Inferring regulatory networks from microarray data

Lecture 3

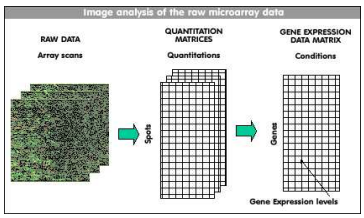
Claudio Altafini

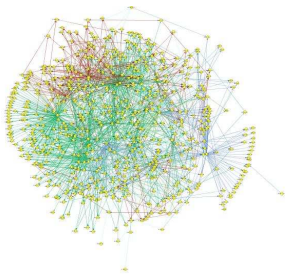
SISSA

http://people.sissa.it/∼altafini

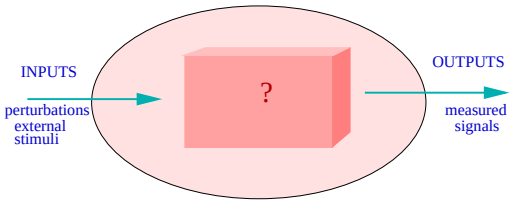
The “reverse engineering” paradigm

- task:





- method:



- measured signals: [mRNA], [proteins] [metabolites]
  - global response: measure the entire “state” vector
    - time series (e.g. cell cycle)
    - single time point (e.g. steady state)
- perturbations: experimental interventions that alter the state of interest

Network inference algorithms

- a few methods
  1. BAYESIAN NETWORK
     - attains a probabilistic graph through a bayesian learning
     - (exact) complexity: superexponential
  2. ASSOCIATION NETWORKS
     - learns a graph through a “similarity measure”
     - polynomial complexity
  3. LINEAR ODES MODELS
     - linear complexity
     - suffers from underdetermination
     - model-dependent
- other methods (not discussed): boolean, automata & formal languages, PDEs, stochastic master eq. etc....

Association networks

Association networks: similarity measures

- given n genes $X_1, \ldots, X_n$
- given a set of m expression profiles
- compute a similarity measure between the genes
- associate through edges in a graph genes with the highest similarity measure
- which similarity measure?

1. Pearson correlation of X_i and X_j

$$R(X_i, X_j) = \frac{\sum_{\ell=1}^{m}(x_i(\ell) - \bar{x}_i)(x_j(\ell) - \bar{x}_j)}{(n-1)\sqrt{v_i v_j}} \in [-1, 1]$$

   - linear
   - variants: Spearman correlation (for the ranks $\Longrightarrow$non-parametric)
2. Mutual information

Mutual information

- X_i = discrete random variable with alphabet $\mathcal{A}$
- Shannon entropy

$$H(X_i) = -\sum_{\phi \in \mathcal{A}} p(\phi) \log p(\phi), \quad \text{where } p(\phi) = Pr(X_i = \phi), \quad \phi \in \mathcal{A}$$

- joint entropy of X_i, X_j

$$H(X_i, X_j) = -\sum_{\phi,\, \psi \in \mathcal{A}} p(\phi, \psi) \log p(\phi, \psi)$$

- Mutual information of X_i and X_j

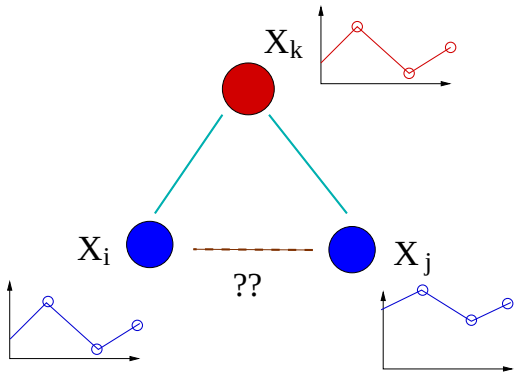
$$I(X_i; X_j) = \sum_{\phi,\, \psi \in \mathcal{A}} p(\phi, \psi) \log \frac{p(\phi, \psi)}{p(\phi)p(\psi)} \geqslant 0$$

- when the joint probability factorizes, the MI vanishes

$$p(\phi, \psi) = p(\phi)p(\psi) \quad \Longrightarrow \quad I(X_i; X_j) = 0.$$

Conditioned similarity measures

- both $R(X_i, X_j)$ and $I(X_i; X_j)$ cannot distinguish between direct and indirect interactions
- $\Longrightarrow$graph constructed will have *many* false positives
- indirect interactions: $\exists\ X_k$ that explains all the correlation between X_i and X_j?



if "yes" then extract the information due to X_k by means of conditioning

Partial Pearson correlation

- 1st order partial Pearson correlation

$$R(X_i, X_j \mid X_k) = \frac{R(X_i, X_j) - R(X_i, X_k)R(X_j, X_k)}{\sqrt{(1 - R^2(X_i, X_k))(1 - R^2(X_j, X_k))}} \in [-1, 1]$$

- take the minimum w.r.t all X_k

$$\boxed{R_{C_1}(X_i, X_j) = \min_{k \neq i,j} |R(X_i, X_j \mid X_k)|}$$

- if $R_{C_1}(X_i, X_j) \simeq 0$
  $\Longrightarrow$ $\exists\ k$ s.t. X_i and X_j are conditionally independent
  $\Longrightarrow$ Markov triple $X_i \longleftrightarrow X_k \longleftrightarrow X_j$
  $\Longrightarrow$ no edge between X_i and X_j in the graph

Conditional mutual information

- conditional entropy of X_j given X_i

$$H(X_j \mid X_i) = H(X_i, X_j) - H(X_i)$$

- conditional mutual information given X_k

$$I(X_i; X_j \mid X_k) = H(X_i \mid X_k) - H(X_i \mid X_j, X_k) \geqslant 0$$

- take the minimum w.r.t all X_k

$$\boxed{I_C(X_i; X_j) = \min_{k \neq i,j} I(X_i; X_j \mid X_k)}$$

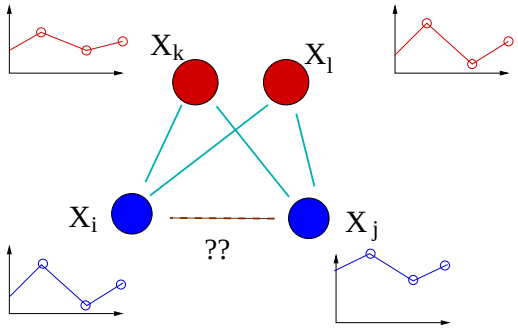
- if $I_C(X_i; X_j) \simeq 0$
  $\Longrightarrow$ $\exists\, k$ s.t. X_i and X_j are conditionally independent
  $\Longrightarrow$ Markov triple $X_i \longleftrightarrow X_k \longleftrightarrow X_j$
  $\Longrightarrow$ no edge between X_i and X_j in the graph

