



Synthetic Biology and Biosystems Control Lab
Valencia UPV



Modeling: ODEs and Hill Functions

Section 1: ODEs, Law of mass action and the central dogma

by Alejandro Vignoni (alvig2@upv.es)

An iGEM Measurement Committee Webinar

Week 2, June 23rd, 2020



$$\begin{cases} \frac{d[LuxI]_i}{dt} = b_i + u([AHL]_i) - \gamma_i [LuxI]_i \\ \frac{d[AHL]_i}{dt} = K_A [LuxI]_i + d([AHL]_i - [AHL]_e) - \gamma_A [AHL]_i \end{cases}$$

Today Webinar's Topics

- ⚠ Section 1: ODEs, the law of mass action, and the central dogma (15 min)
- ⚠ Section 2: Derivation of a Hill function from the law of mass action (15 min)
- ⚠ Section 3: Hill function examples and intuitions: effects of parameters on activators, repressors, hybrid promoters, using a Matlab exploration package. (15min)
- ⚠ Q&A – (at the end of each 15 minutes block, total 15 min)

Types of models

Schematic

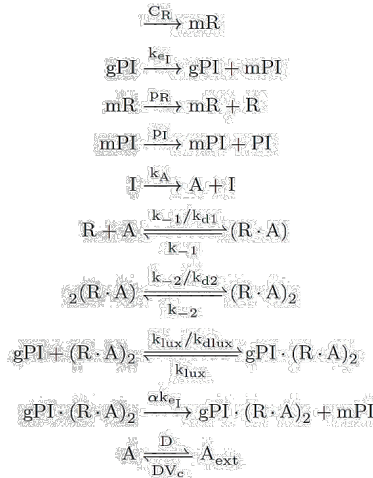
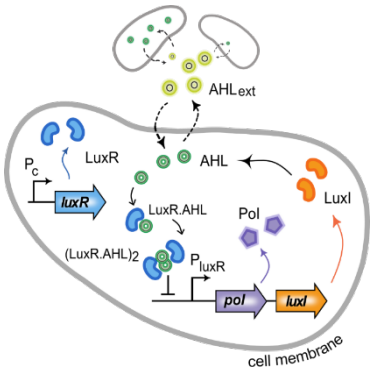
Biochemical
Reactions

Mathematical
Model (ODEs)

Reduced
Mathematical
Model

Part I

Part II



$$\begin{aligned}
 \dot{n}_1^i &= k_{eI} n_7^i + \alpha k_{eI} n_8^i - d_{mI} n_1^i \\
 \dot{n}_2^i &= C_R - d_{mR} n_2^i \\
 \dot{n}_3^i &= p_I n_1^i - d_I n_3^i \\
 \dot{n}_4^i &= p_R n_2^i + k_{-1} n_5^i - d_R n_4^i - \frac{k_{-1}}{k_{d1}} n_9^i n_4^i \\
 \dot{n}_5^i &= 2k_{-2} n_6^i + \frac{k_{-1}}{k_{d1}} n_9^i n_4^i + \left(-k_{-1} - d_{RA} - 2\frac{k_{-2}}{k_{d2}} n_6^i \right) n_5^i \\
 \dot{n}_6^i &= k_{lux} n_8^i + \frac{k_{-2}}{k_{d2}} n_5^i{}^2 + \left(-k_{-2} - d_{RA_2} - \frac{k_{lux}}{k_{dlux}} n_7^i \right) n_6^i \\
 \dot{n}_7^i &= k_{lux} n_8^i - \frac{k_{lux}}{k_{dlux}} n_6^i n_7^i \\
 \dot{n}_8^i &= -k_{lux} n_8^i + \frac{k_{lux}}{k_{dlux}} n_6^i n_7^i \\
 \dot{n}_9^i &= D \left(V_c n_{10} - n_9^i \right) - \left(\frac{k_{-1}}{k_{d1}} n_4^i + d_A \right) n_9^i + k_{-1} n_5^i + k_A n_3^i \\
 \dot{n}_{10} &= D \left(-N V_c n_{10} + \sum_{i=1}^N n_0^i \right) - d_{A_*} n_{10}
 \end{aligned}$$

$$\begin{aligned}
 \dot{n}_1^i &= \frac{C_I p_I}{d_{mI}} \left(\frac{k_{dlux} + a n_3^i}{k_{dlux} + n_3^i} \right) - d_I n_1^i \\
 \dot{n}_2^i &= \frac{C_R p_R}{d_{mR}} + k_{-1} n_6^i - \left(\frac{k_{-1}}{k_{d1}} n_4^i + d_R \right) n_2^i \\
 \dot{n}_3^i &= \frac{k_{-2}}{k_{d2}} (n_6^i)^2 - (k_{-2} + d_{RA_2}) n_3^i \\
 \dot{n}_4^i &= k_{-1} n_6^i + k_A n_1^i + D \left(\frac{n_5^i}{V_c} - n_4^i \right) - \left(\frac{k_{-1}}{k_{d1}} n_2^i + d_A \right)
 \end{aligned}$$

Easier to use and to relate
with experimental data

But what is an Ordinary Differential Equation (ODE)?

