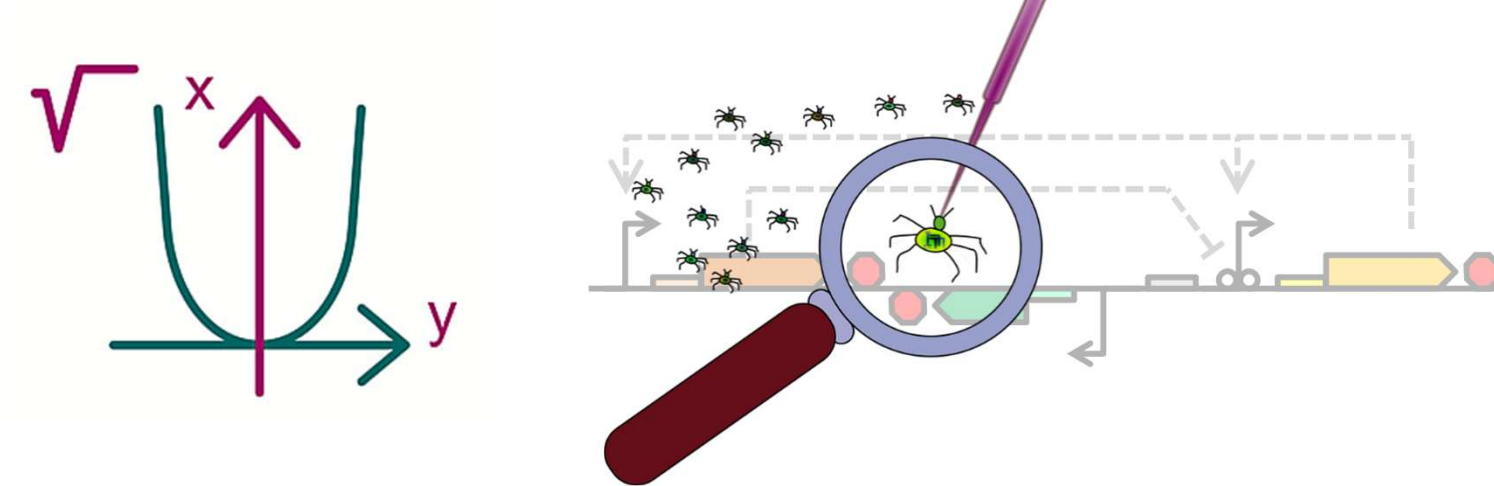


Debugging and Modelling of Genetic Circuits

UE2.1 Biological Parts and Devices

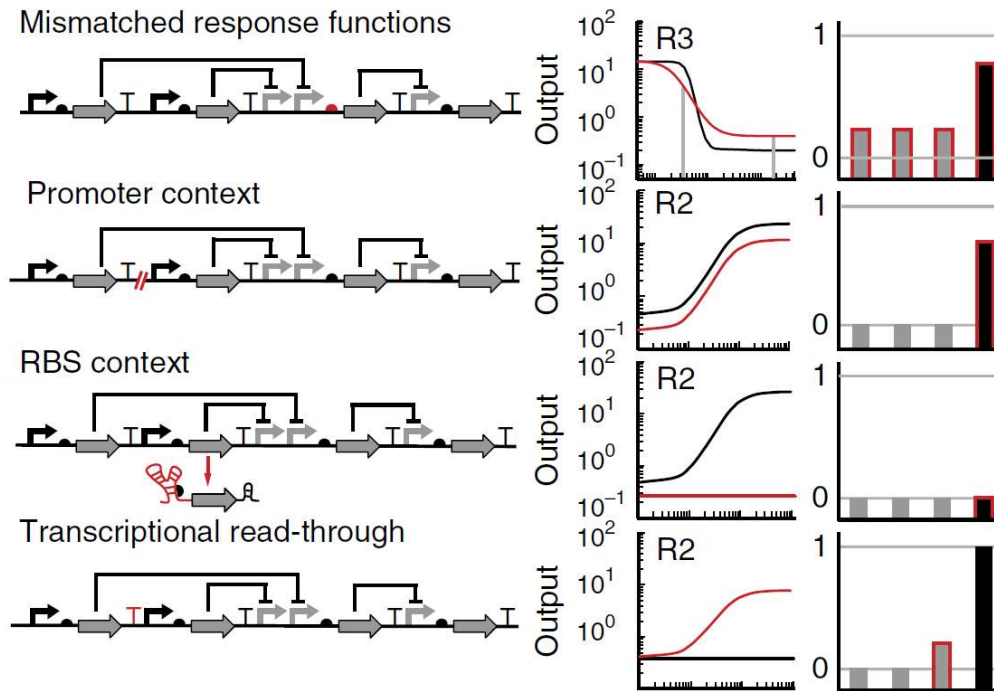
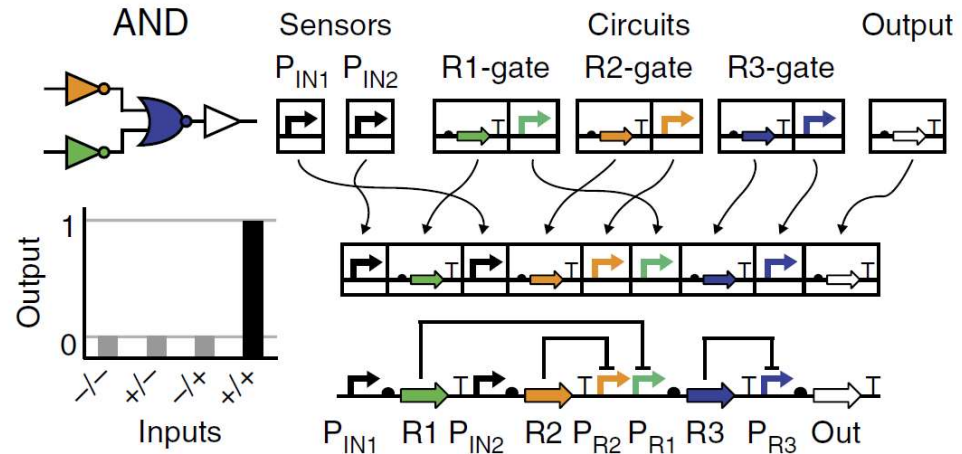


Manish Kushwaha

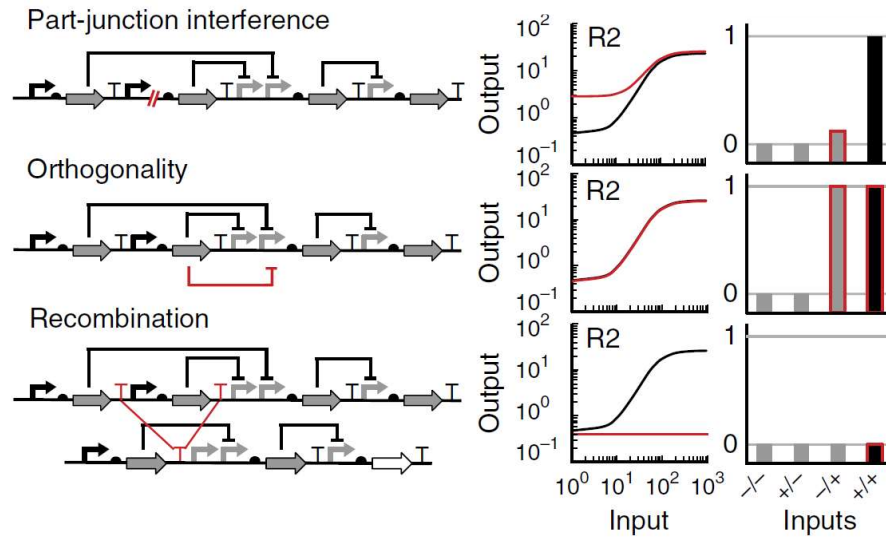
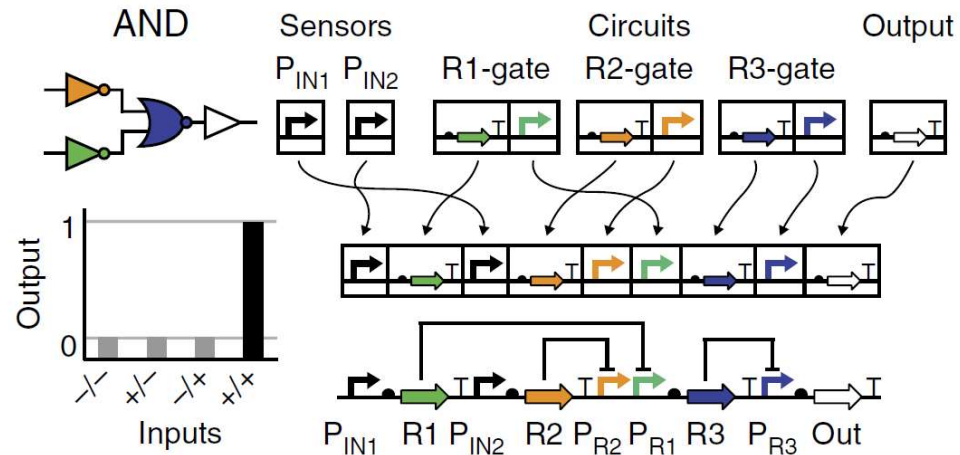
15 October, 2024

How do circuits break?

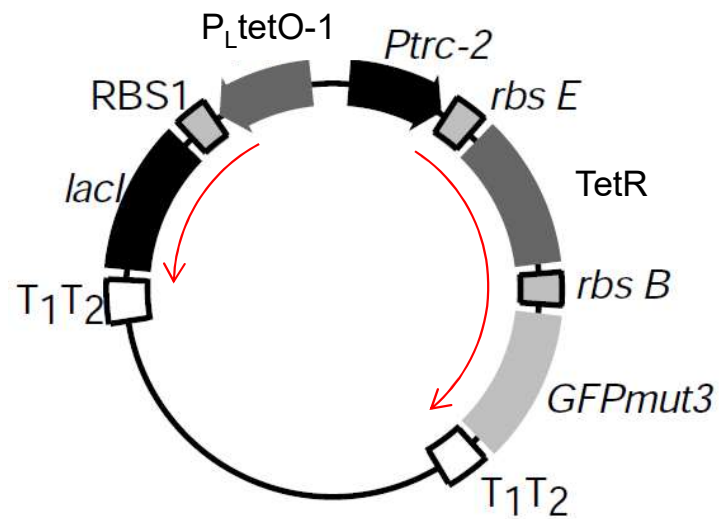
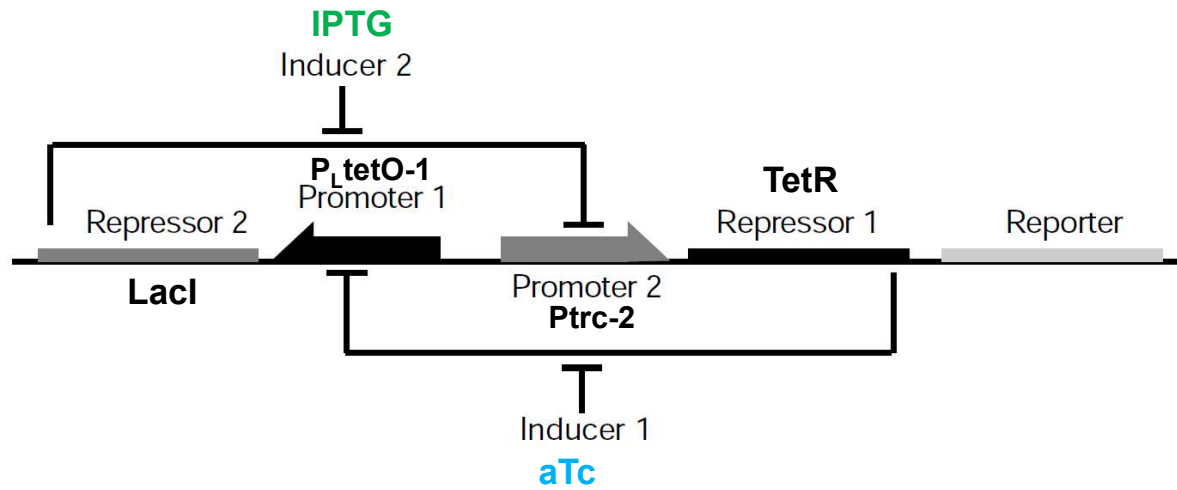
How do circuits break?



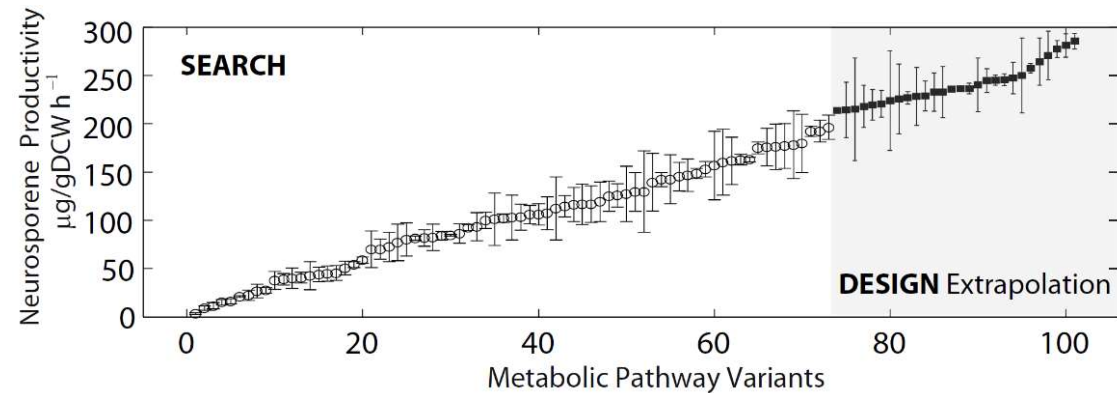
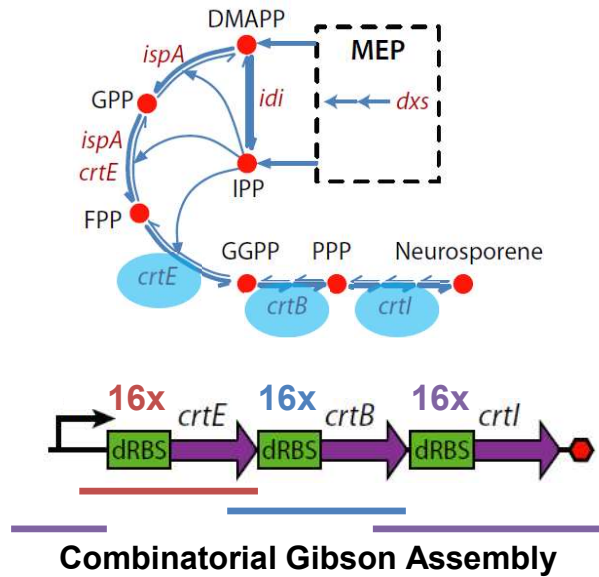
How do circuits break?