Association networks v.s. Bayesian netowrks

- Bayesian network *structure inference* problem: try to learn the positions of the edges on a graph
  - exponential complexity
  - looks for “true” dependences
- Association networks: guarantees that some edges are missing
  - $R(X_i, X_j) \simeq 0 \Longrightarrow X_i$ and X_j independent
  - $R((X_i, X_j) \neq 0 \not\Rightarrow X_i$ and X_j linked, since $R(X_i, X_j | X_k)$ could be $\simeq 0$
  - $\Longrightarrow$guarantees only independencies
  - polynomial complexity $R \in O(n^2)$, $R_{C_1} \in O(n^3)$

Higher order conditioning

- Even if $R(X_i, X_j \mid X_k)$ is high, it could be $R(X_i, X_j \mid X_k, X_\ell) \simeq 0$



- $\Longrightarrow X_k$ and X_ℓ "explain" all the correlation between X_i and X_j
- if I could condition over $n-2$ variables: "true" independent correlation between X_i and X_j
- complication: $O(n^n)$ untreatable

Graphical Gaussian modeling

- theory of "Graphical modeling"

  D. Edwards "Introduction to graphical modelling", Springer, 2000
- given R
- $\Omega(\omega_{ij}) = R^{-1}$ = concentration matrix
- true partial correlation

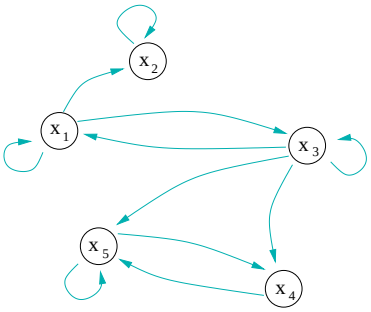
$$R_{C_{\text{all}}}(X_i, X_j) = -\frac{\omega_{ij}}{\sqrt{\omega_{ii}\omega_{jj}}}$$

- $R_{C_{\text{all}}}(X_i, X_j)$ high $\Longrightarrow X_i$ and X_j are linked
- complication: $m < n$ (n. of experiments < n. of genes)
- $\Longrightarrow$ R is normally not full rank
- $\Longrightarrow$ generalized inverses (ill-conditioned)

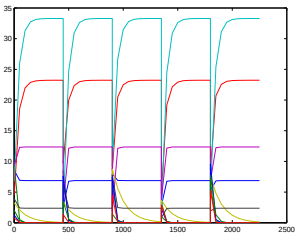
Algorithms evaluation

- to evaluate the power of the algorithms: → artificial netowrk
  - known graph (→ adjacency matrix A)
  - synthetic data as “gene profiles”

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$



$$\frac{dx_i}{dt} = V_i \prod_{j=j_1}^{j_a} \frac{x_j^\nu}{x_j^\nu + \theta_{ij}^\nu} \prod_{k=k_1}^{k_b} \frac{\theta_{ik}^\nu}{x_k^\nu + \theta_{ik}^\nu} - \lambda_i x_i$$



Algorithms evaluation

- for each algorithm:
  - **Input**: true connectivity matrix A_{true}
  - **Output**: matrix of edges weight W
  - $\Longrightarrow$reconstructed adjacency matrix $\hat{A} = W > w_o$
- parameters

TP (true positives)	=	correctly identifi ed true edges
FP (false positives)	=	spurious edges
TN (true negatives)	=	correctly identifi ed zero edges
FN (false negatives)	=	not recognized true edges

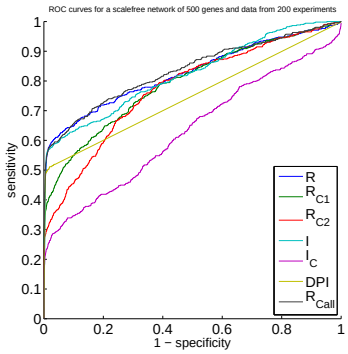
- $\text{edges}(A_{\text{true}}) = TP + FN, \quad \text{edges}(\hat{A}) = TP + FP$
- $\text{zeros}(A_{\text{true}}) = FP + TN, \quad \text{zeros}(\hat{A}) = FN + TN$

Algorithms evaluation

recall (or sensitivity)	true positive rates	$= \frac{TP}{TP+FN} = \frac{TP}{\text{edges}(A_{\text{true}})}$
specifi city	true negative rates	$= \frac{TN}{TN+FP} = \frac{TN}{\text{zeros}(A_{\text{true}})}$
precision	positive predicted values	$= \frac{TP}{TP+FP} = \frac{TP}{\text{edges}(\hat{A})}$

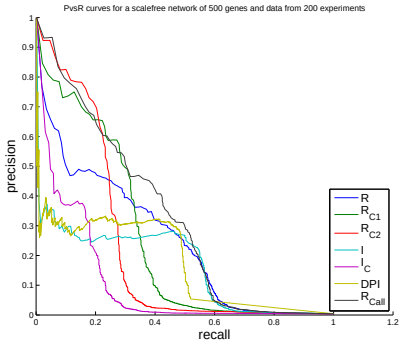
ROC

Receiver Operator Characteristic



PvsR

Precision vs Recall



Regulatory networks in human B cells

K. Basso, A. A. Margolin, G. Stolovitzky, U. Klein, R. Dalla-Favera, A. Califano. *Reverse engineering of regulatory networks in human B cells*, Nature Genetics 37, 382 - 390 (2005)

- goal: network reverse engineering in mammalian cells (here human B cells) as a key in understanding cell physiology and disease
- challenges of mammalian networks:
  - integrative approaches are not yet fully applicable given the very scattered nature of the available mammalian cell information
  - e.g.: systematic (experimental) gene perturbations are technically challenging and time-consuming

Regulatory networks in human B cells

- algorithm used ARACNe (algorithm for the reconstruction of accurate cellular networks)
  - identifies statistically significant gene-gene coregulation by mutual information
  - eliminates indirect relationships, in which two genes are coregulated through one or more intermediaries, by means of the 'data processing inequality' (from transmission theory)
  - $\Longrightarrow$relationships included in the network with high probability represent
    - direct regulatory interactions
    - interactions mediated by post-transcriptional modifiers (undetectable from gene-expression profiles)
    - algorithm complexity: $O(n^3)$ $\Longrightarrow$allows to analyze large scale networks

ARACNe: the algorithm

A A. Margolin, I Nemenman, K Basso, U Klein, C Wiggins, G Stolovitzky, R Dalla Favera, A Califano *ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context*, BCM Bioinformatics, 2006

- assumption: static inter-gene statistical dependencies only
- joint probability distribution

$$P(\{X_i\}) = \frac{1}{Z}\exp\left[-\sum_i \Phi_i(X_i) - \sum_{i,j}\Phi_{i,j}(X_i,X_j) - \sum_{i,j,k}\Phi_{i,j,k}(X_i,X_j,X_k) - \right.$$

$$= \frac{1}{Z}\exp\left[-\mathcal{H}(\{X_i\})\right]$$

where
- n = # of genes,
- m = # of samples,
- Z = partition function
- $\Phi_i(X_i)$ = potentials
- $\mathcal{H}(\{X_i\})$ = Hamiltonian

