3000788 Intro to Comp Molec Biol

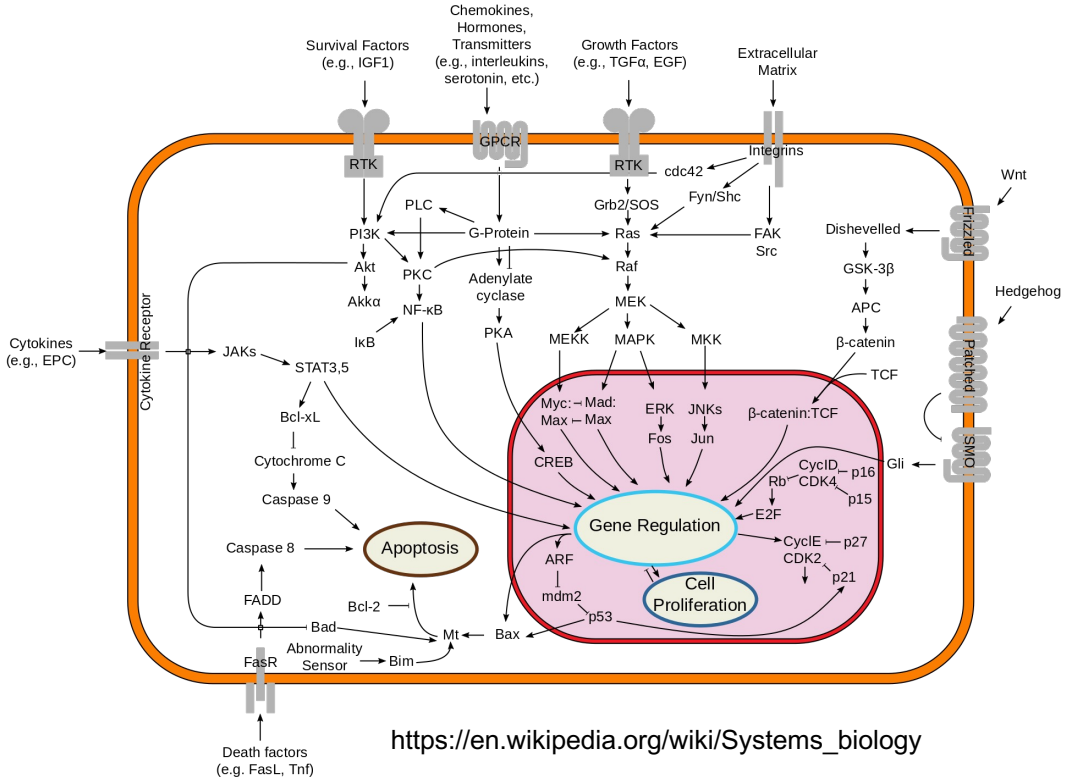
Lecture 21: Dynamics modeling for systems biology

October 30, 2023

Sira Sriswasdi, PhD

- Research Affairs
- Center of Excellence in Computational Molecular Biology (CMB)
- Center for Artificial Intelligence in Medicine (CU-AIM)

Systems biology

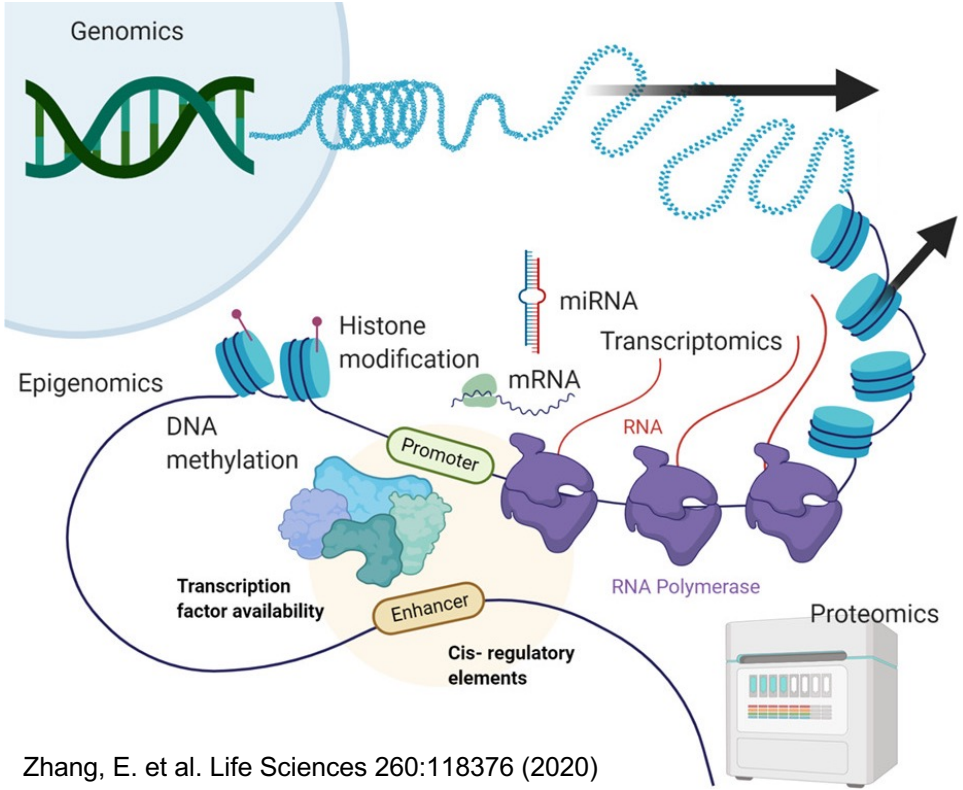


https://en.wikipedia.org/wiki/Systems_biology

- A (biological) system consists of components (genes) and rules (gene expression regulations) that control its characteristics (phenotypes)

