Introducción a la biología de sistemas:
Hacia una biología predictiva con modelos dinámicos no lineales

Elisa Domínguez-Hüttinger

Contents

Introducción

Esta primer sesión será interactiva. Presentaremos y discutiremos algunos conceptos e ideas centrales en biología de sistemas, así como su motivación y retos metodológicos. Los puntos a tratar están **esbozados** en estas primeras páginas; son una invitación a seguir reflexionando e investigando en el tema.

1.1 Biología de sistemas ¿Qué es(tudia)?

- Biología de sistemas: biología de interacciones.

  → Su objeto de estudio son las redes; circuitos regulatorias, de interacción.

- Estudiamos las propiedades emergentes de estas redes. ¿Ejemplos? (multiestabilidad, comportamiento periódico, excitabilidad, sinergismo. Ejemplos de ello: diferenciación celular, ritmo circadiano, potencial de acción.

  → Estas propiedades son no triviales: no se pueden deducir de estudiar solamente las partes (vs. reduccionismo).

  → Sistemas biológicos son dinámicos: Todo(s) cambian. Diferentes escalas temporales: reacciones metabólicas, regulaciones enzimáticas y genéticas, interacciones entre (poblaciones de) células, individuos, establecimiento de comunidades... Todo esto, simultáneamente (acople entre escalas temporales).

→ **Redes de interacción dinámicas**

- ¿Ejemplos de estas redes?

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

1.2 De la biología descriptiva a la predictiva (¿por qué?)

Figure 1.1: La Clairvoyance 1936. `www.ReneMagritte.org`

Queremos responder:

→ ¿de dónde salen los patrones (estadísticos) observados? ¿son comportamientos emergentes?

→ ¿cuáles son los mecanismos subyacentes? ¿cuál es la red regulatoria?

Para ello, formalizamos una hipótesis con un modelo matemático, con el objetivo de reproducir las observaciones empíricas (campo, experimentales, clínicas):

- Dinámicas: Evolución (*sensu lato*) temporal de nuestras variables.
- Curvas dosis-respuesta.
- Relación entre variables.
- Distribuciones

1.3 Necesidad de un enfoque sistémico y dinámico (¿Para qué?)

Con ello, podemos:

- Evaluar qué tan plausible es nuestra hipótesis explicativa. (darle o quitarle peso/credibilidad)
- Predecir la respuesta del sistema a perturbaciones. Por ejemplo: Estímulos ambientales, Fármacos, Mutaciones genéticas, contaminantes en un ecosistema, nutrientes, especies invasoras, etc.
  - → Procesamiento e integración de señales (input-output, sinergismo)
  - → Diagramas de bifurcación, análisis de robustez, análisis de mutantes...
  - → *Complex signal processing machinery*

→ Encontrar las fuentes y evaluar los efectos de la variabilidad de nuestro sistema biológico (modelos estocásticos: ¿de dónde vienen las distribuciones? Caracterización de diferentes fuentes de ruido (ruido intrínseco, extrínseco, estructural; afectando a subconjuntos de variables...).

→ *Parameter sweep* Pacientes virtuales.

1.4 Retroalimentación entre modelación matemática y experimentación / trabajo de campo (¿Cómo?)

- Construcción y análisis de sistemas dinámicos no lineales a partir de datos experimentales.
- Inter-disciplina, colaboración

Ejercicios

- Desde ahora pueden pensar en ir construyendo un mapa general sobre el tipo de modelos que hay; determinista estocástico discreto/ continuo, con ejemplos y tipos de preguntas que responde. Este mapa se irá expandiendo conforme avancemos en el curso.

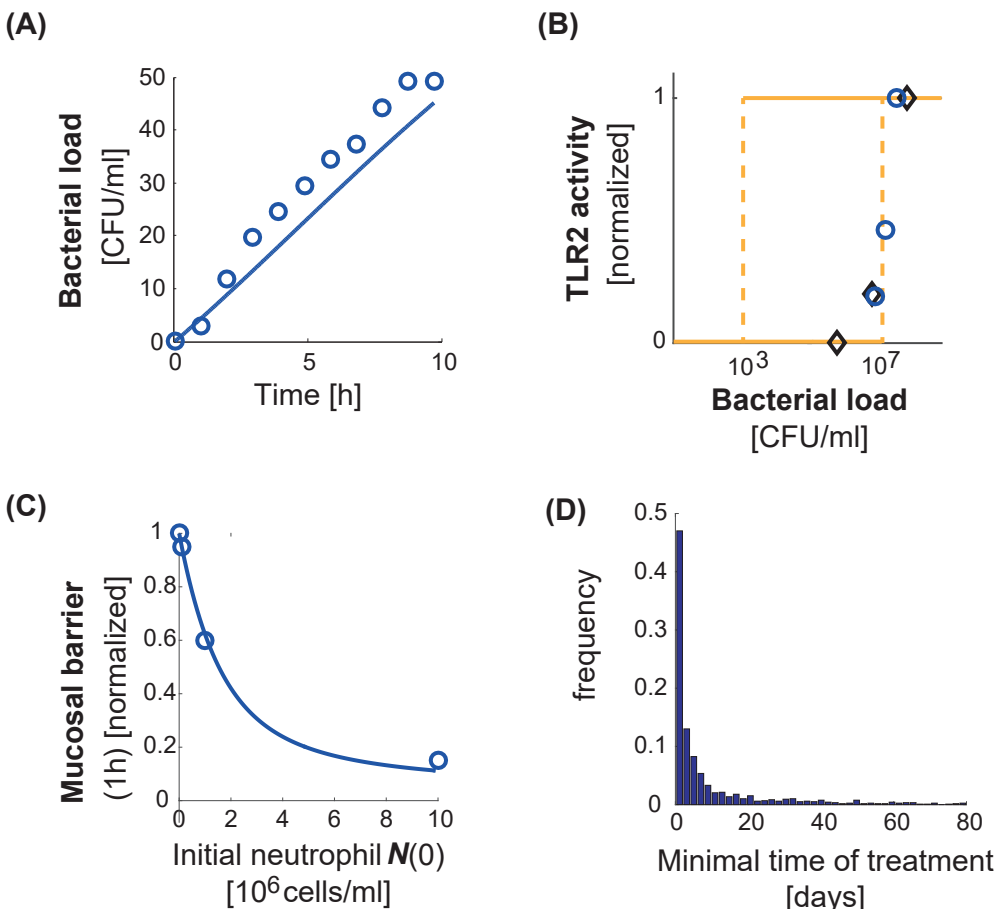


Figure 1.2: De la descripción a la predicción: Modelos matemáticos que reproduzcan (A) dinámicas, (B) Curvas dosis - respuesta, (c) Relaciones entre variables, (D) distribuciones observadas empíricamente. Modificado de `doi:10.3389/fphys.2017.00115` [11]

- Reflexionen: ¿qué es una red biológica? ¿que nos dice, a priori, su estructura? ¿qué papel juegan las conexiones? ¿qué es una entrada del sistema, y en qué difiere (o no) de una perturbación? ¿cómo definirían, operacionalmente, la salida de un sistema? Repitan este ejercicio de reflexión al final del curso.

[**EDH:** añadir :]

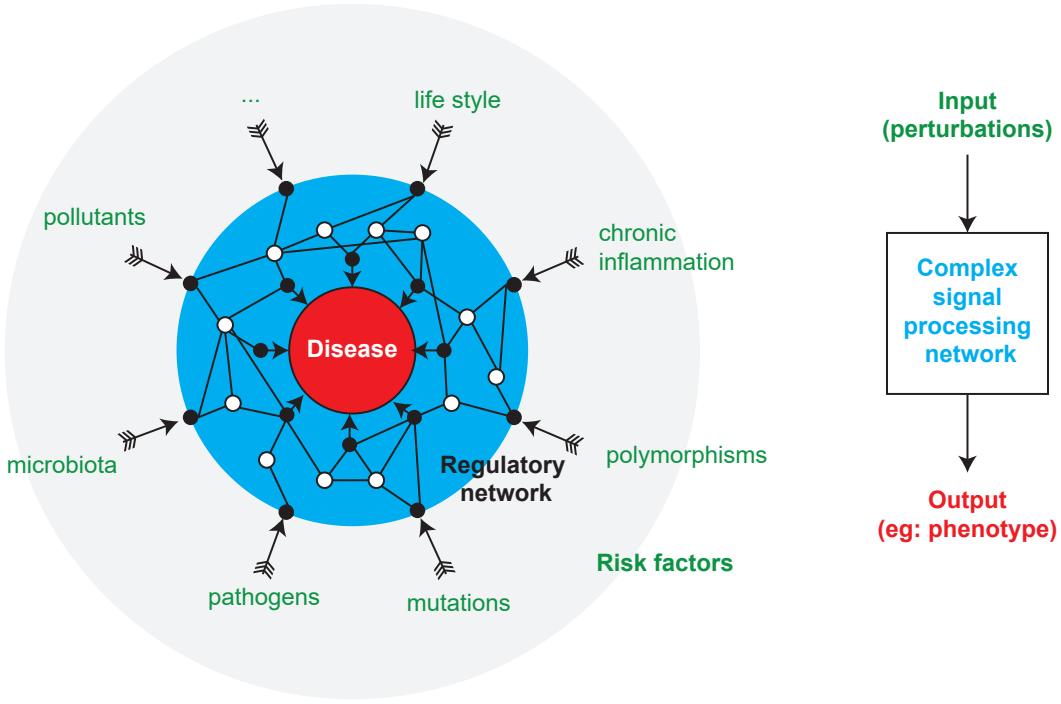


Figure 1.3: Procesamiento de señales en sistemas biológicos: Caracterizar la respuesta a perturbaciones. Modificado de: `http://link.springer.com/10.1007/978-3-319-89354-9` [2]

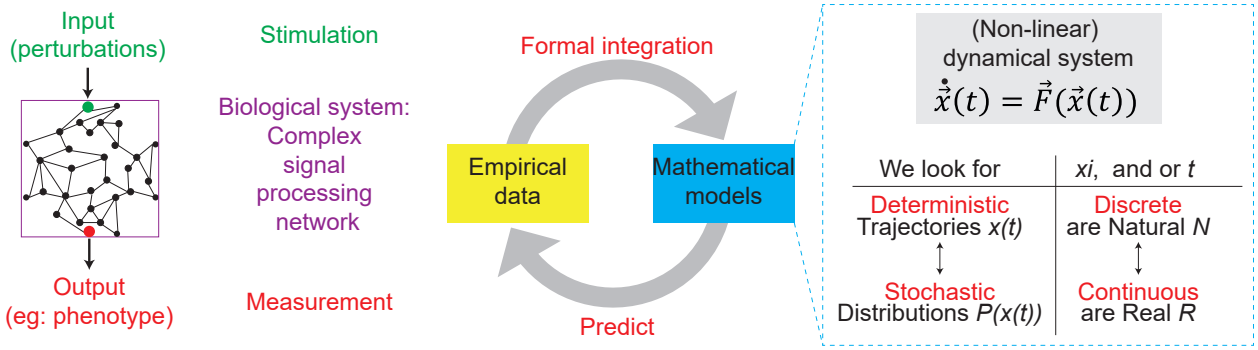


Figure 1.4: Systems biology pipeline

Sistemas de ecuaciones diferenciales ordinarias

Esta sección es una adaptación del capítulo 2.8 del libro [2]. Disculpen que esté en inglés... en algún momento lo vamos a traducir... Y disculpen que los ejemplos estén en `Matlab`.. en algún momento los pasaré a `R`... pero no se preocupen, su práctica sí será en `R`.

Building and analyzing mechanistic ODE models from scratch

In the previous section, Boolean networks were introduced. Although a very powerful to analyse multi-stability of a regulatory network, these discrete-state and discrete-time dynamical models represents some limitations when the problem we want to address with modelling requires:

- The explicit representation of the mechanisms by which the nodes/components/variables of the system interact,
- Consideration of intermediate concentrations, and/or
- Analysis of transient behaviours (see discussion in previous chapters).

In this section, we introduce continuous time and continuous state mechanistic models, namely kinetic models based on Ordinary Differential Equations (ODE). Using a simple example, we explain the pipeline for the formulation, parameter calibration and analysis of a system of ODE, namely:

1. Graphical representation of the biological system, using visual conventions that facilitate the direct translation of the network to a system of equations.

2. Mathematical representation of the biological system: translation of the network to a system of ODEs.

3. Simplification of the mathematical model: Identification of conservation equations.

4. Identification of initial conditions, parameters, experimental data and qualitative behaviours to be reproduced by the model.

5. Finding the equilibrium behaviour of the system: Steady state analysis.

6. Dynamical simulation of the system of ODEs (integration).

7. Parameter optimization: Seeking the best agreement between the mathematical model and the experimental data, using minimization algorithms.

8. Model analysis: Assessing the robustness /plasticity of the model behaviour in response to parametric variations (parametric perturbation analysis, parameter sensitivity analysis, and bifurcation analysis).

In what follows, each of these points is briefly described and exemplified. It is also noteworthy that there are some Systems Biology tools (for example, `Copasi` (`http://copasi.org` or the `SBiology Toolbox for Matlab http://www.sbtoolbox.org/`, [41]) that help to perform most of the steps along this pipeline.

2.0.1 Visual representations of reaction networks

It is a good practice to start the construction of a mathematical model with a visual representation of the system. In the case of ODE models, it is very useful to follow the visual conventions given by the Systems Biology Markup Language community [38], since there is an (almost) one-to-one relation between the graphical and the mathematical representation of the biological system. In fact, software such as `Cell Designer` (`http://www.celldesigner.org`) automatically create ODE models from a user-defined graphical representation of the biological system.
As shown in figure 2.1 (a), the basic building blocks of a system of ODEs are:

- **Constants**. The value c of the constants (per definitions) does not change, and thus there is no need to write an equation for it - they are represented in the mathematical model by parameters.

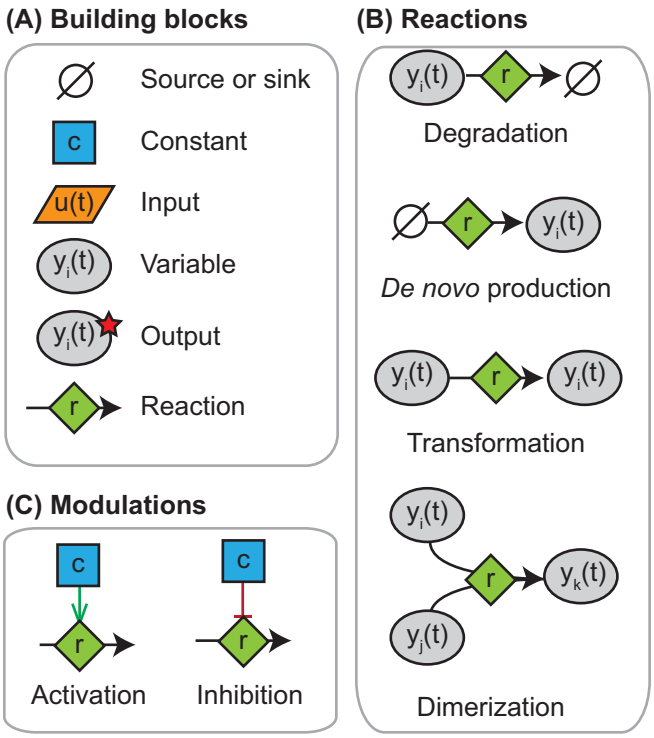


Figure 2.1: Basic building blocks to represent mechanistic reaction networks based on kinetic interactions between biomolecules

- **Inputs** The values of the inputs $u(t)$ can change, but this change is independent of the systems dynamics - it can be controlled by external conditions (for example, by the experimenter). The dynamics of the input $u(t)$ represented in the mathematical model by algebraic equations.

- **Variables** The values of the variables $y_i(t)$ change dynamically, as functions of the reactions. The system of equations describing the coupled dynamical changes of all the variables considered in the system is given by ordinary differential equations of the form $\frac{dy_i}{dt} = F_i(y_1, .., y_n, t)$.

- The **Outputs** of the system correspond to the subset of the system's variables that represent the experimentally observed features of the system that one wants to reproduce with the model (for example, levels of target gene expression). While mathematically, it is not required to distinguish between the outputs and the other variables, it might be useful to make that distinction clear in the reaction network.

- **Reactions** The reactions r_j of the system are the building blocks of the

functions describing the dynamical changes of the variables; the functions F_i are linear combinations of the reactions, i.e.

$$F_i(y_1, .., y_n, t) = \sum_{j=1}^{m} r_j. \tag{2.1}$$

- **Sources or sinks** Are used to represent the output or input of reactions that are outside of the system under consideration - for example, de novo transcription of a gene (source) or degradation of a protein (sink).

Each reaction describes the transformation of a (set of) precursor(s) a to a (set of) product(s) b. Thus, while the **concentrations** of a decrease when this reaction occurs, b increases with this reaction. A typical example of such a reaction is the conversion of a substrate into a product. x, which can be a constant, a input or a variable, is a modulator of such a reaction. This means that its presence increases (when x is an activator) or decreases (when x is an inhibitor) the rate of the reaction. The effect of x on the reaction can be additive (i.e., the reaction occurs even in the absence of x, like an *or* Boolean function), or multiplicative (the reaction occurs only in the presence/absence of x, like an (n)*and* Boolean function). It is important to point out that the concentrations of the modulator x are not affected by the reaction they affect. A classic example of a modulator is an enzyme that catalyses the conversion from a substrate a to a product b.

2.0.2 Mathematical representation of the system

To construct a mathematical model that represents the reaction network, each of the regulatory interactions must be translated into a mathematical expression -particularly, a rate. Collectively, these rates form a system of differential equations that describe the inter-dependent dynamics of the different components of the reaction network.
ODEs to describe the dynamics of the system in a *deterministic* and *continuous* manner. Translating a reaction network into a system of ODE is a standard methodology in systems biology that has been discussed widely, for example in [52, 7, 48, 36]. The basic principle used to construct the individual rates of the system is the *Law of Mass Action*. It assumes that rate of change in the concentration of species X is proportional to the concentration of precursors X_{pre}, the effectors E and the kinetic rate constants k_{i}, thus:

- A production reaction of X is represented by the term $\frac{dX}{dt} = X_{\text{pre}} k_{\text{prod}} E_{\text{prod}}$.