ARACNe: the algorithm

- simplest possible model: all genes are idependent
$$\mathcal{H}(\{X_i\}) = \sum_i \Phi_i(X_i)$$
- next: pairwise interactions
$$\mathcal{H}(\{X_i\}) = \sum_i \Phi_i(X_i) - \sum_{i,j} \Phi_{i,j}(X_i, X_j)$$
$\rightarrow$ we take this truncation as our "joint"
- statistical independent vs non-interacting genes
  - X_i, X_j statistically independent if $P(X_i, X_j) \simeq P(X_i)P(X_j)$
  - X_i, X_j non-interacting if $\Phi_{ij}(X_i, X_j) \simeq 0$

"statistical independent" $\underset{\not\Leftarrow}{\Rightarrow}$ "non-interacting"

$$\left(P(X_i, X_j) \simeq P(X_i)P(X_j) \underset{\not\Leftarrow}{\Rightarrow} \Phi_{ij}(X_i, X_j) \simeq 0\right)$$

- when is this happening?
when there are indirect interactions!

ARACNe: the algorithm

- testing all potential interactions $\Phi_{ij}(X_i, X_j)$ is computationally heavy and sample demanding $\Longrightarrow$methods to reduce the cost
- step 1: identify candidate interactions by the pairwise *mutual information*
$$I_{ij} = I(X_i, X_j)$$
  - put a threshold:
$$\text{if } I_{ij} \leqslant I_0 \text{ then } \Phi_{ij} = 0$$
  - $\rightarrow$ Relevance network approach
  - problem: does not detect indirect interactions
  - i.e., co-regulated genes may have $I_{ij} > I_0$ but still $\Phi_{ij} = 0$

ARACNe: the algorithm

- step 2: use the *data processing inequality* (DPI)
  - meaning: if X_1 and X_3 interact only through a third gene, then it must be

$$I(X_1, X_3) \leqslant \min\left(I(X_1, X_2), I(X_2, X_3)\right)$$

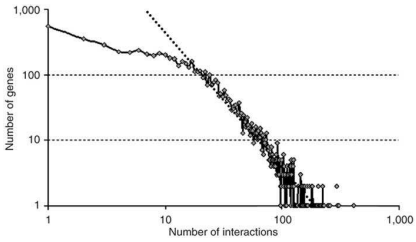
    $\Longrightarrow \Phi_{13}$ can be removed (indirect interaction)
  - checing all triplets above statistical significance I_{ij}, I_{jk}, I_{ki} the DPI removes the least significant arc
    $\Longrightarrow$ Markov triple $X_1 \longleftrightarrow X_2 \longleftrightarrow X_3$
- Theorem: if I_{ij} are correct ("asymptotic behavior," meaning "many data") and if the network is a tree $\Longrightarrow$the ARACNe reconstructs the network exatly
  - meaning of the DPI: interactions decorrelate rather quickly
  - caution: 3-node loops are always opened!
  - algorithm focuses on a network that locally is a tree
  - long loops may survive the "pruning"

Data available

- $\sim$ 6000 genes
- 336 gene expression profiles representative of perturbations of B cell phenotypes:
  - normal cells: resting pre-germinal center naive B cells, proliferating germinal center B cells (centroblasts and centrocytes) and post-germinal center memory B cells
  - Transformed cells more than ten subtypes of B cell malignancies
  - Experimentally manipulated cells treated in vitro to induce specific signal transduction pathways or engineered for the expression of several transcription factors
- organism- and tissue-specific perturbations $\Longrightarrow$highly specific interactions

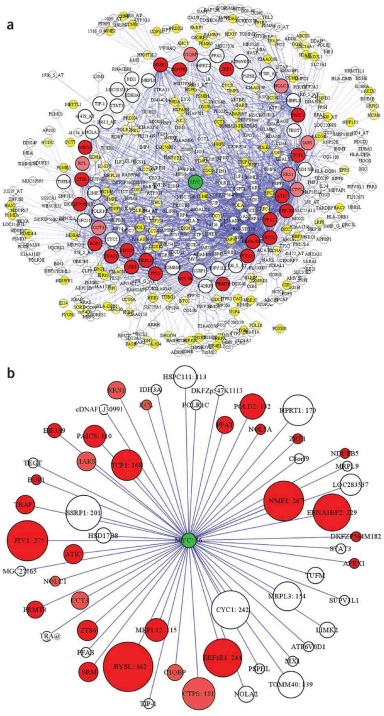
Results

- network of $\sim$120.000 interactions
- connectivity graph of the network has a *power-law tail*
- suggests a scale-free network
  - few *hubs* (highly connected)
  - many nodes with low connectivity
  - 5% of nodes (major hubs) account for $\sim$ 50.000 connections
  - hierarchical structure: hubs tend to communicate a lot among each other



Zooming in: MYC subnetwork

- the proto-oncogen MYC is knonw to be an “important node”
- the algorithm confirmed it as a major hub
  - 56 first neighbors,
  - 2007 second neighbors



- 30% of first neighbors are large hubs
- hierarchical structure
- redundacy
- robustness

Validation of the MYC subnetwork

- 29 out of 56 first neighbors were known MYC targets
- known targets are more significant classified as first neighbors (51.8%) than second neighbors (19.4%)
- 37.5% of first neighbors were validated in vivo by chromatin immunoprecipitation (ChIP)
  $\Longrightarrow$binding of MYC to their promoter region was shown in new candidate MYC target
  $\Longrightarrow$validation *in vivo* of the regulatory pathways

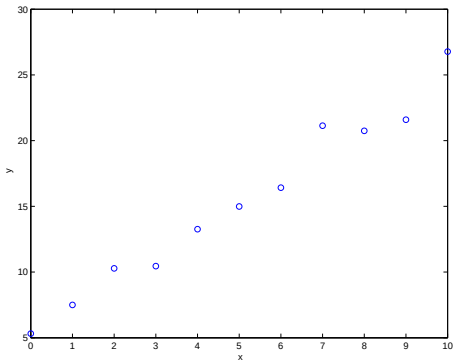
Discussion & Limitations

- on the positive side:
  - resuts are validated experimentally (remarkable)
  - large part (40%) of the data collected on a decade on MYC was correctly represented
- limitations (in the paper):
  - edges lack directionality (i.e., they do not indicate which gene is 'upstream' or 'downstream');
  - some direct connections may involve unknown intermediates, as not all biochemical species participating in cellular interactions are represented on the microarray
  - some direct interactions may have been incorrectly removed by the DPI
- futher limitations (in my biased opinion)
  - the # of gene expression profiles they start with is probably too limited
  - we do not find the DPI very effective

Linear ODEs networks

Parameters fitting

- how do you fit the parameters in a model in order to match experimental data?
  - $\longrightarrow$ linear regression analysis
  - simplest case: want to fit a straight line to a set of measured values (x_1, y_1), (x_2, y_2), $\ldots$, (x_m, y_m)