These are equations with **variables** and their **derivatives**

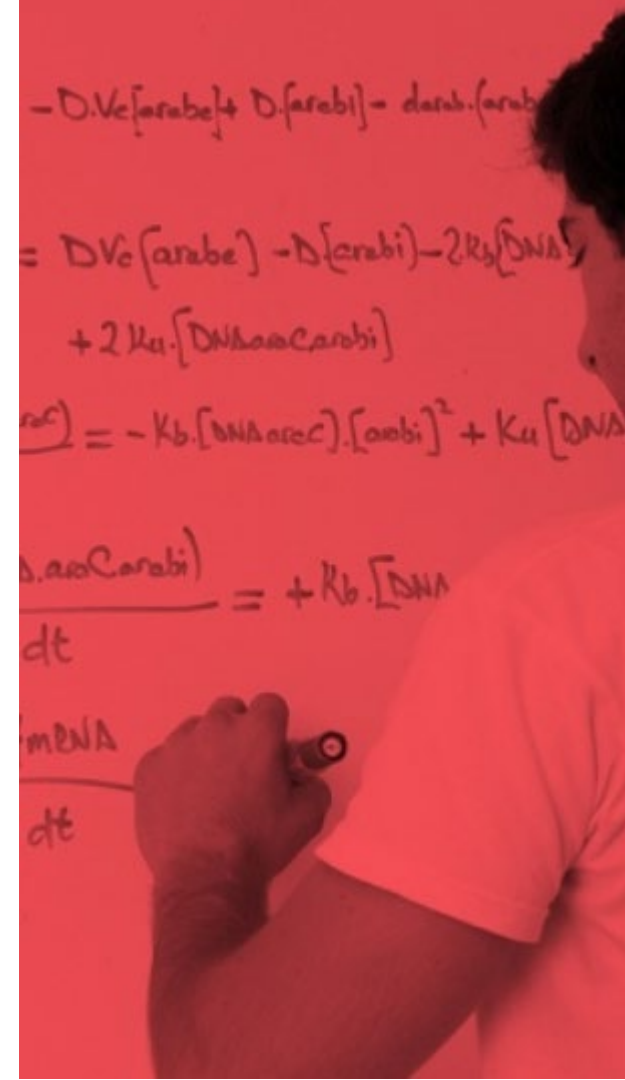
If we have any function (the typical one):

$$y = f(t) \quad (y \text{ only depends on the variable } t, \text{ but we could have } y = f(t, x_1, x_2, \dots, x_n))$$

Do you remember the definition of the derivative of a function?

$$\dot{y} = \frac{df(t)}{dt} = \lim_{h \rightarrow 0} \frac{f(t+h) - f(t)}{h} \quad (\text{we can have higher order derivatives } y'', y''', y^{(n)})$$

But they can be very
challenging and difficult to solve!!



But if they're so complicated... how do we solve them?

We can solve differential equations in two ways:

⚠ **Analytically:** solving for the unknown...



⚠ **Numerically:** in an approximate way.

$$\dot{y} \simeq \frac{f(h+h)-f(x)}{h}$$

(with an h very small)

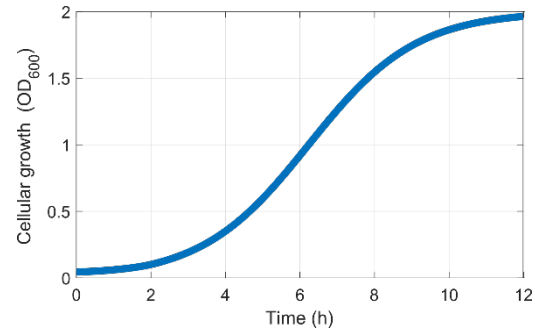


Why do we use them?

Differential equations describe biological behaviour, physical laws, human activities, and much more....



$$\frac{dx}{dt} = \mu x \left(1 - \frac{x}{x_{max}} \right)$$

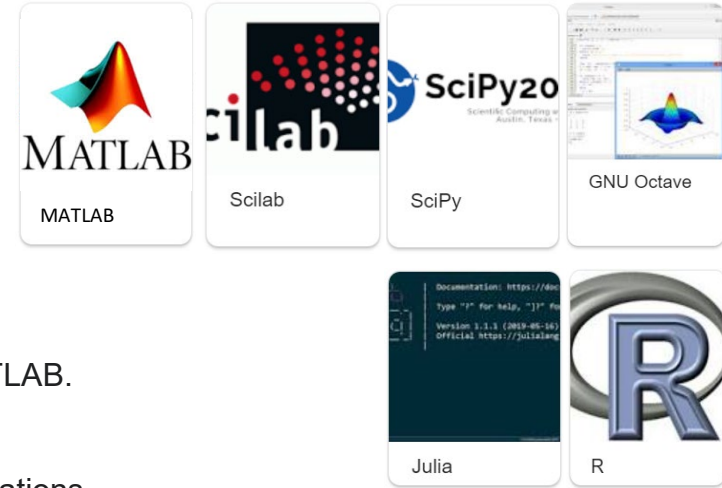


And the set of equations that describe a system, a phenomenon...
is known as ODE model

Software for Ordinary Differential Equations (ODEs) solving

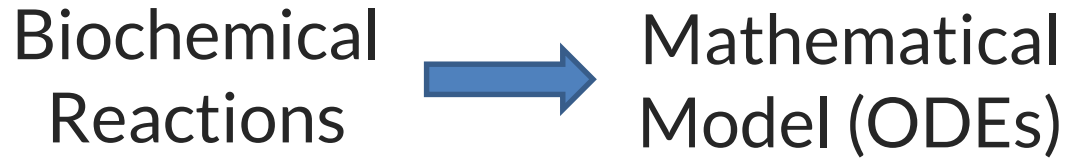
- [MATLAB](#), a technical computing application (MATrix LABoratory)

FREE LICENSE WITH iGEM and the Measurement Committee has some software programed in MATLAB for flow cytometry and plate reader data analysis and calibration.



