

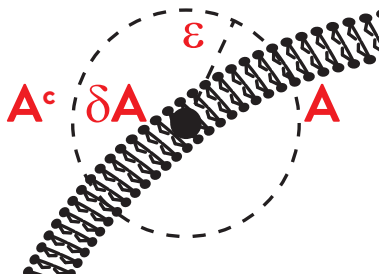
Biología de sistemas - un enfoque continuo

Elisa Domínguez-Hüttinger

Instituto de Investigaciones Biomédicas UNAM

Introducción

Sistemas biológicos: Interacción dinámica con su ambiente



Systems Biology
IIB - UNAM

"The time has come when further progress in our understanding of nature requires that we consider the relationship between the outside and the inside, between organism and environment".

"The characteristic of a living object is that it reacts to external stimuli rather than being passively propelled by them".

"The softness of the boundary between inside and outside is a universal characteristic of living systems".

Lewontin, Triple Helix, 2000.

Acople funcional y dinámico entre organismo - ambiente

$$\frac{dE(t)}{dt} = f(E(t), O(t)) \quad (1a)$$

$$\frac{dO(t)}{dt} = g(E(t), O(t)) \quad (1b)$$

- Fenotipo celular mediada por transducción de señales
- Construcción de nicho / comunidad ecológica
- Usted.

"We need to determine the norms of reaction and the role of [developmental] noise [for each case of interest]".

Lewontin, Triple Helix, 2000.

Sistemas biológicos:

- Dinámicos (diferentes escalas temporales)
- Heterogéneos (diferentes niveles de variabilidad)

entender, formalizar

Retroalimentación - regulación

"In their normal state organisms are buffered against the effects of many internal and external changes by homeostatic regulatory devices"

Lewontin, Triple Helix, 2000.

Objeto de estudio: Redes biológicas dinámicas

"a (developing) organism is a dynamical system in a dynamical structure"

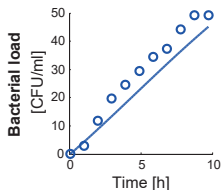
De la biología predictiva a la descriptiva



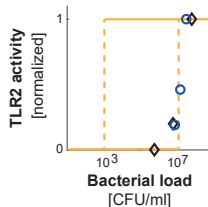
Figure: La Clairvoyance 1936. www.ReneMagritte.org

¿De dónde salen los patrones observados?

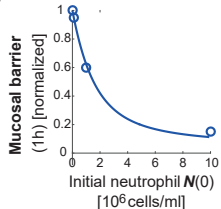
(A)



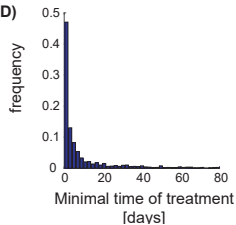
(B)



(C)



(D)



Biología de sistemas

Input
(perturbations)



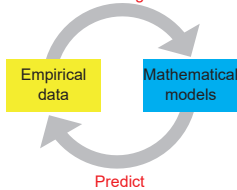
Output
(eg: phenotype)

Stimulation

Biological system:
Complex
signal
processing
network

Measurement

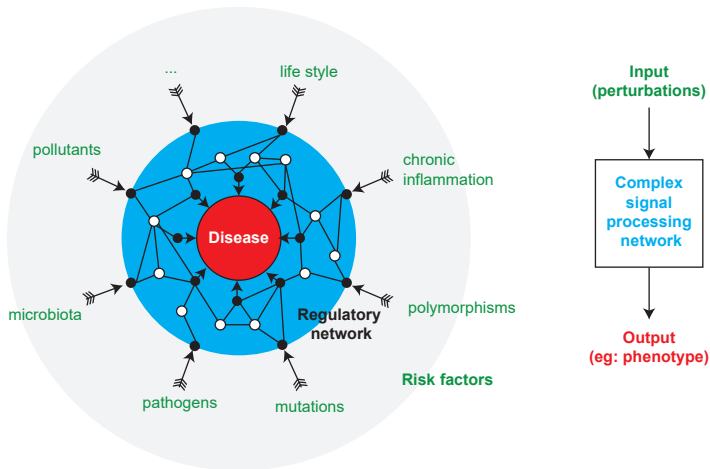
Formal integration



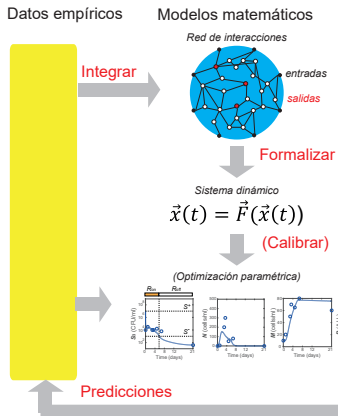
(Non-linear)
dynamical system
 $\dot{\vec{x}}(t) = \vec{F}(\vec{x}(t))$

We look for	x_i , and or t
Deterministic Trajectories $x(t)$	Discrete are Natural N
↑	↓
Stochastic Distributions $P(x(t))$	Continuous are Real R

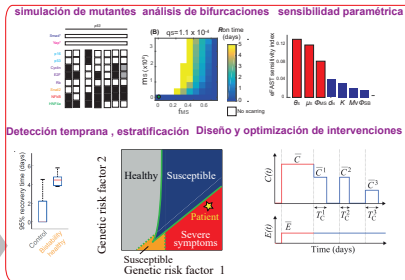
Para predecir la respuesta del sistema a perturbaciones



Systems biology pipeline



Análisis de perturbaciones



"Progress in biology depends not on revolutionary new conceptualizations, but on creation of new methodologies that make questions answerable in practica in a world of finite resources".

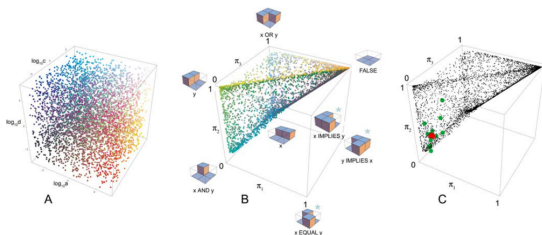
Lewontin, Triple Helix, 2000.

Modelos mecanicistas continuos: Ecuaciones diferenciales

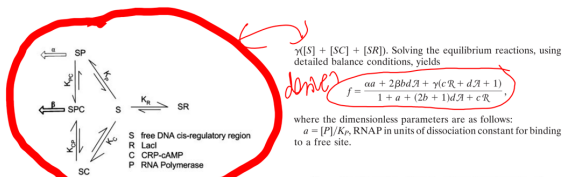
Bienvenidos al mundo real.

(sensu stricto), donde $\bar{x} \in \mathbb{R}^n$ y $t \in \mathbb{R}$. *dejamos atrás el mundo discreto booleano*

¿De dónde salen las compuertas lógicas?

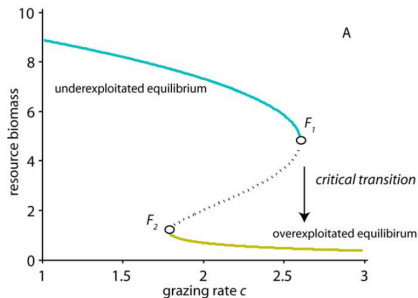


Mayo, A. E., Setty, Y., Shavit, S., Zaslaver, A. Alon, U. PLoS Biol. 4, 555–561 (2006).



Setty, Y., Mayo, A. E., Surette, M. G. Alon, PNAS 100, 7702–7 (2003).

¿Cómo se traducen cambios (ambientales) graduales en decisiones (fenotípicas) abruptas?

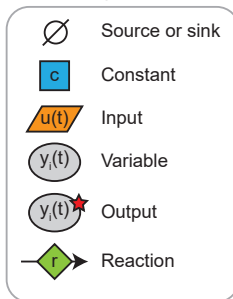


Dakos, V. et al. PLoS One 7, (2012). (¿y cómo se previenen?)

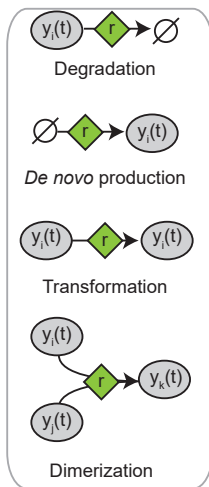
Modulación cuantitativa.

Ley de acción de masas: Notación gráfica

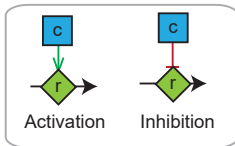
(A) Building blocks



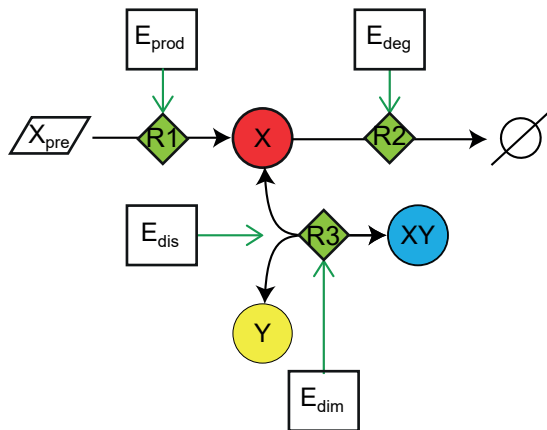
(B) Reactions



(C) Modulations

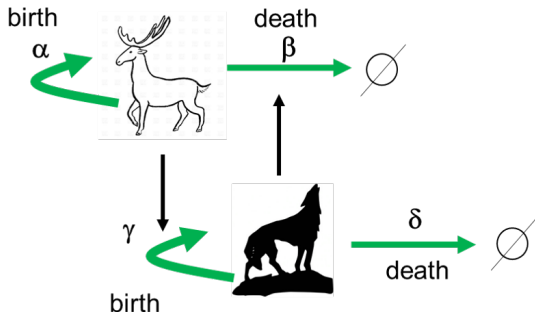


Red de interacción (ejemplo)

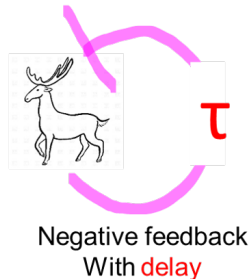


Otro ejemplo, mismo tipo de notación

Regulatory network



Control structure

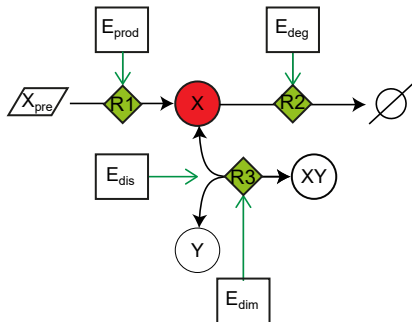


De la red al sistema de ecuaciones

$$F_i(y_1, \dots, y_n, t) = \sum_{j=1}^m R_j. \quad (2)$$

$$R_j = \prod_{p_i=1}^{p_i^{max}} (Pre_{p_i} \kappa_{p_i}) \prod_{e_i=1}^{e_i^{max}} (Enz_{m_i} \mu_{e_i}) \quad (3)$$

Red de interacción (ejemplo)



$$\frac{dX(t)}{dt} = \underbrace{X_{pre} k_{prod} E_{prod}}_{\text{R1: de novo production}} - \underbrace{X(t) k_{deg} E_{deg}}_{\text{R2: degradation}}$$

$$+ \underbrace{-X(t)Y(t)k_{dim}E_{dim}}_{\text{R3+: dimer formation}} + \underbrace{XY(t)k_{dis}E_{dis}}_{\text{R3-: dimer dissociation}} .$$

¿qué nos falta para poder describir a $x(t)$?