Parameters fitting

- model to fit

$$y = \alpha + \beta x + \epsilon$$

  - α, β *regression coefficients*
  - $y =$ response, $x =$ regressor
  - $\epsilon \in N(0, \sigma^2)$ random error
- task: *estimate α and β from the measured data* $\Longrightarrow \hat{\alpha}$, $\hat{\beta}$
- solution: least squares

  observed $y =$ fitted y + residuals
  - fitted y: reflects the straight line $\hat{y} = \hat{\alpha} + \hat{\beta} x$
  - residuals: random deviation from the straight line

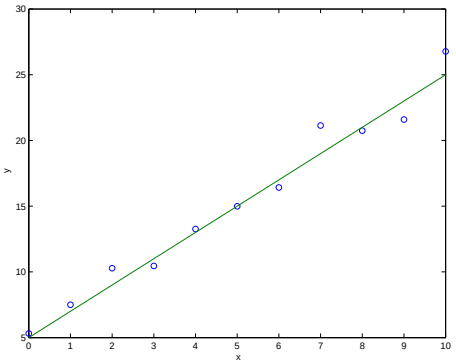
$$\begin{aligned} \min_{\hat{\alpha},\hat{\beta}} S &= \sum_{i=1}^{m} (\text{observed } y - \text{ fitted } y)^2 \\ &= \sum_{i=1}^{m} (y_i - \hat{\alpha} - \hat{\beta} x_i)^2 \end{aligned}$$

Parameters fitting

- $\hat{\alpha}$, $\hat{\beta}$ can be computed explicitely from the partial derivatives

$$\left.\begin{aligned} \frac{\partial S}{\partial \hat{\alpha}} &= -2\sum_{i=1}^{m}(y_i - \hat{\alpha} - \hat{\beta} x_i) = 0 \\ \frac{\partial S}{\partial \hat{\beta}} &= -2\sum_{i=1}^{m} x_i (y_i - \hat{\alpha} - \hat{\beta} x_i) = 0 \end{aligned}\right\} \Longrightarrow \left\{\begin{aligned} \hat{\beta} &= \frac{\sum_{i=1}^{m} x_i (y_i - \bar{y})}{\sum_{i=1}^{m} x_i (x_i - \bar{x})} \\ \hat{\alpha} &= \bar{y} - \hat{\beta}\bar{x} \end{aligned}\right.$$

  where $\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i$ and $\bar{y} = \frac{1}{m}\sum_{i=1}^{m} y_i$ (averages)
- meaning of the fitted model $\hat{y} = \hat{\alpha} + \hat{\beta} x$: line passing through the centroid $(\bar{x}, \bar{y})$ and rotated until the squared deviations is least.
  - in Matlab:
    - `lsqr` function
    - `regression` function (Statistics toolbox)



Least squares fitting of nonlinear models

- this to fit a straight line: How about fitting a more complicated curve?
- linear regression: linear in the coefficients α, β , etc., not in the model structure
  - example: quadratic model

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \epsilon$$

  is solvable directly by means of linear regression

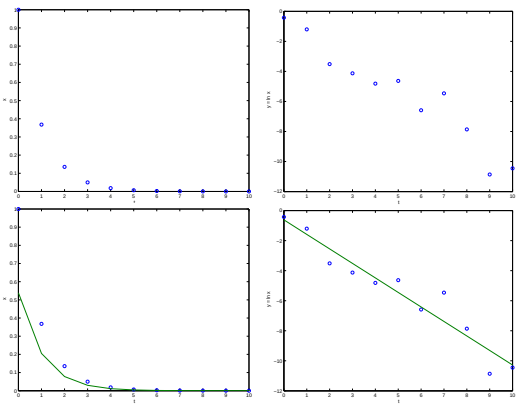
- how about nonlinear regression?

Least squares fitting of nonlinear models

- example: Maltus law $x(t) = x_o e^{-\gamma t}$
  - recall that this is the integral of the linear ODE $\frac{dx}{dt} = -\gamma x$ and describes for exampe the degradation of a substance
  - least square fitting: assume I measure a few time values $x(t_1) = x_1, \ldots, x(t_m) = x_m$
  - how do I find γ?
  - take the logarithm on both sides of $x(t) = x_o e^{-\gamma t}$:

$$\underbrace{ln\, x(t)}_{y(t)} = \underbrace{ln\, x_o}_{\alpha} - \gamma t$$

  $\Longrightarrow$ linear regression



- alternatively: measure $x(t)$ and $\frac{dx}{dt}$ simultaneously and fit directly on the ODE

Least squares fitting of nonlinear models

- how about a Hill function? $h^+(x,\theta,n) = \frac{x^n}{\theta^n + x^n}$
  - consider $\frac{dx}{dt} = kh^+(x,\theta,n)$
  - assume x and $\frac{dx}{dt}$ are measured
  - k = value at saturation $\Longrightarrow$ known from $\frac{dx}{dt}$ as $t \to \infty$
  - to find θ and n
    - manipulating: $\dot{x} = \frac{kx^n}{\theta^n + x^n} \Longrightarrow x^n = \frac{\dot{x}\,\theta^n}{k - \dot{x}}$
    - take the logarithm

      $ln(x^n) = \ln\left(\frac{\dot{x}}{k-\dot{x}}\theta^n\right) = \ln\left(\frac{\dot{x}}{k-\dot{x}}\right) + ln(\theta^n)$
    - i.e.,

$$\underbrace{n}_{\beta}\underbrace{ln(x)}_{x} = \underbrace{nln\theta}_{\alpha} + \underbrace{ln\left(\frac{\dot{x}}{k-\dot{x}}\right)}_{y}$$

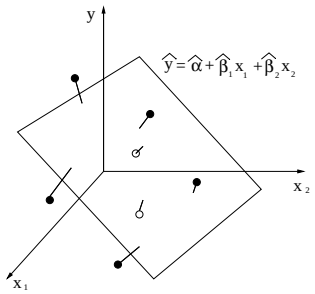
$\Longrightarrow$ linear regression applies

Multiple linear regression

- when observation depends on 2 or more independent variables

$$y = \beta_o + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

$\rightarrow$ multiple linear regression model



- procedure is the same: sample regression equations

$$y^i = \beta_o + \beta_1 x_1^i + \beta_2 x_2^i + \ldots + \beta_n x_n^i \quad i = 1, \ldots, m$$

rewritten in matrix form $\boxed{\mathbf{y} = \mathbf{X}\beta + \epsilon}$

where

$$\mathbf{y} = \begin{bmatrix} y^1 \\ \vdots \\ y^m \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_1^1 & \ldots & x_n^1 \\ 1 & x_1^2 & \ldots & x_n^2 \\ \vdots & \vdots & & \vdots \\ 1 & x_1^m & \ldots & x_n^m \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_o \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}, \quad \epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_m \end{bmatrix}$$

