2002), 99(20):12841-12846

Modular Response Analysis (MRA)

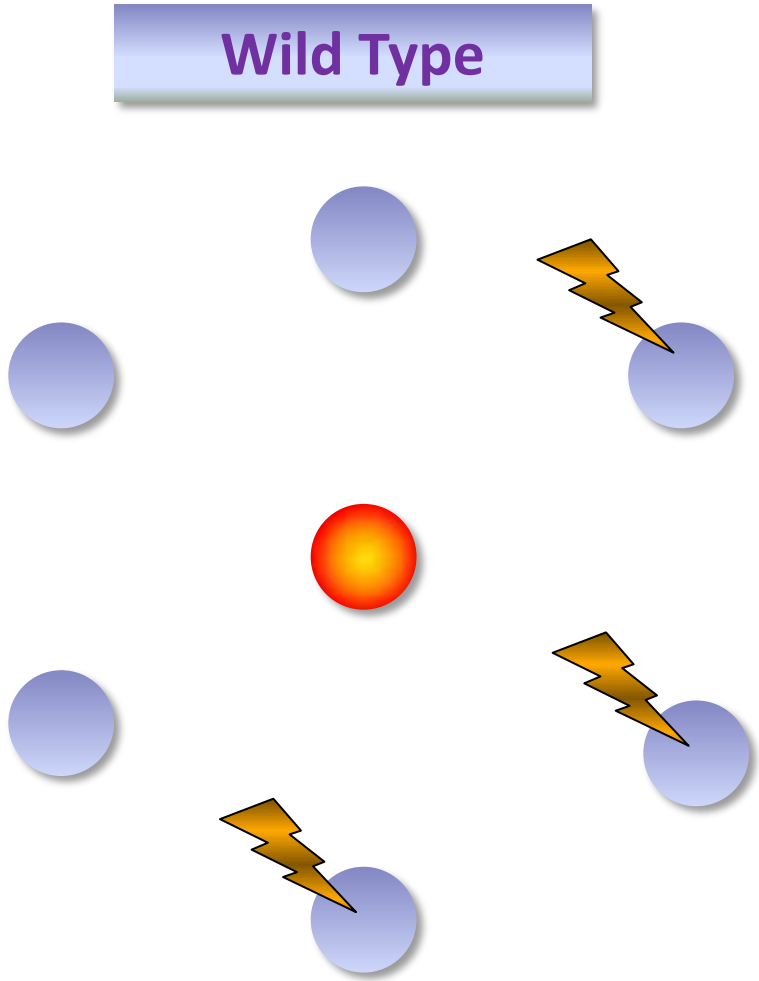
- The name Modular Response Analysis stresses the fact that the same theoretical framework can be applied to modules

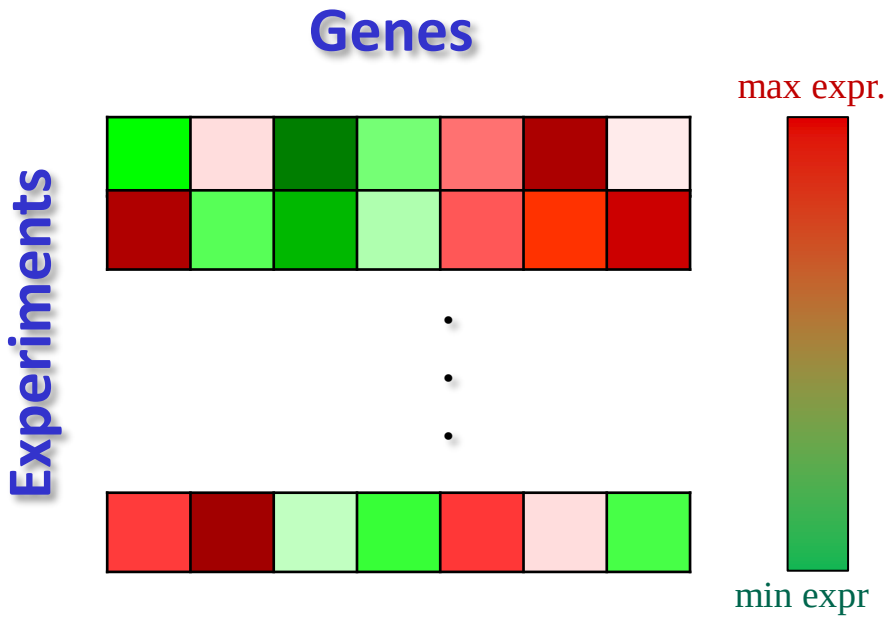
- Required measurements: only the communicating intermediates

- The internal dynamics of the j-th module are summarized by the function $f_j(x, p)$



Construction of the perturbation matrix





- Goal: Infer the regulators of the central node
- Perturb all the nodes except that one
- Measure the expression changes of all nodes each condition

MRA application example

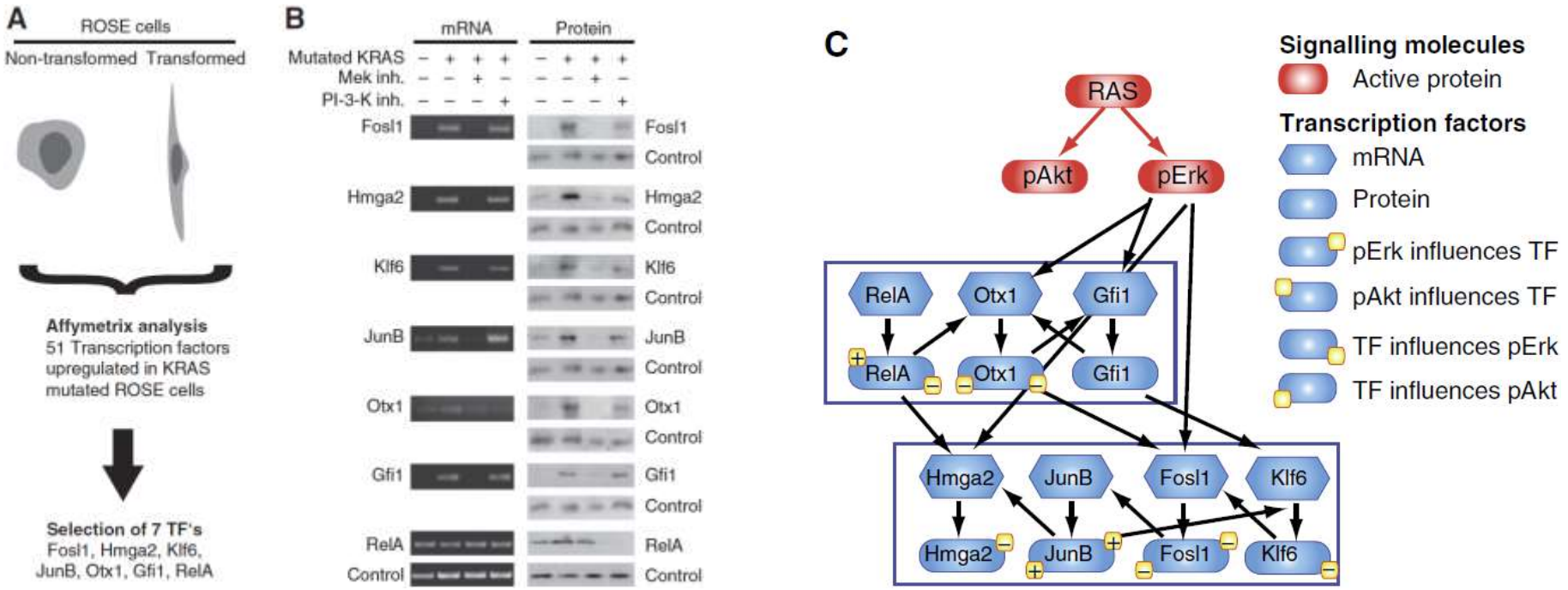
Molecular Systems Biology 8; Article number 601; doi:10.1038/msb.2012.32
Citation: *Molecular Systems Biology* 8:601

www.molecularsystemsbiology.com

molecular systems biology

Reverse engineering a hierarchical regulatory network downstream of oncogenic KRAS

Iwona Stelniec-Klotz[1,5], Stefan Legewie[2,5], Oleg Tchernitsa[1,3], Franziska Witzel[1,4], Bertram Klinger[1,4], Christine Sers[1], Hanspeter Herzel[4], Nils Blüthgen[1,4,6] and Reinhold Schäfer[1,3,6*]



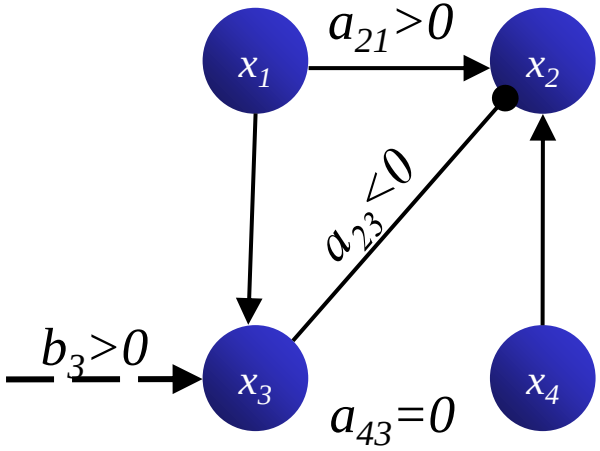
Regression methods for network inference

Inference via Least Squares regression

- Assumption: the system is operating at a stable steady-state
- It is based on the identification of the linearized system

$$x_i(t_k) = \sum_{j=1}^{N} a_{ij} x_j(t_k) + b_i u(t_k) \qquad \begin{matrix} i = 1, \ldots, N \\ k = 1, \ldots, M \end{matrix}$$

- Least Squares (LS) regression methods can be used to estimate the coefficients of the dynamical matrix, a_{ij}, and those of the input matrix, b_i
- $a_{ij} \neq 0$ denotes the presence of an edge in the digraph, between nodes i and j, whereas a nonzero b_i indicates that the node i is directly affected by the perturbation



Least squares regression approaches

- Assume a static linear relationship between a dependent variable y and an independent one x, given h experimental measurements,

$$y^{(k)} = \sum_{i=1}^{n} \theta_i \, x_i^{(k)} + \nu^{(k)} = \theta^T x^{(k)} + \nu^{(k)}, \qquad k = 1, \ldots, h$$

where ν is gaussian noise with zero mean and σ^2 variance.

- The Least Squares (LS) method allows the computation of the optimal value of the vector θ that minimizes the difference between the output of the model

$$\hat{y} = x^T \theta \qquad\qquad e^{(k)} = y^{(k)} - \hat{y}^{(k)}$$

and the measured output y in the sum-of-squared-errors sense.

Least squares regression

- The problem can be conveniently reformulated in matrix form as

$$\min_{\theta} e^T e$$

$$\text{s.t.} \quad e = y - \hat{y} = y - X\theta$$

where (superscripts denote the experiment)

$$X := \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & & \vdots \\ x_1^{(h)} & x_2^{(h)} & \cdots & x_n^{(h)} \end{pmatrix} \qquad y := \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(h)} \end{pmatrix} \qquad \hat{y} := \begin{pmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \\ \vdots \\ \hat{y}^{(h)} \end{pmatrix} \qquad e := \begin{pmatrix} e^{(1)} \\ e^{(2)} \\ \vdots \\ e^{(h)} \end{pmatrix}$$

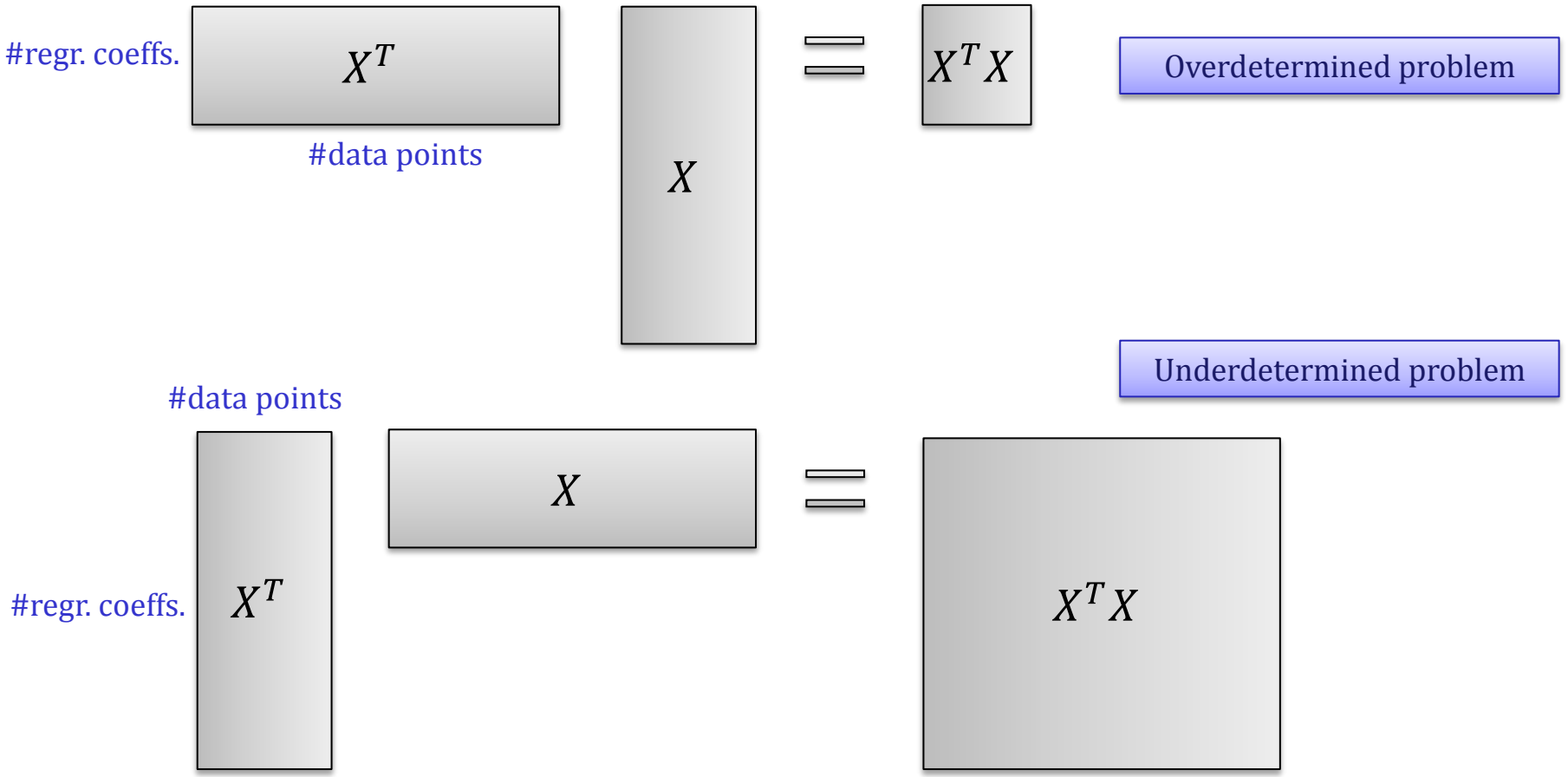
Regressors

- The well-known solution is

$$\hat{\theta} = \left(X^T X\right)^{-1} X^T y$$

Over- and under-determined problems

- To get a full-rank invertible $X^T X$, the #data points must be greater or equal than the #regression coefficients