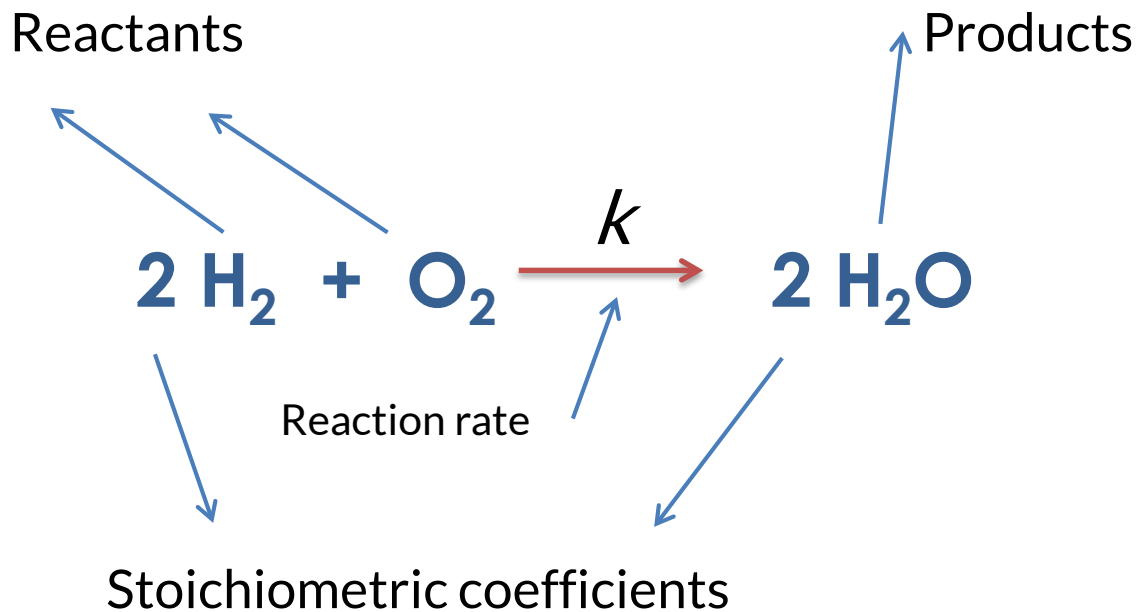
- [Maxima](#), an open-source [computer algebra system](#).
- [COPASI](#), a free software package for the integration and analysis of ODEs.
- [GNU Octave](#), a high-level language, basically a open-source versión of MATLAB.
- [Scilab](#), an open source application for numerical computation.
- [Maple](#), a proprietary application for symbolic calculations.
- [Mathematica](#), a proprietary application primarily intended for symbolic calculations.
- [Julia \(programming language\)](#), a high-level language primarily intended for numerical computations.
- [SageMath](#), an open-source application that uses a Python-like syntax with a wide range of capabilities spanning several branches of mathematics.
- [SciPy](#), a Python package that includes an ODE integration module.
- [GNU R](#), an open source computational environment primarily intended for statistics, which includes packages for ODE solving.

Let us begin this journey **Part I** from:



Reminder: Law of mass action and kinetic equations

Example: Reaction of Water



Law of mass action and kinetic equations

Reaction of Water – Kinetics of H_2



Rate of change
of $[H_2]$

Decreases or
it is consumed

$$[\dot{H}_2] = -2k[H_2]^2[O_2]$$

Stoichiometric
coefficient of $[H_2]$ times
the reaction rate k

product of the concentrations of the
reactants ($[H_2] \times [H_2] \times [O_2] = [H_2]^2[O_2]$)

Law of mass action and kinetic equations

Reaction of Water – Kinetics of O_2



Rate of change
of $[O_2]$

Decreases or
it is consumed

$$[\dot{O}_2] = - k [H_2]^2 [O_2]$$

Stoichiometric
coefficient of $[O_2]$ times
the reaction rate k

product of the concentrations of the
reactants ($[H_2] \times [H_2] \times [O_2] = [H_2]^2 [O_2]$)

Law of mass action and kinetic equations

Reaction of Water – Kinetics of H_2O



Rate of change
of $[H_2O]$

Increases or
it is produced

$$[H_2O] = +2k[H_2]^2[O_2]$$

Stoichiometric
coefficient of $[H_2O]$
times the reaction rate k

product of the concentrations of the
reactants ($[H_2] \times [H_2] \times [O_2] = [H_2]^2[O_2]$)