- Systems biology = integration of data and model to fully understand the system

Multi-omics approach

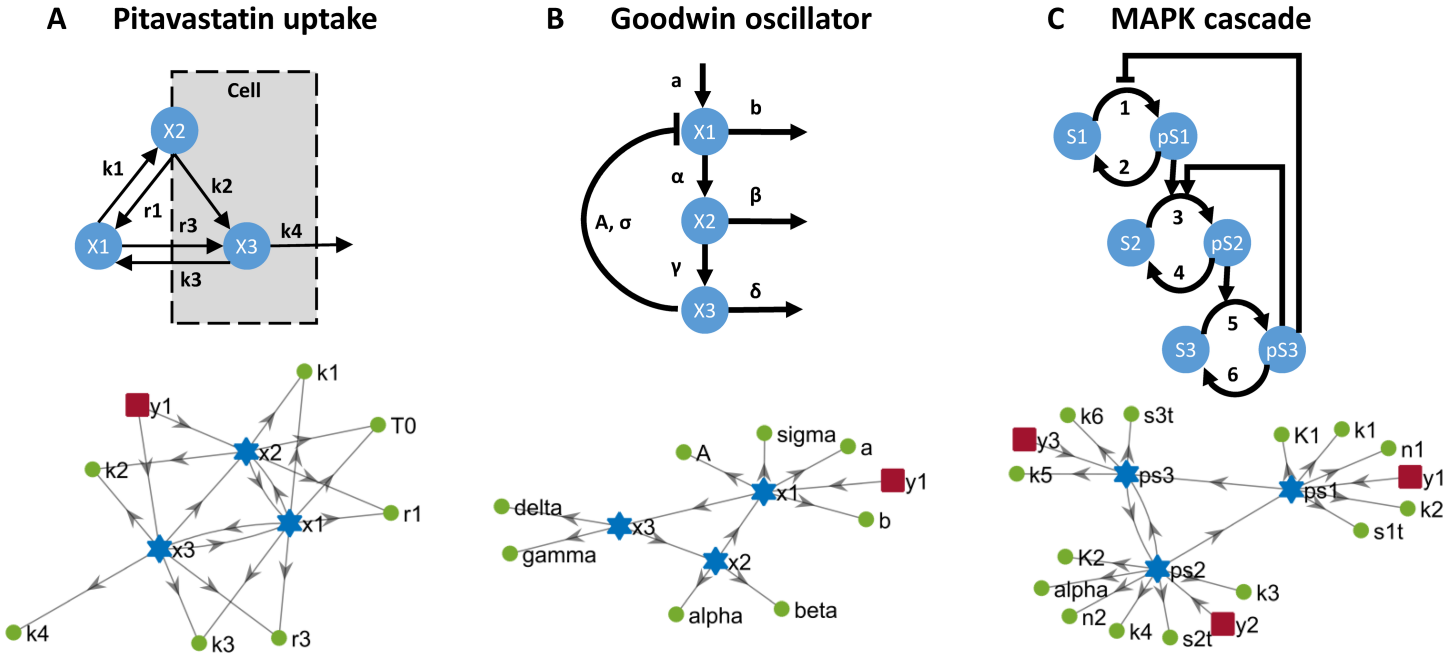


Zhang, E. et al. Life Sciences 260:118376 (2020)

- Genomics → gene state
- Epigenomics → chromatin state
- Transcriptomics → gene expression
- Proteomics → protein expression
- Other assays → protein function
- Provide mechanistic understanding

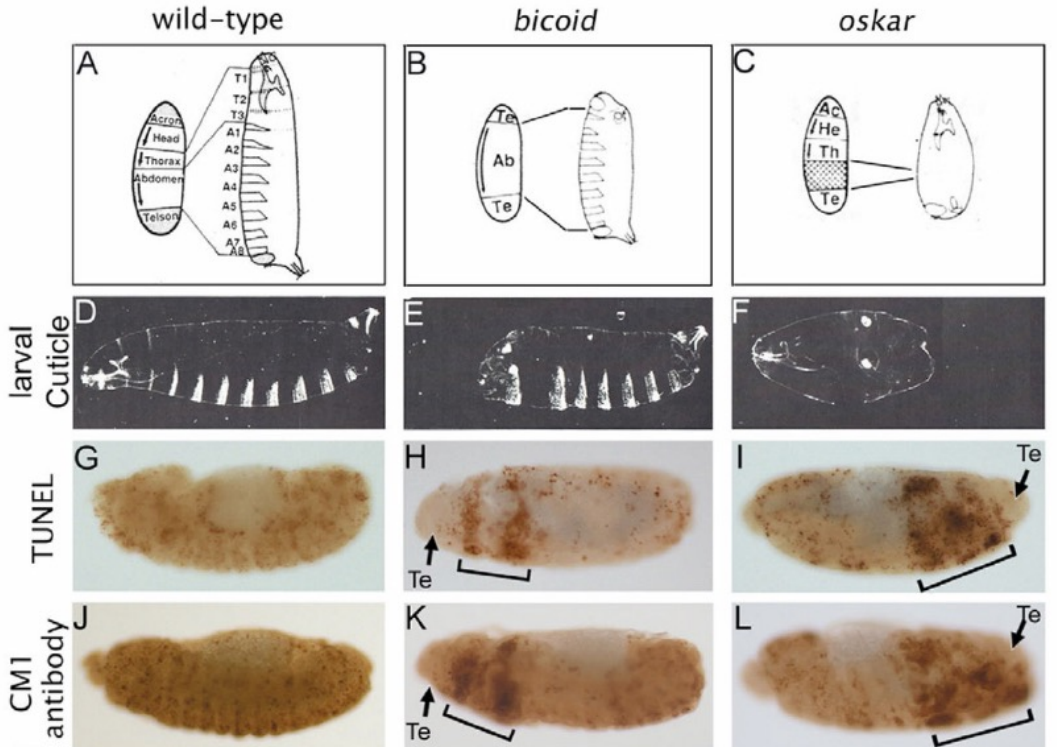
Modeling approach

Villaverde, A.F. PLoS Comp Biol 12:e1005153 (2016)



- Represent molecular interaction as network of physical-chemical reactions

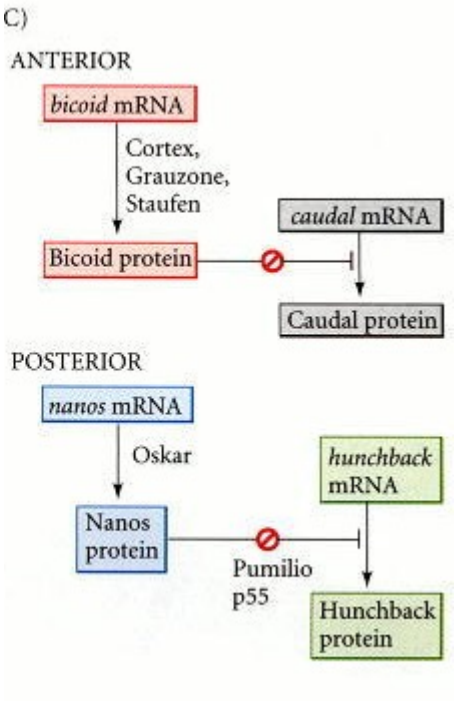
Why is systems biology interesting?