- The degradation of X is represented by $\frac{dX}{dt} = -Xk_{\mathrm{deg}}E_{\mathrm{deg}}$.
- The reversible dimerization of X and Y, by $\frac{dXY}{dt} = X * Yk_{\mathrm{dim}}E_{\mathrm{dim}} - XYk_{\mathrm{dis}}E_{\mathrm{dis}}$.

In general, we can write these individual reactions as:

$$R_i = \Pi_{p_i=1}^{p_i^{max}}(Precursor_{pi}\kappa_{\mathrm{p_i}})\Pi_{m_i=1}^{m_i^{max}}(Modulator_{mi}\mu_{\mathrm{m_i}}) \tag{2.2}$$

Effectors are distinguished from precursors in that their concentrations are not affected by the reaction they catalyse.
Negative regulation is often represented in a phenomenological way, by multiplying the rate on which the repressor is acting by a function that decreases monotonically with the concentration of the repressor. It would also be possible to derive this sort of functions from basic biochemistry, by explicitly representing, again using the Law of Mass Action, the mechanism by which the repressor exerts its action (depending on how the repressor acts: for instance, trapping the effector molecule). However, for convenience (less parameters and equations) this level of mechanistic detail is often omitted.
One of the main advantages of mathematical description of a reaction network is that *all* the reactions that affect X can be represented and studied simultaneously (this is why it is called a *systems biology approach*). Thus, if for example, X is being produced, degraded and also it forms a heterodimer with another molecule Y (reaction network depicted in figure 2.2), then its full dynamics are described by simply adding up the individual reactions explained above. In this particular example, this procedure would retrieve the expression that describes $X(t)$ as:

$$\frac{dX(t)}{dt} = \overbrace{X_{\mathrm{pre}}k_{\mathrm{prod}}E_{\mathrm{prod}}}^{\text{R1: de novo production}} - \overbrace{X(t)k_{\mathrm{deg}}E_{\mathrm{deg}}}^{\text{R2: degradation}} - \overbrace{X(t)Y(t)k_{\mathrm{dim}}E_{\mathrm{dim}}}^{\text{R3+: dimer formation}} + \overbrace{XY(t)k_{\mathrm{dis}}E_{\mathrm{dis}}}^{\text{R3-: dimer dissociation}} . \tag{2.3}$$

The dynamics of X given by equation 2.3 depend on *constant parameters*, such as k_{deg}, k_{dim} and k_{dis}, but also on other, time varying variables, such as $Y(t)$ and $XY(t)$. Hence, to mathematically analyse the behaviour of $X(t)$, we need to consider the equations for $Y(t)$:

$$\frac{dY(t)}{dt} = -\overbrace{X(t)Y(t)k_{\mathrm{dim}}E_{\mathrm{dim}}}^{\text{R3+: dimer formation}} + \overbrace{XY(t)k_{\mathrm{dis}}E_{\mathrm{dis}}}^{\text{R3-: dimer dissociation}} , \tag{2.4}$$

and for $XY(t)$:

$$\frac{dXY(t)}{dt} = \overbrace{X(t)Y(t)k_{\text{dim}}E_{\text{dim}}}^{\text{R3+: dimer formation}} - \overbrace{XY(t)k_{\text{dis}}E_{\text{dis}}}^{\text{R3-: dimer dissociation}} = -\frac{dY(t)}{dt}. \tag{2.5}$$

Collectively, the set of coupled equations 2.3, 2.4 and 2.5 that describe all the inter-dependent variables of the system form the system of differential equations describing the reaction network.

More generally, our example is a Initial Value Problem of the form:

$$\frac{d\bar{x}(t)}{dt} = \bar{f}(\bar{x}(t), \bar{P}) \tag{2.6a}$$

$$\bar{x}(0) = \bar{x}_0 \tag{2.6b}$$

where $\bar{x}(t)$ represents the $n-th$ dimensional vector of n model variables ($\bar{x}(t) = (X(t), Y(t), XY(t))$ in our example equations 2.3), $\bar{f}$ is the $n-th$ dimensional function describing the dynamics of $\bar{x}(t)$ (i.e. equation 2.1, for example right-hand side of our example equations 2.3), and $\bar{x}_0$ are the initial conditions.

At a first glance it might seem that thse type of LMA-based models are best suited for chemical reacions. Hoewever, it is easy to imagine that this formalism can be applied to any (biologial) network in which the rates at which the *processes* (i.e., the reactions) occur depend on the collision between state variables (nodes; molecules; species), and thus are proportional to their concentrations. For example, think about the famous Lotka-Volterra model, which describes the dynamical interactions between a prey and a predator as

$$\overbrace{\frac{dX(t)}{dt}}^{\text{prey dynamics}} = \overbrace{x(t)\alpha}^{\textbf{R1: } \text{linear growth}} - \overbrace{x(t)\beta y(t)}^{\textbf{R2: } \text{death by being eaten by the predator}}, \tag{2.7a}$$

$$\overbrace{\frac{dY(t)}{dt}}^{\text{predator dynamics}} = \overbrace{y(t)x(t)\delta}^{\textbf{R3: } \text{prey-dependent predator growth}} - \overbrace{y(t)\gamma}^{\textbf{R4: } \text{linear death}} \tag{2.7b}$$

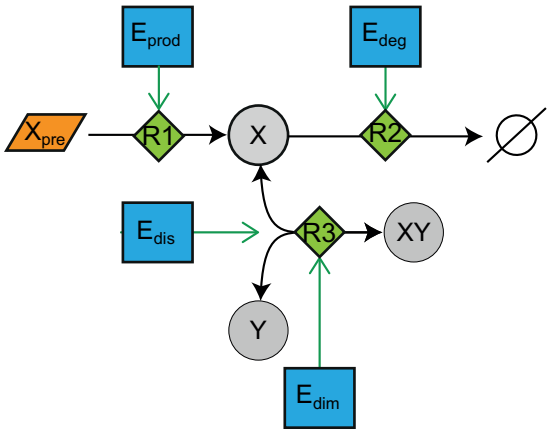


Figure 2.2: **Example reaction network.** The reaction network is given by the interactions between the time-varying biochemical species X, Y and the heterodimer XY (shown in grey circles). The concentrations of these species are determined by the reactions $R1$: *de novo* production of X from a non-rate-limiting precursor X_{pre}; $R2$: degradation of X, and $R3^{+/-}$: and reversible formation of the heterodimer XY. These reactions are catalysed by the enzymes E_{prod}, E_{deg}, and E_{dis} and E_{dim} (blue squares), whose concentrations are assumed to remain constant. Figure taken from [10] (URI: `http://hdl.handle.net/10044/1/47969` (published under a Creative Commons Attribution Non-Commercial No Derivatives Licence `https://creativecommons.org/licenses/by-nc-nd/3.0/`

2.0.3 Model simplification: identification of the conservation equations

Let's try to make some simplifications by identifying sets of variables (or species) that together, do not change over time. In other words, we are looking for sub-sets of $\bar{x}$ that satisfy:

$$\sum_{j=1}^{k} \frac{dX_j(t)}{dt} = 0 \tag{2.8a}$$

$$\rightarrow \sum_{j=1}^{k} X_j(t) = X_k^{\text{total}} = constant. \tag{2.8b}$$

Equations 2.8 are known as **conservation equations**, because they describe conserved amounts of species in a system. Dynamical systems in which

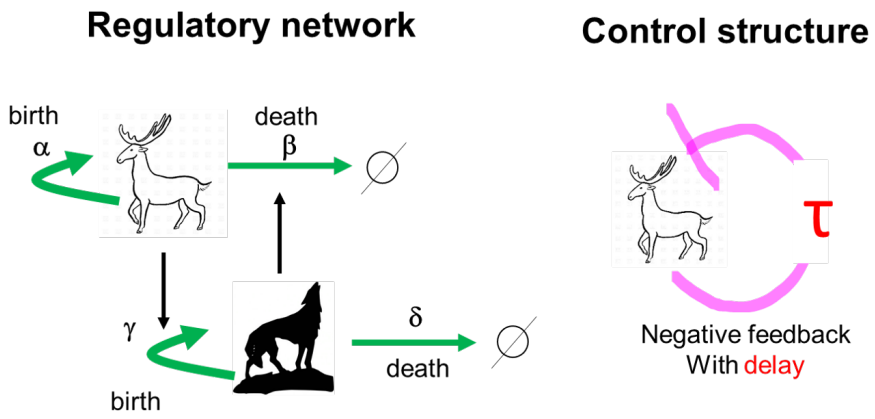


Figure 2.3: Lotka Volterra Model

sets of species are related by such equations can be simplified, since equation 2.8 implies that (at least) one of the variables X_ξ of the subset X_k can be described algebraically, as a function of X_k^{total} and $X_j, j = 1, ..., k-1$:

$$X_\xi(t) = X_k^{\text{total}} - \sum_{j=1}^{k-1} X_j(t) \tag{2.9}$$

and thus, there is no need to solve (integrate) $\frac{dX_\xi(t)}{dt}$. The n-th-dimensional system of ODEs 2.6 can then be simplified to an $n-1$th dimensional system of ODEs with one algebraic equation 2.9.

Note however that not all systems can be simplified by conservation equations, only those in which at least some of the variables are neither produced nor degraded.

In our example, we can see already from the reaction network in figure 2.2 that although variable X is being produced *and* degraded, neither the monomer Y nor the heterodimer XY are being produced *de novo* nor degraded - they are good candidates for our conservation equations. Looking at the corresponding ODEs, we can indeed $\frac{dXY(t)}{dt} + \frac{dY(t)}{dt} = 0$. In other words, the total amount of Y is conserved, i.e. the sum of free Y and heterodimer-Y (XY)

is constant, and thus, defining Y_{T} as the total amount of Y ($Y_{\mathrm{T}} = Y + XY$):

$$\frac{dY_{\mathrm{T}}(t)}{dt} = \frac{dXY(t)}{dt} + \frac{dY(t)}{dt} = 0 \rightarrow Y_{\mathrm{T}} = Y(t) + XY(t) = constant. \quad (2.10)$$

These conservation equations 2.10 imply that $Y(t) = Y_{\mathrm{T}} - XY(t)$ (or, equivalently, $XY(t) = Y_{\mathrm{T}} - Y(t)$) and thus, there is no need to solve (integrate) $\frac{dY(t)}{dt}$ (or $\frac{dXY(t)}{dt}$) to obtain $Y(t)$ (or $XY(t)$). The 3-dimensional system of ODEs given by the coupling of equations 2.3, 2.4 and 2.5 can then be simplified to the 2-dimensional system of ODEs:

$$\frac{dX(t)}{dt} = X_{\mathrm{pre}} k_{\mathrm{prod}} E_{\mathrm{prod}} - X(t) k_{\mathrm{deg}} E_{\mathrm{deg}} - X(t) Y(t) k_{\mathrm{dim}} E_{\mathrm{dim}} + (Y_{\mathrm{T}} - Y(t)) k_{\mathrm{dis}} E_{\mathrm{dis}}, \quad (2.11a)$$

$$\frac{dY(t)}{dt} = -X(t) Y(t) k_{\mathrm{dim}} E_{\mathrm{dim}} + (Y_{\mathrm{T}} - Y(t)) k_{\mathrm{dis}} E_{\mathrm{dis}}. \quad (2.11b)$$

2.0.4 Identification of the initial conditions, parameters, experimental data and qualitative behaviours to be reproduced by the model

Before proceeding to the analysis of the system, it is necessary to gather all the relevant experimental information to which the model simulations will be compared. Although this is an important step in all types of mathematical modelling, it is particularly important for mechanistic ODEs, since its behaviour can be strongly affected by the chosen model parameters. Indeed, while the behaviour of the boolean network models we revised earlier in this book (namely, number, composition and topology of the attractors) depend solely on the network structure (i.e, there is one and only one behaviour per boolean model), depending on the parameter choice a single ODE-based model can have many different behaviours. For example, the model of the NFκB response pathway proposed in [30] can show either an oscillatory or a bistable behaviour. Also, the model of Atopic dermatitis [12] can display monostability, bistability or oscillations. Thus, prior knowledge on the initial conditions, critical parameters, experimental data and expected qualitative behaviours to be reproduced by the model is very useful to constrain the analysis of the system of ODEs. For example:

- What is the typical range of concentrations /numbers in which we expect to find the model variables?
- How is the input-output relation? i.e. are dose-response curves available?
- Is there any critical parameter known to drastically change this input -output response? (mutations, further environmental factors..)
- How is the dynamic response of the measurable output to changes in the input? what is the time-resolution of these experiments?

2.0.5 Finding the equilibrium behaviour of the system: Steady state analysis

We will start the mathematical analysis of our system of ODEs by identifying the steady state behaviour of the system. Per definition, when a system is in steady state, its rate-of-change is equal to zero. These **steady state values** $\bar{x}_{\text{ss}}$ satisfy that $\bar{x}_{ss}(t) = \bar{x}_{ss}(t + \Delta t) \forall \Delta t > 0$ - equivalent to the *attractors* of the boolean networks. These conditions are fulfilled when the rate of change of $\bar{x}$, given by $\frac{d\bar{x}(t)}{dt} = \bar{f}(\bar{x}(t), \bar{P})$ (equations 2.6), is equal to zero, i.e., if $\bar{x}_{\text{ss}}$ is a steady state value then $\frac{d\bar{x}_{\text{ss}}}{dt} = 0 \rightarrow \bar{f}(\bar{x}_{\text{ss}}, \bar{P}) = 0$.

To obtain the steady state value(s) $\bar{x}_{\text{ss}}$, it thus necessary to solve the algebraic equation:

$$\bar{f}(\bar{x}_{\text{ss}}, \bar{P}) = 0 \tag{2.12}$$

For some systems of equations, it is possible to analytically derive an expression for $\bar{x}_{\text{ss}}$ as a function of parameters. I.e., by solving 2.13, one can obtain the function $G(\bar{P})$ such that:

$$\bar{x}_{\text{ss}} = G(\bar{P}) \tag{2.13}$$

For example, the steady state behaviour $[X_{\text{ss}}, Y_{\text{ss}}]$ of our reaction network depicted in figure 2.2 and described by the system of equations 2.11 is obtained by simultaneously solving: $\frac{dX(t)}{dt} = 0$ and $\frac{dY(t)}{dt} = 0$:

$$X_{\text{ss}} = \frac{E_{\text{prod}}\, X_{\text{pre}}\, k_{\text{prod}}}{E_{\text{deg}}\, k_{\text{deg}}}, \tag{2.14a}$$

$$Y_{\text{ss}} = \frac{E_{\text{deg}}\, E_{\text{dis}}\, Y_{\text{tot}}\, k_{\text{deg}}\, k_{\text{dis}}}{E_{\text{deg}}\, E_{\text{dis}}\, k_{\text{deg}}\, k_{\text{dis}} + E_{\text{dim}}\, E_{\text{prod}}\, X_{\text{pr}}e\, k_{\text{dim}}\, k_{\text{prod}}}. \tag{2.14b}$$

While the expressions for X_{ss} and Y_{ss} (eqs. 2.14 can be easily obtained by hand, it is often useful to use software to obtain such steady state values. In the box 2.0.5, we exemplify how `Matlab` can be used to obtain expressions 2.14:

Using software to compute analytical expressions of steady states

```
% (1) declare parameters as symbolic variables
syms Xpre kprod Eprod kdeg Edeg kdim Edim kdis Edis Ytot
% ... and also the variables
syms X Y

% (2) write the equations:
XY=Ytot-Y;
dXdt=Xpre*kprod*Eprod-X*kdeg*Edeg-X*Y*kdim*Edim+XY*kdis*Edis;
dYdt=-X*Y*kdim*Edim+XY*kdis*Edis;

% (3) obtain the steady states
[Xss, Yss]=solve([dXdt==0, dYdt==0], [X, Y]);
```

These expressions (eqns. 2.11) can be used to assess the parameter dependency on the long term behaviour of the system. For example, figure 2.4 shows the steady state behaviour of the example kinetic reaction network as a function of the dimerization constant k_{dim} from 0 to 1 while keeping the other parameter values constant ($X_{pre} = 10, k_{prod} = 1, E_{prod} = .5, k_{deg} = 1, E_{deg} = .5, E_{dim} = 10, k_{dis} = 1, E_{dis} = 1, Y_{tot} = 6$). Increasing the dimerization constant k_{dim} leads to a monotonous decrease in Y_{ss} (equation 2.14b), mirrored by an increase in XY_{ss}. The values of X_{ss} are unaffected by changes in this parameter.

Note that, in general, neither the existence nor the uniqueness of a steady state can be guaranteed.

For example, consider the simplest, zero-order ODE describing the constant production of $B(t)$ ($\emptyset \rightarrow B$ in our graphical notation 2.1) $\frac{B(t)}{dt} = a, a > 0$. This equation has no steady state solution (i.e. $\nexists B_{ss} \frac{B_{ss}(t)}{dt} = 0$) - which makes sense, since we are considering a constant production of B.

In the other extreme, there are also many biologically relevant systems that (depending on parameter choices) can display have *multiple* steady states. As

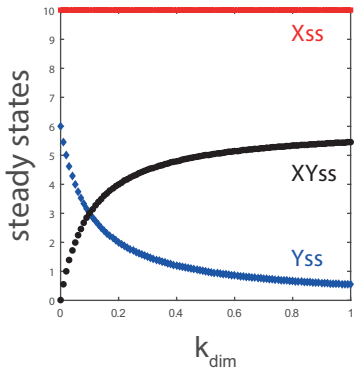


Figure 2.4: **Steady state behaviour of the example kinetic reaction network as a function of the dimerization constant** k_{dim} Increasing the dimerization constant k_{dim} leads to a monotonous decrease in Y_{ss} (equation 2.14b), mirrored by an increase in XY_{ss}. The values of X_{ss} are unaffected by changes in this parameter. (Parameter values: $X_{\text{pre}} = 10, k_{\text{prod}} = 1, E_{\text{prod}} = .5, k_{\text{deg}} = 1, E_{\text{deg}} = .5, E_{\text{dim}} = 10, k_{\text{dis}} = 1, E_{\text{dis}} = 1, Y_{\text{tot}} = 6$ and $k_{\text{dim}} = [0 : 0.01 : 1]$).

in boolean network models displaying multiple attractors, such multi-stable ODE systems can be used to represent the phenotypic plasticity displayed by biological systems, and are hence a particularly important class of ODE models. The following box 2.0.5 is devoted to its description.