Información sobre sus compañeros

Por esto es biología de sistemas: No podemos entender X sin entender también a las otras variables de la red.

Sistema de ecuaciones

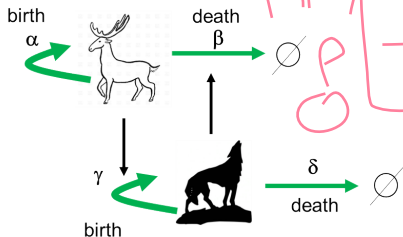
$$\frac{dX(t)}{dt} = \overbrace{X_{\text{pre}} k_{\text{prod}} E_{\text{prod}}}^{\text{R1: de novo production}} - \overbrace{X(t) k_{\text{deg}} E_{\text{deg}}}^{\text{R2: degradation}}$$

$$- \overbrace{X(t) Y(t) k_{\text{dim}} E_{\text{dim}}}^{\text{R3+: dimer formation}} + \overbrace{XY(t) k_{\text{dis}} E_{\text{dis}}}^{\text{R3-: dimer dissociation}} .$$

$$\frac{dY(t)}{dt} = - \overbrace{X(t) Y(t) k_{\text{dim}} E_{\text{dim}}}^{\text{R3+: dimer formation}} + \overbrace{XY(t) k_{\text{dis}} E_{\text{dis}}}^{\text{R3-: dimer dissociation}} , \quad (4)$$

$$\frac{dXY(t)}{dt} = \overbrace{X(t) Y(t) k_{\text{dim}} E_{\text{dim}}}^{\text{R3+: dimer formation}} - \overbrace{XY(t) k_{\text{dis}} E_{\text{dis}}}^{\text{R3-: dimer dissociation}} = - \frac{dY(t)}{dt} . \quad (5)$$

Lo mismo con nuestro ejemplo "ecológico"



prey dynamics

$$\frac{dx(t)}{dt} = \underbrace{x(t)\alpha}_{\text{R1: linear growth}} - \underbrace{x(t)\beta y(t)}_{\text{R2: death by being eaten by the predator}}, \quad (6a)$$

predator dynamics

$$\frac{dy(t)}{dt} = \underbrace{y(t)x(t)\gamma}_{\text{R3: prey-dependent predator growth}} - \underbrace{y(t)\delta}_{\text{R4: linear death}}, \quad (6b)$$

Handwritten red notes: t u(t)

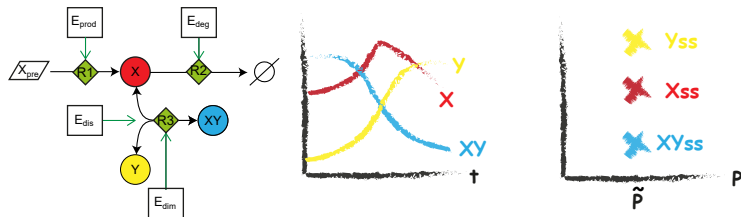
En general: Construimos sistema de ecuaciones diferenciales

Problema de valores iniciales

$$\frac{d\bar{x}(t)}{dt} = \bar{f}(\bar{x}(t), \bar{P}) \quad (7a)$$

$$\bar{x}(0) = \bar{x}_0 \quad (7b)$$

¿Qué le queremos preguntar a nuestro sistema de ecuaciones?



- Queremos encontrar $X(t)$ (y $Y(t), XY(t)$)
- Queremos encontrar X_{ss} (y Y_{ss}, XY_{ss}) ($t \rightarrow \infty$)
- Conocemos: $\frac{dX}{dt}, \frac{dY}{dt}, \frac{dXY}{dt}$

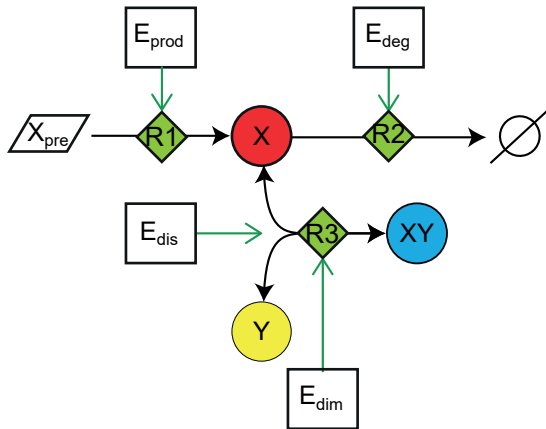
Ecuaciones de conservación

Simplificación del sistema de ecuaciones (reducción del número de dimensiones de las variables dinámicas).

$$\sum_{j=1}^k \frac{dX_j(t)}{dt} = 0 \quad (8a)$$

$$\rightarrow \sum_{j=1}^k X_j(t) = X_k^{\text{total}} = \text{constant}. \quad (8b)$$

Hint: Buscar variables que no se produzcan *de novo* o degraden (sub-sistema cerrado).



Favor de identificar a las variables redundantes.

Ecuaciones de conservación:

$$\frac{dY_T(t)}{dt} = \frac{dXY(t)}{dt} + \frac{dY(t)}{dt} = 0 \rightarrow Y_T = Y(t) + XY(t) = \text{constant.} \quad (9)$$

Reducimos el sistema original (de 3D) a 2D:

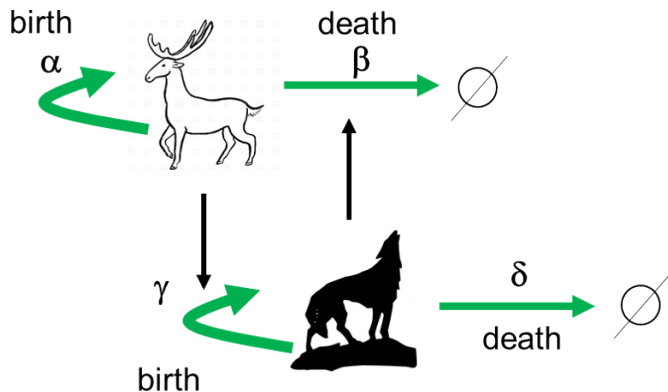
$$\frac{dX(t)}{dt} = X_{\text{pre}} k_{\text{prod}} E_{\text{prod}} - X(t) k_{\text{deg}} E_{\text{deg}}$$

$$- X(t)Y(t)k_{\text{dim}} E_{\text{dim}} + (Y_T - Y(t))k_{\text{dis}} E_{\text{dis}}, \quad (10a)$$

$$\frac{dY(t)}{dt} = -X(t)Y(t)k_{\text{dim}} E_{\text{dim}} + (Y_T - Y(t))k_{\text{dis}} E_{\text{dis}}. \quad (10b)$$

Nota: Claramente no todos los sistemas de ecuaciones pueden reducirse por ecuaciones de conservación.

Contraejemplo



Ok, regresamos a nuestra ecuación (reducida o no).

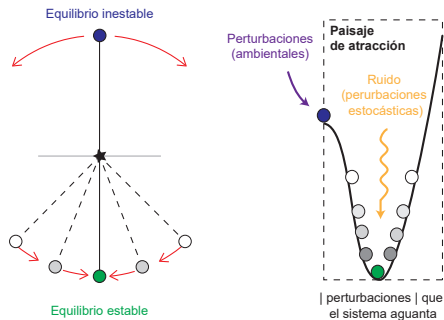
Empezamos por buscar las condiciones de equilibrio

$$\bar{f}(\bar{x}_{ss}, \bar{P}) = 0 \quad (11)$$

En general, no es sencillo (incluso, frecuentemente no es posible) resolver esta ecuación de manera analítica (polinomios de grado alto).
Recurrimos a métodos numéricos: Algoritmo de Newton-Raphson o de Continuación Numérica. **Encontrar las raíces de una función.**

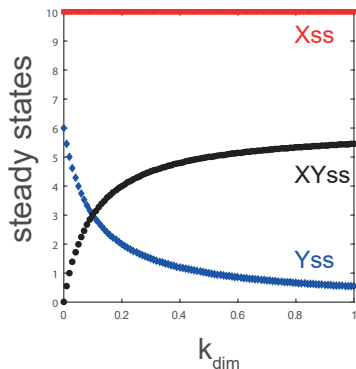
Nota breve sobre estabilidad

Una vez que encontramos los puntos de equilibrio, es importante evaluar su estabilidad. En otras palabras, nos preguntamos: ¿qué sucede si nos alejamos del equilibrio? ¿el sistema regresa? ¿cuántas perturbaciones aguanta?



Linearización (J) para evaluar la estabilidad local.

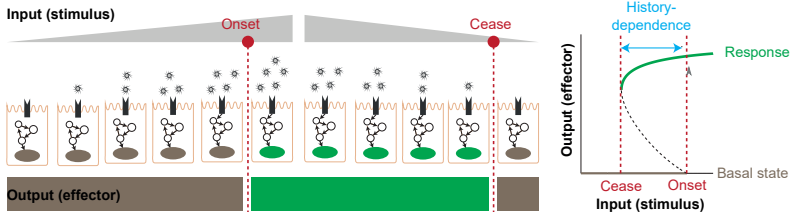
Condiciones de equilibrio



$$[X, Y, XY]_{ss} = \bar{F}(\bar{P}) \quad (12)$$

El punto de equilibrio depende de los parámetros del sistema (diagrama de bifurcación: mapea relación entre parámetro y equilibrio(s)).