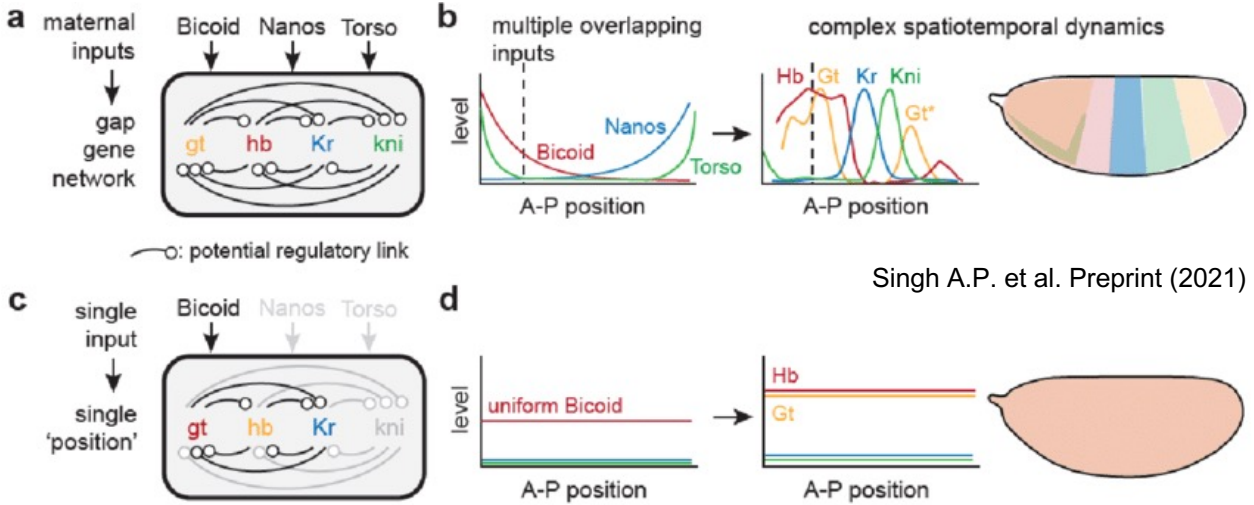


- How does gene and protein know where to activate in the body during development?
- Each cell starts in identical state!

Werz, C. Development 132:5343-52 (2006)

Why is systems biology interesting?





Singh A.P. et al. Preprint (2021)

- Gene regulatory dynamics lead to specific expression pattern over time

https://www.ncbi.nlm.nih.gov/books/NBK10039/

A simple enzymatic reaction

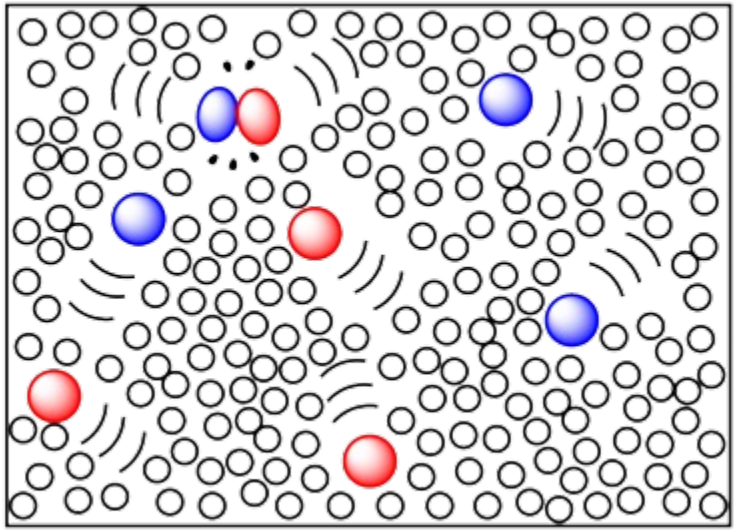
$$E + S \underset{k_{off}}{\overset{k_{on}}{\leftrightarrow}} ES \xrightarrow{k_{react}} E + P$$

- $\frac{d[S]}{dt}$ = Rate of change in $S = -k_{on}[E][S] + k_{off}[ES]$
- $\frac{d[E]}{dt}$ = Rate of change in $E = -k_{on}[E][S] + k_{off}[ES] + k_{react}[ES]$
- $\frac{d[P]}{dt}$ = Rate of change in $P = k_{react}[ES]$

A key mental image

Rate of E + S → ES =
Rate of E meeting S x
Rate of binding

Rate of E + S → ES =
[E][S] x k



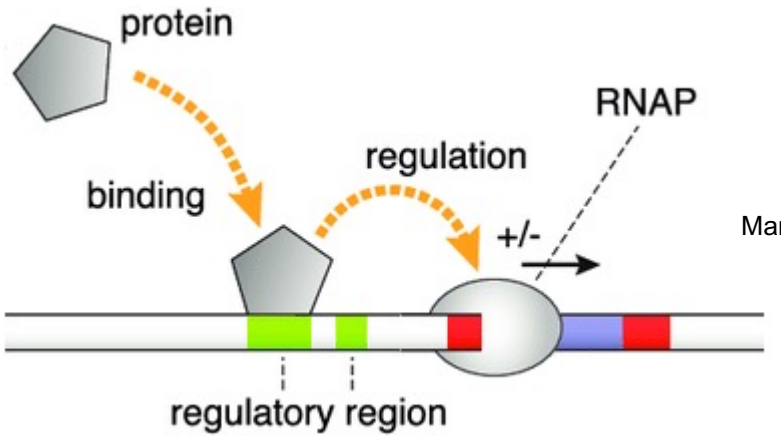
https://employees.csbsju.edu/cschaller/Reactivity/kinetics/rkphase.htm

Rate of E meeting S scales linearly with [E] and [S]

Rate of binding is a constant for E and S

- Molecules must find each other in 3D to bind and interact
- Reaction takes time (and typically energy)

From enzymatic reaction to transcription



Marbach, D. et al. EvoBIO 2007

- $TF + DNA_{inactive} \leftrightarrow TF\text{-}DNA$
- $RNAP + TF\text{-}DNA \rightarrow RNAP + TF\text{-}DNA + RNA$
- $RNA \rightarrow$ degraded RNA

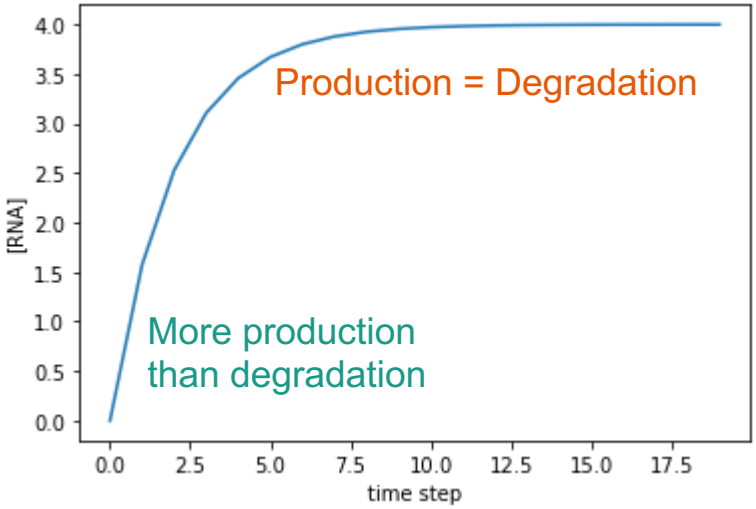
A differential equation for gene expression

$k_{transcription}$

$$DNA \rightarrow DNA + RNA$$

$$RNA \rightarrow \emptyset$$

$k_{degradation}$



- $\frac{d[RNA]}{dt}$ = rate of change of RNA = $k_{transcription} - k_{degradation}[RNA]$
- What would the graph of [RNA] looks like?