Bistability - fragmentation of the phenotype space

A switch-like dose response behaviour refers to the relation between a input (commonly, a ligand) and the steady state concentrations of an output, where small changes in the input can drive large changes in the output [23]. A particular class of such a switch like behaviour is *bistability*, in which this abrupt change in output concentration is also *history-dependent*. In such a bistable dose-response behaviour, the critical concentration of the input at which the abrupt switching *onset* from low to high values occurs is different from the critical input concentration that triggers the *ceasing* of the switch, back from high to low values. The region between the two threshold values for cease and onset of the output response is termed *bistable region*, because the output can have two possible values, high or low, depending on the *previous* values of the output; if previous values

are low, then the system remains at the low branch, and *vice versa*. This property confers the system with *memory*, also termed as *hysteresis*, since the current state depends on past values (fig. 2.5).

For cellular systems, the existence of bistability (or multi-stability in general, as discussed in previous sections of this chapter) has enormous functional implications. If a state of a cell is interpreted as a phenotype, then the multi-stability of a cellular system corresponds to the spectrum of different phenotypes that can be attained by a particular cell with a particular *reaction network configuration*. Each of these states, or phenotypes, has an associated *basin of attraction*, the size of which is related to the stability and the robustness to stochastic, intrinsic (eg. genetic) and environmental perturbations (fig. 2.6 (A)).

Indeed, using the genetic deficiency as a bifurcation parameter, it is possible to systematically assess how the properties of the basin of attraction of a particular cell state are affected by the strength of the genetic deficiency [3] (analogously to the analysis of the changes in the epigenetic landscape elicited by genetic perturbations, as schematically shown in fig. 2.6 (B)). Additionally, given that external (environmental) perturbations are required to force the system from one state to the other, forcing the state to cross the sepparatrix that divides the basins of attraction (fig. 2.6 (B)), with a mathematical model one can assess how the genetic mutations change the *minimal magnitude or duration* of an external challenge required to drive a phenotype transition is affected by underlying genetic perturbations (fig. 2.6 (C)). Such a *susceptibility* analysis can be used to characterize how disease progression can result from the complex interplays between genetic and environmental risk factors (fig. **??**).

In the case of GRN, the co-existence of multiple attractors of the corresponding discrete-time and discrete-state boolean models (discussed in previous sections of this chapter) provide a good description of the fragmentation of the phenotypic space associated to transcriptional control In this section, we focus on the continuous time and continuous state ODE models. It is important to point out that due to the theoretical and computational constrains of the current methods to analyse systems of non-linear differential equations (which give rise to multi-stability), in general a ODE approach to analyse multi-stability should be favoured over a boolean approach only if the system under study and the number of attractors or phenotypes to be described are small enough, and if there is a explicit need to analyse the network assuming continuous time and continuous

state variables. For example, many small signalling networks controlling abrupt, all-or-nothing phenotypic have been modelled with ODEs which result in bistability. Examples include apoptosis [26, 15], cell cycle progression [68], commitment to meiosis [14], oocite maturation [19, 18], quorum sensing in bacteria [64], and immune responses elicited by Dendritic Cells [43], keratinocytes in psioratic [62] and Atopic Dermatitis [54] lesions, T cells [33, 28], lymphocytes [67], chondrocytes [37], macrophages [49] and endothelial cell [35], among others. In general, computational [65, 42, 57] and theoretical [8, 3] analysis of these and other biochemical networks have shown that bistability can result from biochemical networks displaying positive feedback with cooperativity.

In principle, to determine if a ODE system shows bistablity, it is enough to solve equation 2.13 and find that there are three (two locally stable, one locally unstable; stability can be determined by the analysis of the corresponding Jacobian matrix) steady state solutions. However, in practice, most of the ODE systems that can show bistablity are highly non-linear (indeed, non-linear positive feedback interactions are required for such a qualitative behaviour [8, 3]), and thus, the solution to equation 2.13 cannot be determined analytically. Numerically, one can either integrate the ODEs from varying initial conditions (discussed in the next section), or compute the different roots of equation 2.13 (eg. using the Newton–Raphson algorithm). The difficulty lies on the fact that one cannot know *a priori* if a given ODE system with a specific parameter set shows bistability -even if it displays the structural features for this behaviour. Thus, for each parameter set, multiple initial conditions (numerical integration) or initial guess (numerical determinations of the roots of the algebraic equation) have to be tested.

Fortunately, despite these difficulties, there are several software toolboxes which can be used to construction of such bifurcation diagrams. Examples of such tools are `Matcont` or the
`Dynamical Systems Toolbox`, both for `Matlab`, `Oscill8`, `XPPAUT`, `GRIND` and `COPASI`. All of these programs use *numerical continuation algorithms* to computationally approximate the long term behaviour of the non linear ODEs as a function of a bifurcation parameter [61, 31]. Such results can be graphically represented by bifurcation diagrams, which in a bistable system describe the abrupt switching between two stable steady states when the bifurcation parameter reaches its *cease* or *onset* thresholds (fig. 2.5).

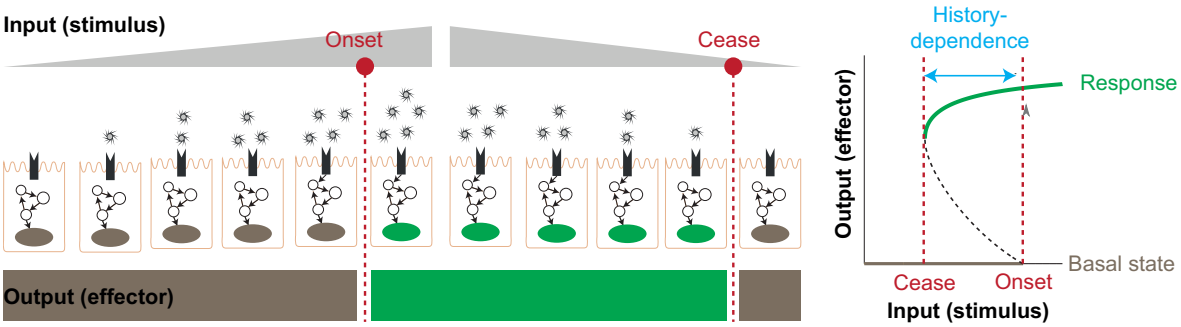


Figure 2.5: Schematic representation of a bistable dose-response (bifurcation) behaviour describing the relation between the concentration of the input (stimulus) and the steady state concentration of the output (effector). Effector concentrations remain at low values until a critical threshold for onset in the stimulus concentration is reached, triggering the abrupt activation of the effector. High effector values can be decreased only if the ceasing threshold ($<$ onset threshold) is reached. The history-dependent region comprised between ceasing and onset thresholds is termed bistable region. Example of such bistable dose-response behaviour is the abrupt and history-dependent onset and cease of innate immune responses that characterize AD flares and are triggered in response to pathogens that come in contact with viable epidermal cells.

2.0.6 Dynamical simulation of the system of ode

When we formulate an ODE-based mechanistic model, such as in the coupling between equations 2.3, 2.5 and 2.4, what we describe are the **rates of change** of the dynamic variables, i.e. $\frac{d\bar{x}(t)}{dt}$ with $\bar{x}(t) = [X(t), Y(t), XY(t)]$.

But wait! What we actually want to know is not exactly $\frac{d\bar{x}(t)}{dt}$, but rather $\bar{x}(t)$. So, what we have to do is to *deduce* $\bar{x}(t)$ from the information that we have, i.e., from $\frac{d\bar{x}(t)}{dt}$. This is, we have to integrate, or solve, $\frac{d\bar{x}(t)}{dt}$. How? There are bad news and good news.

The bad news is that the vast majority of the biologically relevant mechanistic ODE based models are non-linear systems of ODEs describing the rate of change of more than one state variable - and such models don't have an analytical solution [48]. This means that we cannot aim to obtain a *function* $\bar{G}(t, \bar{P})$ such that $\bar{x}(t) = \bar{G}(t, \bar{P})$.

The good news is that there are many numerical methods to estimate $\bar{x}(t)$ from $\frac{d\bar{x}(t)}{dt}$ (and the initial conditions $x(0)$), for example, the Euler or the Runge-Kutta methods [46]. It is beyond the scope of this book to discuss in

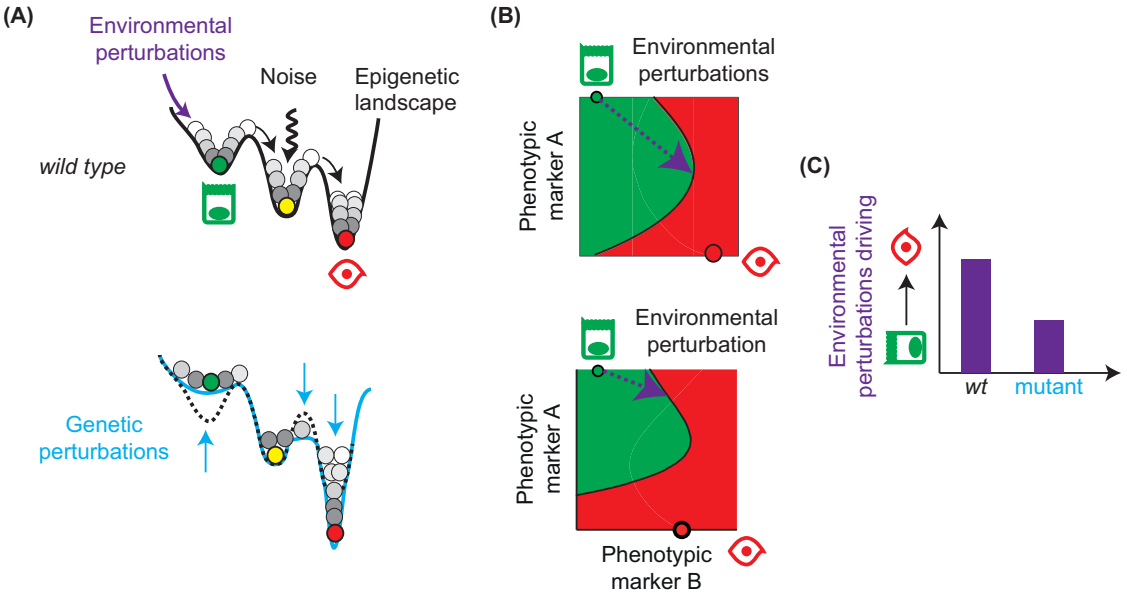


Figure 2.6: Genetically (or micro-environmentally driven) changes in the size and structure of the basins of attraction affect the susceptibility of changing the phenotype in response to environmental perturbations.

detail how these methods work (interested readers are referred to [48] or [46]), but the basic principles are illustrated with the simplest method of method for numerical integration of ODEs, namely the Euler method, as follows:

Consider the Initial Value Problem given in eqs. 2.6. To numerically integrate this model, one takes a current, known value $\bar{x}(t_0) = \bar{x}_0$, and using the knowledge about the expected dynamics, comprised in the derivative $\bar{f}(\bar{x}(t), \bar{P})$, one can estimate the value in the next time-step $t_1 = t_0 + \Delta t$ (with $\Delta_t \to 0$ a sufficiently small time step) as $\bar{x}(t_0 + \Delta t) = \bar{x}_0 + \Delta t \bar{f}(\bar{x}(t_0), \bar{P})$. Repeating this procedure iteratively from t_0 to t_n, one can obtain an numerical estimation for the dynamics for $\bar{x}(t)$. The smaller the time step, the more accurate the calculation - but, if done manually, also more cumbersome the procedure!

Fortunately, there are many software choices with build -in ODE solvers. For example, the statistical programming language `R`, already widely discussed in previous sections of this book, has a library (`deSolve`) with many functions to preform this task [46]. Also `Matlab` has many build-in functions for this (for example, `ode45`), and extensive documentations and examples.

In any case, regardless of the software used to integrate toe ODE numerically, the steps are:

1. Declare the ODE function (eq. 2.6a)
2. Define the parameter value ($\bar{P}$)
3. Define the initial condition ($\bar{x}(t_0) = \bar{x}_0$)
4. Define the integration interval (note: most of the computational ODE solvers "chose" Δ_t, hence it is only necessary to set the initial and the final values.
5. Call the ODE solver - and obtain $\bar{x}(t)$.

Let's see how these steps can be implemented in `Matlab`.

First, we write the system of ODEs (eqs. 2.11 in a separate `m-file`. The name of the file should be the name of the function - in our case, we call it `dimerFormation.m`.

```
function dydt=dimerFormation(~,y ,Xpre, kprodEprod, kdegEdeg,
    kdimEdim, kdisEdis, Y_tot)

dydt = zeros(2,1);

X_t=y(1);
Y_t=y(2);
XY_t=(Y_tot-Y_t);

dydt(1)=Xpre*kprodEprod-X_t*kdegEdeg-X_t*Y_t*kdimEdim+XY_t*kdisEdis;
dydt(2)=-X_t*Y_t*kdimEdim+XY_t*kdisEdis;
```

Now we are ready for the numerical integration:

```
% (1) Define the constant parametre values
Xpre=10; kprodEprod=.5; kdegEdeg=.5; kdimEdim=10; kdisEdis=1;
    Ytot=6;
% (2) Define the initial condition
y0=[0 Ytot];
% (3) Define the integration interval
tspan = [0 5];
% (4) Call the ODE solver
```

```
[t,y] = ode45(@(t,y)dimerFormation(t,y ,Xpre, kprodEprod,
    kdegEdeg, kdimEdim, kdisEdis, Ytot),tspan,y0);
```

We can then visualize the results, obtaining in our case the dynamic trajectories for $\bar{x}(t) = [X(t), Y(t), XY(t)]$ shown in figure 2.7.

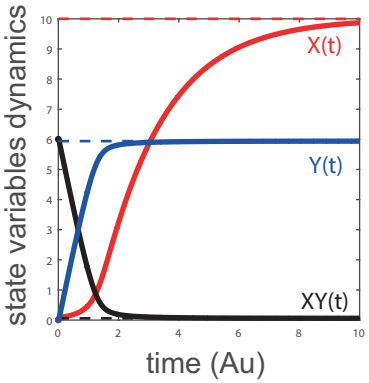


Figure 2.7: Dynamic behaviour of the example kinetic reaction network with the settings specified in the code 2.0.6. The horizontal dotted lines represent the steady state values (eqs. 2.14).

Using the tools that we have introduced so far, it is straightforward to analyse a more "ecologically meaningful" example, the Lotka-Volterra system of ODEs (see equations 2.7 and figure 2.3):

```
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,
    delta),tspan,IC);

% pintamos la funcion
figure
plot(t, y(:,1),'color','b', 'LineWidth',2)
xlabel('Tiempo');
ylabel('Lotka Volterra');
hold on
plot(t, y(:,2),'color','r', 'LineWidth',2)
legend('Presa', 'Depredador')
axis square
set(gcf, 'Position', [100 100 300 300]);
title(['x_0=' num2str(x0) ', y_0=' num2str(y0), ', \alpha='
    num2str(alpha) ', \beta=' num2str(beta) ', \gamma='
    num2str(gamma) ', \delta=' num2str(delta)]);

%
figure
plot(y(:,1), y(:,2),'color','k', 'LineWidth',2)
xlabel('Presa');
ylabel('Depredador');
axis square
set(gcf, 'Position', [100 100 300 300]);

end

function dydt = Lotka_Volterra(t,y, alpha, beta, gamma, delta)

dydt =[alpha*y(1)-beta*y(1)*y(2); % linear growth of pray, death
    by the predator
       delta*y(1)*y(2)-gamma*y(2)]; % growth of predator depends
           on prey, linear death

end
```

The results of this simulation are shown in figure 2.8

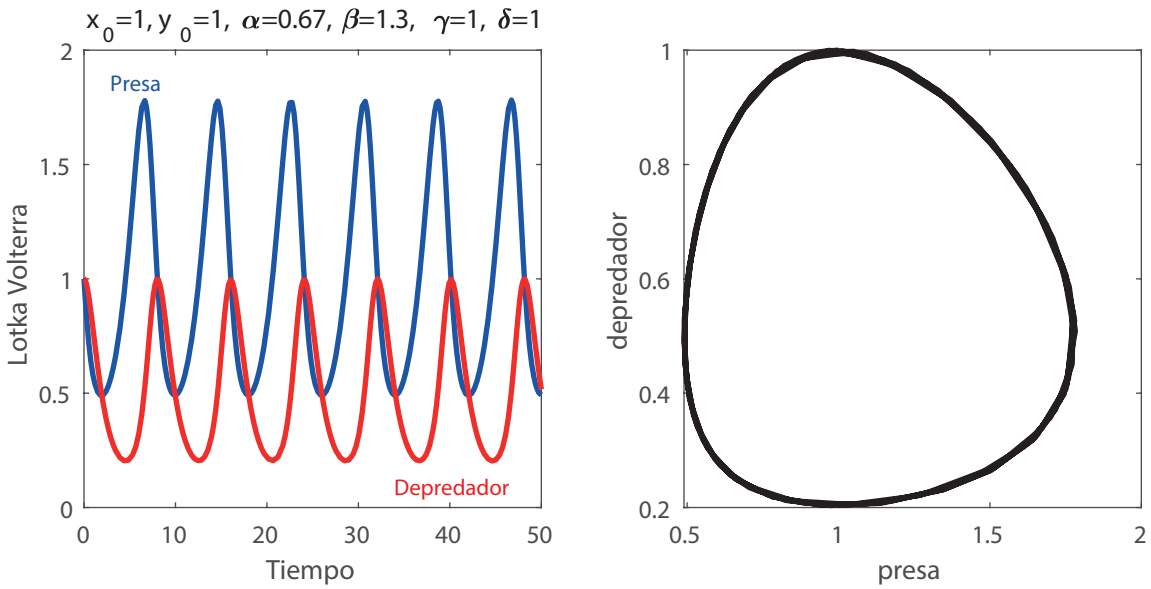


Figure 2.8:

2.0.7 Parameter optimization: Seeking the best agreement between the mathematical model and the experimental data, using minimization algorithms

ODE models are quantitative. This means that there is an explicit dependency on the dynamic and steady state behaviours of the state variables on the choice of parameter values. So, how to choose the best parameter set $\bar{P}_{\text{opt}}$?

