Control and Computation in Biological Systems
Scribe note collection

Fall 2025

Lecturer: Fangzhou Xiao
Website: https://chemaoxfz.github.io/ccbs

Westlake University

Contents

Chapter 1

Order of Magnitude (OoM) reasoning in physics and biology

Scribe: Qinguo Liu+ Yihang Ding
Lecturer: Fangzhou Xiao
Lecture #1 20250904

Contents

1.1 Order of Magnitude(OoM) reasoning in Phisics

The world is connected, and our "eye" to see such connections are orders of magnitude reasoning. Seemingly unrelated observations could be in fact deeply constraining each other. To practice this "vision", we explore some calculations below, with contexts gradually shifting from the macroscopic world we are more familiar with to the microscopic world of molecules and cells.

1.1.1 Hangzhou Exodus

To get a feeling for order of magnitude reasoning, let us start with an estimate for the following problem. Imagine Hangzhou is in a sudden crisis, so we need to evacuate the population of Hangzhou, what would be the best way to do it? By Car? Train? Airplane? Remember that Hangzhou has a population of about 10 million.

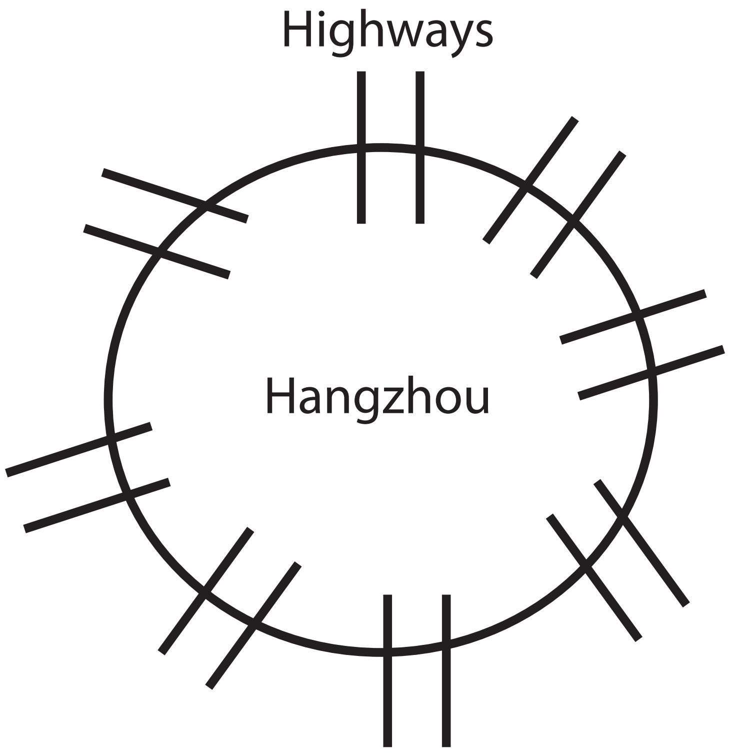
We probably want to leave Hangzhou fast. So let us consider the transportation method that is fastest for individuals.

- By airplane:
  - We have 1 airport in Hangzhou
    Let's say under emergency we need $2\ \mathrm{min}$ for an airplane to leave airport.
    $$\begin{aligned} 2\ \mathrm{min/airplane} &\Longrightarrow 60 \times 24 \approx 10^3\ \mathrm{min/day} \\ &\Longrightarrow \approx 500\ \mathrm{airplane/day} \\ 10^2\ \mathrm{people/airplane} &\Longrightarrow 5 \times 10^4\ \mathrm{people/day} \\ &\Longrightarrow 200\ \mathrm{days} \end{aligned}$$

200 days is too long for to escape from an apocalypse. Is there a method that can be faster than this? We observe that the key bottleneck of the airplane method is that we only have one airport in Hangzhou. The airplane itself is fast, but the number of runways is too low.

Given this, let us leave the estimate for trains as an exercise (we would guess it is too slow overall due to the same reason of bottleneck as the airplane method), and directly jump to escape by cars. Cars have many lanes on highways to travel, which may resolve the previous bottleneck.

Figure 1.1



- By cars
  - Let's say totally we have 10 highways to leave Hangzhou (Roughly check the map, see Fig 1.1)
    $4\ \mathrm{lanes/way} \Longrightarrow 8\ \mathrm{lanes\ total/way}$ (Emergency state: all lanes going out.)
    $\Longrightarrow \approx 10^2\ \mathrm{lanes}$
  - Properties of cars

Speed	$v \approx 80\ \mathrm{km/h} \approx 20\ \mathrm{m/s}$
Distance between cars	$L \approx 40\ \mathrm{m/car}$ (let's say we need 2s to react)
Capacity	$N = 4\ \mathrm{people/car}$

$$\text{FLuxPerLane} = \frac{N \cdot v}{L} = \frac{4 \text{ people/car} \times 20 \text{ m/s}}{40 \text{ m/car}} \approx 2 \text{ people/s}$$

$\Longrightarrow$100 lanes $\times$ 2 people/(s $\cdot$ lane) $\times$ 3600 s/hour $\times$ 24 h/day $\approx 2 \times 10^7$ people/day
So roughly 1 day all people are evacuated.

Roughly 1 day to evacuate all people in Hangzhou, which is very nice indeed! However, we assumed the only bottleneck is the number of lanes and ignored other possible constraints. Could new limiting factors come into play? For example, we assumed there are enough cars in Hangzhou, and every person can get into a car. Could the actual number of cars be limiting? What if people drive slower or faster than the speed we assumed?

To reason about these problems, let us investigate the real bottleneck of the cars method. We notice that the flux of people per lane is dependent on both the speed of driving and the distance between cars, and the latter is limited by our time to react, τ. But τ is the same due to fundamental human reflex, which cannot be drastically reduced. So this is the actual bottleneck. Let us rewrite our calculation in terms of τ.

- Could it be that number of Cars is not enough?
  (A new bottom neck? Could be, but does it matter?)
  Let τ be the time to react to guarantee safety.And $\underline{\tau \text{ is fixed }}$as we are all human.
  $$\text{Flux} = \frac{Nv}{L} = \frac{N}{\tau} \quad \left(\tau = \frac{L}{v}\right)$$
  $\Longrightarrow$Car = Walking
  e.g. we walk 1 m/s, need 2 m distance between people.
  So, we could just WALK ON HIGHWAYS to evacuate.

The Hangzhou exodus example shows how the dimensions of key quantities dictate the solution to a problem. Next, let us consider another problem that further demonstrates this dimensional reasoning, and how it can be used to extend our intuition from the macroscopic world we are familiar with to other scales.

1.1.2 Dimensional reasoning: jump height in animals

Many animals can jump while across different sizes, we humans can jump, cats can jump, fleas can jump. We may be wondering if there is some relationship between jump heights and animal sizes?

- Example: A student can jump:
  30 cm for Karla; Maybe 60 cm for a professional athlete.
  (There is no change of orders of magnitude within humans.)
- To infer the relationship, let's analyze from the energy:
  $$\begin{aligned} E_g &= mgh = E_m && (E_g: \text{ gravitational potential energy}) \\ m &\propto L^3 && (m \propto \text{Volume} \propto \text{Length}^3) \\ E_m &\propto L^3 f_m && (f_m: \text{ volume fraction of muscle}) \end{aligned}$$
  $\Longrightarrow\ h \propto L^0 f_m$
  Jump height is independent of mass/volume/size.

The above result is saying, the height an animal can jump has nothing to do with its mass, volume or size, the only thing that matters is f_m, the muscle fraction, as long as the animal still uses the same muscle mechanism to jump. And this is roughly true:

- Rat : $50\ \mathrm{cm}$
  Flea : $25\ \mathrm{cm}$
  Cats : $100\ \mathrm{cm}$ (Roughly same order of magnitude)

The next question is time, as we have the same h, we got the same speed leaving the ground.

$$\frac{1}{2}mv^2 = mgh \Longrightarrow v \text{ is constant as } h \text{ is constant}$$

So humans, rats, fleas and cats all share roughly the same speed when leaving the ground. Imagine fleas leaving the ground 3 meters per second! This is like thousands of body lengths per second, which scales to human would be kilometers per second!

If we observe Karla's jumping action carefully, we can find this jump takes her about $0.5\ \mathrm{s}$ from stretching the leg to leaving the ground. This duration is reasonable for muscles since she has about $0.5\ \mathrm{m}$ to release the muscle energy. But what about other animals?

- Time cost to release the energy in the muscle

  The time τ is proportional to the height one can use to speed up. Thus

$$\tau \propto \frac{L}{v} \quad (\tau: \text{ time to release energy})$$

$$\begin{array}{ll} \text{Human:} & v \approx 3\ \mathrm{m/s} \Longrightarrow \frac{L}{v} = \frac{1\ \mathrm{m}}{3\ \mathrm{m/s}} \approx 0.3\ \mathrm{s} \\ \text{Cat:} & v \approx 3\ \mathrm{m/s} \Longrightarrow \frac{L}{v} = \frac{0.2\ \mathrm{m}}{3\ \mathrm{m/s}} \approx 0.1\ \mathrm{s} \\ \text{Flea:} & v \approx 3\ \mathrm{m/s} \Longrightarrow \frac{L}{v} = \frac{2\ \mathrm{mm}}{3\ \mathrm{m/s}} \approx 1\ \mathrm{ms}\ !! \end{array}$$

Even fast muscles need about $0.1\ \mathrm{s}$ to contract its half length, $1\ \mathrm{ms}$ is far too short. The truth is fleas choose to store energy in the bending of their shells to go beyond this limitation on the the speed of release due to muscle mechanisms.

From rough calculations, we could use scaling to go from numbers we are familiar with and scale up to what is very different scales animals are doing.

For the next example, let's heat up everything, to see what intuition that heating up potatoes can bring us. If we know how long to heat up a potato, can we assume how long to cool down the moon?

1.1.3 Scaling: heating potatoes, and the moon

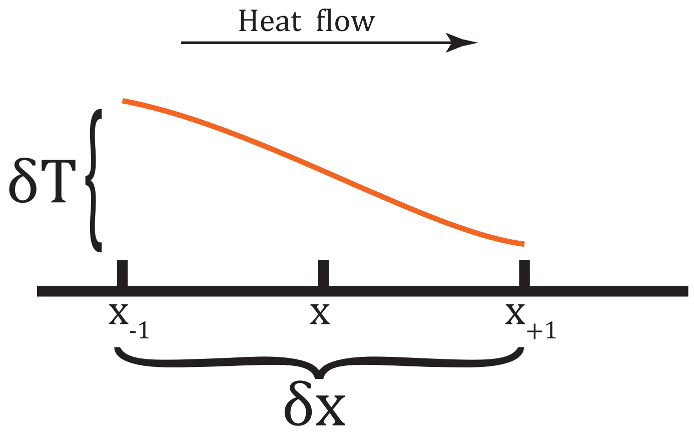
Firstly we need to link together the time and the heat change, to find how item's size can involve this process. For simplicity, let's consider a rod 1.2, with a temperature gradient along its length, decreasing from left to right.

- Heat diffusion on rod
  From intuitive, heat flux should be proportional to the local gradient of temperature,this gives:

$$J = -K\nabla T \qquad (J: \text{ Flux of the heat})$$

$$\nabla T \approx \frac{\delta T}{\delta x} \quad (\nabla T: \text{ Gradient of Temperature })$$

Figure 1.2



And also, the neat change of temperature is reflected by the stored energy.

$$\Delta E \approx C_V \delta T \quad (C_V : \text{ Heat capacity per Volume, with unit: Heat/T})$$
$$\text{(Change in } E \text{ per volume taken } T \text{ change by } \delta T)$$

To derive how temperature changes over time, we will apply the principle of energy conservation. Consider a small segment of the rod with length δx, The net heat flux (the heat flowing in minus the heat flowing out) results in a change in the segment's internal energy. This energy balance leads to the following relationship:

- Temperature changing in time
  From conservation of energy: at x change temperature by δT in δt time
  then

$$\delta x C_v \delta T = (J_{x_{-1}} - J_{x_{+1}})\delta t = K(\nabla T_{x_{+1}} - \nabla T_{x_{-1}})\delta t$$
$$\Longrightarrow \frac{\delta T}{\delta t} = \frac{K}{C_V} \cdot \frac{\nabla T_{x_{+1}} - \nabla T_{x_{-1}}}{\delta x}$$
$$\Longrightarrow \frac{\partial T}{\partial t} = \kappa \nabla^2 T \qquad (\kappa = \frac{K}{C_V} \text{ is heat diffusivity})$$

Now we have the partial derivative equation to describe how T changed with time, that says the temperature's change rate on one site is proportional to the temperature's Hessian on this site. The object's size's effect on the temperature change time is embedded within this Hessian. To get the atypical scaling, we can simply remove the partial derivatives. (We can derive the scaling just from variation in a unit time/length/etc).

- Thus we have

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^x}$$
$$\Longrightarrow \frac{T}{t} \approx \kappa \frac{T}{x^2}$$
$$\Longrightarrow t \approx \frac{x^2}{\kappa}$$

So the time grows with size's square. To cool down the moon, now we need another parameter κ, we don't know yet but we can estimate from heating up potatoes.

- Hong long to heat up a potato?
  From experience, about 10 min is need to heat up a potato of size10 cm
  $\Longrightarrow \kappa \approx \frac{x^2}{t} \approx \frac{1\times10^{-2}\ \mathrm{m}^2}{600\ \mathrm{s}} \approx 1.5 \times 10^{-5}\ \mathrm{m}^2/\mathrm{s}$
  Now we get the number of κ, finally we can turn our view to moon!

- Cool down the moon since its birth.

$$L = 2000\ \mathrm{km} = 2 \times 10^6\ \mathrm{m}$$
$$t \approx \frac{x^2}{\kappa} = \frac{4 \times 10^{12}\ \mathrm{m}^2}{1.5 \times 10^{-5}\ \mathrm{m}^2/\mathrm{s}} \approx 3 \times 10^{17} s$$
$$(3600\ \mathrm{s} \times 24\ \mathrm{h} \times 365\ \mathrm{day} \Longrightarrow 3 \times 10^7\ \mathrm{s/year})$$
$$\Longrightarrow t \approx 1 \times 10^{10} years$$

The solar system is approximately: 4.6 billion years $\approx 5 \times 10^9$ years. Our calculation suggests the Moon is not yet fully cooled, but very close. In reality, the Moon is considered fully cooled. This is because, early in its history, its molten core greatly accelerated the cooling process through convection, which is a much more efficient method of heat transfer than conduction alone.

Through these examples, we've seen how order of magnitude reasoning reveals profound connections. These cases demonstrate that by identifying the core physical constraints and understanding how they scale, we can make powerful predictions across vastly different domains.

Now let's dive into the biology world to see what surprise the order of magnitude reasoning can bring us.

1.2 OoM in biology: molecules in cells

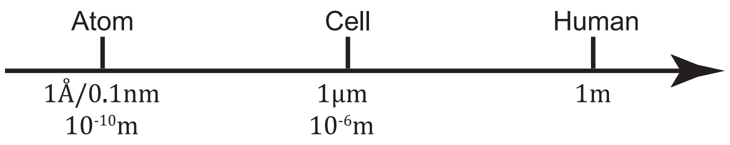
1.2.1 Order of Magnitude(OoM) reasoning in biology

- In biology world, OoM is often very helpful to give a "Null hypothesis","Null model", predicting what should occur based only on fundamental physical or chemical principles, before we spend much time doing literature reading, detailed simulations, or even experiments.

Like physics, in biology world,we often care about some quantities. At the single-item level, we care about properties like length, volume, weight, speed, position... At the population level, quantities like number and concentration etc. become important. Let's start with some of these basic properties of different molecules within cells.

1.2.2 Molecules in cells

- References: cell biology by the numbers. Also, "Snapshot: characteristic rates and timescales in cell biology". s



- Length scale of different cells

E.coli	$1\ \mu\mathrm{m}$
Yeast	$5\ \mu\mathrm{m}$
Mammalian	$20\ \mu\mathrm{m}$($10\ \mu\mathrm{m}$ roughly)

To make the following analysis more specific, let's take E.coli as our standard reference for the microscopic world.

- E.coli
  - Volume of E.coli:
    $1\ \mu\mathrm{m}^2 = 1\times 10^{-18}\ \mathrm{m}^3 = 1\times 10^{-15}\mathrm{L} = 1\ \mathrm{fL}$ (f : femto)
    $1\ \mu\mathrm{L}\ (\mathtt{of\ bacteria}) \Longrightarrow 1\times 10^{9}\mathrm{cells}$
    As $1\ \mathrm{mL}$ of bacterial culture at saturation $\Longrightarrow$ pellet is $1\ \mu\mathrm{L}$

So for $1\ \mathrm{mL}$ of culture media, the E.coli cells are actually takes only 10^{-3} of the volume which is even less dense than our classroom.

  - Mass (of proteins? metabolites?):

Glucose/Nucleotide:	$3\times 10^2\ \mathrm{Da} \Longrightarrow \mathrm{Size}: (3\times 10^2)^{\frac{1}{3}} \approx 6\ \text{Å}$
Amino acid(a.a):	$10^2\ \mathrm{Da}$(a bit lighter (from about $60 \sim 200\ \mathrm{Da}$))

The a.a.'s average MW is $10^2\ \mathrm{Da}$ is because we calculate average in log, $\sqrt{60\times 200}\approx 100$. Further this is because the order of magnitude changes under log space.

Now we need how many a.a. in a protein to calculate the MW of proteins

  - How many a.a. in a protein?

$$\begin{cases} \text{lower bound} & 50\ \mathrm{a.a.} \\ \text{upper bound} & 5\times 10^3\ \mathrm{a.a.} \end{cases} \Longrightarrow \sqrt{2\times 10^5}\approx 300\ \mathrm{a.a.}$$

$$\Longrightarrow \approx 300\ \mathrm{a.a./protein} \Longrightarrow MW = 30\ \mathrm{kDa/mol}$$

$$\Longrightarrow \text{Num of protein per cell} = \frac{1\ \mathrm{pg}}{30\ \mathrm{kDa/mol}}\cdot\frac{6\times 10^{23}\ \mathrm{Da}}{\mathrm{mol}} \approx 2\times 10^7$$

The above result gives us a theoretical upper bound of the number of proteins within one e.coli, as we assume all molecules within one e.coli is protein. While commonly, for one type of protein, this number is from $10^3 \sim 10^4$

  - Concentration:
    What is the concentration of one molecule per cell?
    $1\ \mathrm{molecule/cell} \Longrightarrow \frac{1}{1\times 10^{-15}\ \mathrm{L}}\frac{\mathtt{mol}}{6\times 10^{23}} \approx 10^{-9}\ \mathrm{M} = 1\ \mathrm{nM}$

Metabolites	$10^6 \sim 10^7\ /\mathrm{cell} \Longrightarrow 1\sim 10\ \mathrm{mM}$
Protein	$10^3 \sim 10^4\ /\mathrm{cell} \Longrightarrow 1\sim 10\ \mu\mathrm{M}$

– Volume of Genome, protein
  Genome: E.coli got 5 Million base pairs
  If we first calculate about length:

$$5 \times 10^6 \text{ bp} \times 0.3 \text{ nm/bp} \approx 1 \times 10^6 \text{ nm} = 10^3 \text{ µm}$$

  This result indicate that the genome are so long that it needs to be folded at least $1,000$ times to put into E.coli. But what about its volume?
  Now we calculate its volume:

$$3 \times 10^2 \text{ Å}^3/\text{bp} \cdot 5 \times 10^7 \text{ bp} = 300 \cdot (1 \times 10^{-4} \text{ µm})^3 \cdot 5 \times 10^7 \approx 1.5 \times 10^{-2} \text{ µm}^2$$

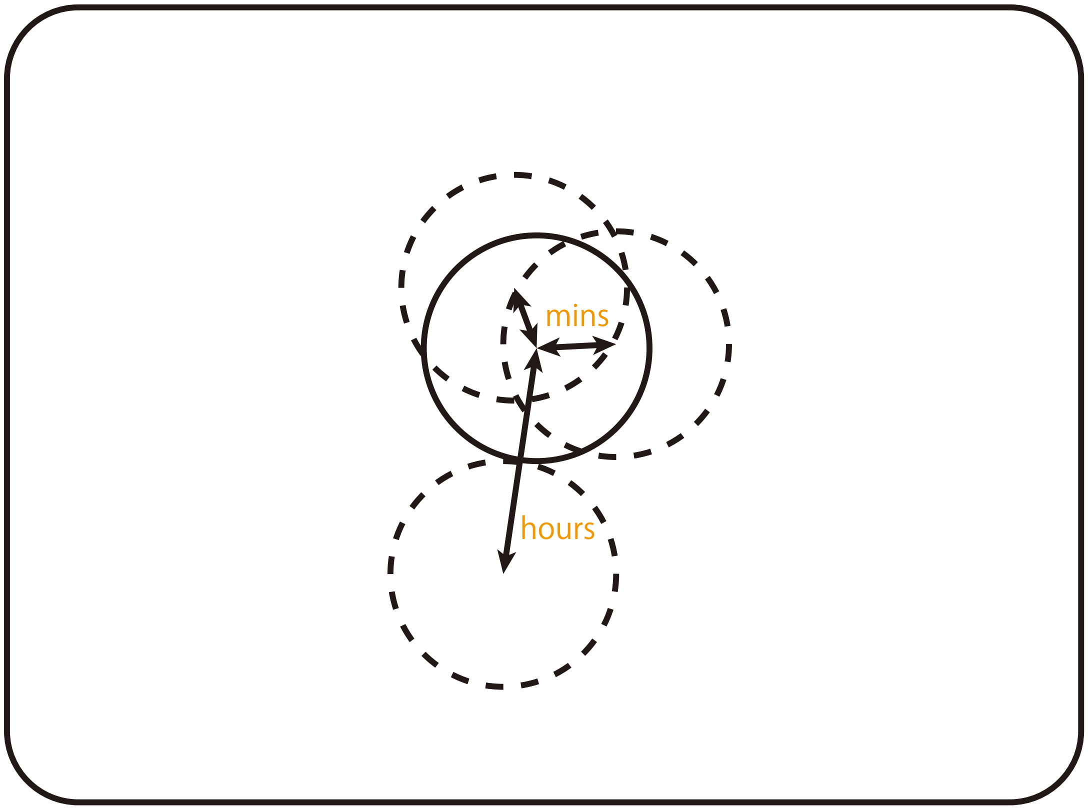
  So while long, Genome only take $\sim 1\%$of volume of a cell.
  What about protein's volume? Roughly one e.coli have about 10^7 proteins, thus:

$$300 \text{ a.a} \times 100 \text{ Å}^3 \times 10^7 \approx 0.3 \text{ µm}^3 \Longrightarrow 20 - 40\% \text{ of the cell}$$

  So unlike genome, proteins are crowded!

From above data and calculation we have built some sense of this micro-world, and static. But what about dynamics?

1.2.3 Rates in cells — Diffusion(How molecules move in cells)



Let's first get some sense of how diffusion happens for this microscopic world.

From our calculations for the heat diffusion case, we know the diffusion behavior has the following scaling relating τ, the duration, and x, the distance the diffusion behavior can reach.

- Diffusion scaling:
  $\frac{1}{\tau} = \frac{D}{x^2}$

The diffusion effect was first studied by Robert Brown, 1827. He put a pollen under water and observed under a microscope, and found the pollen jiggling without any apparent force. And now we can make his experiment as a reference to start up. Let's estimate the diffusion coefficient for pollen first.

- Infer the parameter for pollen.

  How big is a pollen particle? From our daily experience, while we can see the granules that contains lots of pollen particles, we cannot observe individual pollen particles directly by naked eye. However, Brown can observe a pollen particle under a microscope, a microscope in the 1800s so not a very large magnification. So we could estimate that a pollen particle should be around the size of an eukaryotic cell which can be easily observed under the microscope, so $\sim 10\ \mu\text{m}$.

  Some observation is that: When pollen jiggles under microscope, we can see the jiggles in seconds, so the movement should be roughly $\sim 1\ \mu\text{m}$ on seconds timescale. But most movements cancel out.

  To move a significant distance, around one body length $10\ \mu\text{m}$, requires much longer, since we do not see the pollen particles significantly move within minutes of observations. Therefore, $10\ \mu\text{m}$ movement requires about $1\ \text{h}$, so D can be estimated as
  $D = \frac{x^2}{\tau} = \frac{(10\ \mu\text{m})^2}{3600\ \text{s}} \approx 3 \times 10^{-3}\ \mu\text{m}^{-2}/\text{s}$.

Pollen provides a good beginning for us to scale down to proteins, so the key question is, how does D change when the particle size decreases? From our intuition, the smaller the particle, the faster it shall move. In other words, D is negatively correlated with L which denotes the particle size. We can guess $D \propto L^{-1}$, and it turns out to be true.

Why? Diffusion is balance between thermal force (water hitting on pollen) and drag in viscous(low Re,"skin friction drag")

$$\begin{cases} \text{Force} \propto L^2 \\ \text{Friction} \propto L\nu \end{cases} \qquad \text{(two forces are balanced)}$$

$$D = \mu K_B T \propto \mu \propto \frac{\nu}{F} \propto \frac{\nu}{L\nu} \propto L^{-1}$$

Now let's go from pollen to proteins

- Diffusion rate of protein
  Protein size: $(300)^{\frac{1}{3}} \approx 1\ \text{nm}$
  $\frac{D_{\text{protein}}}{D_{\text{Pollen}}} \propto \frac{L_{\text{pollen}}}{L_{\text{protein}}} = \frac{10\ \mu\text{m}}{1\ \text{nm}} = 1 \times 10^4 \Longrightarrow D_{\text{protein}} = 100\ \mu\text{m}^2/\text{s}$
  With the diffusion rate, now we can calculate how long it takes for a protein to move across different cells just by passive diffusion.

- Time for a protein to diffusion in different cell

E.coli:	1 µm	$t \approx \frac{x^2}{D} = \frac{(1\ \mu\mathrm{m})^2}{100\ \mu\mathrm{m}^2/\mathrm{s}} = 1 \times 10^{-2}$ s
Mammalian:	10 µm	$t \approx \frac{x^2}{D} \approx \frac{(10\ \mu\mathrm{m})^2}{100\ \mu\mathrm{m}^2/\mathrm{s}} = 1$ s
Eggs	100 µm	$t \approx \frac{x^2}{D} \approx \frac{(100\ \mu\mathrm{m})^2}{100\ \mu\mathrm{m}^2/\mathrm{s}} = 100$ s
Axon:	1 m	$t \approx \frac{x^2}{D} \approx \frac{(1\times10^6\ \mu\mathrm{m})^2}{100\ \mu\mathrm{m}^2/\mathrm{s}} = 10^{10}$ s $\approx 3 \times 10^2$ years

Thus we see for long cells like neurons, transferring proteins by passive diffusion becomes inefficient as $t \propto x^2$. This suggests that more efficient protein transport mechanisms must exist for these cells.

Chapter 2

Biochemical reaction networks and its dynamics

Scribe: Jiahe Wang + Yiqiao Deng + Zhi Zhou
Lecturer: Fangzhou Xiao
Lecture #2 20250911

Contents

2.1 Biochemical reactions in cells

Cellular changes occur through chemical reactions, which are fundamentally governed by **diffusion** kinetics. These reactions typically take place under the prevailing cellular conditions of **temperature** and **pressure**. Whether and when a reaction proceeds is largely determined by the presence of enzymes, which act as catalysts. From a thermodynamic perspective, many favorable reactions could take years to complete on their own (e.g., spontaneous glucose degradation, peptide bond cleavage, or ATP hydrolysis). However, enzymes accelerate such processes dramatically, often by factors of 10^{10} or more.

2.1.1 notations of (Bio)chemical reactions

Table 2.1: Reaction Types and Notations

Reaction Type	Notation	Meaning	Examples
Elementary	Straight Arrows ($\rightarrow$)	Implies reaction mechanism (all key players included)	E+S $\Leftrightarrow$ C $\rightarrow$ E+P; Gene + RNAP $\Leftrightarrow$ $C_0 \rightarrow C_1 \rightarrow ... \rightarrow C_n \rightarrow$ Gene + RNAP + mRNA
Composite	Squiggly Arrows ($\rightsquigarrow$)	May have hidden mechanism, reaction	S $\rightsquigarrow$ P; Gene $\rightsquigarrow$ Gene + mRNA; (S,P) $\rightsquigarrow$ (S-1,P+1)

"Elementary" "$\rightarrow$", implies reaction mechanism, meaning it's an irreducible reaction step, you CAN'T break it down further, or it's useless to do so for your concern

e.g. $E^+S \rightleftharpoons C \longrightarrow E^+P$
Gene + RNAP $\Leftrightarrow$ $C_0 \rightarrow C_1 \rightarrow ... \rightarrow C_n \rightarrow$ Gene + RNAP + mRNA

"Composite" "$\rightsquigarrow$" it may have Hidden mechanism, meaning it only describes the net change of several steps of reactions

e.g. S $\rightsquigarrow$ P
Gene $\rightsquigarrow$ Gene + mRNA;
(S,P) $\rightsquigarrow$ (S-1,P+1) (another notation for Composite, focused on change of molecular numbers)

Examples:

1. Cell replication, N becomes 2N: N $\rightsquigarrow$ 2N or N $\rightsquigarrow$ N+1
2. Photon doubles in a laser: photon + atom(activated) $\rightarrow$ 2photon + atom(ground)
3. Diffusion into a cell through a passive channelon membrane: $A_{out} \rightarrow A_{in}$
4. ATPase pumping H^+ and of the cell: $H^+{}_{in} \rightsquigarrow H^+{}_{out}$

2.1.2 Estimates of enzymatic reaction rates

Enzymatic reactions occur when an enzyme and its substrate collide and then bind; the enzyme's atoms then orient the substrate in specific ways, ultimately transforming it into a product with atoms in different positions. Thus, there's a two-step process: binding and catalysis.

$$E + S \Leftrightarrow C \rightsquigarrow E + P$$

The fastest reaction rate is limited by association rate (react immediately after association), which is governed by diffusion, so called **"diffusion limited on rate"**.