Simulating differential equation

$$x(t+1) = x(t) + (x(t+1) - x(t)) = x(t) + \frac{x(t+1) - x(t)}{(t+1) - t} = x(t) + \frac{\Delta x}{\Delta t}(t)$$

- $\frac{\Delta x}{\Delta t}(t)$ can be approximated by $\frac{dx}{dt}(t)$, or $x'(t)$
- Differential equation defines $x'(t)$
  - $\frac{d[RNA]}{dt} = k_{transcription} - k_{degradation}[RNA]$

- Start at an initial condition $x(0) = x_0$,
  - Calculate $x(1) = x(0) + x'(0) = x_0 + k_{transcription} - k_{degradation}x_0$
  - Calculate $x(2) = x(1) + x'(1) = x(1) + k_{transcription} - k_{degradation}x(1)$

Simulating differential equation

- $\frac{d[RNA]}{dt} = k_{transcription} - k_{degradation}[RNA]$

- Start at an initial condition $x(0) = x_0$,
  - Calculate $x(1) = x(0) + x'(0) = x_0 + k_{transcription} - k_{degradation}x_0$
  - Calculate $x(2) = x(1) + x'(1) = x(1) + k_{transcription} - k_{degradation}x(1)$
  - $= x_0 + 2 \cdot k_{transcription} - k_{degradation}(x_0 + x(1))$
  - [RNA] after 2 timesteps = initial [RNA] + 2 units of transcription – combined degradation of [RNA] at both timesteps

A toy example

- $\frac{d[RNA]}{dt} = 4 - 0.2\,[RNA]$

- Start at an initial condition $x(0) = 100$,
  - $x(1) = 100 + 4 - 20 = 84$
  - $x(2) = 84 + 4 - 17 = 71$
  - $x(3) = 71 + 4 - 14 = 61$
  - $x(4) = 61 + 4 - 12 = 53$
  - $x(5) = 53 + 4 - 10 = 47$

Another toy example

- $\frac{d[RNA]}{dt} = 4 - 0.2\,[RNA]$

- Start at an initial condition $x(0) = 0$,
  - $x(1) = 0 + 4 - 0 = 4$
  - $x(2) = 4 + 4 - 1 = 7$
  - $x(3) = 7 + 4 - 1 = 10$
  - $x(4) = 10 + 4 - 2 = 12$
  - $x(5) = 12 + 4 - 2 = 14$

scipy.integrate.odeint

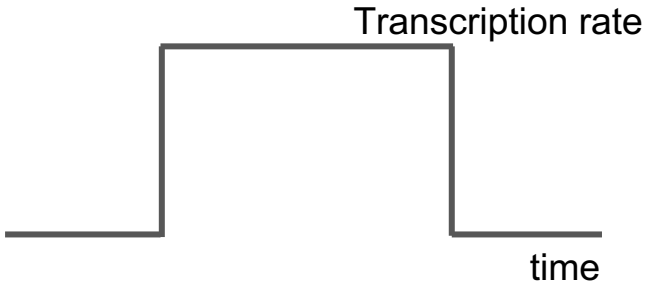
```
from scipy.integrate import odeint
import matplotlib.pyplot as plt
```

```
def simple_transcription(rna, time, k_trans, k_deg):
    return k_trans - k_deg * rna
```

```
initial_rna = 0
times = range(0, 20)

k_trans = 2
k_deg = 0.5

simulated = odeint(simple_transcription, initial_rna, times, args = (k_trans, k_deg))
```

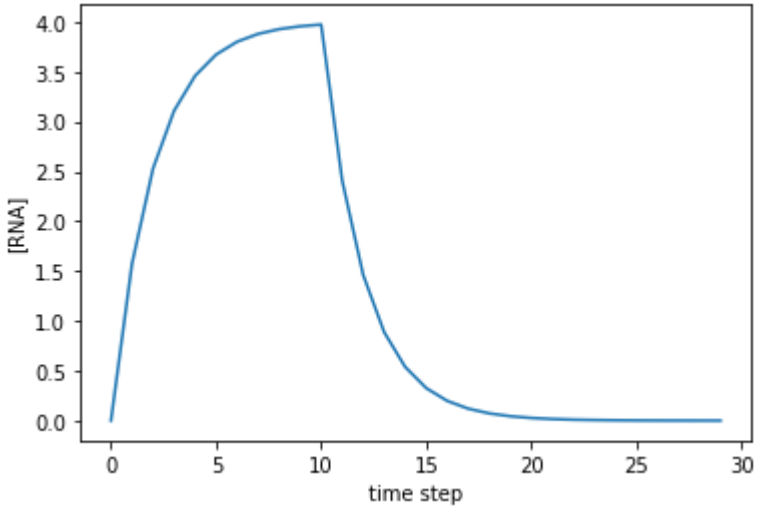
- odeint() takes a function that return derivative(s), initial condition, and time steps

Time-dependent activation

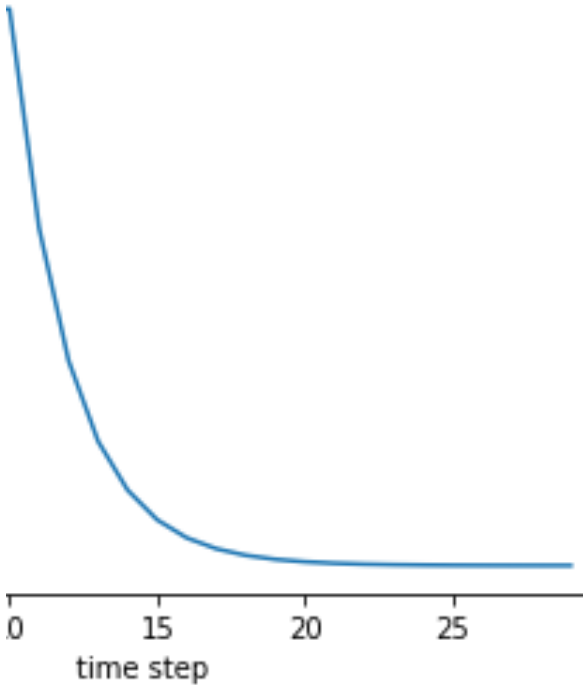



```python
def time_transcription(rna, time, k_trans, k_deg):
    if time < 10:
        return k_trans - k_deg * rna
    else:
        return - k_deg * rna
```