Multiple linear regression

- functional to be minimized

$$S(\beta) = \sum_{i=1}^{k} \epsilon^2 = \|\epsilon\|_2 = \|\mathbf{y} - \mathbf{X}\beta\|_2 \quad \text{(i.e., } = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)\text{)}$$

- the least squares problem is solved by $\hat{\beta}$ such that

$$\left.\frac{\partial S(\beta)}{\partial \beta}\right|_{\hat{\beta}} = -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\hat{\beta} = 0$$

i.e., $\mathbf{X}^T\mathbf{X}\hat{\beta} = \mathbf{X}^T\mathbf{y} \Longrightarrow$ $\boxed{\hat{\beta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}}$

- fitted model

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y} =: \mathbf{H}\mathbf{y}$$

- residuals

$$\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}} = (\mathbf{I} - \mathbf{H})\mathbf{y}$$

Multiple linear regression

- when is $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}$ well-defined?

$$\mathbf{X} = \begin{bmatrix} 1 & x_1^1 & \dots & x_n^1 \\ 1 & x_1^2 & \dots & x_n^2 \\ \vdots & \vdots & & \vdots \\ 1 & x_1^m & \dots & x_n^m \end{bmatrix},$$

$\mathbf{X}$ $m \times (n+1)$ matrix

$\mathbf{X}^T\mathbf{X}$ $(n+1) \times (n+1)$ matrix

- if $rank(\mathbf{X}^T\mathbf{X}) = n+1$ then $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}$ exists
- meaning:
  1. $m \geqslant n+1$
  2. the regressors of $\mathbf{X}$ (i.e., the columns of $\mathbf{X}$) must be linearly independent
- practical meaning
  1. *# of experiments must be $\geqslant$ # of variables*
  2. *data must be collected in different experimental situations*

Nonlinear Multiple regression

- how about nonlinear models?

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x},\, \beta), \qquad \mathbf{f}(\mathbf{x},\, \beta) = \begin{bmatrix} f_1(\mathbf{x},\, \beta) \\ \vdots \\ f_n(\mathbf{x},\, \beta) \end{bmatrix}$$

- if $\mathbf{f}(\mathbf{x},\, \beta)$ is "diagonal"

$$\mathbf{f}(\mathbf{x},\, \beta) = \begin{bmatrix} f_1(x_1,\, \beta) \\ \vdots \\ f_n(x_n,\, \beta) \end{bmatrix}$$

then the ODEs are not coupled
$\Longrightarrow$ each nonlinear problem can be treated separately

- if not, then there is no general rule to deal with it. A common approach it to linearize around an equilibrium point.

Inferring a network of ODEs

- network of n genes $x_1,\, x_2, \ldots, x_n$
- vector ODEs in general form

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}), \qquad \mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix}$$

$f_i(\mathbf{x}) =$ description of how the expression levels of genes $x_1, \ldots, x_n$ are affecting the transcription rate of x_i

- typically
  - $f_i(\mathbf{x}) > 0$ (*activation*) for some combination of $x_1, \ldots, x_n$
  - $f_i(\mathbf{x}) < 0$ (*repression*) for some other combination
  - for $(x_1, \ldots, x_n)$ such that $f_i(\mathbf{x}) = 0$ for all $i = 1, \ldots, n$ then we have a steady state (rate of all x_i stays constant)
- Inferring the network means finding the functions f_i

Inferring a network of ODEs

- simplest form: linear model $\frac{dx_i}{dt} = f_i(\mathbf{x}) = a_{i1}x_1 + \ldots + a_{in}x_n$

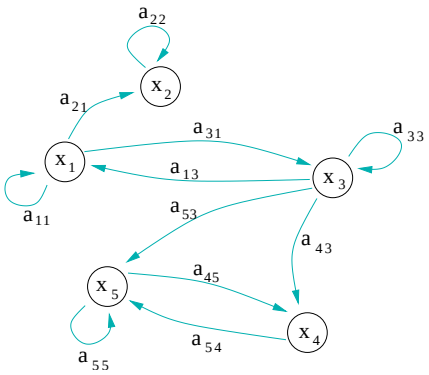
$$\frac{d\mathbf{x}}{dt} = A\mathbf{x}, \qquad A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & \ldots & & a_{nn} \end{bmatrix}$$

  - a_{ij} measure of the interaction strength of gene j over gene i
  - $a_{ij} = 0 \Longrightarrow$gene j is not affecting the transcription rate of gene i
- finding A means finding the connectivity matrix of the network (i.e., its topology)
- in addition, the a_{ij} also quantify the network of connections
- typically only a few a_{ij} are $\neq 0$ on each row
  $\Longrightarrow$low connectivity (the matrix A is sparse)

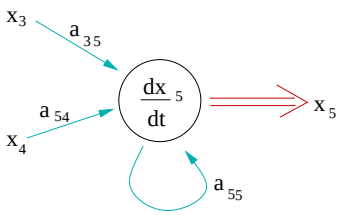
Inferring a network of ODEs

- example

$$A = \begin{bmatrix} a_{11} & & a_{13} & & \\ a_{21} & a_{22} & & & \\ a_{31} & & a_{33} & & \\ & & a_{43} & & a_{45} \\ & & a_{53} & a_{54} & a_{55} \end{bmatrix}$$



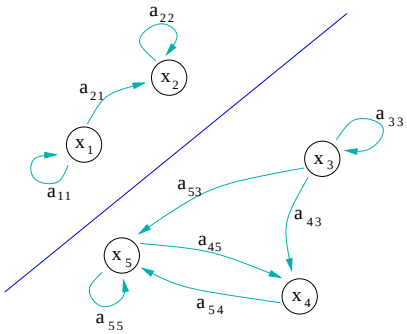
- directed graph $\Longrightarrow$this is *causal information*



Inferring a network of ODEs

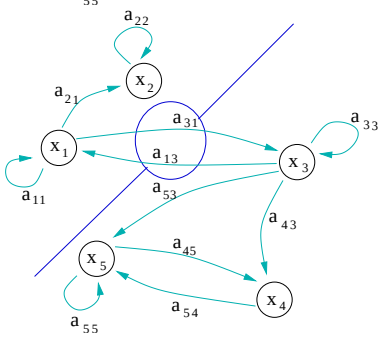
- the structure of A also allows do identify independent blocks

$$A = \begin{bmatrix} a_{11} & & & & \\ a_{21} & a_{22} & & & \\ & & a_{33} & & \\ & & a_{43} & & a_{45} \\ & & a_{53} & a_{54} & a_{55} \end{bmatrix}$$



- as well as "lethal" edges

$$A = \begin{bmatrix} a_{11} & & a_{13} & & \\ a_{21} & a_{22} & & & \\ a_{31} & & a_{33} & & \\ & & a_{43} & & a_{45} \\ & & a_{53} & a_{54} & a_{55} \end{bmatrix}$$