Multi-estabilidad en los sistemas continuos



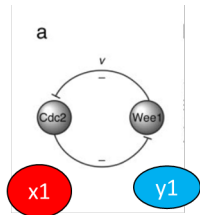
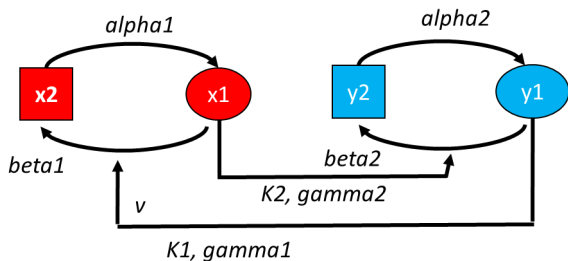
Un sistema de ecuaciones un poco más complejo (más no-linearidades):

David Angeli, James E Ferrell, and Eduardo D Sontag.

Detection of multistability, bifurcations, and hysteresis in a large class of biological positive-feedback systems.

PNAS, 101(7):1822–7, feb 2004.

Para hacer un análisis detallado de este sistema y repasar los temas vistos hasta este punto, haremos el primer ejercicio. En las notas están tanto las preguntas como las respuestas.




$$\dot{x}_1 = \alpha_1(1 - x_1) - \frac{\beta_1 x_1 (\nu y_1)^{\gamma_1}}{K_1 + (\nu y_1)^{\gamma_1}} \quad (13a)$$

$$\dot{y}_1 = \alpha_2(1 - y_1) - \frac{\beta_2 y_1 x_1^{\gamma_2}}{K_2 + x_1^{\gamma_2}} \quad (13b)$$

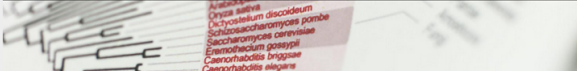
Ojo:


- Ecuaciones de conservación.
- Retroalimentación positiva y cooperatividad.


Universiteit Utrecht

Theoretical Biology & Bioinformatics

About us | Education | Research | Contact





Rob J. de Boer

Home | Research | Publications | Books | Software | CV

Grind

Phase plane analysis is a powerful approach for analyzing mathematical models comprised of a few ordinary differential equations (ODEs). grind.R is an R-script allowing one to easily study such models by numerical integration, phase plane analysis, finding and following steady states, and fitting to data. grind.R is a wrapper around R-functions developed by Martine Beber and colleagues. Grind is easy to use because it defines just a few basic functions:

- `run()` integrates a model numerically and provides a time plot or a trajectory in the phase plane.
- `plane()` shows nullclines and can provide a vector field or phase portrait.
- `steady()` finds steady states (using the Newton-Raphson method) and can provide the Jacobian with its eigenvalues and eigenvectors.
- `continue()` continues a steady state along a parameter, thus providing a bifurcation diagram.
- `fit()` fits an ODE model to data by estimating its parameters, and divides the results in a list object.

The `run()` function calls `ode()` from the deSolve package. The `fit()` function calls `model` from RNL and `newton()` and `continue()` call `steady()` from the respective packages (these 2 packages have to be installed). The `grind.R` functions have several options with well-chosen defaults (also for the state `x` and parameter `a`) of the system, with `fit()` they are typically not helpful when no options are provided. Using `library()` I can use easily create notebooks for analyzing particular models using sliders for changing parameters (see a [self-referencing example](#)).


We use grind.R in our course for biology students at Utrecht University. Because it allows them to easily analyze mathematical models, see their dynamical behavior on the screen, and export high quality graphics for reports. Researchers may also like grind.R because it allows them to easily define and study simple ODE models, add differential equations, news and events, and perform non-linear parameter estimation while fitting several data sets simultaneously having various and altered parameters (using the state of the art environment RNL).

A tutorial, the `grind.R` script, and a number of examples can be downloaded from the [Grinders](#). Grind is basically an extension of an earlier C-program called GRIND. The installation instructions of GRIND are still available on the [web page](#) for details. See the [page](#) for a translation of GRIND commands into grind.

Research groups

- De Boer group
- Snel group
- Hogeweg group
- Kaestli group
- Herman group
- Ten Tusscher group
- Dutilleul group
- Bertrug group

CONTACT >

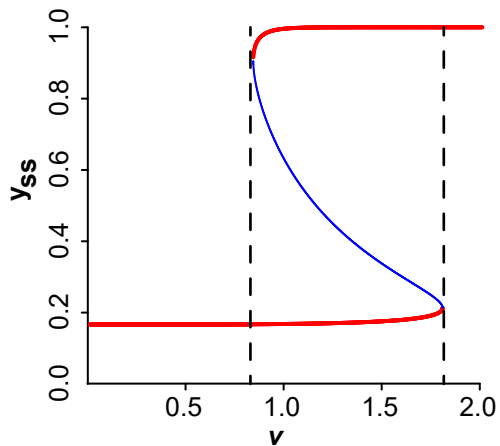

Universiteit Utrecht

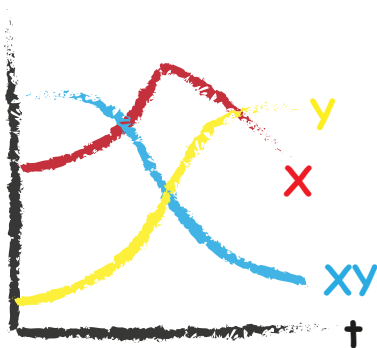
Obtener puntos de equilibrio

Algoritmo de continuación numérica para generar un diagrama de bifurcación (con GrindR)

```
BifDiagAngeli.R
```

Bistabilidad e histéresis



$\bar{X}(t)$ 

Ahora queremos reconstruir $\bar{X}(t)$. Recuerden, lo que conocemos es $\frac{d\bar{X}(t)}{dt} = \bar{F}(\bar{X}(t), \bar{P})$. Con muy pocas excepciones, utilizamos (nuevamente) **métodos numéricos** (Ej. Runge-Kutta, Euler) para deducir $\bar{X}(t)$ a partir de $\frac{d\bar{X}(t)}{dt} = \bar{F}(\bar{X}(t), \bar{P})$.

Integración numérica de ecuaciones diferenciales

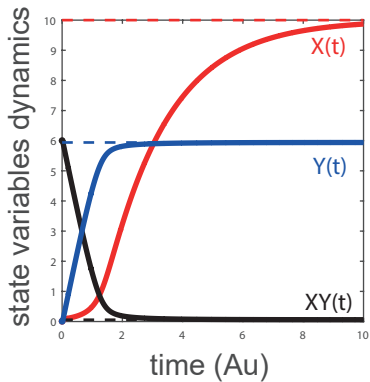
- 1 Declara la función $\frac{d\bar{X}(t)}{dt} = \bar{F}(\bar{X}(t), \bar{P})$
- 2 Declara los valores de parámetro \bar{P}
- 3 Define las condiciones iniciales $\bar{x}(t_0) = \bar{x}_0$
- 4 Define el intervalo de integración $t = [0 : \Delta t : t_{end}]$.
- 5 Invoca al integrador - para obtener $\bar{x}(t)$.

Veamos un ejemplo. Red_ejemplo_Integrar.r

```

1 library(deSolve)
2 # (1) Declara la funcion
3 RedEjemplo <- function(t, y, parms){
4   X_t=y[1]; Y_t=y[2];XY_t=(parms[6]-Y_t);
5   dX=parms[1]*parms[2]-X_t*parms[3]-X_t*Y_t*parms[4]+XY_t*parms[5];
6   dY=-X_t*Y_t*parms[4]+XY_t*parms[5];
7   list(c(dX,dY)) }
8 # (2) Declara los valores de parametro
9 Xpre=10;kprodEprod=.5;kdegEdeg=.5;kdimEdim=10;kdisEdis=1;Ytot=6;
10 parms=c(Xpre,kprodEprod,kdegEdeg,kdimEdim,kdisEdis,Ytot);
11 # (3) Define las condiciones iniciales
12 y0=c(0, Ytot);
13 # (4) Define el intervalo de integracion
14 tspan =seq(from = 0, to = 5, by = 0.01)
15 # (5) Invoca al integrador -
16 out <- ode(y = y0, times = tspan, func = RedEjemplo, parms =
17   parms)
18 # Grafiquemos los resultados
19 par(pty="s")
20 plot(out[,1], out[,2],type = "l", col="red", xlab = "Time", ylab
21   ="[X(t), Y(t), XY(t)]")

```



Lo mismo para nuestro ejemplo de Lotka Volterra

(esta vez se los muestro en Matlab; es equivalente)

LotkaVolterra_LIBRO.m

```
function LotkaVolterra_LIBRO
```

```
% Como siempre, buena costumbre limpiar nuestro espacio de  
trabajo antes
```

```
% de comenzar
```

```
close all; clear all; clc
```

```
% condicion inicial:
```

```
x0=1; y0=1; IC=[x0 y0];
```

```
%Parametros
```

```
alpha = 2/3; beta= 4/3; gamma=1; delta=1;
```

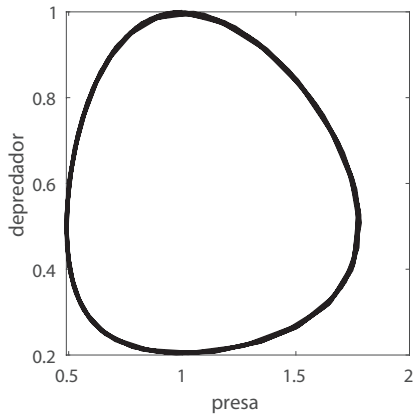
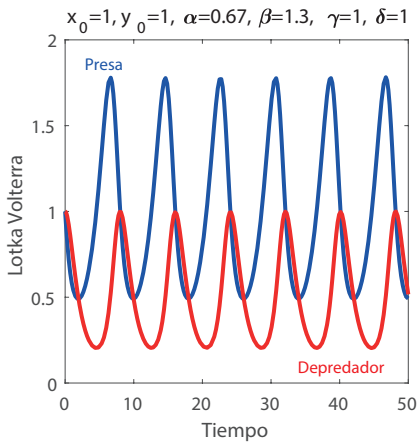
```
%tiempo de integracion
```

```
tspan = [0 50];
```