How to cope with underdetermined problems

- The problem is generally underdetermined: the number of samples is less than the number of regression parameters

- Several strategies can be used to cope with this problem:
  - Limit the number of candidate regulators for each gene
  - Increase the number of data points by interpolation after smoothing
  - Reduce the problem dimension by clustering or PCA

Effect of sampling and noise on network reconstruction

Continuous- vs discrete-time model

- The inference is based on sampled-data → we identify the matrices of the discretized system, A_d and B_d
- What is the relation between the sparsity pattern of the continuous- and discrete-time systems?

$$A_d = e^{AT_s} \qquad B_d = \int_0^{T_s} e^{A\tau} B d\tau$$

- Assume, for the sake of simplicity, that A has n distinct real negative eigenvalues

$$|\lambda_i| < |\lambda_{i+1}|, \; i = 1,\ldots,n$$

- It is then possible to find a nonsingular matrix P such that

$$A = PDP^{-1} \qquad D = \text{diag}(\lambda_1,\ldots,\lambda_n)$$

Continuous- vs discrete-time model

- The matrix A_d can be rewritten as

$$A_d = I + AT_s + \frac{(AT_s)^2}{2!} + \frac{(AT_s)^3}{3!} + \dots$$
$$= P\,\mathrm{diag}\left(e^{\lambda_1 T_s}, \dots, e^{\lambda_n T_s}\right) P^{-1}$$

- If the sampling time $T_s \ll \min 1/|\lambda_i|$, then $|\lambda_i| T_s \ll 1$ and

$$e^{\lambda_i T_s} = \sum_{k=0}^{\infty} \frac{(\lambda_i T_s)^k}{k!} \approx 1 + \lambda_i T_s \quad \Rightarrow \quad A_d \approx I + AT_s$$

Continuous- vs discrete-time model

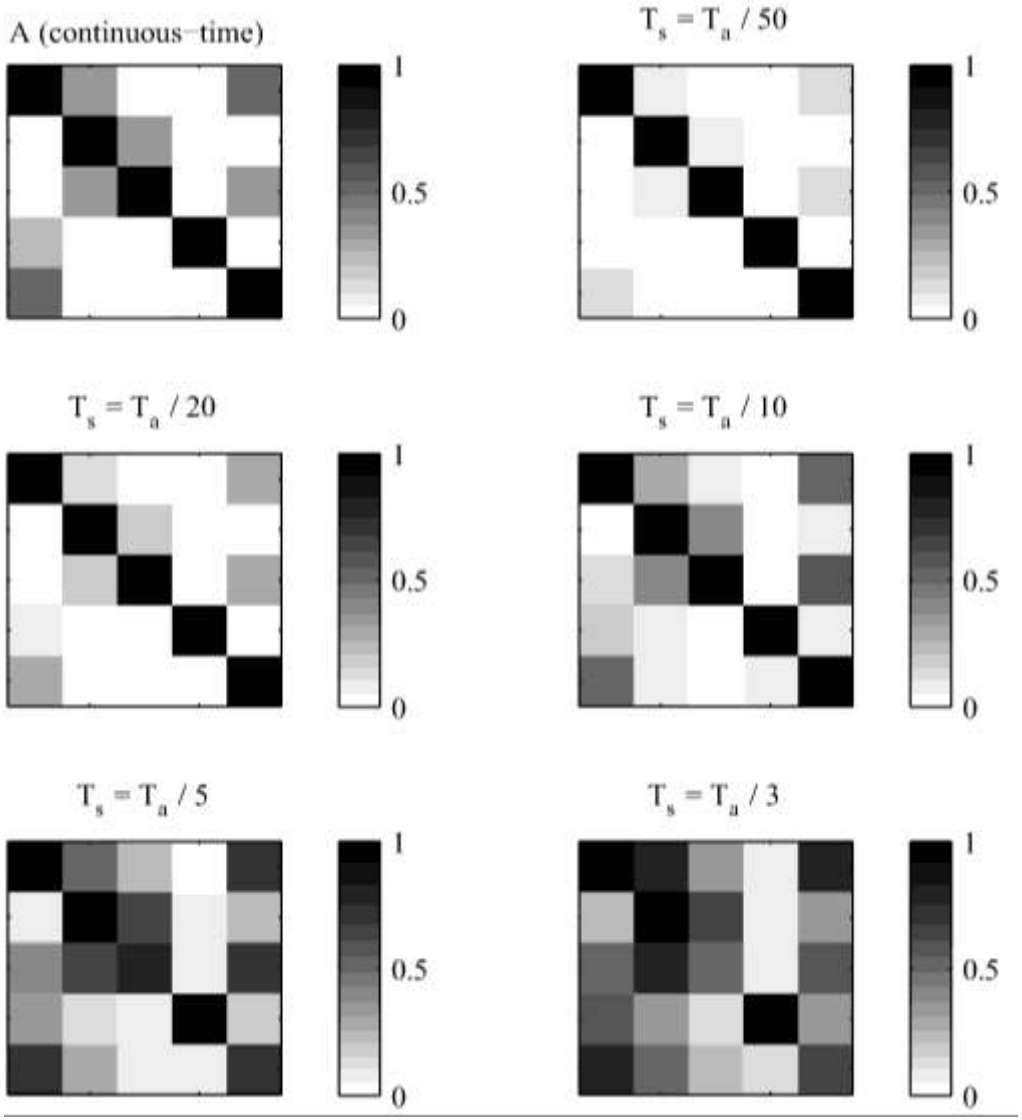
- As for the input matrix, the following approximation holds

$$B_d \approx A^{-1}\left(e^{AT_s} - I\right)B \approx A^{-1}\left(AT_s\right)B = BT_s$$

- Note that the sparsity patterns of $I + AT_s$ and BT_s are identical to those of matrices A and B of the continuous time system, with the exception of the diagonal entries of $I + AT_s$
  - This is not an issue, because the diagonal entries are not considered as targets of the inference methods: they are just assumed to be nonzero
- A typical approach to the reconstruction of the sparsity pattern: consider only the elements of A_d and B_d that fall above a certain threshold

Continuous- vs discrete-time model

- However, the diagonal elements might become dominant and mask the effect of the other ones
- A careful choice of the sampling time is paramount for the successful inference of the network
- Problem: typically, the dynamics of the system are not known beforehand



Cosentino and Bates (2011) Feedback Control in Systems Biology. CRC Press.

Effect of noise on the regression coefficients

- The quality of the model can be *a posteriori* assessed by examining the residuals
- Under the following hypotheses
  a) the linear model is a good approximation of the real system
  b) the regressors are uncorrelated
  c) the process is affected only by additive gaussian zero-mean noise

  the residuals are also gaussian zero-mean
- Moreover, it is possible to compute the covariance of the regression coefficients θ as

$$\operatorname{cov}\left(\hat{\theta}\right) = E\left\{\left(\hat{\theta} - E\left\{\hat{\theta}\right\}\right)\left(\hat{\theta} - E\left\{\hat{\theta}\right\}\right)^T\right\} = \sigma^2\left(X^T X\right)^{-1}$$

Correlation between regressors increases covariance

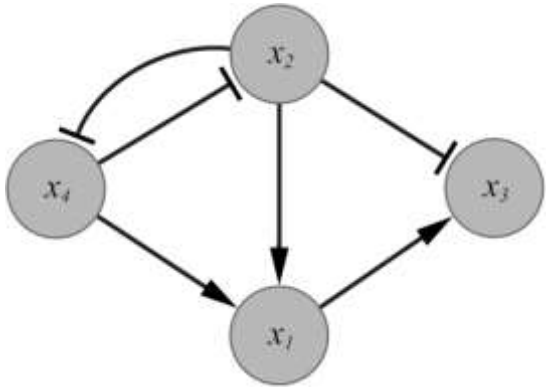
Effect of noise: a simple example

- Let us consider the static linear model with additive gaussian noise

$$y = Ax + Bu + \nu$$

where

$$A = \begin{pmatrix} 0.7035 & 0.3191 & 0 & 0.0378 \\ 0 & 0.4936 & 0 & -0.0482 \\ 0.3227 & -0.4132 & 0.2450 & 0 \\ 0 & -0.3063 & 0 & 0.7898 \end{pmatrix}, \quad B = \begin{pmatrix} -1.2260 \\ 1.1211 \\ -1.1653 \\ 0.1055 \end{pmatrix}$$



and ν is a vector of normally distributed random variables with zero mean and σ^2 variance

- Assume this describes the static relationship between the perturbation input and the steady-state level of the node variables

- Consider h=20 simulated experiments and apply LS to infer the network

Effect of noise: a simple example

- Cast the problem as

$$\hat{Y} = Z\,\Theta$$

where $\hat{Y} \in \mathbb{R}^{h\times n}, Z \in \mathbb{R}^{h\times(n+1)}, \Theta \in \mathbb{R}^{(n+1)\times n}$. The estimated system's matrices are given by $\Theta^T = [A \quad B]$

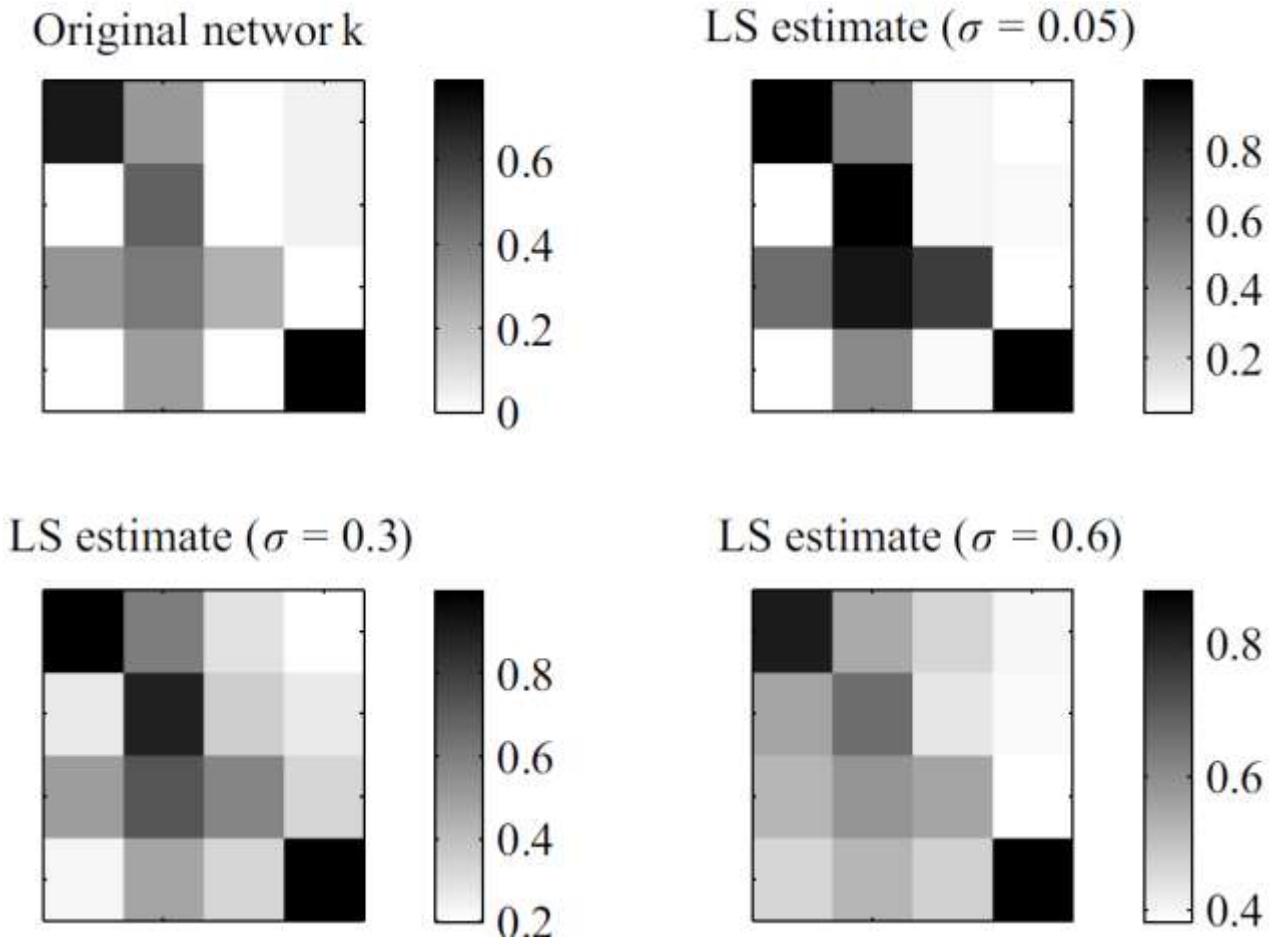
- After computing the LS estimate, we normalize the estimated adjacency matrix

$$\tilde{A}_{ij} = \frac{\hat{A}_{ij}}{\left(\|\hat{A}_{\star j}\| \cdot \|\hat{A}_{i\star}\|\right)^{1/2}}$$

$\hat{A}_{\star j}$:= j-th column of $\hat{A}$

$\hat{A}_{i\star}$:= i-th row of $\hat{A}$

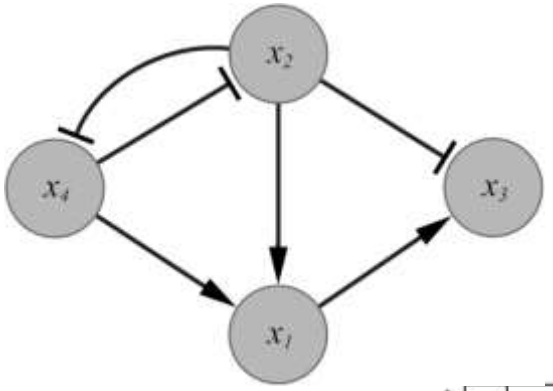
Effect of noise: a simple example



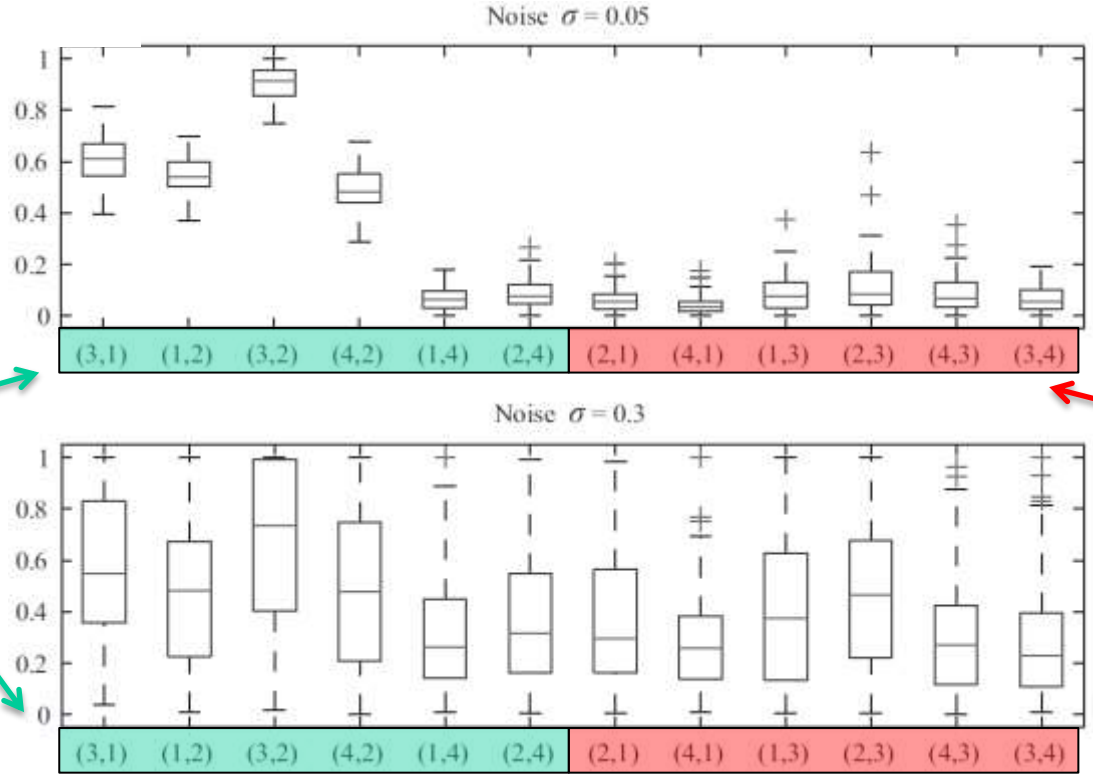
Median of the absolute values of $\hat{A}$ over 100 identification tests with different noise realizations

Cosentino and Bates (2011) Feedback Control in Systems Biology. CRC Press.

