

Reverse-modeling of biological networks

Introduction to ODEs and state-space models

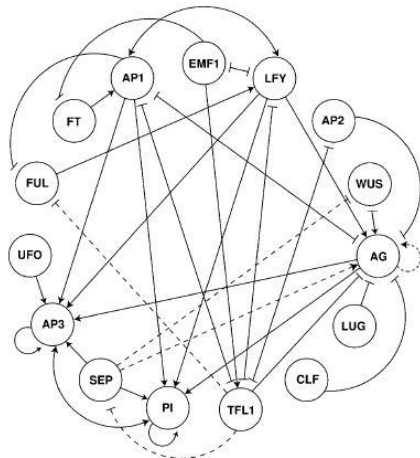
Florence d'Alché-Buc

Laboratoire Informatique, Biologie Intégrative et Systèmes Complexes
FRE 3190 CNRS, Université d'Évry-Val d'Essonne, France
Currently visiting **URA CNRS 2171 - Institut Pasteur, Paris, France**
Email: [florence.dalche at ibisc.fr](mailto:florence.dalche@ibisc.fr)

Outline

- 1 Introduction
- 2 Dynamical models
- 3 Dynamical graphical models
 - Parameter estimation

Gene regulatory networks



Protein-protein interaction networks (rice blast fungus)

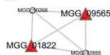


Ion transport
 $7.83 \cdot 10^{-4}$

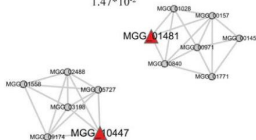
Neurological system process
 $1.98 \cdot 10^{-3}$



Phosphorus metabolic process
 $2.58 \cdot 10^{-7}$



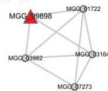
Protein transport
 $1.47 \cdot 10^{-2}$



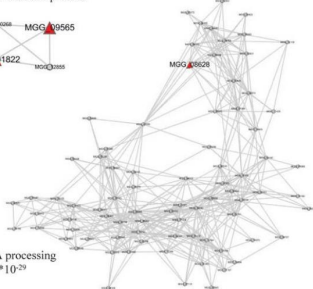
Chromatin silencing
 $7.82 \cdot 10^{-3}$

[Eisen et al. '02](#)

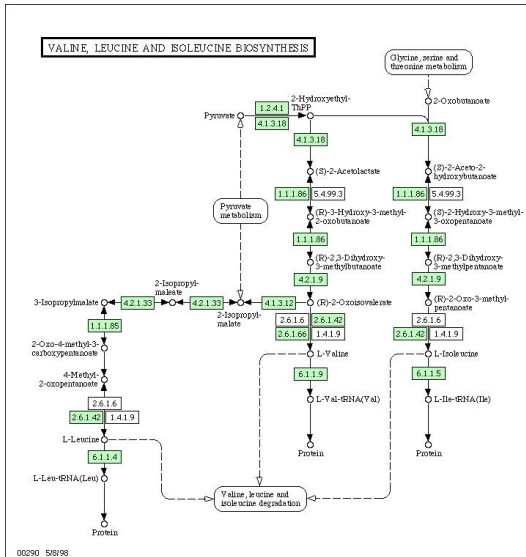
Symbiosis, encompassing mutualism through parasitism
 $1.75 \cdot 10^{-2}$



RNA processing
 $3.15 \cdot 10^{-29}$



Metabolic networks



Motivation

- Networks:
 - signaling pathways
 - gene regulatory networks
 - protein-protein interaction networks
 - metabolic pathways
- Goals:
 - Identify and understand complex mechanisms at work in the cell
 - Potential outputs: therapeutical targetting
- Means:
 - Use experimental data to identify networks: here comes machine learning

How to learn biological networks from data ?

- **Predictive approaches** : predict (only) edges in an unsupervised or supervised way (scale (10^3) nodes)
- **Modeling approaches** : model the network as a complex system and identify it (scale : 10 to a few 10^2 nodes)

Motivation for modeling approaches

- A model reflect assumptions about the behaviour of the network as a dynamical system
- Once learnt, a model can be used in simulation and prediction
- Be careful, a model IS NOT the truth, it is just a tool to test hypothesis

Biological networks as dynamical systems

Two families of models

- 1 **Differential Equations:** bifurcation analysis, control theory, theory of dynamical systems
- 2 **Dynamical Graphical models:** frequentist and Bayesian approaches, incomplete data, building blocks in mixture models (switching models, modular approaches)