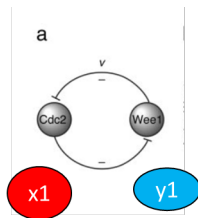
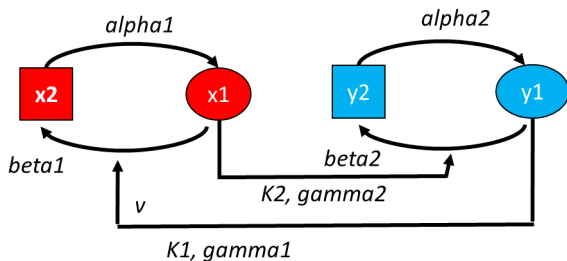
```
%integramos numericamente:
```

```
[t,y] = ode45(@(t,y)Lotka_Volterra(t,y, alpha, beta, gamma,
```

Lotka Volterra



... Y para nuestro ejemplo bi-estable

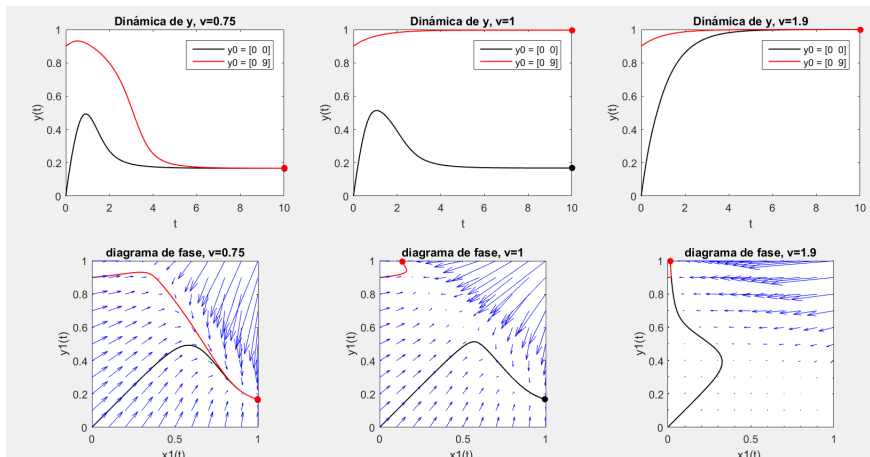


$$\dot{x}_1 = \alpha_1(1 - x_1) - \frac{\beta_1 x_1 (\nu y_1)^{\gamma_1}}{K_1 + (\nu y_1)^{\gamma_1}} \quad (14a)$$

$$\dot{y}_1 = \alpha_2(1 - y_1) - \frac{\beta_2 y_1 x_1^{\gamma_2}}{K_2 + x_1^{\gamma_2}} \quad (14b)$$

(veamos un poco cuáles son las consecuencias dinámicas de la biestabilidad)

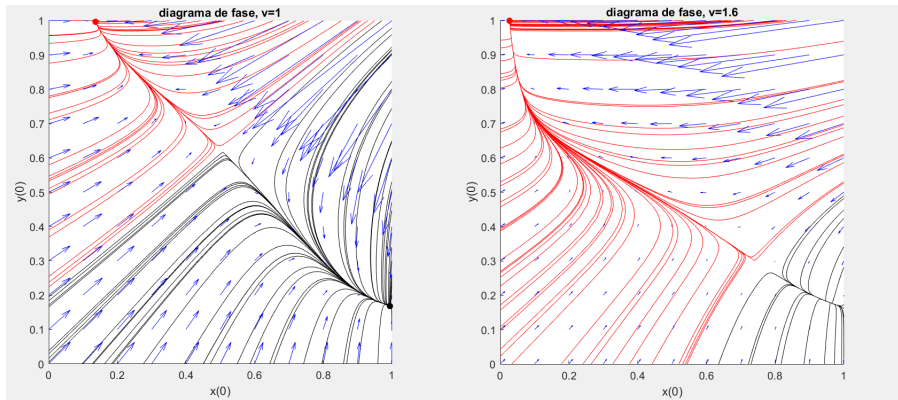
Bi-estabilidad: dependencia de condiciones iniciales



(Figuras hechas en matlab: PracticaAngeli.m; lo mismo obtendrán con Practica_2_ODEs_Angeli.r).

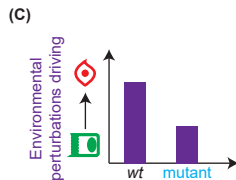
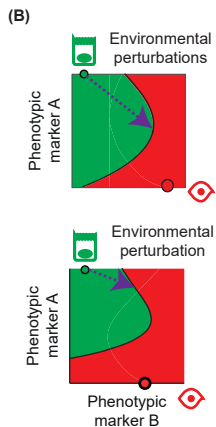
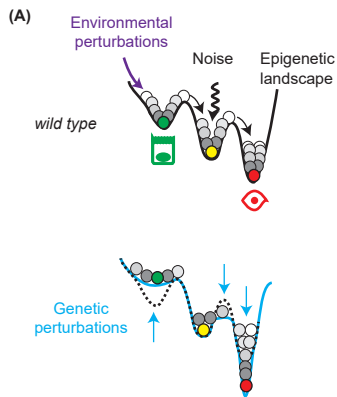
Espacio fase

Separatriz depende del valor del parámetro de bifurcación.

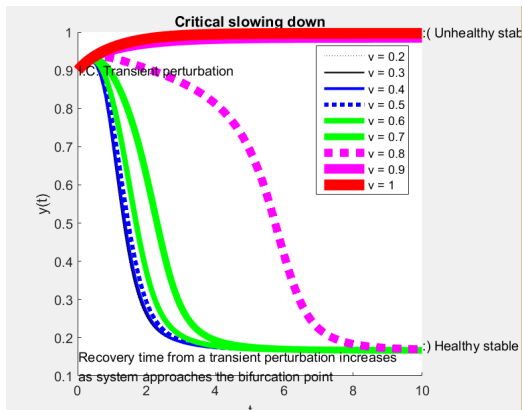


Nota: *Encontrar separatrices no es trivial.*

Vasijas de atracción



Señales de alerta temprana: Alentamiento crítico



El sistema presenta cambios en su comportamiento dinámico (y estadístico) cuando se aproxima una bifurcación. (**Teoría de catástrofes**)

Optimización paramétrica


Supongamos que tenemos la fortuna de contar con datos datos (¡Felizmente, es una suposición razonable en este curso!) Queremos encontrar el conjunto de parámetros \bar{P}) tal que nuestro modelo coincida "lo mejor posible" con los datos experimentales \bar{x}_{exp} . Dicho de otro modo, queremos **minimizar**:

$$\text{cost}(\bar{P}) = \sum_{i=1}^k (\bar{x}_{\text{exp}}(t_i) - \bar{x}(t_i, \bar{P}))^2 \quad (15)$$

Es decir, encontrar:

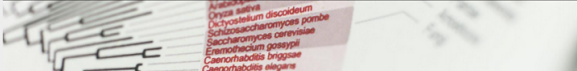
$$\min(\text{cost}(\bar{P})) = \text{cost}(\bar{P}_{\text{opt}}) \quad (16)$$


Una vez más, utilizamos para ello un algoritmo de optimización numérica, ej. **Neadler-Mead** (ojo, es local) (o la opción más elegante: usar un algoritmo global (ej. **Simmulated Annealing**).


Universiteit Utrecht

Theoretical Biology & Bioinformatics

About us | Education | Research | Contact





Rob J. de Boer

Home | Research | Publications | Books | Software | CV

Grind

Phase plane analysis is a powerful approach for analyzing mathematical models comprised of a few ordinary differential equations (ODEs). `grind.R` is an R-script allowing one to easily study such models by numerical integration, phase plane analysis, finding and following steady states, and fitting to data. `grind.R` is a wrapper around `nl2plane` developed by Martine Beber and colleagues. `Grind` is easy to use because it defines just a few basic functions:

- `run()` integrates a model numerically and provides a time plot or a trajectory in the phase plane.
- `plane()` shows nullclines and can provide a vector field or phase portrait.
- `steady()` finds steady states (using the Newton-Raphson method) and can provide the Jacobian with its eigenvalues and eigenvectors.
- `continue()` continues a steady state along a parameter, thus providing a bifurcation diagram.
- `fit()` fits an ODE model to data by estimating its parameters, and divides the results in a `lmplot`.

The `run()` function calls `ode()` from the deSolve package. The `fit()` function calls `lmfit` from `nl2plane` and `steady()` and `continue()` call `steady()` from the respective packages (these 2 packages have to be installed). The `grind.R` functions have several options with well-chosen defaults (also for the state `(x)` and parameter `(a)` of the system), with `fit()` being typically not helpful when no options are provided. Using `library()` it can also easily create nl2plane for analyzing particular models using `nl2plane` for changing parameters (see a `subdir/fit` example).


We use `grind.R` in our course for biology students at Utrecht University because it allows them to easily analyze mathematical models, see their dynamical behavior on the screen, and export high quality graphics for reports. Researchers may also like `grind.R` because it allows them to easily define and study simple ODE models, add differential equations, solve and analyze, and perform non-linear parameter estimation while fitting several data sets simultaneously having various and altered parameters (using the state of the art environment `nl2plane`).

A tutorial, the `grind.R` script, and a number of examples can be downloaded from the `grinders`. `Grind` is basically an extension of an earlier C-program called `GRIND`. The installation instructions of `GRIND` are still available on the `web page` for `distutils`. See the `page` for a translation of `GRIND` commands into `grind`.