Ideally, one should use high-dimensional *quantitative* data to find this optimal parameter set. To do so, the idea is to find the optimal parameter set $\bar{P}_{\text{opt}}$ such that the model $\bar{x}(t) = \bar{H}(t, \bar{P}_{\text{opt}})$ match as closely as possible the experimental data $\bar{x}_{\text{exp}}$. In other words, we are looking for the parameter set $\bar{P}_{\text{opt}}$ that **minimizes** the difference between the experimental data and the

model. i.e. the **cost** given in equation 2.15:

$$cost(\bar{P}) = \sum_{i=1}^{k} (\bar{x}_{\text{exp}}(t_{\text{i}}) - \bar{x}(t_{\text{i}}, \bar{P}))^2 \tag{2.15}$$

To find the solution to this minimization problem:

$$min(cost(\bar{P})) = cost(\bar{P}_{\text{opt}}) \tag{2.16}$$

One can use non-linear optimization algorithms, such as the Nelder-Mead simplex algorithm. These can be implemented by most of the softwares with non-linear ODE analysis capabilities. Below we will give an example of how this algorithm can be implemented in `Matlab`, by using the build-in function `fminsearch`. For this, we will seek the optimal agreement between simulations of our reaction network representing the regulatory interactions controlling the formation of the heterodimer XY (eqs. 2.11) and the experimental data from [44] (Fig. 5B). This data describes the binding of the activated transcription factor Smad1 (corresponding to our variable X; since Smad1 can be activated/produced, degraded, and forms a heterodimer) to the promoter of the PPARγ gene (corresponding to our variable Y, since DNA sequences such as promoters are neither produced nor degraded in this time-scale (hours); their monomeric, free concentrations are only affected by dimerization with regulatory proteins such as Transcription Factors), in response to stimulation with BMP. For our optimization, we will assume that the parameter values $X_{\text{pre}} = 10, k_{\text{dis}} = k_{\text{prod}} = k_{\text{deg}} = 0.5, E_{\text{dim}} = E_{\text{prod}} = E_{\text{dis}} = 1$ are fixed (this could be justified if these parameter values were calculated from other empirical data). We will find the optimal $\bar{P}_{\text{opt}}[k_{\text{dim}}, Y_{\text{tot}}]$ parameter pair that best reproduces the dataset.

Numerical optimization

```
function xOPT=example_optimization_dimerFormation
%% Experimental data
t_exp= [0 1 3 ]; % hours post-stimulation
XY_exp=[.5 2.5 3 ]./.5; %Smad1- promoter complex
%% Optimization
% Give an initial guess for the parameter
xinit= [2 7];
% Run the optimization!
```

```
[x,J,flag]=fminsearch(@(x)CostFunction(x,t_exp, XY_exp),xinit);
xOPT=x;
end
function Cost=CostFunction(x,t_exp, XY_exp)
kdimEdim=x(1);
Ytot=x(2);
%% Solve the ODE - with this parameter
%constant parameters (those not minimized)
Xpre=10; kprodEprod=.5; kdegEdeg=.5; kdisEdis=1;
%initial conditions
X_0=0; Y_0=Ytot-XY_exp(1);
y0=[X_0 Y_0]; %initial condition [X, Y]
%integration interval
tspan = [0 t_exp(end)]; %integration interval
% Call the ODE solver
[t,y] = ode45(@(t,y)dimerFormation(t,y ,Xpre, kprodEprod,
    kdegEdeg, kdimEdim, kdisEdis, Ytot),tspan,y0);
% Focus on the variable to be compared with data
XY_t=Ytot-y(:,2);
% interpolate those values corresponding to the measurements
XY_predicted = interp1(t,XY_t,t_exp);
%% Calculate the cost of the predition vs. the experimental data
Cost=(sum((XY_predicted-XY_exp).^2));
end
```

We can then visualize the results of our optimization by calculating $\bar{P}_{\text{opt}} = [k_{\text{dim}}, Y_{\text{tot}}]$ with our optimization function (`example_optimization_dimerFormation.m` in our example), and then plotting the model $\bar{x}(t, \bar{P}_{\text{opt}}$ together with the experimental data used for the optimization.

```
close all; clear all; clc
%% Run the ODE with the optimal parameters
% constant parameter values
Xpre=10; kprodEprod=.5; kdegEdeg=.5; kdisEdis=1;
% minimized parameter values
xOPT=example_optimization_dimerFormation;
kdimEdim=xOPT(1); Y_tot=xOPT(2);
% experimental data
```

```
t_exp= [0 1 3 ]; XY_exp=[.5 2.5 3 ]./.5;
% Initial conditions
X_0=0; Y_0=Y_tot-XY_exp(1);
y0=[X_0 Y_0];
% integration interval
tspan = [0 t_exp(end)+.5];
% Call the ODE solver
[t,y] = ode45(@(t,y)dimerFormation(t,y ,Xpre, kprodEprod,
    kdegEdeg, kdimEdim, kdisEdis, Y_tot),tspan,y0);

%% calculate the final cost
XY_pred = interp1(t,(Y_tot-y(:,2)),t_exp);
Cost=(sum((XY_pred-XY_exp).^2));

%%Plot the results
figure;
scatter(t_exp, XY_exp, 'k'); hold on
plot(t, Y_tot-y(:,2), 'k'); axis square
ylabel('XY dynamics [fold increase]')
xlabel('time [hours]')
title(['minimal cost:' num2str(Cost) ' kdim=' num2str(xOPT(1)) '
    Ytot=' num2str(xOPT(2))]);
```

These results are shown in figure 2.9.

Let us conclude this small section on parameter optimization by with a word of caution:

- Over-fitting of parameters might occur if the ratio of parameters to be optimized relative to high-quality experimental information is unfavourable. Thus, the more coherent (i.e. from the same experiment, ideally) empirical data we have for the parameter optimization, the better. It is important to aim for a (parametrized) model that **robustly** reproduces the expected behaviours [47, 32]

- The optimal solution $\bar{P}_{\text{opt}}$ might not be unique. In fact, a practical problem when searching for $\bar{P}_{\text{opt}}$ is that the minimization algorithm can be trapped in a local minimum, i.e. where the resulting cost is low but not the (globally) lowest. To avoid these complications, whenever possible, a global optimization algorithm (eg. simulated annealing [1]) should be preferred over local minimization algorithms such as `fminsearch`.

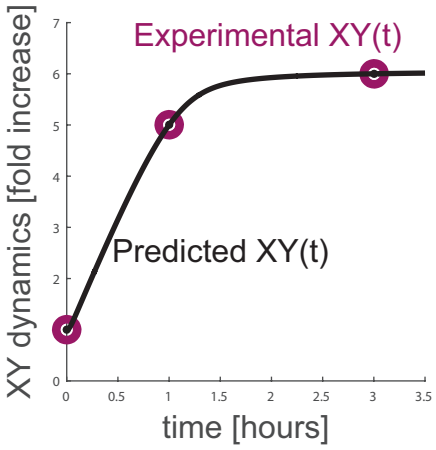


Figure 2.9: Optimal agreement between mathematical model (eqns. 2.11 and experimental data [44]. The optimal parameters are $\bar{P}_{\text{opt}} = [k_{\text{dim}} = 8.5, Y_{\text{tot}} = 6.1]$, which result in the minimal cost (eq. 2.15) 2.5×10^{-11} - That's small indeed!

- Other, simpler (with less variables and parameters), models might be able to better explain or reproduce the experimental data. If in doubt over the regulatory interactions underlying the behaviour of the model variables, it is advisable to start by proposing a set of plausible models (with different kinetic reactions and thus different network topology and structure of the mathematical model), systematically testing how well these models can fit the experimental data, and **selecting the simplest model able to best reproduce the experimental data** - for example, with the Akaike Information Criterion or other statistical techniques [59].

In conclusion, parameter optimization is a powerful tool that can help to find the parameter set with which the proposed model can best describe a given set of experimental data. Often, this technique is used simply as a methodological step, to parameterize the ODE model for further mathematical analysis. However, as more high-throughput and quantitative experimental data becomes available, parameter optimization routines can be used to directly address clinically relevant research problems. For example, parameter optimization has been used to deduce from data the underlying cause of a pathogenic transformation of the liver [55, 56, 63, 27], stratify patient cohorts for differential treatments [20, 21, 25].

2.0.8 Model analysis: Assessing the robustness / plasticity of the model behaviour in response to perturbations

Once we have an experimentally calibrated, mechanistic model, we can start with the analysis. In most cases, the underlying question is: How do structural (i.e., in the model equations) or quantitative (i.e., in model parameters) *perturbations* affect the behaviour of the model?

To answer these questions, we first have to ask: which *feature* of our model are we interested in analyse, in terms of its robustness? For example, is it the steady state value? is it the existence of multiple equilibrium points? The existence of oscillations? The frequency or amplitude of the transient response? The time - to-relaxation? It is important to be specific about these features, since in order to assess their robustness we must be able to bring them to formal terms.

Once identified the feature, we can proceed to its robustness analysis. Since both structural and quantitative features can be modulated by parameteric variations (note that the presence or absence of a specific equation, i.e. structural changes, can be represented by setting the corresponding kinetic parameter to a value ¿0 or =0).

Usually, this analysis is done by:

- **Robustness analysis**: Randomly varying all the model parameters around the nominal (optimized) value, and assessing the response of the system to these perturbations, for example by computing the proportion of parametric model variants displaying the desired behaviour. See for example the robustness analysis of a host-commensal bacteria interaction network reported in [11]).

- **Sensitivity analysis**: Systematically varying all the model parameter combinations, and evaluating which parametric changes are responsible for the largest deviation in the model behaviours [5, 34].

- **Bifurcation analysis**: Follow the model's behaviour in response to changes in a subset of parameters -the bifurcation parameters. Generally, this type of analysis is performed when a sharp, qualitative transition in response to a small, quantitative change in a bifurcation parameter is expected, for example, when looking for a hysteretic switch as depicted in figure 2.5 (see for example [13, 53]

2.1 Multi-scale modelling: Understanding the interplay between regulatory networks and the (micro)-environment

In this section, we explore mathematical tools to analyse biological systems with multiple time scales. Specifically, we consider the interplay between biological processes occurring at two time scales: Fast biochemical processes that regulate the phenotypic decision-making of cells in response to micro-environmental conditions, and the slow, tissue level processes regulating the dynamics of the micro-environment (the "bifurcation parameter" in section 2.0.5). As discussed previously, the clinical relevance of considering such multi-scale systems comes from the fact that the characteristic gradual aggravation of the chronic degenerative diseases emerges from aberrations in the phenotype-micro-environment interactions (figs. **??** and **??**).

In previous sections, we saw that phenotypes can be mathematically represented as attractors of the underlying regulatory networks, and that transitions between these attractors can be driven not only by stochastic fluctuations (section **??**, but also by changes in the micro-environmental conditions (subsection 2.0.5). In this section, we want to pose the following question: what if these environmental fluctuations are actually changing *as a consequence* of the phenotype changes driven by the individual cells in the tissue (fig. **??**? How to account for tissue-level risk factors, which might propagate across this multi-scale regulatory network, giving rise to the gradual phenotypic deterioration (fig. **??**)?

To model these kind of systems, we will simultaneously consider the changes in the activation state of biochemical reaction networks controlling phenotypic decisions, a the tissue-level processes underlying micro-environmental fluctuations. While the biochemical reactions are fast, in the time-scale of minutes to hours, the dynamics of the surrounding tissue-level conditions stabilize within days to weeks. To account for these two different time-scales, we will perform a *time scale separation*, also known as Quasi-Steady-State Assumption (QSSA): The relation between the micro-environmental factor and the phenotype is described algebraically, by the mapping of the bifurcation parameter S to the stationary solution $\hat{X}_{\text{ss}}(S)$ of eq. $\dot{\hat{X}}(t, S) = 0$. The bifurcation parameter S, in turn, is dynamically described by $\dot{\hat{S}} = F(\tau, \hat{X}_{\text{ss}}(S))$, with t and τ the time-scales of the fast and the slow system, respectively. Note that the governing function $F(\tau, \hat{X}_{\text{ss}}(S))$ for the dynamics of S explicitly considers the algebraic variable $\hat{X}_{\text{ss}}(S)$. In other words, in such a model the changes in the bifurcation

parameter depend on the proportion of phenotypes within the tissue.

Assuming such differences in time-scales in fact greatly simplifies the analysis of such multi-dimensional systems, described by the coupling between $\dot{\hat{X}}$ and $\dot{\hat{S}}$, with $\hat{X}$ and $\hat{S}$ n and m dimensional vectors, respectively. To illustrate this, let's consider the typical example of a biochemical network described by a system of ODEs and simplified by the QSSA: The Briggs-Haldane version of the Michaelis-Menten equations [60, 4]).

The system of equations:

$$\frac{d[E]}{dt} = -k_{\mathrm{f}}[E][S] + k_{\mathrm{r}}[ES] + k_{\mathrm{cat}}[ES], \tag{2.17a}$$

$$\frac{d[S]}{dt} = -k_{\mathrm{f}}[E][S] + k_{\mathrm{r}}[ES], \tag{2.17b}$$

$$\frac{d[ES]}{dt} = k_{\mathrm{f}}[E][S] - k_{\mathrm{r}}[ES] - k_{\mathrm{cat}}[ES], \tag{2.17c}$$

$$\frac{d[P]}{dt} = k_{\mathrm{cat}}[ES], \tag{2.17d}$$

represents the dynamic interactions between the catalysing enzyme E, the substrate S, the enzyme-substrate complex ES, and the product of the enzymatic reaction, P, represented in the reaction network in figure 2.10. In these equations, it is considered that the total amount of enzymes is conserved (i.e., no *de novo* production of E), which can be seen directly from the conservation equations:

$$\frac{d[E]}{dt} + \frac{d[ES]}{dt} = 0,$$

which imply

$$[E] + [ES] = [E]_0 \tag{2.18}$$

The key assumption to simplify equations 2.17 is that the enzyme-substrate formation $[ES]$ is infinitely fast respect to the rest of the dynamics i.e. $\frac{d[ES]}{dt} = 0$. From this QSSA, it follows that

$$k_{\mathrm{f}}[E][S] = [ES](k_{\mathrm{r}} + k_{\mathrm{cat}}).$$

Using the conservation equation 2.18, $k_{\mathrm{f}}[E][S]$ can be rewritten as:

$$k_{\mathrm{f}}[E]_0[S] - k_{\mathrm{f}}[ES][S] = [ES](k_{\mathrm{r}} + k_{\mathrm{cat}}),$$

from which one can isolate the variable $[ES]$ as

$$[ES] = \frac{k_{\mathrm{f}}[E]_0[S]}{((k_{\mathrm{r}} + k_{\mathrm{cat}}) + k_{\mathrm{f}}[S])},$$

which can be used to rewrite $\frac{d[P]}{dt}$ as

$$\frac{d[P]}{dt} = k_{\mathrm{cat}} \frac{k_{\mathrm{f}}[E]_0[S]}{((k_{\mathrm{r}} + k_{\mathrm{cat}}) + k_{\mathrm{f}}[S])}.$$

Defining

$$K_{\mathrm{M}} = \frac{k_{\mathrm{r}} + k_{\mathrm{cat}}}{k_{\mathrm{f}}},$$

one can recognize the simple, one dimensional system representing the dynamics of product formation, known as Michaelis -Menten equation:

$$\frac{d[P]}{dt} = k_{\mathrm{cat}} \frac{[E]_0[S]}{K_{\mathrm{M+}}[S]}$$

.

So, using QSSA we were able to reduce a 4-dimensional dynamical system to a one dimensional ODE!

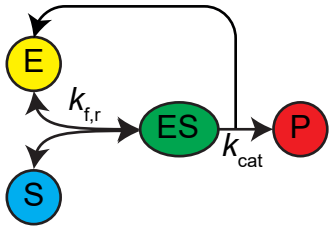


Figure 2.10: Reaction network of the dynamic interactions between enzyme (E), substrate (S), enzyme-substrate complex (ES) and product of the enzymatic reaction (P) represented in equations 2.17.

Back to our original problem of coupling phenotypic decisions to micro-environmental changes. Let's consider the simplest multi-stable system in which gradual environmental conditions drive abrupt phenotype changes, namely a bistable system (fig. 2.5). As discussed in section 2.0.5, mapping the relation between the bifurcation parameter and the stable steady state solutions can be tricky, since analytical steady solutions for high-order non-linear systems rarely exist (since they are roots of high order polynomials), and numerical methods require exhaustive explorations of the parameter space (including initial conditions) and are often stuck in local solutions. Thus, iteratively solving such

multi-scale problems during the numerical integration of slow variables can be computationally very intensive, and might often even fail to find the desired steady state solutions. To overcome this difficulty, it is possible to phenomenologically (as opposed to mechanistic, corresponding to algebraic relations of steady state solutions of ODEs as functions of bifurcation diagrams) describe the previously characterized bistable switch by by a piecewise-affine (PWA) functions [6]. Such PWA approximation provides a rule that maps the input (stimulus) to the output (effector) (figure 2.5). For example, assuming a perfect switch, the effector can be approximated by two constant values, E_{low} and E_{high}, representing the "low" or "high" branches of the bifurcation diagram, respectively. Now, let's consider that our bifurcation parameter, this is, the input, changes dynamically in the time-scale τ. Then the relation describing how the output $E(\tau)$ is determined by the input $S(\tau)$ and by the previous output values $E(x < \tau)$ can be approximated as follows:

- If $S(\tau) < S^-$, then $E(\tau) = E_{low}$ (effector is low if the stimulus concentration is low).
- If $S(\tau) > S^-$, then $E(\tau) = E_{high}$ (effector is high if the stimulus concentration is high).
- If $S(\tau) \in [S^-, S^+]$, then:
  - if $E(x < \tau) = E_{low}$, then $E(\tau) = E_{low}$, or
  - if $E(x < \tau) = E_{high}$, then $E(\tau) = E_{high}$,

  corresponding to the history-dependent determination of the effector value when the stimulus is in the bistable region.