Behavies just like heat diffusion:

flux at a point on E within a surface:

$$j(a) = D\frac{\partial C}{\partial R}$$

Roughly,

$$j(a) = D\frac{\partial C}{\partial R}|_a \backsim \frac{C_\infty}{a} \Rightarrow j = D\frac{C_\infty}{a}$$

More details: cons. of mass ⇒ $4\pi\, R^2 j$ is constant, so

$$j \propto \frac{1}{R^2}\frac{\partial C}{\partial R} = j \propto \frac{1}{R^2} \Rightarrow C(R) = \frac{A}{B} + C$$

A,B is constant, Boundary condition: C(a)=0, C(∞)=C∞, so B=C∞, A=-aC∞

$$J = j \cdot 4\pi a^2 = 4\pi Da \cdot C_\infty \;\Rightarrow\; K_{on}C_\infty \tag{2.1}$$

$$k_{on} = 4\pi Da \;\backsim\; 10 \cdot 10^2\,\mu m^2/s \cdot 1\,nm \cdot \frac{6 \cdot 10^{23}}{\text{mol}} \cdot \frac{1L}{10^{15}\,\mu m^3} \;\backsim\; 10^9\,s^{-1}\,mol^{-1}$$

So, if $C \backsim 1\,\text{mM}$, then it's about 10^6 reactions/s per enzyme.

However, most enzymes operate much more slowly, being bottlenecked by the catalysis step, typically around 10—10^2 reactions/s.
Note that for elementary reactions, i.e., reactions that occur immediately upon collision, the flux of $E + S \to C$ is given by $K_{on}ES$ (where E and S denote the concentrations), since each enzyme reacts at a rate of $K_{on}S$. This relationship is called the **law of mass action**, and it describes how the flux scales with reactant concentrations.

Detailed Derivation of Diffusion-Limited Flux

Based on the steady-state diffusion model, we consider an enzyme E as the center point, with substrate molecules diffusing throughout the solution. Under steady-state conditions, the flux through any spherical surface of radius R centered at the enzyme remains constant, i.e., $4\pi R^2 \cdot j = \text{constant} \neq 0$. This conservation relation indicates that the amount of substrate C consumed at the enzyme surface is exactly balanced by the amount diffusing into the sphere from the external environment.

Derivation of the Spherically Symmetric Diffusion Equation The general diffusion equation (Fick's second law) in three dimensions is:

$$\frac{\partial C}{\partial t} = D\nabla^2 C$$

where ∇^2 is the Laplace operator.

In spherical coordinates (r, θ, φ), the Laplace operator is expressed as:

$$\nabla^2 C = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial C}{\partial r}\right) + \frac{1}{r^2 \sin\theta}\frac{\partial}{\partial \theta}\left(\sin\theta\frac{\partial C}{\partial \theta}\right) + \frac{1}{r^2\sin^2\theta}\frac{\partial^2 C}{\partial \varphi^2}$$

Due to the spherical symmetry of the problem (enzyme is spherical and substrate diffusion is isotropic), the concentration C depends only on the radial distance r:

$$C = C(r), \quad \frac{\partial C}{\partial \theta} = 0, \quad \frac{\partial^2 C}{\partial \varphi^2} = 0$$

Substituting these conditions into the Laplace operator gives:

$$\nabla^2 C = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial C}{\partial r}\right)$$

Under steady-state conditions ($\frac{\partial C}{\partial t} = 0$, i.e., $4\pi R^2 \cdot j =$ constant (non-zero)), the diffusion equation simplifies to:

$$D \cdot \frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{dC}{dr}\right) = 0$$

Since $D \neq 0$, we obtain the spherically symmetric steady-state diffusion equation:

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{dC}{dr}\right) = 0$$

Solving the Diffusion Equation Multiplying both sides by r^2 and integrating once:

$$\frac{d}{dr}\left(r^2\frac{dC}{dr}\right) = 0 \quad \Rightarrow \quad r^2\frac{dC}{dr} = A$$

where A is an integration constant.

Rearranging and integrating again:

$$\frac{dC}{dr} = \frac{A}{r^2} \quad \Rightarrow \quad C(r) = -\frac{A}{r} + B$$

where B is another integration constant.

Applying Boundary Conditions Applying the boundary conditions:

- At the enzyme surface ($r = a$): $C(a) = 0$
- At infinity ($r \to \infty$): $C(\infty) = C_\infty$

Substituting the boundary conditions to solve for the integration constants:

$$\begin{cases} 0 = -\frac{A}{a} + B \\ C_\infty = B \end{cases} \Rightarrow A = aC_\infty, \quad B = C_\infty$$

Thus, the concentration distribution is:

$$C(r) = C_\infty\left(1 - \frac{a}{r}\right)$$

Calculating Flux and Reaction Rate Calculating the concentration gradient:

$$\frac{\partial C}{\partial r} = \frac{aC_\infty}{r^2}$$

The gradient at the enzyme surface ($r = a$):

$$\left.\frac{\partial C}{\partial r}\right|_{r=a} = \frac{C_\infty}{a}$$

According to Fick's first law, the flux is:

$$j = D \cdot \frac{C_\infty}{a}$$

The total reaction rate is the flux multiplied by the enzyme surface area:

$$J = j \cdot 4\pi a^2 = 4\pi D a C_\infty$$

Defining the reaction rate constant $k_m = 4\pi D a$, we obtain $J = k_m C_\infty$.

This derivation provides a rigorous mathematical foundation for understanding diffusion-limited reaction kinetics, following the approach consistent with the file's content.

How to understand $j = D\frac{\partial c}{\partial R}$?

From the perspective of dimensional analysis: j represents the number of molecules passing per unit area per unit time.

- Dimension of j: $\frac{\text{molecules}}{\text{area}\cdot\text{time}} = \frac{N}{L^2 \cdot T}$
- Dimension of D: diffusion coefficient, $\frac{\text{length}^2}{\text{time}} = \frac{L^2}{T}$
- Dimension of $\frac{\partial c}{\partial R}$: concentration gradient, $\frac{\text{concentration}}{\text{length}} = \frac{(N/L^3)}{L} = \frac{N}{L^4}$

Therefore, dimensional verification of $j = D\frac{\partial c}{\partial R}$:

$$[D] \cdot \left[\frac{\partial c}{\partial R}\right] = \frac{L^2}{T} \cdot \frac{N}{L^4} = \frac{N}{L^2 \cdot T} = [j]$$

The dimensions are consistent, validating the physical relationship.

2.1.3 Dynamics of chemical reaction networks(CRN)

A generic chemical reaction (Elementary or Composite)

$$\alpha_1 X_1 + \alpha_2 X_2 + ... + \alpha_n X_n \rightarrow \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n \alpha_1 X_1 + \alpha_2 X_2 + ... + \alpha_n X_n \rightsquigarrow \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n$$

α stand for reaction stoichiometry, β stand for product stoichiometry
Reaction stoichiometry: $\gamma_j = \beta_j - \alpha_j (j = 1, 2, 3, ..., n)$
Net change notation: $(X_1, X_2, ..., X_n) \rightarrow (X_1 + P_1, X_2 + P_2, ..., X_n + P_n)$
Every time this reaction happens, x_j~ flux caused by this reaction is $\gamma_j * \nu$

Several reactions form a network (GRN)

m stand for reactions, $i = 1, 2, ..., m$
n stand for species, $j - 1, 2, 3, ..., n$

$$\alpha_{i1} X_1 + \alpha_{i2} X_2 + \cdots + \alpha_{in} X_n \quad \rightarrow \quad \beta_{i1} X_1 + \beta_{i2} X_2 + \cdots + \beta_{in} X_n$$

$$\frac{d}{dt}x_j = \sum_i \gamma_{ij}\nu_i = \sum_{i:\gamma_{ij}>0} V_{ij}\nu_i - \sum_{i:\gamma_{ij}<0} V_{ij}\nu_i = f_j^+ - f_j^- = f_j$$

$$\frac{dx}{dt} = \Gamma\nu = f^+ - f^-$$

Note that this hold always, for both elementary and composite, Since all we're done is "accounting", for where molecules went. Only depends on γ, not α or β, so "net change notation" is enough. We can write this even if we don't know the regulation mechanism. But to have a full description of the dynamics.

Does ν depend on x ? i.e. reaction kinetics

Kinetics: Law of mass actin Elementary reactions' kinetics follow the law of mass action:

$$\alpha_1 X_1 + \alpha_2 X_2 + \cdots + \alpha_n X_n \ \xrightarrow{k \text{ (reaction rate constant)}} \ \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$$

$$\nu(x) \ = \ \text{reaction rate} \ = \ \text{flux} \ = \ k\, X_1^{\alpha_1} X_2^{\alpha_2} \cdots X_n^{\alpha_n}$$

e.g. $E + S \xrightarrow{K} C, rate = K_{on}ES(K = K_{on})$

Now with several reactions forming a network

$$\alpha_{i1} X_1 + \alpha_{i2} X_2 + \cdots + \alpha_{in} X_n \ \xrightarrow{k_i} \ \beta_{i1} X_1 + \beta_{i2} X_2 + \cdots + \beta_{in} X_n$$

$$\nu_i \ = \ k_i X_1^{\alpha_{i1}} \cdots X_n^{\alpha_{in}} \ = \ k_i X^{\alpha_i}$$

$$\frac{d}{dt}x_j \ = \ \sum_{i:\gamma_{ij}>0} \gamma_{ij}\nu_i \ - \ \sum_{i:\gamma_{ij}<0} |\gamma_{ij}|\nu_i \ = \ f_j^+(x) - f_j^-(x)$$

$$\frac{dx}{dt} \ = \ \Gamma\nu(x) \ = \ \Gamma \wedge_k x_\alpha \ = \ f(x) \ = \ f^+(x) - f^-(x)$$

2.1.4 Analysis of 1D and 2D dynamics by phase portrait

These are autonomous dynamical systems $\frac{dx}{dt} = f(x)$. How to understand their dynamics? We start simple from low dimensions:

Exp1

$$\frac{dx}{dt} = f(x) = -x$$

Fixed point: x^* s.t. $F(x^*) = 0$
Feature of $f(x) = -x$

Fixed point: You **DON'T** change if you start from a fixed point, if you start at a different initial condition, you'll go to $x = 0$ until you are stable at a fixed point.

Exp2

$$\frac{dx}{dt} = f(x) = x^3$$

Fixed point: x^* s.t. $F(x^*) = 0$
Feature of $f(x) = x^3$ Fixed point: You **DON'T** change if you start from Fixed point, but you'll go further and further if you're not

Exp3

$$\frac{dx}{dt} = f(x) = x^2 - r$$

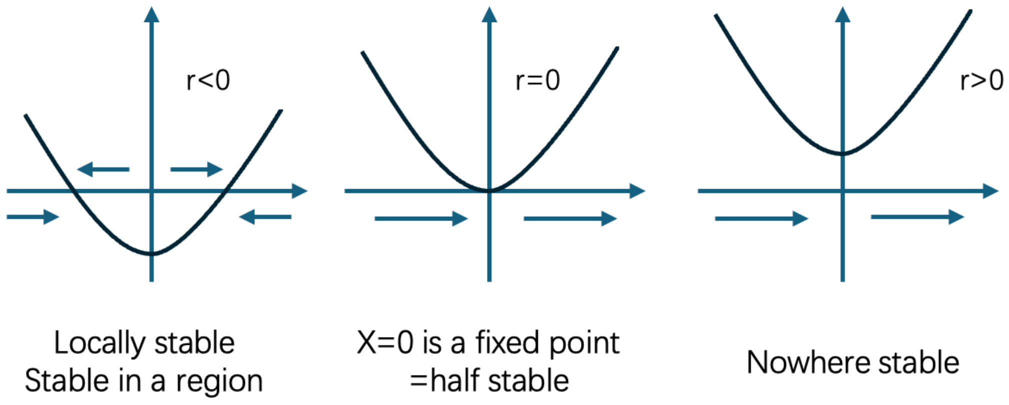


Figure 2.1 Exp3 $f(x) = x^2 - r$

1D can only define stable or unstable, "go away" or "coverage", because trajectory is 1D as well, and the point **CAN'T go back**

Exp4 Bio product minus degradation

$$\dot{x} = \mu - x$$

Fixed point: $x = \mu$ ***Exp5***: bistable

$$\dot{x} = f^+(x) - x$$

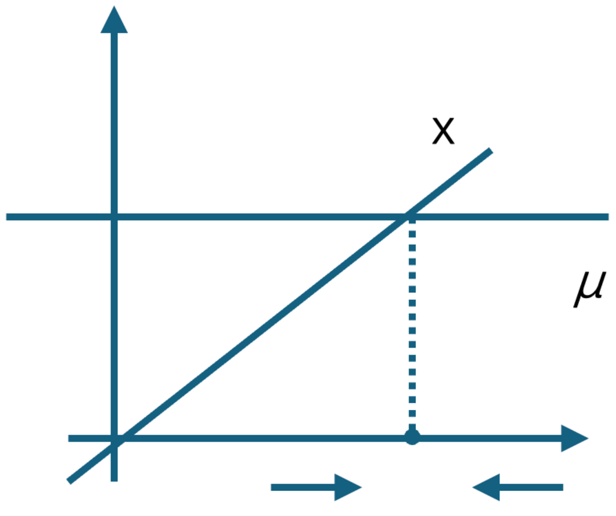


Figure 2.2 $\dot{x} = \mu - x$

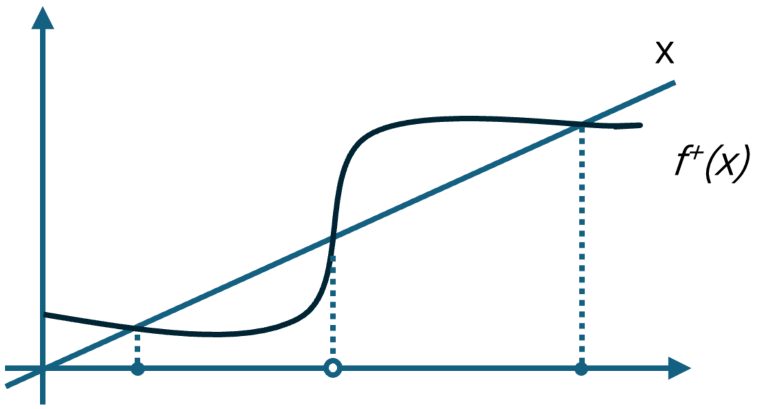


Figure 2.3 $\dot{x} = f^{+}(x) - x$

How to create a 1D (Bio) bistable system?

2D system In 2D system trajectories **can "go back"** along any axis, but trajectories **can't overlap itself** On top of fixed points: $f_1(X^*) = f_2(x^*) = 0$, also nullelines $f_1(X^*) = 0 | f_2(x^*) = 0$, These are enough to determined dynamics.

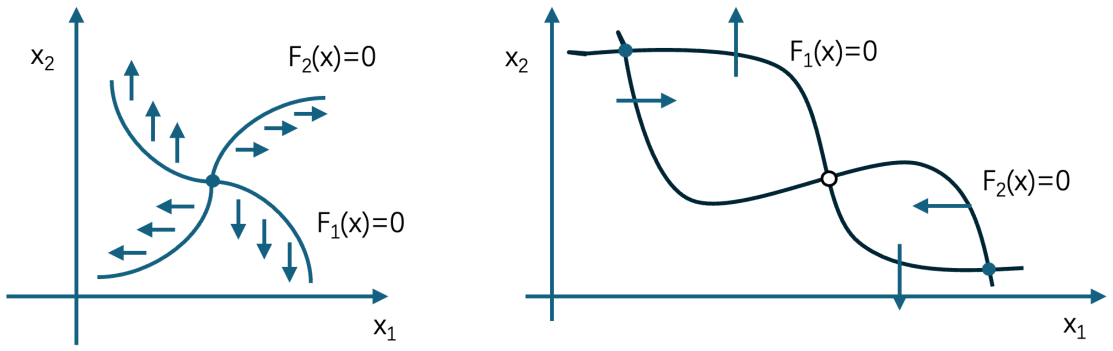


Figure 2.4

2.1.5 Types of dynamics in 2D

Dynamics about a fixed point graph needed

New in 2D: center *Exp1* Spring question

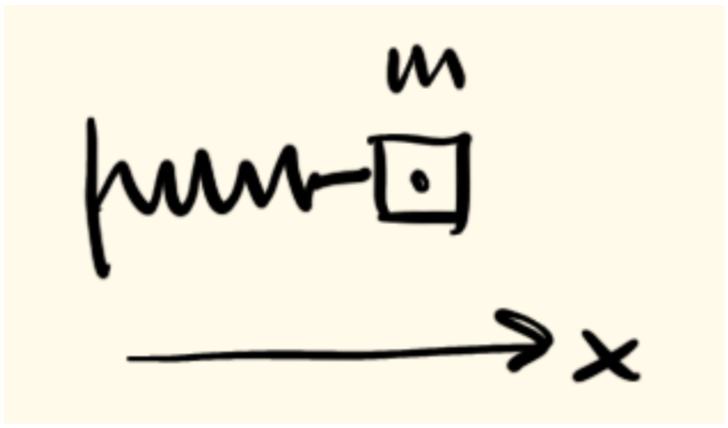


Figure 2.5 Spring

$$F = m\ddot{x}F = -kx \Rightarrow m\ddot{x} + kx = 0 \Rightarrow x_1 = x_1,\ x_2 = \dot{x}\ \text{,so } m\dot{x}_2 + kx_1 = 0\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Trajectories neither spiral in or out, $x = 0$ is a **center**, not stable

Also, we can look at energy, which is conserved:

$$\frac{1}{2}mx_2^2 + \frac{1}{2}kx_1^2 = E\frac{dE}{dt} = mx_2\dot{x}_2 + mx_1\dot{x}_1 \Rightarrow \frac{dE}{dt} = -mx_2\frac{k}{m} + kx_1x_2 \Rightarrow -kx_1x_2 + kx_1x_2 = 0$$

If there's friction:

$$m\ddot{x} + k_f\dot{x} + k_sx = 0 \tag{2.2}$$

Define the state variables $x_1 = x$, $x_2 = \dot{x}$. Then

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_s/m & -k_f/m \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{2.3}$$

The total mechanical energy is

$$E = \tfrac{1}{2}mx_2^2 + \tfrac{1}{2}k_s x_1^2. \tag{2.4}$$

Taking the time derivative gives

$$\frac{dE}{dt} = -mx_2\big(\tfrac{k_s}{m}x_1 + \tfrac{k_f}{m}x_2\big) + k_s x_1 x_2 = -k_f x_2^2 < 0. \tag{2.5}$$

means the circle shrunks overtime.

3D and beyond

3d-system Trajectories can roam freely, no constraint anymore due to dimensions, e.g. graph needed The best way to analyze is to **stimulate** and to **look**, lack of structure, **Nothing** very useful can be said.

system beyond 3D To say something for >=3 dimensions, still by dimension reduction
⇒ Reduce to 0 dimension: fixed point
⇒ Reduce to 1 dimension: fixed point, limit cycles
⇒ Reduce to 2 dimensions: then analyze on 2D....

2.1.6 Local stability analysis

Local dynamics in general ⇒ Local is always Linear
assuming $\frac{d}{dt}(xt\Delta x) = f(x)$ ⇒ consider small perturbationΔx around x

$$\frac{d}{dt}(xt\Delta x) = \frac{d}{dt}\Delta x = f(x + \Delta x) \approx f(x) + \frac{\alpha f}{\alpha x}(x)\Delta x$$

Assuming at $x = x^*$, a fixed point, so $f(x^*) = 0$

$$\frac{d\Delta x}{dt} = A\Delta x \Rightarrow A = \frac{\alpha f}{\alpha x}(x^*)$$

$$\frac{dx}{dt} = ax \quad \Rightarrow \quad x(t) = x(0)e^{at}$$

Eigenvector (λ, v) s.t. $Av = \lambda v$, let $x(0) = y(0)v$, then:

$$\frac{1}{\Delta t}\big(x(\Delta t) - x(0)\big) = Ax(0)$$

$$\frac{1}{\Delta t}\big(y(\Delta t) - y(0)\big)v = Avy(0) = \lambda vy(0)$$

$$\Rightarrow \frac{dy}{dt} = \lambda y \quad \Rightarrow \quad y(t) = y(0)e^{\lambda t}$$

$$\Rightarrow x(t) = y(0)e^{\lambda t}v$$

So, behaviors can be decomposed in terms of Eigenvector

When $\mathtt{Re}(\lambda) > 0$, Eigenvector is unstable, When $\mathtt{Re}(\lambda) < 0$, Eigenvector is stable
A is **Hurwitz**: When $\mathtt{Re}(\lambda) < 0$, for all $\lambda \in \mathtt{Eigenvector}(A)$, this guarantees $x = 0$ is stable.

Chapter 3

Diversity of bioregulation by time scale separation and Adaptation biomachines

Scribe: Jiacheng Wei + Chengqian Li + Xia Yao
Lecturer: Fangzhou Xiao
Lecture #4 20250925

Contents

3.1 The explanation of chemical reaction network equation

At first, let us review how we describe a system of chemical reaction network (CRN). According to our lecture, we have equation below:

$$\dot{\vec{x}} = \frac{dx}{dt} = \Gamma \Lambda_k x^{\alpha}$$

But this equation looks confusing because it lacks an explanation of each character. Let us write this equation in a more understandable form:

$$\dot{\vec{x}} = \frac{d\vec{x}}{dt} = \Gamma \Lambda_k \vec{x}^{\alpha}$$

Here, $\dot{\vec{x}}$ means derivative of x with respect to t, so it is equal to dx / dt. And x is called by me the **concentration vector of biomolecules** because it represents every concentration of biomolecules we consider. It can be written as $[x_1, x_2......x_n]^T$. So $\dot{\vec{x}}$ means the conception change of every biomolecule we consider, it can be written as $[dx_1/dt, dx_2/dt......dx_n/dt]^T$.

For Γ, it is the uuppercase form of gamma (γ), it is called **the matrix of stoichiometric number change** by me. To understand this, we must first figure out the definition of "stoichiometic number chnange". For a chemical reaction:

$$4x_1 + x_2 = 2x_1 + 3x_2$$

The coefficient in front of the variable x_1 and x_2 is called the stoichiometric number. To make a difference of the coefficient in front of these reactants and products, let us use $_1$ to represent the coefficient "4" in front of reactant x_1, use $_2$ to represent coefficient "1" in front of reactant x_2. Similarly, we can also use $_1$ and $_2$ to represent cofficient in front of products x_1 and x_2. We notice the x_1 and X_2 appear in both reactants and products, so we can then definite another physical quantity γ, for γ_1, it equals to $\alpha_1 - \beta_1$, for γ_2, it equals to $\alpha_2 - \beta_2$. The γ shown here is "stoichiometic number chnange". For every variable, or biomolecules we consider, they have a unique γ. And when $\gamma < 0$, that means after this reaction happens, the quantity of this biomolecule will decrease. For example, for biomolecule x_1, after one single reaction happens, it will consume $4x_1$ biomolecules and generates 2, so it consumes $2x_1$ in total. And for $\gamma > 0$, that means after this reaction happens, the quantity of this biomolecule will increase. Notably, for many biochemical reactions or even chemical reactions that occurred in tubes or reaction kettles, the molecules will rarely appear on both both reactant side and the product side, because they will be invited. One example is alkaline hydrogen oxygen fuel cell, the positive pole half reaction is:

$$2H_2 - 4e^- + 4OH^- = 4H_2O$$

and the negative pole half reaction is:

$$O_2 + 4e^- + 2H_2O = 4OH^-$$

when we consider the whole reaction of this battery, we just simply combine this 2 half reactions together, and it becomes:

$$2H_2 + O_2 + 2H_2O = 4H_2O$$

For water, it appears in both reactant side and product side, we can divide $2H_2O$ from both side, so we get:

$$2H_2 + O_2 = 2H_2O$$

but we can also save the divided 2 water molecules in both sides, because they have practical significance, it means 2 water molecules is consumed in negative pole, and 4 water molecules generated in positive pole. For water, its α equals to 4, its β equals 2, and its γ equals 4-2=2.

So, for "the matrix of stoichiometric number change", it is the matrix formed by γ, in other words, every element in this matrix is γ.

So how do these γ arrange? We first need to definite the chemical reaction network (CRN). Now this CRN can be written as:

$$\alpha_{11}x_1 + \alpha_{12}x_2 + ... + \alpha_{1n}x_n = \beta_{11}x_1 + \beta_{12}x_2 + \beta_{1n}x_n$$
$$\alpha_{21}x_1 + \alpha_{22}x_2 + ... + \alpha_{2n}x_n = \beta_{21}x_1 + \beta_{22}x_2 + \beta_{2n}x_n$$
$$......$$
$$\alpha_{m1}x_1 + \alpha_{m2}x_2 + ... + \alpha_{mn}x_n = \beta_{m1}x_1 + \beta_{m2}x_2 + \beta_{mn}x_n$$

Here, M and N are positive integers.

For the first row of the matrix, it represents the first reaction of CRN written above, it is:

$$[\gamma_{11}, \gamma_{12}......\gamma_{1n}]$$

The second row of the matrix is:

$$[\gamma_{21}, \gamma_{22}......\gamma_{2n}]$$

The last row, which is the number m row, is:

$$[\gamma_{m1}, \gamma_{m2}......\gamma_{mn}]$$

For Λ_k, it is called **diagonal matrix of reaction rate constant**. In linear algebra, we often use Λ to represent a diagonal matrix. The diagonal matrix means every element don't locate in diagonal line is zero. In other words, for element a_{ij} in matrix $A_{m\times n}$, where $1 \leqslant i \leqslant m$, $1 \leqslant j \leqslant n$, unless i=j, $a_{ij} = 0$. But that doesn't mean when i=j, $a_{ij} \neq 0$. Another question is what is "reaction rate constant", that means for a reaction $4x_1 + x_2 = 2x_1 + 3x_2$, according to the Law of mass action the reaction rate equals to $v = k \times x_1^4 \times x_2$, the k shown here is "reaction rate constant". So, for this diagonal matrix, it can be written as k_1, k_2......k_n located on the diagonal line.

The last one in this equation needed to explain is $\vec{x}^{\alpha}$, I call it **mass action expression vector**. For every element in this vector, its form is:

$$x_{i1}^{\alpha_{i1}} + x_{i2}^{\alpha_{i2}} + ... + x_{in}^{\alpha_{in}}$$

with $1 \leqslant i \leqslant m$.

This so-called "mass action expression vector" is formed by mass action expression of each reaction in CRN. And the first reaction occupies the first row of this vector, the second reaction occupies the second row of this vector...... In other words, if the reaction "$\alpha_{11}x_1 + \alpha_{12}x_2 + ... + \alpha_{1n}x_n = \beta_{11}x_1 + \beta_{12}x_2 + \beta_{1n}x_n$" is represented by "reaction 1", the mass action expression of reaction 1 is $x_{11}^{\alpha_{11}} + x_{12}^{\alpha_{12}} + ... + x_{1n}^{\alpha_{1n}}$. If the mass action expression of reaction 1 is represented by "M1", the "mass action expression vector" can be written as:

$$[M_1, M_2......M_m]^T$$

That is all the characters participated in the equation describing the chemical reaction network (CRN). And now I will briefly introduce how can we get this equation. Now we only consider there is one reaction in the CRN, the CRN can be described by:

$$\alpha_1 x_1 + \alpha_2 x_2 = \beta_1 x_1 + \beta_2 x_2$$

What does $d\vec{x}/dt$ mean here? It is $[dx_1/dt, dx_2/dt]^T$, so we solve these two elements of this vector separately. For dx_1/dt, it equals:

$$dx_1/dt = \gamma_1 \times v_1 = \gamma_1 \times k_1 \times x_2^{\alpha_1} \times x_2^{\alpha_2}$$

The first "=" above comes from the relationship between molecule changing rate and reaction rate. Considering in 1 second, the reaction happens N times, and the γ represents when there is 1 reaction happens, the molecules change (increase/decrease) is $\gamma = \alpha - \beta$, the molecule changes is $\gamma \times N$. Notably, the γ can indicate whether molecules increase or decrease, as we discussed above. The second "=" is the Law of mass action.