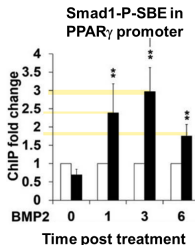
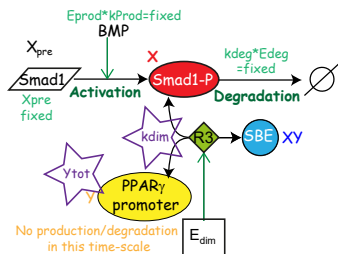
Research groups

- De Boer group
- Snel group
- Hogeweg group
- Kaestli group
- Herman group
- Ten Tusscher group
- Dutilh group
- Bertram group

CONTACT >


Universiteit Utrecht

Regresemos a nuestra red ejemplo



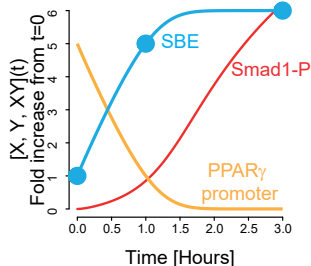
- **Data from:** Sunmee Shin, Je Kyung Seong, and Yun Soo Bae. Ahnak stimulates BMP2-mediated adipocyte differentiation through Smad1 activation. *Obesity*, 24(2):398–407, 2016.
- **Model:** our toy model (equations 27)

Optimizacion_Parametrica.r

Optimización paramétrica en R (con *GrindR*)

```
1 model <- function(t, state, parms){
2   with(as.list(c(state,parms)), {
3     dX_t=Xpre*kprodEprod-X_t*kdegEdeg-X_t*Y_t*
4       kdimEdim+(Ytot-Y_t)*kdisEdis;
5     dY_t=-X_t*Y_t*kdimEdim+(Ytot-Y_t)*kdisEdis;
6     dXY_t=-dY_t;
7     return(list(c(dX_t, dY_t, dXY_t)))
8   }) }
9
10 p=c(Xpre=10, kprodEprod=.5, kdegEdeg=.5, kdimEdim=10,
11     kdisEdis=1, Ytot=6);
12
13 s <- c(X_t=0,Y_t=5, XY_t=1) # s is the state
14
15 data <-data.frame("time"= c(0, 1, 3) , "XY_t"=c(.5, 2.5, 3 )/.5 )
16
17 w <- c("kdimEdim", "Ytot") # free parameters # to be optimized
18
19 f=fit(legend=FALSE, free=w, tstep=0.01) #method="Nelder-Mead"
```

Resultados para nuestra red ejemplo



- Error (diferencia [mínima] entre modelo y datos):
 $SSR = 2.61319 \times 10^{-14}$
- Parámetros óptimos: $k_{dim} \times E_{dim} = 2.056827$, $Y_{tot} = 1.65 \times 10^{-06}$

Optimización paramétrica: ¿para qué?

- Deducir lo que no podemos medir.
- Evaluar qué tan plausible es un modelo (+ penalización por número de parámetros + evaluar su capacidad de predicción de datos nuevos). Ver por ejemplo:

Cedersund, G. and Roll, J. Systems biology: Model based evaluation and comparison of potential explanations for given biological data. FEBS J. 276, 903–922 (2009).

Análisis de robustez

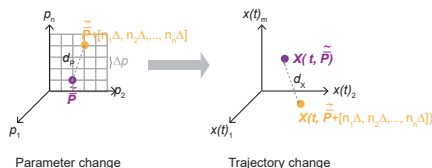


Divergencia de un estado nominal del sistemas a partir de simulación de perturbaciones (parámetros, estructural, ruido interno, forcing...)

Más adelante (modelos estocásticos) veremos cómo simular perturbaciones aleatorias.

Análisis de sensibilidad paramétrica

¿quién es el culpable? (¿qué tan robusto es el modelo ante alteraciones paramétricas?)



Queremos entender cómo se relacionan las variaciones en el espacio de parámetros \bar{p} ("genotipos") con las variaciones en el espacio de variables dinámicas $\bar{x}(t)$

- Morris's Elementary Effects:

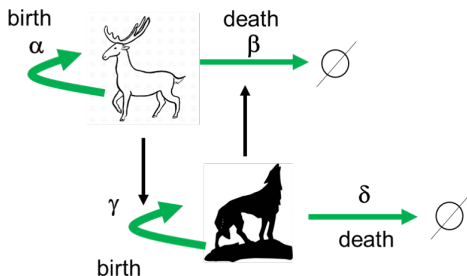
$$\frac{\partial f}{\partial x_i} \quad (17)$$

- Sobol's sensitivity index (Variance decomposition):

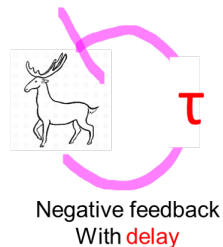
$$D_I = \text{Var}(f_I(x_I)) \text{Var}(f(x)) \quad (18)$$

Análisis de sensibilidad paramétrica: ejemplo

Regulatory network



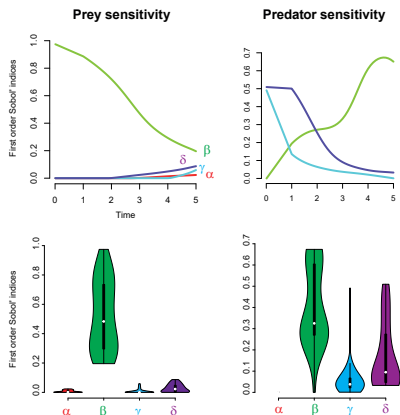
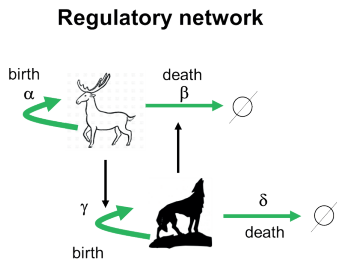
Control structure



Análisis de sensibilidad paramétrica en R

```
1 library(deSolve); library(sensitivity)
2 library(checkmate); library(ODEnetwork)
3 library(ODEsensitivity)
4 ##### Lotka-Volterra equations #####
5 LVmod <- function(Time, State, Pars) {with(as.list(c(State,
6     Pars)), {
7     dPrey      <- alpha*Prey-beta*Prey*Predator
8     dPredator <- gamma*Prey*Predator- delta*Predator
9     return(list(c(dPrey, dPredator))))}}
10 # The parameters, their lower and upper boundaries:
11 LVpars <- c("alpha", "beta", "gamma", "delta")
12 LVbinf <- c(0.05, 0.05, 0.05, 0.05); LVbsup <- c(1.00, 3.00,
13     0.95, 0.95)
14 LVinit <- c(Prey = 1, Predator = 2) #initial conditions
15 LVtimes <- c(0.01, seq(1, 5, by = 0.1)) # The timepoints of
16     interest:
17 set.seed(59281)
18 LVres_sobol= ODEsobol(mod = LVmod, pars = LVpars,state_init =
19     LVinit, times = LVtimes,
```

Análisis de sensibilidad paramétrica: ejemplo



Ecuaciones diferenciales ordinarias: Recapitulación

- Construcción de red: Nodos y procesos; "moléculas" y "reacciones". Efectores, precursores.
- Formalización a un modelo matemático: Ley de acción de masas.
 $\dot{\bar{x}}(t) = \bar{F}(\bar{x}(t), \bar{P})$
- Evaluación de puntos de equilibrio: Resolver $\dot{\bar{x}}(t) = 0$ (algoritmo de Newton-Raphson, de continuación numérica). Análisis de bifurcaciones.
- Integrar $\dot{\bar{x}}(t) = \bar{F}(\bar{x}(t), \bar{P})$ (Runge-Kutta, Euler)
- Optimización paramétrica.
- Análisis de sensibilidad paramétrica.

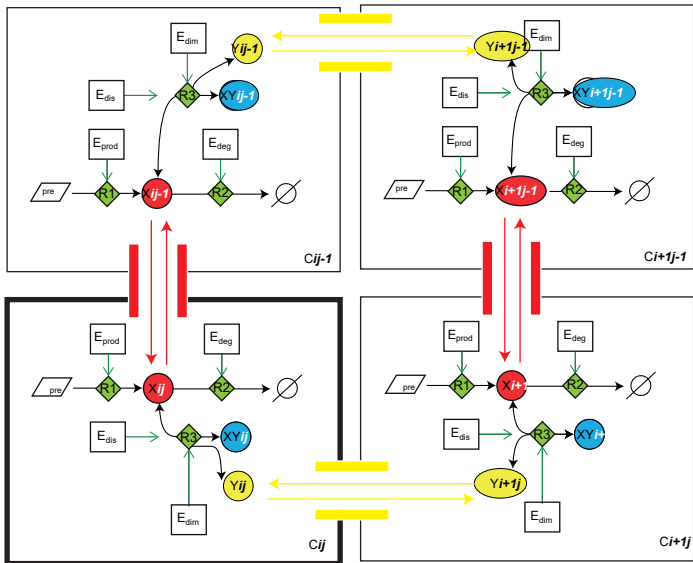
... y si eso no fuera suficiente:

Ecuaciones diferenciales ordinarias: Un buen punto de partida para abordar problemas más complejos. Dependiendo de nuestros requerimientos, es decir, **de la pregunta que queremos responder**, podemos "fácilmente" extender nuestro modelo para incluir, explícitamente:

- Heterogenidad espacial: Modelos en compartimentos.
- Forcing externo: *events function*
- Más escalas temporales: *Quasi-steady state assumption*
- Retardos: *Delay differential equations*
- Ruido
 - ▶ Intrínseco: *Chemical Master Equation* (algoritmo de Gillespie)
 - ▶ Extrínseco (muestrear parámetros y condiciones iniciales de distribuciones)
 - ▶ Estructural (ruido aditivo)

Stochastic differential equations

Heterogeneidad espacial: Modelo en compartimentos



Heterogeneidad espacial: Modelo en compartimentos

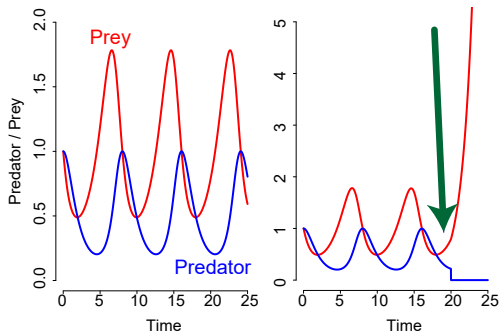
- Ley de acción de masas: "precursores de compartimentos".
- Cuidado con cómo crece el sistema con el número de compartimentos ($\#$ variables \times $\#$ de compartimentos)
- Si $\#$ compartimentos es grande ($\rightarrow \infty$): PDEs. (añadir un término difusivo: ej. ecuaciones de reacción-difusión).
- Otras opciones de modelos espaciales: modelo basado en agentes.

Forcing externo: *Events function*

Por ejemplo: (con Grind.R):

```
run(after="if (t==20)state[\"N\"]<-0")
```

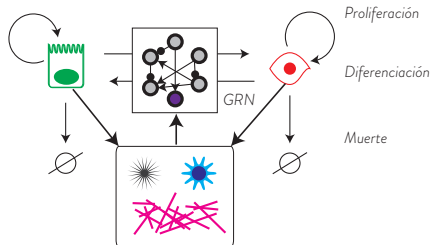
Extinción de los depredadores al tiempo 20.