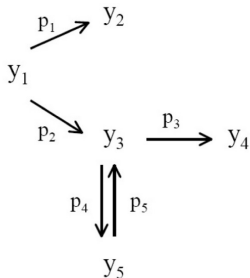
System of Ordinary Differential Equations (ODE)

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t); \theta)$$

- gene regulatory networks
- $\mathbf{x}(t)$: state variables at time t
 - $\mathbf{p}(t)$: protein concentrations
 - $\mathbf{r}(t)$: mRNA concentrations
- \mathbf{f} : the form of f encodes the nature of interactions (and their structure)
 - linear/nonlinear models
 - Michaelis-Menten kinetics
 - Mass action kinetics
 - ...
- θ : parameter set (kinetic parameters, rate constants,...)
- $\mathbf{u}(t)$: input variables at time t

Examples of ODEs

- Isomerization of α -pinene
- JAK-STAT pathways
- Repressilator

Isomerization of α -pinene network

• ODEs:

$$\dot{x}_1 = -(p_1 + p_2)x_1$$

$$\dot{x}_2 = p_1 x_1$$

$$\dot{x}_3 = p_2 x_1 - (p_3 + p_4)x_3 + p_5 x_5$$

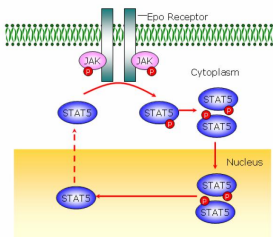
$$\dot{x}_4 = p_3 x_3$$

$$\dot{x}_5 = p_4 x_3 + p_5 x_5$$

- Isomerization of α -pinene (y_1) to dipentene (y_2) and allo-ocimen (y_3)
- which in turn yields α - and β -pyronene (y_4) and a dimer (y_5)

JAK-STAT signaling pathway

[Swameye, PNAS 2003]



- ODEs:

$$\begin{cases} \dot{x}_1(t) &= -a_1 x_1(t)u(t) + 2a_4 x_4(t)1_{\{t \geq \tau\}} \\ \dot{x}_2(t) &= a_1 x_1(t)u(t) - 2a_4 x_2^2(t) \\ \dot{x}_3(t) &= -a_3 x_3(t) + x_2^2(t) \\ \dot{x}_4(t) &= a_3 x_3(t) - a_4 x_4(t)1_{\{t \geq \tau\}} \end{cases}$$

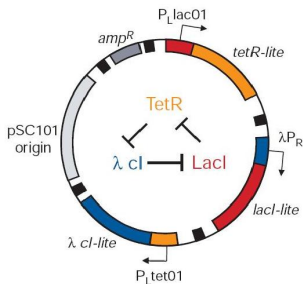
- Observed variables

$$y_1 = x_2 + 2x_3$$

$$y_2 = x_1 + x_2 + 2x_3$$

- Experimental data: 16 time points
- $\theta = (a_1, a_3, a_4)^\top$ is the parameters to be estimated

Repressilator (Elowitz and Leibler 2000)



$$\frac{dr_1}{dt} = v_1^{\max} \frac{k_{12}^n}{k_{12}^n + p_2^n} - k_1^{\text{mRNA}} r_1$$

$$\frac{dr_2}{dt} = v_2^{\max} \frac{k_{23}^n}{k_{23}^n + p_3^n} - k_2^{\text{mRNA}} r_2$$

$$\frac{dr_3}{dt} = v_3^{\max} \frac{k_{31}^n}{k_{31}^n + p_1^n} - k_3^{\text{mRNA}} r_3$$

$$\frac{dp_1}{dt} = k_1 r_1 - k_1^{\text{protein}} p_1$$

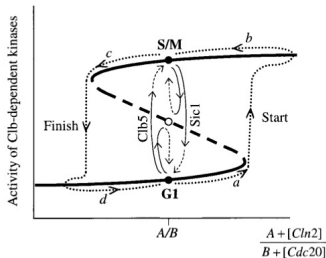
$$\frac{dp_2}{dt} = k_2 r_2 - k_2^{\text{protein}} p_2$$

$$\frac{dp_3}{dt} = k_3 r_3 - k_3^{\text{protein}} p_3$$

- mRNAs are observed, proteins are hidden
- mRNA and protein degradation rate constants are supposed to be known
- Estimate 9 parameters