External control over components?



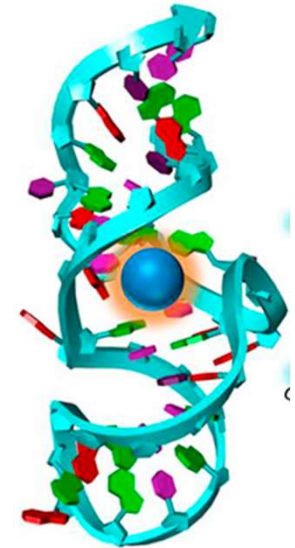
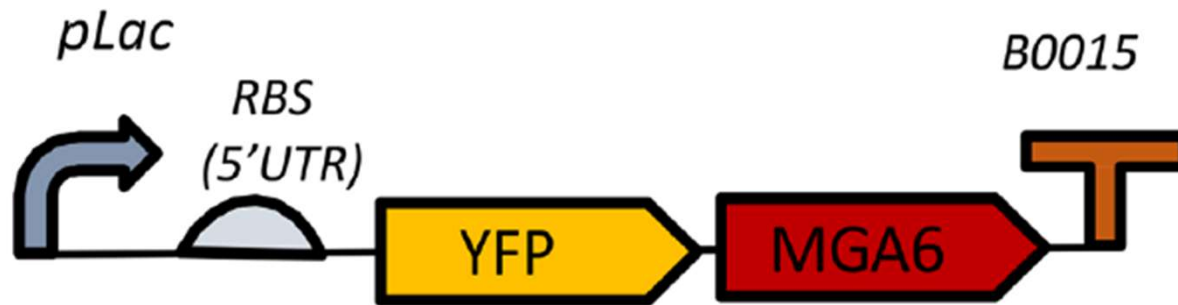
Diversity and Redundancy in Design?



- RBS (or promoter) libraries with several expression levels

Directly visualize the components?

- Visualise transcribed RNA directly: malachite green stabilization by RNA aptamer

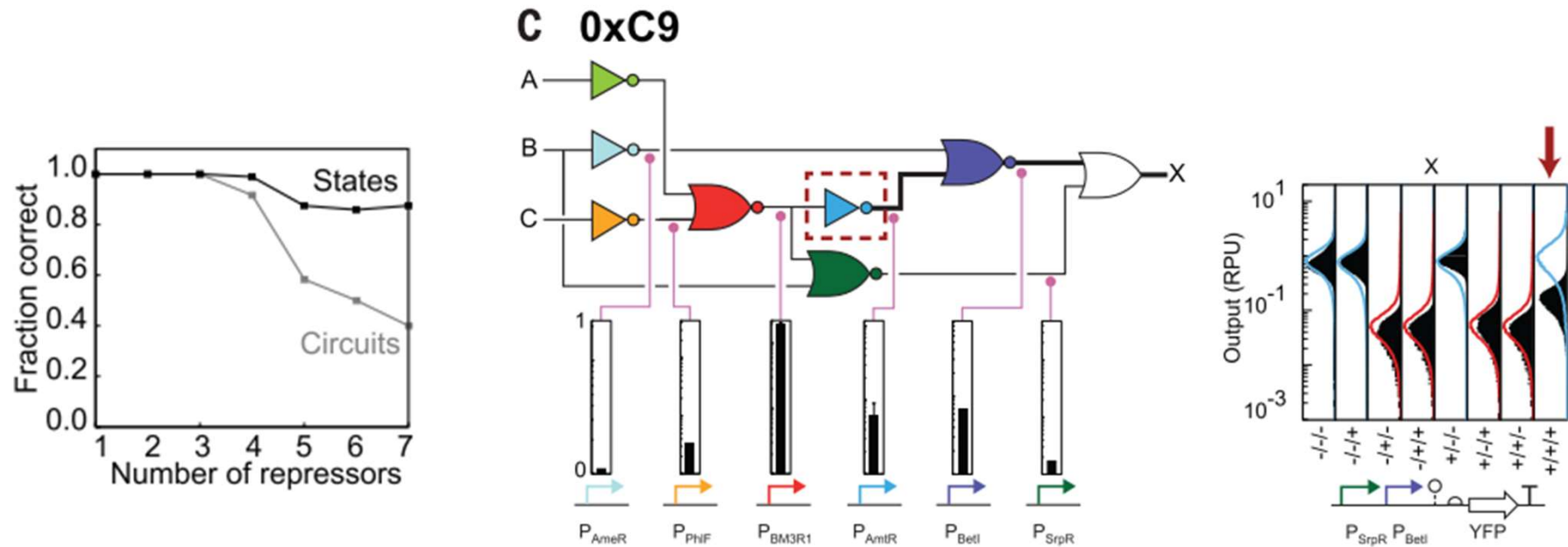


Aptamer binding

- Visualise translated protein: YFP measurement

Circuit failures: Under the hood

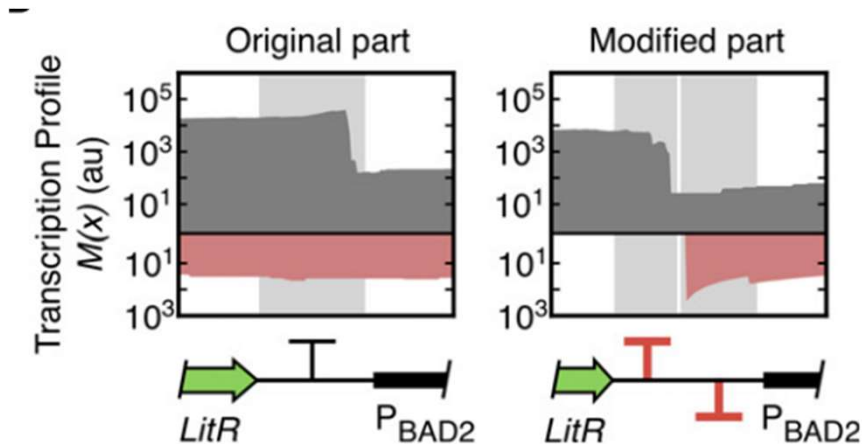
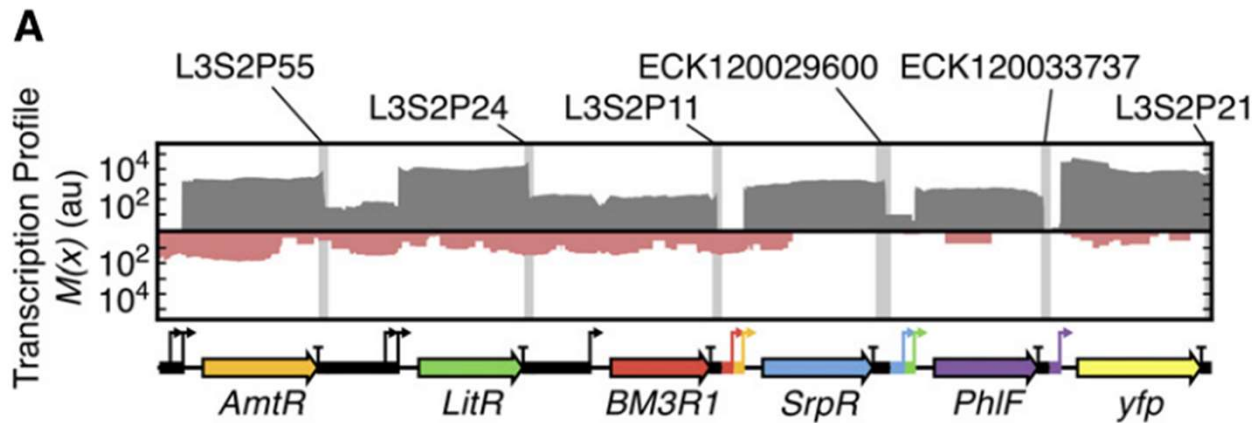
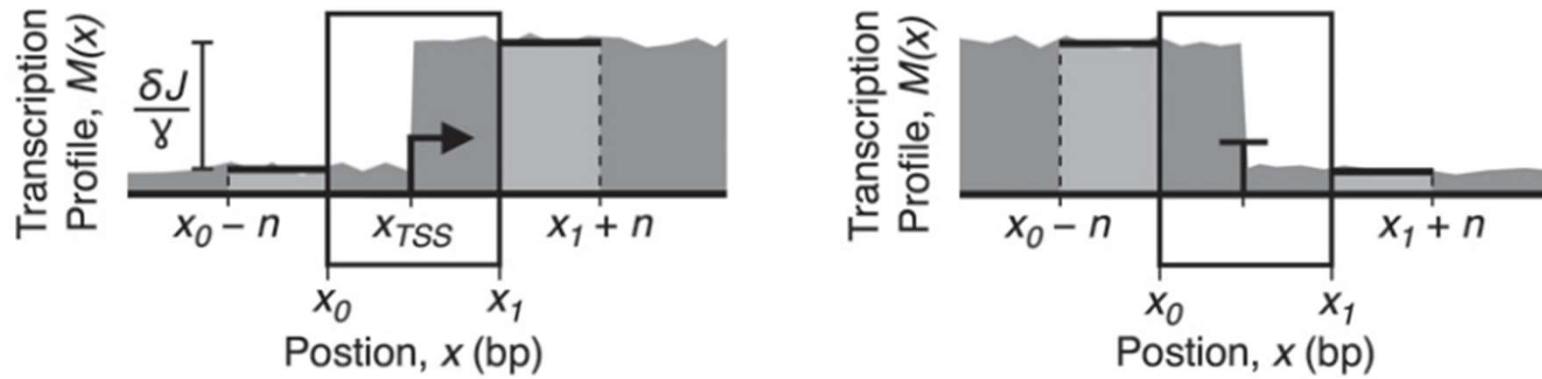
- CELLO: Increasing number of parts increases chances of errors



- Fluorescent protein fusions can be used to analyse each step separately
- From this analysis, most of the circuit failures point to unexpected behavior from the anhydrotetracycline (aTc) sensor (seven circuits) or AmtR gate (two circuits).

Debugging broken circuits

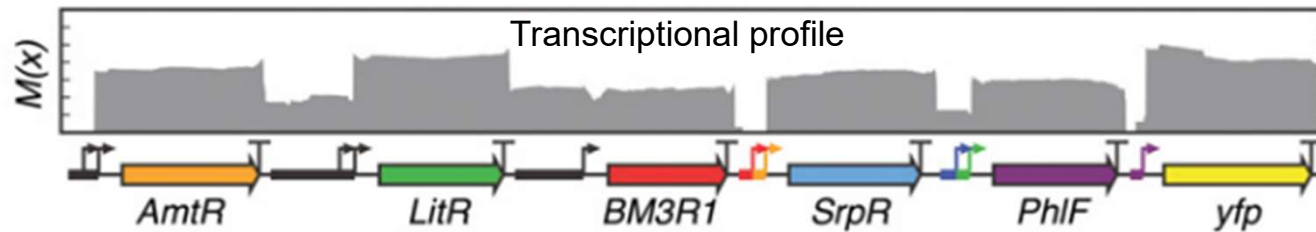
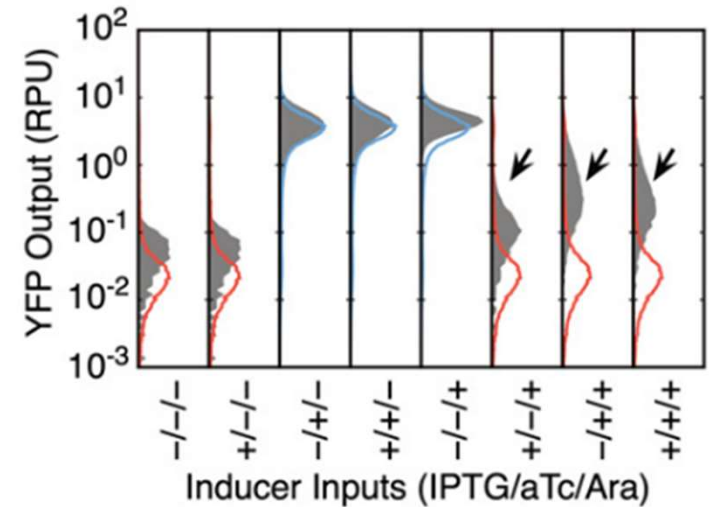
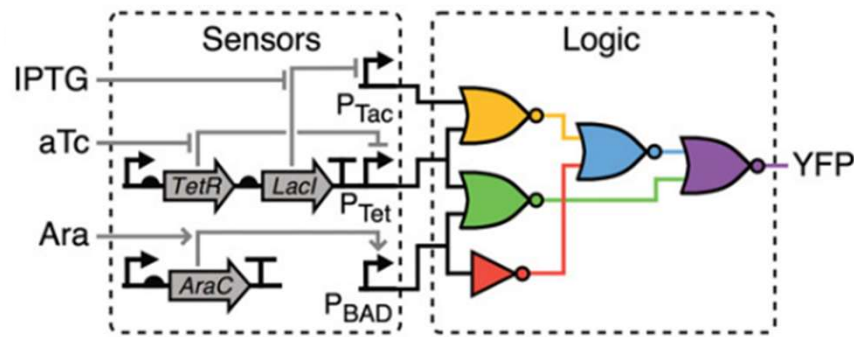
- Direct analysis of transcription states by RNAseq