Effect of noise: edges variance



$$A = \begin{pmatrix} 0.7035 & 0.3191 & 0 & 0.0378 \\ 0 & 0.4936 & 0 & -0.0482 \\ 0.3227 & -0.4132 & 0.2450 & 0 \\ 0 & -0.3063 & 0 & 0.7898 \end{pmatrix}$$



True edges

False edges

Cosentino and Bates (2011) Feedback Control in Systems Biology. CRC Press.

Steady-state vs time-series exps

- Several types of experiments are used to unravel gene interactions
  - Gene knock-out/down, overexpression
  - RNA-interference
  - Perturbations through drug injections
  - …
- Subsequently, two types of measurements strategies are used:
  - The cellular system is measured after a long time, to ensure that a steady-state condition has been achieved
  - In other cases (especially perturbation exps), a whole time-course is taken, to study the transient behavior (expensive, less frequent)

Steady-state vs time-series

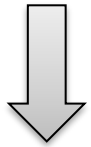
- When using time-courses, the LS solution is no longer consistent
- Let us consider the dynamical system

$$x(t_{k+1}) = A_d\, x(t_k) + B_d\, u(t_k) \qquad k = 0,\ldots,h$$

$$\hat{Y} := \begin{pmatrix} x_1(t_h) & x_2(t_h) & \cdots & x_1(t_h) \\ x_1(t_{h-1}) & x_2(t_{h-1}) & \cdots & x_2(t_{h-1}) \\ \vdots & \vdots & & \vdots \\ x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \end{pmatrix}$$

$$\hat{Z} := \begin{pmatrix} x_1(t_{h-1}) & x_2(t_{h-1}) & \cdots & x_1(t_{h-1}) & u(t_{h-1}) \\ x_1(t_{h-2}) & x_2(t_{h-2}) & \cdots & x_2(t_{h-2}) & u(t_{h-2}) \\ \vdots & \vdots & & \vdots & \vdots \\ x_1(t_0) & x_2(t_0) & \cdots & x_n(t_0) & u(t_0) \end{pmatrix}$$

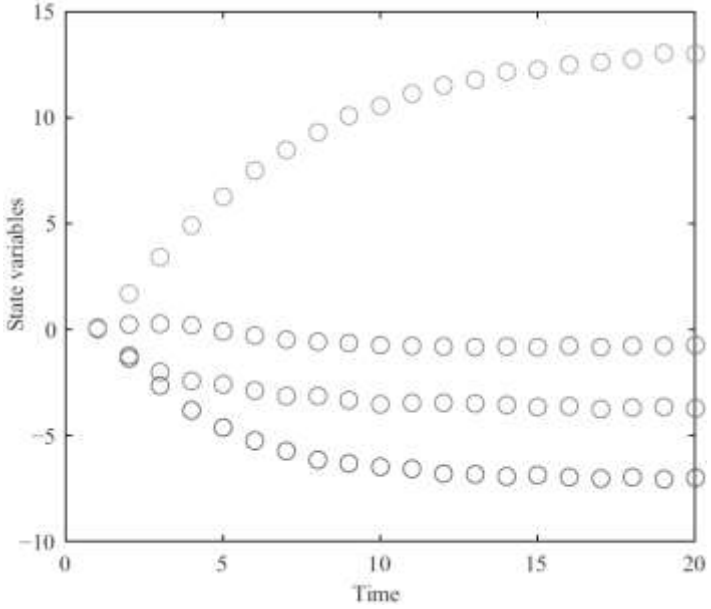
$$\hat{Y} = \hat{Z}\,\Theta$$

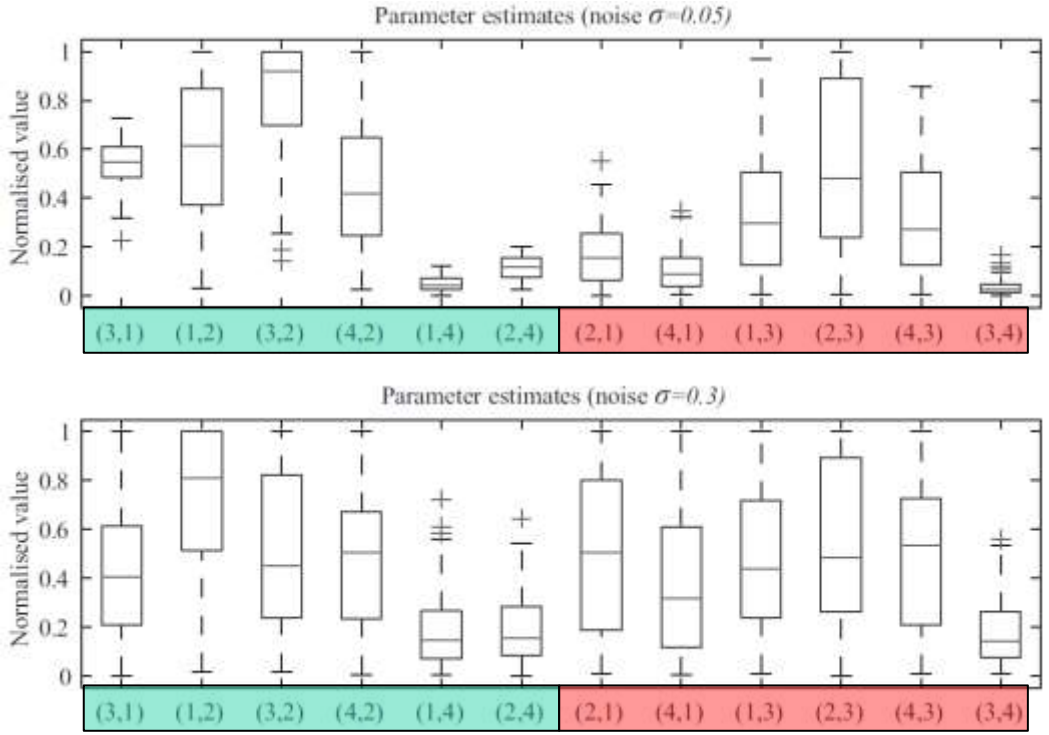


Least Squares?

Regression on time-series:example

- Take $(A_d, B_d) = (A, B)$ matrices considered in the static example before
- LS yield a worse performance wrt to the static case, even with small noise





Cosentino and Bates (2011) Feedback Control in Systems Biology. CRC Press.

Regression on time-series: drawbacks

- Issues:
  - Noise affects both sides of the relation
  - Regressors are correlated, indeed they are made up of values of the same variables at consecutive time points

$$\hat{Y} := \begin{pmatrix} x_1(t_h) & x_2(t_h) & \cdots & x_1(t_h) \\ x_1(t_{h-1}) & x_2(t_{h-1}) & \cdots & x_2(t_{h-1}) \\ \vdots & \vdots & & \vdots \\ x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \end{pmatrix}$$

$$\hat{Y} = \hat{Z}\,\Theta$$

$$\operatorname{cov}(\hat{\theta}) = \sigma^2 (Z^T Z)^{-1}$$

$$\hat{Z} := \begin{pmatrix} x_1(t_{h-1}) & x_2(t_{h-1}) & \cdots & x_1(t_{h-1}) & u(t_{h-1}) \\ x_1(t_{h-2}) & x_2(t_{h-2}) & \cdots & x_2(t_{h-2}) & u(t_{h-2}) \\ \vdots & \vdots & & \vdots & \vdots \\ x_1(t_0) & x_2(t_0) & \cdots & x_n(t_0) & u(t_0) \end{pmatrix}$$

Regression on time-series: drawbacks

- How to improve the performance?
  - Decrease the sampling time?
    - Increases correlation between samples at consecutive time points
  - Increase the observation interval?
    - Not useful if the system has already reached the steady-state: in this case, again, the additional regression vectors are correlated with the previous ones
- A possible answer is to exploit different types of experiments, however
  - Normalization of measurements taken with different experimental set-ups and techniques is problematic
- Other System Identification approaches, e.g.
  - Instrumental Variables: iterative filtering of the residuals to decorrelate them (Ljung, System Identification Theory, 1999)
    - More suitable for identification of predictive autoregressive models, computationally demanding

Total Least Squares

- The classical LS method amounts to solving the following optimization problem

$$\left\{\hat{\Theta}_{\text{ls}}, \Delta Y_{\text{ls}}\right\} := \arg\min_{\Theta, \Delta Y} \left\|\Delta Y\right\|_F$$

$$\text{s.t.} \quad X\,\Theta = Y + \Delta Y$$

- In this setting, the given data matrix X and Y are treated asymmetrically: X is assumed to be certain, whereas Y subject to additive noise

- The Total LS (TLS) method recasts the problem in a symmetric form

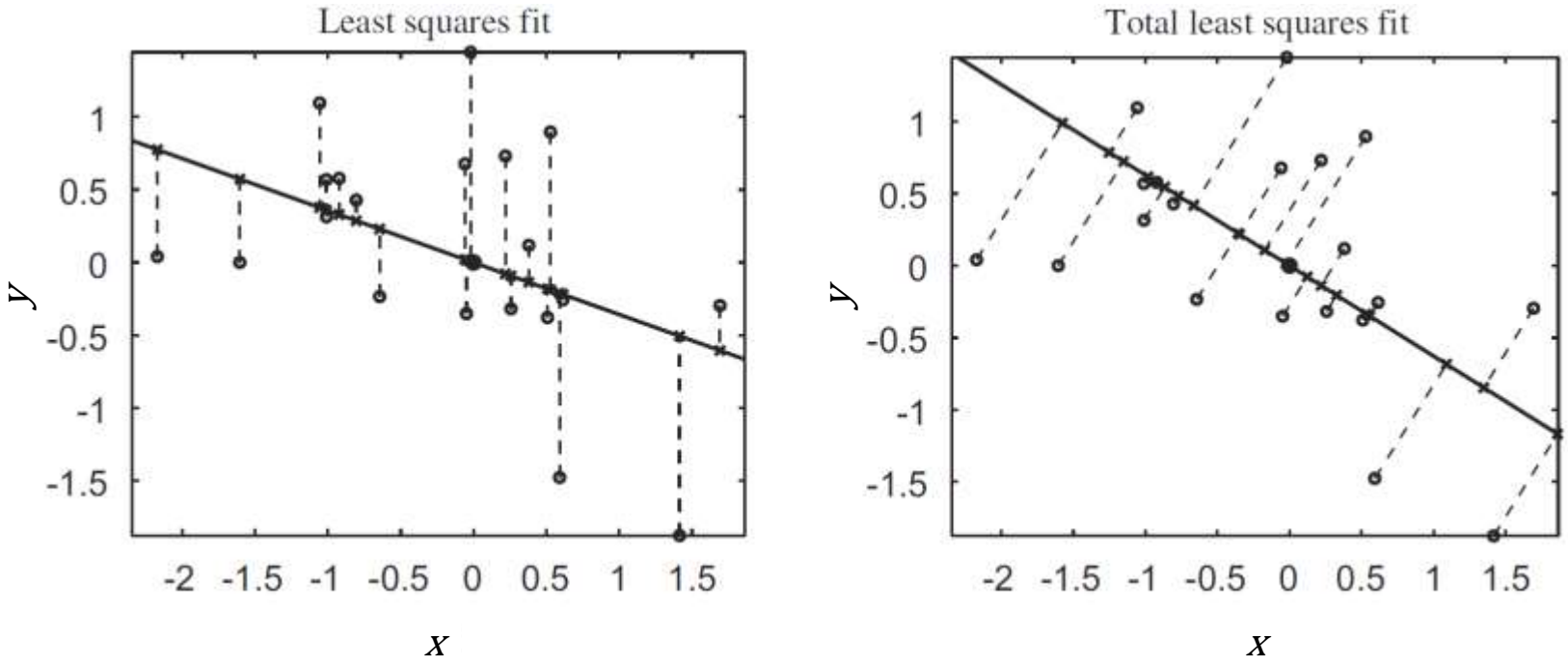
$$\left\{\hat{\Theta}_{\text{ls}}, \Delta X, \Delta Y_{\text{ls}}\right\} := \arg\min_{\Theta, \Delta X, \Delta Y} \left\|\Delta X \quad \Delta Y\right\|_F$$

$$\text{s.t.} \quad \left(X + \Delta X\right)\Theta = Y + \Delta Y$$

I. Markovsky, S. Van Huffel, Overview of total least-squares methods, Signal Processing 87 (2007) 2283–2302.

Total Least Squares

- The difference between LS and TLS is fairly evident looking at the following fits



- The solution of a TLS problems can be found (if existing) through the singular value decomposition of $[X \quad Y]$