More formally, these conditions can be represented by the PWA given in equation 2.19 (adapted from [40]):

$$E(\tau) = \begin{cases} E_{low} & \text{if } (S(\tau) < S^-) \text{ or } \{S(\tau) \in [S^-, S^+] \text{ and } E(x < \tau) = E_{\text{low}}\} \\ E_{high} & \text{if } (S(\tau) > S^+) \text{ or } \{S(\tau) \in [S^-, S^+] \text{ and } E(x < \tau) = E_{\text{high}}\} \end{cases}. \tag{2.19}$$

Note that equation 2.19 implicitly assumes two time-scales:

- A **fast** time-scale t that governs the stabilized biochemical interactions that underlie the bistable dose-response behaviour. These biochemical reactions can be represented by a system of ODEs $\dot{\mathbf{E}}(t, S, \mathbf{E})$ that operates at time-scale t and has a input S that does not change significantly

($S(t) \approx constant$) while $E(t)$ reaches its equilibrium value (given by E_{low} or E_{high}, respectively).

- A **slow** time-scale τ that determines the dynamics of the input $S(\tau)$ by the governing equation $\dot{S}(\tau) = F(\tau, S)$.

A special case of the system 2.19, which is of particular interest here, when considering the complex **interplays** between phenotype decisions (described by the different states of the bifurcation diagram) and microenvironmental conditions, occurs when the slowly changing input $S(\tau)$ is itself determined by its quickly stabilizing output $E(t)$ (and vice-versa). In such a case, also the dynamics of $S(\tau)$ (that depend on $E(\tau)$) can be descried by the PWA given in equation 2.20 (adapted from [40]):

$$\dot{S}(\tau) = \begin{cases} F_{low}(S) & \text{if } E(\tau) = E_{low} \\ F_{high}(S) & \text{if } E(\tau) = E_{high}, \end{cases} \tag{2.20}$$

where F_{low} and F_{high} are the two *governing equations* that determine the dynamics of S when $E(\tau) = E_{low}$ or $E(\tau) = E_{high}$, respectively.
Accordingly, the long term behaviour of S is given by the *focal points* S_{ss}^{low} and S_{ss}^{high}, corresponding to the steady state values given by the solution to $F_{low} = 0$ and $F_{high} = 0$, respectively [40].
The coupling between equations 2.19 and 2.20 represents a hybrid system that has been extensively discussed and analysed in [6, 40]. The long term behaviour of the coupled variable $S(\tau)$ and $E(t)$ is determined by the relative position of the focal points S_{ss}^{low} and S_{ss}^{high} respect to the threshold values S^- and S^+, as follows (figure 2.11):

- A resting, homeostatic ("low") steady state occurs when $S_{\text{ss}}^{low} \leq S^+$ and $S_{\text{ss}}^{high} < S^-$.
- A chronically inflamed steady state occurs when $S_{\text{ss}}^{low} > S^+$ and $S_{\text{ss}}^{high} \geq S^-$.
- Bistability in the two-time-scale dynamical system occurs when $S_{\text{ss}}^{low} \leq S^+$ but $S_{\text{ss}}^{high} \geq S^-$.
- Oscillations occur when $S_{\text{ss}}^{low} > S^+$ and $S_{\text{ss}}^{high} < S^-$.

In conclusion, this methodology allows the derivation of analytical conditions required for different qualitative behaviours of a complex dynamical

system that operates in two time-scales, reducing the need for numerical methods. Note however that the agreement between the dynamical behaviour that is analytically derived from the hybrid system representation and the numerical simulations of the model must be verified for the particular mathematical model that is analysed using this approach, to ensure that neither the discontinuities of the hybrid representation of the system, nor the transient behaviour that is not captured by the focal point analysis detailed above, affect the dynamics of the unsimplified mathematical model.

The model described in section **??** provides an example in which this focal point analysis is used to systematically determine the effects of risk factors affecting tissue level processes on the development of early phases of AD. Such framework can be applied not only to micro-environment - phenotype interactions discussed here, but in general to model (biological) systems in which there is a co-existence and inter-dependence of processes operating at different time-scales. Examples include multi-scale networks considering the interplay between:

- Metabolism and signalling [55].
- Metabolism and gene expression [40].
- Cellular-level population dynamics and biochemical processes [39, 24, 50, 51].

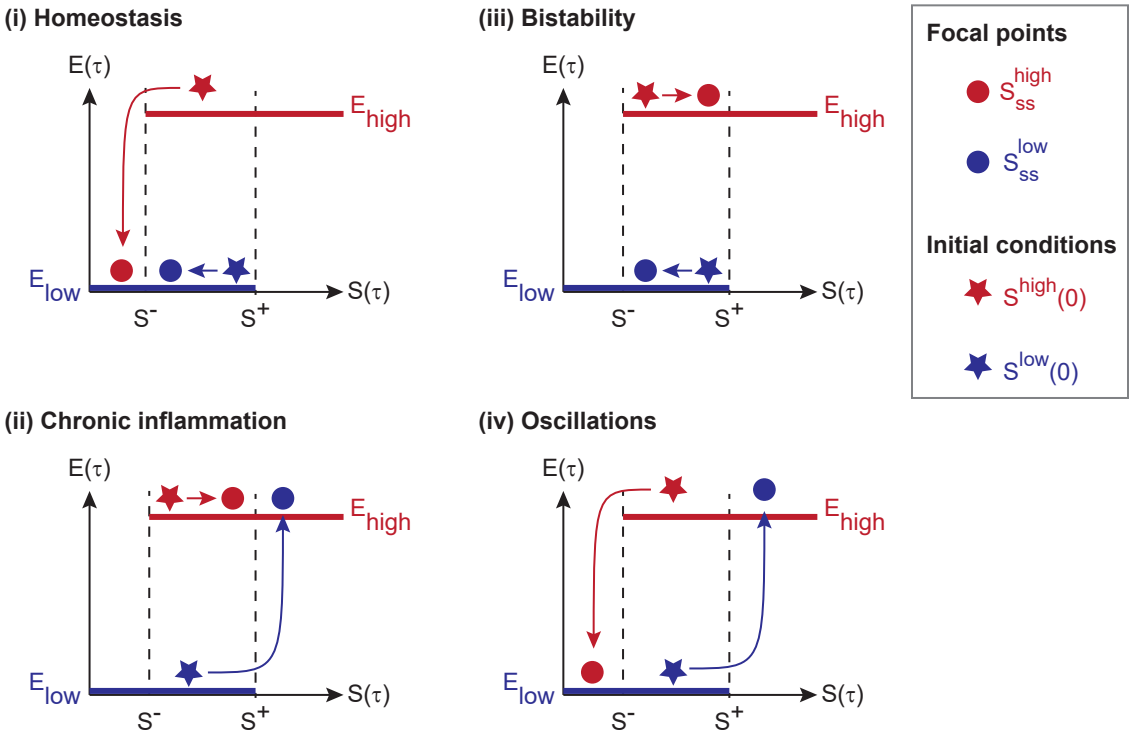


Figure 2.11: **Schematic representation of the qualitative dynamic behaviours of the hybrid system described in the coupled equations 2.19 and 2.20.** The long term dynamical behaviour of the hybrid system 2.19 and 2.20 is determined by the position of the focal points $S_{\rm ss}^{low}$ and $S_{\rm ss}^{high}$ respect to the threshold values S^- and S^+. (i) $S_{\rm ss}^{low} \leq S^+$ and $S_{\rm ss}^{high} < S^-$ lead to homeostasis, (ii) chronic inflammation occurs when $S_{\rm ss}^{low} > S^+$ and $S_{\rm ss}^{high} \geq S^-$, (iii) Bistability arises from $S_{\rm ss}^{low} \leq S^+$ but $S_{\rm ss}^{high} \geq S^-$, and (iv) Oscillations result from $S_{\rm ss}^{low} > S^+$ and $S_{\rm ss}^{high} < S^-$. Figure taken from [10] (URI: `http://hdl.handle.net/10044/1/47969` (published under a Creative Commons Attribution Non-Commercial No Derivatives Licence `https://creativecommons.org/licenses/by-nc-nd/3.0/`

Práctica 2: Sistemas continuos multi-estables

Les recomiendo hacer esta práctica en casa. Probablemente no nos dará tiempo de hacerla en el taller. Pero vienen las respuestas al final de la sección.

En esta práctica, regresaremos al tema de la multi-estabilidad en sistemas biológicos, pero, esta vez, desde el punto de vista de modelos matemáticos continuos (específicamente, con ecuaciones diferenciales). Para ello, revisaremos a detalle el modelo matemático propuesto en [3]:

$$\dot{x}_1 = \alpha_1(1 - x_1) - \frac{\beta_1 x_1 (\nu y_1)^{\gamma_1}}{K_1 + (\nu y_1)^{\gamma_1}} \tag{3.1a}$$

$$\dot{y}_1 = \alpha_2(1 - y_1) - \frac{\beta_2 y_1 x_1^{\gamma_2}}{K_2 + x_1^{\gamma_2}} \tag{3.1b}$$

3.0.1 De la ecuación a la gráfica

Considera el Sistema de ecuaciones diferenciales que se analiza en el artículo:

$$\dot{x}_1 = \alpha_1 x_2 - \frac{\beta_1 x_1 (\nu y_1)^{\gamma_1}}{K_1 + (\nu y_1)^{\gamma_1}}, \tag{3.2a}$$

$$\dot{x}_2 = -\alpha_1 x_2 + \frac{\beta_1 x_1 (\nu y_1)^{\gamma_1}}{K_1 + (\nu y_1)^{\gamma_1}}, \tag{3.2b}$$

$$\dot{y}_1 = \alpha_2 y_2 - \frac{\beta_2 y_1 x_1^{\gamma_2}}{K_2 + x_1^{\gamma_2}} \tag{3.2c}$$

$$\dot{y}_2 = -\alpha_2 y_2 + \frac{\beta_2 y_1 x_1^{\gamma_2}}{K_2 + x_1^{\gamma_2}} \tag{3.2d}$$

1. ¿De cuántas dimensiones es este sistema?

2. ¿Cuántas reacciones hay, qué representan, y cómo afectan éstas la razón de cambio de cada una de las variables? Hint: La reacción $\frac{\beta_1 x_1 (\nu y_1)^{\gamma_1}}{K_1 + (\nu y_1)^{\gamma_1}}$ representa una retroalimentación negativa de y_1 sobre x_1: y_1 está modulando la conversión de x_1 de regreso a x_2. Este efecto de y_1 sobre x_1 es saturante (i.e., llega a un máximo), lo que se representa como $\frac{(\nu y_1)^{\gamma_1}}{K_1 + (\nu y_1)^{\gamma_1}}$. La constante ν representa la intensidad o fuerza de esta retroalimentación. El exponente γ_1 representa cooperatividad en el sistema; si γ_1 es mayor a 1, podemos pensar que se requiere más de una molécula de y_1 para que se lleve a cabo la reacción.

3. ¿qué puedes decir de $\dot{x}_1 + \dot{x}_2$, así como de $\dot{y}_1 + \dot{y}_2$? ¿qué interpretación biológica tiene esto (en términos de producción de novo / degradación)?.

4. Realiza una gráfica que represente la red regulatoria representada por este sistema de ecuaciones diferenciales. ¿a qué estructura de control corresponde?

3.0.2 Ecuaciones de conservación para reducir el sistema

En la pregunta anterior, vimos que el sistema 3.2 está cerrado; es decir, que no hay producción de novo ni degradación de x_i ni de y_i ($i = 1, 2$); sólo activación y de-activación. Esto significa que $x_1 + x_2 =$ constante 1, y que $y_1 + y_2 =$constante 2.

1. Formalmente, es decir, en el modelo matemático, ¿cómo podemos ver esto? (Hint: fíjate la suma de las derivadas).

2. Las ecuaciones $x_1 + x_2 =$ constante 1, y $y_1 + y_2 =$constante 2 se llaman ecuaciones de conservación, e implican que podemos expresar x_2 y y_2 en términos de x_1 y de y_1, respectivamente, y con ello, nos "ahorramos" dos ecuaciones diferenciales. Explica cómo se llega entonces del sistema de 4 dimensiones (eq. 3.2 al sistema 2D (eq. 3.1) con el que se trabaja en el resto del artículo.

Ahora sí, prendan su computadora y entremos en materia.

3.0.3 Dinámica del sistema

Usando dos condiciones iniciales ($[x1(0), y1(0)] = [0, 0]$ y $[0, 9]$:

1. Genera tres diagramas de tiempo vs- variable y_1, para tres valores del parámetro de bifurcación $\nu = 0.75, 1$ y 1.9.

2. Grafica esta misma información en un diagrama-fase; i.e. $x_1(t)$ vs $y_1(t)$. (opcional: añadir a este diagrama el campo vectorial, dado por $[\dot{x}_1(t), \dot{y}_1(t)]$.

3.0.4 Cuencas de atracción

Regresemos al valor del parámetro de bifurcación $\nu = 1$. Probando varias condiciones iniciales, gráfica, en una misma figura, las trayectorias $x_1(t)$ vs $y_1(t)$. ¿te puedes dar una idea del tamaño de las cuencas de atracción de cada uno de los atractores? ¿qué sucede si aumentas el valor de ν a 1.6?

3.0.5 Señales de alerta temprana (1)

Los sistemas bifurcantes presentan un comportamiento característico, llamado "ralentización crítica", cuando se acercan al punto de bifurcación. ¿podemos ver algo así en este sistema? Para comprobarlo, gráfica, en la misma figura, t vs $y_1(t)$ para $\nu = 0.2 : 0.1 : 1$, usando la condición inicial $[0, 0.9]$. ¿qué observas?

(Nota: Retomaremos esto en la siguiente sesión).

3.0.6 Biestabilidad e histéresis

Concentrémonos ahora en los puntos de equilibrio ("atractores") del sistema. (ver sección 2.0.5. En preguntas pasadas vimos que, dependiendo del valor de ν (nuestro parámetro de bifurcación) el sistema de ecuaciones 3.1 puede converger a valores de y_{ss} bajos (cercanos a cero), altos (cercanos a 1) **o ambos**. Ahora, vamos a inspeccionar con más cuidado cómo la estabilidad del sistema de ecuaciones 3.1 depende del parámetro de bifurcación ν. Específicamente, vamos a construir un **diagrama de bifurcaciones** de ν vs. y_{ss}. Para ello, usaremos `grind.r`, una implementación de un algoritmo de continuación numérica en `r` (pueden consultar detalles en: `http://theory.bio.uu.nl/rdb/grind.html`).

Diagrama de bifurcación en R (con GrindR)

```
source('Grind.r') # descargado de
    {http://theory.bio.uu.nl/rdb/grind.html}

# definir la funcion
model <- function(t, state, parms){
  with(as.list(c(state,parms)), {
    dx = alpha1*(1-x)-beta1*x*(v*y)^gamma1/(K1+(v*y)^gamma1)
    dy = alpha2*(1-y)-beta2*y*x^gamma2/(K2+x^gamma2)
    return(list(c(dx, dy)))
  })
}

# Declarar los valores de parametros que permanecen constantes
p <-  c(alpha1=1, alpha2=1, beta1=200, beta2=10, gamma1=4,
    gamma2=4, K1=30, K2=1, v=1)

# "Initial guess" para buscar raices con el algoritmo de Newton
    Raphson
s <- c(x=0,y=0)

# Grafica las ceroclinas para este valor de v
par(pty="s") #ejes cuadrados
plane(xmin=0, xmax=1, ymin=0, ymax=1)

# ahora obtengamos estos tres puntos de equilibrio
mid <- newton(s,plot=T)
low <- newton(c(x=1,y=0),plot=T)
hig <- newton(c(x=0,y=1),plot=T)

par(pty="s")
continue(state=hig, parms=p, odes=model, x="v", step=0.001,
    xmin=0, xmax=2,y="y", ymin=0, ymax=1.1) # log="", time=0,
    positive=TRUE, add=TRUE)
continue(state=low, parms=p, odes=model, x="v", step=0.001,
    xmin=0, xmax=2,y="y", ymin=0, ymax=1.1, log="", time=0,
    positive=TRUE, add=TRUE)
```

```
30 continue(state=mid, parms=p, odes=model, x="v", step=0.001,
    xmin=0, xmax=2,y="y", ymin=0, ymax=1.1, log="", time=0,
    positive=TRUE, add=TRUE)
```

Corriendo el código 3.0.6 obtenemos el diagrama de bifurcación representado en figura 3.1:

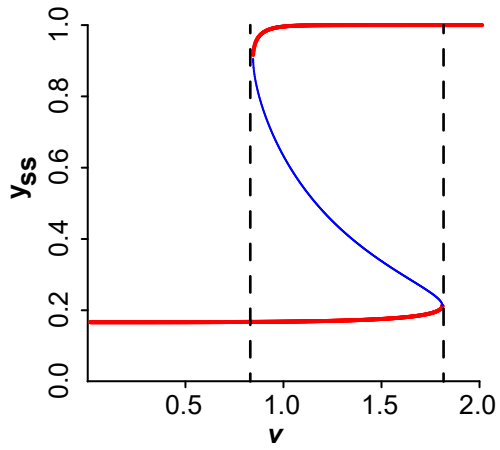


Figure 3.1: Diagrama de bifurcación del sistema de ecuaciones 3.1

3.0.7 ¿qué sucede cuando ν es mayor a 0.83 y menor a 1.8?

Decimos que sistemas que presentan dos puntos de equilibrio estables, como éste, tienen un tipo de memoria llamada histéresis, pues cuando el input o parámetro de bifurcación (ν en este caso) está entre los dos valores umbral (0.8 y 1.8 en este caso), el output del sistema (punto de equilibrio estable) puede ser uno (bajo) o el otro (alto); esto depende de los valores anteriores del parámetro de bifurcación: Si los valores anteriores son bajos, output es bajo, y viceversa.