(Goroehowski et al., 2017. Mol Sys Biol)

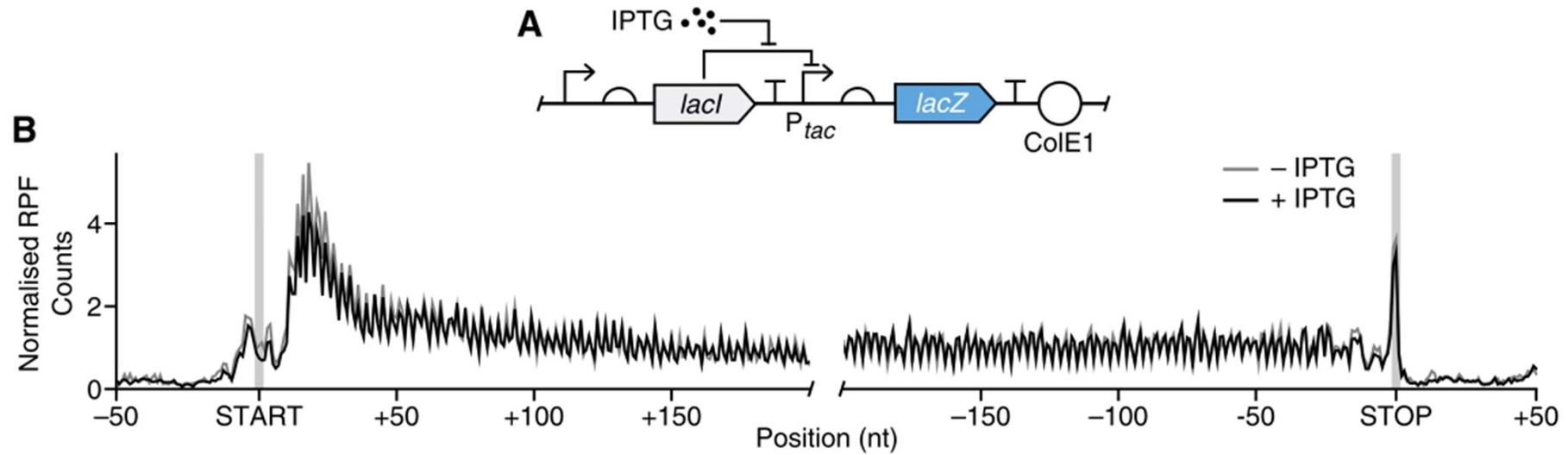
Debugging broken circuits

Inputs			Output
A	B	C	X
0	0	0	0
1	0	0	0
0	1	0	1
1	1	0	1
0	0	1	1
1	0	1	0
0	1	1	0
1	1	1	0



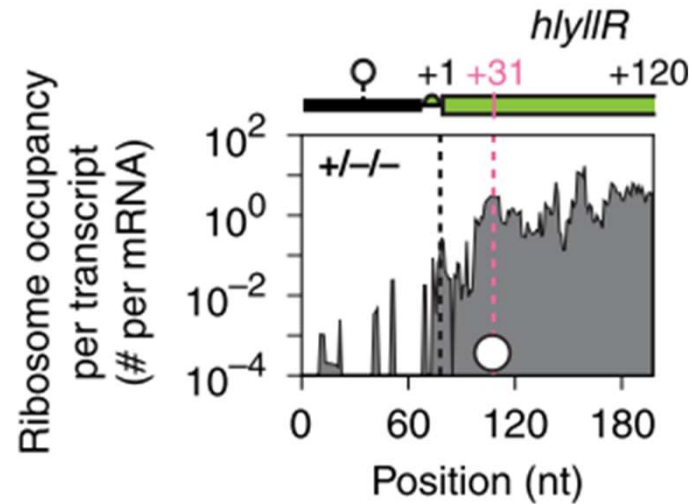
- Transcriptional profiling revealed a number of unexpected reasons of failure:
 - (1) cryptic antisense promoters
 - (2) terminator failure
 - (3) sensor malfunction due to media-induced changes in host gene expression

Debugging broken circuits



- Translation profiling reveals ribosome occupancy

Debugging broken circuits



- Internal start codon can affect genetic circuits
- Eliminating internal translation start site can fix the problem

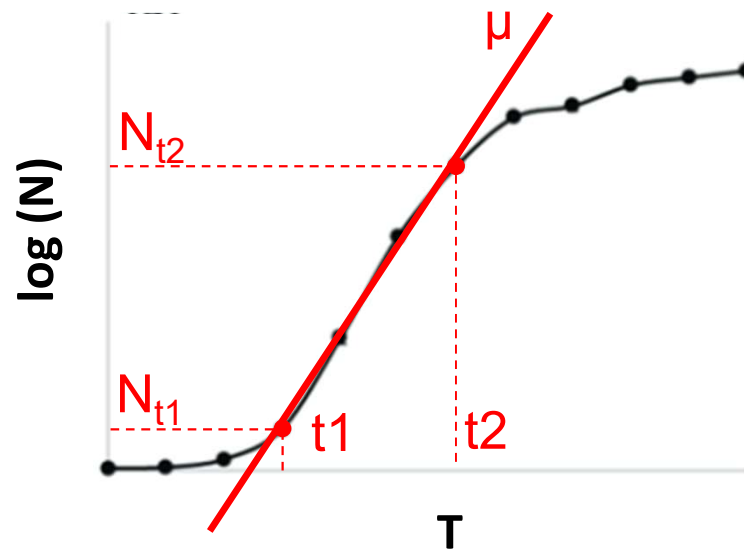
What is a model?

What is a model?

- Some informative/ conceptual representation of a system
- One or more equations that describe the relationship between different components/ properties of the system
- For example, the Exponential Growth Model:

$$N_{t2} = N_{t1} * (2^g)$$

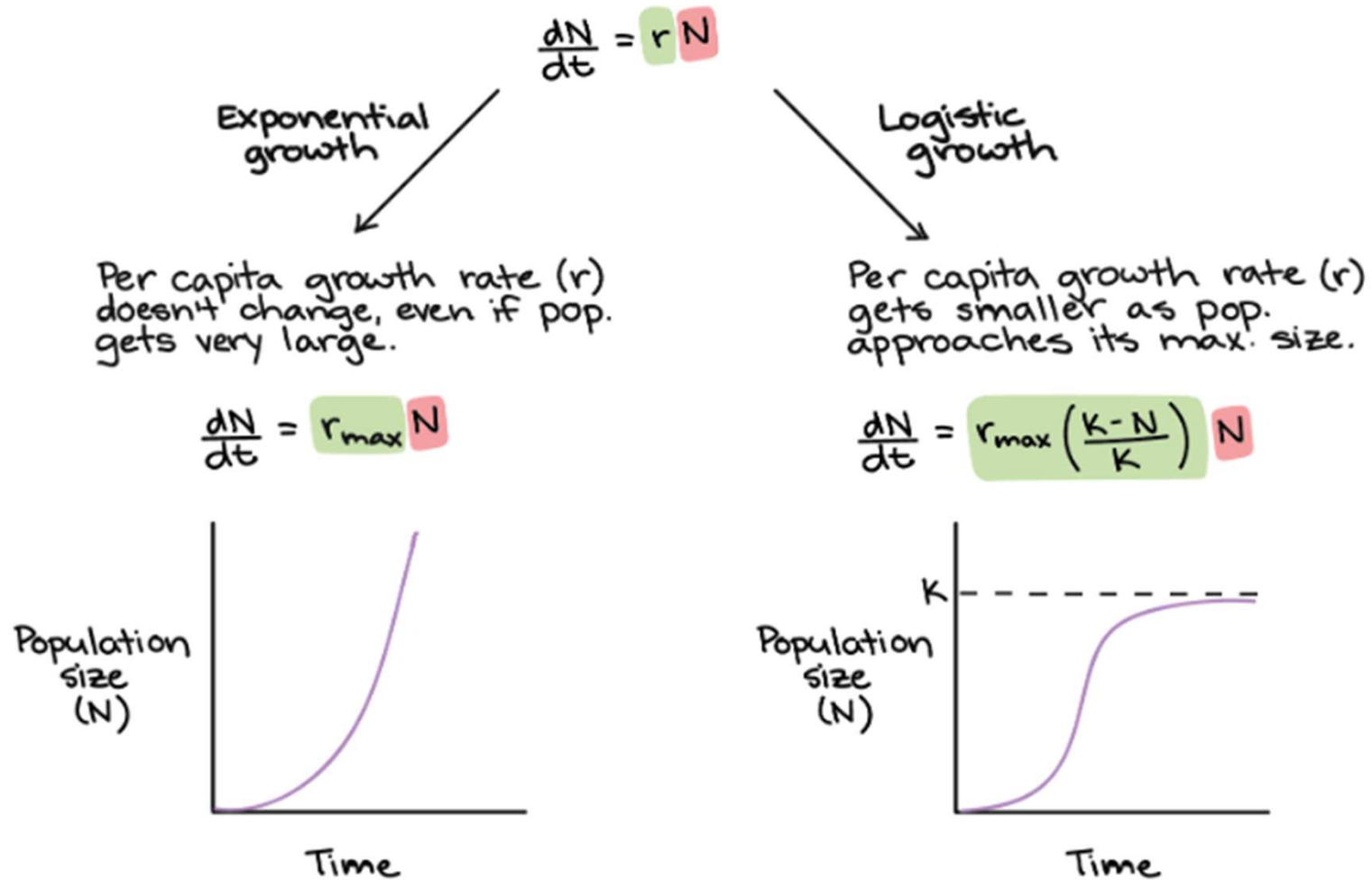
$$\mu = \log (N_{t2} / N_{t1}) / (t2 - t1)$$



$$[r = \mu]$$

Growth model: boundless or bounded

- The Exponential Growth Model assumes **infinite resources**.
- The Logistic Growth Model defines a **fixed carrying capacity**.



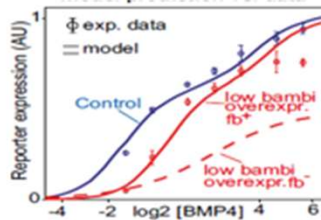
Biological Modelling

Using models for experimental design & refining models based on data

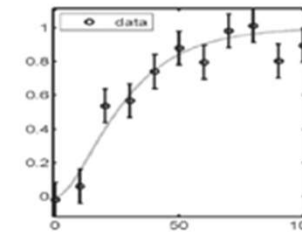
Model formulation



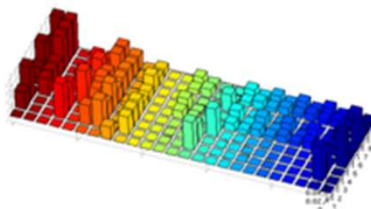
Experimental validation



Modeling & Simulation



Model analysis



Which network nodes are sensitive to perturbations?

Setting up your computer for simple modelling

- Essential: python
- Useful: jupyter Notebook (local) / Google Collab (cloud)

Some installation / help links:

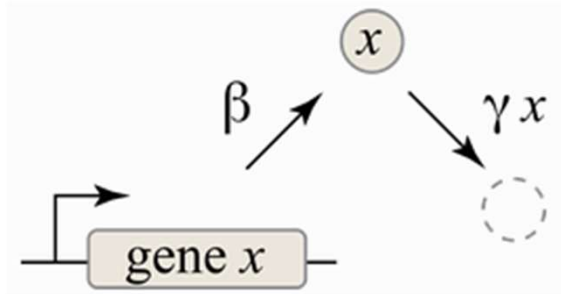
- Introduction to Python
https://biocircuits.github.io/appendices/appB_python/index.html
- Python + Jupyter on Linux WSL in Windows
<https://harshityadav95.medium.com/jupyter-notebook-in-windows-subsystem-for-linux-wsl-8b46fdf0a536>
- Jupyter Notebook Complete Beginner Guide 2023
https://www.youtube.com/watch?v=5pf0_bpNbkw
- Python code in your **Google Drive / Colaboratory**
<https://www.freecodecamp.org/news/google-colaboratory-python-code-in-your-google-drive/>



<https://tinyurl.com/mssb24colab>

Expression Modelling

- A system of Ordinary Differential Equations (ODEs) to model gene expression



$$\frac{dx}{dt} = \beta - \gamma x$$

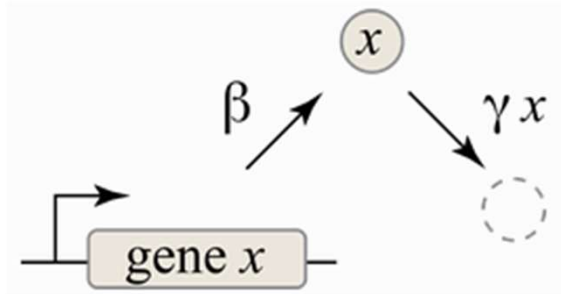
$$\gamma = \gamma_{\text{dilution}} + \gamma_{\text{degradation}}$$

Production (β) and Degradation (γ) rates

- Some assumptions: well mixed, real concentrations of molecules, first-order decay

Expression Modelling

- A system of Ordinary Differential Equations (ODEs) to model gene expression



$$\frac{dx}{dt} = \beta - \gamma x$$

$$\gamma = \gamma_{\text{dilution}} + \gamma_{\text{degradation}}$$

Production (β) and Degradation (γ) rates

- Some assumptions: well mixed, real concentrations of molecules, first-order decay

$$\frac{dx}{dt} = \beta - \gamma x = 0$$

$$\Rightarrow x_{\text{ss}} = \beta/\gamma$$

- Steady-state concentration can be determined as a **ratio of production and degradation rates**

ChatGPT: your coding / learning assistant?