Grind_events.r

Modelos multi-escala

- *Regulatory interplay between fast phenotype changes and slow microenvironmental dynamics* (muy ecológico)
- *Quasi-steady state assumption (QSSA)*
- *Events location function*



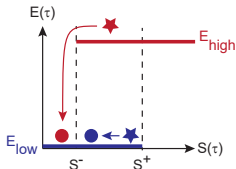
Ejemplos ecológicos:

- Sucesión ecológica (ej en REPSA);

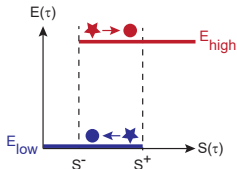
Paseando en el diagrama de bifurcación

Cuando el parámetro de bifurcación (de un sistema bi-estable) es función (dinámica) del punto de equilibrio del sistema. → Análisis de puntos focales: comportamientos emergentes (nuevas bifurcaciones)

(i) Homeostasis



(iii) Bistability



Focal points

● S_{ss}^{high}

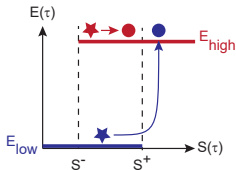
● S_{ss}^{low}

Initial conditions

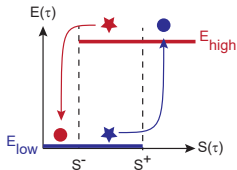
★ $S^{high}(0)$

★ $S^{low}(0)$

(ii) Chronic inflammation



(iv) Oscillations



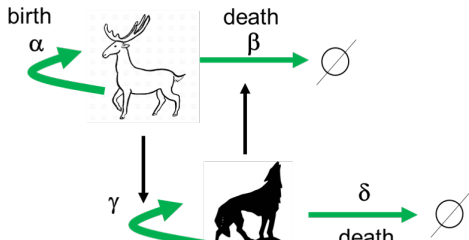
Retardos

Delay differential equations Algunas cosas toman tiempo

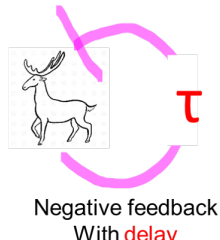
$$\frac{d\bar{x}}{dt} = \bar{F}(\bar{x}(t), x(t - \tau)) \quad (19)$$

$$\frac{dP(t)}{dt} = \alpha P(t) - \beta P(t)P(t - \tau) \quad (20)$$

Regulatory network



Control structure

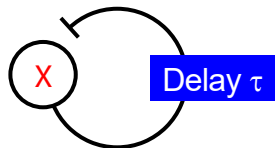
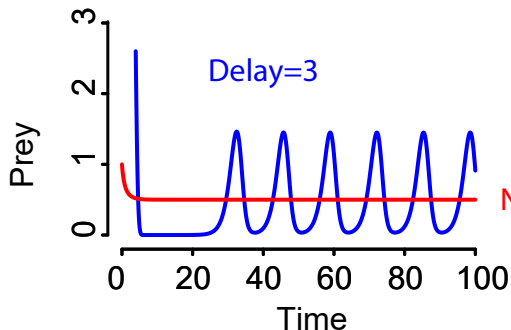


Nuevamente: Grind.R|

```
1 model <- function(t, state, parms) {
2   with(as.list(c(state,parms)), {
3     tlag <- t - Delta
4     if (tlag < 0) lags <- 0 # no initial predation
5     else lags <- lagvalue(tlag) # returns lags of P
6     dP <- alpha*P - beta*P*lags[1]
7     return(list(dP)) })}
8 p <- c(alpha=2/3,beta=4/3,Delta=3)
9 run(delay=TRUE, ylab="Prey", ymax=5, tmax=100, tstep=0.1,
10    legend=FALSE)
11 p <- c(alpha=2/3,beta=4/3,Delta=0)
12 run(delay=TRUE, ylab="Prey", ymax=5, tmax=100, tstep=0.1,
13    legend=FALSE)
```

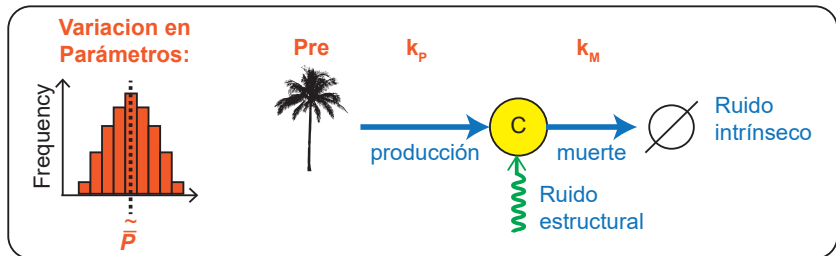
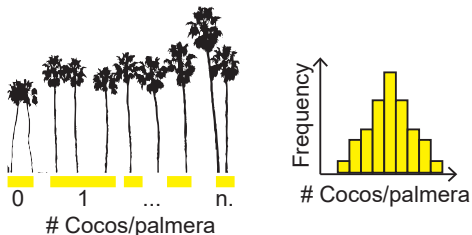
DDE_lotka_volterra.R

Emergencia de oscilaciones en un sistema uni-dimensional



Ruido.

¿de dónde salen las distribuciones poblacionales?



Fuentes de ruido (y cómo modelarlas)

- Extrínseco (muestrear parámetros y condiciones iniciales de distribuciones)
- Estructural (ruido aditivo)
- Intrínseco: *Chemical Master Equation* (algoritmo de Gillespie)

Ecuaciones diferenciales estocásticas

¿de dónde salen las distribuciones?

The rest is noise

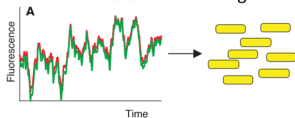


Now its about populations (Remark on Darwin; Dialectical biologist, Lewis and Lewontin)

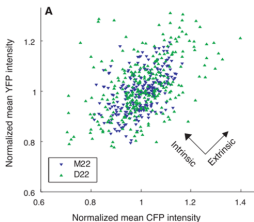
Intrinsic vs extrinsic noise

No intrinsic noise \rightarrow correlation

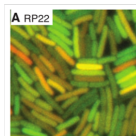
Different total amounts of $r+g \rightarrow$ extrinsic



Operationally, intrinsic noise for a given gene may be defined as the extent to which the activities of two identical copies of that gene, in the same intracellular environment, fail to correlate.

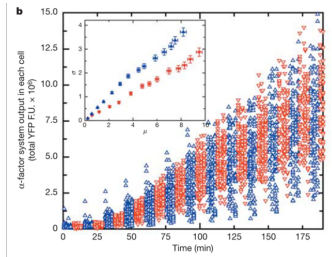
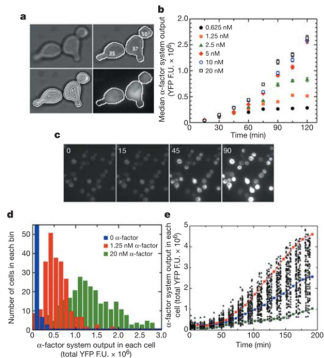


*Assuming each cell has the same number of components:
Intrinsic: variations within the same cell
Extrinsic: variations between cells.*



Elowitz, M. B., Levine, a J., Siggia, E. D. and Swain, P. S. Stochastic gene expression in a single cell. Science (80-.). 297, 1183–1186 (2002).

¿Cómo se modulan las distribuciones?

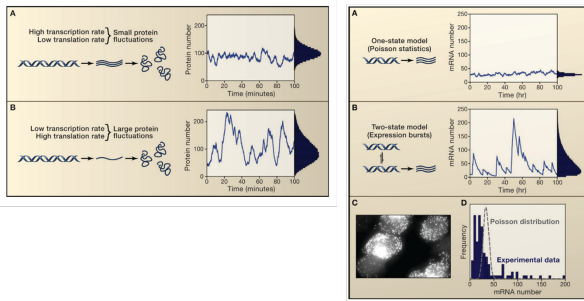


“inhibir a la molécula X aumenta el ruido”

(\rightarrow feedback negativo)

Colman-lerner, A. et al. Regulated cell-to-cell variation in a cell-fate decision system. Nature 437, (2005).

¿De dónde salen las distribuciones? (¿de qué tipo de topologías regulatorias?)



Raj, A. and van Oudenaarden, A. Nature, Nurture, or Chance: Stochastic Gene Expression and Its Consequences. *Cell* 135, 216–226 (2008).

Modulación de distribuciones (Biología sintética)

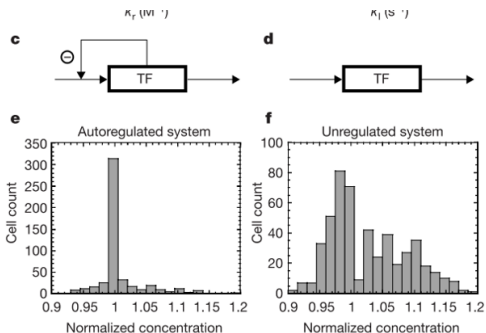


Figure 1 Stability properties of gene circuits. **a,b**, For calculating the value of relative

Becskei, a y Serrano, L. Engineering stability in gene networks by autoregulation. Nature 405, 590–593 (2000).

¡Robustez!

Importancia de las colas largas.

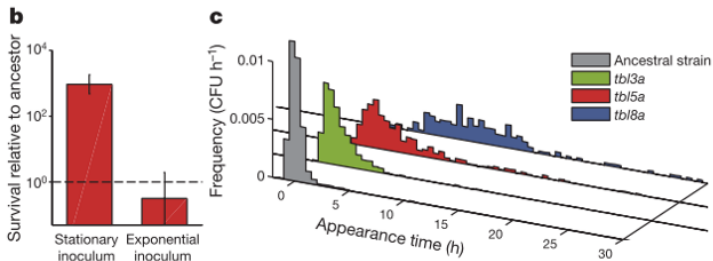
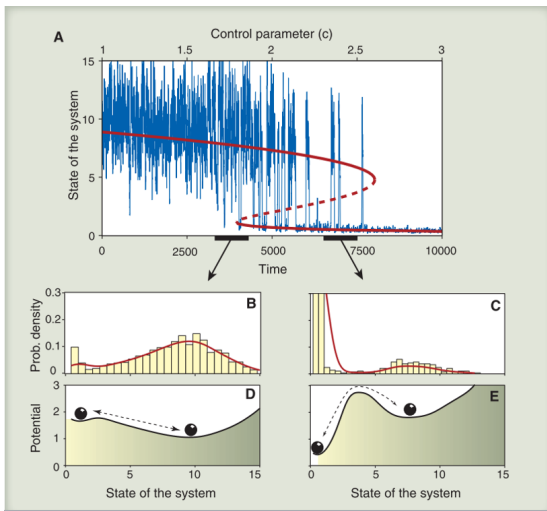


Figure 2 | Changes in the single-cell lag-time distributions underlie antibiotic tolerance. a, Phase-contrast images of time-lapse microscopy of

Fridman, O., Goldberg, A., Ronin, I., Shores, N. and Balaban, N. Q. Optimization of lag time underlies antibiotic tolerance in evolved bacterial populations. *Nature* 513, 418–421 (2014).