Este tipo de comportamientos son sumamente relevantes en biología, pues subyacen decisiones fenotípicas abruptas en respuesta a estímulos (ambientales) continuos. Ejemplos de sistemas bi-estables incluyen la regulación de la apoptosis [26, 15], la entrada a ciclo celular [68], y la respuesta inmune ante estímulos ambientales [54, 28, 49], entre otros.

Quizás se preguntarán cómo se puede ver experimentalmente que un sistema es bi-estable. Pues, básicamente, esperaríamos encontrar distribuciones bimodales [57, 29, 17, 45] ... aunque no todas las distribuciones bimodales implican bi-estabilidad [58].

> En el próximo capítulo veremos cómo podemos analizar qué tipo de circuitería da lugar a qué tipo de distribuciones....

Bueno, espero haberlos convencido de que sistemas biológicos bi-estables son frecuentes e importantes. Entonces, una pregunta relevante es: ¿qué tipo de circuitería/red genera sistemas bi-estables? Mucha gente se ha hecho esta pregunta, y se ha abordado desde varias perspectivas, incluyendo métodos de fuerza bruta [42], métodos algebráicos (mucho más elegantes, pero complicados y con resultados no totalmente generalizables) [8, 9].

Lo que este tipo de estudios han encontrado es que, en general, para generar bi-estabilidad se requiere retroalimentación positiva y cooperatividad.

Última pregunta:

> ¿pueden encontrar estos dos elementos (retroalimentación positiva y cooperatividad) en el sistema que acaban de analizar?

3.1 Respuestas

3.1.1 Parte que no involucra código

1. El sistema tiene 4 dimensiones.
2. Hay cuatro reacciones:
   - R1: $\alpha_1 x_2$: conversión de x_2 a x_1, afecta positivamente a x_1 y negativamente a x_2.
   - R2: Conversión de x_1 a x_2, mediada por y_1, cooperativamente y con efecto saturado. Afecta positivamente a x_2 y negativamente a x_1.
   - R3: $\alpha_3 y_2$, conversión de y_2 a y_1, afecta positivamente a y_1 y negativamente a y_2.
   - R4: Conversión de y_1 a y_2, mediada por x_1, cooperativamente y con efecto saturado. Afecta positivamente a y_2 y negativamente a y_1.

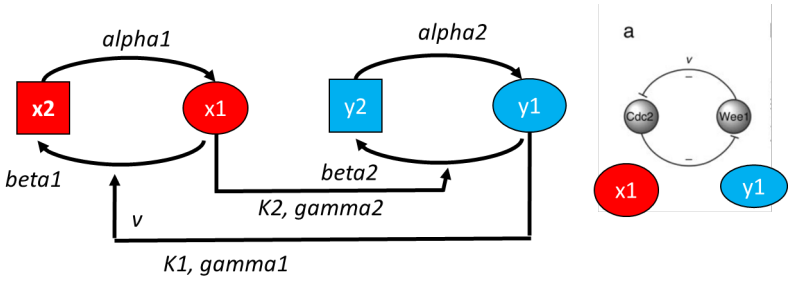


Figure 3.2: Red de regulación representada en 3.2

3. Ecuaciones de conservación para reducir el sistema

   - $\dot{x}_1 + \dot{x}_2 = 0$ y $\dot{y}_1 + \dot{y}_2 = 0$.
   - Asumimos que $x_2 = (1 - x_1)$ y $y_2 = (1 - y_1)$ (es decir, $x_{total} = y_{total} = 1$.

4. Biestabilidad e histéresis:

   - Hay dos puntos de equilibrio estables y uno inestable.
   - sí: y_1 inhibe a x_1 que a su vez inhibe a y_1, formado, de principio a fin, un asa de retroalimentación positiva (dos inhibiciones seguidas dan una activación, menos por menos es más, ver figura 3.2). La cooperatividad se da por el exponente gamma mayor a 1.

3.1.2 Parte que involucra código

Práctica 2 en R

El código para responder todas las preguntas en R:

```
### Prica 2: Ansis de un sistma de ecuaciones diferenciales
    acopladas y no lineales:

##. Angeli, D., Ferrell, J. E. & Sontag, E. D. Detection of
    multistability, bifurcations, and hysteresis in a large class
    of biological positive-feedback systems. PNAS 101, 1822-7
    (2004).
```

```
4
5  # Borramos todo
6  rm(list=ls())
7
8  # Nos ubicamos donde queremos
9  setwd("C:/Users/Elisa/Dropbox/Taller de Introducci a la Biolog
       de Sistemas - CCM UNAM Nov 2018/Pricas/Prica 2 - Sistemas de
       ecuaciones diferenciales ordinarias (EDH)")
10
11
12 # Instalar la paqueter que necesitamos
13 library(deSolve)
14 library(phaseR)
15
16 # Declarar los valores de partros que permanecen constantes
17 alpha1=1; alpha2=1; beta1=200; beta2=10; gamma1=4; gamma2=4;
18 K1=30; K2=1;
19 #v=1;
20
21 Angeli2004 <- function(t, y, parms){
22     #         1      2       3      4       5     6      7   8   9
23   #parms=(alpha1, alpha2, beta1, beta2, gamma1, gamma2, K1, K2, v)
24         dX <-
               parms[1]*(1-y[1])-parms[3]*y[1]*(parms[9]*y[2])^parms[5]/(parms[7]+(parms[9
25         dY <-
               parms[2]*(1-y[2])-parms[4]*y[2]*y[1]^parms[6]/(parms[8]+y[1]^parms[6]);
26         list(c(dX,dY))
27 }
28
29 # Condiciones iniciales
30 ini_1 <- c(0,0); ini_2 <- c(0,0.9)
31
32 # Tiempo de integraci
33 tspan <- seq(from = 0, to = 10, by = 0.01)
34
35 ############### PREGUNTA 1: DINMICA DEL SISTEMA
       ######################
36
37 #  partro  de bifurcaci -
38 for (v in c(0.75, 1, 1.9)){
```

```
39
40   parms=c(alpha1, alpha2, beta1, beta2, gamma1, gamma2, K1, K2, v)
41
42  # A integrar!
43  out1 <- ode(y = ini_1, times = tspan, func = Angeli2004, parms =
        parms)
44
45  plot(out1[,1], out1[,2],type = "l", ylim=c(0,1),col="red", xlab
        = "Time", ylab = "X(t)", main = paste("v=", toString(v),
        sep=" "))
46
47  #Ahora con la segunda condici inicial
48  out2 <- ode(y = ini_2, times = tspan, func = Angeli2004, parms =
        parms)
49  lines(out2[,1], out2[,2],type = "l", col="blue")
50  legend=c(paste("I.C.=", toString(ini_1), sep=" "),
        paste("I.C.=", toString(ini_2), sep=" "))
51
52  #   Amos   a este diagrama de espacio fase un campo vectorial
53  Angeli2004.flowField <- flowField(Angeli2004, xlim = c(0, 1),
        ylim = c(0, 1), parameters = parms, points = 10, add = FALSE)
54  Angeli2004.trajectory <- trajectory(Angeli2004, y0 = ini_1, tlim
        = c(0,10), parameters = parms, col = "blue")
55  Angeli2004.trajectory <- trajectory(Angeli2004, y0 = ini_2, tlim
        = c(0,10), parameters = parms, col = "red")
56  }
57
58  ############ PREGUNTA 2: Cuencas de atracci ###############
59
60    #  partro  de bifurcaci -
61    for (v in c(1, 1.6)){
62      parms=c(alpha1, alpha2, beta1, beta2, gamma1, gamma2, K1, K2,
          v)
63
64      #   Amos   a este diagrama de espacio fase un campo vectorial
65      flowField(Angeli2004, xlim = c(0, 1), ylim = c(0, 1),
          parameters = parms, points = 10, add = FALSE)
66
67      # Genera n condiciones iniciales al azar, pero sobre el men
          ([x=0,1; y=rand] y vice versa)
```

```
    for (ii in seq(1,20,1) ){
      # generate 3 random numbers
      r1=runif(1); r2=runif(1); r3=runif(1);

      if (r1<0.5){
      ini = c(as.numeric(r2<0.5), r3)
      } else {
      ini = c(r3, as.numeric(r2<0.5))
      }

    trajectory(Angeli2004, y0 = ini, tlim = c(0,10), parameters =
        parms, col = "blue")
    }
  }



############ PREGUNTA 3: Ses de alerta temprana ###############

LineWidth=1
#  partro  de bifurcaci -
for (v in seq(0.2,1,0.1)){

  parms=c(alpha1, alpha2, beta1, beta2, gamma1, gamma2, K1, K2, v)

  # A integrar!
  out <- ode(y = ini_2, times = tspan, func = Angeli2004, parms =
      parms)

  if (v==0.2){
  plot(out[,1], out[,3],type = "l", ylim=c(0,1),col="black", xlab
      = "Time", ylab = "X(t)", lwd=LineWidth, main="Alentamiento
      crco" )
  } else {
  lines(out[,1], out[,3],type = "l", col="black", lwd=LineWidth)
  }

  LineWidth=LineWidth+0.5

}
```

```
104
105
106 ############### PREGUNTA 4: Diagrama de bifurcaci
        ######################
107
108 # Corre el  co  anexo:
109 source('Grind.r') # puedes hacerlo abriolo y corriolo, o ir a la
        carpeta en la que est con setwd(..) y luego
110
111
112 # Declarar los valores de partros que permanecen constantes
113 alpha1=1; alpha2=1; beta1=200; beta2=10; gamma1=4; gamma2=4;
114 K1=30; K2=1;
115 #v=1;
116
117 model <- function(t, state, parms){
118   with(as.list(c(state,parms)), {
119     dx = alpha1*(1-x)-beta1*x*(v*y)^gamma1/(K1+(v*y)^gamma1)
120     dy = alpha2*(1-y)-beta2*y*x^gamma2/(K2+x^gamma2)
121     return(list(c(dx, dy)))
122   })
123 }
124
125
126 p <-  c(alpha1=1, alpha2=1, beta1=200, beta2=10, gamma1=4,
        gamma2=4, K1=30, K2=1, v=1)
127 s <- c(x=0,y=0)
128 plane(xmax=4)
129 mid <- newton(s,plot=T)
130 low <- newton(c(x=1,y=0),plot=T)
131 hig <- newton(c(x=0,y=1),plot=T)
132
133 continue(state=hig, parms=p, odes=model, x="v", step=0.001,
        xmin=0, xmax=2,y="y", ymin=0, ymax=1.1) # log="", time=0,
        positive=TRUE, add=TRUE)
134 continue(state=low, parms=p, odes=model, x="v", step=0.001,
        xmin=0, xmax=2,y="y", ymin=0, ymax=1.1, log="", time=0,
        positive=TRUE, add=TRUE)
135 continue(state=mid, parms=p, odes=model, x="v", step=0.001,
        xmin=0, xmax=2,y="y", ymin=0, ymax=1.1, log="", time=0,
```

```
    positive=TRUE, add=TRUE)
```

Practica_2_ODEs_Angeli.R

Práctica 2 en Matlab

Discuplen pero las figuras salen más bonitas en matlab. Acá les va el código:

```
function Practica_Angeli

close all
clear all
clc

%%%%% Ejemplo de un sistema bi-estable%%%%%%%%
%  Angeli D, Ferrell JE, Sontag ED. Detection of multistability,
    bifurcations, and hysteresis in a large class of biological
    positive-feedback systems. PNAS [Internet] 2004;101:18227.
    Available from:
    http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=357011&tool=pmcentrez&ren

%% Pregunta A) Usando dos condiciones iniciales para cada
    condici (y0_a=[x1(0), y1(0)]=[0,0] y y0_b=[0,9]):
% i) Generar tres diagramas de tiempo vs- variable y1, para tres
    valores del partro de bifurcaci v=0.75, 1 y 1.9.
% ii) Graficar esta misma informaci en un diagrama-fase; i.e.
    x1(t) vs y1(t).
%(opcional:  ar  a este diagrama el campo vectorial, dado por
    [dx1(t)/dt, dy1(t)/dt]
%(?6 figuras).

% constant parameter values:
% Parametros:
alpha1=1; alpha2=1; beta1=200; beta2=10;gamma1=4;
gamma2=4; K1=30; K2=1;%v=1;

% Primero, un ansis  dinco  - integraci  numca  del sistema,
tspan = [0 10];
```

```
y0A = [0 0];
y0B = [0 0.9];

% for the vector plot
[xP,yP] = meshgrid(0:0.1:1,0:0.1:1);

figure_num=1;
ii=1;
for v=[0.75, 1, 1.9];%1.9; % 1.9 only the high steady state %1;
    bistable 0.75 only the low one

[tl,yl] = ode45(@(t,y)Angeli(t,y,v),tspan,y0A);
[th,yh] = ode45(@(t,y)Angeli(t,y,v),tspan,y0B);

subplot(2,3,figure_num)
plot(tl,yl(:,2), 'LineWidth', ii,'Color', 'k')
hold on
plot(th,yh(:,2), 'LineWidth', ii,'Color', 'r')
legend('y0 = [0 0]', 'y0 = [0 9]');
scatter(tl(end), yl(end,2), 'k', 'filled');
scatter(th(end),yh(end,2), 'r', 'filled');
xlabel('t');
ylabel('y(t)');
title([' Dinca  de y, v=' num2str(v)])
 %%
subplot(2,3,3+figure_num)
plot(yl(:,1),yl(:,2), 'LineWidth', ii,'Color', 'k')
hold on
plot(yh(:,1),yh(:,2), 'LineWidth', ii,'Color', 'r')
scatter(yl(end,1), yl(end,2), 'k', 'filled');
scatter(yh(end,1), yh(end,2), 'r', 'filled');
dx1dt=alpha1.*(1-xP)-beta1.*xP.*(v.*yP).^gamma1./(K1+(v.*yP).^gamma1);
dy1dt=alpha2.*(1-yP)-beta2.*yP.*xP.^gamma2./(K2+xP.^gamma2);
quiver(xP, yP,dx1dt,dy1dt,4, 'b')
xlabel('x1(t) ');
ylabel('y1(t)');
title(['diagrama de fase, v=' num2str(v)])
axis square
xlim([0,1]);
ylim([0,1]);
```

```
62
63
64 figure_num=figure_num+1;
65 end
66
67 %% Pregunta B: Caracterizar las separatrices
68 % Regresemos al valor del partro de bifucaci v=1.
69 % probando varias condiciones iniciales, grafica, en una misma
       figura, las
70 % trayectorias x1(t) vs y1(t). te puedes dar una idea del tama
       de las
71 % cuencas de atracci de cada uno de los atractores? qu sucede si
72 % aumentas el valor de v a 1.6?
73
74 jj=1;
75 figure;
76 for v=[1, 1.6]
77 subplot(1,2,jj);
78 hold on
79
80 for ii=1:1:100
81
82     % generate 3 random numbers
83     r1=rand;
84     r2=rand;
85     r3=rand;
86
87     if r1<0.5
88
89     y0 = [r2<0.5 r3];
90     else
91     y0 = [r3 r2<0.5];
92     end;
93
94 [~,yBi] = ode45(@(t,y)Angeli(t,y,v),tspan,y0);
95
96 if abs(yBi(end,2)-0.15)<0.2
97  plot(yBi(:,1),yBi(:,2),'k')
98
99 else
```

```
 plot(yBi(:,1),yBi(:,2),'r')

end

end;

xlabel('x(0)');
ylabel('y(0)');
title(['diagrama de fase, v=' num2str(v)])

[~,yll] = ode45(@(t,y)Angeli(t,y,v),tspan,y0A);
[~,yhh] = ode45(@(t,y)Angeli(t,y,v),tspan,y0B);
scatter(yll(end,1), yll(end,2), 'k', 'filled');
scatter(yhh(end,1), yhh(end,2), 'r', 'filled');
dx1dt=alpha1.*(1-xP)-beta1.*xP.*(v.*yP).^gamma1./(K1+(v.*yP).^gamma1);
dy1dt=alpha2.*(1-yP)-beta2.*yP.*xP.^gamma2./(K2+xP.^gamma2);
quiver(xP, yP,dx1dt,dy1dt,4, 'b')
axis square
xlim([0,1]);
ylim([0,1]);
jj=jj+1;
end;
%% Pregunta (C)
% Los sistemas bifurcantes presentan un comportamiento
    caracterico,
% llamado "alentamiento crco", cuando se acercan al punto de
% bifurcaci . podemos ver algo as en este sistema? para
    comprobarlo,
% grafica, en la misma figura, t vs y1(t) para v=0.2:0.1:1. qu
    observas?
LineWidth=0.1;
LineStyle=[':', '--', ':','--', ':', '--', ':','--', ':','--'];
figure;
ii=1;
col=['k', 'k', 'b','b', 'g', 'g', 'm','m', 'r', 'r'];
for v=0.2:0.1:1;
legendInfo{ii} = ['v = ' num2str(v)];

%[tl,yl] = ode45(@(t,y)Angeli(t,y,v),tspan,y0A);
[th,yh] = ode45(@(t,y)Angeli(t,y,v),tspan,y0B);
```

```
137
138 %plot(tl,yl(:,2), 'LineWidth', LineWidth,'Color', 'k')
139 hold on
140 plot(th,yh(:,2), 'LineWidth', LineWidth,'Color', col(ii),
        'LineStyle', LineStyle(ii));
141 LineWidth=LineWidth+1;
142 ii=ii+1;
143 end;
144 legend(legendInfo)
145 xlabel('t');
146 ylabel('y(t)');
147 title('Critical slowing down');
148 text(0, 0.15, 'Recovery time from a transient perturbation
        increases')
149 text(0, 0.1,'as system approaches the bifurcation point');
150 text(0, 0.9, 'I.C: Transient perturbation');
151 text(10, 1, ':( Unhealthy stable state');
152 text(10, 0.18, ':) Healthy stable state');
153
154 axis square
155
156 function dydt = Angeli(~,y,v)
157
158 % Parametros:
159 alpha1=1; alpha2=1; beta1=200; beta2=10;gamma1=4;
160 gamma2=4; K1=30; K2=1;%v=1;
161
162 dydt = zeros(2,1);
163
164 dydt(1) =
        alpha1*(1-y(1))-beta1*y(1)*(v*y(2))^gamma1/(K1+(v*y(2))^gamma1);
165 dydt(2) =
        alpha2*(1-y(2))-beta2*y(2)*y(1)^gamma2/(K2+y(1)^gamma2);
```