Write the following ODE as python code:



$$dx/dt = \beta - \gamma x$$



You can use Python to solve the given ordinary differential equation (ODE) using numerical methods such as Euler's method. Here's a Python code snippet to solve this ODE:



python

Copy code

```
import numpy as np
import matplotlib.pyplot as plt

# Define the ODE parameters
beta = 0.2 # β
gamma = 0.1 # γ

# Define the ODE function
def ode(x, t):
    dxdt = beta - gamma * x
    return dxdt

# Initial condition
x0 = 0.0
```

ChatGPT: your coding / learning assistant?

```
python Copy code  
  
import numpy as np  
import matplotlib.pyplot as plt  
  
# Define the ODE parameters  
beta = 0.2 #  $\beta$   
gamma = 0.1 #  $\gamma$   
  
# Define the ODE function  
def ode(x, t):  
    dxdt = beta - gamma * x  
    return dxdt  
  
# Initial condition  
x0 = 0.0  
  
# Time span  
t = np.linspace(0, 10, 100) # You can adjust the time span as needed  
  
# Solve the ODE using Euler's method  
x = np.zeros(len(t))  
x[0] = x0  
  
for i in range(1, len(t)):  
    delta_t = t[i] - t[i-1]  
    x[i] = x[i-1] + ode(x[i-1], t[i-1]) * delta_t  
  
# Plot the solution  
plt.figure()  
plt.plot(t, x)  
plt.xlabel('Time (t)')  
plt.ylabel('x(t)')  
plt.title('Solution of dx/dt =  $\beta - \gamma x$ ')  
plt.grid(True)  
plt.show()
```

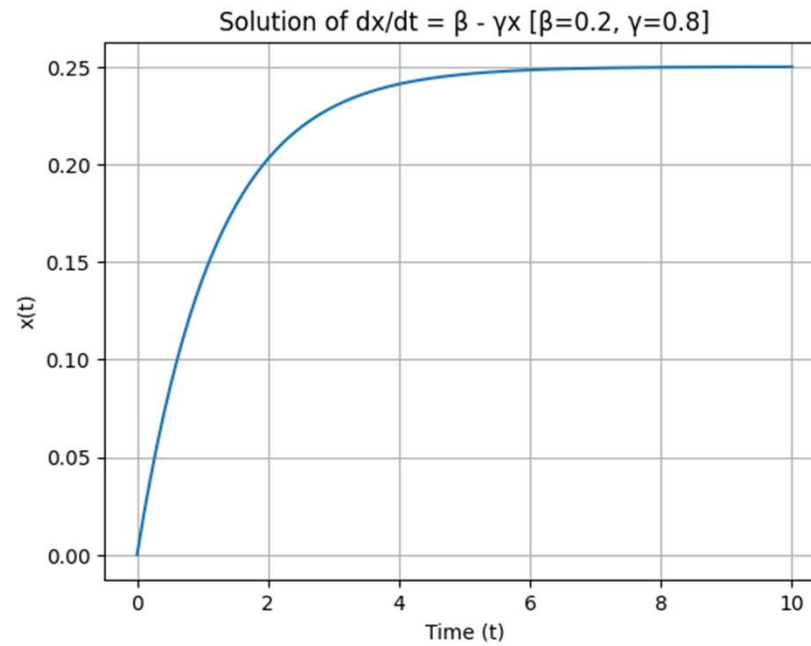
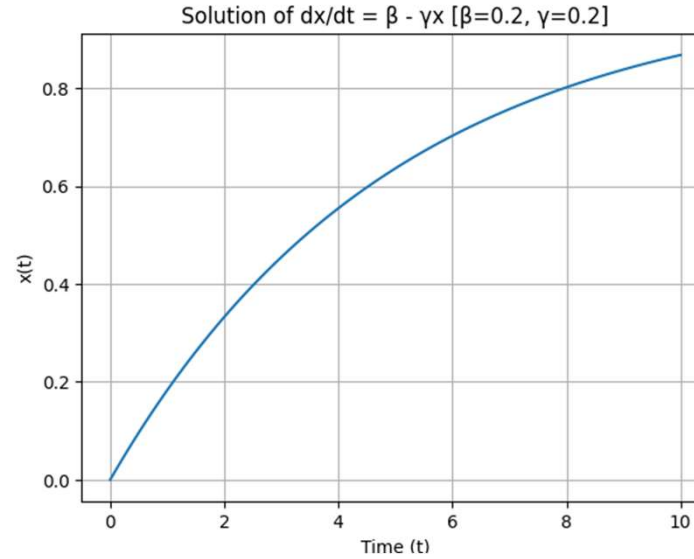
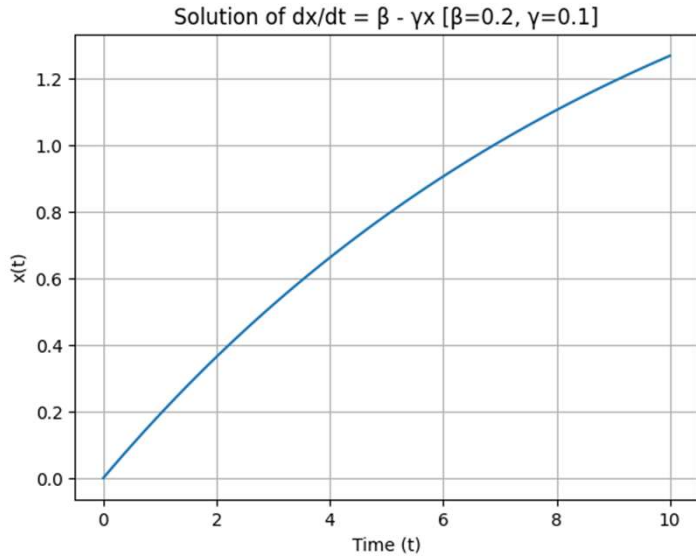
$$\frac{dx}{dt} = \beta - \gamma x$$

In this code, we first define the ODE function `ode(x, t)` as `dx/dt = $\beta - \gamma x$` . Then, we use Euler's method to numerically solve the ODE over a specified time span. You can adjust the values of `beta`, `gamma`, the initial condition `x0`, and the time span `t` to suit your specific problem. The code also includes a plot of the solution.

ChatGPT + Google Colab

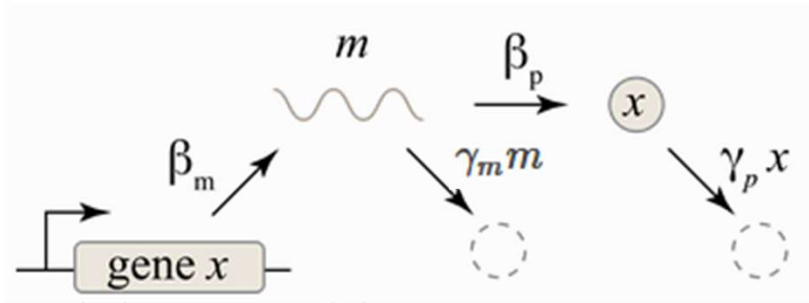
$$\frac{dx}{dt} = \beta - \gamma x$$

<https://tinyurl.com/mssb24colab>



Transcription-Translation Modelling

- A system of Ordinary Differential Equations (ODEs) to model transcription and translation steps



$$\frac{dm}{dt} = \beta_m - \gamma_m m$$
$$\frac{dx}{dt} = \beta_p m - \gamma_p x$$

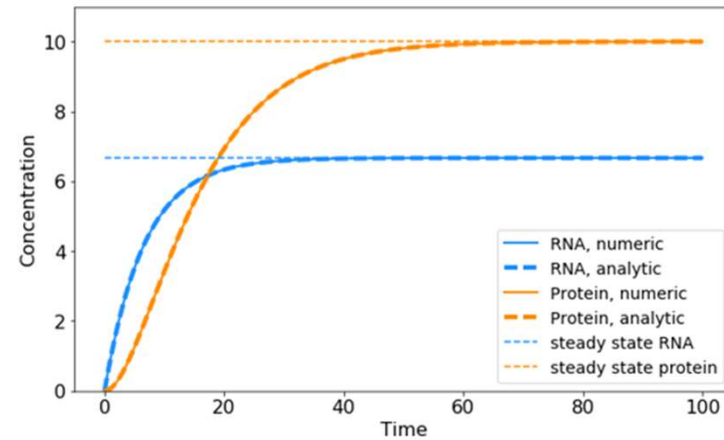
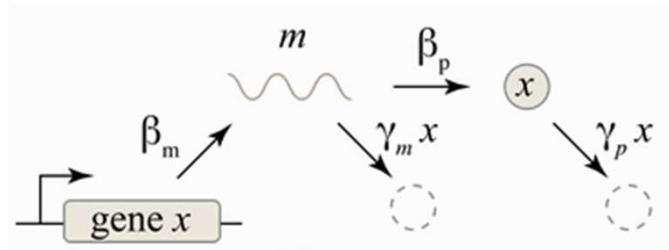
$$m_{ss} = \beta_m / \gamma_m$$

$$x_{ss} = \frac{\beta_p m_{ss}}{\gamma_p} = \frac{\beta_p \beta_m}{\gamma_p \gamma_m}$$

- Steady-state values

- Steady-state concentration can be determined as a ratio of production and degradation rates
- Transcription proportionally changes both mRNA and protein levels

Transcription-Translation Modelling



- Time course of mRNA and protein expression

$$\frac{dm}{dt} = \beta_m - \gamma_m m$$

$$\frac{dx}{dt} = \beta_p m - \gamma_p x$$

$$m_{ss} = \beta_m / \gamma_m$$

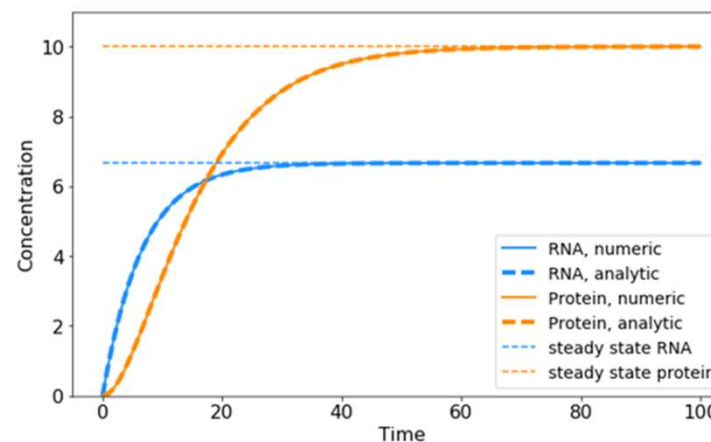
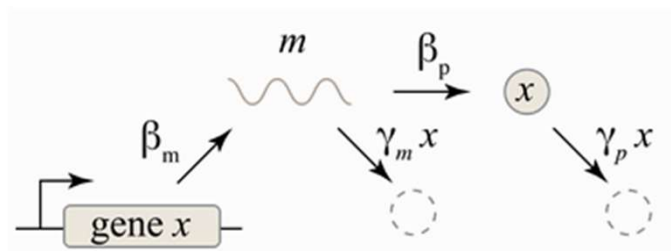
$$x_{ss} = \frac{\beta_p m_{ss}}{\gamma_p} = \frac{\beta_p \beta_m}{\gamma_p \gamma_m}$$

$$m(t) = m_{ss} \cdot (1 - e^{-\gamma_m t})$$

$$x(t) = x_{ss} \cdot \left(1 - \frac{\gamma_m e^{-\gamma_p t} - \gamma_p e^{-\gamma_m t}}{\gamma_m - \gamma_p}\right)$$

Instantaneous concentrations

Transcription-Translation Modelling



- Time course of mRNA and protein expression

$$\frac{dm}{dt} = \beta_m - \gamma_m m$$

$$\frac{dx}{dt} = \beta_p m - \gamma_p x$$

mRNA degradation rate (γ_m)

changes final steady state
and response time

$$m_{ss} = \beta_m / \gamma_m$$

$$x_{ss} = \frac{\beta_p m_{ss}}{\gamma_p} = \frac{\beta_p \beta_m}{\gamma_p \gamma_m}$$

protein degradation rate (γ_p)

changes final steady state
and response time

$$m(t) = m_{ss} \cdot (1 - e^{-\gamma_m t})$$

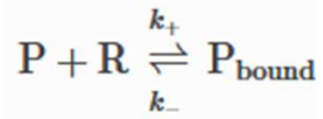
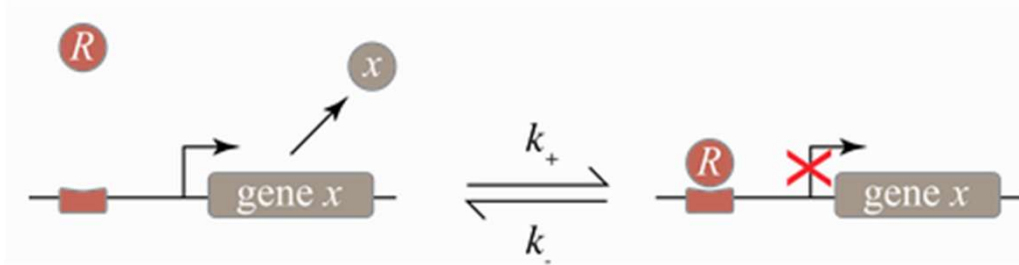
$$x(t) = x_{ss} \cdot \left(1 - \frac{\gamma_m e^{-\gamma_p t} - \gamma_p e^{-\gamma_m t}}{\gamma_m - \gamma_p}\right)$$