ODE: analysis of the behavior

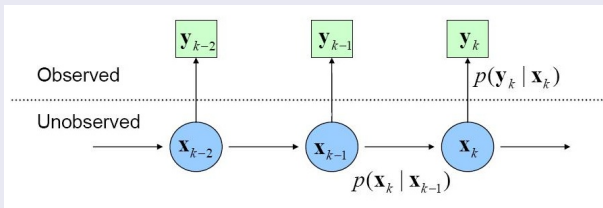
Bistability and hysteresis (example): budding yeast



Graphical models : from simple interactions models to complex ones

- Graphical Gaussian Model model : estimating partial correlation as a measure of conditional independency
- Bayesian networks : modeling directed interactions
- Dynamic Bayesian Networks : modeling directed interactions through time
- **State-space models** : modeling observed and hidden dynamical processes as well

State-space models



Equations



$$\mathbf{x}_{t+1} = \mathbf{F}_t \mathbf{x}_t + \mathbf{w}_t \quad (1)$$

$$\mathbf{y}_t = \mathbf{H}_t \mathbf{x}_t + \mathbf{v}_t \quad (2)$$

where:

- \mathbf{F}_t is the transition matrix, \mathbf{H}_t is the measurement matrix
- $\mathbf{w}_t \sim N(0, \Sigma_w)$, $\mathbf{v}_t \sim N(0, \Sigma_v)$
- and $\mathbf{x}_0 \sim N(\mathbf{x}_{0|-1}, P_{0|-1})$.

What to do with a state-space model and a dataset ?

- Infer hidden states (given parameters)
- Estimate unknown parameters: a learning problem that requires hidden states inference

Hidden state estimation: Kalman filtering and smoothing

- The problem of state estimation is addressed in the Bayesian framework
- Compute the filtering distribution $p(\mathbf{x}_t | \mathbf{y}_{0:t})$
- Method : recursive estimation
- Kalman's proposal (1960) is to recursively estimate these states using the linearity of the model and the gaussianity of distributions at hand
- Various estimates can be built from the filtering distribution: MMSE, MAP (see for instance Chen's review on filtering)

Minimum Mean Squared Error estimate

- Find $\hat{\mathbf{x}}_t$ that minimizes

$$\int (\mathbf{x}_t - \hat{\mathbf{x}}_t)^T (\mathbf{x}_t - \hat{\mathbf{x}}_t) p(\mathbf{x}_t | y_{0:t}) d\mathbf{x}_t$$

- Solution : mean posterior $E[\mathbf{x}_t | y_{0:t}]$
- Also estimate error covariance matrix

Bayesian optimal filtering

- Suppose that the parameters are known, recursively calculate the filtering distribution of the states $p(\mathbf{x}_t | \mathbf{y}_{0:t})$

Two steps

1 Prediction: $p(\mathbf{x}_{t+1} | \mathbf{y}_{0:t}) = \int p(\mathbf{x}_{t+1} | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{y}_{0:t}) d\mathbf{x}_t$

2 Update:

$$p(\mathbf{x}_{t+1} | \mathbf{y}_{0:t+1}) = \frac{p(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}) p(\mathbf{x}_{t+1} | \mathbf{y}_{0:t})}{p(\mathbf{y}_{t+1} | \mathbf{y}_{0:t})}$$

where:

$$p(\mathbf{y}_{t+1} | \mathbf{y}_{0:t}) = \int p(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}) p(\mathbf{x}_{t+1} | \mathbf{y}_{0:t}) d\mathbf{x}_{t+1}$$

Derivation of Bayesian optimal filtering for the linear case (1)

- Assuming that the posterior distribution at previous step $p(\mathbf{x}_t | \mathbf{y}_{0:t})$ is a Gaussian density $N(\mathbf{x}_t | \mathbf{m}_t, P_t)$
- Then $p(\mathbf{x}_{t+1} | \mathbf{x}_t)$ is a Gaussian density of the following form: $N(\mathbf{x}_{t+1} | F_t \mathbf{m}_t, \Sigma_w)$ using SSM equations
- How to compute $p(\mathbf{x}_{t+1} | \mathbf{y}_{0:t})$?
 - $p(\mathbf{x}_{t+1} | \mathbf{y}_{0:t}) = \int p(\mathbf{x}_{t+1} | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{y}_{0:t}) d\mathbf{x}_t$
 - Gaussianity of the two distributions gives the prediction step:
 - $p(\mathbf{x}_{t+1} | \mathbf{y}_{1:t}) = N(\mathbf{x}_{t+1} | F_t \mathbf{m}_t, F_t P_t F_t^T + \Sigma_w)$
 - Mean: $m_{t+1|t} = F_t \mathbf{m}_t$
 - Covariance: $P_{t+1|t} = F_t P_t F_t^T + \Sigma_w$