I. Markovsky, S. Van Huffel, Overview of total least-squares methods, Signal Processing 87 (2007) 2283–2302.

Constrained Total Least Squares

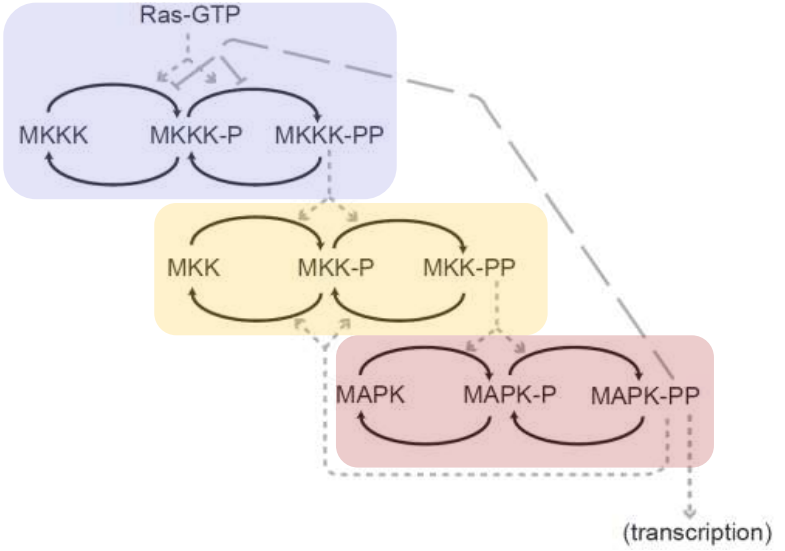
- When the data matrices contain time-series measurements of the same variables, the TLS can be specialized to a Constrained TLS problem
- CTLS preserves the information about the structure of the data matrices in the optimization matrices ΔX and ΔY, that is

$$Y + \Delta Y = (X + \Delta X)\Theta \qquad \Delta Y_i = \begin{pmatrix} \nu_i(t_h) \\ \nu_i(t_{h-1}) \\ \vdots \\ \nu_i(t_1) \end{pmatrix} \qquad \Delta X = \begin{pmatrix} \nu_1(t_{h-1}) & \nu_2(t_{h-1}) & \cdots & \nu_1(t_{h-1}) \\ \nu_1(t_{h-2}) & \nu_2(t_{h-2}) & \cdots & \nu_2(t_{h-2}) \\ \vdots & \vdots & & \vdots \\ \nu_1(t_0) & \nu_2(t_0) & \cdots & \nu_M(t_0) \end{pmatrix}$$

- The correction terms $\nu_i(t_k)$ are the optimization variables
- Drawback: no computionally effective algorithm to solve this problem!
  → limited to low-order systems

Total Least Squares application

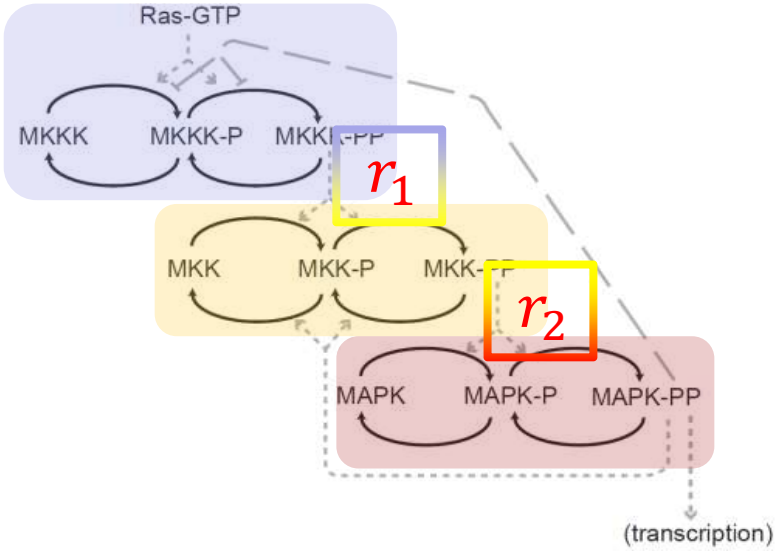
- An thourough analysis of the application of TLS to the reverse-engineering of a MAPK network can be found in (Andrec et al, 2005)
- In particular, the authors investigate the effect of noise and the probability of inferring a qualitatively wrong interaction between modules
- They cast the inference problem in the framework of *Modular Response Analysis* (MRA) (Kholodenko et al, PNAS 2002.)
- MRA allows reducing the complexity and focusing only on the communicating intermediates between modules



M. Andrec, B.N. Kholodenko, R.M. Levy, E. Sontag, Inference of signaling and gene regulatory networks by steady-state perturbation experiments: structure and accuracy. J. Theor. Biol. 232 (2005) 427–441.

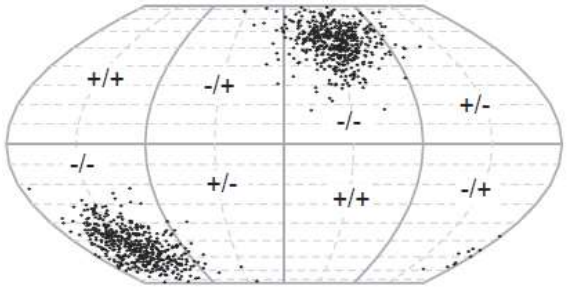
Total Least Squares application

- The polar plots show the estimated value of the connection coefficients vector (r_1, r_2)



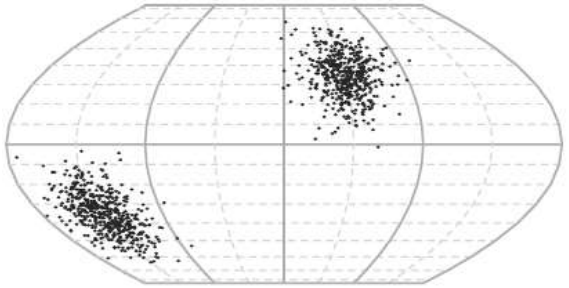
- r_1 and r_2 small ⇒ it is more likely to mis-estimate only one (panel a)
- r_1 and r_2 large ⇒ it is more likely to mis-estimate both (panel c)

True values: $r_1 = r_2 = -0.5$



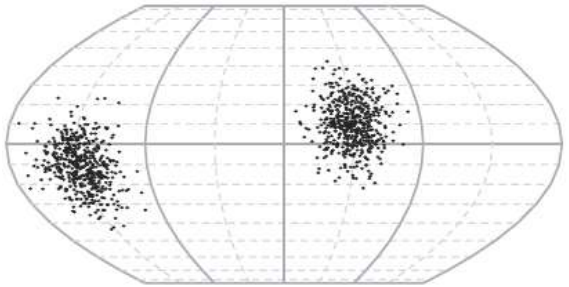
(a)

True values: $r_1 = r_2 = -1$



(b)

True values: $r_1 = r_2 = -4$



(c)

Subset selection and shrinkage methods

Inference entails subset selection

- A typical inference algorithm based on ODE models entails two phases:
  a) Selection of a subset of the regression coefficients
  b) Identification of the current model
- The process is often iterative
  - At each step, new regression coefficients are added or removed to the regression model
  - This amounts to pruning or expanding the inferred network
- Phase a) is termed *subset (or feature) selection*, where the subset elements are the regression coefficients not set to zero

Feature selection basic strategy

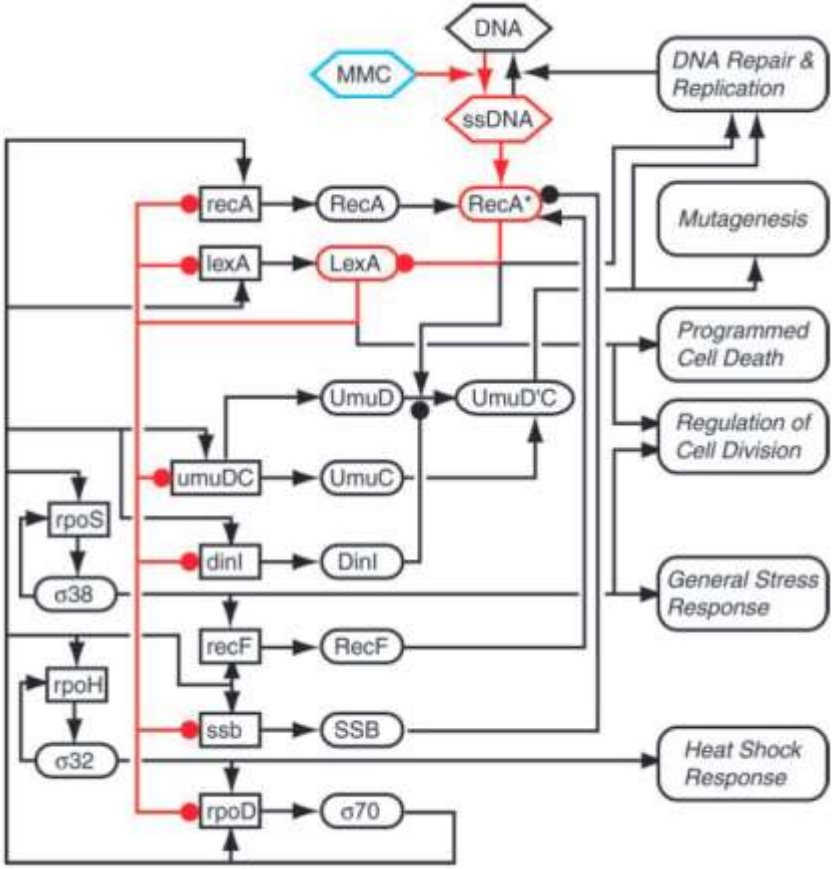
- The most basic feature selection strategy is the one adopted by the Network Inference by Reverse-engineering (NIR) algorithm (Gardner et al., Science, 2003)
  - Multiple linear regression, a maximum of $k_{\max}$ regulatory interactions is assumed for each gene
  - Exhaustive search over all the possible k-tuples, with $k < k_{\max}$
  - Eventually, the subset of regressors yielding the smallest sum of squared errors (SSE) is chosen

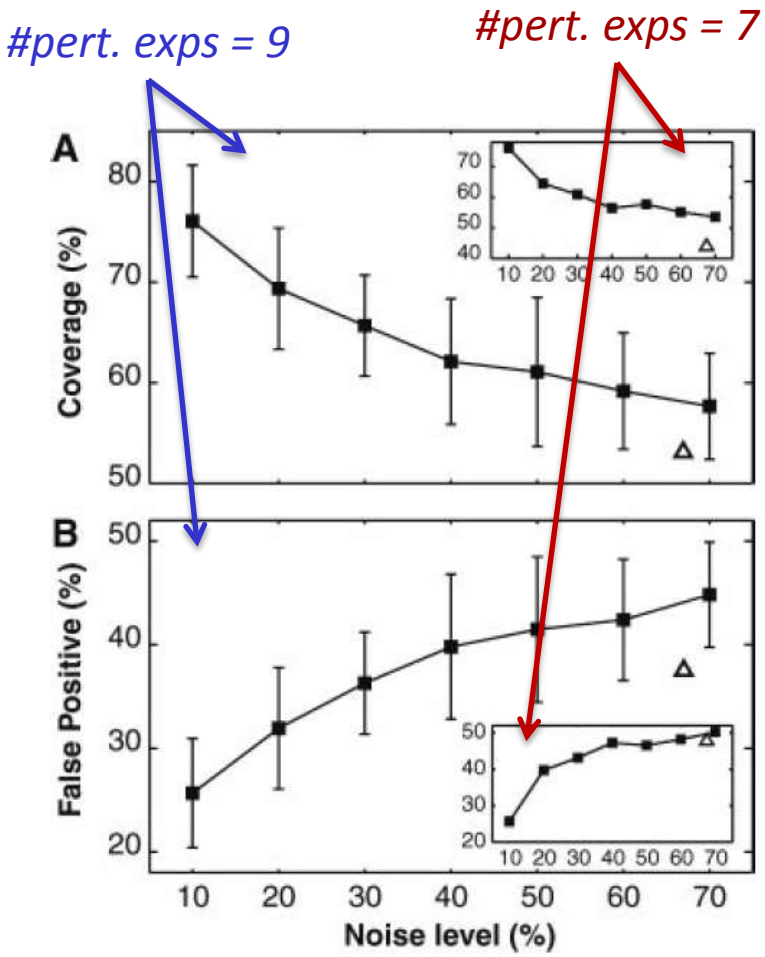
#regression problems to solve: $$n\sum_{k=1}^{k_{\max}} \frac{n!}{k!(n-k)!}$$

  - Possible only with very small networks
  - E.g. with $n = 50$, $k_{\max} = 5$ → #regr. problems $\sim 10^8$

A case-study: the SOS pathway in *E. coli*

- To keep the problem treatable, Gardner *et al*. have picked a subnetwork of only 9 genes





Forward subset selection

- Given the linear-in-the-parameter model

$$y_j = \sum_{i=1}^{M} x_{ij}\,\theta_i + \varepsilon_j$$

i: regr. coeffs index $\in [1, \dots, M]$

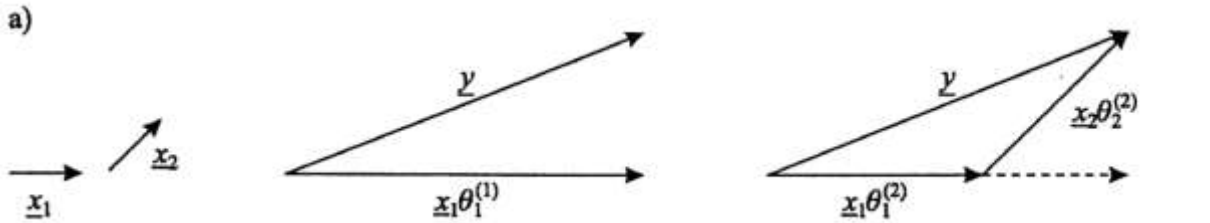
j: exps index $\in [1, \dots, N]$

- Each regressor is used in a single-regressor test and the corresponding residual is evaluated
- The regressor x_{A} yielding the best approximation of y is selected
- The part of y not explained by x_{A} is $y_{\mathrm{A}} = y - x_{\mathrm{A}}\hat{\theta}_A$
- Each of the non-selected regressors is tested against y_{A}
- The regressor x_{B} yielding the best approximation of y_{B} is selected
- ...