Linearization

- in reality the function f will typically be nonlinear
- reconstructing f nonlinear: the # of parameters may be high (ex: each Hill function has 3 parameters) typically then one considers the linear model obtained by means of linearization in a neighbohood of a stable steady state x^* (i.e. such that $f(x^*) = 0$)

$$A = \left.\frac{\partial f}{\partial x}\right|_{x=x^*} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & & \\ \frac{\partial f_n}{\partial x_1} & \cdots & & \frac{\partial f_n}{\partial x_n} \end{bmatrix}\right|_{x=x^*} = \text{Jacobian}$$

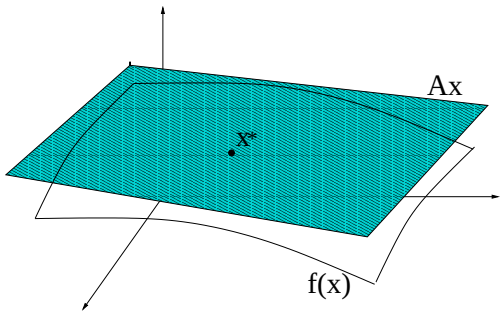
i.e., $$a_{ij} = \left.\frac{\partial f_i(x)}{\partial x_j}\right|_{x=x^*}$$

Linearization

- calling $\mathbf{x} = \mathbf{x}^* + \delta\mathbf{x}$ ($\delta\mathbf{x}$ small)
  $\Longrightarrow$Taylor expansion around $\mathbf{x}^*$

$$\frac{d\mathbf{x}}{dt} = \underbrace{f(\mathbf{x}^*)}_{=0} + \underbrace{\left.\frac{\partial f}{\partial x}\right|_{x=x^*} \mathbf{x}}_{A\mathbf{x}} + \underbrace{o(\delta\mathbf{x})}_{\approx 0} = A\mathbf{x}$$

- linearized system = plane tangent to the full nonlinear model computed at x^*



Information extrapolated from A

- compute the eigenvalues of A

  eigenvalues of A = real or complex numbers λ such that $A\mathbf{v} = \lambda\mathbf{v}$ for some vector $\mathbf{v}$ (called *eigenvector*): they give the *characteristic modes* of the ODE $\frac{d\mathbf{x}}{dt} = A\mathbf{x}$

- A is stable $\iff \mathrm{Re}[\lambda] < 0$ ($\mathrm{Re}[\lambda] \leqslant 0$ and multiplicity(λ)$\leqslant 1$)
  - $\Longrightarrow$all modes are decaying
  - $\Longrightarrow$ $\mathbf{x}(t) \xrightarrow{t\to\infty} \mathbf{x}^*$ stable stationary steady state
  - $\Longrightarrow$if I disturb the initial condition: $\mathbf{x}(t) + \delta\mathbf{x}(t) \xrightarrow{t\to\infty} \mathbf{x}^*$
- if $\mathrm{Re}[\lambda] > 0$ for some eigenvalue $\lambda \Longrightarrow A$ is unstable and $\mathbf{x}(t)$ may grow unbounded
- λ complex number $\Longrightarrow \mathbf{x}(t)$ has oscillations
- a_{ii} = self-regulating coefficient: includes degradation rate

$$A = \tilde{A} - \begin{bmatrix} \gamma_1 & & \\ & \ddots & \\ & & \gamma_n \end{bmatrix}$$

Inferring linear ODEs

- how to compute A?
- If we
  1. can perturb each gene individually
  2. measure simultaneously
     - gene expression level $\mathbf{x}$
     - amount of the perturbation b^i
     - rate $\frac{d\mathbf{x}}{dt}$
- $\Longrightarrow$finding A becomes a multilinear regression problem
- example: choosing a linear additive model for the perturbation

$$\underbrace{\begin{bmatrix} \dot{x}_1^1 & \dots & \dot{x}_1^n \\ \dot{x}_2^1 & \dots & \dot{x}_1^n \\ \vdots & & \vdots \\ \dot{x}_n^1 & \dots & \dot{x}_n^n \end{bmatrix}}_{1^{st}\text{exp.}\dots n^{th}\text{exp.}} = A \underbrace{\begin{bmatrix} x_1^1 & \dots & x_1^n \\ x_2^1 & \dots & x_1^n \\ \vdots & & \vdots \\ x_n^1 & \dots & x_n^n \end{bmatrix}}_{1^{st}\text{exp.}\dots n^{th}\text{exp.}} + \underbrace{\begin{bmatrix} b^1 & \dots & 0 \\ 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & b^n \end{bmatrix}}_{1^{st}\text{exp.}\dots n^{th}\text{exp.}}$$

Inferring linear ODEs

$$\dot{X} = AX + B$$

- $\Longrightarrow$finding A becomes a multilinear regression problem

$$A = \text{argmin}\|AX + B - \dot{X}\|_2$$

$$\left(\text{i.e., for each row } \mathbf{a} \text{ of } A\text{: } \hat{\mathbf{a}}^T = \left(XX^T\right)^{-1} X(\dot{\mathbf{x}}_{\text{row}}^T - \mathbf{b}_{\text{row}}^T)\right)$$

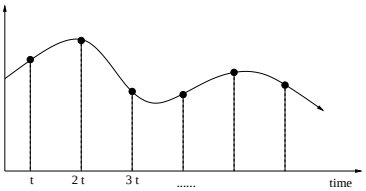
- properties
  - perturbations need not form a diagonal matrix B: all is needed is a matrix X such that $\text{rank}(X) \geqslant n$ i.e., at least n arrays of "independent" measurements
  - drawback: A is very sensitive to noise in X and B
  - drawback: need to measure the rates $\frac{d\mathbf{x}}{dt} \rightarrow$ this is normally done by finite difference schemes + interpolation/smoothing provided you have a time series of data

Continuos vs Discrete time

- model to fit

$$\dot{x} = Ax$$

- measures (constant sampling time T)



$$x(T),\ x(2T),\ x(3T), \ldots, x(mT)$$

- fitting a discrete dynamical model:
  - since measurements are in discrete time, one can choose to infer a discrete-time state update matrix F:

$$x((k+1)T) = Fx(kT), \qquad k = 1, 2, \ldots$$

  - multilinear regression procedure

$$\underbrace{[x(nT)\ x((n-1)T) \ldots x(2T)]}_{\text{measured}} = F \underbrace{[x((n-1)T)\ x((n-2)T) \ldots x(T)]}_{\text{measured}}$$

  - inferring F needs no observation of the $\frac{dx}{dt}$

Continuos vs Discrete time

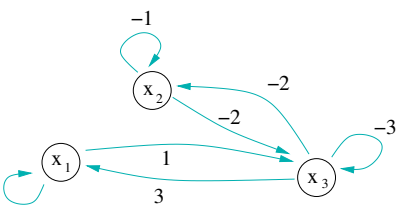
- what is the relation between A and F??