Instantaneous concentrations

mRNA and protein synthesis rates (β_m, β_p)

change only final steady state

Regulated Transcription: Repressor



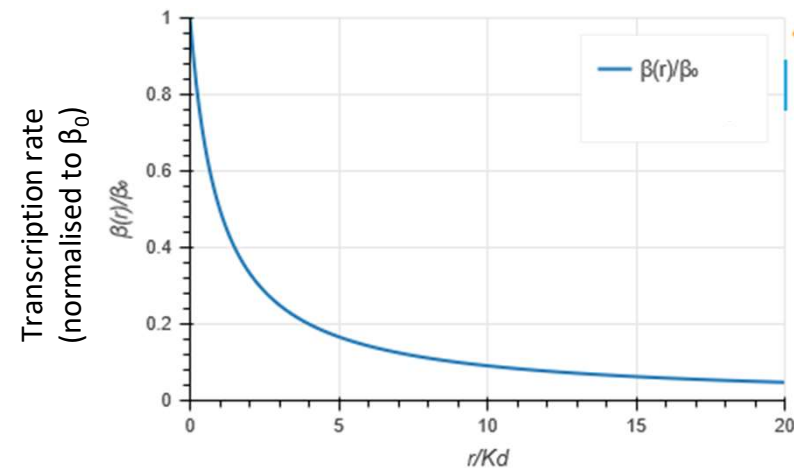
$$\frac{dp}{dt} = -k_+ p r + k_- p_{\text{bound}} \quad (p = \text{Free pr conc.})$$

$$K_d = \frac{k_-}{k_+}$$

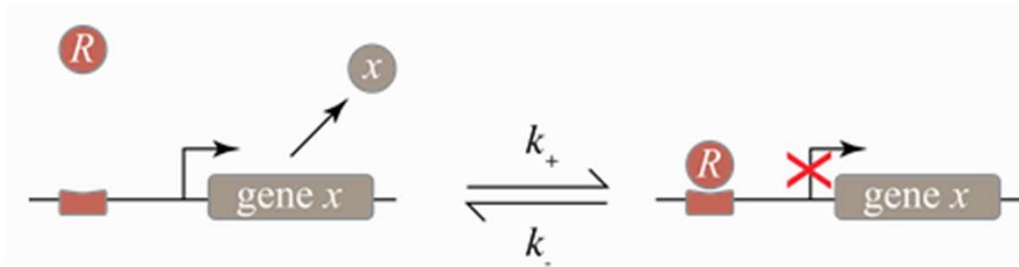
(Dissociation constant)

$$\beta(r) = \beta_0 \frac{p}{p_{\text{tot}}} = \frac{\beta_0}{1 + r/K_d}$$

- β_0 is the unregulated transcription rate
- Assumes separation of time-scales

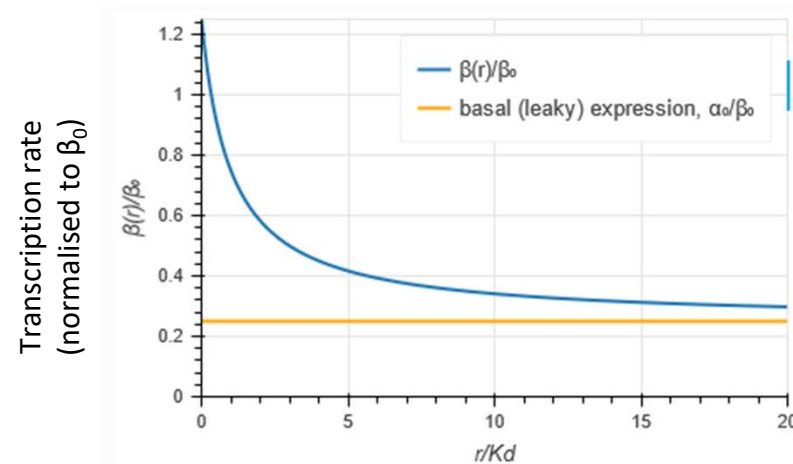


Regulated Transcription: Repressor

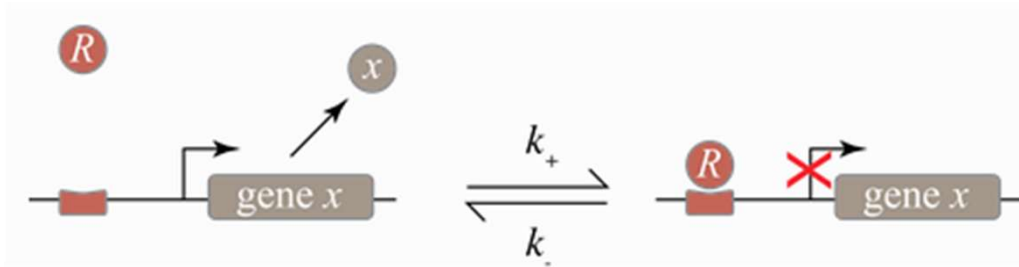


$$\beta(R) = \alpha_0 + \beta_0 \frac{p}{p_{\text{tot}}} = \alpha_0 + \frac{\beta_0}{1 + r/K_d}$$

(with leaky transcription)

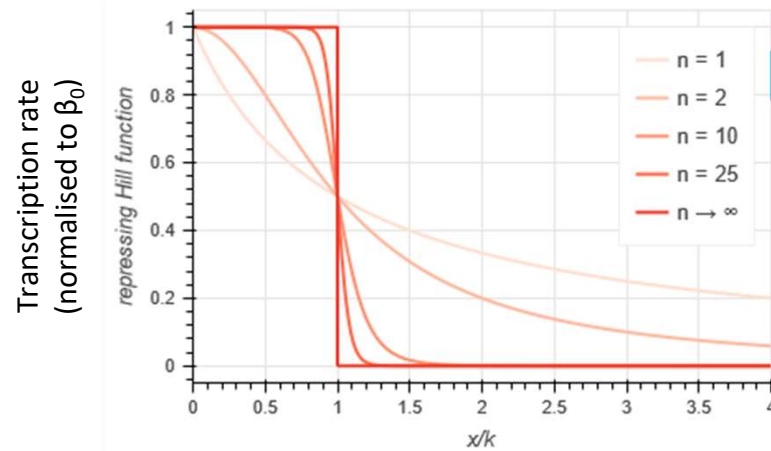


Regulated Transcription: Repressor

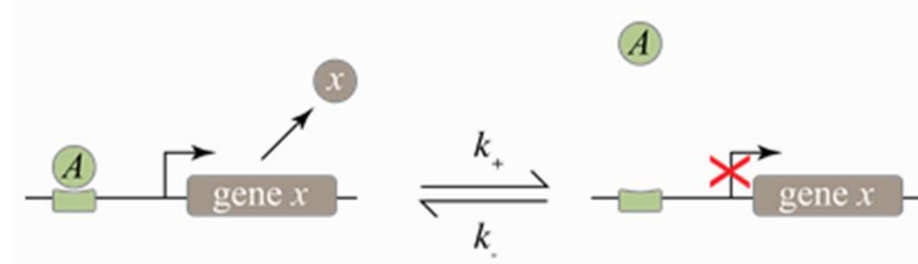


$$\beta(r) = \frac{\beta_0}{1 + (r/K_d)^n}$$

(Hill function kinetics)

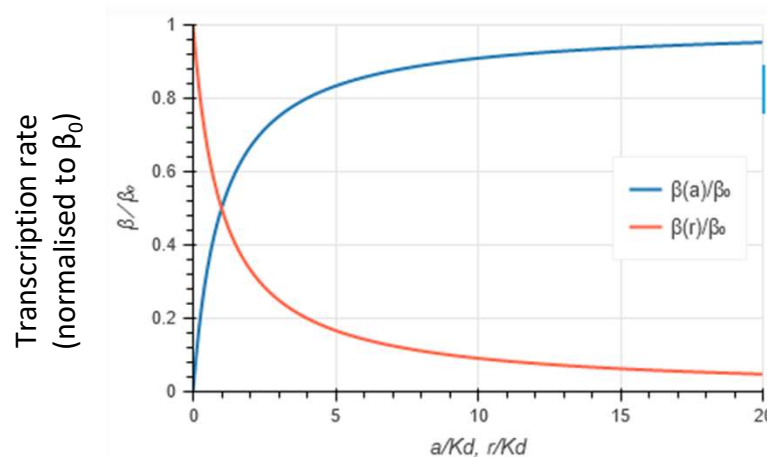


Regulated Transcription: Activator



$$\beta(a) = \beta_0 \frac{p_{\text{bound}}}{p_{\text{tot}}} = \beta_0 \frac{a/K_d}{1 + a/K_d}$$

(Activated expression)

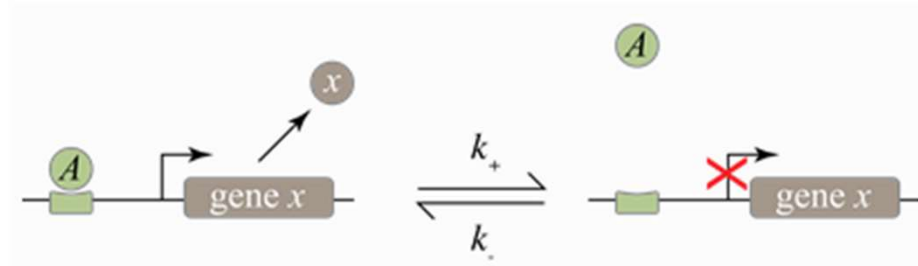


$$\beta(r) = \beta_0 \frac{p}{p_{\text{tot}}} = \frac{\beta_0}{1 + r/K_d}$$

(Repressed expression)

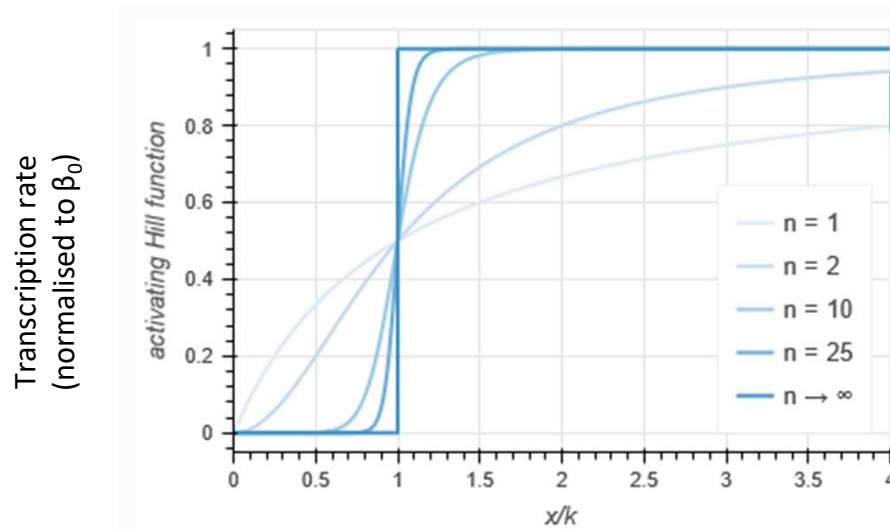
- β_0 is the unregulated max transcription rate
- Assumes separation of time-scales

Regulated Transcription: Activator

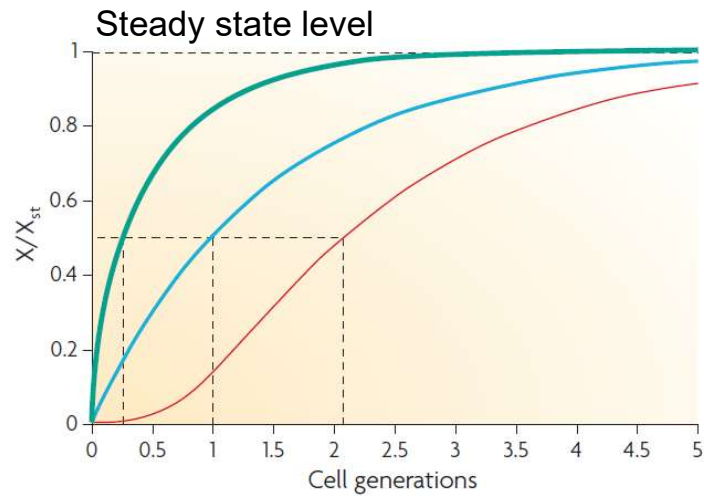


$$\beta(a) = \beta_0 \frac{(a/K_d)^n}{1 + (a/K_d)^n}$$

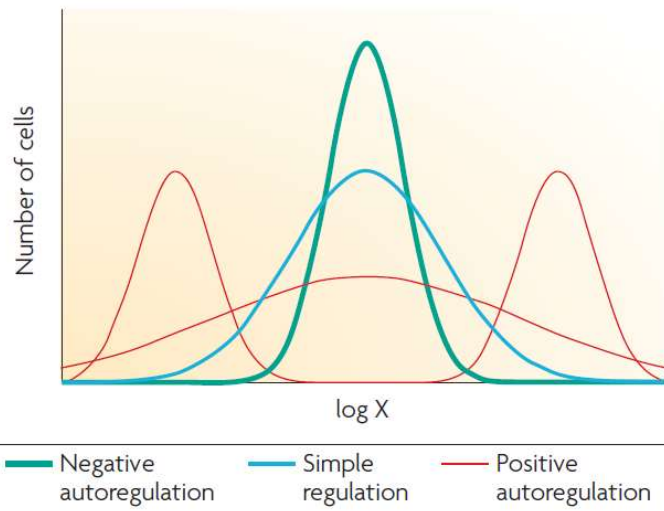
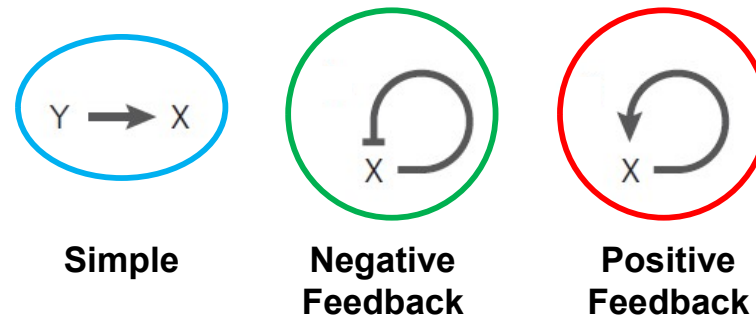
(Hill function kinetics)