Law of mass action and kinetic equations

Reaction of Water – Kinetics



$$[\dot{H}_2] = -2k[H_2]^2[O_2]$$

$$[\dot{O}_2] = -k[H_2]^2[O_2]$$

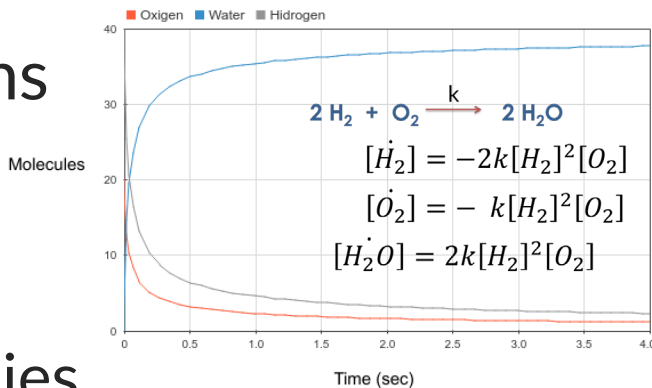
$$[\dot{H}_2O] = 2k[H_2]^2[O_2]$$



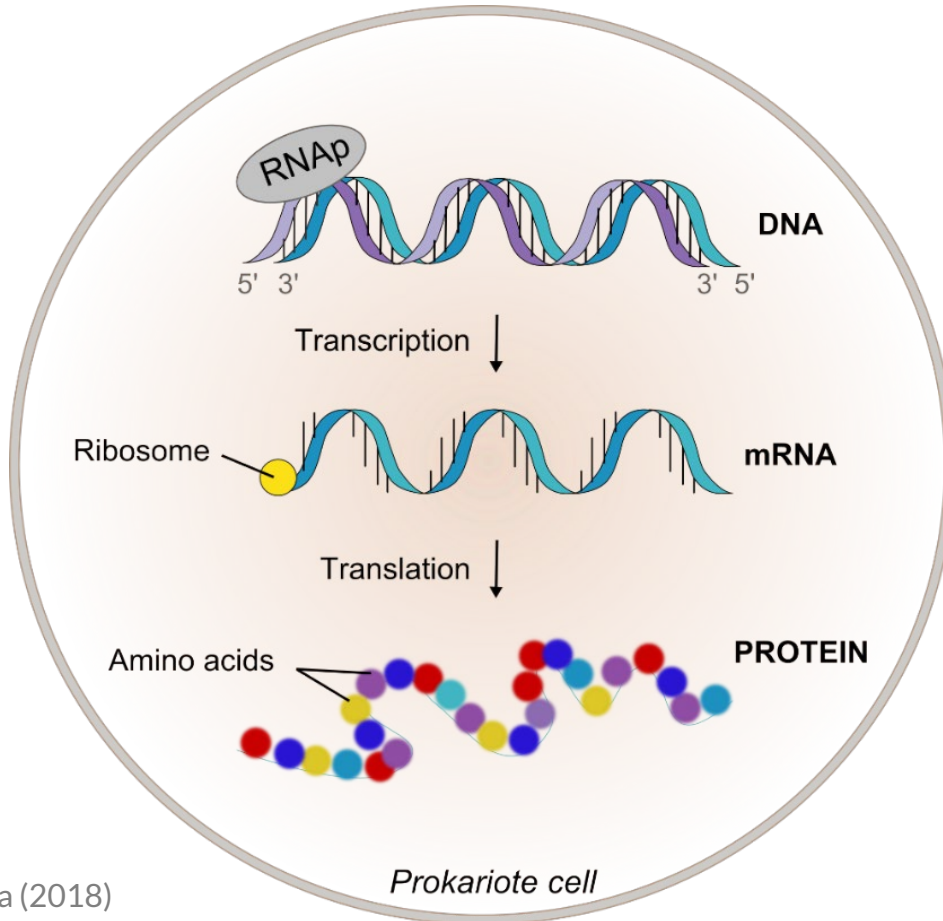
Law of mass action and kinetic equations

Summary

- ⚠ One equation for each one of the species
- ⚠ Rate of change of $[A]$ (concentration of species A) is proportional to:
 - ⚠ Stoichiometric coefficient of A time reaction rate k
 - ⚠ The product of the concentrations of the reactants