$$F = e^{AT}$$

  - if A sparse $\not\Rightarrow$ F is sparse
  - $\Longrightarrow$if A = connectivity matrix, $F = e^{AT}$ is not a connectivity matrix!
  - not obvious (neither easy) to reconstruct A given F (i.e., to find the network from F)

- example

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



$$F = e^{AT} = \begin{bmatrix} 2.7 & 1 & 20.1 \\ 1 & 0.4 & 0.1 \\ 2.7 & 0.1 & 0.1 \end{bmatrix}$$

Continuos vs Discrete time

- how to get A out of F?
  1. exact caculation: matrix logarithm $\boxed{A = \frac{\ln(F)}{T}}$
     - complication: $m < n$
       $\Longrightarrow F$ is not full-rank $\Longrightarrow$cannot take log
  2. approximation 1.: Euler discretization

$$\frac{dx}{dt} \simeq \frac{x((k+1)T) - x(kT)}{T} \implies x((k+1)T) = (I + TA)x(kT)$$
$$= Fx(kT)$$

$$\boxed{A = \frac{F - I}{T}}$$

  3. approximation 2.: bilinear approximation

$$\boxed{A = \frac{2F - I}{TF + I}}$$

- both approximations are inadequate when T is not very small

Inferring a linear model via SVD

Yeung MKS, Tegner J and Collins JJ. Reverse engineering gene networks using singular value decomposition and robust regression. PNAS 99: 6163-6168 (2002)

- most times: # of experiments $\ll$ # of variables: $\boxed{m \ll n}$
  $\Longrightarrow$reverse engineering problem is underdetermined
- how to recover A?
  - infinitely many possible solutions
  - $\Longrightarrow$many network architectures fit the data
- one possible solution: SVD Singular Value Decomposition

$$\underbrace{\dot{X}}_{n\times m} = A \underbrace{X}_{n\times m} + \underbrace{B}_{n\times m}$$

Inferring a linear model via SVD

- use SVD to decompose X:

$$X = U\Lambda V^T$$

U $n \times m$ orthogonal $UU^T = I_n$
V $m \times m$ orthogonal $VV^T = I_m$

where $\ell = \mathrm{rank}(X)$
$\lambda_1, \ldots, \lambda_\ell$ singular values:
$\lambda_j = \sqrt{\mu_j}$, $\mu_j = \mathrm{eig}(XX^T)$

$$\Lambda = \begin{bmatrix} 0 & & & & & \\ & \ddots & & & & \\ & & 0 & & & \\ & & & \lambda_1 & & \\ & & & & \ddots & \\ & & & & & \lambda_\ell \end{bmatrix}$$

- one particular solution is given by the Moore-Penrose pseudoinverse

$$\boxed{A_o = (\dot{X} - B)U\Lambda^\dagger V^T}, \qquad \Lambda^\dagger = \begin{bmatrix} \frac{1}{\lambda_1} & & & & & \\ & \ddots & & & & \\ & & \frac{1}{\lambda_\ell} & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix}$$

Inferring a linear model via SVD

- SVD solution is *still* the least squares solution:

$$A_o = \mathrm{argmin}\|AX + B - \dot{X}\|_2$$

- general solution: affine space

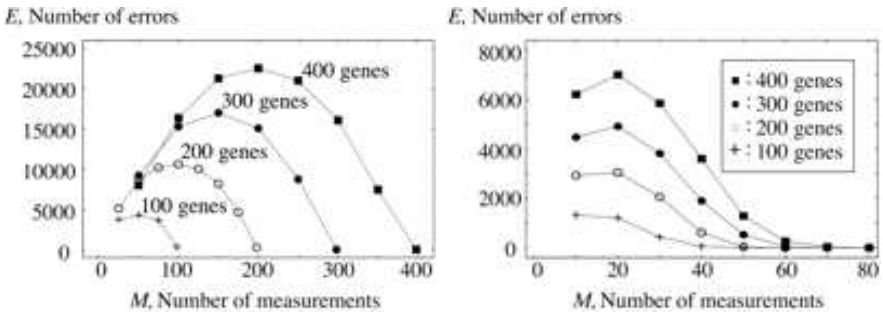
$$A = A_o + CV^T \qquad C = \begin{bmatrix} c_{11} & \ldots & c_{1\ell} & 0 & \ldots & 0 \\ \vdots & & & & & \vdots \\ c_{n1} & \ldots & c_{n\ell} & 0 & \ldots & 0 \end{bmatrix}$$

  - c_{ij} = all degrees of freedom that can be used to optimize some extra criterion: e.g. the sparsity of A
  - $\Longrightarrow$maximize # of zeros in A
  - impose $A = 0$ i.e., $A_o = -CV^T$

Inferring a linear model via SVD

- want to satisfy as many constraints as possible: solution is different from least squares solution
- robust regression problem: L_1 problem exact fit leaving as few outlier as possible

$$\hat{A} = \text{argmin}\|AX + B - \dot{X}\|_1$$



Yeung MKS, Tegner J and Collins JJ. . PNAS 99: 6163 (2002)

Linear model via steady state measurements

Gardner TS, di Bernardo D, Lorenz D and Collins JJ. Inferring genetic networks and identifying compound mode of action via expression profi ling. Science 301: 102-105 (2003)

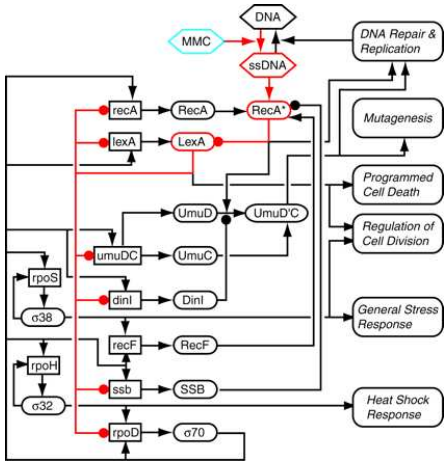
- problem with previous method: still requires to measure the rates $\frac{d\mathbf{x}}{dt}$
- often times there is no time series available: only apply perturbations and wait for the system to resettle at a new steady state $\Longrightarrow \frac{d\mathbf{x}}{dt} = 0$
- if $\mathbf{x}^*$ is stable, the new steady state is nearby $\Longrightarrow$linear methods still make sense
- after a perturbation:

$$AX + B = 0 \quad \Longrightarrow AX = -B$$

$\Longrightarrow$still the same multilinear regression problem

Linear model via steady state measurements

- SOS pathways in E. coli: regulates cell survival and repair after DNA damage
- network
  - red: primary pathway (well studied)
  - 9 genes
  - each perturbed singularly
  - perturbations: overexpression of one single gene



$$A \underbrace{\begin{bmatrix} x_1^1 & \dots & x_1^9 \\ x_2^1 & \dots & x_1^9 \\ \vdots & & \vdots \\ x_9^1 & \dots & x_9^9 \end{bmatrix}}_{1^{st}\text{exp.}\dots 9^{th}\text{exp.}} = - \underbrace{\begin{bmatrix} b^1 & \dots & 0 \\ 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & b^9 \end{bmatrix}}_{1^{st}\text{exp.}\dots 9^{th}\text{exp.}}$$