Network Motifs modify circuit behaviour



NFL speeds up response time



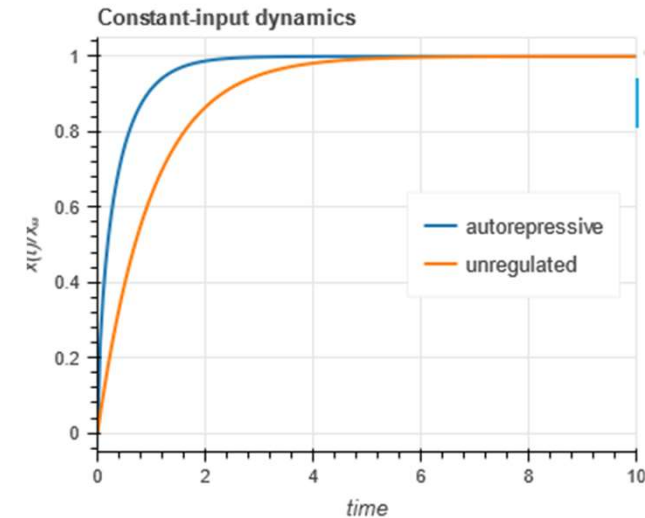
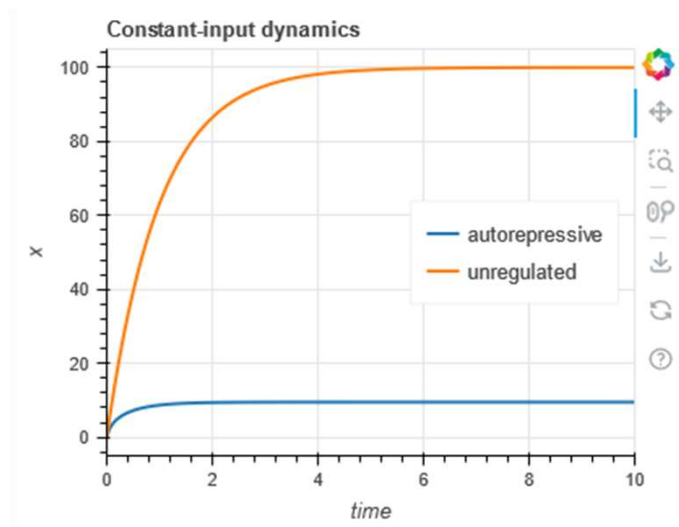
NFL reduces cell-to-cell variation

- NFL results in steeper rise in the concentration of protein X.
- NFL reduces noise

NFL speeds up response time

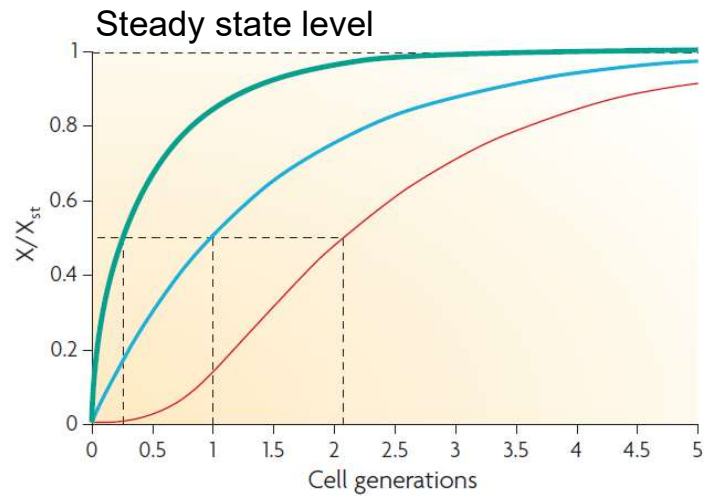
$$m(t) = m_{ss} \cdot (1 - e^{-\gamma_m t})$$
$$x(t) = x_{ss} \cdot \left(1 - \frac{\gamma_m \cdot e^{-\gamma_p t} - \gamma_p \cdot e^{-\gamma_m t}}{\gamma_m - \gamma_p}\right)$$

Instantaneous concentrations

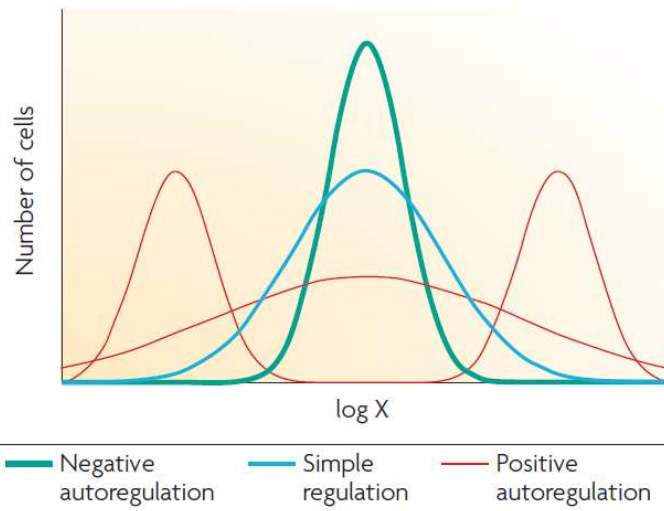
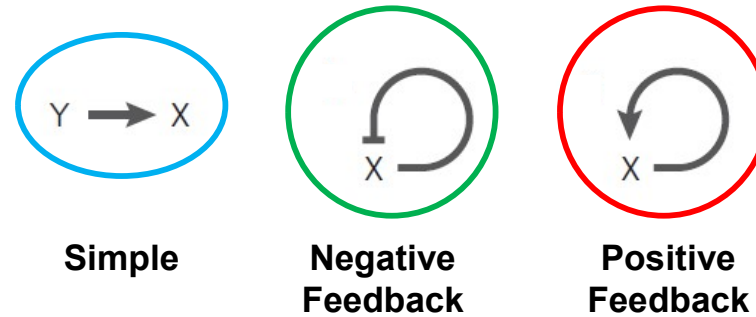


- NFL results in steeper rise in the concentration of protein X.

Network Motifs modify circuit behaviour



PFL slows down response time

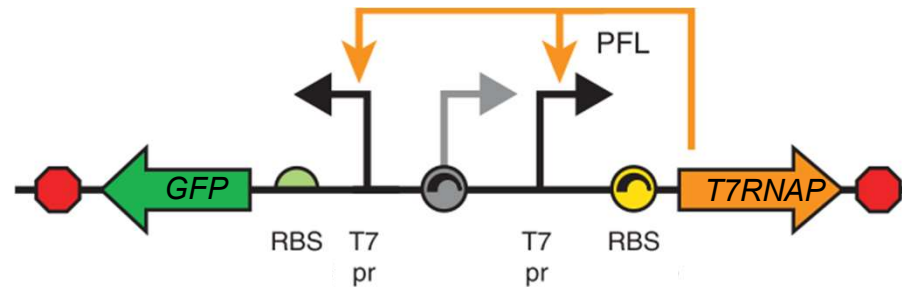


PFL increases cell-to-cell variation

- PFL results in slower rise in the concentration of protein X.
- PFL increases noise

Transcription-Translation Modelling with Reaction Mechanism

- In the previous examples the β_m , β_x , γ_m , γ_x are experimentally observed
- Accounting for DNA concentration ($CopyN$) and leaky transcriptions
- Model-calculated translation rate (β_x)



ODEs

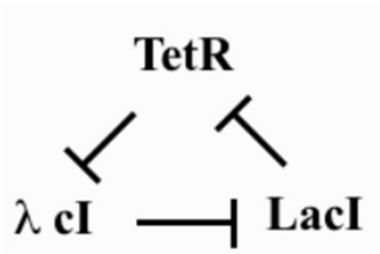
$$\frac{d[mRNA_{T7RNAP}]}{dt} = CopyN \cdot (primingR_{T7RNAP} + R_{T7prT7RNAP}) - \delta_{mRNA} \cdot [mRNA_{T7RNAP}] \quad (1)$$

$$\frac{d[T7RNAP]}{dt} = PR_{T7RNAP} \cdot [mRNA_{T7RNAP}] - \delta_{T7RNAP} \cdot [T7RNAP] \quad (2)$$

$$\frac{d[mRNA_{GFP}]}{dt} = CopyN \cdot (leakyR_{GFP} + R_{T7prGFP}) - \delta_{mRNA} \cdot [mRNA_{GFP}] \quad (3)$$

$$\frac{d[GFP]}{dt} = PR_{GFP} \cdot [mRNA_{GFP}] - \delta_{GFP} \cdot [GFP] \quad (4)$$

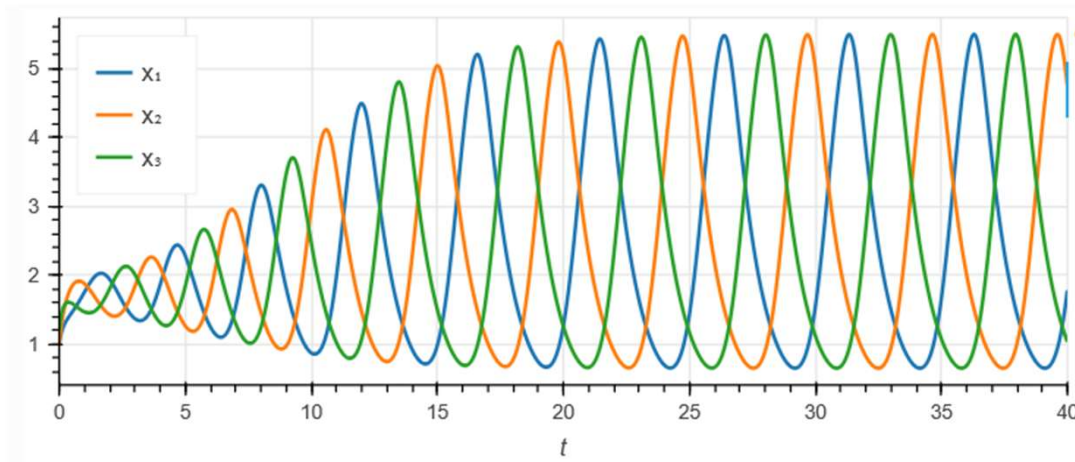
Expression Modelling: Repressilator



$$\frac{dx_1}{dt} = \frac{\beta}{1 + (x_3/k)^n} - \gamma x_1$$

$$\frac{dx_2}{dt} = \frac{\beta}{1 + (x_1/k)^n} - \gamma x_2$$

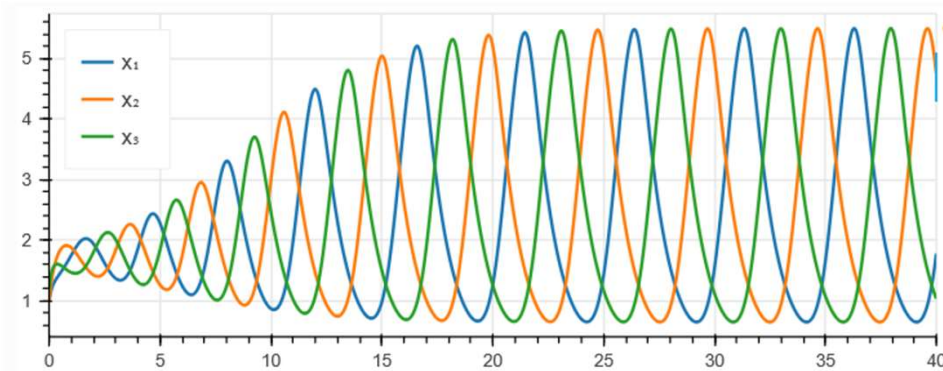
$$\frac{dx_3}{dt} = \frac{\beta}{1 + (x_2/k)^n} - \gamma x_3$$