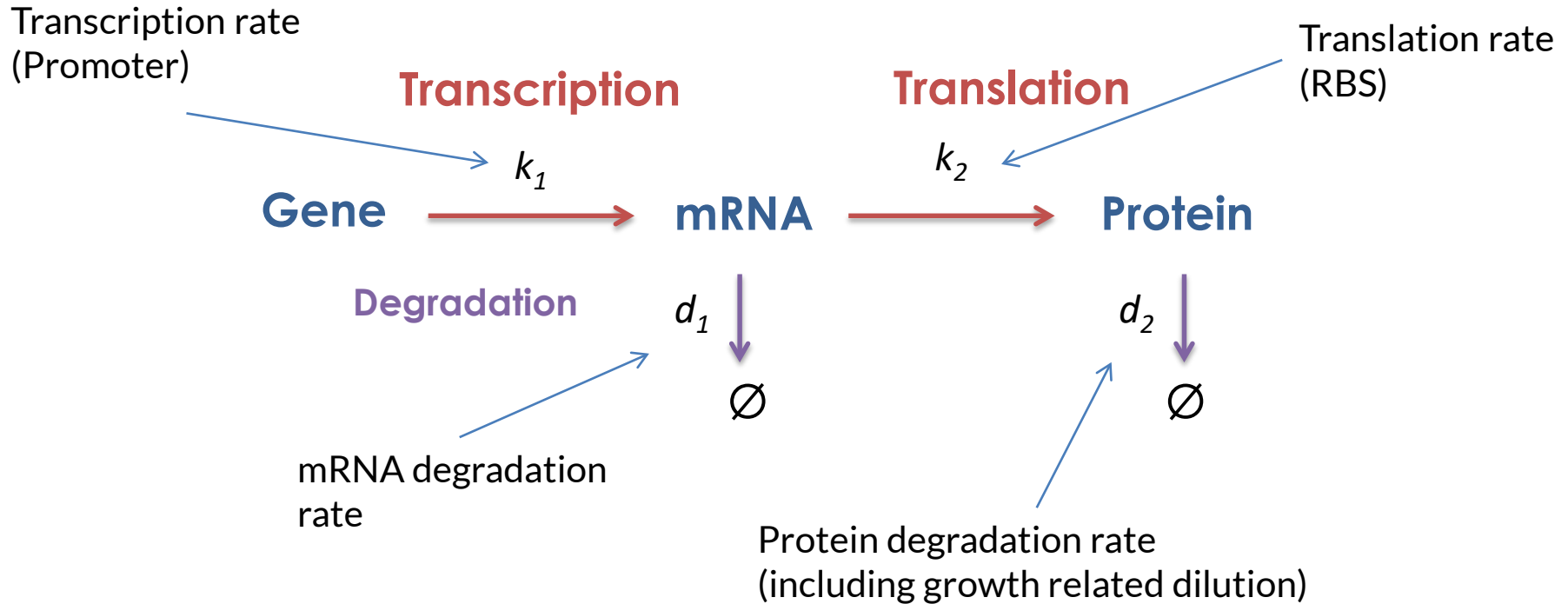
The central dogma of molecular biology



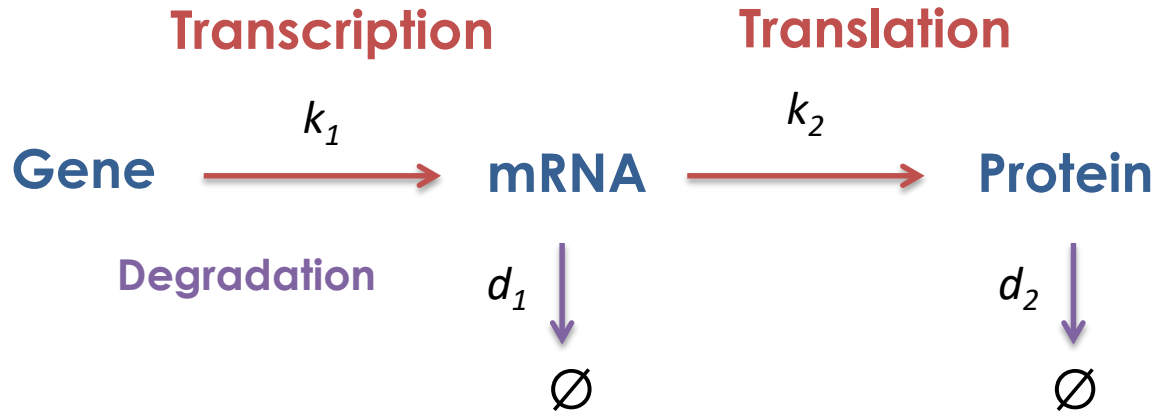
Transcription of DNA
by RNA polymerase

Translation of mRNA by
Ribosomes

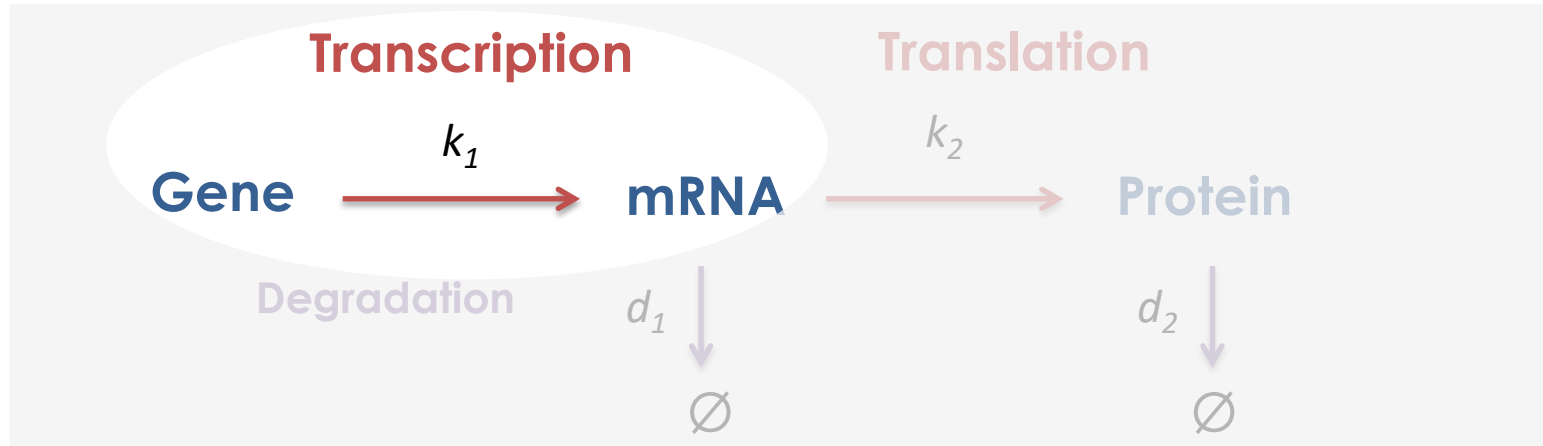
Constitutive gene expression (Simplified version)



Constitutive gene expression (Simplified version)