When the reaction number increases to m, the total molecule number increase to n, the equation of chemical reaction network becomes:

$$\dot{x} = \frac{d\vec{x}}{dt} = \Gamma \Lambda_k \vec{x}^{\alpha}$$

3.2 Time Scale Separation

What is the meaning of "Time Scale Separation"? I think it is the thought to consider those "fast reaction" and "slow reaction". The meaning of "fast reaction" is that it can become chemical equilibrium quickly. For example, we consider the enzyme catalysis process:

$$E^+S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_{cat}} E^+P$$

Here we consider the first step (binding reaction). In this step, Enzyme (E) binds substance (S) and becomes complex ES, and ES can dissociate into E and S. In other words, this step is reversible. For any reversible chemical reaction, we can use reaction equilibrium constant K (uppercase) to describe how thorough is the reaction, which is a thermodynamic parameter. And there is also reaction rate constant k (lowercase) to describe how fast can the reaction go, which is a kinetic parameter and obey the Law of Mass Action.

$$K = \frac{k_1}{k_{-1}} = \frac{c(E) \times c(S)}{c(ES)}$$

For the dissociation constant K_d, it describes how easy the complex ES can dissociate, which is the reciprocal of K.

$$K_d = \frac{1}{K}$$

The second step (catalytic reaction) is the complex ES becomes E and product (P). This process is considered as irreversible, so there is only one reaction rate constant called K_{cat} representing the k of catalysis.

When we use the principle of "Time Scale Separation", we consider the binding reaction as fast reaction, and the catalytic reaction as slow reaction. That means we can consider the binding reaction has reached chemical equilibrium. Notably, this opinion (fast reaction has reached chemical equilibrium) is used by Michaelis and Menten in 1913 to explain enzyme kinetics. And in 1925, G.E.Briggs and James B.S.Haldane use steady-state approximation (SSA) to explain enzyme kinetics, and SSA is considered as a better model in biochemistry textbook. In SSA, it no longer considers the binding reaction reaches equilibrium, it considers the concentration of ES complex is not changed, in other words, it has $\frac{dc(ES)}{dt} = 0$, so it satisfies the following equation:

$$k_1 \times E \times S = k_{-1} \times ES + k_{cat} \times ES$$

But in the following part, we use "Time Scale Separation", that means we should remember the binding reaction is in equilibrium, and this equation is true in any case.

$$K = \frac{E \times S}{ES}$$

$$K = \frac{E \times S}{C}$$

Here, C and ES has the same meaning, they are the concentration of enzyme-substance complex. **Here, I am not sure why Fangzhou thinks here he uses Quasi-steady state assumption (QSSA), I think QSSA means $\frac{dc(ES)}{dt}0$, but it changes very slow, much slower than the changes of S, in other words,** $\frac{dc(ES)}{dt} \ll \frac{dc(S)}{dt}$.

With this background knowledge, **let us start to understand Fangzhou's regime thoery!** Let us start with a 2-dimension graph, the x-axis is S_{tot}, which means it is the total substance concentration. For an enzyme kinetics assay, it is the final substance concentration you add in the tube. The y-axis is the E_{tot}, which means the total concentration of enzyme. The graph can be divided into three regimes, each regime has distinct biological meaning.

Interestingly, the term "regime" carries profound religious connotations, originally referring to the specific parishes overseen by missionaries from different churches during the Middle Ages. This is similar to the word "dogma" in the central dogma—which initially denoted doctrines in the Bible. Likewise, "transcript" originally referred to the copying of religious scriptures; "translation" initially meant translating religious texts, such as rendering written Latin religious works into spoken English; and "canonical" derives from the religious term "canon," meaning authoritative texts or norms.

So why this graph divides the whole space as three regimes? Fangzhou pointed out the restrictions of every regime, that is like the version of the Bible used by missionaries in this parish (regime).

In regime 1, the restriction is $E_{tot} = E, S_{tot} = S$. The biological meaning of this regime is the binding affinity of enzyme and substance is low, there is very little ES complex in the solution. Because $E_{tot} = E$ and $E_{tot} = E + C$, E represents free enzyme that doesn't bind substance, C represents ES complex, as described above. So we can get $E \gg C$ and $E_{tot} \gg C$. Similarly, we can get $S \gg C$ and $S_{tot} \gg C$. Now we focus on the equilibrium equation $K_d = \frac{E \times S}{C}$, we can get $\frac{S}{K_d} = \frac{C}{E}$. And because we have $E_{tot} = E, S_{tot} = S$ in regime 1, we can get $\frac{S_{tot}}{K_d} = \frac{C}{E_{tot}}$. And because we previously proved $E_{tot} \gg C$, we have $\frac{S_{tot}}{K_d} = \frac{C}{E_{tot}} \ll 1$, that is $S_{tot} \ll K_d$. Similarly, we can also get $E_{tot} \ll K_d$. So, we know when we set the restriction $E_{tot} = E, S_{tot} = S$, it equals to $S_{tot} \ll K_d$ and $E_{tot} \ll K_d$, and it occupies the region surrounded by x-axis, y-axis, $x = K_d$ and $y = K_d$. In my figure, I set K_d as 1.

In regime 2, the restriction is $E_{tot} = E, S_{tot} = C$, we need to figure out where regime 2 located in 2D graph, And the answer is $E_{tot} \gg K_d$ and $E_{tot} \gg S_{tot}$. Here I will get the proof process. Because $E_{tot} = E$ and $E_{tot} = E + C$, we have $E_{tot} \gg C$. And because $S_{tot} = C$, we prove $E_{tot} \gg S_{tot}$. Then is the other border. Because $S_{tot} = S + C$ and $S_{tot} = C$, we get $C \gg S$. And because of the equilibrium equation $K_d = \frac{E \times S}{C}$, we can get $\frac{E}{K_d} = \frac{C}{S}$. Because we get $C \gg S$ before, we have $\frac{E}{K_d} \gg 1$, that is $E \gg K_d$, and that becomes $E_{tot} \gg K_d$. So we get two borders: $E_{tot} \gg K_d$ and $E_{tot} \gg S_{tot}$. That means the y-axis, $y = K_d$ and $x = y$ is the border of regime 2, as my figure shows. The regime 2 shows there are many enzyme, so almost all substance is binding with enzyme, and there is still many free enzyme left. In other words, in this tube, the dominant component is free enzyme, then the less one is enzyme-substance complex, and there is very little free substance here.

In regime 3, the restriction is $E_{tot} = C$ and $S_{tot} = S$. This restriction equals $S_{tot} \gg K_d$ and $S_{tot} \gg E_{tot}$. The proof process is the same as regime 2, just changes character E to character S. And in regime 3, the biological meaning is there are many substance in the tube, and almost all enzyme is binding with substance. The concentration of component is $S \gg ES \gg E$, and in regime 2 is $E \gg ES \gg S$.

So, we can draw the graph with 3 regimes now! It is as in Figure 1.

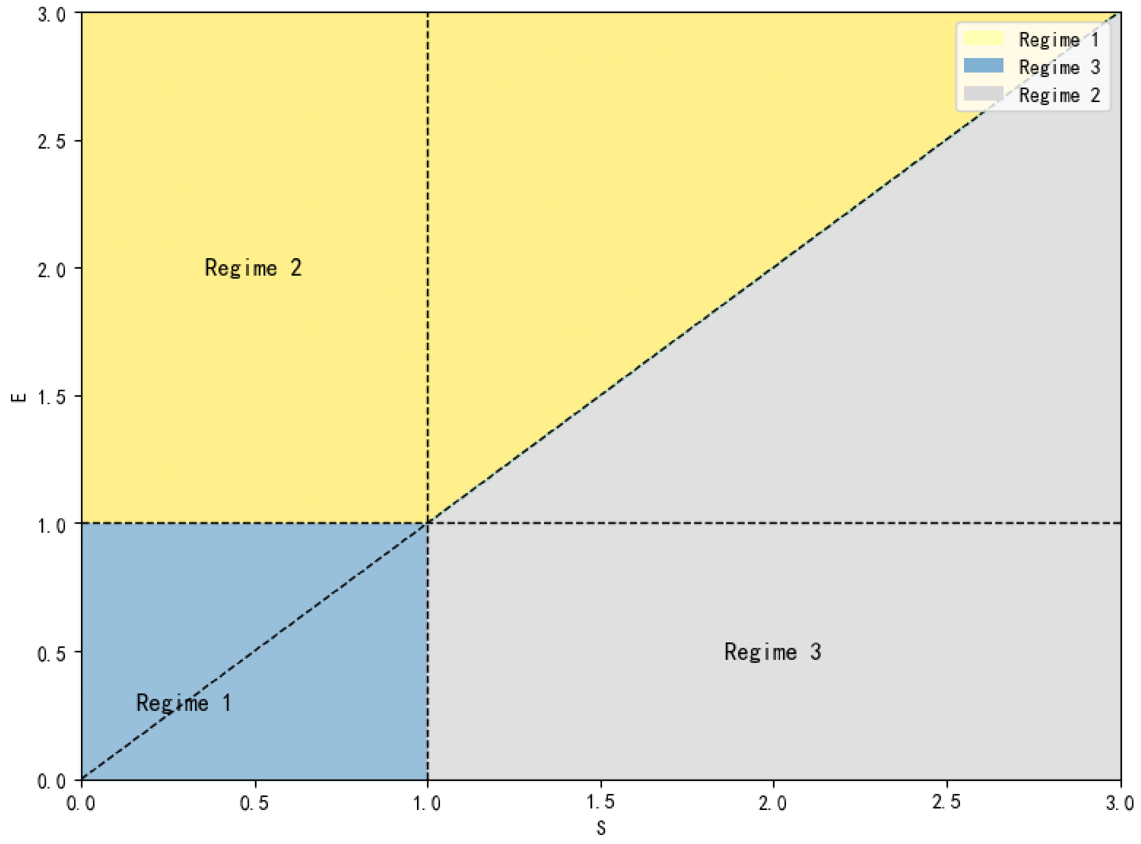


Figure 3.1 Three Regimes

3.3 “LEGO” of Bioregulation

After we analyzed the 3 regimes, we can try to find what behaviors will have in this enzyme kinetic system. **In our class, Fangzhou gives us 3 situations, or so-called ”LEGO”. They are saturation, bottleneck and ultrasensitivity.** Now let us discuss each of them!

For saturation and bottleneck, they appear in the 2D graph, whose x-axis is S_{tot} and y-axis is C_{tot}, or call it ES_{tot}.

Let us talk about saturation now: imagine a line crossing regime 1 and regime 2, that means we don’t change the total enzyme concentration, and increase the concentration of total substance. That is like we are doing a substance tritation assay to draw the Michaelis-Menten function. For this assay, the x-axis is the concentration of substance, that is the same as our figure, the S_{tot}. And the y-axis is the reaction rate, which has the relationship $v = k_{cat} \times C$ or $v = k_{cat} \times c(ES)$, that is the same as our y-axis, too.

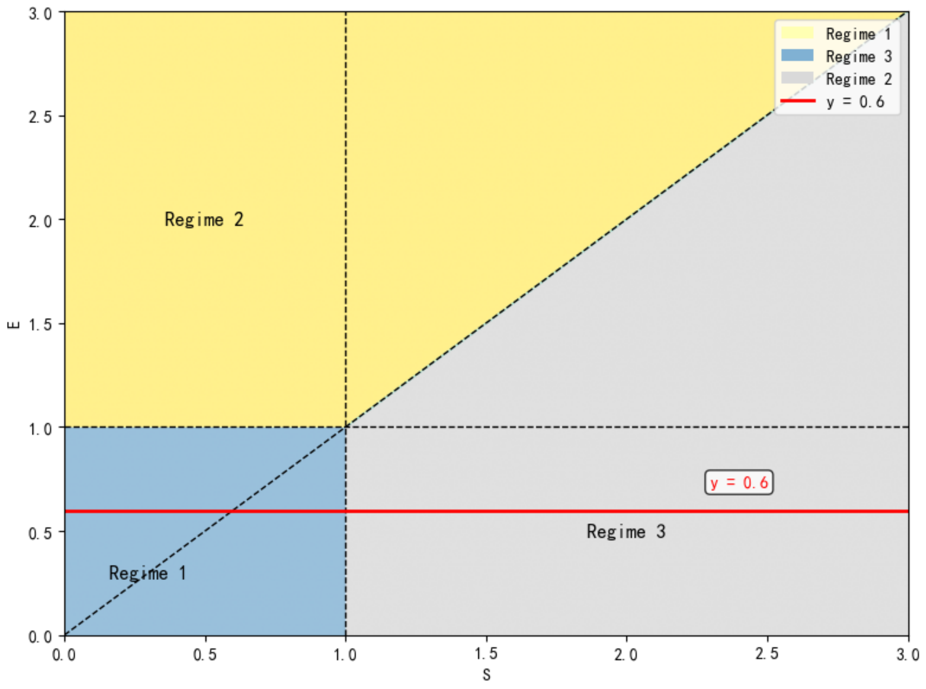


Figure 3.2

In regime 1, according to the equilibrium equation $K_d = \frac{E \times S}{C}$, we can get $C = \frac{E \times S}{K_d}$. Because in regime 1, we have $E_{tot} = E$, $S_{tot} = S$, so we get the equation of C:

$$C = \frac{E_{tot} \times S_{tot}}{K_d}$$

Because K_d and E_{tot} is constant now, because we just move along a line parallel to x-axis, the concentration of total enzyme is the same, and the K_d can’t change when we choose a pair of special enzyme and substance. So the equation of C is at the same form of $y = kx$, which is positive proportional function, and the slope of C is $\frac{E_{tot}}{K_d}$.

In regime 3, the restriction is $E_{tot} = C$, so the equation of C is:

$$C = E_{tot}$$

Because E_{tot} is a constant, C becomes constant now.

To link these two equation of C, we can use Hill function. The form of Hill function is as below.

$$\theta = \frac{E}{E + C}$$

The θ here is called Hill coefficient, which is proposed by Archibald Vivian Hill. The Hill coefficient is well-known by the explanation of the binding of oxygen and hemoglobin. For our enzyme catalysis process, the concentration of ES complex can be described as below.

$$C = (E + C) \times \frac{C}{E + C} = E_{tot} \times \frac{C}{E + C} = E_{tot} \times (1 - \theta)$$

Now we call $1 - \theta$ as φ, and it can be written as below.

$$\varphi = 1 - \theta = \frac{C}{E + C} = \frac{\frac{E \times S}{K_d}}{E + \frac{E \times S}{K_d}} = \frac{\frac{S}{K_d}}{1 + \frac{S}{K_d}}$$

So, the C can be written as below.

$$C = E_{tot} \times (1 - \theta) = E_{tot} \times \varphi = \frac{\frac{E_{tot} \times S}{K_d}}{1 + \frac{S}{K_d}}$$

Because in regime 1 and 3, we alaways have $S = S_{tot}$, so the equation can be written as below.

$$C = \frac{\frac{E_{tot} \times S}{K_d}}{1 + \frac{S}{K_d}} = \frac{\frac{E_{tot} \times S_{tot}}{K_d}}{1 + \frac{S_{tot}}{K_d}}$$

When in regime 1, we have $S_{tot} \ll K_d$, so we get $\frac{S_{tot}}{K_d} \ll 1$, we can ignore the $\frac{S_{tot}}{K_d}$ in the denominator, the equation changes to this form.

$$C = \frac{\frac{E_{tot} \times S_{tot}}{K_d}}{1} = \frac{E_{tot} \times S_{tot}}{K_d}$$

When in regime 3, we have $S_{tot} \gg K_d$, so we get $\frac{S_{tot}}{K_d} \gg 1$, we can ignore the "1" in the denominator, the equation changes to this form.

$$C = \frac{\frac{E_{tot} \times S_{tot}}{K_d}}{\frac{S_{tot}}{K_d}} = E_{tot}$$

So, this equation can link both regime 1 and regime 3, let us review the equation described C again.

$$C = E_{tot} \times \varphi = \frac{\frac{E_{tot} \times S_{tot}}{K_d}}{1 + \frac{S_{tot}}{K_d}}$$

So when crossing regime 1 and regime 3, in other words, when the total concentration of substance crossing K_d, the concentration of C is changing smoothly. In the language of mathematics, the left limit of $S_{tot} = K_d$ is equal to the right limit.

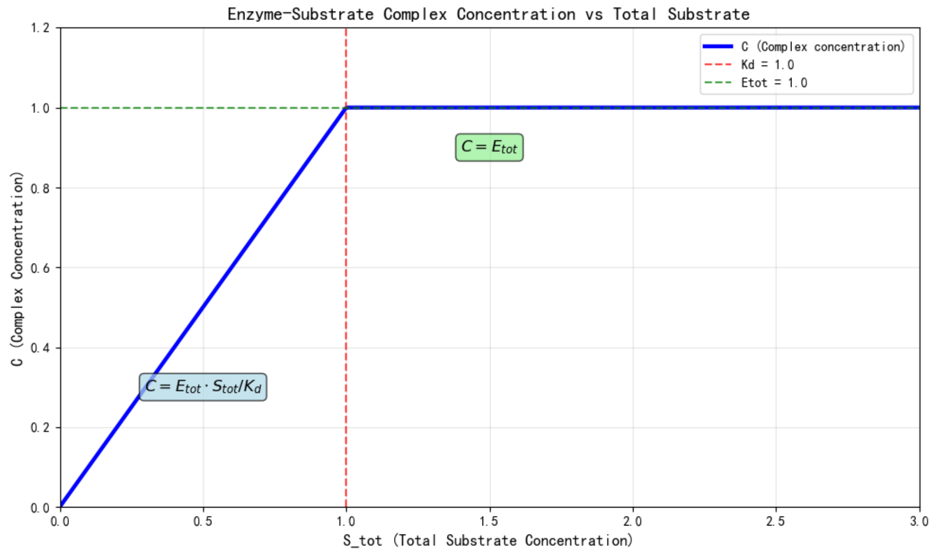


Figure 3.3

Then let us talk about another behavior, or another "LEGO". It is called "**bottleneck**". Bottleneck appears when the concentration of total substance moves along a line parallel with x-axis and crosses regime 2 and regime 3. That means the total concentration of enzyme is relatively high (higher than K_d). And the concentration of total substance increases from relatively low ($S_{tot} \ll K_d$) to relatively high ($S_{tot} \gg E_{tot} \gg K_d$). When in regime 2, the restriction is $E_{tot} = E$ and $S_{tot} = C$. So, the concentration of C is very obvious.

$$C = S_{tot}$$

And in regime 3, the restriction is $E_{tot} = C$, so the equation of C is:

$$C = E_{tot}$$

To link these two situation, the concentration of C can be written as below.

$$C = \min(S_{tot}, E_{tot})$$

This form looks like the concentration of C is controlled by the smaller one of the total concentration of enzyme or substance.

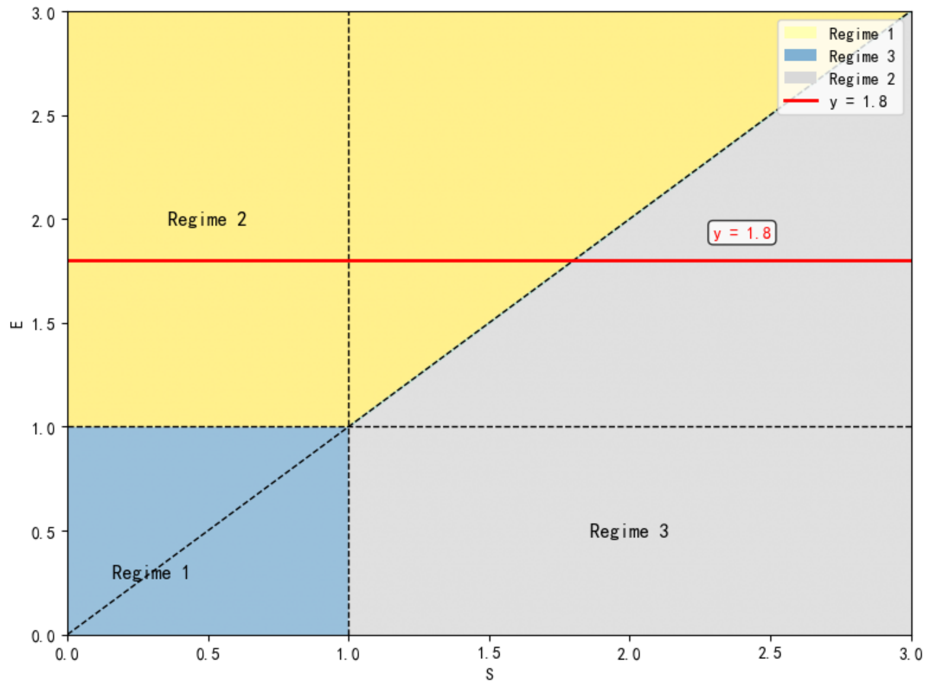


Figure 3.4

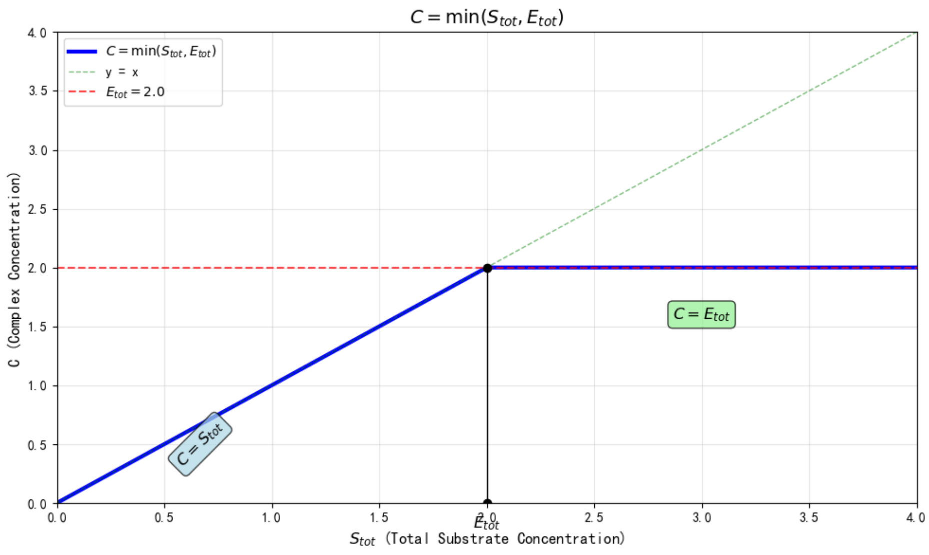


Figure 3.5

The third behavior or ”LEGO” appears in the $E_{tot} - S_{tot}$ graph, which means the x-axis is S_{tot} and the y-axis is E_{tot}. The third behavior is called ”ultrasensitivity”. we continues to imagine a line parallel to x-axis crossing regime 2 and 3, in regime 2, the restriction is $E_{tot} = E, S_{tot} = C$, so the concentration of enzyme is as below.

$$E = E_{tot}$$

In regime 3, the restriction becomes $S = S_{tot}$ and $E_{tot} = C$. So the concentration of enzyme is no longer equals E_{tot}. Considering the chemical equilibrium equation, the form can switch as below.

$$K_d = \frac{E \times S}{C} = \frac{E \times S_{tot}}{E_{tot}}$$

Here, we use $S = S_{tot}$ and $E_{tot} = C$ to change the E and C in the equation. So the concentration of enzyme is as below.

$$E = \frac{E_{tot} \times K_d}{S_{tot}}$$

So, if we consider the left limit and right limit when $S_{tot} = E_{tot}$, we will find they are different. Why we consider the border is $S_{tot} = E_{tot}$ here? That is because the border of regime 2 and 3 is no longer $S_{tot} = K_d$, but it is $S_{tot} = E_{tot}$.

For the left limit, it is as below.

$$\lim_{S_{tot} \to E_{tot}^{+}} E = E_{tot}$$

For the right limit, it is as below.

$$\lim_{S_{tot} \to E_{tot}^{-}} E = \frac{E_{tot}^2}{K_d}$$

So, there is a gap when S_{tot} is crossing the border of regime 2 and regime 3. And we can also notice when E_{tot} is bigger, the gap will become bigger. That is because the length of the gap equals to:

$$length = \frac{E_{tot}^2}{K_d} - E_{tot} = E_{tot}(\frac{E_{tot}}{K_d} - 1) = E_{tot} \times \frac{E_{tot} - K_d}{K_d}$$

When $E_{tot} - K_d$ becomes bigger, the length of the gap will become bigger, the ultrasensitivity will become much more obvious!

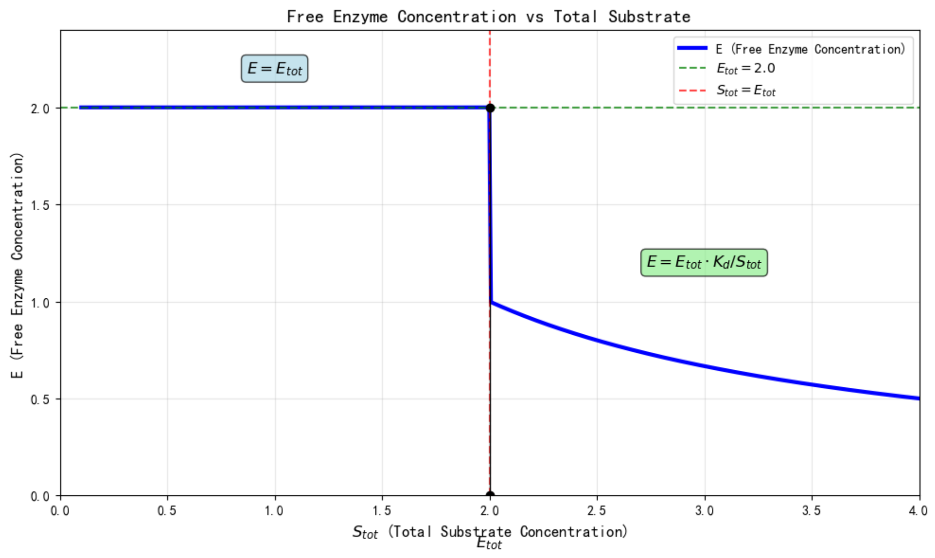


Figure 3.6

For ultrasensitivity, there is an example given in our class: imagine the binding of a kind of ligand and receptor is very tight (almost irreversible), that means when c(ligand) is bigger than c(receptor), there is almost no free receptor, almost all receptor is binding with ligand. And now the free ligands start to degrade, for example, they are ubiquitinated protein inhibitor. So when the total concentration of ligand crosses the border of $S_{tot} = E_{tot}$, here S_{tot} means the total concentration of ligand and E_{tot} means the total concentration of receptor, the concentration of receptor will undergo a very shape increase from almost zero to almost E_{tot} (the total concentration of receptor). This is just like the substance or inhibitor of enzyme with Hill coefficient larger than 1, for example, the figure shown below.

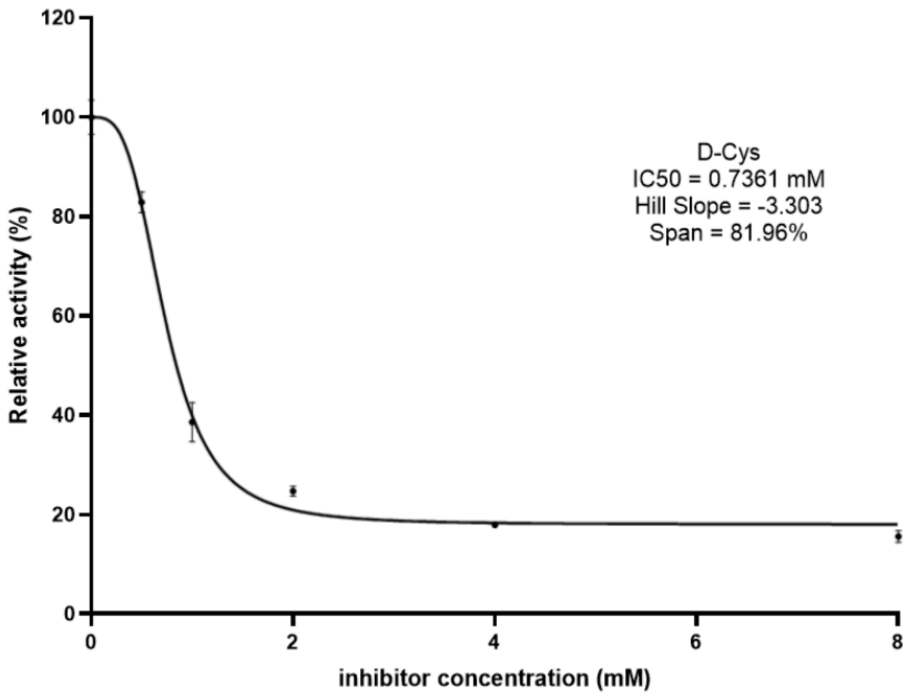


Figure 3.7

3.4 Adaptation Biomachines

In the section, Fangzhou told us a way to control the system, avoiding being disturbed. This well-known and widely-used control method is Proportion Integration Differentiation (PID). We can describe PID controller as the equation below.

$$u(t) = K_p \times y(t) + K_I \times \int_0^t y(t)\,dt + K_D \times \frac{dy(t)}{dt}$$

Here, u(t) is the output of a PID controller, and y(t) is the disturbance. The PID controller contains three parts: the present, the past, and the future.