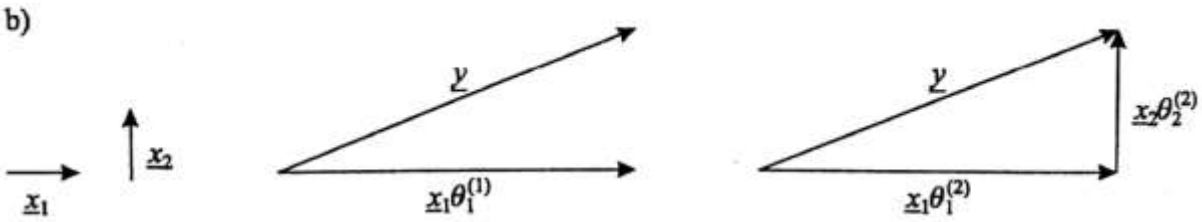
Forward subset selection: drawbacks

- FSS is a greedy algorithm: the subset at step $i+1$ includes all the elements selected at previous steps
- Major drawback: does not consider the interaction between regressors

Non-orthogonal regressors



Orthogonal regressors



- The final susbet is only suboptimal
- Computationally efficient: $M-i+1$ one-parameter regression at step i

Orthogonal LS and forward subset selection

- The OLS method is based on the orthogonalization of the regressors, which yields the equivalent model

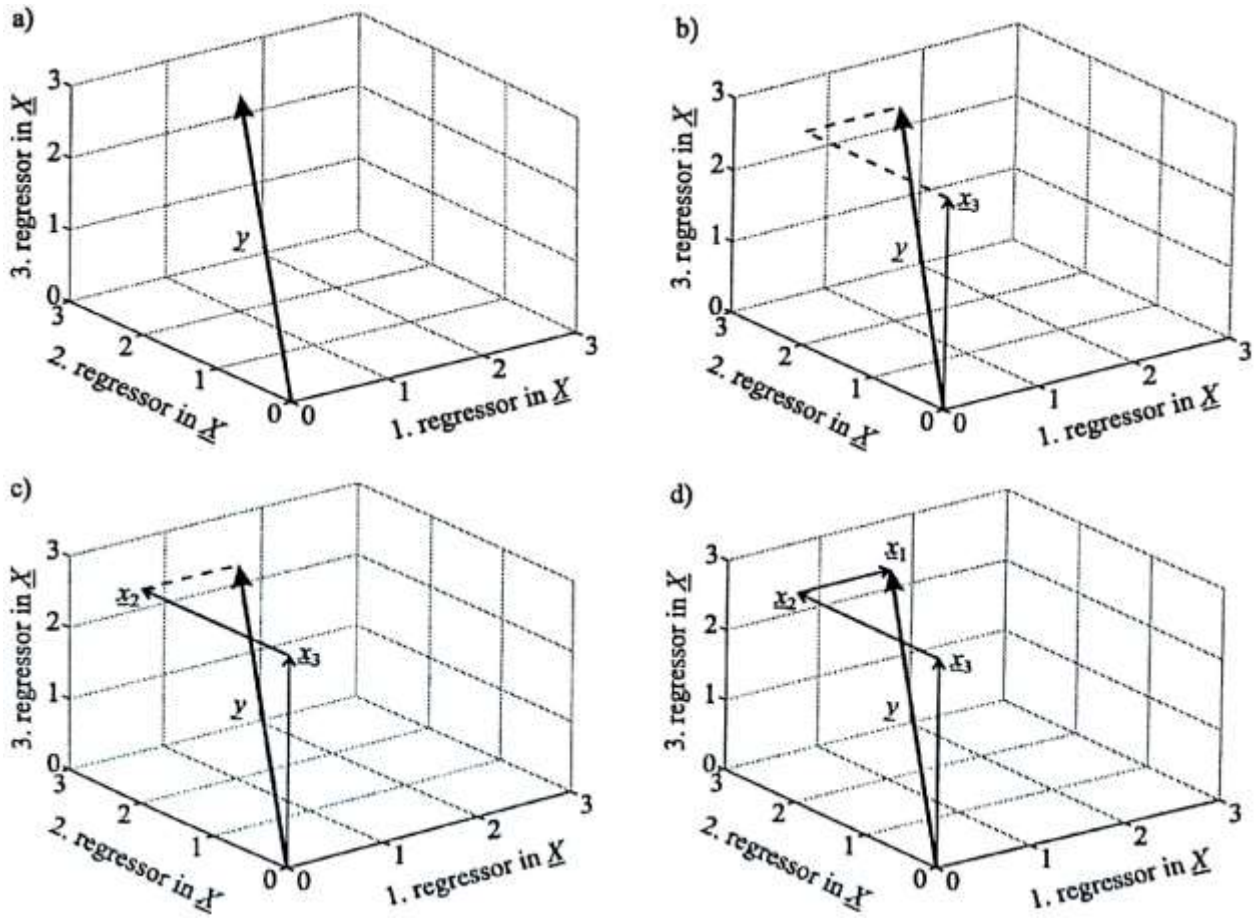
$$y_j = \sum_{i=1}^{M} w_{ij}\, g_i + \varepsilon_j \qquad\qquad w_i^T w_j = 0, \;\; \forall i \neq j$$

- The parameters of the orthogonal model can be computed as

$$g_i = \frac{\sum_{j=1}^{N} y_j w_{ij}}{\sum_{j=1}^{N} w_{ij}^2} = \left|\text{proj}_{w_i}(y)\right|$$

Forward selection with OLS

- This is the case with three orthogonal regressors



O. Nelles, Nonlinear System Identification. (2001). Springer-Verlag.

Orthogonal LS and forward subset selection

- A key advantage of the orthogonalization model is the possibility to compute the *Error Reduction Ratio (ERR)* associated to each regressor

$$ERR_i = \frac{g_i^2 \langle w_i, w_i \rangle}{\langle y, y \rangle}$$

- Forward subset selection includes the regression coefficients in the model in descending order of ERR_i value, with termination condition

$$1 - \sum_{i=1}^{p} ERR_i = \rho$$

- Unfortunately, OLS selects the coefficients of the orthogonal model, not those of the original one (which correspond to the network edges)!

Chen, S., S. A. Billings and W. Luo. 1989. Orthogonal Least Squares Methods and Their Application to Non-linear System Identification. International Journal of Control. 50:1873–1896.

Backward elimination

- Alternative to forward selection:
  - Start with a model made up of all the M regressors
  - Iterate regression removing the least significant regressor at each step
- Typically not applicable in biological network inference:
  - #number of regressors of the full model $\gg$ #data points

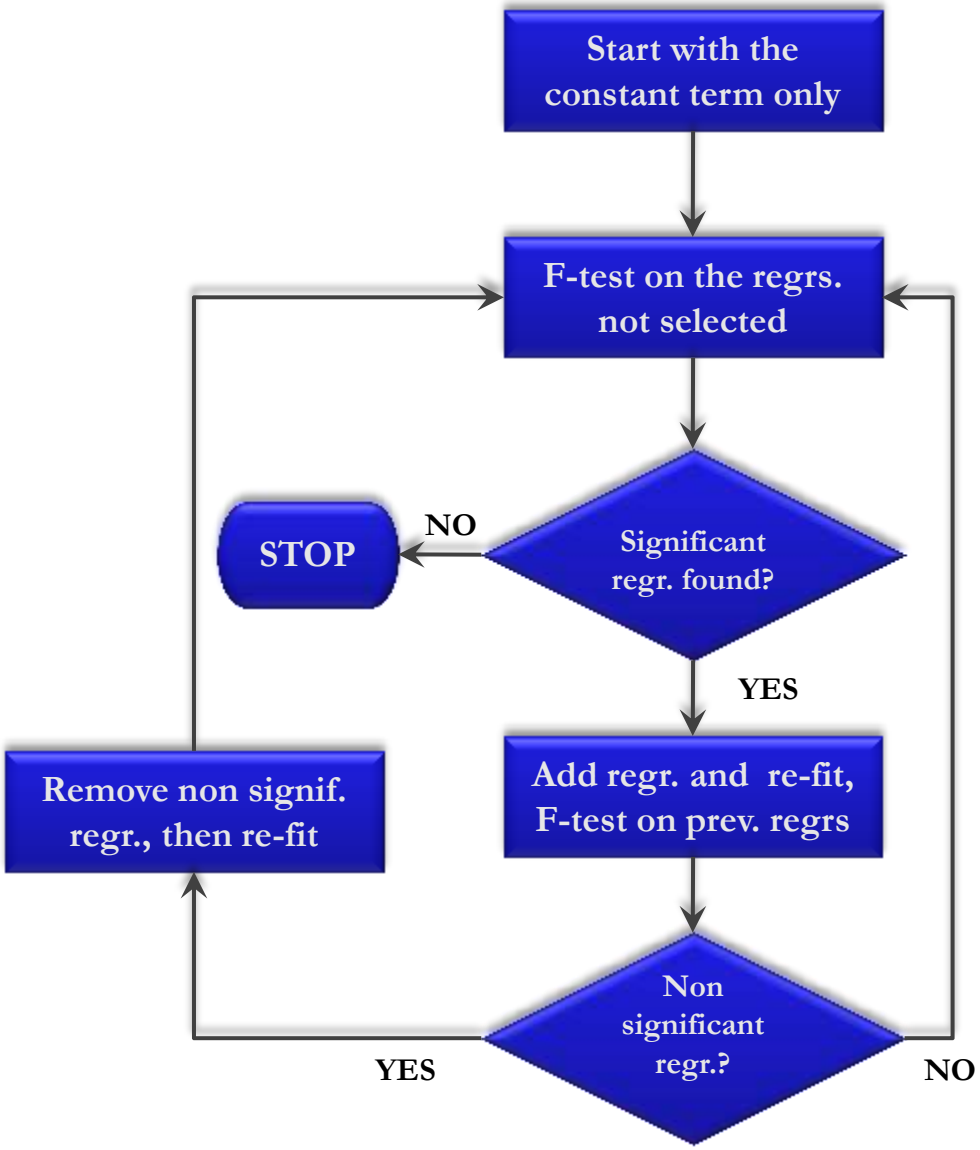
Stepwise regression

- Stepwise regression addresses the drawbacks of forward selection by allowing the removal of variables selected at previous steps
- The addition/removal are based on the Residual Sum of Squares (RSS); in particular the following normalized variables are considered

$$R_{\text{add}} = \frac{RSS_p - RSS_{p+1}}{RSS_{p+1}/(n-p-2)} \qquad R_{\text{rem}} = \frac{RSS_{p-1} - RSS_p}{RSS_p/(n-p-1)}$$

- Under gaussian noise hypothesis, these variables (approximately) exhibit a Fisher distribution
  - Their use allows a statistical significance test to be performed for each regression coefficient
  - Thresholds F_{add} and F_{rem} can be computed from the distribution, to achieve desired significance

Stepwise regression algorithm



Ridge regression

- Ridge regression is a method that introduces a penalty term on the size of the regression coefficients θ (aka *problem regularization*)

$$\min_{\theta} \left\| y - x^T \theta \right\|_2^2 + \lambda \left\| \theta \right\|_2^2$$

λ: ridge parameter

- The solution is

$$\hat{\theta} = \left(X^T X + \lambda I \right)^{-1} X^T y$$

- Note that

$$\lambda \to 0 \quad \Rightarrow \quad \hat{\theta}^{\text{ridge}} \to \hat{\theta}^{\text{ls}}$$

$$\lambda \to \infty \quad \Rightarrow \quad \hat{\theta}^{\text{ridge}} \to 0$$

- Ridge regression shrinks the coefficients towards zero

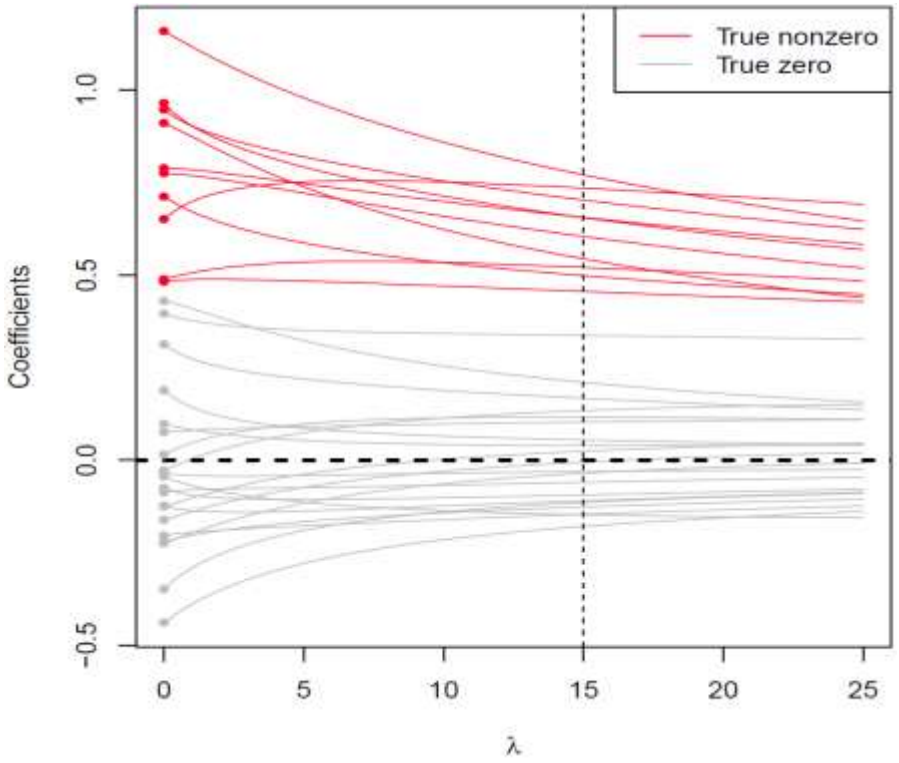
Ridge regression

- Typical behaviour of the coefficients in a ridge regression (ridge trace)

- Ridge regression does not actually implement a subset selection strategy

- However, it regularizes the problem, by imposing a lower bound on the minimum eigenvalue of $X^T X$

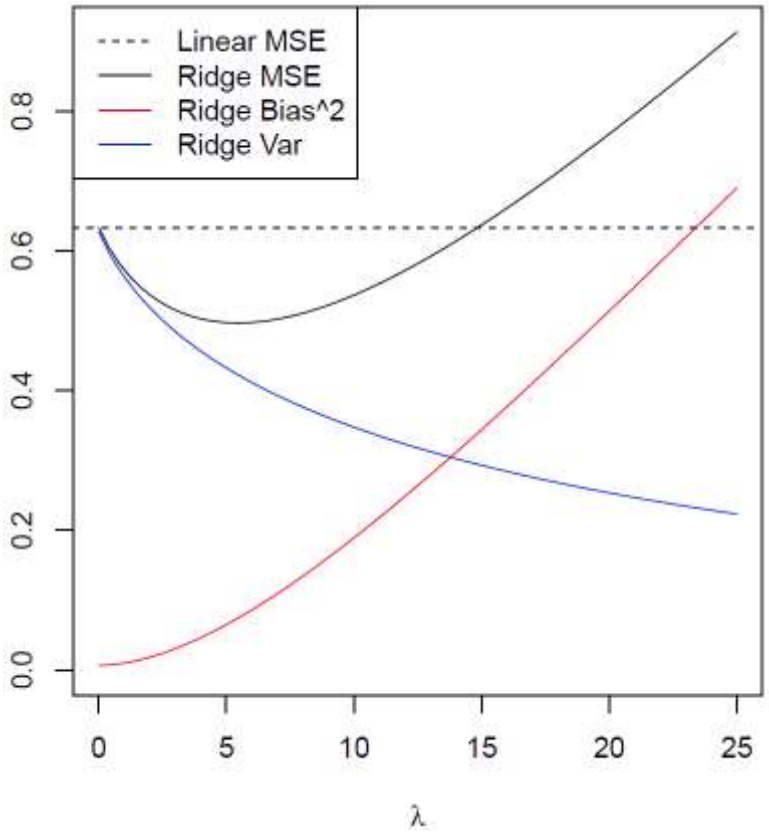
- Note: the estimate is biased
  - $\lambda \uparrow \Rightarrow$ bias $\uparrow$
  - $\lambda \uparrow \Rightarrow$ variance $\downarrow$



Ryan Tibshirani. "Data Mining" course lecture notes – Spring 2013 – Carnegie Mellon University

Ridge regression

- The major issue in the application of ridge regression is the choice of the optimal value of λ
- Above a certain value, the mean square error (MSE) becomes greater than LS
- The value of λ can be chosen via (K-fold) cross-validation methods
- However, this technique is aimed at *prediction accuracy*, not at *recovering the true model*