- First half of the dynamics is the same as before
- Second half is when RNA degrades until the [RNA] falls back to the basal level

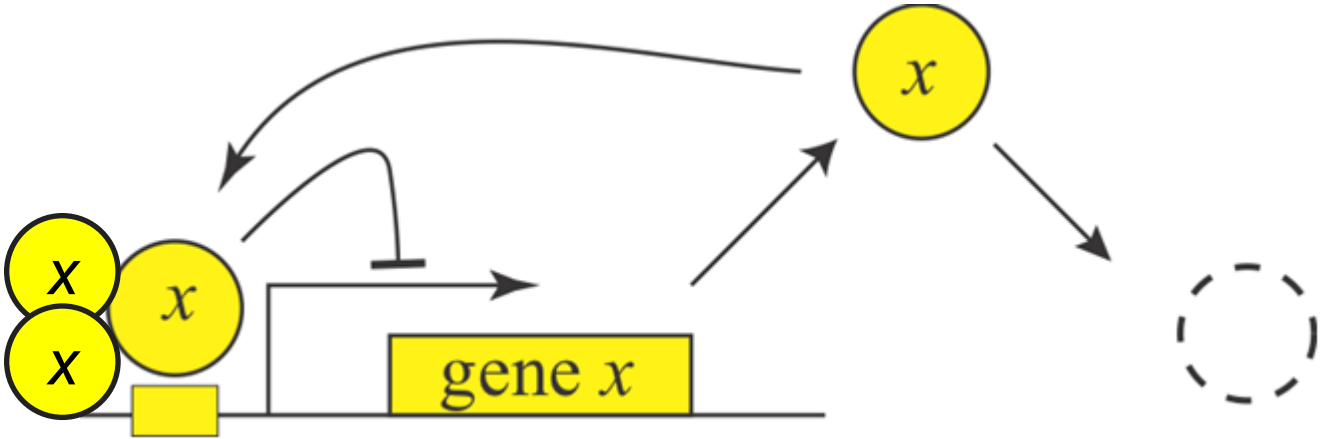


Exponential decay



- $\frac{d[RNA]}{dt} = -k_{\text{degradation}}[RNA]$
- Fast decay in the beginning because there are a lot of RNA molecules
- Slower decay at the end
- $\frac{de^{-kt}}{dt} = -ke^{-kt}$

Negative auto-regulation



http://be150.caltech.edu/2019/handouts/03_small_circuits.html

- $\frac{d[X]}{dt} = k_{transcription}[DNA_{active}] - k_{degradation}[X]$
- $[DNA_{active}]$ also depends on $[X]$ through binding dynamics
- Two variables!

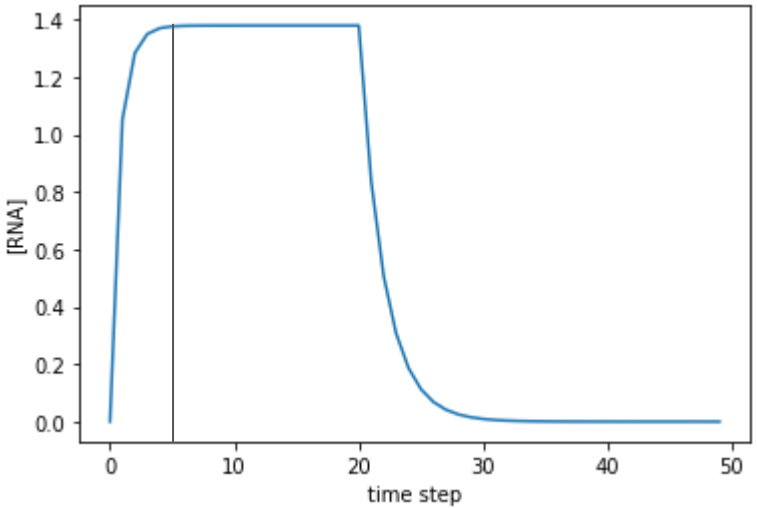
Negative auto-regulation

$$nX + DNA_{active} \underset{k_{unbind}}{\overset{k_{bind}}{\leftrightarrow}} DNA_{inactive}$$

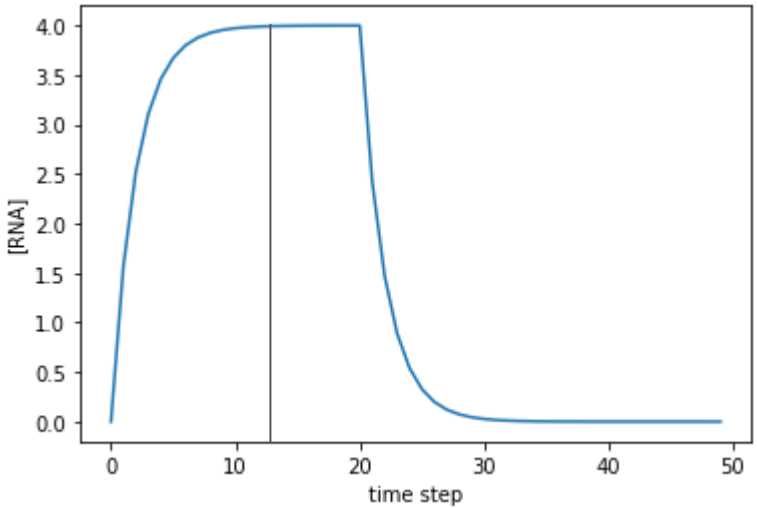
- $\frac{dDNA_{inactive}}{dt} = k_{bind}[X]^n[DNA_{active}] - k_{unbind}[DNA_{inactive}]$
- At binding equilibrium, fraction of unbound (active) DNA $= \frac{k_{bind}[X]^n}{k_{unbind}+k_{bind}[X]^n}$
- Also known as Hill function