$K_p \times y(t)$ represents the present, it is to give a force in the opposite direction.

And $K_I \times \int_0^t y(t)\,dt$ presents the past, because it is the integral over all time in the past. It is used to control the steady-state error. The steady-state error means the error is relatively small and will not grow when time goes. But if we allow this small error to accumulate, the system will become far from steady state over time.

And the $K_D \times \frac{dy(t)}{dt}$ represents the future, because it contains the derivative of y(t), it can reveal the change of y(t). This item can prevent the system from falling into shock, or control the over regulation caused by the first two items.

Chapter 4

Feedback, Feedforward, and Noise in Biosystems

Scribe: Shuo Wang, Kaijun Wang, Xiaowen Zhang
Lecturer: Fangzhou Xiao
Lecture #05 2025.10.09

Contents

4.1 Introduction: The Biomachine Perspective

4.1.1 Motivation and Principles

Biomachine is a perspective we can take on understanding biological systems. We treat them like machines.

Biological systems perform various functions just like engineered systems and machines. So principles on how to design machines with these functions can be used to understand why biological systems are designed the way they are.

However, biological systems *implement* the same principles using different physical components. So there are comparisons and contrasts. Even new theories and principles from biology can flow back to engineering.

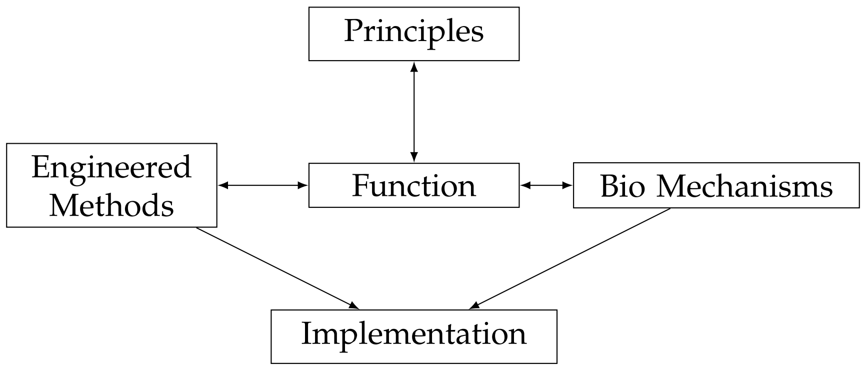


Figure 4.1 The relationship between principles, functions, and implementations in biological and engineered systems

4.1.2 Adaptation: The Function We Started With

Adaptation is a hallmark for biological systems and autonomous machines we build.

In control theory, we model a system as:

$$\dot{x} = f(x, w) \tag{4.1}$$
$$y = h(x, w) \tag{4.2}$$

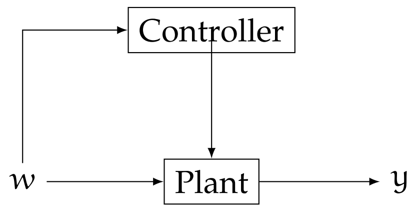
where w is a disturbance input and y is the output.

Perfect adaptation means: $y \to y_0$ as $t \to \infty$ for all constant w.

4.1.3 Controller Design Problem

There are two main control architectures:

1. **Feedforward Control:**



2. **Feedback Control:**

$$\begin{tikzcd}
w \arrow[r] & \boxed{\text{Plant}} \arrow[r] & y \arrow[d, no head] \\
& \boxed{\text{Controller}} \arrow[u] & {} \arrow[l]
\end{tikzcd}$$

From a simple example we saw:

1. **Feedforward** requires perfect matching of parameters. Can't do adaptation.
2. **Feedback** uses the power of time, observes the past, acts accordingly.

4.2 PID Control and Integral Feedback

4.2.1 PID Controller

e also saw, naturally, a **PID controller** emerge, which can achieve perfect adaptation:

$$u = k_P y + k_I \int y \, dt + k_D \dot{y} \tag{4.3}$$

Question: Which part of PID is responsible for adaptation?

Answer: The **Integral** part.

4.2.2 Internal Model Principle

The **Internal Model Principle** states: Adapt to all w from Σ_w $\Leftrightarrow$ Internal model of Σ_w.

$$\begin{tikzcd}
0 \arrow[r] & \boxed{\Sigma_w} \arrow[r, "w"] & \Sigma \arrow[r] & y
\end{tikzcd}$$

Perfect adaptation $\Leftrightarrow$ **Integral Feedback** (+ stable steady state).

4.2.3 Integral Feedback is Sufficient

Integral feedback is sufficient:

$$u = k_I \int y \, dt \quad \Rightarrow \quad \dot{u} = k_I y \tag{4.4}$$

At steady state: $y = 0$.

But requires **stability**!

4.2.4 Power of Integral Feedback

This is powerful: Adaptation *regardless of the plant*! (as long as the system can reach steady state)

Pretty much independent of everything!

(Look back at the simple example with only integral feedback.) Oscillation prevents perfect adaptation.

4.3 Implementation of Adaptation in Engineered Systems

4.3.1 Goal

Recall our goal: Use machine control principles to understand biology.

Control theory tells us that Perfect adaptation $\Leftrightarrow$ integral feedback.

But how to *implement* integral feedback? i.e., physically make it happen.

4.3.2 Electrical Circuits

Electrical circuits: variables are voltages V, currents I.

1. **Resistors:** $V = RI$, $I = \frac{V}{R}$
   This is **proportional control**.

2. **Inductor:**
$$V(t) = L\frac{dI(t)}{dt} \tag{4.5}$$
$$I(t) = I(0) + \int_0^t V(\tau)d\tau \tag{4.6}$$
   Integral or Derivative.

3. **Capacitor:**
$$V(t) = V(0) + \int_0^t I(\tau)d\tau \tag{4.7}$$
$$I(t) = C\frac{dV(t)}{dt} \tag{4.8}$$
   Integral or Derivative.

So, **PID control is very natural in electrical circuits!**

4.4 Implementation in Biological Reaction Networks (BRNs)

4.4.1 Constraints in Biology

But what about in biological reaction networks (BRNs)?

- Variables are **concentrations**, all **positive**. (Rule [P])
  (in contrast to V, I or position, velocity)
- Dynamics are, by default, **constrained polynomials**. (Rule [CP])

$$\frac{dx}{dt} = \Gamma^n x^\alpha \tag{4.9}$$

4.4.2 Control Types in BRNs

Proportional Control

Proportional: e.g., $u = k_P y = k_P(x_r - x)$

Often easy... $\dot{x} = \mu - px$, where $\mu = k_P x_r$, $p = k_P$.

Another example: $y = \frac{x}{x_r}$ (more natural!)

In fact, any static regulation $u(y)$ works as proportional feedback (upon linearization).

Derivative Control

Derivative: can often be done, but not often used.

Since biomolecular processes are **noisy**, and derivative control amplifies noise...

(not used much in highly noisy scenarios...)

Integral Control - The Challenge

Integral — important for perfect adaptation (if and only if).

But **not naturally achieved!** because of rule [CP].

$$z = \int y \, dt = \int (x_r - x) \, dt \tag{4.10}$$

$$\Rightarrow \quad \dot{z} = x_r - x \tag{4.11}$$

z **can't be a species** due to rule [CP]!

4.4.3 Strategies to Achieve Integral Control in Biology

To achieve integral control in biology needs further sophistication!

Strategy 1: Virtual Variable (Antithetic Integral Controller)

This is called **antithetic integral controller**. (Cell Systems 2016, Nature 2019, Mustafa Khammash group)

$$\dot{z}_1 = \mu - C \tag{4.12}$$

$$\dot{z}_2 = x - C \tag{4.13}$$

where $z_1, z_2 \xrightarrow{C} \varphi$ (annihilation reaction).

Let $z = z_1 - z_2$, then $\dot{z} = \mu - x$.

(Recall: this is like the dual rail strategy in last lecture to circumvent rule [P].)

But the problem is that rule [PU] makes it hard to directly use z in other variable's dynamics.

e.g., $\dot{x} = kz - \gamma x = kz_1 - kz_2 - \gamma x$

This requires kz_1 and kz_2 to have perfectly matching parameters, violating rule [PU]!

So, the best we can do is $\dot{x} = kz_1 - \gamma x$ for example.

This doesn't hurt Perfect Adaptation, since the integral variable still exists.

Strategy 2: Constrain to Certain Regimes

e.g.,

$$\dot{z} = \mu - C, \quad x + z \xrightarrow{k} x_f \tag{4.14}$$

In regime $C \ll x$, we have $\dot{z} \approx \mu - x$.

$$\begin{tikzcd} C \arrow[r] \arrow[d] & x_f \\ x_f + z & \end{tikzcd}$$

4.5 Incoherent Feedforward Loops (IFFL)

4.5.1 Nonlinear Structure Revealed by IFFL Motifs

Previously, we adopted a machine perspective on biological systems.

Since adaptation is a hallmark behavior in biology, we reasoned that an *adaptation machine* must achieve adaptation following similar design principles as engineered adaptation machines.

According to the internal model principle in control theory,

$$\text{perfect adaptation} \iff \text{integral feedback.} \tag{4.15}$$

Due to physical constraints on biomolecular reactions, it then seemed that biochemical reaction networks (BRNs) should achieve perfect adaptation by implementing integral feedback, perhaps via more sophisticated strategies. However, this is not an accurate picture of what is happening in reality.

4.5.2 Empirical Evidence for Feedforward Network Prevalence

In 2002, Shen-Orr *et al.*, *Nature Genetics*, examined all three-node network topologies formed by transcription factors and operons in *E. coli*. The study catalogued 115 transcription factors and 424 operons and analyzed the resulting interaction topologies.

They found that certain *network motifs*—subgraphs that occur far more frequently than in randomized networks—tend to be *feedforward*. This is surprising from the engineered-systems viewpoint: in engineered control systems, perfect adaptation cannot be achieved with purely feedforward architectures because it requires perfect matching of the parameters of two processes.

Thus the questions arise:

- How come biology is different?
- What happened to the design rule "perfect adaptation ⇔ integral feedback"?

Figure 4.2 shows a schematic three-node IFFL motif, representative of the structures overrepresented in transcriptional networks.

$$\begin{tikzcd}
 & X_1 \arrow[dd, -|] \\
W \arrow[ur] \arrow[dr] & \\
 & X_2
\end{tikzcd}$$

Figure 4.2 Incoherent feedforward loop (IFFL) motif: a three-node feedforward network where the upstream node W regulates both X_1 and X_2, and X_1 also regulates X_2.

4.5.3 The "Sniffer" as a Canonical IFFL Example

A concrete example of an incoherent feedforward loop is the so-called *"Sniffer"* circuit. The underlying reactions are:

$$W \xrightarrow{k_1} W + X_1, \tag{4.16}$$

$$W \xrightarrow{k_2} W + X_2, \tag{4.17}$$

$$X_1 + X_2 \xrightarrow{\delta} X_2, \tag{4.18}$$

$$X_2 \xrightarrow{\gamma} \varnothing. \tag{4.19}$$

The corresponding dynamics for the concentrations x_1 and x_2 of X_1 and X_2 are

$$\frac{dx_1}{dt} = k_1 W - \delta x_1 x_2, \tag{4.20}$$

$$\frac{dx_2}{dt} = k_2 W - \gamma x_2. \tag{4.21}$$

$$\begin{tikzcd}
W \arrow[rr] \arrow[dr] & & X_1 \\
 & X_2 \arrow[ur, -|] \arrow[r] & \varnothing
\end{tikzcd}$$

Figure 4.3 The "Sniffer" IFFL reaction network. The input W produces both X_1 and X_2; X_1 promotes the degradation of X_2, and X_2 is additionally degraded to a sink.

At steady state, from (4.21) we have

$$x_2^* = \frac{k_2}{\gamma} W. \tag{4.22}$$

Substituting into (4.20) and setting $\frac{dx_1}{dt} = 0$,

$$x_1^* = \frac{k_1 W}{\delta x_2^*} = \frac{k_1 W}{\delta \frac{k_2}{\gamma} W} = \frac{k_1 \gamma}{k_2 \delta}. \tag{4.23}$$

Thus,

$$x_1^* \text{ is invariant with respect to } W, \tag{4.24}$$

which is precisely *perfect adaptation*. Crucially, this invariance is independent of the exact parameter values in the sense of fine-tuned equality between gains; no perfect matching of parameters is required.

4.5.4 Reaction Orders as the Source of Nonlinearity

The key to understanding this behavior lies in the *exponents* appearing in the rate laws, that is, the *reaction orders*. For a generic reaction rate of the form

$$v = kx^{\alpha}, \tag{4.25}$$

the exponent α is the mass–action order in simple reactions, but can be more complicated when there are additional mechanisms or time-scale separation.

This structure—nonlinear dependence on concentration via reaction orders—is a feature that biological systems can exploit. It is typically not present, or is not fully utilized, in engineered systems, which are often modeled and analyzed after linearization.

4.5.5 Identifying the Implicit Integral Variable

Given the perfect adaptation property of the Sniffer circuit, it is natural to ask: *Where is the integral variable?*

Define the composite variable

$$z = k_2x_1 - k_1x_2. \tag{4.26}$$

Differentiating and substituting from (4.20)–(4.21) gives

$$\begin{aligned}
\dot{z} &= k_2\dot{x}_1 - k_1\dot{x}_2 && (4.27)\\
&= k_2(k_1W - \delta x_1x_2) - k_1(k_2W - \gamma x_2) && (4.28)\\
&= -k_2\delta x_1x_2 + \gamma k_1x_2 && (4.29)\\
&= x_2\big(\gamma k_1 - \delta k_2x_1\big). && (4.30)
\end{aligned}$$

The dynamics of z integrate the mismatch between the term δk_2x_1 and the constant γk_1, revealing how an effective integral feedback is hidden in the nonlinear reaction structure of the IFFL. This provides a mechanism for perfect adaptation in a purely feedforward network, reconciling biological observation with control-theoretic principles.

4.5.6 Comparison: Feedforward vs. Feedback

Although both Integral Feedback and IFFL can achieve adaptation, their mechanisms and philosophies differ significantly:

1. **"Cheating" vs. Robustness:** Integral Feedback is robust to *unknown* disturbances. It monitors the error and corrects it regardless of the source. In contrast, IFFL can be viewed as a form of "cheating." The system "knows" the specific noise/input (because it measures w directly) and calculates a precise cancellation via the parallel path. This requires the parameters (e.g., the ratio of production/degradation rates) to be finely tuned. If the nature of the disturbance changes, IFFL might fail, whereas Integral Feedback would still adapt.

2. **Specific vs. General Components:** In electrical engineering, Integral Feedback is often implemented using general-purpose components (like Op-Amps) designed to handle all types of noise. Biological systems, however, tend to evolve unique, specific feedback or feedforward loops for specific parts. There is no "universal integrator" protein; instead, each pathway has its own specific regulatory wiring tailored to its specific biological context.

3. **Mathematical Relationship:** In linear systems, at steady state, IFFL and IFB both require $\mathbf{d} - \mathbf{c}\mathbf{A}^{-1}\mathbf{b} = \mathbf{0}$, but with different structure. IFFL can have $\mathbf{d} > \mathbf{0}$ and $\mathbf{c}\mathbf{A}^{-1}\mathbf{b} > \mathbf{0}$, but IFB requires both $\mathbf{d} = \mathbf{0}$ and $\mathbf{c}\mathbf{A}^{-1}\mathbf{b} = \mathbf{0}$.

$$\begin{tikzcd}
 & X_1 \arrow[dd, -|, "\mathbf{c}"] \\
W \arrow[ru, "-\mathbf{A}^{-1}\mathbf{b}"] \arrow[rd, "\mathbf{d}"'] & \\
 & X_2
\end{tikzcd}$$

Figure 4.4 Three Nodes Again: a three-node feedforward network where the upstream node W regulates both X_1 and X_2, and X_1 also regulates X_2.

4.6 Noise in Biosystems

4.6.1 Introduction: The Biomachine Perspective

The biomachine perspective on biology we took was a systems engineering perspective.

It underlies the field of **systems and synthetic biology** that started in 2000.

With the completion of the human genome project, we had a glance at "all components in a cell", so we could start to reason about it as a system, and engineer it like a machine (like an electrical circuit).

$\Rightarrow$ The phrase: **biocircuit**.

4.6.2 The Repressilator

Repressilator. Elowitz 2000 Science. Was one of first work pioneering this perspective. (reading in HW 1).

Three genes' expression oscillates in E.coli:

$$\begin{tikzcd}
 & R_1 \arrow[rd, bend left] & \\
R_3 \arrow[ru, bend left] & & R_2 \arrow[ll, bend left]
\end{tikzcd}$$

Prediction: Smooth oscillations (single cell trajectory).

Reality: Not much of an oscillation.

Limitations of theory for design at the time... (ours based on regimes doesn't make Hill-type limiting assumptions).

But another big observation: **Noise!** between cells and within one cell.

GFP under the microscope.

4.6.3 Origin of Noise: Stochasticity

But where does noise come from, and when does it matter?

- E.coli cells from the same clone, so not genetic difference.

- $\Rightarrow$ Origin of noise is in **stochasticity of cellular processes**. Namely, chemical reactions.

Reaction events happen by molecules $E + S \xrightarrow{k}$ products encountering each other.

What's the distribution for the # of reaction events in a time interval Δt?

- Reactions happen when E and S hits
- Reaction rate is constant $a = kES$, unless E, S changes
- The events are independent of each other; they only depend on E and S's conc. and k

$\Rightarrow$ **Poisson distribution** with parameter $\lambda = a\Delta t$ (average # events in this interval).

Chemical reaction rate: $a = kES$.

4.6.4 Poisson Distribution

$X \sim \text{Poisson}(\lambda)$:

$$P\{X = k\} = \frac{\lambda^k}{k!} e^{-\lambda} \tag{4.31}$$

$$\mathbb{E}(X) = \text{Var}(X) = \lambda \tag{4.32}$$

$$\Rightarrow \text{CV} = \frac{\text{Var}(X)}{\mathbb{E}(X)} = \frac{\sqrt{\lambda}}{\lambda} = \frac{1}{\sqrt{\lambda}} \tag{4.33}$$

Coefficient of Variation (CV):

- $\lambda = 1$: $\text{CV} = 1$
- $\lambda = 100$: $\text{CV} = \frac{1}{10}$
- $\lambda = 10000$: $\text{CV} = \frac{1}{100}$

4.6.5 When Does Process Noise Matter?

Process noise becomes important when # events is small!

Translation to Molecule Numbers

# events $\rightarrow$ production events in balance with dilution by growth (i.e., no active degradation).

Then, if N_X = number of molecule X in cell:

- $\Rightarrow N_X$ produced every generation
- If one molecule X produced every event $\Rightarrow N_X$ events

Typical molecule numbers:

- $N_X \sim 10^2$ to 10^3 for proteins in E.coli: $\text{CV} \sim 0.1$
- $N_X \sim 10^5$ to 10^6 for metabolites in E.coli: $\text{CV} \sim 10^{-3}$

So, noise typically doesn't matter, even in E.coli.

- (Eukaryotes? Even less so...)

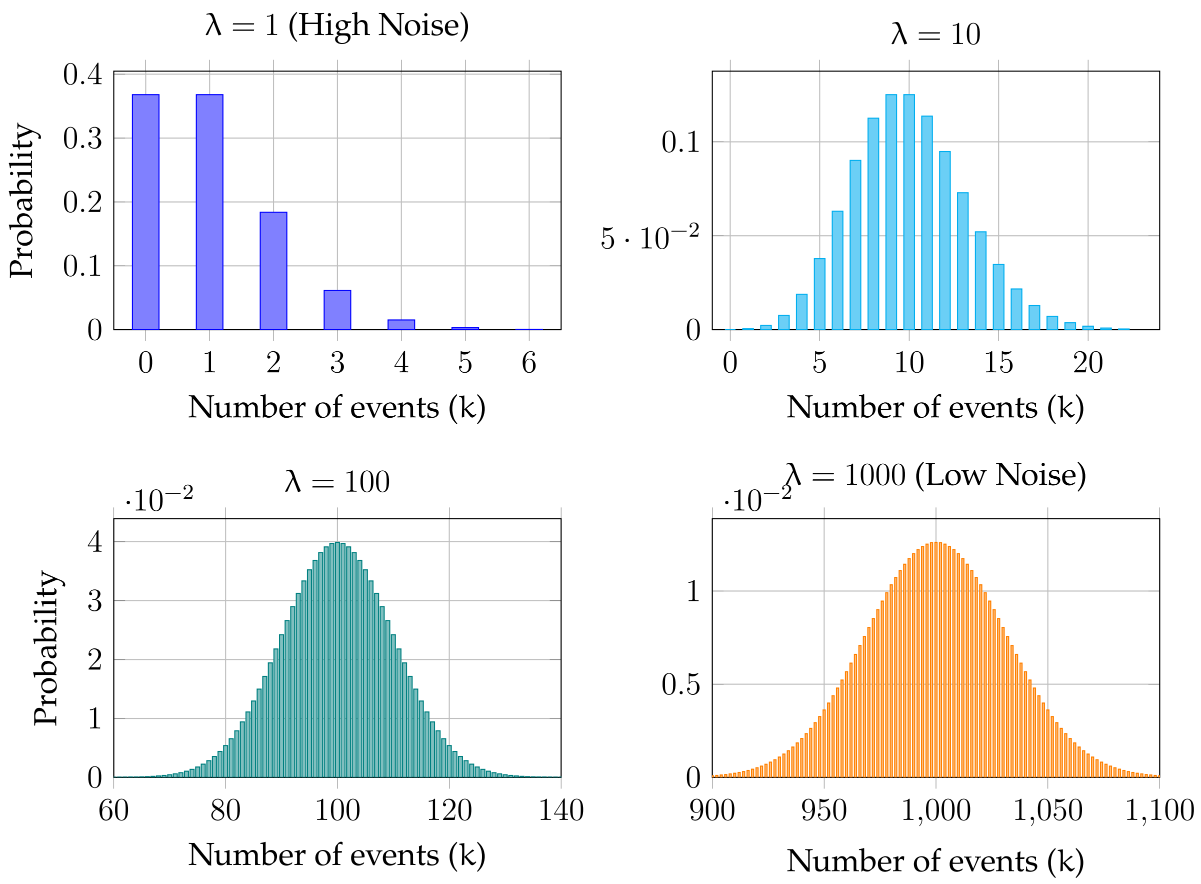


Figure 4.5 Poisson distributions for $\lambda = 1, 10, 100, 1000$. Note that as λ increases, the relative width (CV) decreases, representing lower noise relative to the mean.

- (Active degradation? More events, less noise.)

What's Missing? Burstiness in Gene Expression

We do see a lot of heterogeneity in single cells under the microscope!

What's missing? Burstiness in gene expression.

Several proteins produced per event, amplified by the **TX-TL process** (Transcription-Translation).

If 100 proteins per event:

$$N_{\text{event}} \sim \frac{N_X}{100} \sim 1 \text{ to } 10 \text{ for proteins in E.coli}$$

So, noise starts to matter.

"Critical Threshold" Concept

Typical burstiness ~ # proteins in cells.

- Low expression → noise dominates
- High expression → noise doesn't matter

Could be a strategy for bacteria:

- e.g., bet hedging... utilizing noise...
- Yet noise's role/importance is <u>tunable</u>.

Despite Burstiness: Noise Doesn't Play a Big Role in Most Biological Processes!

For noise to be important, need following factors (typically):

- Small cell size (e.g., not Eukaryotes)
- Low flux (not much degradation)
- Low concentration (e.g., genome)
- Large molecule size (e.g., not metabolites)
- Burstiness (e.g., not metabolic reactions)

CAUTION! Important Distinction

Comment: Noise $\neq$ Stochasticity (in processes) vs. Unknown (mechanisms)

Intrinsic randomness of underlying mechanisms vs. lack of information.

We are talking about noise from **process stochasticity** here, not "a wide spread in data" that also includes unknown mechanisms.

Example: A deterministic process can look "noisy" in data just because we don't know the mechanism.

- Like whether you bring umbrella to school: if you don't observe the weather.

Noise in data is a completely different topic because it's more about **inference of the unknown mechanisms**, not about stochasticity.

4.6.6 Chemical Master Equation (CME)

To formally describe this stochasticity, we use the CME.

Stochasticity in chemical reactions.

Now we have some confidence on the role noise plays in biological systems. We would like a way to formally describe and analyze them.

Setup: A Simple Reaction System (Elementary)

Consider: $\emptyset \xrightarrow{\mu} X \xrightarrow{k} \emptyset$

We could write this in **net-change form**:

$$X \xrightarrow{\nu_1} X+1 \quad \nu_1 = \mu \tag{4.34}$$

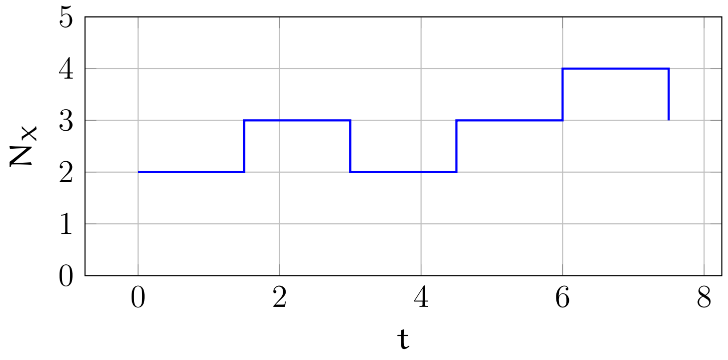
$$X \xrightarrow{\nu_2} X-1 \quad \nu_2 = kX \tag{4.35}$$

(Deterministic) rate equation:

$$\frac{dX}{dt} = \mu - kX$$

Stochastic case: N_X is discrete. $N_X = 0, 1, 2, \ldots$

It's a **random variable** that changes over time $\Rightarrow$ a **stochastic process**.



So its dynamics is described in terms of **probabilities**:

$$P_n(t) = \mathbb{P}\{N_X(t) = n\}$$

What's the Dynamics of $P_n(t)$? How Do Reactions Change It?

Consider the degradation reaction: $X \xrightarrow{\nu_2} X - 1$

This says: $\mathbb{P}\{\text{reaction happen in } [t, t+dt]\} = \nu_2(t)\,dt = kN_X(t)\,dt$

Deterministic: rate equation $\frac{dX}{dt} = \mu - kX$

Stochastic: N_X is discrete, $N_X = 0, 1, 2, \ldots$

State Transitions

$$\begin{tikzcd} n-1 \arrow[r, bend left, "\mu"] & n \arrow[l, bend left, "kn"] \arrow[r, bend left, "\mu"] & n+1 \arrow[l, bend left, "k(n+1)"] \end{tikzcd}$$

Derivation for Degradation Only

So, just consider this one reaction. It's:

$$\begin{aligned} P_n(t+dt) &= P_n(t) \cdot \mathbb{P}\{\text{no reaction in } [t, t+dt]\} \quad (4.36) \\ &\quad + P_{n+1}(t) \cdot \mathbb{P}\{\text{reaction in } [t, t+dt]\} \quad (4.37) \end{aligned}$$

(At $t \to 0$: either one reaction or no reaction.)

$$\begin{aligned} P_n(t+dt) &= P_n(t)(1 - kn\,dt) + P_{n+1}(t) \cdot k(n+1)\,dt \quad (4.38) \\ &= P_n(t) + k(n+1)P_{n+1}(t)\,dt - knP_n(t)\,dt \quad (4.39) \end{aligned}$$

$$\Rightarrow \frac{P_n(t+dt) - P_n(t)}{dt} = k(n+1)P_{n+1}(t) - knP_n(t)$$

$$\Rightarrow \text{as } dt \to 0, \quad \frac{dP_n}{dt} = k(n+1)P_{n+1} - knP_n \quad (n = 0, 1, 2, \ldots)$$

Now Add Production: $\emptyset \xrightarrow{\mu} X$

$$\frac{dP_n(t)}{dt} = k(n+1)P_{n+1}(t) - knP_n(t) + \mu P_{n-1}(t) - \mu P_n(t)$$

Chemical Master Equation:

$$\boxed{\frac{dP_n}{dt} = \mu P_{n-1} - \mu P_n + k(n+1)P_{n+1} - knP_n}$$

for $n = 0, 1, 2, 3, \ldots$ (Assume $P_{-1} = 0$ always.)

Interpretation

This is the **Chemical Master Equation**, about distributions.

Viewed as a dynamical system, this is **infinite dimensional**.

But it has **strong structure**. So we can simulate/analyze accordingly.