Linear model via steady state measurements

- Reverse enginnering

Association networks

Linear ODEs networks
- Nonlinear Fitting
- Multiple linear regression
- Nonlinear Multiple regression
- Inferring a network of ODEs
- Linearization
- Information extrapolated from A
- Inferring linear ODEs
- Continuos vs Discrete time
- Inferring a linear model via SVD
- steady state inference
- model-based drug design

- 9 genes, 9 perturbations $\Longrightarrow$enough to compute

$$A = \text{argmin}\|AX + B\|_2$$

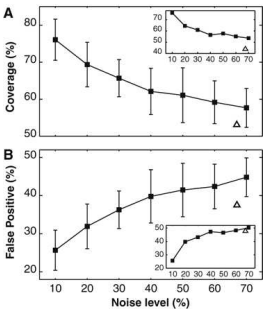
  although result will be very noisy
- in general however $m \ll n$ $\Longrightarrow$underdetermined problem
- instead of using SVD, assume degree of connectivity is $k \simeq m$ (meaning: each row of A has at most k nonzero elements)
  $\Longrightarrow$conditioning is better in presence of a limited number of experiments
- solution is still Moore-Penrose pseudoinverse
- drawbacks of the method
  - to find the "best" row of k paramters a_{ij} is a combinatorial problem: $\frac{n!}{k!(n-1)!}$ ("n choose k")
    $\Longrightarrow$computational explosion
    $\Longrightarrow$heuristic solutions

Linear model via steady state measurements

previously mapped regulatory structure

	recA	*lexA*	*ssb*	*recF*	*dinI*	*umuDC*	*rpoD*	*rpoH*	*rpoS*
recA	+	−	−	+	+	−	+	0	0
lexA	+	−	−	+	+	−	+	0	0
ssb	+	−	−	+	+	−	+	0	0
recF	0	0	0	0	0	0	+	0	+
dinI	+	−	−	+	+	−	+	0	0
umuDC	+	−	−	+	+	−	+	0	0
rpoD	+	−	−	+	+	−	+	+	0
rpoH	0	0	0	0	0	0	+	+	0
rpoS	0	0	0	0	0	0	+	0	+

coverage



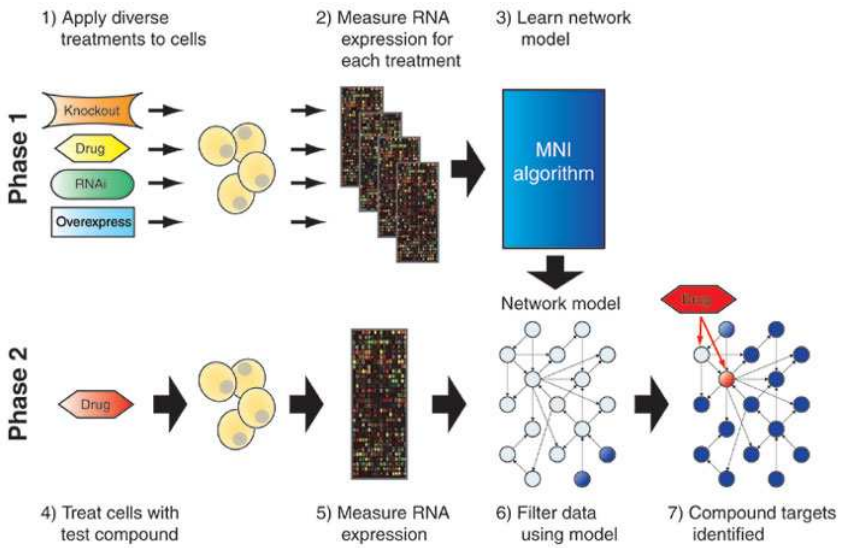
false positives

identified network

	recA	*lexA*	*ssb*	*recF*	*dinI*	*umuDC*	*rpoD*	*rpoH*	*rpoS*
recA	-0.597	-0.179	-0.010	0	0.096	0	-0.011	0	0
lexA	0.387	-1.670	-0.014	0	0.087	-0.068	0	0	0
ssb	0.044	-0.189	-1.275	0	0.053	0	0.027	0	0
recF†	-0.1808	0.2377	-0.0251	-1	-0.0554	0	0	0	0.39
dinI	0.281	0	0	0	-2.094	0.156	-0.037	0.012	0
umuDC	0.112	-0.403	-0.016	0	0.205	-1.147	0	0	0
rpoD	-0.171	0	-0.017	0	0.025	0	-1.513	0.021	0
rpoH	0.096	0	0.001	0	-0.009	-0.031	0	-0.483	0
rpoS	0.217	0	0	-1.678	0.672	0	0.077	0	-3.921

What lies ahead? Model-based drug design

di Bernardo D, Thompson MJ, Gardner TS, Chobot SE, Eastwood EL, Wojtovich AP, Elliott SJ, Schaus SE, Collins JJ. *Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks* Nat Biotechnol. 2005 ,23(3):377-83



What lies ahead? Model-based drug design

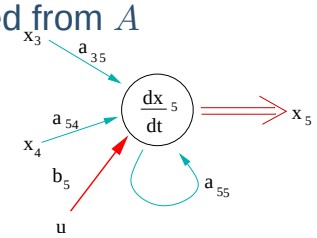
- assume to have identified the matrix A in

$$\dot{x} = Ax$$

- assume you want to test the effect of an external stimulus u (e.g. drug) on the system
- task: compute the mode of action of u:
  - applying u to the linear system

$$\dot{x} = Ax + bu$$

  - b = $n \times 1$ vector of coefficients = pathways through which the drug is acting $\rightarrow$ directly regulated genes
  - all indirect regulations must be obtained from A $\Longrightarrow$fundamental prerequisite: have a sound knowledge of A
  - finding b is easier than finding A (similar procedure)



What lies ahead? Model-based drug design

- Reverse enginnering

Association networks

Linear ODEs networks
- Nonlinear Fitting
- Multiple linear regression
- Nonlinear Multiple regression
- Inferring a network of ODEs
- Linearization
- Information extrapolated from A
- Inferring linear ODEs
- Continuos vs Discrete time
- Inferring a linear model via SVD
- steady state inference
- model-based drug design

- task: compute time-depdendent $u(t)$ such that

$$x(t) \rightarrow x_{\text{desired}}$$

- x_{desired} = particular array of expression levels of the genes of the system that I would like to achieve
- model can tell you when this is possible / not possible for a certain u or a linear combination of us:

$$\dot{x} = Ax + \begin{bmatrix} b_1 \dots b_p \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_p \end{bmatrix}$$

$\rightarrow$ linear (or nonlinear) control theory?