Practica_Angeli.m

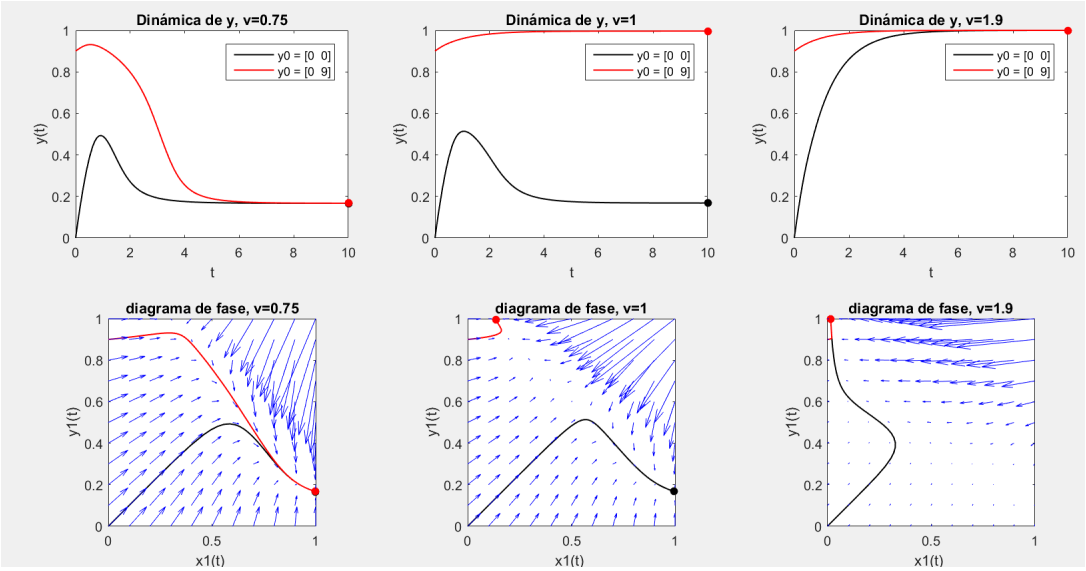


Figure 3.3:

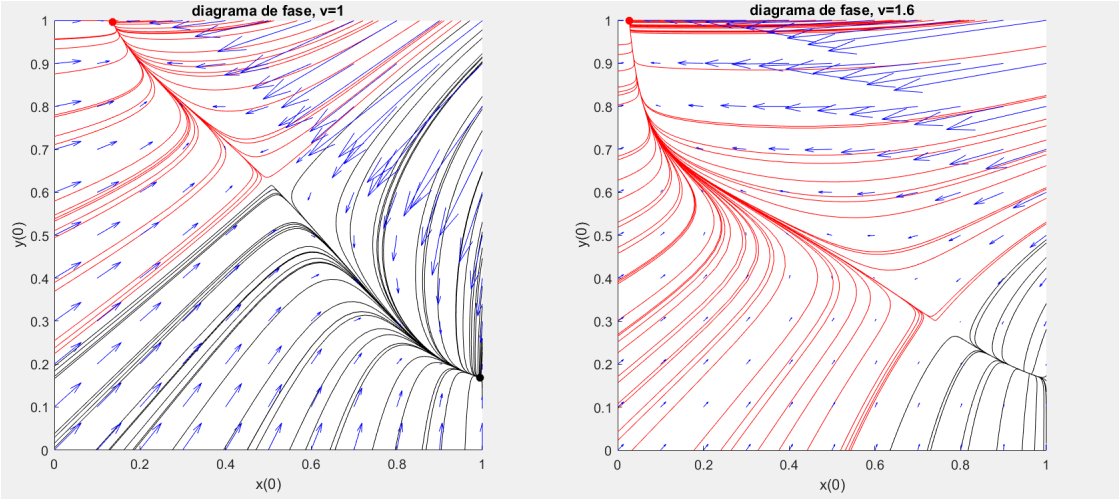


Figure 3.4:

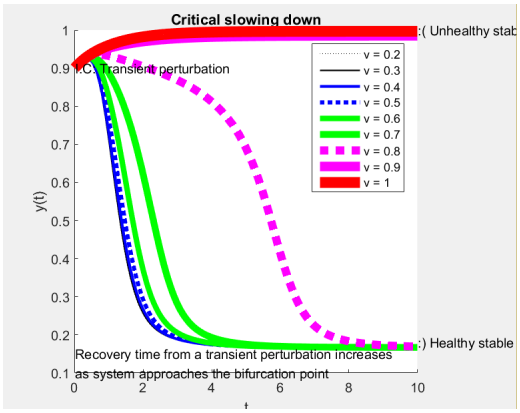

Critical slowing down
:(Unhealthy stab
:) Healthy stable
Transient perturbation
Recovery time from a transient perturbation increases
as system approaches the bifurcation point
v = 0.2
v = 0.3
v = 0.4
v = 0.5
v = 0.6
v = 0.7
v = 0.8
v = 0.9
v = 1
y(t)
t
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
2
4
6
8
10

Figure 3.5:

Práctica 3: Simulación y análisis de modelos estocásticos con el algoritmo de Gillespie

En su artículo de 2002 [16], Elowitz *et al* reportan, quizás por vez primera, mediciones de los niveles de expresión de la proteína verde fluorscente (GFP, por sus siglas en inglés, *Green Flourescent Protein*) de cada una de las células bacterianas en un cultivo celular. Estos datos les permiten constíuir empíricamente las distribuciones poblacionales de los niveles de expresión de la GFP. Desde entonces, se han acentuado los esfuerzos y aportaciones teóricos dirigidos a entender *de dónde surgen / emergen estas distribuciones poblacionales*. Una de las posibles fuentes de ruido / generador de variabilidad en este tipo de sistemas celulares es el **ruido intrínseco**. Éste se debe a que las interacciones bioquímicas que regulan los niveles de expresión genética son procesos estocásticos. En esta práctica, vamos a simular este proceso estocástico de producción y decaimiento de la GFP. Para ello, utilizaremos el algoritmo de Gillespie [22].

1. El sistema bioquímico de formación y degradación de la GFP a considerar en esta práctica está dado por las reacciones representadas por:
   $\emptyset$ -¿[k_1] GFP -¿[k_2] $\emptyset$. Describe con tus palabras estas reacciones.

2. Ignoremos por un momento la cuestión estocástica. Describe este sistema con una Ecuación Diferencial Ordinaria. Esta ecuación, ¿tiene solución analítica? Si sí, ¿cuál es?

3. Ahora, elije una condición inicial y valores para los parámetros. Utilizando `R` o `Matlab`, simula estocásticamente el sistema (una iteración).

4. Simula ahora la ecuación determinista, y compara con su contraparte

estocástica. ¿se parecen? Argumenta, basándote en la gráfica que obtengas.

5. Vuelve a simular estocásticamente el sistema, utlizando las mismas condiciones iniciales y los mismos parámetros. Esta realización, ¿se parece a tu primer simulación? En comparación, ¿qué sucede si simulas nuevamente tu ecuación diferencial determinista? Argumenta, basándote en la gráfica que obtengas.

6. Simula varias veces más (mínimo 50) tu sistema estocástico, guardando cada vez el vector dinámico que obtengas. Saca el promedio poblacional. ¿Se parece a la dinámica de GFP determinista? ¿qué pasa si aumentas/ disminuyes el número de iteraciones? Argumenta, basándote en la gráfica que obtengas. [**Hint:** Recuerda que, si utilizas el algoritmo de Gillespie para simular dinámicamente el sistema, comparar las diferentes trayectorias de GFP(t) va a requerir regularizar tus vectores GFP(t) en una gradilla uniforme. Si usas `Matlab`, puedes usar `interpol` para ello (aunque no es la solución más elegante). Si utilizas `R`, puedes utilizar la función (adaptada de: [66]):

```
discretize <- function(out){
  events=length(out$t)
  start=0; end=out$t[events]; dt=0.01
  len=(end-start)%/%dt
  x=vector("numeric", len)
  target=0; t=0; j=1;
  for(i in 1:events){
    while (out$t[i]>=target){
      x[j]=out$x[i]
      j=j+1; target=target+dt
      t[j]=target
      } } return(list(tdisc=t, xdisc=x)) }
```

7. Finalmente, obtén la distribución poblacional para el tiempo final de tu simulación estocástica. Compara la media poblacional que obtengas con el estado estacionario del sistema determinista. Argumenta, basándote en la gráfica que obtengas. (Nota: simula tu modelo estocástico por *suficiente* tiempo - y argumenta a qué nos referimos con "suficiente tiempo".

4.1 Respuestas

- Los niveles de GFP están controlados por una producción constante (i.e., una reacción de órden zero), y una degradación lineal (i.e., una reacción de primer órden).
- La ODE está dada por:

$$\frac{\partial}{\partial t}\mathrm{GFP}(t) = \mathrm{k1} - \mathrm{k2}\,\mathrm{GFP}(t) \tag{4.1}$$

  cuya solución analítica es:

$$\frac{\mathrm{k1} - \mathrm{e}^{-\mathrm{k2}\,t}\ (\mathrm{k1} - \mathrm{GFP}(0)\,\mathrm{k2})}{\mathrm{k2}} \tag{4.2}$$

- Realizaciones individuales (correspondientes a células individuales en el experimento de Ellowitz et al) de la ecuación estocástica difieren entre sí. A mayor número de iteraciones, la media poblacional converge a su contraparte determinista.
- De igual manera, las distribuciones estacionarias tienden a converger al valor estacionario determinista al aumentar el número de iteraciones.

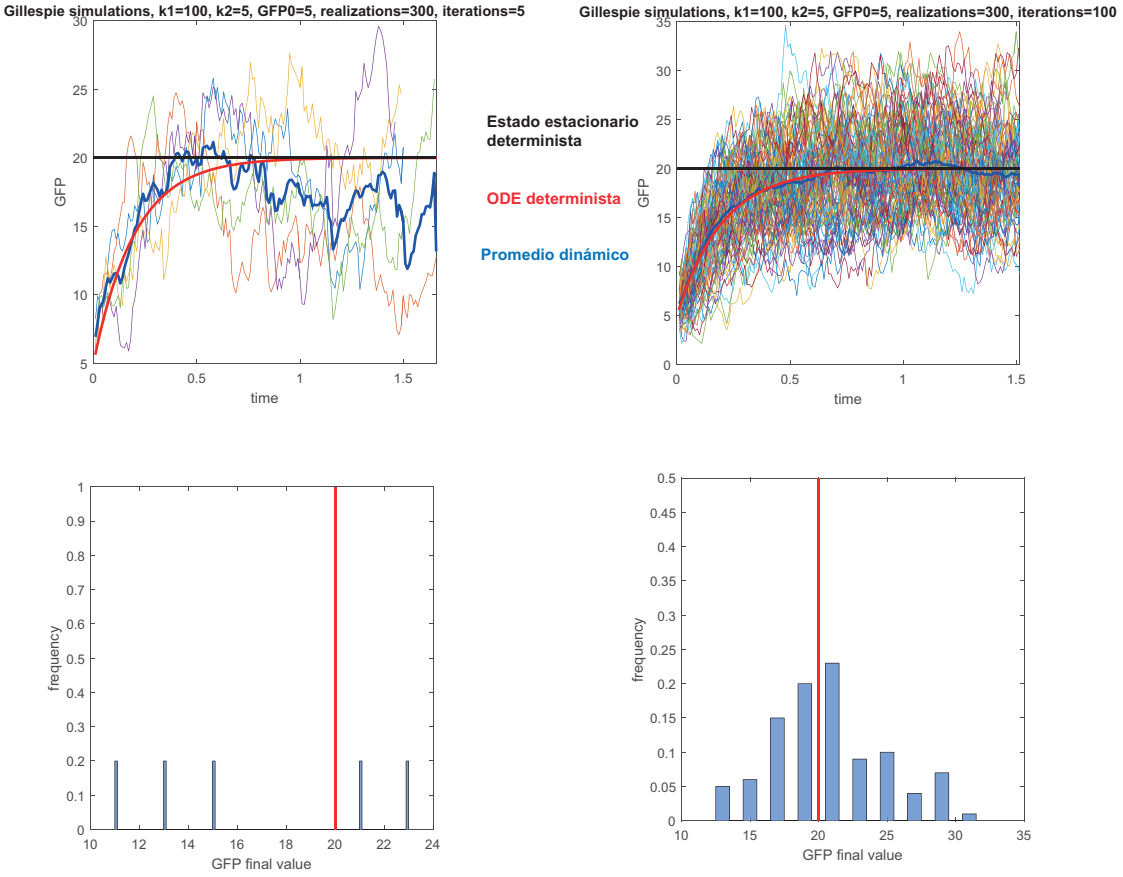


Figure 4.1: Análisis del sistema de formación y degradación de GFP en Matlab..

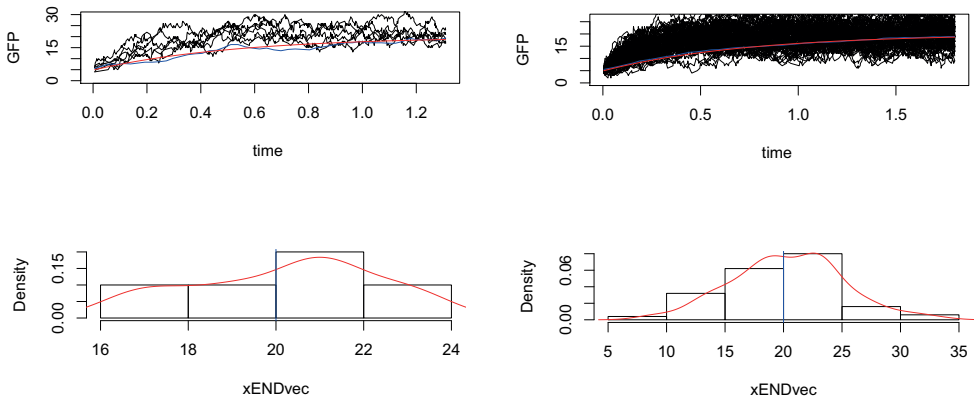


Figure 4.2: .. y en R.

Sistemas estocásticos: ¿de dónde salen las distribuciones estadísticas?

Como seguramente ya notaron, el tipo de modelos matemáticos de sistemas biológicos que estamos aprendiendo a construir y analizar en este curso tienen hipótesis cada vez más relajadas. Empezamos con redes booleanas, asumiendo que el mundo es binario: 0 o 1, prendido o apagado. Y que las reglas que lo "gobiernan" son reglas lógicas: las señales se integran como compuertas lógicas "And" "or", Not". En este mundo

Después, nos sumergimos al mundo de las ecuaciones diferenciales ordinarias, construidas con base en la ley de acción de masas. Aquí aceptamos explícitamente que el mundo es real (en el sentido matemático de la palabra): las variables y el tiempo con contínuos.

Recapitulación

Revisitemos lo que vimos el primer día (1), a la luz del compendio de herramientas matemáticas y computacionales (sistemas dinámicos no lineales en diferentes colores y sabores (contínuos / discretos, deterministas / estocásticos) que hemos introducido.

Ahora: ¡A trabajar en sus propios sistemas! (Trabajo en equipo).

Se tratará de proponer un *pipeline* de análisis a la biología de sistemas que les sirva para responder alguna pregunta de investigación (por ejemplo, de su tesis)

6.1 Plantear la red de interacciones

Primero, tomen una(s) hoja(s) de papel y hagan un esquema del sistema biológico que quieren modelar.

A considerar:

- ¿cuáles son los nodos, cuáles son las interacciones?
- ¿qué variables pueden medir, cuáles no? (subconjunto de variables medibles)
- ¿de qué interacciones hay certeza (y cuál es esta), de cuáles especulan / quieren probar qué tan plausibles son?
- ¿hay experimentos que correspondan a un sub-sistema de su red? (Ej: ensayos de muerte: co-cultivos entre neutrófilos y bacterias para caracterizar la respuesta inmune a estas bacterias.
- ¿Hay perturbaciones externas al sistema?
- Congruencia en la notación gráfica.

Output (a presentar): Una red.

6.2 Formalización

Vimos una gama de formalismos: sistemas dinámicos no lineales en diferentes colores y sabores. ¡A elegir el formalismo más adecuado para tu red! (y justificar tu elección).

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

A considerar:

- Granularidad: ¿cuántas variables tenemos? (modelos cualitativos vs. cuantitativos: pocas vs. muchas variables)
- Definición: ¿tenemos datos cualitativos o cuantitativos? (modelos cualitativos vs. cuantitativos: pocas vs muchas variables)
- Resolución: ¿cuál es la resolución temporal? (puede ser nula: entonces: ¿asumimos que el sistema está en equilibrio? en otras palabras, ¿importa el transitorio?)
- ¿Conocemos la distribución de los datos, o sólo tenemos información sobre la media poblacional (el primer momento, dirían los probabilistas)? (estocásticos vs. deterministas)
- ¿Qué predicciones nos importan? ¿estructurales o cuantitativas?

Output (a presentar): Un formalismo (¡justificado!). Y, de ser posible, un primer esbozo del modelo, con este formalismo. Recuerden: ¡Nos pueden preguntar!

6.3 ¿Qué análisis queremos hacer?

En esta parte, queremos que se detengan un momento a pensar bien cuál es la pregunta biológica que tienen. ¿qué tipo de análisis requieren para responderla? ¿cómo formalizarían esta pregunta?

Quizás les sirva pensar en análisis de perturbaciones (en el sentido amplio de la palabra). Por ejemplo:

- **Análisis de mutantes** (para evaluar el efecto de alteraciones estructurales sobre el sistema)
- **Análisis de bifurcaciones** para evaluar el efecto de alteraciones paramétricas sobre el sistema.
- **Análisis de robustez** para cuantificar "cuántos golpes aguanta la piñata (el sistema biológico) antes del colapso".
- **Análisis de sensibilidad paramétrica** para encontrar los parámetros que más probablemente afectan el sistema?
- **Análisis de optimización** para encontrar las condiciones mínimas / máximas tal que *algo* ocurra.
- *Simulación de modelos estocásticos bifurcantes* en búsqueda de señales de alerta temprana.
- **Optimización paramétrica** para cuantificar el error mínimo entre el modelo y los datos experimentales.