4.7 Credit

Contributor	Sections
Kaijun Wang	• 1 Introduction: The Biomachine Perspective • 2 PID Control and Intergral Feedback • 3 Implementation of Adaptation in Engineered Systems
Xiaowen Zhang	• 4 Implementation in Biological Reaction Networks • 5 Incoherent Feedforward Loops(IFFL)
Shuo Wang	• 5.6 Comparison: Feedforward vs. Feedback • 6 Noise in Biosystems

Chapter 5

Noise in Biology and Equilibrium Physics of Bioregulation

Scribe: Chenxiao Wang & Yihan Gong & Jiahe Wang
Lecturer: Fangzhou Xiao
Lecture #06 20251016

Contents

5.1 Course Overview and Review

CCBS Lecture 06 Noise in Bio & Equilibrium Physics of Bioregulation.

5.1.1 Review from Last Time

- Noise: Intro.
- Chemical master equation.

5.1.2 This Time's Content

- Gillespie algorithm.
- Noise analysis (simple, at steady state).
- Some stochastic phenomena.

References: Kardar's book; Erban, Chapman, Maini (2007), "Practical guide to stoch. sim of reaction diffusion processes"; Phillips, PBOC, etc.; Molecular Switch.

5.2 Equilibrium Physics of Bioregulation

5.2.1 Energy and Equilibrium: A Physics Perspective

Systems tend towards equilibrium.

- Equilibrium in physics – entropy, etc.
- Microscopic world – Boltzmann distr.
- Detailed balance.

This is the fundamental idea of equilibrium.

5.2.2 Applied to Enzymatic and Gene Regulation

Focus on single molecule's states.

- Michaelis-Menten.
- Allostery (MWC).
- Lac operon.

5.2.3 Beyond Equilibrium: Markov Chains

For molecular state transitions.

- e.g. Metabolism, phosphorylation cascades.

5.2.4 Steady State Distr. and Hitting Times

Steady state distr. and hitting times.

5.3 Gillespie Algorithm (or SSA – Stochastic Simulation Algorithm)

5.3.1 Simulation First, Because...

1. Deterministic already hard to analyze for general case. This is much harder for stochastic case.

2. Distribution vs Trajectory. Exact analysis, even when doable, is often only possible for steady state distribution, or distribution dynamics. But that's different from trajectory dynamics.
3. e.g.

   [Sketch: Smooth distribution over time vs fluctuating trajectory]
4. Usually analysis can be done after approximation, such as linearization. Analysis can give the full picture, but approximate. So, always helpful to check with simulations.

5.3.2 Distribution vs Trajectory

Can't directly simulate the distribution $\to$ inf. dim ODE. $P_n(t)$. We can simulate trajectories, then distr. can be obtained from averages over lots of traj.

5.3.3 Simplest Idea: Just Like ODE Sim.

$\frac{dx}{dt} = f(x)$ (Euler integration)$\to$ $X(t+dt) = X(t) + f(X(t))dt$.

For $X \xrightarrow{\nu=kX} X-1$,

$$N_X(t+dt) = \begin{cases} N_X(t) - 1 & \text{w/ prob } a dt = kN_X(t)dt \\ N_X(t) & \text{w/ prob } 1 - a dt \end{cases}$$

5.3.4 But Need dt Small to Have a Good Approx...

Could be very costly. Could we simulate $N_X(t)$ exactly? Yes, by transforming randomness from whether an event happens in an interval to when does an event happen.

5.3.5 Start at t, Want τ. s.t. $t+\tau$ is Next Reaction.

Let $f(N_X(t), s)ds \leftarrow$ an infinitesimal pdf for τ.

pdf for τ:

$$f(N_X(t), s)ds = P\{N_X(t) \text{ molecules at time } t, \text{ and the next reaction occurs in time interval } [t+s, t+s+ds)\} \tag{5.1}$$

$g(N_X(t), s) = P\{\text{No reaction in interval } [t, t+s)\}$

Denote reaction with rate $\nu(N_X)$. (e.g. $\nu = kN_X$)

$$\Rightarrow f(N_X(t), s)ds = g(N_X(t), s) \cdot \nu(N_X(t+s))ds = g(N_X(t), s) \cdot \nu(N_X(t))ds.$$

(Since no reaction $\to N_X(t+s) = N_X(t)$).

5.3.6 Let's Solve for $g(N_X(t), s)$.

(Memoryless or independent)

$$g(N_X(t), s+ds) = g(N_X(t), s)[1 - \nu(N_X(t))ds] \tag{5.2}$$

$$\frac{dg}{ds} = -\nu g \quad \Rightarrow \quad g = e^{-\nu s} \tag{5.3}$$

$\Rightarrow$

$$f(N_X(t), s)ds = \nu e^{-\nu s} ds \tag{5.4}$$

$\Leftarrow$ pdf (prob density function) for exponential distr. $\tau \sim \text{Exp}(\nu)$. pdf: $\nu e^{-\nu\tau}$. cdf: $1 - e^{-\nu\tau}$.

We want τ s.t. $t+\tau$ is time for next reaction. For a reaction with rate ν. Then $\tau \in [0,\infty)$ is a random number $\tau \sim \mathrm{Exp}(\nu)$.

This is exact! e.g. $X \xrightarrow{\nu=kX} X-1$ At t: draw $\tau \sim \mathrm{Exp}(kN_X(t))$. Then $N_X(t+\tau) = N_X(t)-1$.

5.3.7 What if Multiple Reactions?

The reactions are independent, each with rate $\nu_1, \ldots, \nu_m$.

Let τ_0 be time til' any reaction happens. $\Rightarrow \tau_0 \sim \mathrm{Exp}(\nu_1 + \cdots + \nu_m)$.

Which reaction? $P\{\text{it's reaction } j\} = \frac{\nu_j}{\nu_1+\cdots+\nu_m}$.

This completes the Gillespie algorithm, or SSA. Exact simulation of stochastic trajectories by sampling event times.

5.4 Analysis of Steady State Distributions

5.4.1 Simulations Can't Give the Full Picture, Over All Parameters

$\Rightarrow$ Analysis via moments, from CME. Mean, Var.

5.4.2 Exact Analysis. Example. of Moments.

$X \xrightarrow{\mu} X+1$ $X \xrightarrow{kx} X-1$

$$\frac{dP_n}{dt} = k(n+1)P_{n+1} + \mu P_{n-1} - (kn+\mu)P_n$$

Mean: $M(t) = \sum_{n=0}^{\infty} nP_n$

Variance: $V(t) = \sum_{n=0}^{\infty}(n-M)^2 P_n = \sum n^2 P_n - M^2$

$$\begin{aligned}
\frac{dM}{dt} &= \frac{d}{dt}\sum_{n=0}^{\infty} nP_n \\
&= k\sum_{n=0}^{\infty} n(n+1)P_{n+1} + \mu\sum_{n=0}^{\infty} nP_{n-1} - k\sum_{n=0}^{\infty} n^2P_n - \mu\sum_{n=0}^{\infty} nP_n \\
&= k\sum_{n=1}^{\infty}(n-1)nP_n + \mu\sum_{n=1}^{\infty} n(n-1)P_{n-1} - k\sum n^2P_n - \mu\sum nP_n \quad \text{(reindex)}
\end{aligned}$$

After calculation: $= \mu - kM$. (This is just like deterministic rate eqn. $\frac{dx}{dt} = \mu - kx$. Warning: Not always so.)

Similarly, observe $\sum_{n=0}^{\infty} n^2 P_n = V + M^2$

$$\begin{aligned}
\frac{d}{dt}(V+M^2) &= \frac{d}{dt}\sum n^2 P_n \\
&= k\sum n^2(n+1)P_{n+1} + \mu\sum n^2 P_{n-1} - k\sum n^3 P_n - \mu\sum n^2 P_n \\
&= k\sum (n-1)^2 n P_n + \mu\sum (n+1)^2 P_{n+1} - k\sum n^3 P_n - \mu\sum n^2 P_n \\
&= \sum [k((n^2-2n+1)n)P_n + \mu(n^2+2n+1)P_n - kn^3P_n - \mu n^2 P_n] \quad \text{(reindex for sums)} \\
&= \sum [k(n^3-2n^2+n)P_n + \mu(n^2+2n+1)P_n - kn^3P_n - \mu n^2 P_n] \\
&= \sum [kn^3 - 2kn^2 + kn + \mu n^2 + 2\mu n + \mu - kn^3 - \mu n^2]P_n \\
&= \sum [-2kn^2 + kn + 2\mu n + \mu]P_n \\
&= -2k\sum n^2 P_n + (k+2\mu)\sum nP_n + \mu \\
&= -2k(V+M^2) + (k+2\mu)M + \mu
\end{aligned}$$

Then

$$\begin{aligned}
\frac{d(V+M^2)}{dt} &= -2k(V+M^2) + (k+2\mu)M + \mu \\
\frac{dV}{dt} + 2M\frac{dM}{dt} &= -2kV - 2kM^2 + kM + 2\mu M + \mu
\end{aligned}$$

But $\frac{dM}{dt} = \mu - kM$, so $2M\frac{dM}{dt} = 2M(\mu - kM) = 2\mu M - 2kM^2$

$$\begin{aligned}
\Rightarrow \frac{dV}{dt} &= -2kV - 2kM^2 + kM + 2\mu M + \mu - 2\mu M + 2kM^2 \\
&= -2kV + kM + \mu \\
&= \mu + kM - 2kV
\end{aligned}$$

$\frac{dM}{dt} = \mu - kM$

At s.s. $M = \frac{\mu}{k}$, from $\mu + kM - 2kV = 0 \Rightarrow V = \frac{\mu + kM}{2k}$, and with $M = \frac{\mu}{k}$, $V = \frac{\mu+\mu}{2k} = \frac{2\mu}{2k} = \frac{\mu}{k} = M$.

So. Mean = Variance. (Poisson! In fact, it is...) You can solve P_n at s.s. explicitly...

5.4.3 But This Doesn't Always Work.

That the moments form a finite number of equations is called Moment Closure. Not closed if, e.g. $E(X)$ depends on $E(X^2)$ depends on $E(X^3)$... this happens when $X \xrightarrow{\nu = x^2} X - 1$

5.4.4 For Example.

Write reactions in net change form $x \xrightarrow{\nu(x)} x + \nu$. x is a vector of species, molecular counts. $x = (x^j)_{j=1}^m$. $x = (x^j)$. $j = 1, \dots, m$.

Example: $X \xrightarrow{f^+(x)} X + 1$ $X \xrightarrow{f^-(x)} X - 1$

$$\Rightarrow \frac{dp(x,t)}{dt} = f^+(x-1)p(x-1,t) - f^+(x)p(x,t) + f^-(x+1)p(x+1,t) - f^-(x)p(x,t)$$

$\langle X \rangle = E(X)$.

Just another notation

$$\frac{d\langle X\rangle}{dt} = \langle f^+(x)\rangle - \langle f^-(x)\rangle$$

e.g. $f^+(x) = \mu$, $f^-(x) = x^2$. then $\frac{d\langle x\rangle}{dt} = \mu - \langle x^2\rangle$.

5.4.5 Cases with Moment Closure

– Linear, i.e. 1st order or 0th order reactions. ν_j are all degree 1 polynomials of x. e.g. C, $c + x_1$, x_2, but not $x_1 x_2$.

– Feedforward structure $\dot{x}_1 = \mu - x_1$, $\dot{x}_2 = x_1^2 - x_2$, $\dot{x}_3 = x_1 x_2 - x_3$.

This excludes many interesting cases though...

5.4.6 More Generally, How to Analyze Steady State Moments?

$\Rightarrow$ Linear noise approximation (LNA).

– Just like using linearization to analyze nonlinear dynamical systems. We can also do linearization for stochastic processes.

– approximate, $\nu_i(x) \approx \nu_i(x^*) + \sum_j \frac{\partial \nu_i}{\partial x^j}(x - x^*)$.

– For the example, but with $x^* = \langle x\rangle$.

$$\begin{aligned}\frac{d\langle x\rangle}{dt} &= \langle f^+(x)\rangle - \langle f^-(x)\rangle \\ &\approx f^+(\langle x\rangle) + \frac{\partial f^+}{\partial x}\langle x - \langle x\rangle\rangle - f^-(\langle x\rangle) - \frac{\partial f^-}{\partial x}\langle x - \langle x\rangle\rangle \\ &= f^+(\langle x\rangle) - f^-(\langle x\rangle)\end{aligned}$$

– General solution and application of LNA, see homework.

5.5 Stochasticity in Biological Systems

How biological noise arises from molecular mechanisms and impacts cellular processes like gene expression and cell fate decisions.

5.5.1 Gene Expression Noise & Burstiness

Gene expression is often "bursty" or "noisy," meaning protein/mRNA levels fluctuate significantly over time in identical cells. This noise is largely attributed to transcriptional bursting—genes switching between active (ON) and inactive (OFF) states, producing a "burst" of mRNA molecules when ON. Understanding this noise is crucial for explaining phenotypic variability in genetically identical cell populations.

5.5.2 Modeling Transcriptional Bursting

Attempt 1: Simple Burst Model

Reaction:

$$G \xrightarrow{k} G + bP \tag{5.5}$$

(where b is the average burst size, i.e., number of proteins produced per activation event).

Limitations:

Assumes a constant, fixed burst size $\mathtt{b}$.

Attempt 2: Random Burst Size:

Improvement:

Model burst size $\mathtt{b}$ itself as a random variable (e.g., drawn from a geometric distribution).

Limitation:

Still a phenomenological model; doesn't explain the *origin* of the variability in $\mathtt{b}$.

Attempt 3: Mechanistic Two-State (ON/OFF) Model

Mechanism:

Explicitly models the gene's promoter switching.

$$G_{off} \rightleftharpoons G_{on} \tag{5.6}$$

(stochastic switching between states)

$$G_{on} \xrightarrow{K_{+x}} G_{on} + P \tag{5.7}$$

(protein production only in the ON state)

The observed "burst size" emerges naturally from the time the gene spends in the ON state and the rate of transcription. Different genes have different switching kinetics, explaining gene-specific noise profiles.

This model illustrates how the structure of a mechanism (ON/OFF switching) directly organizes the features of observed variation (burstiness).

5.5.3 From Multistability to Multimodality: The Role of Noise in Dynamical Systems

In dynamical systems theory, particularly in biological and chemical contexts, the relationship between deterministic structure and stochastic behavior is fundamental. This document formalizes the connection: **multistability combined with noise yields multimodal probability distributions**.

Deterministic Model

$$\frac{dx}{dt} = f(x), \quad \text{where } f(x) \text{ has a cubic (N-shaped) form} \tag{5.8}$$

This model has the following behavior:

- Two stable fixed points: "ON" (x_{on}) and "OFF" (x_{off})
- One unstable saddle point (x_{saddle}) in between
- Basins of attraction partition the state space

Interpretation: Initial condition determines final state $\Rightarrow$ perfect bistability. Cells are permanently in one fate or the other.

Adding Stochasticity (Noise)

The equation becomes:

$$dX = f(X)\,dt + \sigma\,dW \quad \text{(Stochastic Differential Equation)} \tag{5.9}$$

where W is a Wiener process (Brownian motion) and σ quantifies noise intensity. Noise enables transitions between basins $\Rightarrow$ **stochastic switching** between ON and OFF states.

Steady-state distribution (SSD): At $t \to \infty$, probability distribution $P_{ss}(x)$ becomes **bimodal**, with peaks at the two stable states.

Population view: A heterogeneous population emerges, with fractions in each state corresponding to the peaks of $P_{ss}(x)$.

5.5.4 Extinction and Zero

Autocatalysis in Biological Growth

Many biological processes involve autocatalysis:

$$X \xrightarrow{K} X + X \quad \text{or} \quad X \xrightarrow{f(X)} X + 1 \tag{5.10}$$

Key feature: You need X to make more X.

The Extinction Problem:

- If $X = 0$, the system cannot produce more X
- This creates an **absorbing state** at $X = 0$
- Once the system reaches $X = 0$, it remains there forever

Deterministic Model

For a simple autocatalytic system:

$$\frac{dx}{dt} = kx - rx = (k - r)x \tag{5.11}$$

Fixed points:

- $x^* = 0$ (unstable if $k > r$)
- Growth occurs for any $x(0) > 0$

With External Disturbances

Adding external mortality/removal:

$$\frac{dx}{dt} = \frac{kx}{K + x} - (r + \mu)x \tag{5.12}$$

- Can create a **stable** extinction state if μ is large enough
- But still deterministic: either always extinct or never extinct

5.5.5 Ergodicity

Definition: A system is ergodic if, starting from any point, over time it can visit **every point** in the state space.

For a Markov process, ergodicity requires:

1. **Irreducible:** For any states i and j, there exists $t > 0$ such that:

$$P(X(t) = j \mid X(0) = i) > 0 \tag{5.13}$$

2. **Positive recurrent:** The expected return time to any state is finite

A stochastic process $\{X(t)\}_{t\geqslant 0}$ is ergodic if:

$$\lim_{T\to\infty} \frac{1}{T} \int_0^T \mathbf{1}_{\{X(s)\in A\}}\, ds = \pi(A) \tag{5.14}$$

where π is the stationary distribution.

Obtaining Distributions from Trajectories

Ensemble Approach (Many Trajectories)

Take N independent trajectories $\{X_i(t)\}_{i=1}^N$:

$$\hat{p}_t(x) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{\{X_i(t)=x\}} \tag{5.15}$$

As $N \to \infty$:

$$\hat{p}_t(x) \to P(X(t) = x) \tag{5.16}$$

Steady-State Distribution

If the system has a stationary distribution π:

$$\pi(x) = \lim_{t\to\infty} P(X(t) = x) = \lim_{t\to\infty} \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{\{X_i(t)=x\}} \tag{5.17}$$

Time-Average Approach (Single Trajectory)

For an ergodic system, follow one trajectory $X(t)$:

$$\pi(x) = \lim_{T\to\infty} \frac{1}{T} \int_0^T \mathbf{1}_{\{X(t)=x\}}\, dt \tag{5.18}$$

5.6 Equilibrium physics of bioregulation

The world is connected, and our "eye" to see such connections are orders of magnitude reasoning. Seemingly unrelated observations could be in fact deeply constraining each other. To practice this "vision", we explore some calculations below, with contexts gradually shifting from the macroscopic world we are more familiar with to the microscopie world of molecules and cells.

5.6.1 Energy and Equilibrium in statistical physics

Energy is a concept from physics that a closed system (i.e without energy input) would dissipate energy and

- **Energy** is a concept from physics that a closed system (i.e without energy input) would dissipate energy and
- **Side note:** But Bio is not closed, so not in equilibrium!
- Answer:
- But energy In other words, equilibrium is **"easier to implement"**.
- Also, driven processes can still have several behaviors that "look like" equilibrium, i.e., they balance just like an equilibrium system due to other constraints. e.g. network topology (no cycles).
- How to use equilibrium?
  - Statistical Mechanics. A system consists of lots of particles, so we only need to care about statistics of particles.

$$\text{Microstate} \quad \text{(all particle states)} \qquad \xrightarrow{\text{Multiplicity}} \qquad \text{Macrostate} \quad \text{(statistical states)}$$

$$x \qquad\qquad E(x) \text{ (energy)}$$

$$\text{weight } w(x) \qquad \text{energy } E(x) \qquad \text{distribution } P(x)$$

$$\langle X\rangle \text{ (observation)}$$

- Equilibrium: a distribution over microstates with expected property. equilibrium distribution. Namely, the following are equivalent characterizations:

1. **Boltzmann distribution**

   Every (micro) state has an energy $E(x)$, and the equilibrium distribution is

   $$p(x) \propto e^{-\beta E(x)}, \qquad \beta = \frac{1}{kT}.$$

   where $\beta = \frac{1}{k_B T}$, that we often omit it.

2. **Detailed balance**

   For every transition between two microstates,

   $$p_A k_{A\to B} \rightleftharpoons p_B k_{B\to A}.$$

   Detailed balance says **forward flux = reverse flux**, for every state transition.

   $$k_1 P_A = J_{A\to B} = J_{B\to A} = k_2 P_B$$

$$\begin{tikzcd}
X_1 \arrow[rr, shift left] \arrow[dr, shift left] & & X_2 \arrow[ll, shift left] \arrow[dl, shift left] \\
& X_3 \arrow[ul, shift left] \arrow[ur, shift left] &
\end{tikzcd}$$

$$\Rightarrow \frac{p_A}{p_B} = \frac{k_2}{k_1}$$

So we can define E_A, E_B, s.t. $p_A \propto e^{-E_A}$, $p_B \propto e^{-E_B}$, then $\frac{p_A}{p_B} = e_{-(E_A-E_B)} = \frac{k_2}{k_1}$

- Transition rates and energies are related.

3. **No cyclic flux**

   Detailed balance:

$$k_{12}p_1 = k_{21}p_2$$

$$\Rightarrow J_{\circlearrowleft} = (k_{12}p_1 + k_{23}p_2 + k_{31}p_3) - (k_{21}p_2 + k_{13}p_1 + k_{32}p_3) = 0$$

   Equilibrium constrains transition rates:

$$p_1 = \frac{k_{21}}{k_{12}}p_2 = \frac{k_{21}}{k_{12}}\frac{k_{32}}{k_{23}}p_3 = \frac{k_{21}}{k_{12}}\frac{k_{32}}{k_{23}}\frac{k_{13}}{k_{31}}p_1$$

   In a system with states $X_1 \leftrightarrow X_2 \leftrightarrow X_3$, the absence of cyclic flux implies

$$\frac{k_{1\to2}\, k_{2\to3}\, k_{3\to1}}{k_{2\to1}\, k_{3\to2}\, k_{1\to3}} = 1.$$

   i.e

$$k_{1\to2}\, k_{2\to3}\, k_{3\to1} = k_{2\to1}\, k_{3\to2}\, k_{1\to3} \tag{5.19}$$

   which is called **"Cycle condition"**.

5.6.2 Equilibrium in bioregulation

- Equilibrium is very powerful
  - Distribution directly obtained. Only need to know the **states**, no need to know reaction mechanisms!

Example 1 (Enzymatic reaction). $S \overset{v}{\rightsquigarrow} P$, catalyzed by enzyme E. Consider the reversible reaction

$$E + S \rightleftharpoons ES.$$

We focus on a *single enzyme molecule*. Here S denotes the substrate. Assume the enzyme has two possible macrostates:

- free state: E
- bound state: ES

Energy.

$$E(E) = 1, \qquad E(ES) = \Delta G_b.$$

Multiplicity (number of microstates). Multiplicity is the number of microstates corresponding to each macrostate:

$$\Omega_E = 1, \qquad \Omega_{ES} = \frac{S_{tot}}{C_0},$$

where S_{tot}/C_0 is the dimensionless concentration factor.

Statistical weight.

$$w_E = 1, \qquad w_{ES} = \frac{S_{tot}}{C_0}\, e^{-\Delta G_b}.$$

Probability of being in a bound state. From equilibrium statistical mechanics:

$$p_{bound} = \frac{w_{ES}}{w_E + w_{ES}} = \frac{(S_{tot}/C_0)\, e^{-\Delta G_b}}{1 + (S_{tot}/C_0)\, e^{-\Delta G_b}}.$$

This matches the Michaelis–Menten occupancy formula. If $S_{tot} = E_{tot}$, one may define

$$C = \frac{S_{tot}}{E_{tot}}, \qquad K = Ce^{-\Delta G_b}.$$

Note that. This simplified derivation considers only *one enzyme molecule*. Therefore, the system only has 2 macro states while a full system with many enzymes has many possible microstates.

Then, this can be applied to **several** enzyme molecules by **assuming each enzyme is *i.i.d*(independent and identically distribution)**, so $E = p_b E_{tot} = E_{tot} \frac{S_{tot} e^{-\Delta G_b}}{S_{tot} e^{-\Delta G_b} + 1}$

For example, if

$$N_E, \quad N_S, \quad N_{ES}$$

denote molecular counts, then they satisfy the constraint

$$N_E + N_S + N_{ES} = N_{E,tot},$$

and the number of microstates can become very large.

$\triangle$

Example 2 (MWC Model: The Second Secret of Life — **Allostery**). Enzyme activities can be regulated by ligands or substrates. The MWC (Monod–Wyman–Changeux) model describes allostery: enzymes have **multiple conformations** and **binding states**.

- **Allostery:** an enzyme may have multiple conformations.
- **Multiple conformations + binding:** an enzyme can bind ligands in different conformational states.
- **Independent contributions:** conformational state and ligand-binding state contribute multiplicatively to statistical weight.

We consider two conformations:

$$\text{active (A)} \qquad \text{and} \qquad \text{inactive (I)}.$$

Each conformation can be either ligand-free or ligand-bound.

State Energies and Weights The four macrostates are:

$$A,\ A + \text{bound},\ I,\ I + \text{bound}.$$

Their energies and statistical weights are:

State	Energy	Weight	Bio Notation
Active	E_A	e^{-E_A}	
Active + bound	$E_A + E_{A,b}$	$e^{-(E_A+E_{A,b})}\,\frac{S_{tot}}{C_0}$	$e^{-\epsilon_A}\frac{S_tot}{K_A}$
Inactive	E_I	e^{-E_I}	
Inactive + bound	$E_I + E_{I,b}$	$e^{-(E_I+E_{I,b})}\,\frac{S_{tot}}{C_0}$	$e^{-\epsilon_I}\frac{S_tot}{K_I}$

Here S_{tot}/C_0 is the multiplicity factor (number of accessible microstates of ligand binding).

Microstate Counting When the "Single-enzyme" Assumption is Removed If the assumption of a single enzyme molecule is removed, the number of microstates can become very large or even infinite.

For example, the system may include:

$$N_E, \quad N_S, \quad N_{ES},$$

with the constraint

$$N_E + N_S + N_{ES} = N_{E,tot}.$$

The combinatorial number of microstates grows rapidly with molecule counts. This is why equilibrium statistical mechanics is useful — it allows us to compute distributions without enumerating all microstates.

Interpretation The MWC model explains how enzymes can switch activity states depending on ligand concentration, through changes in relative statistical weights of conformational states.

△

Activation Probability in the MWC Model From the statistical weights of the four states (A, A+bound, I, I+bound), the probability of being in the active conformation is

$$P_{\text{active}} = \frac{e^{-E_A}\left(1+\frac{S_{\text{tot}}}{K_A}\right)}{e^{-E_A}\left(1+\frac{S_{\text{tot}}}{K_A}\right)+e^{-E_I}\left(1+\frac{S_{\text{tot}}}{K_I}\right)}.$$

Limit as $S_{\text{tot}} \to 0$.

$$P_{\text{active}}(0) = \frac{e^{-E_A}}{e^{-E_A}+e^{-E_I}} = \frac{1}{1+e^{E_A-E_I}}.$$

If the inactive state is lower in energy (e.g. $E_I < E_A$ by 2–3 $k_B T$, about a hydrogen bond), then

$$P_{\text{active}}(0) \approx \tfrac{1}{10}.$$

Limit as $S_{\text{tot}} \to \infty$.

$$P_{\text{active}}(\infty) = \frac{e^{-E_A}\frac{S_{\text{tot}}}{K_A}}{e^{-E_A}\frac{S_{\text{tot}}}{K_A}+e^{-E_I}\frac{S_{\text{tot}}}{K_I}} = \frac{e^{-E_A}/K_A}{e^{-E_A}/K_A+e^{-E_I}/K_I} = \frac{1}{1+e^{E_A-E_I}\frac{K_A}{K_I}}.$$

If $K_A < K_I$, the ligand binds tighter in the active state. For example, if $K_I/K_A \approx 100$, then

$$P_{\text{active}}(\infty) \approx \frac{1}{1+10^{-2}} \approx 0.9.$$

Cooperativity: Example of a Dimer

Consider a dimeric enzyme where each binding site becomes active/inactive independently and can bind/free ligand independently.

In this case, the active-state weight is squared:

$$P_{\text{active}} = \frac{e^{-E_A}\left(1+\frac{S_{\text{tot}}}{K_A}\right)^2}{e^{-E_A}\left(1+\frac{S_{\text{tot}}}{K_A}\right)^2+e^{-E_I}\left(1+\frac{S_{\text{tot}}}{K_I}\right)^2}.$$

High-ligand limit.

$$P_{\text{active}}(\infty) = \frac{e^{-E_A}\left(\frac{S_{\text{tot}}}{K_A}\right)^2}{e^{-E_A}\left(\frac{S_{\text{tot}}}{K_A}\right)^2+e^{-E_I}\left(\frac{S_{\text{tot}}}{K_I}\right)^2} = \frac{1}{1+e^{E_A-E_I}\left(\frac{K_A}{K_I}\right)^2}.$$

This yields a *sharper transition* in P_{active} as a function of ligand concentration, characteristic of cooperativity.