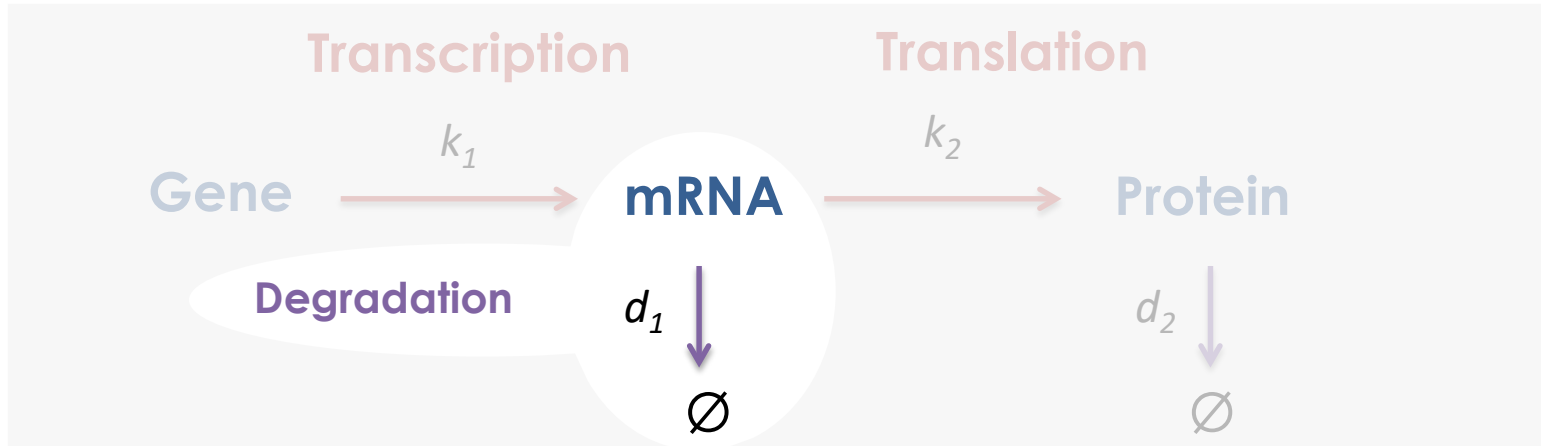


Constitutive gene expression (Simplified version)



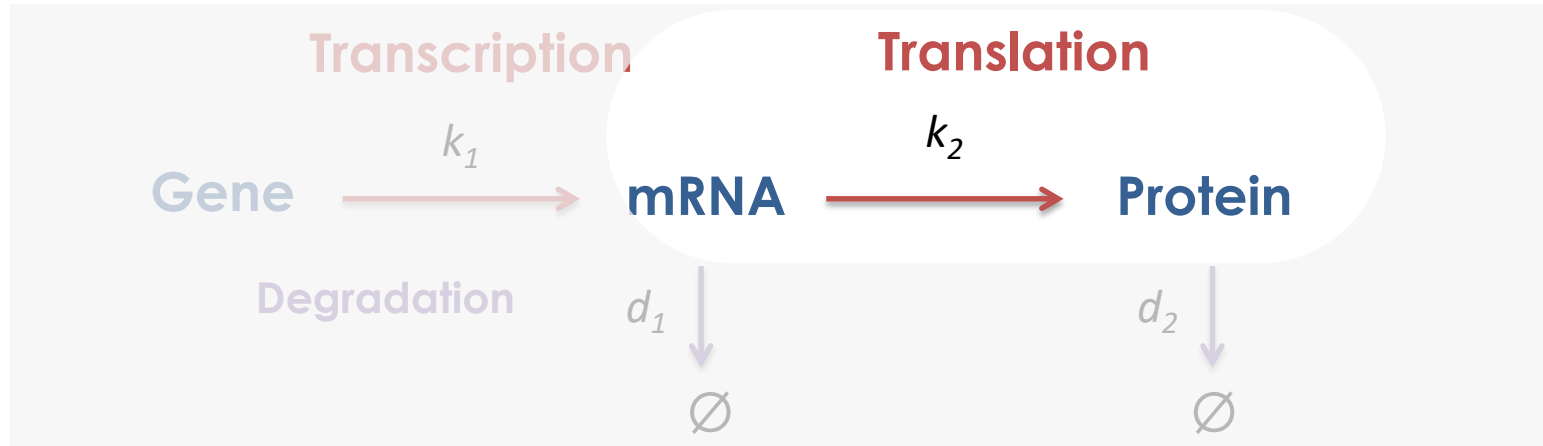
$$\dot{[mRNA]} = k_1 [Gene]$$

Constitutive gene expression (Simplified version)