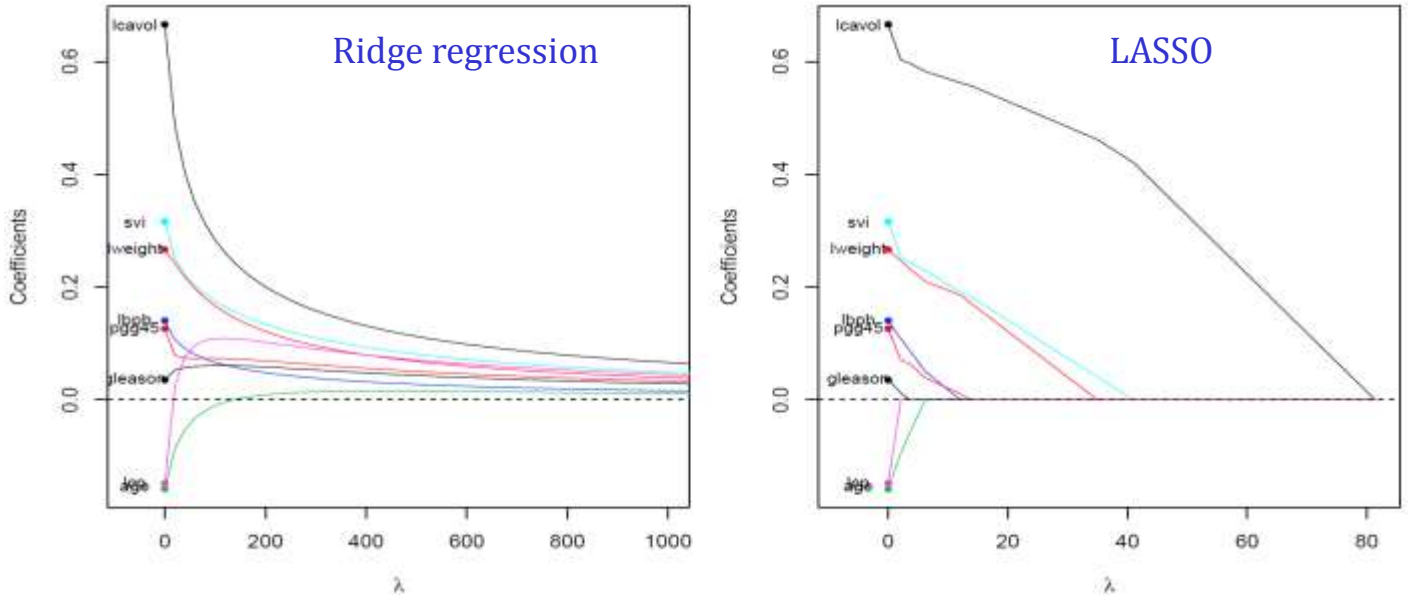
Ryan Tibshirani. "Data Mining" course lecture notes – Spring 2013 – Carnegie Mellon University

LASSO regression

- *Least Absolute Selection and Shrinkage Operator (LASSO)* is similar to ridge regression, but uses an ℓ_1 penalty term

$$\min_{\theta} \left\| y - x^T \theta \right\|_2^2 + \lambda \left\| \theta \right\|_1$$

- This causes the coefficients to shrink exactly to zero as $\lambda \to \infty$, thus implementing a true variable selection method, e.g.



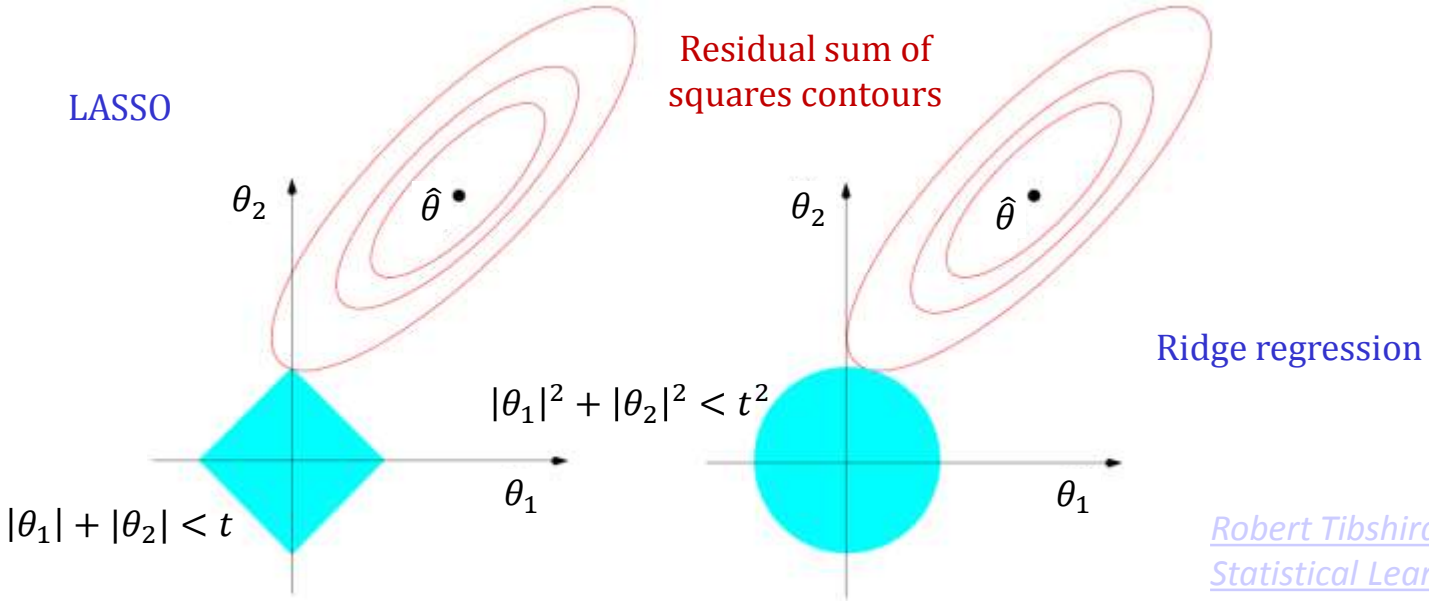
Ryan Tibshirani. "Data Mining" course lecture notes – Spring 2013 – Carnegie Mellon University

Ridge regression vs LASSO

- It is informative to look at the alternative formulation of Ridge Regression and LASSO as constrained optimization problems

$$\hat{\theta}^{\text{ridge}} = \arg\min_{\theta} \left\| y - x^T\theta \right\|_2^2 \quad \text{subject to} \quad \|\theta\|_2^2 < t^2$$

$$\hat{\theta}^{\text{lasso}} = \arg\min_{\theta} \left\| y - x^T\theta \right\|_2^2 \quad \text{subject to} \quad \|\theta\|_1 < t$$

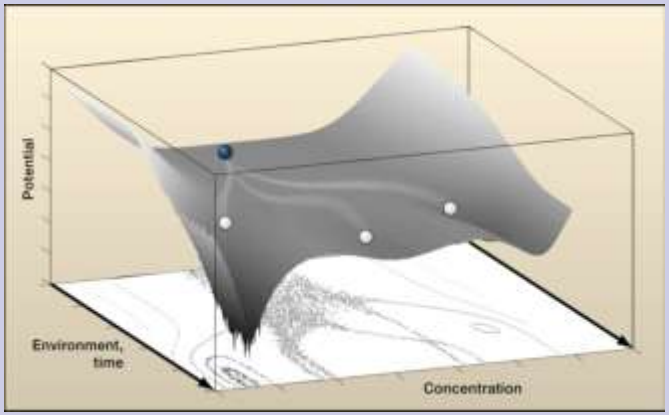
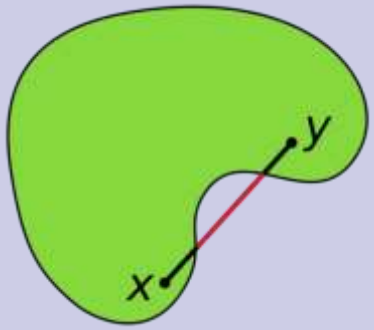
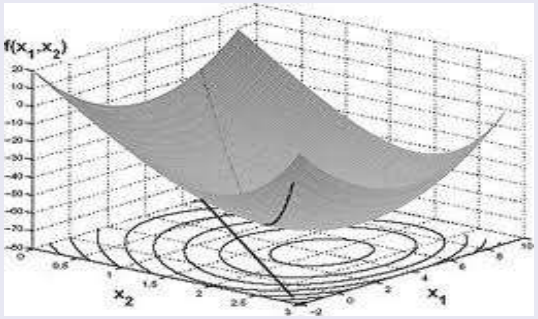
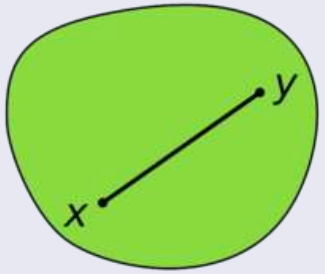


Robert Tibshirani. The Elements of Statistical Learning. Springer. 2013.

Convex optimization methods and prior knowledge exploitation

Convex Optimization

- A problem is convex when both the admissible solution space and the objective function are convex
- Convex problems can be solved very efficiently!

	Objective function (find min)	Solution space (constraints)
Non-Convex		
Convex		

Inference as a convex optimization problem

- We have seen that the regression problems can be cast as (constrained) optimization problems

$$\hat{\theta}^{\text{ls}} = \arg\min_{\theta} \left\| y - x^T \theta \right\|_2^2$$

$$\hat{\theta}^{\text{ridge}} = \arg\min_{\theta} \left\| y - x^T \theta \right\|_2^2 \quad \text{subject to} \quad \left\| \theta \right\|_2^2 < t^2$$

$$\hat{\theta}^{\text{lasso}} = \arg\min_{\theta} \left\| y - x^T \theta \right\|_2^2 \quad \text{subject to} \quad \left\| \theta \right\|_1 < t$$

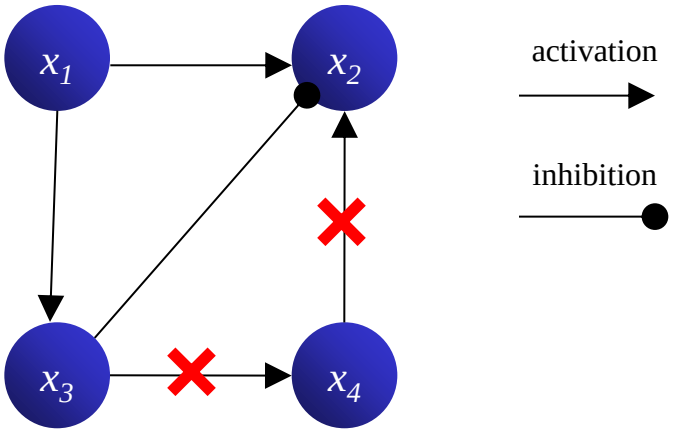
- The good news is that this type of problems is convex and is therefore solvable by means of efficient off-the-shelf numerical tools, e.g. CVX
- Why is it convenient to recast them as convex optimization problems?

Stephen Boyd, Lieven Vandenberghe. Convex Optimization. (2004) Cambridge University Press (http://stanford.edu/~boyd/cvxbook/)

Convex optimization methods

- The basic idea is to improve linear ODE–based methods by exploiting available prior knowledge about the network topology
- Indeed, it is much likely that the network topology is partially known from literature and biological databases
- Known interactions → Sign constraints on the regression coefficients → Smaller admissible solution space

	x_1	x_2	x_3	x_4
x_1		?	?	?
x_2	>		<	0
x_3	>	?		?
x_4	?	?	0	



Cosentino et al, IET Systems Biology 1(3): 164–173, 2007

Example: regression as an LMI problem

- Assume that $h+1$ experimental observations are available, then

$$\Xi := \begin{pmatrix} x(h) & \dots & x(1) \end{pmatrix} = \Theta\Omega\,,$$

where

$$\Theta = \begin{bmatrix} \hat{A} & \hat{B} \end{bmatrix}, \quad \Omega := \begin{pmatrix} x(h-1) & \dots & x(0) \\ u(h-1) & \dots & u(0) \end{pmatrix}$$

- The identification problem can be cast as

$$\min_{\Theta} \varepsilon$$
$$s.t. \qquad (\Xi - \Theta\Omega)^T (\Xi - \Theta\Omega) < \varepsilon I$$

- The constraint is quadratic in the optimization variable Θ, but using Schur complements it can be transformed into a LMI (convex constraint)

$$\begin{pmatrix} -\varepsilon I & (\Xi - \Theta\Omega)^T \\ (\Xi - \Theta\Omega) & -I \end{pmatrix} < 0$$

Plus additional inequality constraints for prior knowledge on specific edges

Cosentino et al, IET Systems Biology 1(3): 164–173, 2007

Edges Ranking w/ Preferential Attachment

- First identify a full *A* matrix, that will be used to weight the relative influence of each parameter on the system's dynamics, and normalize it

$$\tilde{A}_{ij} = \frac{\bar{A}_{ij}}{\left(\|\bar{A}_{\star,j}\| \cdot \|\bar{A}_{i,\star}\|\right)^{1/2}}$$

- At the *k*-th iteration, the edges ranking list

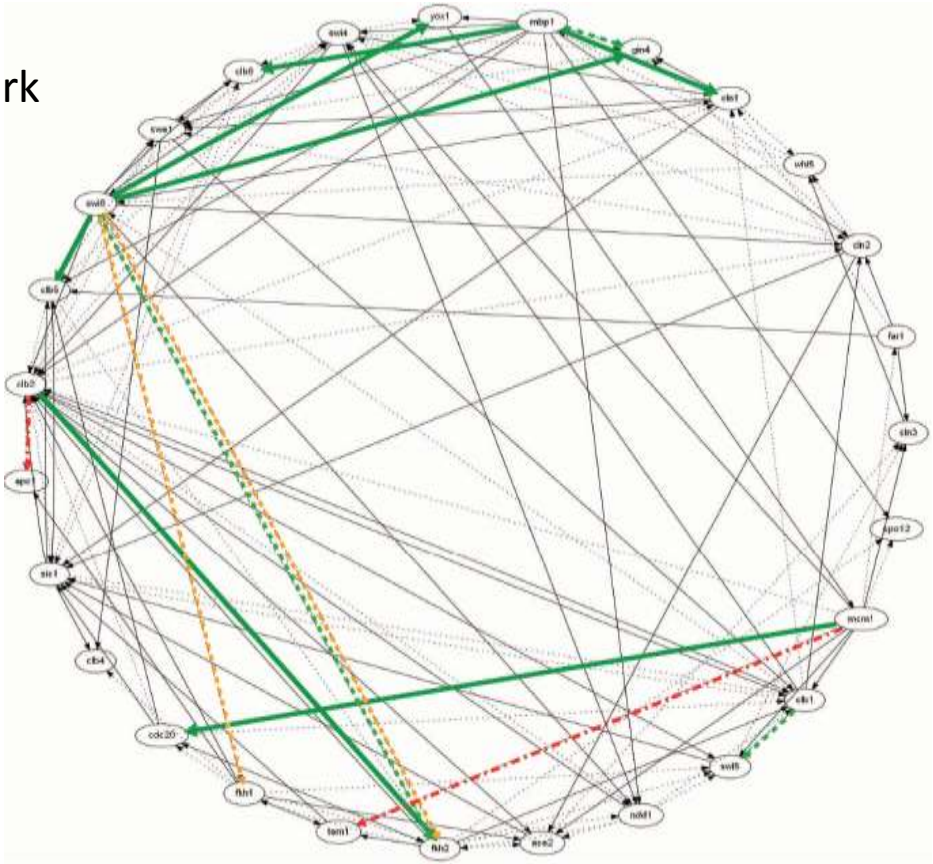
$$\tilde{G}_{ij}^{(k)} = \frac{|\tilde{A}_{ij}| p_j^{(k)}}{\sum_{l=1}^{n} p_l^{(k)} |\tilde{A}_{il}|} \qquad p_j^{(k)} = \frac{K_j^{(k)}}{\sum_{l=1}^{n} K_l^{(k)}}$$