Marcadores estadísticos para predicción de catástrofes.



Scheffer, M. et al. Anticipating critical transitions. *Science* (80-.). 338, 344–348 (2012).

A modelar se ha dicho.

Idea general

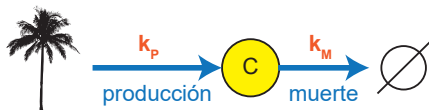
Generaremos ahora familias de soluciones: corresponden a individuos en una población. Se trata de eventos, i.e. realizaciones de un proceso estocástico. Colectivamente, generan

Distribuciones poblacionales

Wilkinson, D. J. Stochastic Modelling for Systems Biology. (Chapman and Hall, 2006).

Un ejemplo sencillo (lineal)

Producción y degradación.



¿cuál sería el modelo determinista?

$$dC(t)/dt = +K_p - K_m * C(t)$$

Hint: esto guarda relación con el Ej. 2 EDH

El sistema determinista

$$dx/dt = k_p - xk_d \quad (21)$$

Conocemos ya varias cosas:

- Su punto de equilibrio:

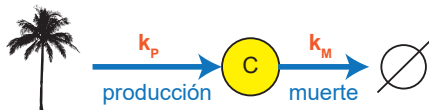
$$x_{ss} = k_p/k_d \quad (22)$$

(ustedes díganme por qué)

- Su solución analítica (disculpen, no lo vimos, pero al ser lineal esta solución \exists y es fácil encontrarla - y bueno, siempre podemos recurrir a un método numérico)

$$x(t) = k_p - e^{-k_d t} \times (k_p - k_d * x(0)) \quad (23)$$

, incluyendo su solución analítica



Ruido extrínseco

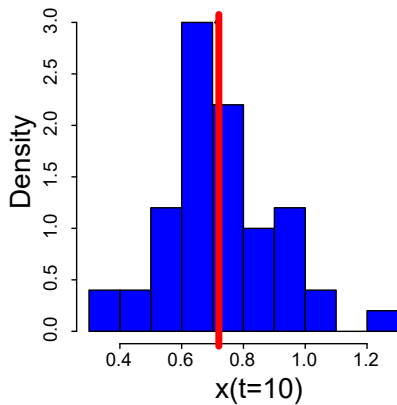
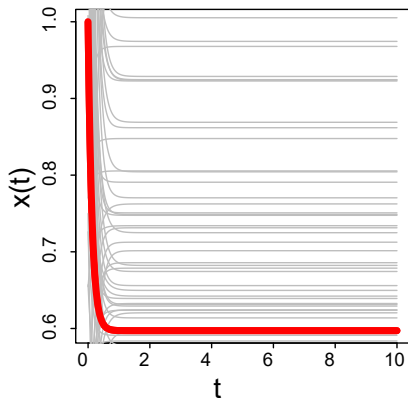
Construimos la distribución de x_{ss} (o $x(t_n)$) por medio de muestrear parámetros (y condiciones iniciales) de una distribución.

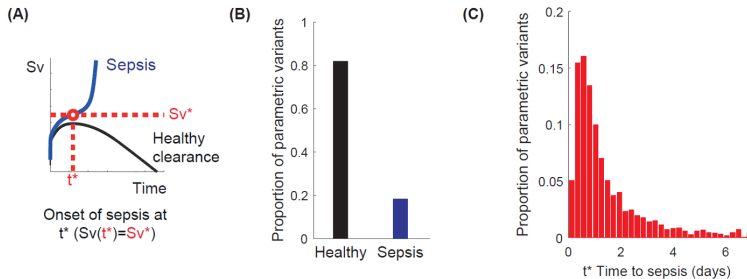
`Extrinsic_noise.R`

```

library(deSolve)
cocos <- function(t, X, parms){dX= kP-X*kD
list(dX)}
kP_nominal=5; kD_nominal=7; x0_nominal=1;
tspan =seq(from = 0, to = 10, by = 0.01)
iterations=50
x_t_family=matrix(data=NA, nrow=length(tspan), ncol = iterations)
for (i in 1:iterations){
  kP=rnorm(1, mean=kP_nominal, sd=1); kD=rnorm(1,
    mean=kD_nominal, sd=1)
  x0=rnorm(1, mean=x0_nominal, sd=1); parms=c(kP, kD)
  x= ode(y = x0, times = tspan, func = cocos, parms = parms)
  x_t_family[,i]=x[,2]
  lines(x[,1], x[,2], col="grey", type="l", lwd=1) }# end for
  iteraciones
lines(x_n[,1], x_n[,2], col="red", type="l", lwd=5,
  ylab="cocos", xlab="time" )
hist(x_t_family[1001,], freq=FALSE, col="blue", main=" ",
  xlab="x(t=10)")
points(kP_nominal/      kD_nominal, 3)

```





Domínguez-Hüttinger, E., Boon, N. J., Clarke, T. B. and Tanaka, R. J. Mathematical Modeling of *Streptococcus pneumoniae* Colonization, Invasive Infection and Treatment. *Front. Physiol.* 8, 1–14 (2017).

Ruido aditivo

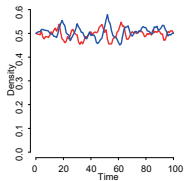
ALgoritmo de Euler-Maruyama (pasitos pequeños + muestreo de una variable distribuída gaussianamente) `GrindR_additive_noise_notes.R`
(Lotka-Volterra)

```

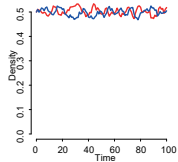
1 # Run or Source Grind.R
2 model <- function(t, state, parms) {
3   with(as.list(c(state,parms)), {
4     dR <- r*R*(1 - R/K) - a*R*N
5     dN <- c*a*R*N - delta*N
6     return(list(c(dR, dN))) })}
7 p <- c(r=1,K=1,a=1,c=1,delta=0.5) # p is a named vector of
   parameters
8 s <- c(R=1,N=0.01) # s is the state
9 ##### ADD GAUSSIAN NOISE TO BOTH VARIABLES #####
0 # START FROM THE EQUILIBRIUM STATE
1 f <- newton(c(R=0.5,N=0.5))
2 X=run(state=f,after="state<-state+rnorm(2,mean=0,sd=0.01)",
   legend=FALSE) #,ymax=1)
3 #Note that rnorm(2,0,0.01) provides two #random values, that are
   added to the two variables, respectively.
4 iterations=50
5 # prealoca la matrix para guardar los resultados de las
   simulaciones
6 x_t_family=matrix(data=NA, nrow=iterations, ncol = 2)
7 for (i in 1:iterations){

```

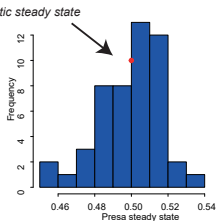
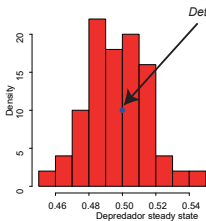

Iteration 1



Iteration 2



Final-time distribution (50 iterations)



Ruido intrínseco:

- Incluso individuos (genéticamente) **idénticos (parámetros y condiciones iniciales)** pueden diferir fenotípicamente.
- Las reacciones (procesos) son, en esencia, **eventos probabilísticos**, que ocurren con determinada **probabilidad**
- Volvemos a **discretizar** \bar{x} , y nos enfocamos en cómo cambia el **número de individuos** en cada (sub)población.
- Queremos conocer la **probabilidad de que haya x individuos/moléculas de X al tiempo t_n , para $x \in 0, 1, \dots, \mathbb{N}$ y para $t \in \mathbb{R}^+$**
- Ecuación Maestra, simulada de manera exacta con el **Algoritmo de Gillespie**

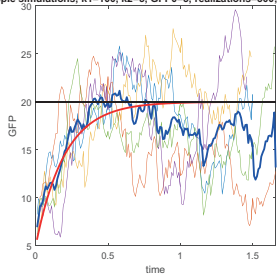
Wilkinson, D. J. Stochastic Modelling for Systems Biology. (Chapman and Hall, 2006).

Ruido intrínseco: Algoritmo de Gillespie

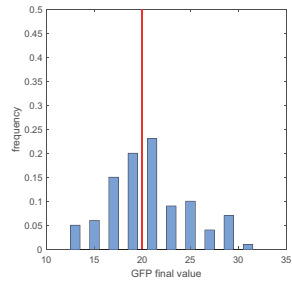
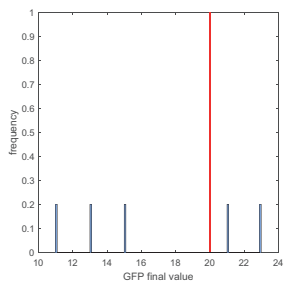
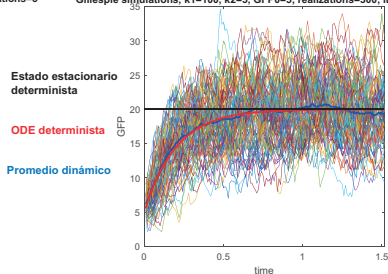
GFP_gillespie_Notas.m

```
1 k1=100; k2=5; th=[k1,k2]; %parameters (rate constants)
2 h=@(y, th)[th(1), th(2)*y]; % Hazard functions
3 x0=5; n=300;% #IC; number of steps
4 tt=0; % # initialize time
5 tvec=zeros(1,n); % # create a vector for the time, n number of
   realizations
6 xvec=zeros(1,n); % # vector for state variable
7 for i=1:1:n
8     totH= sum(h(x, th));%# total hazard
9     tt=tt+log(1/rand).*(1/ totH); %sample time to next event
   (Poisson process: wating times exponentially distributed)
10    prob_reac=h(x, th)./totH; % choose a reaction to occur
11    if rand<=prob_reac(1) % choose first reaction
12        %# update state vector
13        x=x+1; % # GFP formed
14    else
15        x=x-1; % # decay
16    end
```