$$\dot{[\text{mRNA}]} = k_1 [\text{Gene}] - d_1 [\text{mRNA}]$$

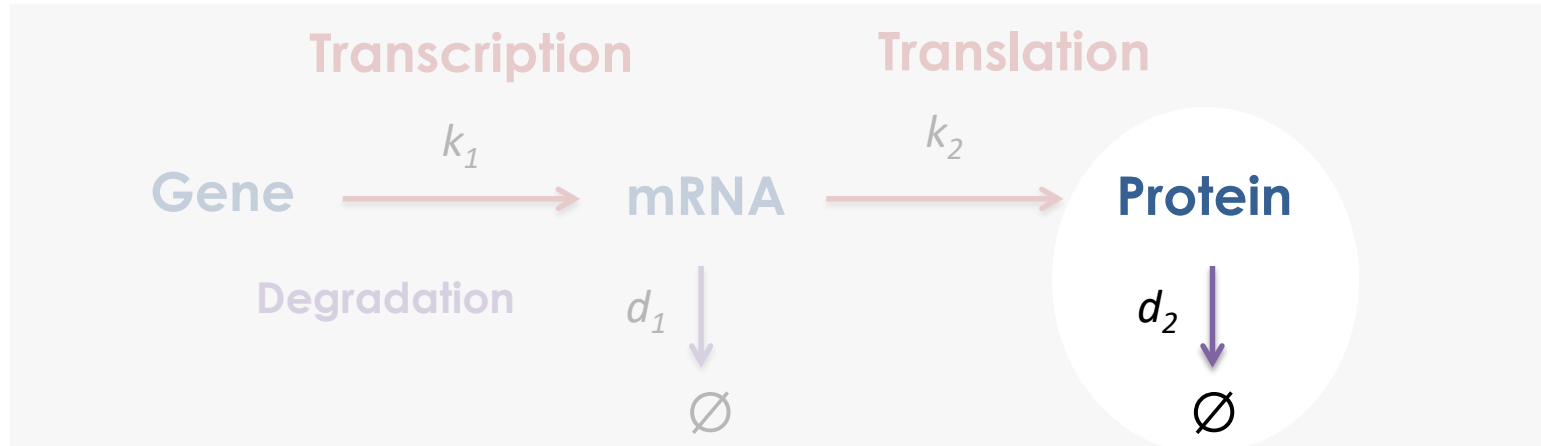
Constitutive gene expression (Simplified version)



$$\dot{[\text{mRNA}]} = k_1 [\text{Gene}] - d_1 [\text{mRNA}]$$

$$\dot{[\text{Protein}]} = k_2 [\text{mRNA}]$$

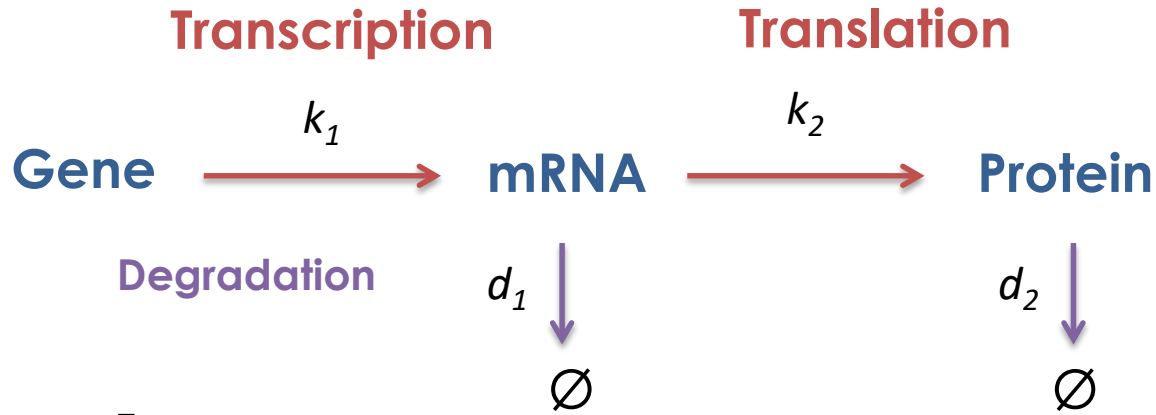
Constitutive gene expression (Simplified version)



$$\dot{[\text{mRNA}]} = k_1 [\text{Gene}] - d_1 [\text{mRNA}]$$

$$\dot{[\text{Protein}]} = k_2 [\text{mRNA}] - d_2 [\text{Protein}]$$

Constitutive gene expression (Simplified version)



$$\frac{d[\text{mRNA}]}{dt} = \dot{[\text{mRNA}]} = k_1[\text{Gene}] - d_1[\text{mRNA}]$$

$$\frac{d[\text{Protein}]}{dt} = \dot{[\text{Protein}]} = k_2[\text{mRNA}] - d_2[\text{Protein}]$$

Constitutive gene expression - Simulation

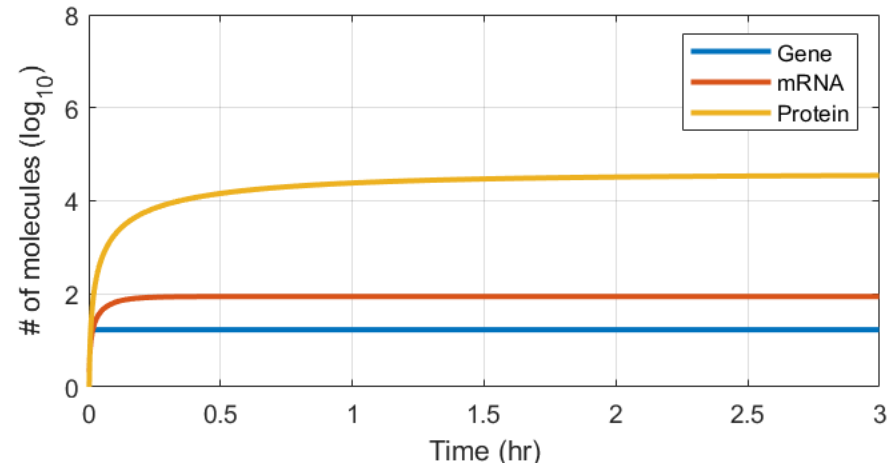
ODE model

$$[mRNA] = k_1[Gene] - d_1[mRNA]$$

$$[Protein] = k_2[mRNA] - d_2[Protein]$$

Function defines the ODE model

```
% Constitutive gene expression model.
% Updated 17/06/2020 Alejandro Vignoni
function [dxdt] = model_const(t,x,p)
%x1 = mRNA
dxdt(1,1) =p.CN*p.k1-p.d1*x(1);
%x2 = Protein
dxdt(2,1)=p.k2*x(1)-p.d2*x(2);
end
```