$K_j^{(k)} \rightarrow$ Connection degree of node j

- Starting with an empty network, at each iteration insert a number of edges according to such ranking list
- Update accordingly the list of constraints and iterate the identification (stop when residuals converge)
- The score assigned by the ranking list blends the *preferential attachment* with the weights computed at the first identification step

Case study: cell cycle in yeast

- The method has been first validated by means of a large number of in silico tests
- Then, it has been applied to a subnetwork involved in the cell cycle of the yeast *S. cerevisiae*, using microarray data
- The network is composed of 27 genes, comprising genes encoding for transcription factors and for regulatory proteins (cyclins and CDKs)
- The gold standard network has been derived from the BioGRID database, it comprises 119 interactions



Montefusco et al, IET Systems Biology 4(5): 296–310, 2010

Performance Evaluation

- Performance indexes based on True/False Positives and Negatives
  - Sensitivity (Sn) (how many of the existing edges are inferred?)

$$\mathrm{Sn} = \frac{TP}{TP + FN}$$

  - Positive Predictive Value (PPV) (how reliable is a predicted interaction?)

$$\mathrm{PPV} = \frac{TP}{TP + FP}$$

- The performance indexes are computed taking into account both the directed and the undirected inferred networks

Results w/o prior knowledge

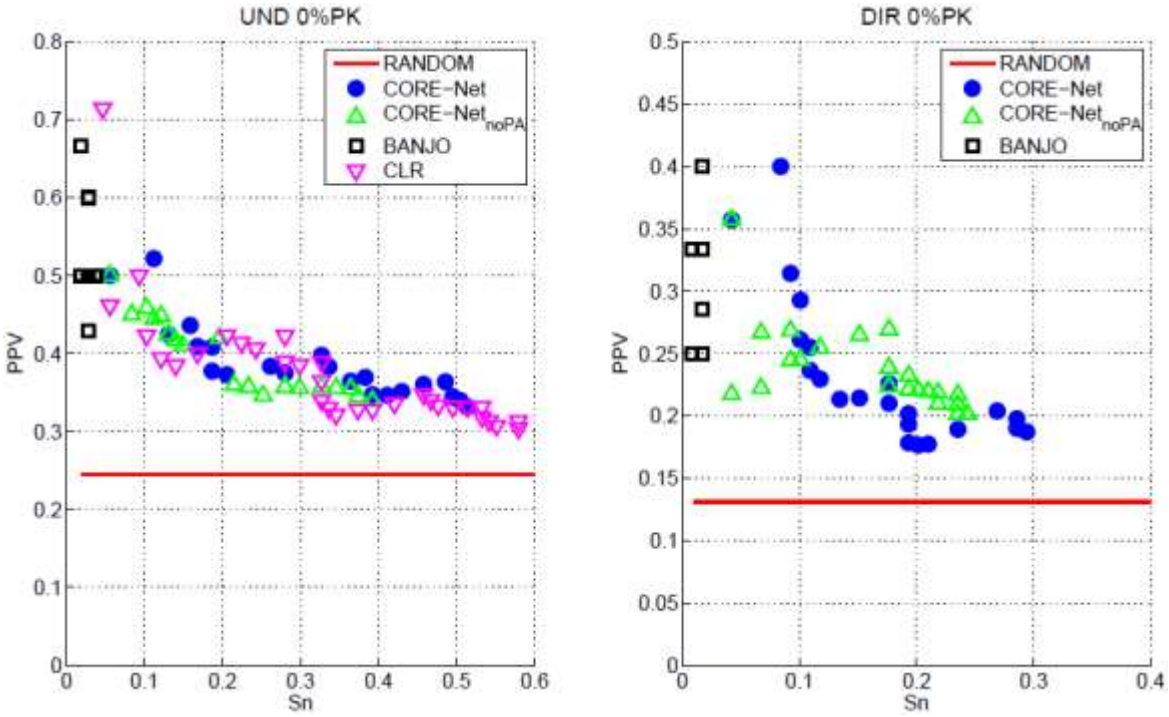


Figure 5: Results for the cell cycle regulatory subnetwork of *Saccharomyces cerevisiae* obtained by the different techniques, without assuming prior knowledge (PK=0%).

Montefusco et al, IET Systems Biology 4(5): 296–310, 2010

Results w/ prior knowledge

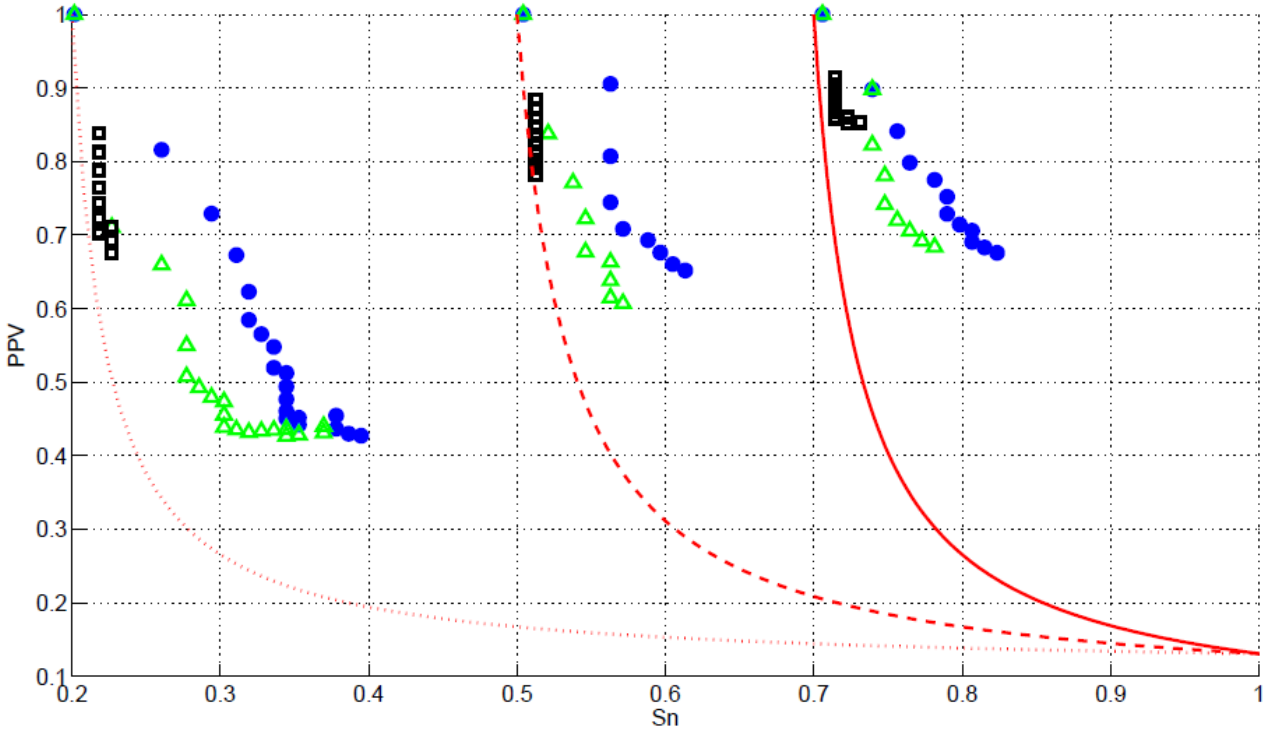


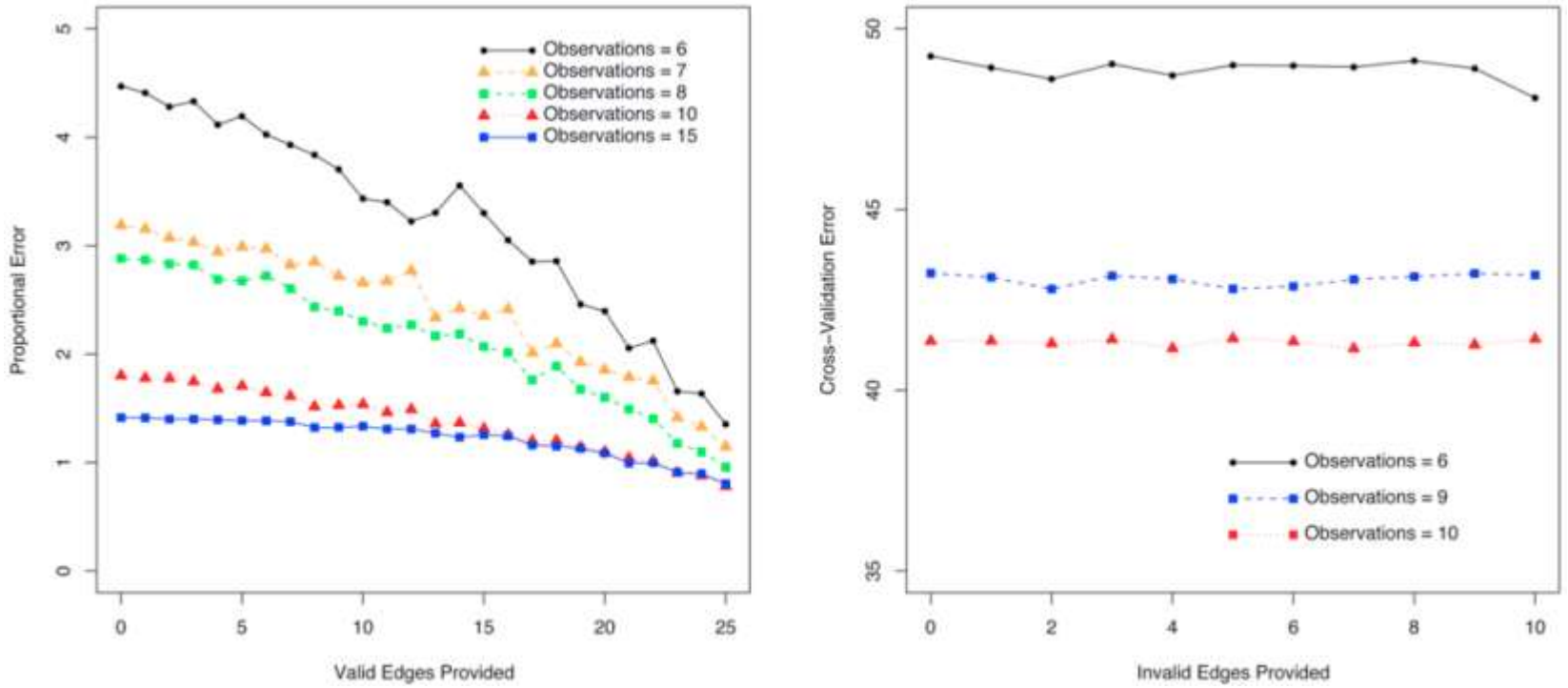
Figure 6: Results for the cell cycle regulatory subnetwork of *Saccharomyces cerevisiae* assuming different levels of prior knowledge (PK=20,50,70%): random reconstruction algorithm with 20% PK (···), 50% PK (--) and 70% PK (–) and performance of CORE–Net (•), CORE–Net$_{\text{noPA}}$ (△) and Banjo (□).

Montefusco et al, IET Systems Biology 4(5): 296–310, 2010

Example: Christley et al, PLoS ONE 2009

- Use of a penalization term for the elements that are not present in a prior information matrix W_0

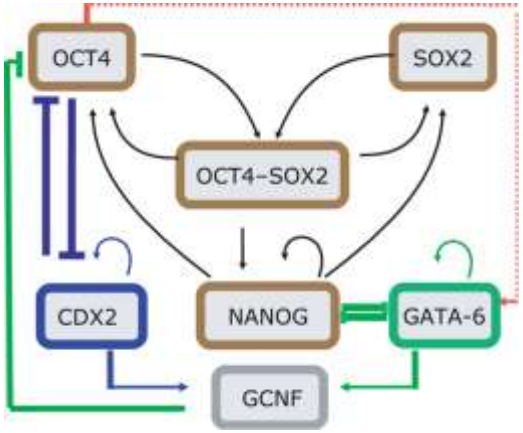
$$\hat{W} = \arg\min_{W} g(W) = f(W) + \alpha \| W \|_1 + \beta \left\| W \circ W^0 \right\|_1$$



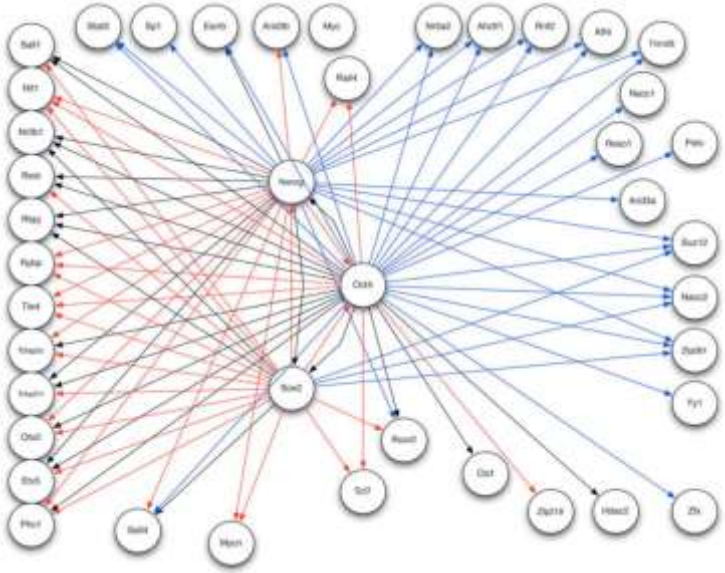
Christley et al. (2009) Incorporating Existing Network Information into Gene Network Inference. PLoS ONE 4(8):e6799

Example: Christley et al, PLoS ONE 2009

- Application: inference of the interactors of a core regulatory module of embryonic stem cells fate
- Using prior information
  - 34 known edges are retained (only 25 w/o p.i.)
  - The core module is preserved (not w/o p.i.)