Faster response time with negative auto-regulation

With negative auto-regulation

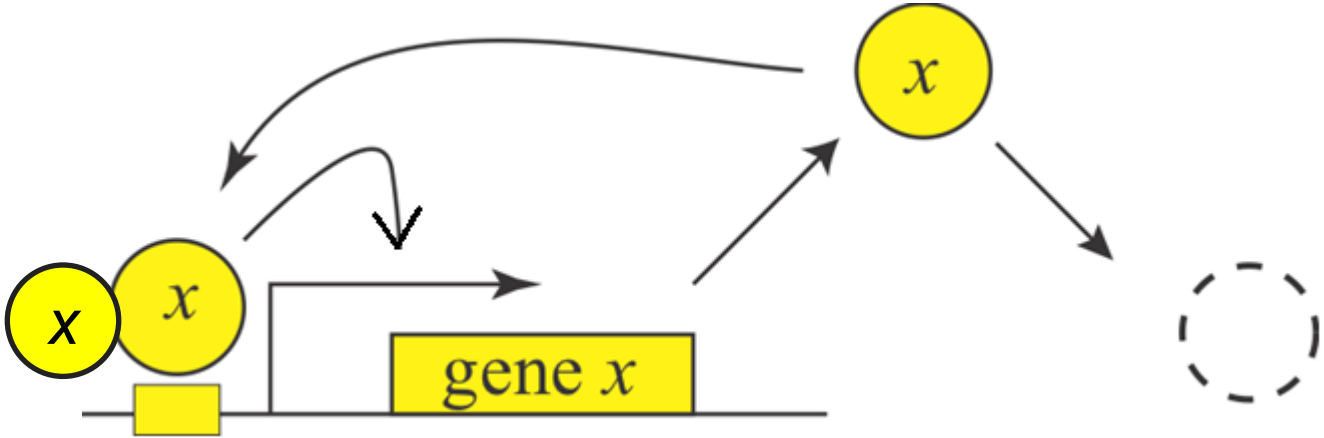


Without negative auto-regulation



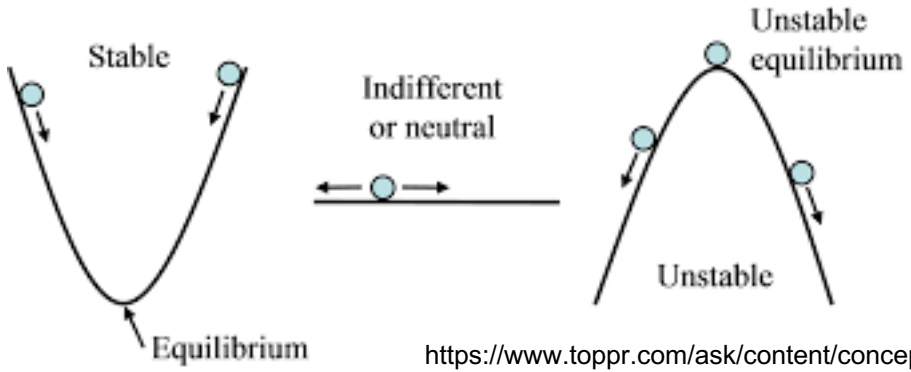
- $\frac{d[X]}{dt} = \frac{k_{transcription}}{1+ (k[X])^n} - k_{degradation}[X]$

Positive auto-regulation



- $\frac{d[X]}{dt} = \frac{k_{transcription}(k[X])^2}{1+(k[X])^2} - k_{degradation}[X]$
- Can you guess how this system will behave?
  - How many equilibria are there?

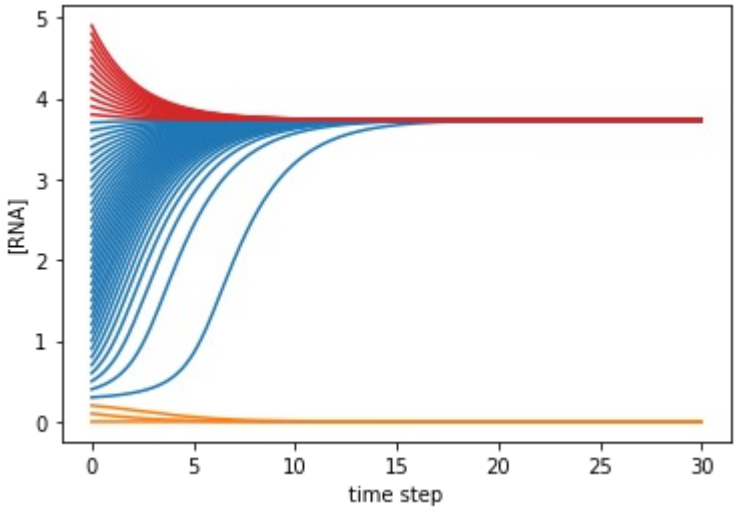
Stability of an equilibrium



https://www.toppr.com/ask/content/concept/equilibrium-and-its-types-208616/

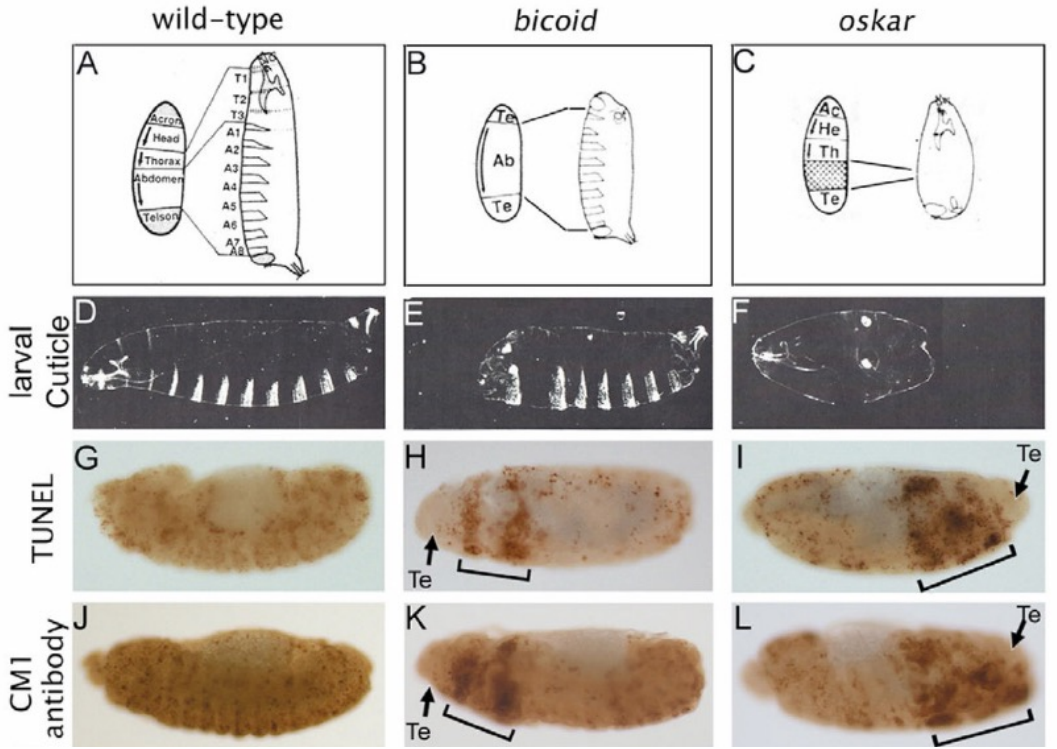
- At equilibrium, $\frac{k_{transcription}(k[X])^2}{1+(k[X])^2} = k_{degradation}[X]$
- There are three solutions, one of which is $[X] = 0$
  - One solution is unstable, two are stable
- Depending on the initial $[X]$, the system can converge to either stable equilibrium