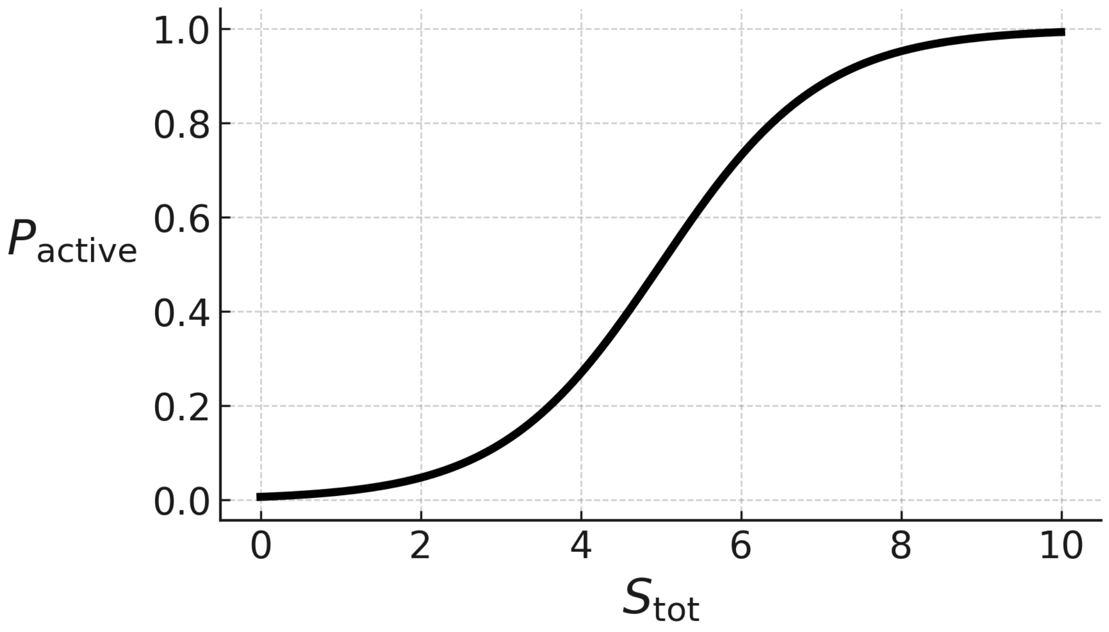


Figure 5.1 $P_{active} vs S_{tot}$

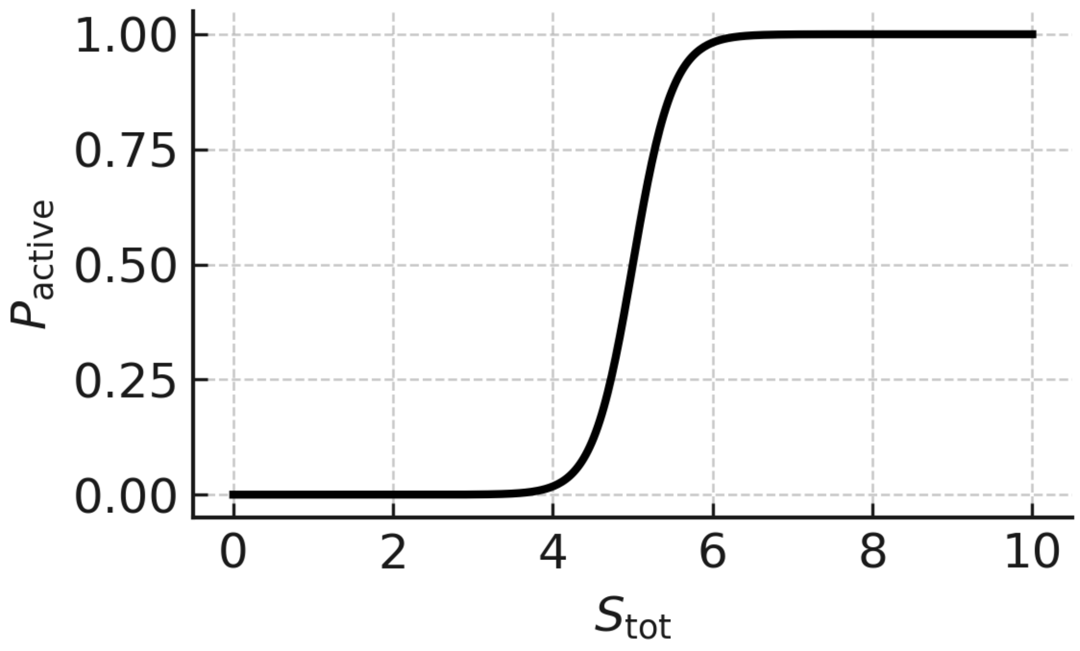


Figure 5.2 Caption

Example 3 (Gene Expression)**.** Consider the states pf the **gene** under repression: We enumerate the possible promoter states, their energies, and statistical weights.

State	**Energy**	**Weight**
Promoter free	0	1
RNA polymerase (RNAP) bound	$\Delta\varepsilon_p$	$e^{-\Delta\varepsilon_p} \dfrac{P}{N_{NS}}$
Repressor bound	$\Delta\varepsilon_r$	$e^{-\Delta\varepsilon_r} \dfrac{R}{N_{NS}}$

Where:

- P: number of RNA polymerase molecules - R: number of repressor molecules - N_{NS}: number of non-specific DNA binding sites - $\Delta\varepsilon_p$: RNAP binding energy - $\Delta\varepsilon_r$: repressor binding energy

These statistical weights can be used to compute promoter occupancy and the probability of transcription initiation.

△

5.6.3 Beyond Equilibrium – Markov chains

- Equilibrium has the powerful property that we don't need to know the detailed kinetic mechanisms, just the **thermodynamics** (i.e. interacting energies...).
- But what if I encounter a behavior?
  - my behavior of interest can **only** be achieved out of equilibrium...
  - e.g. Kinetics proofreading... (super precise)

  How to analyze that?
- For the special case of **finite number of states**, if we know the **state transition rates**, this is a *Markov Chain*. (A special case of chemical master equation where we can get the **full distribution**.)

Chapter 6

Equilibrium Physcis of Bioregulation

Scribe: Yiqiao Deng, Shaocong Fang, Chengqian Li
Lecturer: Fangzhou Xiao
Lecture #7 20251023

Contents

6.1 What is Ergodicity

Ergodicity is crucial for equilibrium. Ergodicity means that, given sufficient time, a system can evolve into any possible state. In a closed system with ergodicity, after evolving for a sufficiently long time, it will inevitably reach an equilibrium state, which unifies temporal equilibrium and spatial equilibrium. Here, temporal equilibrium refers to the average value of the trajectory of a single point in the system after evolving for a sufficiently long time, while spatial equilibrium refers to the average value formed by the superposition of all states of all points in the system at a specific moment. In an ergodic system, the equilibrium state ensures that the temporal equilibrium equals the spatial equilibrium. Such ergodic systems provide us with a simplified method for solving their equilibrium states. We only need to simulate a single point, starting from a certain initial state, and let this point evolve for a sufficiently long time. The temporal average obtained from its trajectory will then equal the equilibrium state of the system. For example, consider the velocity distribution of ideal gas molecules at a certain temperature. We only need to simulate one molecule, let it evolve under the system's conditions for a sufficiently long time, and the distribution of the velocities it has assumed will correspond to the velocity distribution of all the ideal gas molecules in the system—that is, the equilibrium state of the system.

Furthermore, if a system is not ergodic, then its equilibrium state is neither unique nor determinate, which means we cannot discuss its equilibrium in a definitive manner. For example, a system loses ergodicity if one of its state variables can go extinct. By "extinct," we mean that once this state variable reaches zero, it becomes trapped at zero forever. Consider a state variable x, where dx/dt depends only on x, and when x equals zero, $dx/dt = 0$. In this case, once the system reaches $x = 0$, x will be permanently trapped at that point. Therefore, even if the system possesses several stable points (or equilibrium states) where $x \neq 0$, once x reaches zero, it can never access those non-zero stable points, even over an infinite amount of time. This system thus loses ergodicity. Consequently, the equilibrium of such a system is indeterminate. Depending on the initial conditions (i.e., the value of x at $t = 0$), the system might sometimes reach a non-zero stable point, and other times become trapped at $x = 0$. In this scenario, the stable points and equilibrium states are not uniquely determined.

6.2 Similarities and differences between Steady State and Equilibrium

Next, we will discuss the similarities and differences between steady state and equilibrium. First, we need to clarify how steady state and equilibrium are described mathematically.

6.2.1 Mathematical Definitions

Steady state refers to a situation where all state variables in a system do not change with time, i.e.,

$$\frac{dx_i}{dt} = 0 \quad \text{for all state variables } x_i. \tag{6.1}$$

Equilibrium (specifically, detailed balance) refers to a condition where the transitions between different state variables are balanced. That is, for any state x_i, the total rate of transitions into x_i from all directly connected states x_j equals the total rate of transitions out of x_i to all directly connected states x_k. Mathematically,

$$\sum_j \nu_{ji} = \sum_k \nu_{ik}, \tag{6.2}$$

where ν_{ji} denotes the rate from state x_j to x_i, and ν_{ik} denotes the rate from state x_i to x_k. This condition is known as **detailed balance**.

6.2.2 Are Steady State and Equilibrium Equivalent?

Now, is steady state equivalent to equilibrium? For linear (chain) networks, yes; but for cyclic (ring) networks, not necessarily. We will discuss two examples separately.

Example 1: Linear Network $x_1 \leftrightarrow x_2 \leftrightarrow x_3$

In a linear network, the endpoints are x_1 and x_3. If the system is in a steady state, then for the endpoints, we have:

For x_1:

$$\frac{dx_1}{dt} = \nu_{21} - \nu_{12} = 0 \quad \Rightarrow \quad \nu_{21} = \nu_{12}. \tag{6.3}$$

For x_2:

$$\frac{dx_2}{dt} = (\nu_{12} + \nu_{32}) - (\nu_{21} + \nu_{23}) = (\nu_{12} - \nu_{21}) + (\nu_{32} - \nu_{23}) = 0. \tag{6.4}$$

Since $\nu_{12} - \nu_{21} = 0$ from the steady state condition of x_1, it follows that $\nu_{32} - \nu_{23} = 0$. Thus, the steady state condition propagates along the chain, ensuring that detailed balance holds at every node. Therefore, in a linear network, steady state implies equilibrium.

Example 2: Cyclic Network $x_1 \leftrightarrow x_2 \leftrightarrow x_3 \leftrightarrow x_1$

In a cyclic network, there are no endpoints. Consider the steady state condition for x_1:

$$\frac{dx_1}{dt} = (\nu_{21} + \nu_{31}) - (\nu_{12} + \nu_{13}) = (\nu_{21} - \nu_{12}) + (\nu_{31} - \nu_{13}) = 0. \tag{6.5}$$

This equation can be satisfied without requiring $\nu_{21} - \nu_{12} = 0$ and $\nu_{31} - \nu_{13} = 0$ individually. For example, if $\nu_{21} - \nu_{12} < 0$ and $\nu_{31} - \nu_{13} > 0$, their sum can still be zero. This corresponds to a net flow in the cycle: for instance, $x_1 \to x_2 \to x_3 \to x_1$, with more flow from x_3 to x_1 than from x_1 to x_3, and more flow from x_1 to x_2 than from x_2 to x_1. Thus, steady state does not necessarily imply equilibrium in a cyclic network.

6.2.3 Energy Consumption and Biological Implications

Achieving a steady state without equilibrium in a cyclic network requires energy input. This is due to the second law of thermodynamics: in a closed system, a net flow would increase entropy, so maintaining a non-equilibrium steady state requires external energy. Biological systems often exploit such non-equilibrium steady states in cyclic networks to perform functions.

Two classic examples are the **Citric Acid Cycle** (Krebs cycle) and the **Calvin-Benson Cycle**:

- In the **Citric Acid Cycle**, metabolites remain at approximately constant concentrations under stable conditions (steady state), but the cycle is not at equilibrium. It continuously consumes acetyl-CoA and produces CO_2, NADH, and $FADH_2$. If it were at equilibrium, no net production of these energy carriers would occur, preventing aerobic respiration and ATP synthesis.
- In the **Calvin-Benson Cycle**, metabolites are also in steady state, but the cycle consumes CO_2 and produces organic carbon for biomass synthesis. Equilibrium would halt carbon fixation. This cycle is driven by energy from light reactions, via NADPH.

The ability of these cycles to maintain steady-state concentrations despite external perturbations reflects their **robustness**.

6.3 Consider Equilibrium under Macro and Micro View

In the macroscopic perspective, equilibrium is described using equilibrium constants. In the microscopic perspective, equilibrium is described using statistical mechanics. We will now consider how to connect these two approaches.

We begin by outlining the tools we will use. In the macroscopic perspective, we will use the van't Hoff equation, which relates the Gibbs free energy change to the equilibrium constant. Its mathematical form is:

$$\Delta G = -kT \ln K_{eq} \tag{6.6}$$

where ΔG is the Gibbs free energy change, k is the Boltzmann constant, T is the temperature, and K_{eq} is the equilibrium constant for the reaction.

In the microscopic perspective, statistical mechanics uses the Boltzmann distribution as a fundamental postulate. The Boltzmann distribution is suitable for describing any non-quantum (or classical) system in thermodynamic equilibrium. The Boltzmann distribution states that the probability P_i of a system being in state i is:

$$P_i = \frac{1}{Z} e^{-E_i/kT} \tag{6.7}$$

where k is the Boltzmann constant, T is the temperature, and Z is the partition function. Here, the partition function serves primarily as a normalization factor to ensure that the probabilities sum to one. In my view, the Boltzmann distribution essentially tells us that as the energy of a state increases, its likelihood of existence decreases exponentially.

Macroscopic Thermodynamic Starting Point

Thermodynamics provides the relationship between the standard Gibbs free energy change ΔG° and the equilibrium constant K_{eq} (van 't Hoff isotherm):

$$\Delta G^\circ = -RT \ln K_{eq} \tag{6.8}$$

For the reaction $A \rightleftharpoons B$, the equilibrium constant is defined as:

$$K_{eq} = \frac{[B]}{[A]} \tag{6.9}$$

Statistical Mechanics Interpretation

In the framework of statistical mechanics, the concentration ratio can be interpreted as the probability ratio of the system being in different macroscopic states. For a randomly selected molecule:

$$\frac{[B]}{[A]} = \frac{P_B}{P_A} \tag{6.10}$$

where P_A and P_B represent the probabilities of the molecule being in state A and B, respectively.

According to the Boltzmann distribution, the probability of the system being in a particular macroscopic state is proportional to its partition function:

$$P_A \propto Z_A, \quad P_B \propto Z_B \tag{6.11}$$

Therefore, the probability ratio equals the partition function ratio:

$$\frac{P_B}{P_A} = \frac{Z_B}{Z_A} \tag{6.12}$$

Microscopic Nature of the Partition Function

The microscopic definition of the partition function Z is the sum of Boltzmann factors over all possible microstates. For macroscopic state B:

$$Z_B = \sum_{\text{all microstates } i \text{ belonging to } B} e^{-E_i^B/kT} \tag{6.13}$$

We can group microstates by energy. Assuming that at energy $E_{B,j}$, state B has $\Omega_B(E_{B,j})$ degenerate microstates, then:

$$Z_B = \sum_j \Omega_B(E_{B,j}) \cdot e^{-E_{B,j}/kT} \tag{6.14}$$

Similarly, for state A:

$$Z_A = \sum_j \Omega_A(E_{A,j}) \cdot e^{-E_{A,j}/kT} \tag{6.15}$$

Complete Derivation Chain

Combining equations (6.8) through (6.15), we obtain the complete derivation:

$$\begin{aligned}
\Delta G^\circ &= -RT \ln K_{eq} \quad \text{(Macroscopic thermodynamic relation)} \\
&= -RT \ln\left(\frac{[B]}{[A]}\right) \quad \text{(Equilibrium constant definition)} \\
&= -RT \ln\left(\frac{P_B}{P_A}\right) \quad \text{(Statistical interpretation)} \\
&= -RT \ln\left(\frac{Z_B}{Z_A}\right) \quad \text{(Boltzmann distribution)} \\
&= -RT \ln\left(\frac{\sum_j \Omega_B(E_{B,j}) \cdot e^{-E_{B,j}/kT}}{\sum_j \Omega_A(E_{A,j}) \cdot e^{-E_{A,j}/kT}}\right) \quad \text{(Microscopic nature of partition functions)}
\end{aligned}$$

Final Conclusion

We thus arrive at the microscopic statistical expression for the equilibrium constant:

$$\boxed{K_{eq} = \frac{Z_B}{Z_A} = \frac{\sum_j \Omega_B(E_{B,j}) \cdot e^{-E_{B,j}/kT}}{\sum_j \Omega_A(E_{A,j}) \cdot e^{-E_{A,j}/kT}}} \tag{6.16}$$

This result reveals the microscopic nature of the macroscopic equilibrium constant K_{eq}: it is determined by the ratio of statistical weights of competing states A and B, which in turn depend on their respective energy level structures (E) and state degeneracies (Ω).

6.4 Biological Examples: Enzyme Allosteric Regulation, Gene Regulation

Based on lecture notes (pages 7-14), this section applies equilibrium physical theory from previous parts to specific biological processes, demonstrating how to analyze single molecule states and directly calculate probability distributions using energies and weights.

6.4.1 Enzyme Catalytic Reaction: Michaelis-Menten Model

Model Framework

Consider enzyme E catalyzing substrate S to product P:

$$\mathrm{E} + \mathrm{S} \rightleftharpoons \mathrm{ES} \tag{6.17}$$

Using a lattice model with one enzyme and multiple substrate sites.

States and Weights

- **Free state** (E + S): weight $w_{\text{free}} \propto N_s e^{-E_f}$
- **Bound state** (ES): weight $w_{\text{bound}} \propto e^{-E_b}$

where N_s is the number of substrate molecules, E_f and E_b are energies of free and bound states.

Binding Probability Derivation

Binding probability:

$$p_{\text{bound}} = \frac{w_{\text{bound}}}{w_{\text{free}} + w_{\text{bound}}} = \frac{N_s e^{-\Delta E}}{1 + N_s e^{-\Delta E}} \tag{6.18}$$

where $\Delta E = E_f - E_b$.

Concentration Parameters

Define:

- Solution volume V, site volume Ω
- Substrate concentration $C_s = N_s/V$
- Dissociation constant $K_d = C_0 e^{\Delta E}$ (C_0 is standard concentration, 1M)

Rewriting binding probability:

$$p_{\text{bound}} = \frac{C_s/K_d}{1 + C_s/K_d} \tag{6.19}$$

Reaction Rate

Reaction rate is given by:

$$v = kE_{\text{tot}} p_{\text{bound}} = \frac{kE_{\text{tot}} C_s}{K_m + C_s} \tag{6.20}$$

where $K_m \approx K_d$, yielding the classical Michaelis-Menten equation.

Physical Interpretation

- Energy difference $\Delta E < 0$ indicates favorable binding
- Multiplicity N_s represents entropy contribution
- Chemical potential: $\mu \approx E_f + kT \ln(N_s/\Omega)$
- Free energy change: $\Delta G = \Delta E - T\Delta S$

6.4.2 Allosteric Regulation: MWC Model

Model Framework

MWC (Monod-Wyman-Changeux) model describes enzyme allosteric regulation:

- Two conformations: Active (A) and Inactive (I)
- Each conformation can bind substrate

States and Weights

Table 6.1 MWC Model State Weights

State	Energy	Weight
Active free (EA)	E_A	e^{-E_A}
Active bound (EA·S)	$E_A + E_{ab}$	$e^{-(E_A+E_{ab})}$
Inactive free (EI)	E_I	e^{-E_I}
Inactive bound (EI·S)	$E_I + E_{ib}$	$e^{-(E_I+E_{ib})}$

Considering substrate concentration C_s, bound state weights are multiplied by factor C_s/C_0.

Active Probability

Probability of active conformation:

$$p_{\text{active}} = \frac{e^{-E_A}\left(1+\frac{C_s}{K_A}\right)}{e^{-E_A}\left(1+\frac{C_s}{K_A}\right)+e^{-E_I}\left(1+\frac{C_s}{K_I}\right)} \tag{6.21}$$

where $K_A = C_0 e^{E_A - E_{ab}}$, $K_I = C_0 e^{E_I - E_{ib}}$.

Cooperativity Effects

When $K_A \ll K_I$ (tighter binding in active state), p_{active} changes significantly with C_s.

Introducing intersite interaction energy $E_{\text{int}} < 0$, binding probability for dimer system:

$$p_2 \propto \left(\frac{C_s}{K}\right)^2 e^{-E_{\text{int}}} \tag{6.22}$$

producing steeper transition curves (Hill coefficient $n > 1$).

Biological Significance

- Allostery enables enzymes to respond to substrate concentration changes
- Implements biological switching behavior
- Example: Hemoglobin oxygen binding curve

6.4.3 Gene Regulation: Lac Operon Model

Model Framework

Consider Lac operon transcriptional regulation with repressor and RNA polymerase competing for promoter binding.

States and Weights

- **No repressor state** (RNA polymerase bound):

$$w_{\text{pol}} \propto e^{-\Delta E_p} \frac{P}{N_{NS}} \tag{6.23}$$

- **With repressor state** (repressor bound):

$$w_{\text{rep}} \propto e^{-\Delta E_r} \frac{R}{N_{NS}} \tag{6.24}$$

where:

- ΔE_p, ΔE_r: binding energy differences
- P, R: RNA polymerase and repressor numbers
- N_{NS}: number of non-specific sites (multiplicity)

Transcription Probability

Transcription probability (RNA polymerase bound):

$$p_{\text{bound}} = \frac{e^{-\Delta E_p} P/N_{NS}}{e^{-\Delta E_p} P/N_{NS} + e^{-\Delta E_r} R/N_{NS} + \cdots} \tag{6.25}$$

Allosteric Extension

In homework problem 1, LacI repressor has active/inactive forms:

- Active form binds DNA, preventing transcription
- Inactive form does not bind DNA
- Inducer concentration c affects active form probability $p_{\text{active}}(c)$

Deriving fold-change formula:

$$\text{fold-change} = \frac{1}{1 + p_{\text{active}}(c) \frac{R}{N_{NS}} e^{-\Delta E_r}} \tag{6.26}$$

Model Advantages

- Parameters (ΔE_p, ΔE_r) can be measured from independent experiments
- No need for detailed kinetic mechanisms
- Provides quantitative predictive power

6.5 Beyond Equilibrium: Markov Chain

6.5.1 Motivation and Overall Logic

While equilibrium statistical mechanics provides powerful tools for analyzing biological systems at steady state, many crucial biological processes operate **beyond equilibrium** due to continuous energy input and driving forces. Biological systems are not closed–they exchange energy and matter with their environment, leading to behaviors that cannot be captured by equilibrium descriptions alone.

- **Key insight**: Biological systems utilize equilibrium in most components but are **sparsely driven** at critical control points
- **Examples**: Protein degradation with precise timing, phosphorylation cascades, metabolic networks with sustained fluxes
- **Challenge**: When detailed balance is broken, we need new tools to analyze steady states and dynamics

The Markov chain framework provides the mathematical foundation for analyzing these non-equilibrium systems, bridging the gap between equilibrium statistics and non-equilibrium dynamics.

6.5.2 Markov Chain Fundamentals

Definition and Derivation from Chemical Master Equation

For a continuous-time Markov chain describing molecular state transitions:

$$\frac{d\mathbf{P}}{dt} = \mathbf{QP} \tag{6.27}$$

$$Q_{ij} = \text{transition rate from state } j \text{ to state } i \quad (i \neq j) \tag{6.28}$$

$$Q_{jj} = -\sum_{i \neq j} Q_{ij} \quad \text{(conservation of probability)} \tag{6.29}$$

Properties:

- Probability conservation: $\mathbf{1}^T\mathbf{P} = 1$
- Column sum zero: $\mathbf{1}^T\mathbf{Q} = \mathbf{0}$
- Steady state: $\mathbf{Q\Pi} = \mathbf{0}$ (find null space of $\mathbf{Q}$)
- Ergodicity: Irreducible chain $\Rightarrow$ unique steady state $\mathbf{\Pi}$

Detailed Balance Condition:

$$Q_{ij}\Pi_j = Q_{ji}\Pi_i \quad \text{iff no cyclic fluxes (equilibrium)} \tag{6.30}$$

Biological Example: Phosphorylation Cascades

Consider kinase-phosphatase cycles with states representing phosphorylation levels $(0, 1, 2, \ldots)$. The Markov chain captures stochastic phosphorylation/dephosphorylation events:

Process	Transition	Rate
Phosphorylation	$S_n \to S_{n+1}$	$k_n[K]$
Dephosphorylation	$S_n \to S_{n-1}$	$\gamma_n[P]$

6.5.3 Handling Non-Markovian Systems

Many biological processes exhibit memory effects, violating the Markov assumption. We can **embed** these systems into Markov chains by extending the state space:

$$X(t) \to (X(t), X(t-\Delta t), X(t-2\Delta t), \ldots) \tag{6.31}$$

Application: Protein degradation with precise timing control, where current degradation probability depends on previous states.

6.5.4 Dynamic Analysis: Hitting Times

Definition and Computation

The hitting time τ_{ji} represents the expected time to first reach state i starting from state j:

$$\tau_{ji} = \frac{1}{|q_{jj}|} + \sum_{k \neq i} \frac{q_{jk}}{|q_{jj}|} \tau_{ki} \tag{6.32}$$

Matrix formulation: Remove row/column i from $\mathbf{Q}$ to obtain $\mathbf{Q}^{(i)}$, then solve:

$$\mathbf{Q}^{(i)} \boldsymbol{\tau}^{(i)} = -\mathbf{1} \tag{6.33}$$

where $\boldsymbol{\tau}^{(i)}$ is the vector of hitting times to state i from all other states.

Biological Application

In phosphorylation cascades, hitting times quantify:

- Signal propagation speed from unphosphorylated to fully phosphorylated state
- Response time to external stimuli
- Temporal precision of signaling pathways

6.5.5 Steady State Distribution and Finite State Projection

Steady State Calculation

For finite-state systems, solve $\mathbf{Q\Pi} = \mathbf{0}$ subject to $\sum_i \Pi_i = 1$.

Homework Example: Binding reaction $E + S \rightleftharpoons C$ yields steady state:

$$\Pi \propto \frac{K_d^{-N_C}}{N_E! N_S! N_C!} \quad \text{(Poisson-like distribution)} \tag{6.34}$$

At high molecular numbers, the most probable state approaches the deterministic equilibrium.

Finite State Projection

For systems with large or infinite state spaces, project the Chemical Master Equation onto a finite Markov chain:

$$\text{Infinite CME} \to \text{Finite Markov Chain} \to \text{Approximate } \boldsymbol{\Pi} \text{ and dynamics} \tag{6.35}$$

Application: Metabolic cascades with non-equilibrium fluxes that cannot be captured by equilibrium thermodynamics.

6.5.6 Why We Use Markov Chains for Simulation and Analysis

1. **Unified Framework**: Markov chains provide a consistent mathematical structure for analyzing both equilibrium and non-equilibrium systems
2. **Computational Tractability**: The matrix $\mathbf{Q}$ enables efficient numerical computation of steady states and dynamic properties
3. **Connection to Physical Laws**: Markov chains naturally emerge from the Chemical Master Equation, which derives from fundamental stochastic reaction kinetics
4. **Bridge Between Scales**: Captures molecular stochasticity while connecting to macroscopic observables through ergodicity
5. **Experimental Validation**: Hitting times and steady state distributions provide testable predictions for single-molecule and bulk experiments

6.5.7 Key Takeaways

- Markov chains extend equilibrium statistical mechanics to driven biological systems
- The framework captures essential non-equilibrium features: cyclic fluxes, broken detailed balance, and energy dissipation
- Both steady-state properties (distributions) and dynamic properties (hitting times) are accessible
- Finite state projection makes experimentally relevant computations feasible
- Biological implementation requires careful consideration of Markovian assumptions and potential state space extensions

Final Insight: "It takes a lot of driving to not look like equilibrium" – weak driving forces often yield behaviors indistinguishable from equilibrium, but biological systems exploit strong, sparse driving at critical control points.

Chapter 7

Stochastic Kinetics of Markov Chain and Computation Biomachines

Scribe: Xia Yao+Jiacheng Wei+Yihan Gong
Lecturer: Fangzhou Xiao
Lecture #08 2025-10-30

Contents

7.1 Equilibrium Physics of Bio-regulation

7.1.1 The Measurement Problem

The most practical motivation for equilibrium physics is the difficulty of measuring kinetic rates compared to thermodynamic energies.

1.The Kinetic Challenge

Consider a standard biological binding reaction, such as a gene (D) binding to a transcription factor (P), or an enzyme (E) binding to a substrate (S).

The dynamic model is described by the Law of Mass Action:

$$E + S \underset{k_-}{\overset{k_+}{\rightleftharpoons}} ES \tag{7.1}$$

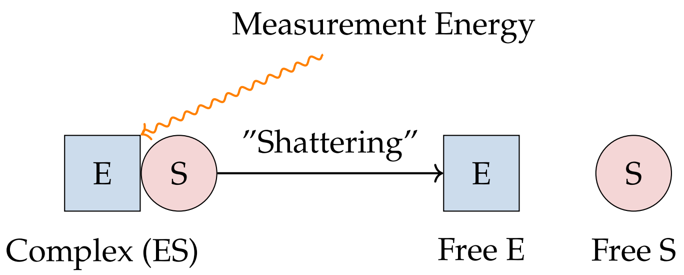
To model this dynamically, we need two parameters:

- k_+: The association rate constant.
- k_-: The dissociation rate constant.