w/ prior information

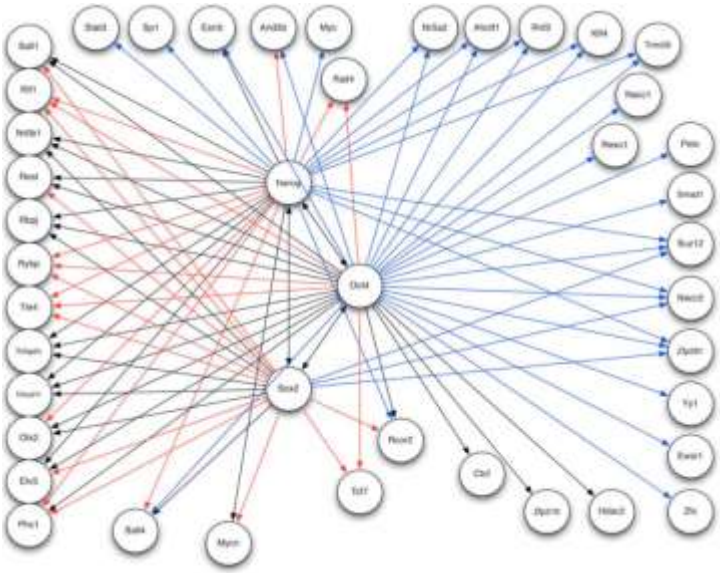


Black: prior information

Red: false positive

Blue: novel interaction

w/o prior information

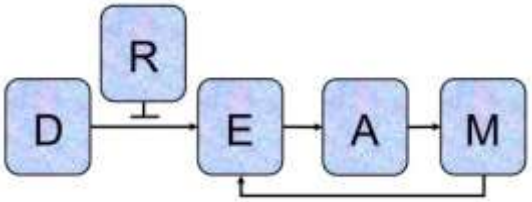


Christley et al. (2009) Incorporating Existing Network Information into Gene Network Inference. PLoS ONE 4(8):e6799

Assessment of network inference methods: the DREAM project

How to assess inference methods

- Dialogue for Reverse Engineering Assessment and Methods (DREAM)
  - Catalyze interaction between experiments and theory in
    - Cellular network inference
    - Quantitative model building in Systems Biology
- IMPROVER
  - Enhanced assessment of complex scientific processes
  - Developmente of robust, repeatable and recognized methodology for
  - Verification of correctness of basic assumptions and methods used in Systems Biology

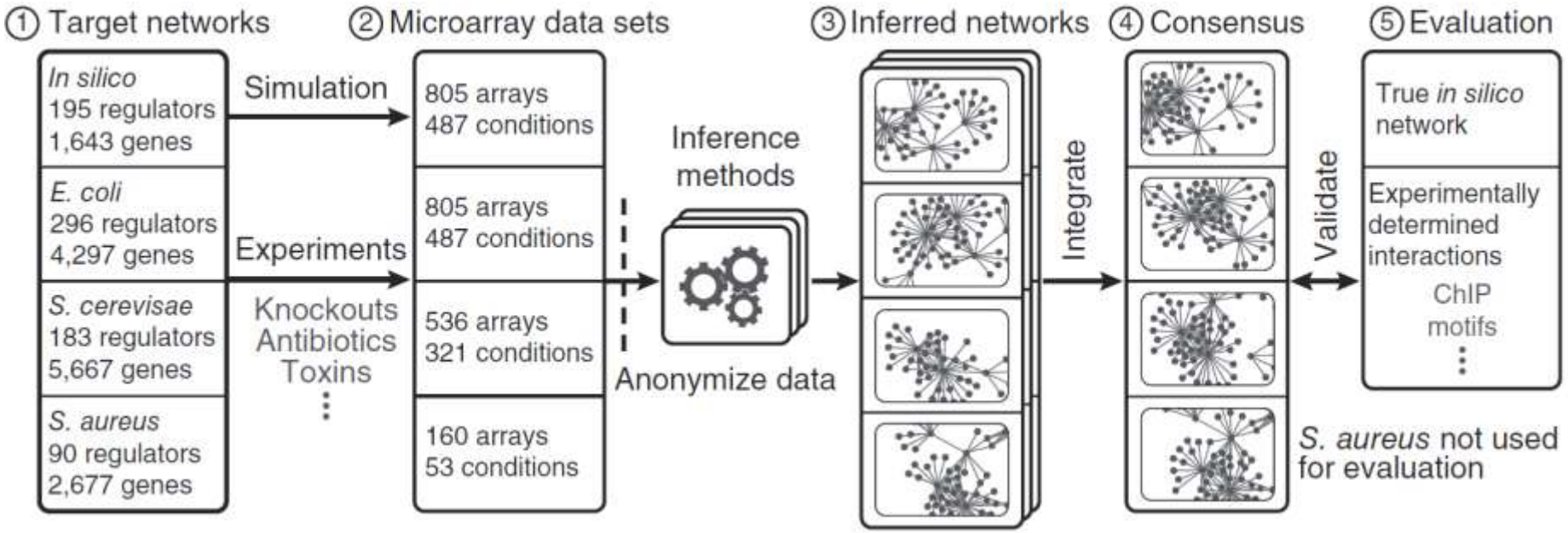


IMPROVER

The DREAM challenges

- DREAM is based on annual challenges (since 2007) comprising
  - Gene network inference
  - Protein-protein network inference
  - Gene expression prediction
  - Signaling response prediction
  - Transcription factos – DNA motif recognition
  - Peptide recognition domain specificity
  - Systems Genetics (phenotype prediction from genetic screening)
  - Parameters estimation for biomolecular models
  - …

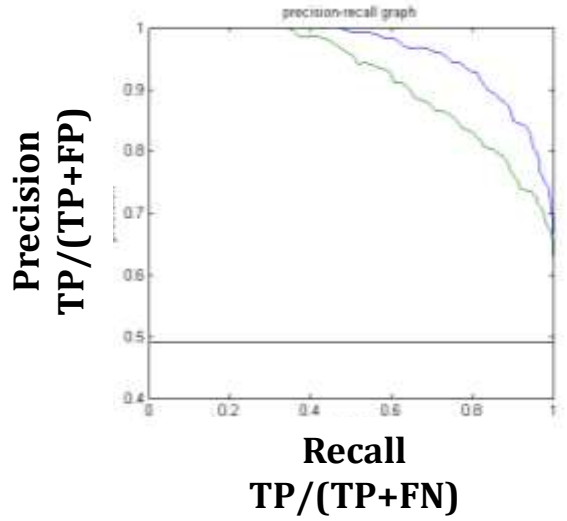
Example: the DREAM5 (2010) network inference challenge

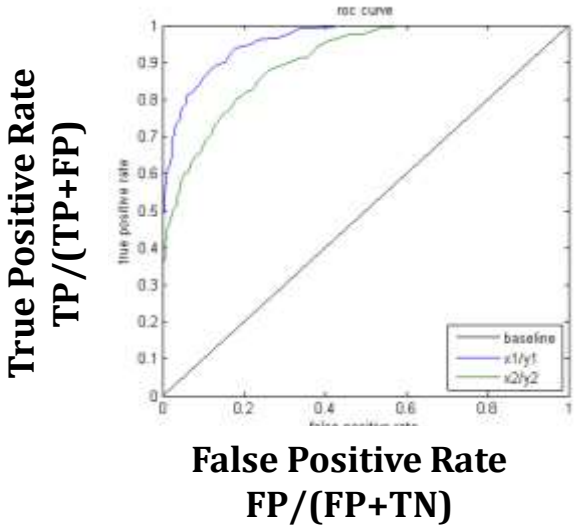


Marbach et al., Wisdom of crowds for robust gene network inference. Nature methods 9, 796–804 (2012).

Performance metrics for network inference

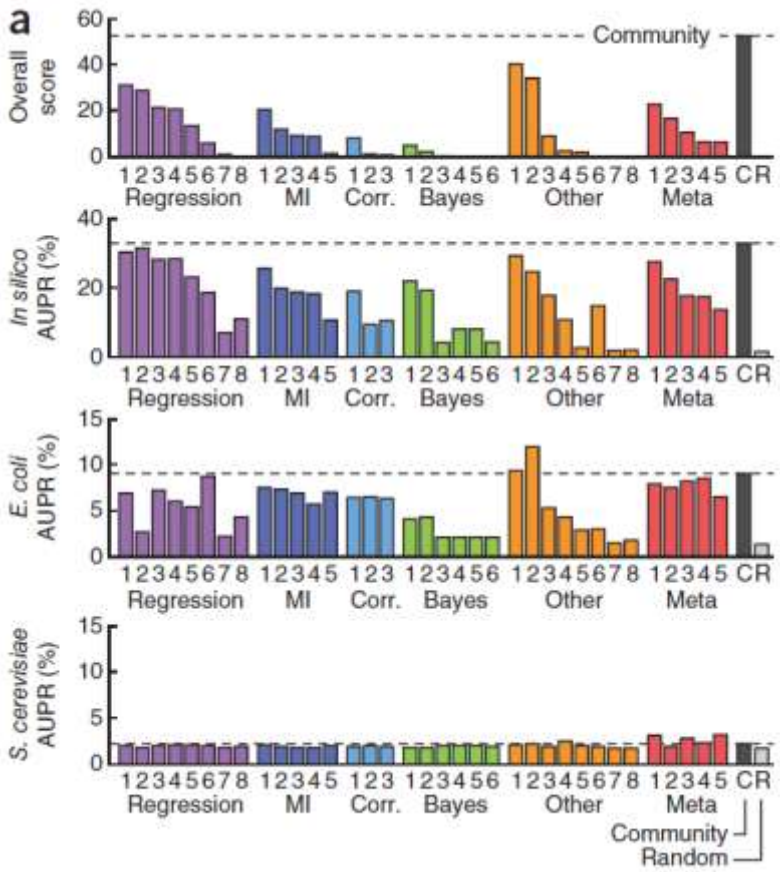
- The performance evaluation is based on the
  - Area Under Precision-Recall curve (AUPR)
  - Area Under Receiver-Operator-Characteristic (AUROC)
- A probability distribution for these value is generated experimentally
- The p-values associated with the inferred network are computed
- The Overall Score is computed based on the p-values of all the predictions

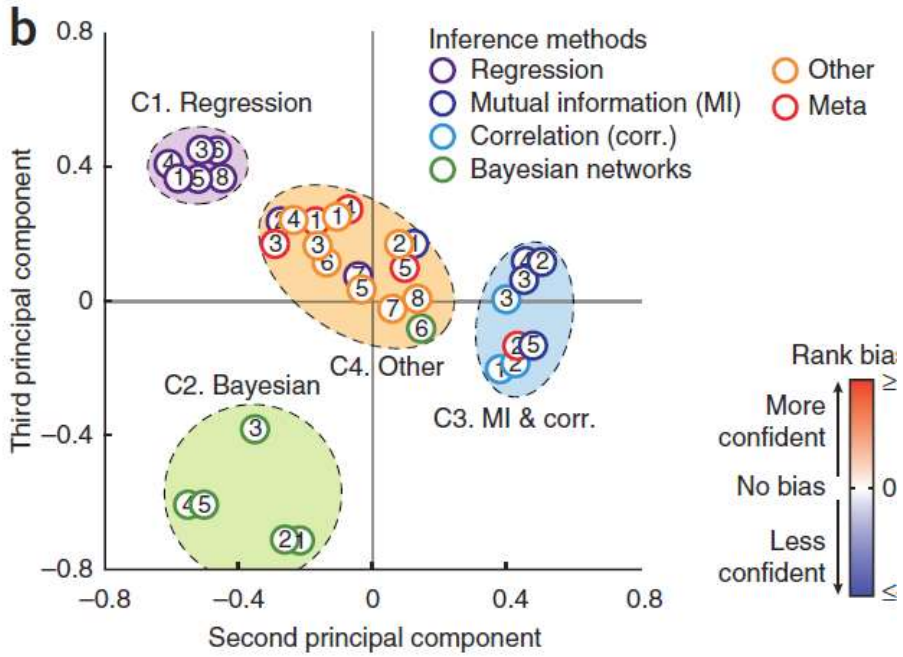




Wisdom of Crowds in Network Inference

- The DREAM Consortium, (Nature Methods, 2012) showed complementarity of different approaches

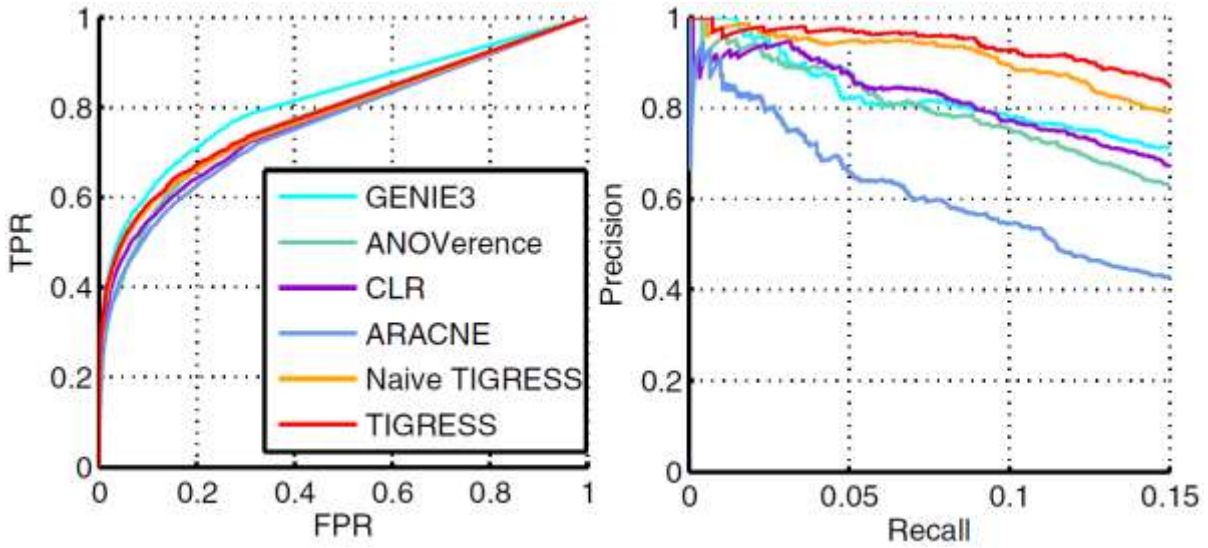




Marbach et al., Wisdom of crowds for robust gene network inference. Nature methods 9, 796–804 (2012).

Top performing regression method in DREAM 5

- The top-performing method in DREAM5 network inference challenge is based on LASSO regression plus
- *Stability selection*:
  - Repeat LASSO many times (bootstrapping the training dataset)
  - Compute frequency of selection for each edge across all runs



Haury et al., TIGRESS: Trustful Inference of Gene REgulation using Stability Selection . BMC Systems Biology 2012, 6:145.

Conclusions

- Biological network inference (and Systems Biology at large) is a very fast growing and highly interdisciplinar field

- The present time is very favorable, we are at the beginning of a revolution in biosciences, which are shifting from qualitative to quantitative disciplines

  - 👍 Good for motivated students looking for a promising research field

  - 👍 👎 Requires (a bit of) biological background and theoretical and computational tools beyond the typical systems and control theory curriculum

- Systems and Control Theory cannot miss the chance to play a key role in this revolution!

That's all folks!

- Thank you for the attention…