Bistability



- For **low** [X], the degradation term wins over the transcription and [X] decays down to zero
- For **intermediate** [X], the transcription term wins over degradation and [X] increases until reaching the other equilibrium
- For **high** [X], the degradation term wins over the transcription and [X] decays down to the nearest equilibrium

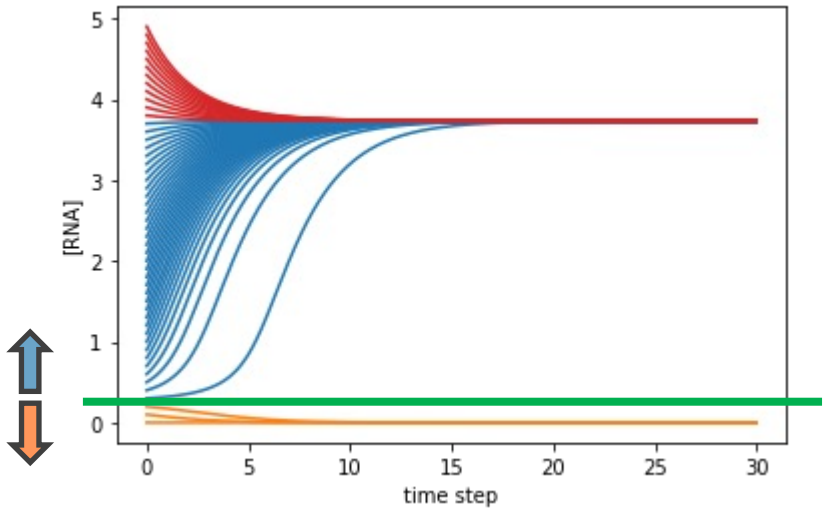
Tunable cell state based on equilibria



- With more genes and regulatory interactions, many more equilibria can exist
- Gradient of gene expression can establish position-specific cell states

Werz, C. Development 132:5343-52 (2006)

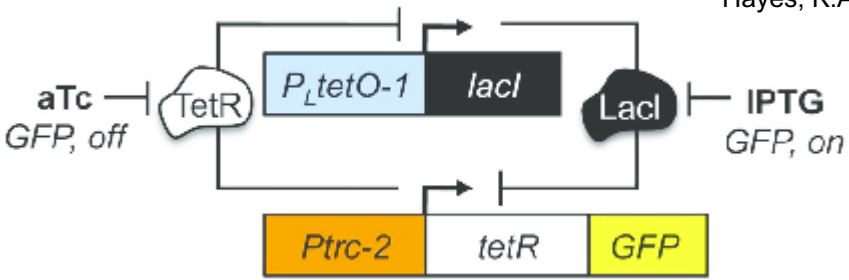
Bistability as memory



- Given a cell in low expression state
- A drug treatment raises the expression level in the cell pass the green threshold
- After stopping the treatment, the cell will “**memorize**” that it is now in the high expression state

Memory from gene toggle switch

Hayes, K.A. et al. J of Cell Biology 187:589-96 (2009)

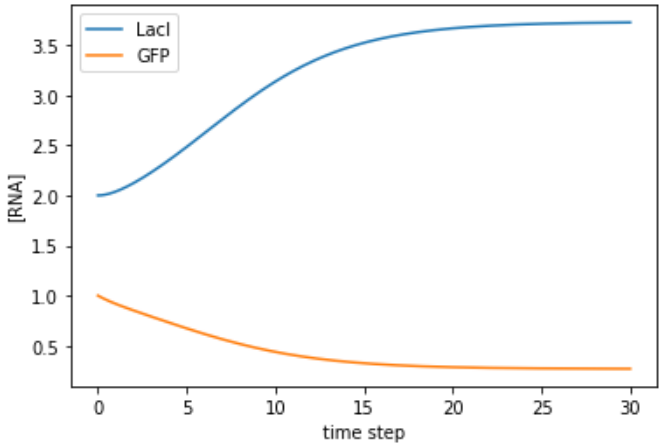


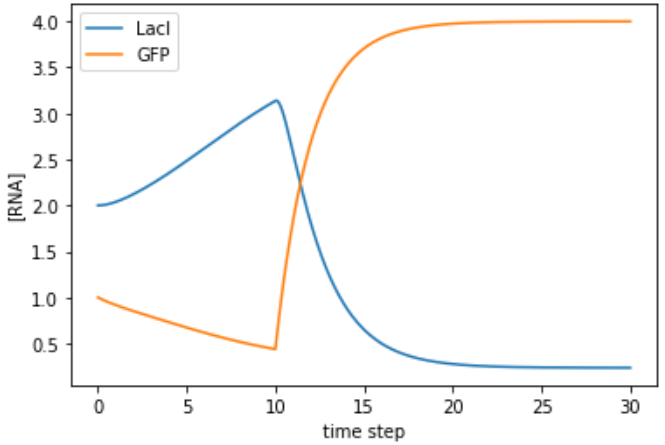
- Two genes repressing each other
- **Two equilibria**: High LacI or high GFP

- If GFP is ON, how can we turn it off? Adding aTc or IPTG?
- Once GFP is OFF, can we stop supplying those molecules to the cell?

Toogle switch simulation

```python
def toggle_switch(rna, time, k_trans, k_deg, n):
    LacI = rna[0]
    GFP = rna[1]

    dLacI_dt = k_trans / (1 + GFP ** n) - k_deg * LacI
    dGFP_dt = k_trans / (1 + LacI ** n) - k_deg * GFP

    return [dLacI_dt, dGFP_dt]
```





- odeint() can handle multiple equations
- Input rna is a list [LacI, GFP]

- **Bottom Panel**: add IPTG to neutralize LacI at time = 10

Two-gene linear system

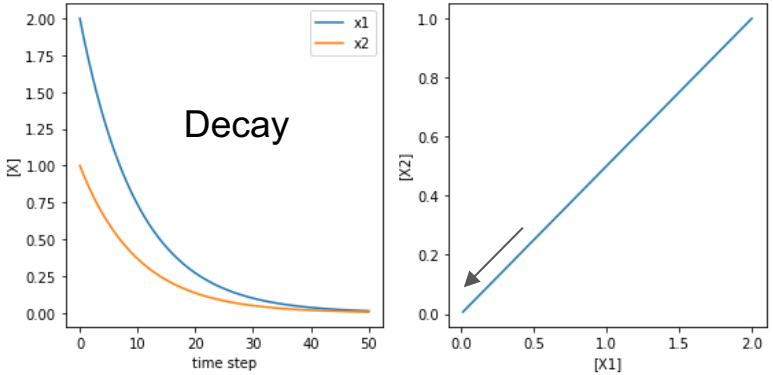
$$\frac{d[X1]}{dt} = k_{1,1}[X1] + k_{1,2}[X2]$$