Derivation of Bayesian optimal filtering for the linear case (2)

- We need to compute: $p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1})p(\mathbf{x}_{t+1}|\mathbf{y}_{0:t})$
- $p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1})p(\mathbf{x}_{t+1}|\mathbf{y}_{0:t}) = p(\mathbf{x}_{t+1}, \mathbf{y}_{t+1}|\mathbf{y}_{0:t})$
- Using Gaussianity, we can write:
 $p(\mathbf{x}_{t+1}, \mathbf{y}_{t+1}|\mathbf{y}_{1:t}) = N([\mathbf{x}_{t+1} \mathbf{y}_{t+1}]^T, \mathbf{m}'', P'')$
 - $\mathbf{m}'' = [\mathbf{m}_{t+1|t}, H_{t+1} \cdot \mathbf{m}_{t+1|t}]^T$
 - Block matrix: $\mathbf{P}'' = [P_{t+1|t}, P_{t+1|t} \cdot H_{t+1}; H_{t+1} \cdot P_{t+1|t}, H_{t+1} \cdot P_{t+1|t} \cdot H_{t+1}^T + \Sigma_v]$

Reminder (gaussianity composition rule)

$$N(\mathbf{x} \mid \mathbf{m}, \mathbf{P}) = \frac{1}{(2\pi)^{n/2} |\mathbf{P}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{m})^T \mathbf{P}^{-1} (\mathbf{x} - \mathbf{m})\right)$$

Let \mathbf{x} and \mathbf{y} have the Gaussian densities

$$p(\mathbf{x}) = N(\mathbf{x} \mid \mathbf{m}, \mathbf{P}), \quad p(\mathbf{y} \mid \mathbf{x}) = N(\mathbf{y} \mid \mathbf{H}\mathbf{x}, \mathbf{R}),$$

Then the joint and marginal distributions are

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim N\left(\begin{pmatrix} \mathbf{m} \\ \mathbf{H}\mathbf{m} \end{pmatrix}, \begin{pmatrix} \mathbf{P} & \mathbf{P}\mathbf{H}^T \\ \mathbf{H}\mathbf{P} & \mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R} \end{pmatrix}\right)$$
$$\mathbf{y} \sim N(\mathbf{H}\mathbf{m}, \mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R}).$$

Derivation of Bayesian optimal filtering for the linear case (3)

Adding a new observation

- We want to compute: $p(\mathbf{x}_{t+1} | \mathbf{y}_{0:t+1}) = N(\mathbf{x}_{t+1} | \mathbf{m}_{t+1}, P_{t+1})$
- To obtain $p(\mathbf{x}_{t+1} | \mathbf{y}_{0:t+1})$, we notice that:
 - $p(\mathbf{x}_{t+1}, \mathbf{y}_{t+1} | \mathbf{y}_{0:t}) = p(\mathbf{x}_{t+1} | \mathbf{y}_{0:t+1})$
- As we have already computed $p(\mathbf{x}_{t+1}, \mathbf{y}_{t+1} | \mathbf{y}_{0:t})$,
- We compute the conditional distribution of \mathbf{x}_{t+1} given \mathbf{y}_{t+1} using rule 2 (gaussianity again)
- mean update:

$$\mathbf{m}_{t+1} = \mathbf{m}_{t+1|t} + K_{t+1}(\mathbf{y}_{t+1} - H_{t+1} \cdot \mathbf{m}_{t+1|t})$$

- covariance update:

$$\mathbf{P}_{t+1} = \mathbf{P}_{t+1|t} - K_{t+1} S_{k+1} K_{t+1}^T$$

- with the following notation:
-

$$S_{t+1} = H_{t+1} \cdot P_{t+1|t} H_{t+1}^T + \Sigma_v$$

Reminder (gaussianity composition rule)

If the random variables \mathbf{x} and \mathbf{y} have the joint Gaussian probability density

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim N \left(\begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix}, \begin{pmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{B} \end{pmatrix} \right),$$

Then the marginal and conditional densities of \mathbf{x} and \mathbf{y} are given as follows:

$$\mathbf{x} \sim N(\mathbf{a}, \mathbf{A})$$

$$\mathbf{y} \sim N(\mathbf{b}, \mathbf{B})$$

$$\mathbf{x} | \mathbf{y} \sim N(\mathbf{a} + \mathbf{C}\mathbf{B}^{-1}(\mathbf{y} - \mathbf{b}), \mathbf{A} - \mathbf{C}\mathbf{B}^{-1}\mathbf{C}^T)$$

$$\mathbf{y} | \mathbf{x} \sim N(\mathbf{b} + \mathbf{C}^T \mathbf{A}^{-1}(\mathbf{x} - \mathbf{a}), \mathbf{B} - \mathbf{C}^T \mathbf{A}^{-1} \mathbf{C}).$$