Why is this hard to measure? To find k_+ and k_-, you must measure the concentrations of the free components (E, S) and the complex (ES) in real-time. However, the biological bond between E and S is usually non-covalent (e.g., hydrogen bonds, Van der Waals forces). These are **weak associations**.

The "Shattering" Effect (Mass Spectrometry) Common measurement tools, like Mass Spectrometry, require imparting energy to the molecules to detect them.

- When you apply energy to measure the system, the weak non-covalent bond breaks.
- You cannot distinguish whether the detected molecules were originally a complex (ES) or two separate molecules ($E + S$).
- The act of measurement destroys the structure you are trying to observe.



2.The Equilibrium Solution: Thermodynamics

Instead of trying to measure k_+ and k_- individually, we look at the system in equilibrium. Here, we care about the ratio, the **Dissociation Constant** (K_d):

The Equilibrium Relation

$$K_d = \frac{[E][S]}{[ES]} = \frac{k_-}{k_+} \tag{7.2}$$

Crucially, statistical physics relates this ratio to **Energy**. The probability of a state is proportional to the Boltzmann factor:

$$P \propto e^{-E_{state}/k_B T} \tag{7.3}$$

Therefore, K_d is directly related to the Binding Free Energy (ΔG):

Energy-Measurement Link

$$K_d \propto e^{\Delta G/k_B T} \tag{7.4}$$

Why is this easier?

- Energy changes (ΔG) correspond to the release or absorption of **Heat**.
- We can measure heat changes very accurately (e.g., Isothermal Titration Calorimetry) without needing to count individual molecules or preserve fragile bonds during detection.
- **Conclusion:** Equilibrium is a "Measurement Science." It transforms a hard counting problem into an easier calorimetry problem.

7.1.2 The Bio-Design Perspective

If you are an engineer or synthetic biologist trying to build a biological function (e.g., a genetic switch), equilibrium mechanisms are often superior.

1.Low Energy Cost

- **Non-Equilibrium:** Requires constant energy input (flux) to maintain a steady state (like keeping a lightbulb on).
- **Equilibrium:** Once the system settles, it stays there forever without consuming fuel (ATP). It is thermodynamically stable.

2.Low Mechanistic Complexity

- Designing a cycle that forces a reaction in one direction (Non-equilibrium) requires complex molecular machinery to couple with energy sources.
- Designing for equilibrium just requires mixing components that have the right binding affinity (ΔG). They will naturally find their way to the desired state.

7.1.3 The Analytical Perspective

From the perspective of system analysis (Mathematical Modeling), equilibrium provides powerful constraints that make unsolvable problems solvable.

1.Dynamics are Hard

- **Deterministic:** Systems of coupled differential equations are often non-linear and hard to solve analytically.
- **Stochastic:** If we treat it as a random process, the Master Equation is usually infinite-dimensional and impossible to solve exactly.

2.Equilibrium is Easy

If we assume the system is at equilibrium, we don't need to solve the dynamics. We know the answer immediately because of the **Boltzmann Law**:

$$P(State_i) = \frac{1}{Z} e^{-E_i/k_B T} \tag{7.5}$$

Where Z is the partition function.

Key Constraints Equilibrium imposes strict physical rules that simplify the math:

1. **Detailed Balance:** The flux between any two states is zero.

$$P_A \cdot k_{A\to B} = P_B \cdot k_{B\to A} \tag{7.6}$$

2. **No Cycle Fluxes:** There is no net rotation around reaction loops.

Summary of Section 1

Even though biology is dynamic, we use equilibrium physics because:

1. We can measure "Energies" (Heat) much easier than "Rates".
2. Equilibrium systems are energy-efficient and simpler to design.
3. Equilibrium math is solvable via the Boltzmann distribution.

7.2 Markov Chains: Stochastic Kinetics and non-equilibrium

Traditional thermodynamics often focuses on equilibrium—the state of a system after an infinite amount of time. However, in biological systems (like cells), we are often interested in kinetics (how fast things happen) and non-equilibrium behaviors (energy consumption).

The Kinetics Problem

Even if we know the final state of a cell, we often ask: *"If a signal changes in the environment, how long does it take for the cell to switch states?"*

- This is known as the First Passage Time problem.
- Simply adding up the inverse rates of steps ($1/k_1 + 1/k_2 \ldots$) is incorrect because stochastic systems can transition backward.

7.2.1 Non-Equilibrium Systems

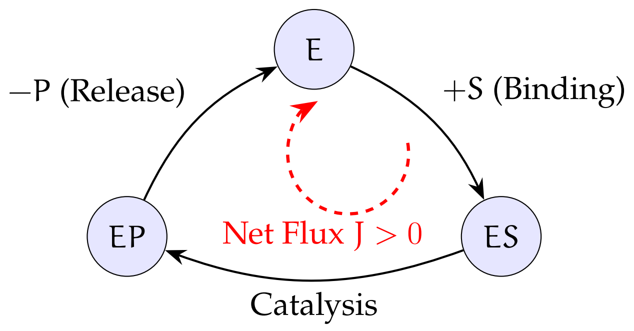
Energy vs. Non-Equilibrium

A common misconception is that if a system consumes energy (e.g., ATP hydrolysis), it is automatically "non-equilibrium."

- Correction: Many energy-consuming systems still exhibit behaviors that fit equilibrium models perfectly.
- True Indicator: A system requires a non-equilibrium model only when it exhibits behaviors impossible in equilibrium, such as non-monotonicity or net cyclic fluxes .

Example: The Enzymatic Cycle

Consider an enzyme E that converts Substrate S to Product P.



- Equilibrium: Forward and backward rates balance perfectly. No net rotation.
- Non-Equilibrium: High concentration of S drives the cycle continuously in a clockwise direction (Flux). This is a "current" in the state space.

7.2.2 Finite Markov Chains & The Master Equation

To model these kinetics, we use Finite State Markov Chains . This is a specific application of the Chemical Master Equation (CME).

From Concentrations to Probabilities

The lecture demonstrates that for 1st-order reactions , the macroscopic rate equation is mathematically identical to the single-molecule probability equation.

Consider a simple reversible transition:

$$A \underset{k_1}{\overset{k_2}{\rightleftharpoons}} B \tag{7.7}$$

(Note: Notation assumes k_2 creates A, k_1 consumes A).

1. Macroscopic View (Rate Equation) Let $A(t)$ and $B(t)$ be the number of molecules. The total number $N_{tot} = A + B$ is constant. The rate of change of A is:

$$\frac{dA}{dt} = k_2 B - k_1 A \tag{7.8}$$

Variable Definitions:

- dA/dt: Change in number of molecules of A per unit time.
- k_1: Rate constant for $A \to B$ (units: time^{-1}).
- k_2: Rate constant for $B \to A$ (units: time^{-1}).

2. Microscopic View (Probability) We define the probability (or fraction) of being in state A as:

$$P_A = \frac{A}{N_{tot}}, \quad P_B = \frac{B}{N_{tot}} \tag{7.9}$$

Since the system is linear, we can divide the rate equation by the constant N_{tot}:

$$\frac{1}{N_{tot}}\frac{dA}{dt} = k_2 \frac{B}{N_{tot}} - k_1 \frac{A}{N_{tot}} \tag{7.10}$$

Substituting the probability definitions:

$$\boxed{\frac{dP_A}{dt} = k_2 P_B - k_1 P_A} \tag{7.11}$$

Conclusion on Linearity

Because the reaction is linear (1st order), the math governing billions of molecules (concentration) is the exact same math governing one molecule's probability . We can simply model one molecule hopping between states.

7.2.3 Mathematical Formalism: The Q-Matrix

We generalize the system to n states using Linear Algebra.

The Probability Vector

Let $p(t)$ be a column vector representing the probability distribution at time t:

$$p(t) = \begin{bmatrix} P_1(t) \\ P_2(t) \\ \vdots \\ P_n(t) \end{bmatrix} \tag{7.12}$$

where $\sum_{i=1}^{n} P_i(t) = 1$.

The Transition Rate Matrix (Q)

The evolution of probability is given by the differential equation:

$$\frac{dp}{dt} = Qp \tag{7.13}$$

The matrix Q is constructed as follows:

- Off-diagonal terms (q_{ij} where $i \neq j$): The rate of transitioning **FROM** state j **TO** state i.
- Diagonal terms (q_{ii}): The negative sum of all outgoing rates from state i.

Mathematical Detail: Conservation of Probability

Since probability cannot be created or destroyed, the columns of Q must sum to zero.

$$\sum_{i=1}^{n} q_{ij} = 0 \tag{7.14}$$

This implies the diagonal term is:

$$q_{jj} = -\sum_{i \neq j} q_{ij} \tag{7.15}$$

Physical meaning: The rate of leaving state j (q_{jj}) is exactly the sum of rates going to all other states.

7.2.4 Steady States and Ergodicity

Steady State Distribution (π)

At steady state, the probability distribution does not change with time.

$$\frac{dp}{dt} = 0 \implies Q\pi = 0 \tag{7.16}$$

Here, π is the Steady State Distribution . Mathematically, it is the Right Eigenvector of matrix Q corresponding to the eigenvalue $\lambda = 0$.

Ergodicity

Definition: Strong Connectivity

A graph is strongly connected if there is a path from every state to every other state.

If the state transition graph is strongly connected (Irreducible):

1. The steady state π is Unique .
2. The system is Ergodic .

Implication for Simulation: Due to ergodicity, the Time Average equals the Ensemble Average .

Time Average of 1 trajectory = Average over infinite populations

You only need to simulate **one** single molecule trajectory for a long time to calculate the distribution of the entire population.

7.2.5 Detailed Balance (Equilibrium vs. Steady State)

A system can be in a steady state (constant concentrations) without being in thermodynamic equilibrium (e.g., a battery powering a circuit).

Detailed Balance Condition

For a system to be in true Equilibrium , the flux between any two specific states must balance out to zero.

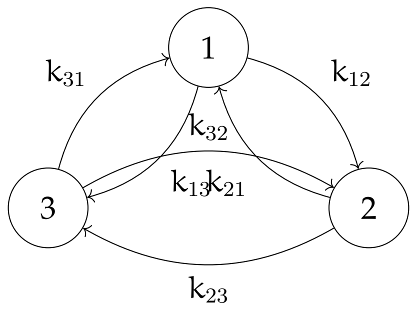
$$\text{Flux}_{j\to i} = \text{Flux}_{i\to j} \tag{7.17}$$

$$\pi_j q_{ij} = \pi_i q_{ji} \tag{7.18}$$

Cycle Condition (Kolmogorov Criterion)

An easier way to check for equilibrium is to look at loops in the graph. For any closed cycle (e.g., $1 \to 2 \to 3 \to 1$):

$$\text{Product of Clockwise Rates} = \text{Product of Counter-Clockwise Rates} \tag{7.19}$$



Equilibrium Requirement:

$$k_{12} \cdot k_{23} \cdot k_{31} = k_{13} \cdot k_{32} \cdot k_{21} \tag{7.20}$$

If this equation does not hold, there is a net flux, and the system is "Non-Equilibrium".

7.2.6 The Philosophy of "State": Markovian vs. Non-Markovian

There is a debate in biology that biological systems depend on history (memory) and are therefore "Non-Markovian."

The Lecturer's Counter-Argument

The lecturer argues that all dynamical systems are Markovian if you define the "State" correctly.

Definition: State

A State is a collection of variables that summarizes **all** necessary information from the past to predict the future evolution of the system.

Example: Hidden Variables

If a system appears to depend on history (x_t depends on x_{t-1} and x_{t-2}), it is only because we defined the state too narrowly.

- Narrow View (Non-Markovian): State = $[x_t]$. The future depends on the past history.
- Expanded View (Markovian): State = $\begin{bmatrix} x_t \\ x_{t-1} \end{bmatrix}$. Now, the current vector contains all the information needed for the next step.

Analogy: In Newtonian mechanics, position x alone is not a state (need velocity). But (x, v) is a state.

Conclusion: When modeling cell biology, assuming a Markov chain is not a restrictive assumption about "lack of memory." It is simply an assumption that the variables we chose (e.g., gene expression levels) are sufficient to define the system's future.

7.2.7 First-Passage Time to an Active State

Consider a CTMC with discrete states. One state is designated as **active** (state A). We want to compute: **How long does it take to reach A starting from some initial state i?**

Notation:

- $X(t)$: state of the system at time t
- $Q = (q_{ij})$: generator matrix
  - $q_{ij} \geqslant 0$ for $i \neq j$ (transition rate $i \to j$)
  - $q_{ii} = -\sum_{j \neq i} q_{ij}$ (total exit rate from i)
- First-passage time: $T_i = \inf\{t \geqslant 0 : X(t) = A \mid X(0) = i\}$
- Mean first-passage time: $m_i = \mathbb{E}[T_i]$
- CDF: $F_i(t) = P(T_i \leqslant t)$

Mean First-Passage Time Calculation

Intuitive Derivation: Starting from state $i \neq A$:

1. Wait in state i for exponentially distributed time with rate $-q_{ii}$. Mean waiting time = $1/(-q_{ii})$.
2. Jump to state $j \neq i$ with probability $\frac{q_{ij}}{-q_{ii}}$.
3. After arriving at j:
   - If $j = A$: done (extra time = 0)
   - If $j \neq A$: need additional mean time m_j

Thus:

$$m_i = \frac{1}{-q_{ii}} + \sum_{j \neq i} \frac{q_{ij}}{-q_{ii}} m_j \tag{7.21}$$

Multiply by $-q_{ii}$:

$$-q_{ii} m_i = 1 + \sum_{j \neq i} q_{ij} m_j \tag{7.22}$$

Since $m_A = 0$, we separate terms:

$$\boxed{-q_{ii} m_i = 1 + \sum_{\substack{j \neq i \\ j \neq A}} q_{ij} m_j} \tag{7.23}$$

This is a system of linear equations for m_i ($i \neq A$).

Matrix Formulation: Make state A absorbing: set $q_{Aj} = 0$ for all j.

Partition states: **transient states** T and **absorbing state** A. Reorder so transient states come first:

$$Q = \begin{pmatrix} \tilde{Q} & \mathbf{r} \\ \mathbf{0} & 0 \end{pmatrix}$$

- $\tilde{Q}$: $n_T \times n_T$ matrix for transient states
- $\mathbf{r}$: exit rates from transient to absorbing state
- $\mathbf{0}$: row for A

Let $\mathbf{m}$ = vector of m_i for $i \in T$, and $\mathbf{1}$ = column vector of ones.

Then:

$$\boxed{\mathbf{m} = (-\tilde{Q})^{-1}\mathbf{1}} \tag{7.24}$$

Proof: From $-q_{ii}m_i = 1 + \sum_{j\in T, j\neq i} q_{ij}m_j$, rewrite as:

$$(-q_{ii})m_i - \sum_{j\in T, j\neq i} q_{ij}m_j = 1$$

This is the i-th row of $(-\tilde{Q})\mathbf{m} = \mathbf{1}$.

7.2.8 Distribution and Example

Distribution of First-Passage Time

Let $F_i(t) = P(T_i \leqslant t)$.

Backward Equation: Condition on first jump:

$$\begin{aligned} F_i(t) &= \int_0^t (-q_{ii})e^{q_{ii}s}\left[\sum_j \frac{q_{ij}}{-q_{ii}}F_j(t-s)\right]ds \\ &= \int_0^t e^{q_{ii}s}\left[\sum_{j\neq i} q_{ij}F_j(t-s)\right]ds \end{aligned}$$

Differentiate with respect to t:

$$\boxed{\frac{dF_i}{dt} = q_{ii}F_i(t) + \sum_{j\neq i} q_{ij}F_j(t)} \tag{7.25}$$

Boundary Conditions:

- $F_A(t) = 1$ for $t \geqslant 0$
- $F_i(0) = 0$ for $i \neq A$

Matrix Form: Let $\mathbf{F}(t)$ = vector of $F_i(t)$ for $i \in T$. Then:

$$\frac{d\mathbf{F}}{dt} = \tilde{Q}\mathbf{F}(t) + \mathbf{r} \tag{7.26}$$

with $\mathbf{F}(0) = \mathbf{0}$.

Solution using matrix exponential:

$$\mathbf{F}(t) = \int_0^t e^{\tilde{Q}s}\mathbf{r}\,ds \tag{7.27}$$

If $\tilde{Q}$ is invertible:

$$\mathbf{F}(t) = \left(e^{\tilde{Q}t} - I\right)\tilde{Q}^{-1}\mathbf{r} \tag{7.28}$$

Example: Three-State System

System Description: States: 1 (initial), 2, A (active).

Transition rates:

$$\begin{gathered} 1 \xrightarrow{k_1} 2 \\ 2 \xrightarrow{k_2} A \\ 2 \xrightarrow{k_b} 1 \\ 1 \xrightarrow{k_f} A \end{gathered}$$

Generator Matrix:

$$Q = \begin{pmatrix} -(k_1 + k_f) & k_1 & k_f \\ k_b & -(k_b + k_2) & k_2 \\ 0 & 0 & 0 \end{pmatrix}$$

Mean First-Passage Times: Transient states: $\{1, 2\}$.

$$\tilde{Q} = \begin{pmatrix} -(k_1 + k_f) & k_1 \\ k_b & -(k_b + k_2) \end{pmatrix}$$

$$-\tilde{Q} = \begin{pmatrix} k_1 + k_f & -k_1 \\ -k_b & k_b + k_2 \end{pmatrix}$$

$$\mathbf{m} = (-\tilde{Q})^{-1}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

The explicit solution:

$$\begin{aligned} \det(-\tilde{Q}) &= (k_1 + k_f)(k_b + k_2) - k_1 k_b \\ &= k_1 k_2 + k_f k_b + k_f k_2 \end{aligned}$$

$$\mathbf{m} = \frac{1}{\det(-\tilde{Q})} \begin{pmatrix} k_b + k_2 + k_1 \\ k_b + k_1 + k_f \end{pmatrix}$$

So:

$$m_1 = \frac{k_b + k_2 + k_1}{k_1 k_2 + k_f k_b + k_f k_2}$$
$$m_2 = \frac{k_b + k_1 + k_f}{k_1 k_2 + k_f k_b + k_f k_2}$$

7.3 Computation Biomachine

Again like in adaptation biomachine, when we consider biological systems, we often consider they have distinct functions, like computation. To achieve such functions, we have built engineered machines, eg, computer. Since biological systems also need to achieve the same tasks, the structures and design principles of engineered machines could be used to better understand why biological systems are built in that way.

7.3.1 Today's Computation biomachine

Biological systems need to process information to respond to environments.

- Example:
  - activate a gene expression
  - to take up new nutrient
  - start a stringent response when harsh conditions hit
  - become a new cell type when growth factors say "differentiate"

So cells for sure need to do computation, but how is it done? How complex a computation does the cell achieve in that way? In ENgineered machines, we built computational units, and then linked them up for the larger and larger computation.

And now let us search for the existing biological computational unit.

logic gates

We built computers consisting of binary logic gates to perform computation. Maybe cells do the same?

- Example:
  - AND Gate
  - OR gate
  - NOT gate

Cells have chemical reactions inside, maybe implement gates via chemical reactions (catalysis)? Input and output turn out to be the concentration of species.

- '0' corresponds to low conc relative to the total conc
- '1' corresponds to high conc relative to the total conc

- Example:
  - The variable Y is represented in two forms, Y0 and Y1. (e.g., like phosphorylation and dephosphoorylation)
  $$Y0 \Leftrightarrow Y1$$
  - Then 'transition' of a signal can be done by catalysis of production and degradation.

How to build an universal turing machine out of logic gate? Namely, we can do whatever computation we want. The answer is NAND gate.

- NAND gate

$$
\begin{aligned}
X_0 + Y_0 + Z_0 &\rightarrow X_0 + Y_0 + Z_1 \\
X_0 + Y_1 + Z_0 &\rightarrow X_0 + Y_1 + Z_1 \\
X_1 + Y_0 + Z_0 &\rightarrow X_1 + Y_0 + Z_1 \\
X_1 + Y_1 + Z_1 &\rightarrow X_1 + Y_1 + Z_0
\end{aligned}
$$

X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

So Chemical reaction (catalysis) networks are Turing universal!

Cells can perform logic gate computations via catalytic chemical reactions. In fact, they can perform arbitrarily complex computations!

- Points to worry:
  - Signal may get corrupted in a cascade (repeat / restore).
  - Such reactions may be hard to implement (e.g. not trimolecular; need to make them first- or second-order).
  - Reactions are stochastic / discrete / noisy (in fact that's even better ...).
  - Maybe reactions are more powerful:
    * Use mass–action catalytic reactions themselves as computational units (similar, maybe a bit better).
- WAIT! But how complex can a cell be in this way?
  - Number of catalytic reactions $\approx$ number of enzyme types.
  - This is about 10^3 (e.g. $\sim 5 \times 10^3$ reactions in databases for all known enzymes, $\sim 10^3$ in a human cell).
  - If we need about 2–8 reactions per gate, that gives $\sim 3 \times 10^2$ logic gates.
- Compare:
  - 1972: first pocket calculators, $\sim 3 \times 10^3$ transistors, with about 2–8 transistors per gate (e.g. 4 for a NAND gate) $\Rightarrow$ roughly 4-bit data bandwidth (one decimal digit at a time).
  - Moon landing computers (1970s): $\sim 5 \times 10^3$ gates ($\sim 20\times$ more than a cell in this estimate).
  - Nowadays: microcontroller chips have $\sim 10^6$ gates.
  - Intel laptop chips typically have $\sim 10^9$ gates ($\sim 10^3\times$ more than a cell).

Maybe cells do computations in smarter ways? More efficient computing than logic gates?

Neural networks

In recent years (2010 on wards), artificial neural networks (ANNs) have taken over in the complexity of certain computational tasks, e.g. computer vision (image recognition), language models and translation, robotics, etc.

ANNs are made more powerful by their *depth*. Instead of just linking computational units in simple cascades, we put them into *layers*, so that the network can achieve an exponential-in-layers growth in representational complexity. The idea is to use layers to build a very complex input–output map that represents the solutions of a computational task, and then let the ANN learn/represent this map.

- The computational units of ANNs are *linear threshold units* (LTUs, perceptrons). For an input vector $x = (x_1, \dots, x_n)$ and a weight vector $w = (w_1, \dots, w_n)$, the output y is

$$y = \begin{cases} 1, & w^\top x = \sum_i w_i x_i > \theta, \\ 0, & w^\top x = \sum_i w_i x_i \leqslant \theta, \end{cases}$$

  where θ is the threshold.

- Example: an LTU network can implement the XOR function $z = \mathrm{XOR}(x, y)$ with two binary inputs $x, y \in \{0, 1\}$. Its truth table is

x	y	z
0	0	0
1	0	1
0	1	1
1	1	0

- How to implement an LTU in cells? We need many inputs to be combined and weighted. A natural candidate mechanism is *gene regulation*: multiple activators and repressors bind to DNA and jointly control the expression level of a gene.

Gene regulatory networks (GRNs) as an implementation:

- **Transcription**

$$G_i \xrightarrow{k_{tx,i}} G_i + T_i,$$

  where G_i is gene i and T_i is its transcript.

- **Translation**

$$T_i \xrightarrow{k_{tl,i}} T_i + R_i,$$

  where R_i is the regulatory species (e.g. a transcription factor).

- **Degradation**

$$T_i \xrightarrow{k_{rd,i}} \varnothing, \qquad R_i \xrightarrow{k_{pd,i}} \varnothing.$$

- **Typical rates**

$$k_{tx,i},\ k_{tl,i} \sim 10^{-3}\ \mathrm{s}^{-1} \quad \text{(per molecule)},$$
$$k_{rd,i},\ k_{pd,i} \sim 10^{-3}\ \mathrm{s}^{-1} \quad \text{(e.g. ~ 20 min timescale)}.$$

- **Regulation of transcription** by regulator R_j:

$$R_j + G_i^0 \underset{k_{ji}^-}{\overset{k_{ji}^+}{\rightleftharpoons}} G_{ij},$$

  where G_i^0 is the unbound promoter state and G_{ij} the promoter bound by regulator j. Typically

$$k_{ji}^+ \sim \frac{1}{\mathrm{nM} \cdot \mathrm{s}}, \qquad k_{ji}^- = K_M\, k_{ji}^+,$$

  with binding constant (dissociation constant) K_M in the range 0.1–10^5 nM.

- Regulation among transcription factors:

$$R_i + L_j \underset{k_{ij}^-}{\overset{k_{ij}^+}{\rightleftharpoons}} D_{ij},$$

$$D_{ij} \xrightarrow{k_{ij}^{pd}} \varnothing,$$

$$D_{ij} + G_k \underset{k_{ijk}^-}{\overset{k_{ijk}^+}{\rightleftharpoons}} G_{ijk}.$$

  These reactions allow transcription factors to form dimers D_{ij} and regulate different genes G_k, effectively changing their transcription rates.

- LTU in titration GRN:

  Represent inputs by normalized variables

$$x_j = \frac{I_j}{\bar{I}_j},$$

  where I_j is the concentration of input j and $\bar{I}_j$ is a reference value.

  Assumptions:

  - I_j act as repressors, so they only affect transcription rates through binding (no extra production term, $k_{tx}^j = 0$).
  - Tight binding between transcription factors R_i and $R_{i'}$, i.e. $K_M^{ii'} \to 0$.
  - Dimers $D_{ii'}$ do not bind promoters G_i or $G_{i'}$, so promoter states with dimers (e.g. $G_{ii'}$, $G_{i'i}$) can be ignored.

  Idea: competitive binding among transcription factors R_i, $R_{i'}$ forms a decision unit that behaves like a linear threshold unit.

- Binding equilibrium with inputs:

  For repressor I_j binding to promoter G_i, let

$$G_{ji}^c = \text{promoter } i \text{ bound by input } j.$$

  At equilibrium,

$$G_{ji}^c = \frac{I_j G_i}{K_M^{ji}}.$$

  The total amount of promoter for gene i is

$$G_i^{tot} = G_i^c + \sum_j G_{ji}^c = G_i + \sum_j \frac{I_j G_i}{K_M^{ji}}$$
$$= G_i \left(1 + \sum_j \frac{I_j}{K_M^{ji}} \right).$$

Hence

$$G_i = \frac{G_i^{tot}}{1 + \sum_j I_j / K_M^{ji}}.$$

- Gene expression of regulator R_i:

  Let

  $$R_{i,tot} = R_i + D_{ii'}$$

  be the total amount of R_i (free plus dimer). Its production–degradation dynamics are

  $$\dot{R}_{i,tot} = k_{tl} T_i - k_{rd} R_{i,tot}.$$

  At steady state,

  $$R_{i,tot} = \frac{k_{tl}}{k_{rd}} T_i.$$

  For the transcript T_i,

  $$\dot{T}_i = g_i k_{tx} G_i - k_{pd} T_i \quad \Rightarrow \quad T_i = \frac{k_{tx}}{k_{pd}} g_i G_i$$

  at steady state. Combining the two expressions gives

  $$R_{i,tot} = \frac{k_{tx} k_{tl}}{k_{pd} k_{rd}} g_i G_i \equiv g_i G_i,$$

  where the prefactor is absorbed into an effective gain g_i (typically of order 1 per copy of G_i^{tot}). Using the expression for G_i,

  $$R_{i,tot} = g_i \frac{G_i^{tot}}{1 + \sum_j I_j / K_M^{ji}}.$$

- Competitive binding / titration between R_i and $R_{i'}$:

  For very tight binding between R_i and $R_{i'}$, the dimer concentration and the remaining free R_i can be approximated by

  $$D_{ii'} = \min\{R_{i,tot}, R_{i',tot}\},$$

  and

  $$R_i = \begin{cases} 0, & R_{i,tot} < R_{i',tot}, \\ R_{i,tot} - R_{i',tot}, & \text{otherwise.} \end{cases}$$

  This titration mechanism implements a nonlinear, threshold-like dependence of the free regulator R_i on the inputs, which is what allows the GRN to behave like an LTU.

- Decision boundary.

  There are regulators R_i' that can bind to the output gene G_0. If $R'_{i,tot} > R_{i,tot}$ (and binding is tight), R_i' represses G_0 and the output is 0. In this case we require

  $$\frac{g_i'}{1 + \sum_j I_j / K_M^{ji'}} > \frac{g_i}{1 + \sum_j I_j / K_M^{ji}}$$

  as the condition for output $= 0$.

Rearranging gives

$$g_i'\Big(1+\sum_j \frac{I_j}{K_M^{ji'}}\Big) > g_i\Big(1+\sum_j \frac{I_j}{K_M^{ji}}\Big)$$
$$(g_i' - g_i) > \sum_j I_j\Big(\frac{g_i'}{K_M^{ji'}} - \frac{g_i}{K_M^{ji}}\Big).$$

Introducing normalized inputs

$$x_j = \frac{I_j}{m},$$

we obtain

$$\underbrace{\frac{g_i' - g_i}{m}}_{\theta_i} > \sum_j x_j \underbrace{\Big(\frac{g_i'}{K_M^{ji'}} - \frac{g_i}{K_M^{ji}}\Big)}_{w_j}.$$

Thus, for output 0,

$$\sum_j w_j x_j < \theta_i,$$

which is exactly the decision rule of a linear threshold unit (LTU).