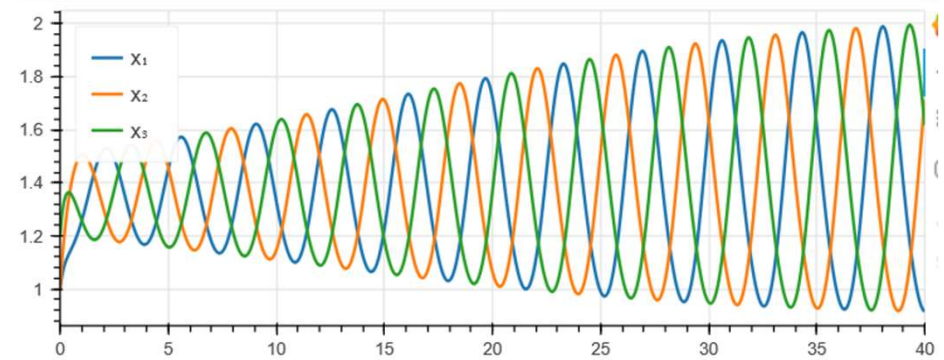
$$\beta = 10$$

$$n = 3$$

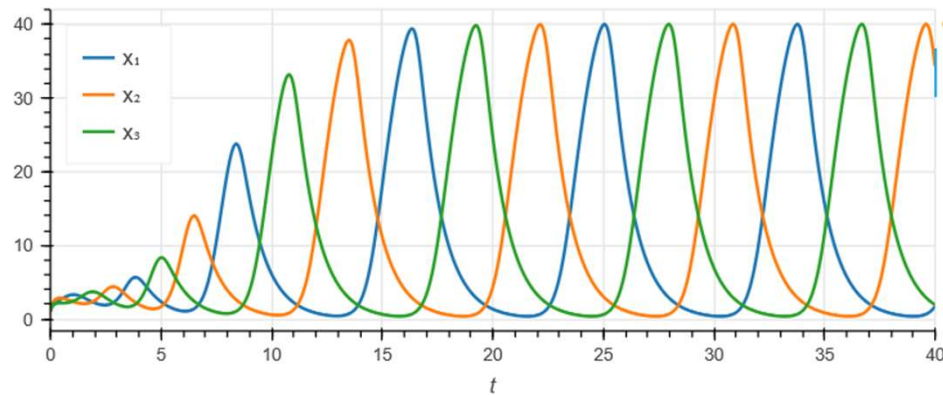
Expression Modelling: Repressilator



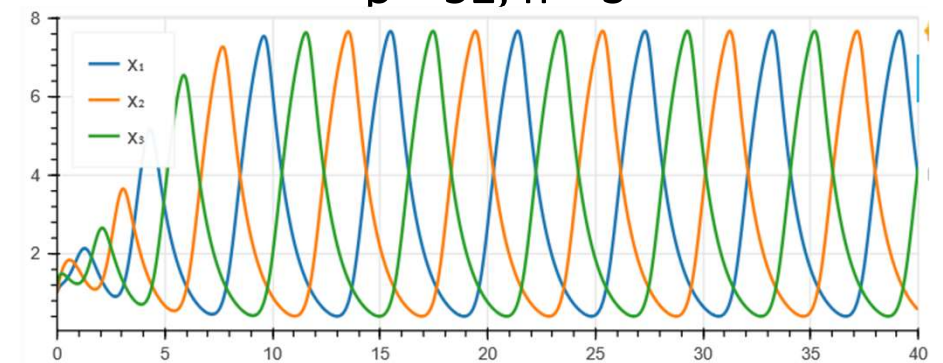
$\beta = 10, n = 3$



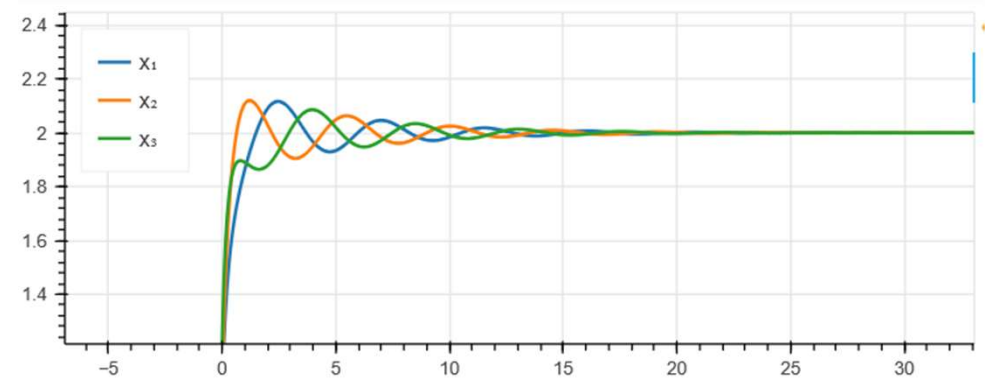
$\beta = 4.6, n = 3$



$\beta = 52, n = 3$



$\beta = 10, n = 4$

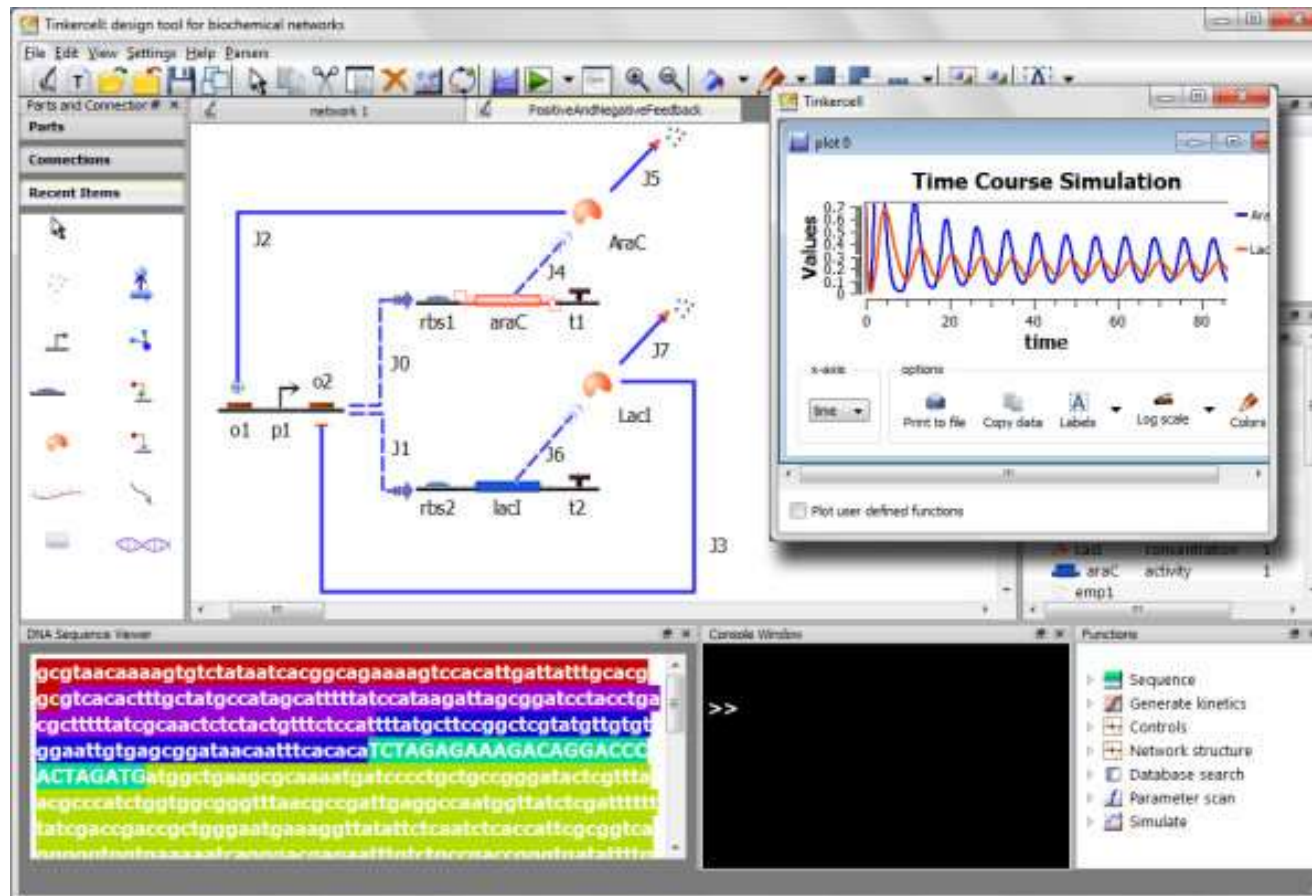


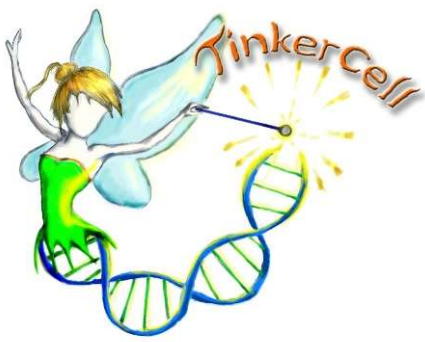
$\beta = 10, n = 2$



TinkerCell

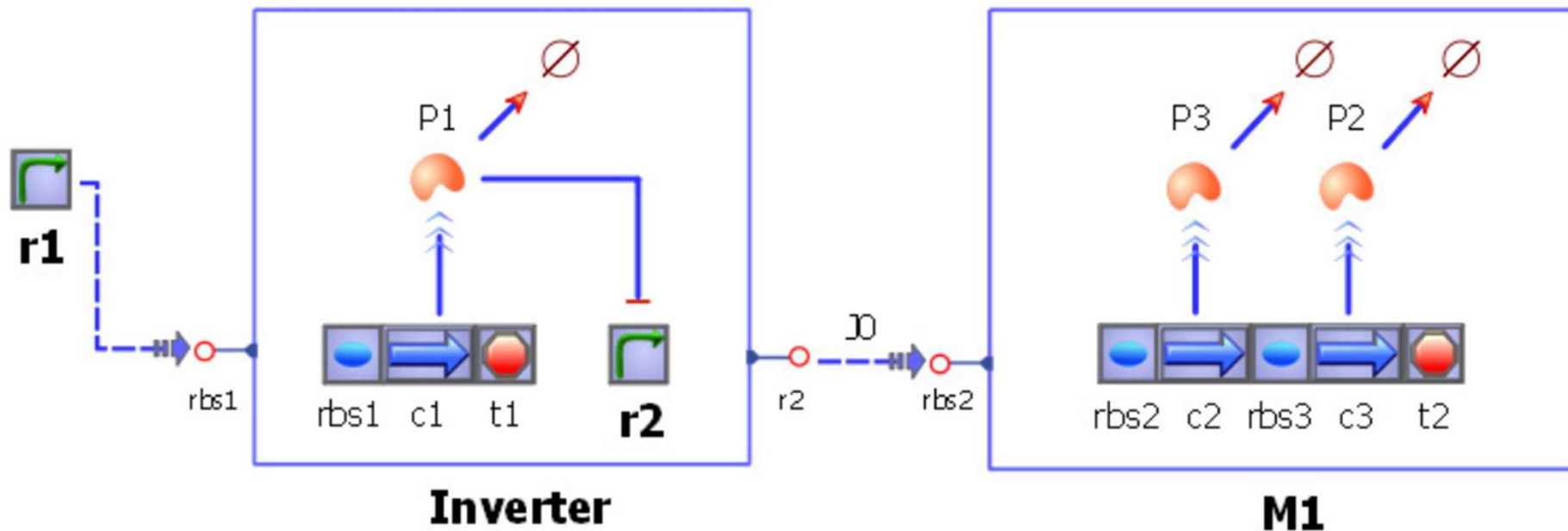
- A visual modelling tool for computer aided design of genetic circuits
<http://tinker-cell.blogspot.com/>

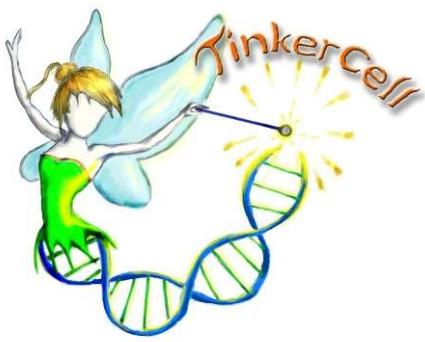




TinkerCell

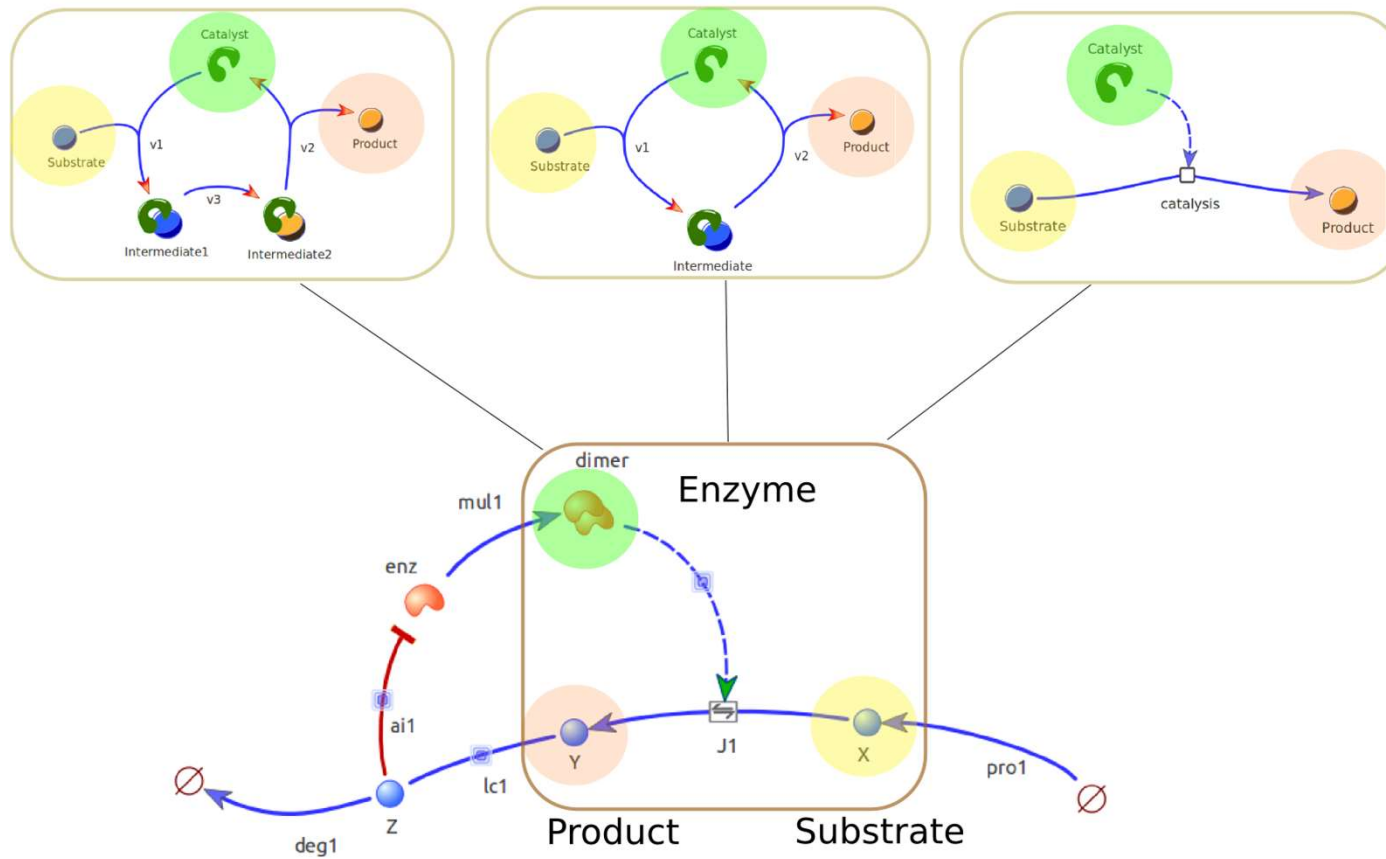
- A visual modelling tool for computer aided design of genetic circuits
<http://tinker-cell.blogspot.com/>