Recursive equations of Kalman filter

- Initialization
- Prediction step
 - $\mathbf{m}_{t+1|t} = F_t \mathbf{m}_t$
 - $P_{t+1|t} = F_t P_t F_t^T + \Sigma_w$
- Update step
 - $\mathbf{z}_{t+1} = \mathbf{y}_{t+1} - H_{t+1} \mathbf{m}_{t+1|t}$
 - $S_{t+1} = H_{t+1} P_{t+1|t} H_{t+1}^T + \Sigma_v$
 - $K_{t+1} = P_{t+1|t} H_{t+1}^T S_{t+1}^{-1}$
 - $\mathbf{m}_t = \mathbf{m}_{t+1|t} + K_{t+1} \mathbf{z}_{t+1}$
 - $P_{t+1} = P_{t+1|t} - K_{t+1} S_{t+1} K_{t+1}^T$

Estimation of hidden state (linear case)

- Kalman Filter computes the Gaussian posterior probability by computing its mean and covariance estimates
- We thus get the MMSE estimate

Improvement by smoothing

- Filtering provides an estimate of the posterior probability $p(\mathbf{x}_t|y_0, \dots, y_t)$
- Smoothing provides an estimate of the posterior probability $p(\mathbf{x}_t|y_0, \dots, y_T)$
- Recursive equations of smoothing can be derived similarly to filtering
- Smoothing is applied backward after filtering

Parameter learning

Several approaches among which:

- 1 Maximum likelihood principle with EM algorithm (also can be used for MAP)
- 2 Augmented approach : parameters as stationary state (for instance Sitz et al. 2002, Quach et al. 2007)

Dealing with nonlinear state-space models

Nonlinear model



$$\begin{aligned}\mathbf{x}(t+1) &= \mathbf{F}(t, \mathbf{x}(t); \theta) \\ \mathbf{y}(t) &= \mathbf{H}(t, \mathbf{x}(t); \theta) + \epsilon(t)\end{aligned}$$

- When \mathbf{F} , \mathbf{H} are nonlinear, the integrals are usually intractable. Approximate solutions are needed!
- Issue : estimate sufficient statistics of $p(\mathbf{x}_t | \mathbf{y}_{0:t})$
 - Extended Kalman Filter
 - Unscented Kalman Filter
 - (sequential) Monte-Carlo strategies

Unscented Kalman Filter

- S. J. Julier and J. K. Uhlmann, A New Extension of the Kalman Filter to Nonlinear Systems, 1997.
- Eric A. Wan and Rudolph van der Merve, The Unscented Kalman Filter for Nonlinear Estimation, 2000.
- Extension of UKF to a broader class of nonlinear estimation problems

Unscented Kalman Filter

Issues

- Estimate $\mathbf{m}_t, P_t, \mathbf{m}_{t+1|t}, P_{t+1|t}$ while transformations are nonlinear
- UKF is based on the Unscented Transform (UT)
- UT is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation

Unscented transform

- Consider f , a nonlinear transform of a random variable $X : Y = f(X)$
- Given $\bar{X} = E[X]$ and $P_X = E[(X - \bar{X})(X - \bar{X})]$,
- Find: $\bar{Y} = E[Y]$ and $P_Y = E[(Y - \bar{Y})(Y - \bar{Y})]$

Unscented Transform

UT algorithm (Julier, 1996)

- 1 here d is the dimension of the variable to approximate
- 2 Compute $2d + 1$ sigma points: $\{\xi_i\} = \bar{X} + (\alpha\sqrt{P_X})_i, i = 1\dots d$ and $\{\xi_i\} = \bar{X} - (\alpha\sqrt{P_X})_i, i = d + 1\dots 2d + 1$
- 3 Transform each sigma point: $\psi_i = f(\xi_i)$
- 4 Compute a weighted sample mean and covariance:
- 5 $\bar{Y} = \sum_{i=0}^{2d} \beta_i \psi_i$ and $P_Y = \sum_{i=0}^{2d} \beta_i (\psi_i - \bar{Y})(\psi_i - \bar{Y})^\top$

Illustration

Figure: Unscented Transform (Wan's paper)