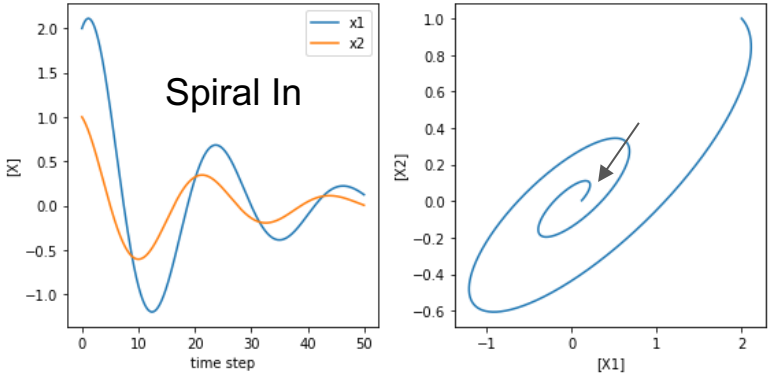
$$\frac{d[X2]}{dt} = k_{2,1}[X1] + k_{2,2}[X2]$$

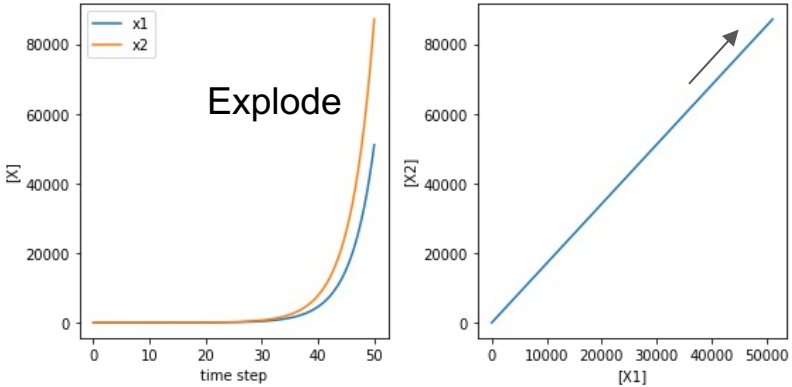
```python
def two_loci_linear(rna, time, k11, k12, k21, k22):
    dx1_dt = k11 * rna[0] + k12 * rna[1]
    dx2_dt = k21 * rna[0] + k22 * rna[1]

    return [dx1_dt, dx2_dt]
```

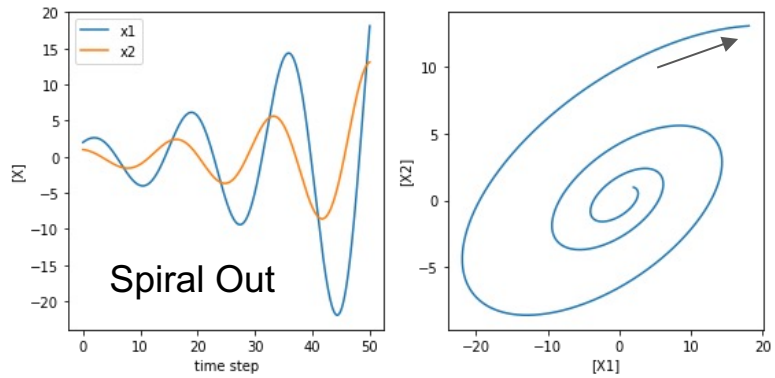
- A simple two-gene system with only linear effects
- Depending on $k_{i,j}$, interesting dynamics can be derived
  - Actually, depending on **Trace** and **Determinant** of the matrix $\{k_{i,j}\}$

Some possible dynamics

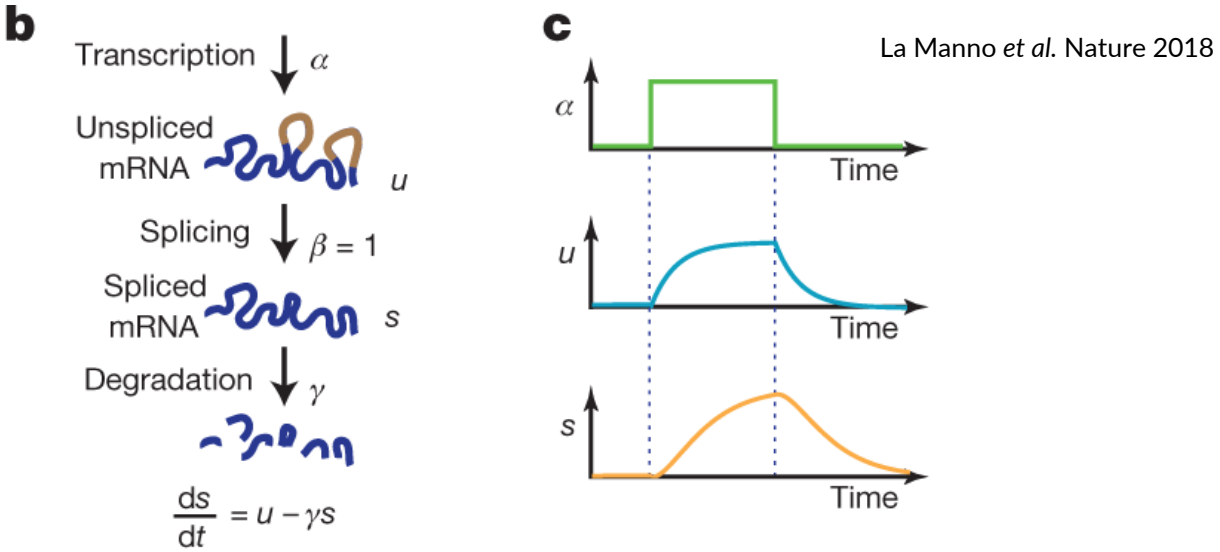








RNA velocity model



La Manno *et al.* Nature 2018

- **Two-step process**: From DNA to unspliced RNA to spliced RNA
- $\frac{d[U]}{dt} = \alpha - \beta[U], \frac{d[S]}{dt} = \beta[U] - \gamma[S]$
- Not the same dynamics as simplifying $\frac{d[S]}{dt} = \alpha - \gamma[S]$

Summary

- Systems biology = gain mechanistic understanding of a biological systems
  - Combine multi-omics data
  - Model the dynamics
- Different stable cell states arise from multiple gene expression equilibria
- Diverse dynamics can originate from a simple two-gene system
- **Next**: Gene expression simulation with Python

Any question?

- See you on November 2