Gillespie simulations, $k_1=100$, $k_2=5$, $GFP_0=5$, realizations=300, iterations=5



Gillespie simulations, $k_1=100$, $k_2=5$, $GFP_0=5$, realizations=300, iterations=100



En R

ALgoritmo de Gillespie GFP_Gillespie_Nota.r

```
1 IC=1; c1=100; c2=5;th=c(c1,c2); n=50; # Parameters / settings
2 h=function(y, th){return(c(th[1], th[2]*y))} # Hazard functions
3 # Gillespie algorithm
4 gillespieGFp <- function(IC, th, h, n)
5 {v=length(th) # number of reactions (IN THIS CASE: same as
6   number of kinetic parameters)
7 tt=0; x=IC; tvec=vector("numeric",n); xvec=vector("numeric", n)
8   # matrix for the u variables of the system
9   for (i in 1:n) {
10     totH= sum(h(x, th))# total hazard
11     tt=tt+rexp(1,totH) # sample time to next event from exp
12     probability distribution
13     j=sample(v,1, prob=h(x, th)) # choose a reaction to occur
14     # update state vector
15     if ( j==1) {x=x+1} # production
16     else # j==2
17       x=x-1 # degradation
18     # done, just fill in the vectors:
```

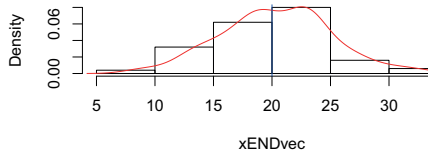
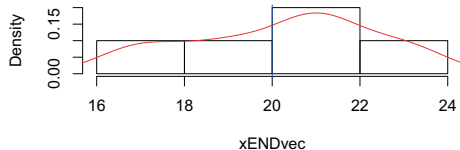
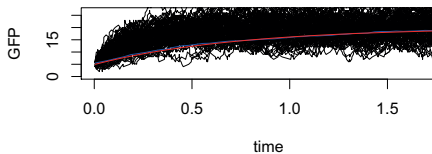
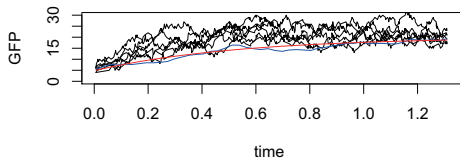


Figure: .. y en R.

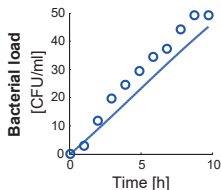
Recapitulación

De la biología descriptiva a la predictiva

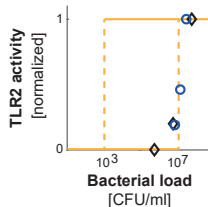


¿De dónde salen los patrones observados?

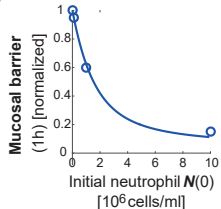
(A)



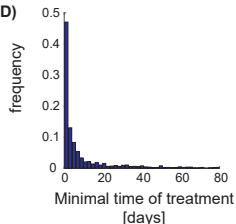
(B)



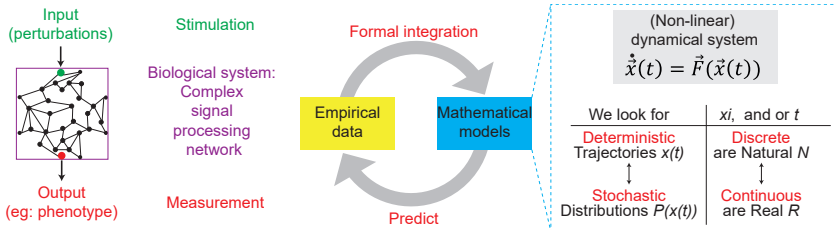
(C)



(D)

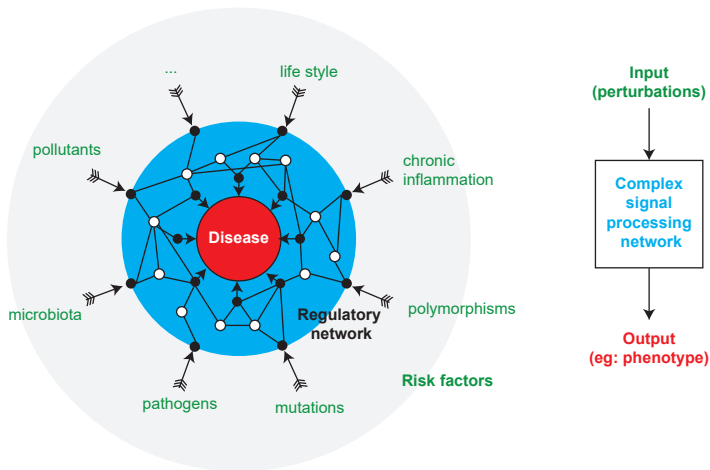


Biología de sistemas



Booleanos estocásticos y deterministas, ecuaciones diferenciales ordinarias y estocásticas, algoritmo de Gillespie.

Para predecir la respuesta del sistema a perturbaciones



Análisis de mutantes, análisis de robustez, análisis de sensibilidad paramétrica, evaluación de efectos de ruido intrínseco / extrínseco / estructural...

Ruido

- Intrínseco: *Chemical Master Equation* (algoritmo de Gillespie)
- Extrínseco (muestrear parámetros y condiciones iniciales de distribuciones)
- Estructural (ruido aditivo)

Stochastic differential equations

¿de dónde salen las distribuciones?

Trabajo en sus propios proyectos

Objetivo

Proponer una estrategia de investigación / análisis para abordar y responder alguna pregunta de investigación (ej. de su tesis)

1) Sketch the net

Input
(perturbations)



Output
(measurables)

- nodes:
- edges:
- input:
- output:

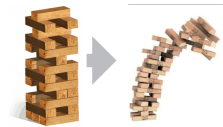
2) Select the formalism

And draft the system of equations

(Non-linear)
dynamical system
 $\dot{\vec{x}}(t) = \vec{F}(\vec{x}(t))$

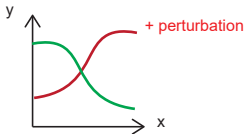
We look for	xi , and or t
Deterministic Trajectories $x(t)$	Discrete are Natural N
↓	↓
Stochastic Distributions $P(x(t))$	Continuous are Real R
-select one (ODEs, SDEs, boolean deterministic / stochastic)	
-justify your choice	

3) Design the dry experiment



- Which nominal behaviour is expected?
- Which perturbations will alter the system?

4) Draft your expected results



- What is x , what is y , what is the perturbation?
- How would your graph look in an ideal world?
- ... in a real world (with noise and experimental error)?
- in a dystopian world in which your hypothesis is wrong?

(1) Plantear la red de interacciones

A considerar:

- Nodos, interacciones
- Medibles
- Certezas vs certidumbres
- tipos de experimentos
- Perturbaciones

Output: Un dibujo de una red.

(2) Formalización

¿Qué formalismo matemático es adecuado para modelar tu red, y por qué?

"Systems biology is about making choices. The important part is to know how justify these choices: It's about conscientious decision-making."

Considera:

- Número de variables
- ¿Datos cualitativos o cuantitativos?
- Resolución temporal
- ¿Conocemos la distribución de los datos?
- ¿Qué predicciones nos importan? ¿estructurales o cuantitativas?

Output : Un formalismo (¡justificado!). (+ primer esbozo del modelo)

(3) ¿Qué análisis queremos hacer?

¿cuál es la pregunta, y cómo formalizarla? ¿su hipótesis se puede evaluar con análisis de perturbaciones?

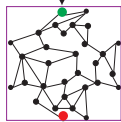
- **Mutantes** (predecir efectos de alteraciones estructurales)
- **Bifurcaciones** (predecir efectos de alteraciones cuantitativas)
- **Análisis de robustez**
- **Análisis de sensibilidad paramétrica**
- **Análisis de optimización**
- *Simulación de modelos estocásticos bifurcantes*
- **Optimización paramétrica**

(4) Resultados esperados

Nuevamente tomen una hoja o varias hojas de papel. Ahora, imagínense un mundo ideal en el que todo sucede como lo planean, y en donde todas sus hipótesis son ciertas. ¿Qué gráfica obtendrían? Hagan un esbozo de esta gráfica esperada e ideal. (en el mundo paralelo: ¿cómo se verían las simulaciones del modelo que falsean su hipótesis?).

1) Sketch the net

Input
(perturbations)



Output
(measurables)

- nodes:
- edges:
- input:
- output:

2) Select the formalism

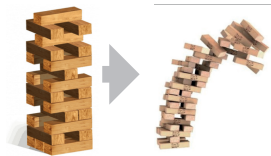
And draft the system of equations

(Non-linear)
dynamical system
 $\dot{\vec{x}}(t) = \vec{F}(\vec{x}(t))$

We look for	x_i , and or t
Deterministic Trajectories $x(t)$	Discrete are Natural N
Stochastic Distributions $P(x(t))$	Continuous are Real R

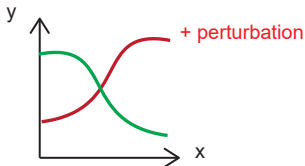
- select one (ODEs, SDEs, boolean deterministic / stochastic)
- justify your choice

3) Design the dry experiment



- Which nominal behaviour is expected?
- Which perturbations will alter the system?

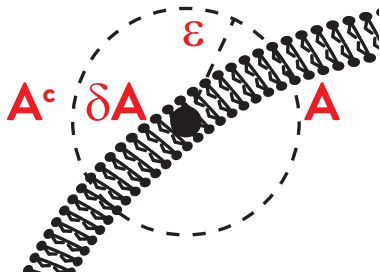
4) Draft your expected results



- What is x , what is y , what is the perturbation?
- How would your graph look in an ideal world?
- ... in a real world (with noise and experimental error)?
- in a dystopian world in which your hypothesis is wrong?

Artículos a exponer

- <http://www.icsb2018-france.com/program/invited-speakers>
- <https://www.sbhdberlin.org/speakers>



Systems Biology

IIB - UNAM