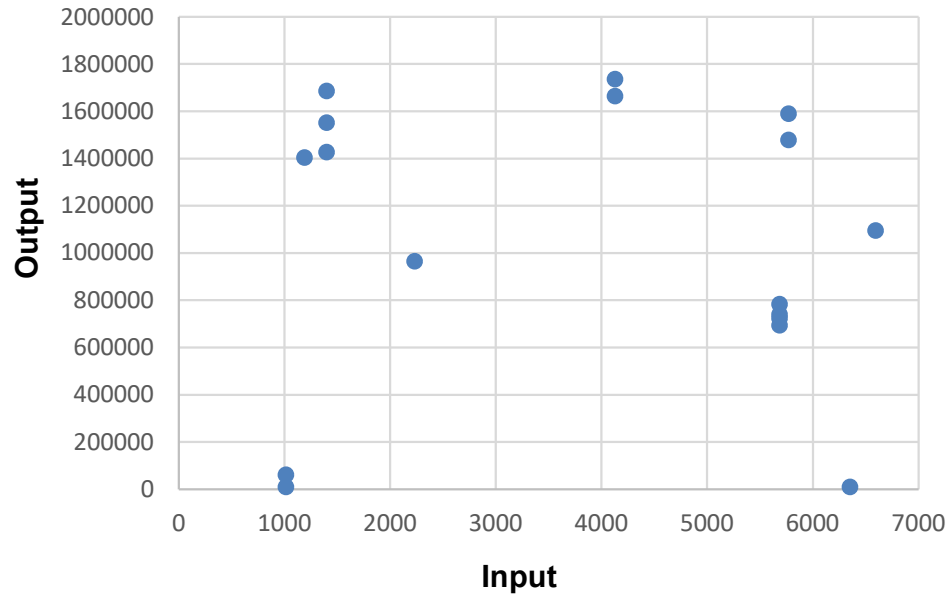
TinkerCell

- A visual modelling tool for computer aided design of genetic circuits
<http://tinker-cell.blogspot.com/>



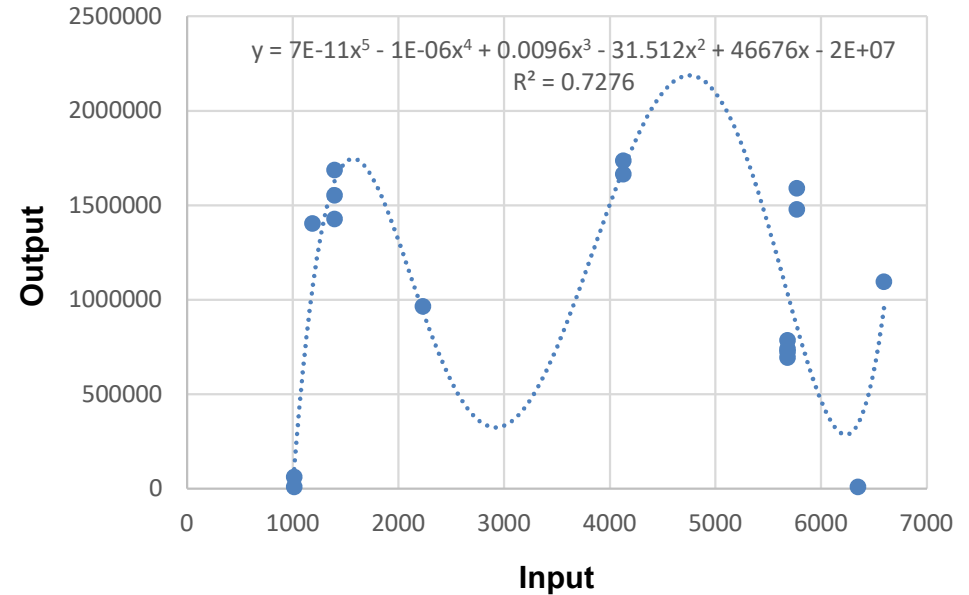
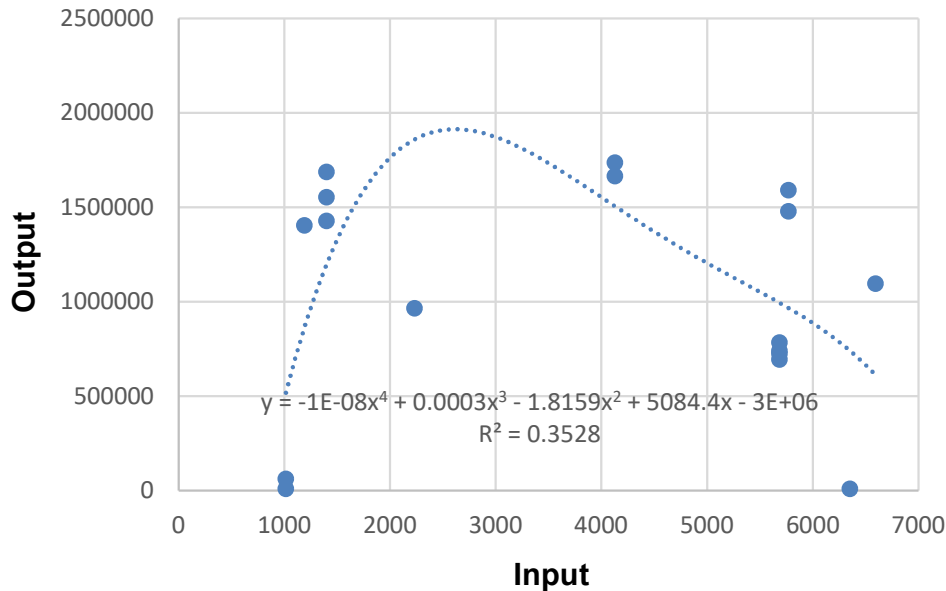
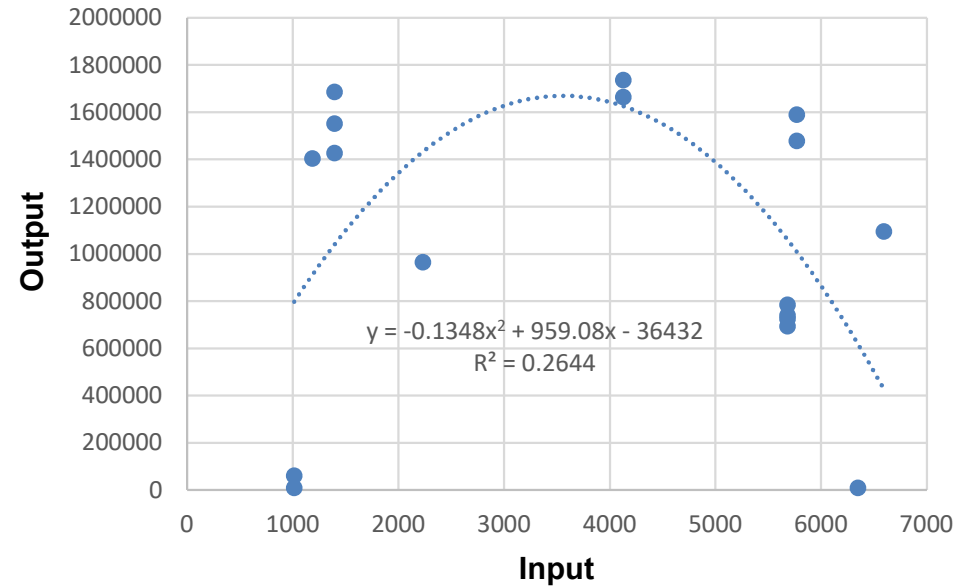
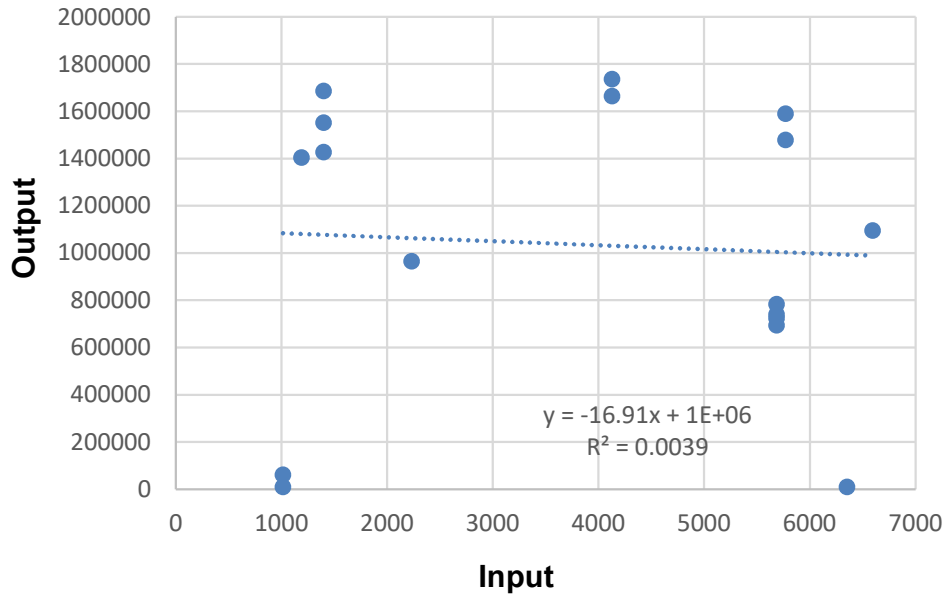
How many Fitted Parameters?

- What function will fit this data?



How many Fitted Parameters?

- Fitting too many parameters may result in less useful models (R^2 notwithstanding)

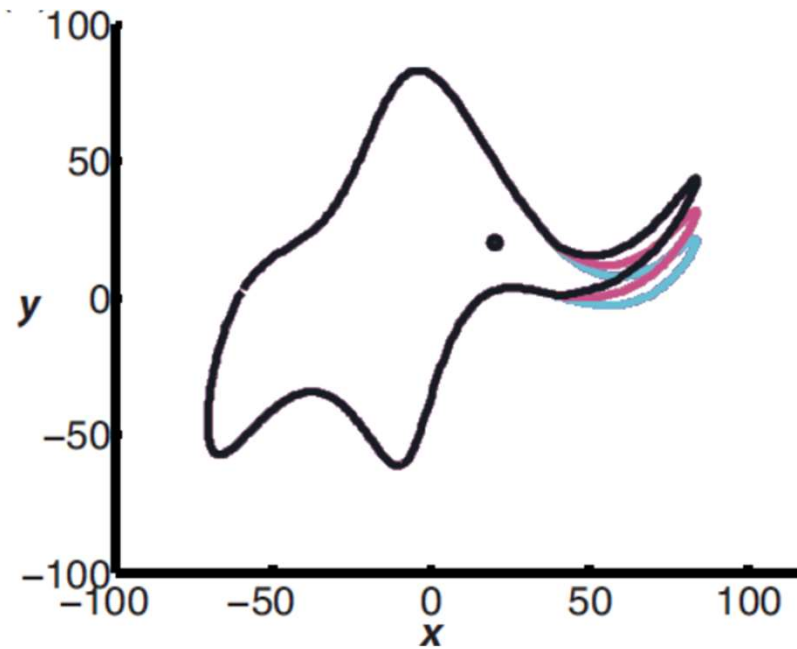


How many Fitted Parameters?

- Fitting too many parameters can result in less useful models

“with four parameters I can fit an elephant, and with five I can make him wiggle his trunk.” -John von Neumann

(as narrated by Enrico Fermi in Dyson 2004, Nature v427, p297)



(Mayer *et al.*, 2010.
American Journal of Physics v78, p648)

How detailed does a model have to be?

- A model may be useful even if it does not represent all the details of the system

“All models are wrong, but some are useful.”

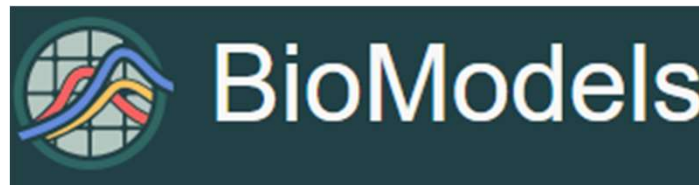
-George Box (1979), "Robustness in the strategy of scientific model building"

“Now it would be very remarkable if any system existing in the real world could be exactly represented by any simple model. However, cunningly chosen parsimonious models often do provide remarkably useful approximations. For example, the law $PV = RT$ relating pressure P , volume V and temperature T of an "ideal" gas via a constant R is not exactly true for any real gas, but it frequently provides a useful approximation and furthermore its structure is informative since it springs from a physical view of the behavior of gas molecules. For such a model there is no need to ask the question "Is the model true?". If "truth" is to be the "whole truth" the answer must be "No". The only question of interest is "Is the model illuminating and useful?".”

Circuit Modelling and Finding Parameters from Literature

Resources

- Biological Circuit Design <https://biocircuits.github.io/index.html>



- BioModels Parameters: <https://www.ebi.ac.uk/biomodels/parameterSearch> (Glont *et al.*, 2020. Bioinformatics.)



- <https://bionumbers.hms.harvard.edu/search.aspx>

Questions welcome.

manish.kushwaha@inrae.fr