Main_const.m

Constitutive gene expression - Simulation

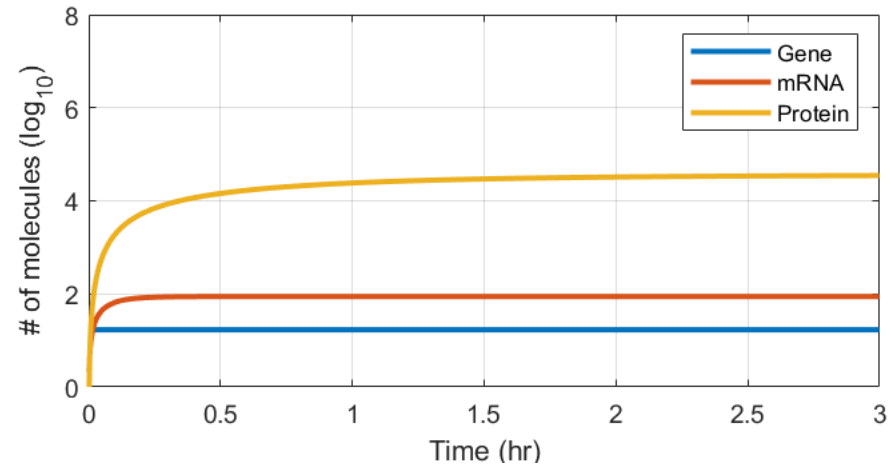
Parameters definition

```
%Parameters
p.CN = 17;           % plasmid number pACYC184 (17 copies/cell)
p.d1 = log(2)/3;     % mRNA degradation rate [1/min]
p.d2 = 0.02;         % degradation rate [1/min]
p.k2 = 8.23;         % translation rate [1/min]
p.k1 = 1.19;         % transcription rate [1/min]
```

Simulation configuration and execution

```
tfin = 60*3;        %simulation final time
step = 0.1;         %simulation step
tspan = 0:step:tfin-step;
% options for ode function
opti = odeset('AbsTol',1e-8,'RelTol',1e-6);
Init = [0 0];       %initial conditions

[t0,x0] = ode23t(@(t,x) model_const(t,x,p),tspan, Init, opti);
```



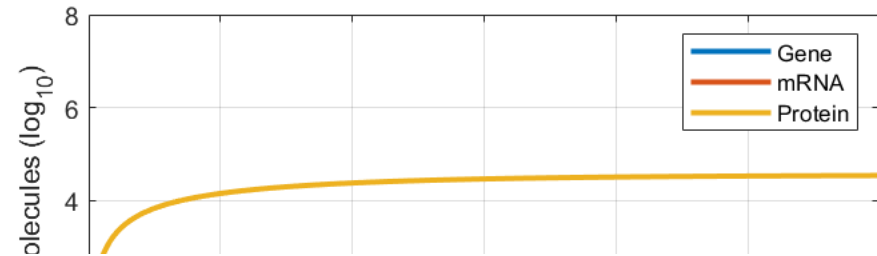
Constitutive gene expression - Simulation

Parameters definition

```
%Parameters
p.CN = 17;           % plasmid number pACYC184 (17 copies/cell)
p.d1 = log(2)/3;     % mRNA degradation rate [1/min]
p.d2 = 0.02;         % degradation rate [1/min]
p.k2 = 8.23;         % translation rate [1/min]
p.k1 = 1.19;         % transcription rate [1/min]
```

Simulation configuration and execution

```
tfin = 60*3;         %simulation final time
step = 0.1;          %simulation step
```



★ Sponsor Webinar - MathWorks: Modeling and Analysis of Synthetic Biology Systems with SimBiology and MATLAB

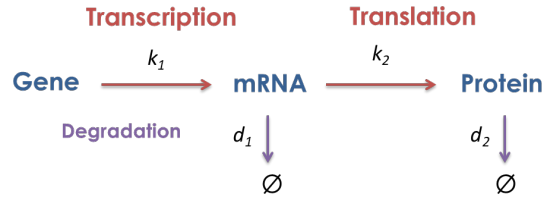
July

👤 Team: MathWorks

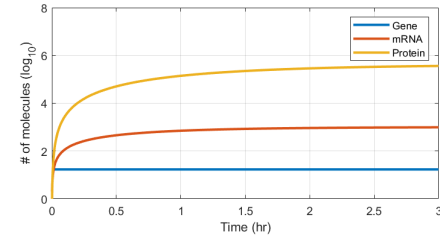
🕒 Date: July 23, 10:00AM EDT

📍 Location: Online

Constitutive gene expression - Remarks



$$\begin{aligned} \dot{[mRNA]} &= k_1 [Gene] - d_1 [mRNA] \\ \dot{[Protein]} &= k_2 [mRNA] - d_2 [Protein] \end{aligned}$$



[Gene] is considered a constant value and depends on: the Origin of Replication and the Plasmid Copy Number where the Gene is cloned.

We are considering:

- RNA polymerase and Ribosomes are sufficient enough so that they are not limiting the kinetics.
- Binding/Unbinding processes are much faster than transcription and translation.
- Protein degradation includes growth associated dilution.

Questions?

Ask writing in the chat or contact me
by email (alvig2 [at] upv [dot] es)

Stay tuned, next Section 2:

Derivation of a Hill function from
the law of mass action (15 min)