6.4 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?).

6.5 Discusión de resultados

Dedicaremos la última hora del jueves para discutir sus propuestas.

Bibliography

[1]

[2] María Elena Álvarez-Buylla Roces, Juan Carlos Martínez-García, José Dávila-Velderrain, Elisa Domínguez-Hüttinger, and Mariana Esther Martínez-Sánchez. *Modeling Methods for Medical Systems Biology*, volume 1069. Springer, 2018.

[3] 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.

[4] John Burdon and Sanderson Haldane. A note on the kinetics of Enzyme Action. *Biochem J*, 19(2):338–339, 1925.

[5] Flavio Cannavó. Sensitivity analysis for volcanic source modeling quality assessment and model selection. *Comput. Geosci.*, 44:52–59, 2012.

[6] Madalena Chaves and Jean-Luc Gouzé. Exact control of genetic networks in a qualitative framework: The bistable switch example. *Automatica*, 47(6):1105–1112, jun 2011.

[7] Carlo Cosentino and Declan Bates. *Feedback Control in Systems Biology*. CRC Press, 2012.

[8] Gheorghe Craciun and Martin Feinberg. Multiple equilibria in complex chemical reaction networks: i. the injectivity property â^—. *SIAM J Appl Math*, 65(5):1526–1546, 2005.

[9] Gheorghe Craciun, Yangzhong Tang, and Martin Feinberg. Understanding bistability in complex enzyme-driven reaction networks. *PNAS*, 103(23):8697–8702, 2006.

[10] Elisa Domínguez-Hüttinger. *Mathematical modelling of epithelium homeostasis*. Phd, Imperial College London, 2015.

[11] Elisa Domínguez-Hüttinger, Neville J Boon, Thomas B Clarke, and Reiko J. Tanaka. Mathematical Modeling of Streptococcus pneumoniae Colonization, Invasive Infection and Treatment. *Front. Physiol.*, 8(115):1–14, 2017.

[12] Elisa Domínguez-Hüttinger, Panayiotis Christodoulides, Kosuke Miyauchi, Alan D. Irvine, Mariko Okada-Hatakeyama, Masato Kubo, and Reiko J. Tanaka. Mathematical Modeling of Atopic Dermatitis Reveals "Double switch" Mechanisms Underlying Four Common Disease Phenotypes. *J. Allergy Clin. Immunol.*, (February), 2017.

[13] Elisa Domínguez-Hüttinger, Masahiro Ono, Mauricio Barahona, and Reiko J Tanaka. Risk factor-dependent dynamics of atopic dermatitis : modelling multi-scale regulation of epithelium homeostasis. *Interface Focus*, 3(February):20120090, 2013.

[14] Andreas Doncic, Oguzhan Atay, Mart Loog, Jan M Skotheim, Andreas Doncic, Oguzhan Atay, Ervin Valk, Alicia Grande, Alan Bush, Gustavo Vasen, Alejandro Colman-lerner, Mart Loog, and Jan M Skotheim. Compartmentalization of a Bistable Switch Enables Memory to Cross a Feedback-Driven Transition. *Cell*, 160(6):1182–1195.

[15] Thomas Eissing, Holger Conzelmann, D Gilles, Frank Allgöwer, Eric Bullinger, Peter Scheurich, and Ernst D Gilles. Mechanisms of Signal Transduction : Bistability Analyses of a Caspase Activation Model for Receptor-induced Apoptosis Bistability Analyses of a Caspase Activation Model for Receptor-induced Apoptosis *. *J. Biol. Chem.*, 279, 2004.

[16] M B Elowitz, a J Levine, E D Siggia, and P S Swain. Stochastic gene expression in a single cell. *Science (80-.).*, 297(5584):1183–1186, 2002.

[17] L. Espinar, M. Dies, T. Cagatay, G. M. Suel, and Jordi García-Ojalvo. Circuit-level input integration in bacterial gene regulation. *PNAS*, 110(17), apr 2013.

[18] James E Ferrell, Joseph R Pomerening, Sun Young Kim, Nikki B Trunnell, Wen Xiong, Chi-Ying Frederick Huang, and Eric M Machleder. Simple, realistic models of complex biological processes: positive feedback and bistability in a cell fate switch and a cell cycle oscillator. *FEBS Lett.*, 583(24):3999–4005, dec 2009.

[19] J. E. Ferrell Jr. The Biochemical Basis of an All-or-None Cell Fate Switch in Xenopus Oocytes. *Science (80-.).*, 280(5365):895–898, may 1998.

[20] Dirk Fey, Yannick Blum, Sung Sik Lee, Hyunryul Ryu, Minhwan Chung, Maciej Dobrzy, Matthias Peter, Boris N Kholodenko, Noo Li Jeon, and Olivier Pertz. Frequency modulation of ERK activation dynamics rewires cell fate. *Mol. Syst. Biol.*, pages 1–14, 2015.

[21] Dirk Fey, Melinda Halasz, Daniel Dreidax, Sean P Kennedy, Jordan F Hastings, Nora Rauch, Amaya Garcia Munoz, Ruth Pilkington, Matthias Fischer, Frank Westermann, Walter Kolch, Boris N Kholodenko, and David R Croucher. Signaling pathway models as biomarkers : Patient-specific simulations of JNK activity predict the survival of neuroblastoma patients. *Sci. Signal.*, 8(408):1–16, 2015.

[22] Daniel T Gillespie. Exact Stochastic Simulation of Coupled Chemical Reactions Danlel. *J. Phys. Chem.*, 93555(1):2340–2361, 1977.

[23] a. Goldbeter and D. E. Koshland. Ultrasensitivity in biochemical systems controlled by covalent modification. Interplay between zero-order and multistep effects. *J. Biol. Chem.*, 259:14441–14447, 1984.

[24] Niels Grabe and Karsten Neuber. A multicellular systems biology model predicts epidermal morphology, kinetics and Ca2+ flow. *Bioinformatics*, 21(17):3541–7, sep 2005.

[25] Melinda Halasz, Boris N Kholodenko, Walter Kolch, and Tapesh Santra. Integrating network reconstruction with mechanistic modelling to predict cancer therapy. 114(i), 2016.

[26] Heather A Harrington, Kenneth L Ho, Samik Ghosh, and K C Tung. Construction and analysis of a modular model of caspase activation in apoptosis. *Theor. Biol. Med. Model.*, 5:26, jan 2008.

[27] B. S. Hijmans, C. a. Tiemann, a. Grefhorst, M. Boesjes, T. H. van Dijk, U. J. F. Tietge, F. Kuipers, Natal Aw van Riel, a. K. Groen, and M. H. Oosterveer. A systems biology approach reveals the physiological origin of hepatic steatosis induced by liver X receptor activation. *FASEB J.*, pages 1–12, 2014.

[28] Thomas Höfer, Holger Nathansen, Max Löhning, Andreas Radbruch, and Reinhart Heinrich. GATA-3 transcriptional imprinting in Th2 lymphocytes: a mathematical model. *PNAS*, 99(14):9364–8, jul 2002.

[29] Sun-young Hwang, Kye-yeon Hur, Jeong-Rae Junil Kim, Kwang-hyun Cho, Seung-hwan Kim, and Joo-yeon Yoo. Biphasic RLR IFN- β Response Controls the Balance between Antiviral Immunity and Cell Damage. *J. Immunol.*, 190:1192–1200, 2013.

[30] Kentaro Inoue, Hisaaki Shinohara, Marcelo Behar, Noriko Yumoto, Gouhei Tanaka, Alexander Hoffmann, Kazuyuki Aihara, and Mariko Okada-Hatakeyama. Oscillation dynamics underlie functional switching of NF-κB for B-cell activation. *npj Syst. Biol. Appl.*, 2(April):16024, 2016.

[31] Yuri A. Kuznetsov. Elements of Applied Bifurcation Theory. *Library (Lond).*, page 591, 1998.

[32] Gabriele Lillacci and Mustafa Khammash. Parameter Estimation and Model Selection in Computational Biology. 6(3), 2010.

[33] Tomasz Lipniacki, Beata Hat, James R Faeder, and William S Hlavacek. Stochastic effects and bistability in T cell receptor signaling. *J. Theor. Biol.*, 254(1):110–22, sep 2008.

[34] Simeone Marino, Ian B Hogue, Christian J Ray, and Denise E Kirschner. A methodology for performing global uncertainty and sensitivity analysis in systems biology. *J. Theor. Biol.*, 254(1):178–96, sep 2008.

[35] Johannes Müller and Thorsten Tjardes. Modeling the Cytokine Network In Vitro and In Vivo. *J. Theor. Med.*, 5(2):93–110, 2003.

[36] J D Murray. *Mathematical biology*. 3 edition, 2003.

[37] Jin Nam, Baltazar D Aguda, Bjoern Rath, and Sudha Agarwal. Biomechanical thresholds regulate inflammation through the NF-kappaB pathway: experiments and modeling. *PLoS One*, 4(4):e5262, jan 2009.

[38] Nicolas Le Novère, Michael Hucka, Huaiyu Mi, Stuart Moodie, Falk Schreiber, Anatoly Sorokin, Emek Demir, Katja Wegner, Mirit I Aladjem, Sarala M Wimalaratne, Frank T Bergman, Ralph Gauges, Peter Ghazal, Hideya Kawaji, Lu Li, Yukiko Matsuoka, Alice Villéger, Sarah E Boyd, Akiya Jouraku, Sohyoung Kim, Fedor Kolpakov, Augustin Luna, Sven Sahle, and Esther Schmidt. The Systems Biology Graphical Notation : Abstract : Nature Biotechnology. *Nat. Biotechnol.*, 27(8):735–742, 2009.

[39] L Olsen, J a Sherratt, and Philip K Maini. A mathematical model for fibro-proliferative wound healing disorders. *Bull. Math. Biol.*, 58(4):787–808, jul 1996.

[40] Diego Oyarzún, Madalena Chaves, and Marit Hoff-Hoffmeyer-Zlotnik. Multistability and oscillations in genetic control of metabolism. *J. Theor. Biol.*, 295:139–53, feb 2012.

[41] Henning Schmidt and Mats Jirstrand. Systems Biology Toolbox for MATLAB: a computational platform for research in systems biology. *Bioinformatics*, 22(4):514–5, feb 2006.

[42] Najaf a Shah and Casim a Sarkar. Robust network topologies for generating switch-like cellular responses. *PLoS Comput. Biol.*, 7(6):e1002085, jun 2011.

[43] Alex K Shalek, Rahul Satija, Xian Adiconis, Rona S Gertner, Jellert T Gaublomme, Raktima Raychowdhury, Schraga Schwartz, Nir Yosef, Christine Malboeuf, Diana Lu, John J Trombetta, Dave Gennert, Andreas Gnirke, Alon Goren, Nir Hacohen, Joshua Z Levin, Hongkun Park, and Aviv Regev. Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. *Nature*, 498(7453):236–40, jun 2013.

[44] Sunmee Shin, Je Kyung Seong, and Yun Soo Bae. Ahnak stimulates BMP2-mediated adipocyte differentiation through Smad1 activation. *Obesity*, 24(2):398–407, 2016.

[45] Hanna L Sladitschek and Pierre A Neveu. The bimodally expressed microRNA miR- 142 gates exit from pluripotency. *Mol. Syst. Biol.*, pages 1–16, 2015.

[46] Karline Soetaert, Jeff Cash, and Francesca Mazzia. *Use R! Solving Differential equations in R.* 2012.

[47] Jörg Stelling, Uwe Sauer, Zoltan Szallasi, Francis J. Doyle, and John Doyle. Robustness of cellular functions. *Cell*, 118:675–685, 2004.

[48] Steven H. Strogatz. *Nonlinear dynamics and chaos.* 2000.

[49] Myong-Hee Sung, Ning Li, Qizong Lao, Rachel a Gottschalk, Gordon L Hager, and Iain D C Fraser. Switching of the Relative Dominance Between Feedback Mechanisms in Lipopolysaccharide-Induced NF-κB Signaling. *Sci. Signal.*, 7(308):ra6, jan 2014.

[50] Thomas Sütterlin, Simone Huber, Hartmut Dickhaus, and Niels Grabe. Modeling multi-cellular behavior in epidermal tissue homeostasis via finite state machines in multi-agent systems. *Bioinformatics*, 25(16):2057–63, aug 2009.

[51] Thomas Sütterlin, Christoph Kolb, Hartmut Dickhaus, Dirk Jäger, and Niels Grabe. Bridging the scales: semantic integration of quantitative SBML in graphical multi-cellular models and simulations with EPISIM and COPASI. *Bioinformatics*, 29(2):223–9, jan 2013.

[52] Zoltan Szallasi, Vipul Periwal, and Jörg Stelling. *System Modeling in Cellular Biology: From Concepts to Nuts and Bolts*. The MIT Press, 2006.

[53] Gouhei Tanaka, Panayiotis Christodoulides, Elisa Domínguez-Hüttinger, Kazuyuki Aihara, and Reiko J. Tanaka. Bifurcation analysis of a mathematical model of atopic dermatitis to determine patient-specific effects of treatments on dynamic phenotypes. *J. Theor. Biol.*, pages 1–38.

[54] Reiko J Tanaka, Masahiro Ono, and Heather a Harrington. Skin barrier homeostasis in atopic dermatitis: feedback regulation of kallikrein activity. *PLoS One*, 6(5):e19895, jan 2011.

[55] Christian a Tiemann, Joep Vanlier, Peter Aj Hilbers, and Natal Aw van Riel. Parameter adaptations during phenotype transitions in progressive diseases. *BMC Syst. Biol.*, 5(1):174, 2011.

[56] Christian a. Tiemann, Joep Vanlier, Maaike H. Oosterveer, Albert K. Groen, Peter a J Hilbers, and Natal Aw van Riel. Parameter Trajectory Analysis to Identify Treatment Effects of Pharmacological Interventions. *PLoS Comput. Biol.*, 9(8), 2013.

[57] Abhinav Tiwari, J Christian J Ray, Jatin Narula, and Oleg a Igoshin. Bistable responses in bacterial genetic networks: designs and dynamical consequences. *Math. Biosci.*, 231(1):76–89, may 2011.

[58] Tsz-Leung To and Narendra Maheshri. Noise Can Induce Bimodality in positive transcriptional feedback loops without bistability. *Science (80-.).*, 1142(February):1142, 2010.

[59] Tina Toni and Michael P H Stumpf. Simulation-based model selection for dynamical systems in systems and population biology. *Bioinformatics*, 26(1):104–110, 2010.

[60] Mark K. Transtrum and Peng Qiu. Bridging Mechanistic and Phenomenological Models of Complex Biological Systems. *PLoS Comput. Biol.*, pages 1–10, 2015.

[61] Said Elnashaie Frank Uhlig and with the assistance of Chadia Affane. *Numerical Techniques for Chemical and Biological Engineers Using MATLAB*.

[62] Najl V Valeyev, Christian Hundhausen, Yoshinori Umezawa, Nikolay V Kotov, Gareth Williams, Alex Clop, Crysanthi Ainali, Christos Ouzounis, Sophia Tsoka, and Frank O Nestle. A systems model for immune cell interactions unravels the mechanism of inflammation in human skin. *PLoS Comput. Biol.*, 6(12):e1001024, jan 2010.

[63] Natal Aw van Riel, Christian a Tiemann, Joep Vanlier, and Peter a J Hilbers. Applications of analysis of dynamic adaptations in parameter trajectories. *Interface Focus*, 3:20120084, 2013.

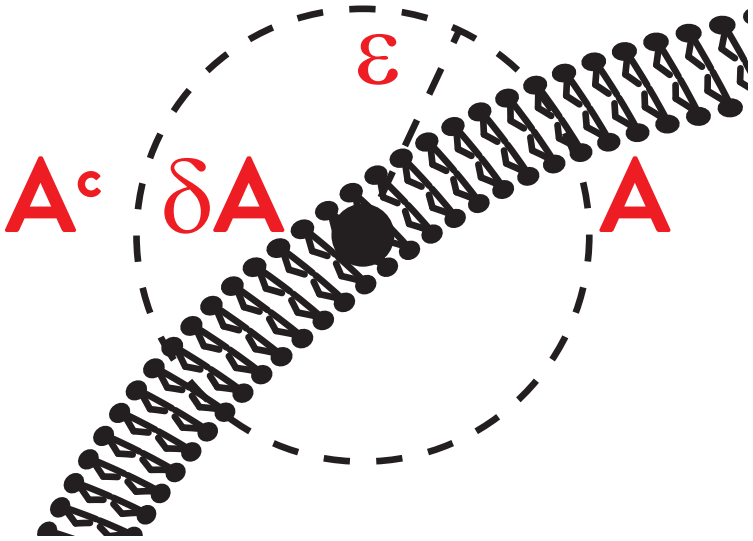
[64] Marc Weber and Javier Buceta. Dynamics of the quorum sensing switch: stochastic and non-stationary effects. *BMC Syst. Biol.*, 7:6, 2013.

[65] Thomas Wilhelm. The smallest chemical reaction system with bistability. *BMC Syst. Biol.*, 9:1–9, 2009.

[66] Darren J Wilkinson. *Stochastic Modelling for Systems Biology*. Chapman and Hall, 2006.

[67] Qiang Zhang, Sudin Bhattacharya, Douglas E Kline, Robert B Crawford, Rory B Conolly, Russell S Thomas, Norbert E Kaminski, and Melvin E Andersen. Stochastic modeling of B lymphocyte terminal differentiation and its suppression by dioxin. *BMC Syst. Biol.*, 4:40, jan 2010.

[68] T. Zhang, B. Schmierer, and B. Novak. Cell cycle commitment in budding yeast emerges from the cooperation of multiple bistable switches. *Open Biol.*, 1:110009–110009, 2011.



Systems Biology

IIB - UNAM