- That's great: we can build an LTU out of just three genes. Cells can in principle achieve much more efficient computation using neural networks built from such genetic units.
- But wait — the complexity of an ANN scales with the number of *weights*.

For a gene regulatory network (GRN), the effective number of weights is roughly

$$\#\text{connections in GRN} \approx \#\text{TFs per gene} \times \#\text{genes}.$$

A rough comparison:

	E. coli	**Yeast**	**Human**
# TFs	~ 300 (7%)	~ 200 (3%)	~ 1800
# TFs per gene	1–2 (0–10)	1–2 (0–20)	10–12 (0–100)
Estimated #weights	$\sim 10^2$	$\sim 10^3$	$\sim 10^4$

For ANNs, the number of weights has grown dramatically:

	1990s	2010s	2020s
Example models	LeNet, MLP on MNIST	AlexNet, VGG-16, ResNet on ImageNet	Transformers (GPT-1–4)
#weights	$\sim 10^3$	$\sim 10^4$–10^6	$\sim 10^7$–10^{10}

So even human cells, with only $\mathcal{O}(10^4)$ effective weights, would correspond to just a small fraction of a classic CNN such as LeNet doing handwritten digit recognition.

Something seems off: cells must be processing information much more efficiently than a naive weight-count comparison with ANNs would suggest.

Side note: binding is much more efficient Binding can implement a rectified–linear (ReLU)–type operation much more efficiently than catalysis or gene expression.

Define

$$\mathrm{ReLU}(x) = \begin{cases} x, & x > 0, \\ 0, & \text{otherwise.} \end{cases}$$

Consider tight binding between two species X and Y:

$$X + Y \rightleftharpoons D.$$

Let X_{tot} and Y_{tot} be their total amounts. Under strong binding, the free X concentration satisfies

$$X_{\text{free}} = \begin{cases} X_{\text{tot}} - Y_{\text{tot}}, & X_{\text{tot}} > Y_{\text{tot}}, \\ 0, & \text{otherwise,} \end{cases}$$

so

$$X_{\text{free}} = f(X_{\text{tot}}, Y_{\text{tot}}) = \mathrm{ReLU}(X_{\text{tot}} - Y_{\text{tot}}).$$

Composition of such functions f can achieve arbitrarily complex behaviour, e.g.

- any piecewise–linear function,
- any logic gate,
- any LTU, etc.

The number of binding reactions in cells is huge. Very roughly:

	E. coli	Yeast	Mammalian
protein–protein	10^4	10^5–10^6	10^6–10^7
enzyme–substrate / cofactor	10^5		
protein–DNA		10^4–10^5	
protein–RNA			
protein–lipid / ion . . .			
Total	10^5–10^6	10^6–10^7	10^7–10^8

Computational tasks in Bio vs Engineered

- Logic gates in computers.
  - Task: logical deduction / symbolic computation.
  - Question: do cells need to do logical deduction?
- LTUs in ANNs.
  - Task: represent / fit / learn complex input–output maps.
  - Question: do cells need to do this kind of generic function approximation?
- One task we clearly see cells doing:
  - In response to changing environments, cells perform different actions, often via changes in gene expression.
  - So at least, a cell needs to encode *one* complex input–output map (though not an arbitrary one) that maps environmental conditions to appropriate actions.

Chapter 8

Computation Biomachine

Scribe: Kaijun WANG, Xinyu WANG, Xiaowen ZHANG
Lecturer: Fangzhou Xiao
Lecture #10 2025.1113

Contents

Again, like what we discussed in adaptation biomachine, when we consider biological systems, we often consider they have distinct functions. e.g. computation. To achieve such functions, we have built engineered machines. e.g. computers. Since biological systems also need to achieve the same tasks, the structures and design principles of engineered machines could be used to better understand why biological systems are built in this way. Although the functions of engineered machines and biomachines are the same (hopefully), their implementations can be different.
Biological systems need to process information to respond to environments. e.g. activate a gene expression to take up new nutrient, start a stringent response when harsh conditions hit because a new cell type when growth factors say ”differentials”...

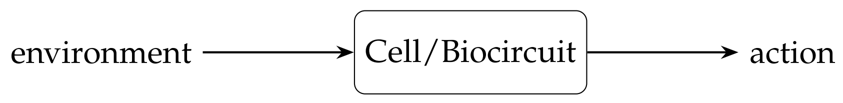


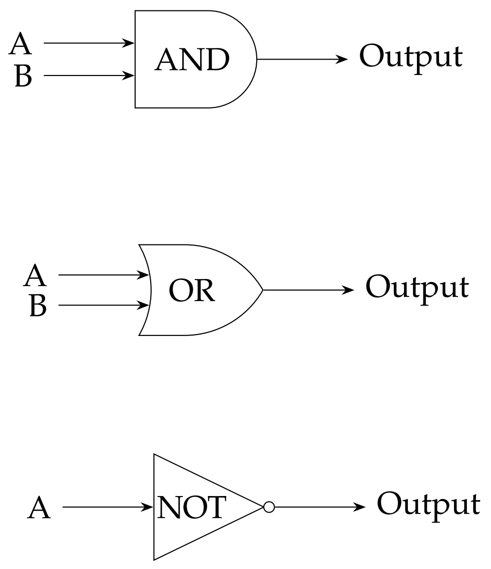
Figure 8.1

How are the computations done? How complex a computation the cell achieve in this way? In engineered machines, we build computational units, then link them up for larger and larger computations. In the following sections, we will search for biological computational units.

8.1 Logic gate computation in cells

8.1.1 Basic logic gate

We built computers consisting of binary logic gates to perform computation. Both the inputs and the outputs of logic gates are binary, either true(1) or false(0). There are three basic logic gates: AND, OR, NOT.



A	B	A·B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

A	−A
0	1
1	0

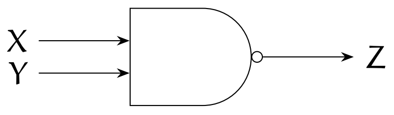
Figure 8.2 Diagrams of AND(top), OR(middle and NOT(down) gates and their corresponding truth tables

8.1.2 Logic gate implementation with catalysis reactions

Maybe cells can do the same thing. Let's assume that cells encode 0/1 as low/high concentration, X_0, X_1. And cells may implement logic gates via catalysis reactions. The following equations show a series of catalysis reactions that achieves a NAND gate. We should note that NAND gate itself is enough to build universal Turing machines.

8.1.3 Comparison of complexity with engineered machines

But let us slow down here. If we assume that three substrates and four reactions are needed to implement a single NAND gate, then a cell—with roughly 10^3 types of enzymes—could effectively generate only about (3×10^2) logic gates, even if all enzymes were devoted to computation. When we compare this level



X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

$$X_0 + Y_0 + Z_0 \to X_0 + Y_0 + Z_1$$
$$X_0 + Y_1 + Z_0 \to X_0 + Y_1 + Z_1$$
$$X_1 + Y_0 + Z_0 \to X_1 + Y_0 + Z_1$$
$$X_1 + Y_1 + Z_1 \to X_1 + Y_1 + Z_0$$

Figure 8.3 Diagrams of NAND gate, truth table and catalysis reactions implementation of NAND

of complexity with engineered machines, early electronics such as a 1972 pocket calculator (about 3×10^3 transistors) and the Apollo Guidance Computer used for the moon landing (around 5×10^3 logic gates) exceeded cellular computational complexity only modestly. As technology progressed, microcontrollers with limited functionality reached about 10^6 gates, and modern laptop processors now contain on the order of 10^9 gates. At the very least, we should recognize that cells operate in a far more complex and dynamic environment than a microcontroller does—yet there remains a vast gap in raw computational complexity between a cell and a modern microcontroller. Thus, we conclude that if cells only use logic gates for computation, the complexity of logic gates is not enough. Maybe cells do computations in smarter ways to achieve more efficient computing than logic gates.

Machine	**Complexity**
cell (all the enzymes process information with gates)	3×10^2 gates
pocket calculator in 1972	3×10^3 transistors
moonlanding in 1970s	5×10^3 gates
microcontroller	10^6 gates
intel laptop chip	10^9 gates

Table 8.1 Complexity gap between cell and real engineered machines when assuming cells process information with logic gates only.

8.2 Neural networks

Recent years (2010 onwards). Artificial neural networks (ANN) have taken over in complexity of certain computational tasks, e.g. computer vision, language models/translation, image recognition, robotics… ANNs are more powerful by "depth". Instead of just linking computational units as cascades,

put them in layers, so they have exponential–in–layer power of representational complexity.
Idea: use layers to build a very complex input–output map that represents the solution of a computa-

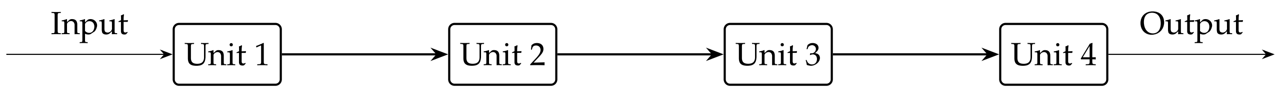


Figure 8.4 Diagrams linking computational units as cascades

tional task. Then learn/represent this map by ANN. (Perceptron.) Computational units of ANN are linear threshold units (LTU).

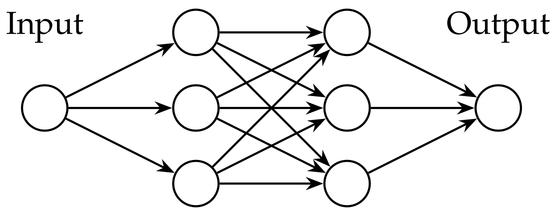


Figure 8.5 Diagrams linking computational units in layers

$$w^\top x = \sum_i w_i x_i > \theta \ : \ y = 1$$

$$w^\top x = \sum_i w_i x_i < \theta \ : \ y = 0$$

$G_i \xrightarrow{k_{tx}} G_i + T_i$	Transcription
$T_i \xrightarrow{k_{tl}} T_i + R_i$	Translation
$T_i \xrightarrow{k_{rd}} \emptyset$	Transcript degradation
$R_i \xrightarrow{k_{pd}} \emptyset$	Regulatory species degradation
$R_j + G_i \underset{k_{ji}^-}{\overset{k_{ji}^+}{\rightleftharpoons}} G_{ij}$	Regulation of transcription
$R_i + R_j \underset{k_{ij}^-}{\overset{k_{ij}^+}{\rightleftharpoons}} D_{ij}$	Regulation among transcription factors
$D_{ij} + G_k \underset{k_k^-}{\overset{k_k^+}{\rightleftharpoons}} G_{ijk}$	Binding of TF complex to gene
$G_{ij} \xrightarrow{k_{ji}^{tx}} G_{ij} + T_i$	Result of regulation (Rate change)
$G_{ijk} \xrightarrow{k_{ijk}^{tx}} G_{ijk} + T_k$	Result of regulation (Rate change)

8.2.1 LTU: Titration GRN Implementation

After given the definitions of the GRN, in this section we try to make it clear that why it represents a LTU(or a neuron element) for further construction of an Artificial Neural Network.

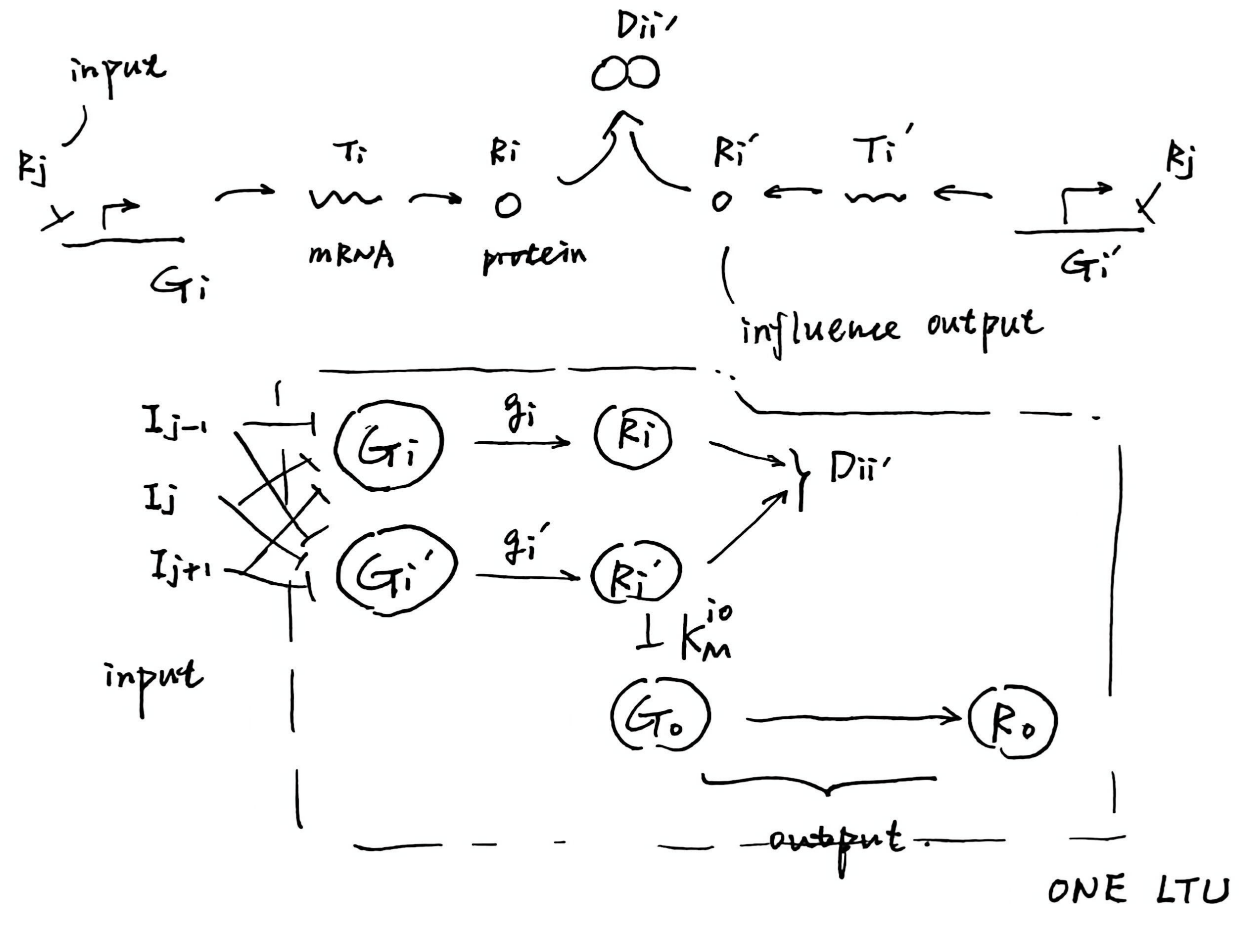


Figure 8.6 LTU implementation with gene regulatory network

We begin by analyzing the two important species: a specific gene (G_i) and a specific regulator (R_i). (In this case we assume the regulator is an repressor of gene). To make the intuition clear, we denote the input repressor as I (which is the resulting output of the previous layer), and the regulators in current layer would be the output variable, which we denote R.

Single Gene G_i We denote the total amount of gene as $G_{tot,i}$, consisting the free state G_i and regulated state G_{ji}. We can first solve the regulated state quantity using the equilibrium of the binding reaction:

$$I_i + G_j \to G_{ij}, \qquad K_M^{ij} := \frac{I_i G_j}{G_{ij}}.$$

Thus the regulated (bound) state is

$$G_{ij} = \frac{I_i\, G_j}{K_M^{ij}}.$$

So the total amount of gene i would be:

$$\begin{aligned} G_{tot,i} &= G_i + \sum_j G_{ij} \\ &= G_i + \sum_j \frac{I_j\, G_i}{K_M^{ij}} \\ &= G_i \left(1 + \sum_j \frac{1}{K_M^{ij}} I_j \right), \end{aligned} \tag{8.1}$$

hence

$$G_i = \frac{G_{tot,i}}{1 + \sum_j \frac{1}{K_M^{ij}} I_j}.$$

Notice that the mathematical form of this sum $\sum_j \frac{1}{K_M^{ij}} I_j$ is analogous to the ANN, with the form of a weighted linear combination. (i is the current node, and the j sums up all the inputs). This is the exact result of design that we use the multiple regulator inputs to assemble the input signals of a single LTU.

Single Regulator R_i Similarly the regulator would have free state (R_i) and the muted dimer ($D_{ii'}$, formed by its complementary regulator R_i'). At steady state, the total regulator produced from gene G_i can be written as

$$R_{tot,i} = g_i\, G_i \frac{k_{+l} k_{+x}}{k_{pd} k_{rd}}.$$

Substituting the expression of G_i above,

$$R_{tot,i} = g_i \frac{k_{+l} k_{+x}}{k_{pd} k_{rd}} \cdot \frac{G_{tot,i}}{1 + \sum_j \frac{1}{K_M^{ij}} I_j}.$$

Note that in the last equation we assumed

$$g_i \frac{k_{+l} k_{+x}}{k_{pd} k_{rd}} \approx \frac{1}{G_{tot}},$$

it's more like a choice of simplification and this whole term represents certain "gain of protein R_i from gene G_i copy number."

This is the output of our GRN version LTU. We anticipate this to be capable of performing the "decision" function of an LTU. Recall that we would have a firing output of the neuron if the linear combination of input signals is higher than a threshold, and on the contrary the neuron would be silent if the linear combination of input signals can't achieve the threshold. This gives the graphical intuition of a "decision boundary."

In our molecular design, this exact decision boundary would be implemented by the result of a "cruel competition" between R_i and R_i': since the binding of two regulators is so tight, we can almost say that R_i occurs only if $R_{tot,i} > R_{tot,i'}$, since otherwise it would all show up in the binding form $D_{ii'}$.

In the previous paragraphs we already analyzed the computation essence (G_i and R_i) of our GRN implementation, and we can now connect the whole body of this one LTU, starting from the decision goal $R_{tot,i} > R_{tot,i'}$:

$$\begin{aligned}
R_{tot,i} &> R_{tot,i'} \\
\frac{\alpha_i}{1+\sum_j \frac{1}{K_M^{ij}} I_j} &> \frac{\alpha_{i'}}{1+\sum_j \frac{1}{K_M^{i'j}} I_j} \qquad \left(\alpha_i := g_i \frac{k_{+l}k_{+x}}{k_{pd}k_{rd}} G_{tot,i}\right) \\
\alpha_i\left(1+\sum_j \frac{1}{K_M^{i'j}} I_j\right) &> \alpha_{i'}\left(1+\sum_j \frac{1}{K_M^{ij}} I_j\right) \\
\sum_j \left(\frac{\alpha_i}{K_M^{i'j}} - \frac{\alpha_{i'}}{K_M^{ij}}\right) I_j &> \alpha_{i'} - \alpha_i. \qquad (8.2)
\end{aligned}$$

This has the LTU form "weighted sum > threshold."

By comparing the last equation with the LTU definition, we can see that we successfully built a LTU element with 3 genes. So theoretically, if we want, we can now build up an ANN in biology.

8.2.2 Practical Considerations

The computational complexity (or specifically, the functional expression capability) of ANN scales with the number of weights, or the number of connections between neuron units. For one GRN-version LTU that we just created, this number of connection is roughly the number of transcription regulatory binding to the genes (like G_i and G_i'). So the number of connection in this network would be the typical number of gene transcription binding times the total number of genes involved.

Biological knowledge would give us a rough taste of this number of connections:

	E. coli	Yeast	Human
# TFs	300 (7%)	200 (3%)	1800
# TFs on a gene (ChIP-seq etc.)	$1 \sim 2$ (0 to 10)	5–12 (0 to 20)	10–12 (0 to 100)
# weights	10^2	10^3	10^4

The number of connections of network is at most the extend of 10^4. We can also take a look at the history of ANN, the success of deep learning and machine learning areas enables amazing ai products and changed the human society. How many connection do we need there?

Era	Representative models/tasks	# weights (order)
1990s	LeNet (LeCun); MLP on MNIST; etc.	$\sim 10^5$
2010s	Deep nets: AlexNet (2012), VGG-16 (2014), ResNet (2015) on ImageNet	10^7–10^8
2020s	Transformers: GPT-1, GPT-2, GPT-3, GPT-4	$10^8,\ 10^9,\ 10^{11},\ 10^{12}$

So with our reasoning and estimation, even human cells achieves 1/10 of LeNet, which is just for simple tasks like hand-written digit recognitions……This is somewhat unsatisfactory and disappointed: we believe that cells are definitely more powerful and efficient, and we are eager to achieve better computation with biology-implemented circuits. But I think it's no suprise that we face this challenge. We just made a hard, superficial analogy to make a molecular network exactly the same as ANN to implement the computation, and this is fairly impossible that cell happens to choose this very strategy. The success of human world ANN, what problems we meet, what we can do, how ANN address computation tasks… would be quite different with the situations for a cell, a biological entity.

8.3 Side Note: Binding is much more efficient!

When we attempt to build bio-computation machines, binding reaction network may be more powerful than the gene regulation network(GRN).

We first show that a simple tight binding would implement a ReLU function. If two species A and B bind very tightly to form complex C, then in the tight-binding limit the free amount of A is approximately

$$A \approx \max(0, A_{\mathrm{tot}} - B_{\mathrm{tot}}) = \mathrm{ReLU}(A_{\mathrm{tot}} - B_{\mathrm{tot}}).$$

The number of binding reactions in cells would be much larger:

Type of binding reaction	*E. coli*	Yeast	Mammalian
protein–protein	10^4–10^5	10^5–10^6	10^6–10^7
enzyme–substrate / cofactor	10^5	—	—
protein–DNA	—	—	—
protein–RNA	—	10^4–10^5	—
protein–lipid / ion …	—	—	—
Total	10^5–10^6	10^6–10^7	10^7–10^8

But it still don't scales to contribute to the computation complexity. What's more, not all bindings are tight. And we should not assume all of these binding reactions serve for just "computation." After all, cells need to do a whole bunch of other tasks.

8.4 Computational Tasks in Biology vs Engineered Systems

8.4.1 Logic Gates, LTUs, and Cellular Computation

In engineered systems such as digital computers, the fundamental computational elements are **logic gates** (e.g. AND, OR, NOT, NAND), which implement Boolean operations and logical deduction. In deep artificial neural networks (ANNs), the basic units are linear threshold units (LTUs); large networks of LTUs can approximate arbitrary complex input–output maps.

A natural question is: **what are the analogous computational tasks in biological systems?** Do cells need to perform logical deduction, or fit arbitrary input–output functions the way engineered ANNs do?

One ubiquitous task is the following: in response to different combinations of environmental signals (nutrients, antibiotics, hormones, morphogens, and so on), a cell switches into different internal states, often realized as distinct gene-expression programs. Thus the cell must implement at least one nontrivial map.

$$\text{environmental inputs} \longrightarrow \text{cellular states / behaviors}.$$

A convenient abstraction is shown in Fig8.7.

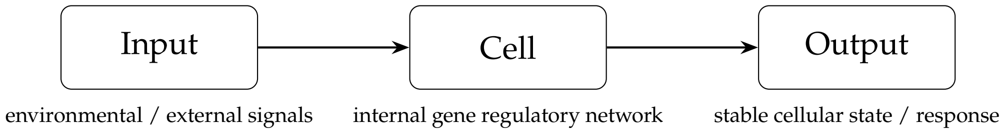

Input
Cell
Output
environmental / external signals
internal gene regulatory network
stable cellular state / response

Figure 8.7 Abstract view of cellular computation: environmental inputs are processed by the cell into distinct output states.

8.4.2 Real Example: Multi-fate Differentiation

A concrete biological example is multi-fate differentiation (e.g. Zhu Ronghui, Elowitz lab, *Science* 2022). The computational task is to generate on the order of 10^2–10^3 distinct cell types using on the order of 10 transcription factors (TFs), rather than 10^2–10^3 distinct TFs.

This is achieved by *combinational encoding*. For example, if three genes A, B, C can each be high or low, there are $2^3 = 8$ possible expression patterns. More generally, N genes provide 2^N combinatorial patterns.

The implementation is through *multistability* in gene-expression space: each stable cell type corresponds to a distinct stable fixed point (attractor) of the underlying dynamical system. Figure 8.8 shows a simple schematic with three stable fixed points in the plane spanned by gene A and gene B.

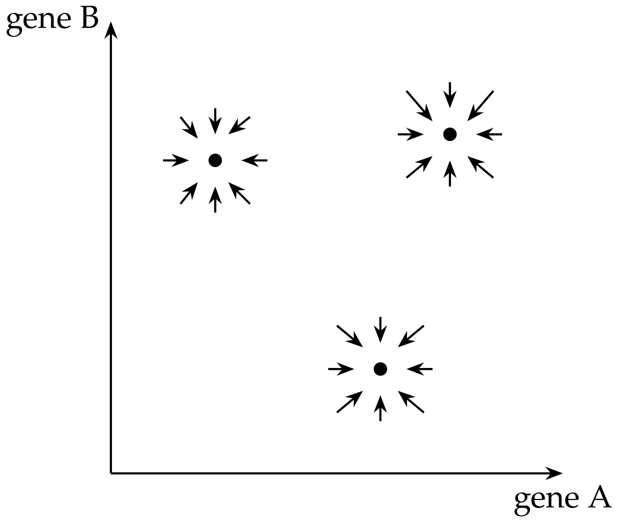


Figure 8.8 Multistability in the gene A–gene B plane. Filled dots are stable fixed points; surrounding arrows indicate flow of cell states toward these attractors.

Another Example of Multistability: Toggle Switch

A canonical synthetic example is the toggle switch with mutual repression (Gardner & Collins 2000). Two genes A and B repress each other's expression via their protein products.

At the gene level, the proteins bind the opposite promoter and inhibit transcription; at the protein level, A and B form a mutually repressive pair. This is summarized schematically in Fig. 8.9.

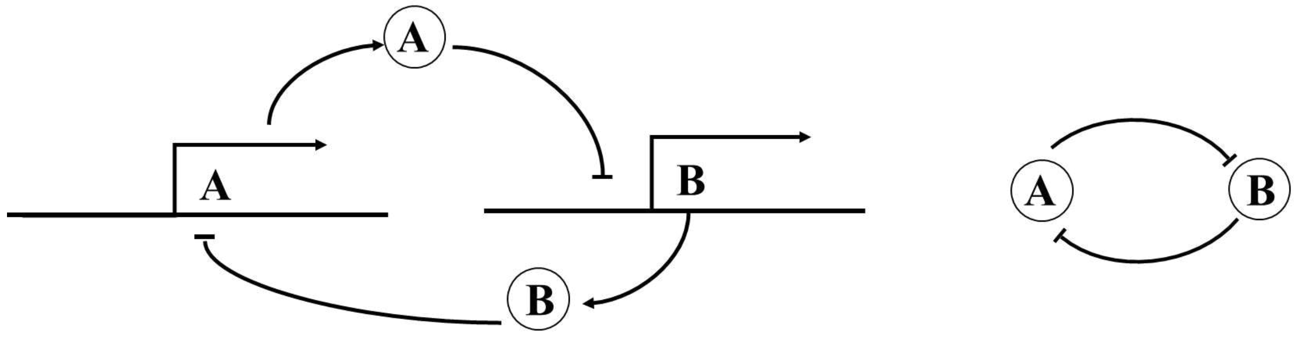


Figure 8.9 Mutual repression between two genes A and B and between their protein products.

Mutual Repression: Dynamics and Nullclines

We write the binding reactions

$$G_A + B \rightleftharpoons G_A^B, \qquad G_B + A \rightleftharpoons G_B^A,$$

and the dynamics for the total protein concentrations A_{tot} and B_{tot} as

$$\frac{dA_{tot}}{dt} = \nu + \beta \frac{G_A}{G_{tot}} - \delta A_{tot}, \tag{8.3}$$

$$\frac{dB_{tot}}{dt} = \nu + \beta \frac{G_B}{G_{tot}} - \delta B_{tot}. \tag{8.4}$$

Assuming $G_{tot} \ll K_A, K_B$,

$$G_A \approx \frac{G_{tot}}{1 + B_{tot}/K_A}, \tag{8.5}$$

$$G_B \approx \frac{G_{tot}}{1 + A_{tot}/K_B}. \tag{8.6}$$

From $dB_{tot}/dt = 0$ we obtain the nullcline

$$B_{tot} = \frac{1}{\delta}\left(\nu + \beta \frac{1}{1 + A_{tot}/K_A}\right).$$

A schematic of the A_{tot} and B_{tot} nullclines is shown in Fig. 8.10; there is only one intersection, so the system is monostable in this simple parameter regime.

To increase nonlinearity, we can let repressors be dimers (or multimers):

$$G_A + 2B \Rightarrow G_A^B, \quad G_A \approx \frac{G_{tot}}{1 + (B_{tot}/K_A)^2}.$$

However, even then we may obtain only two stable states, with a middle unstable fixed point.

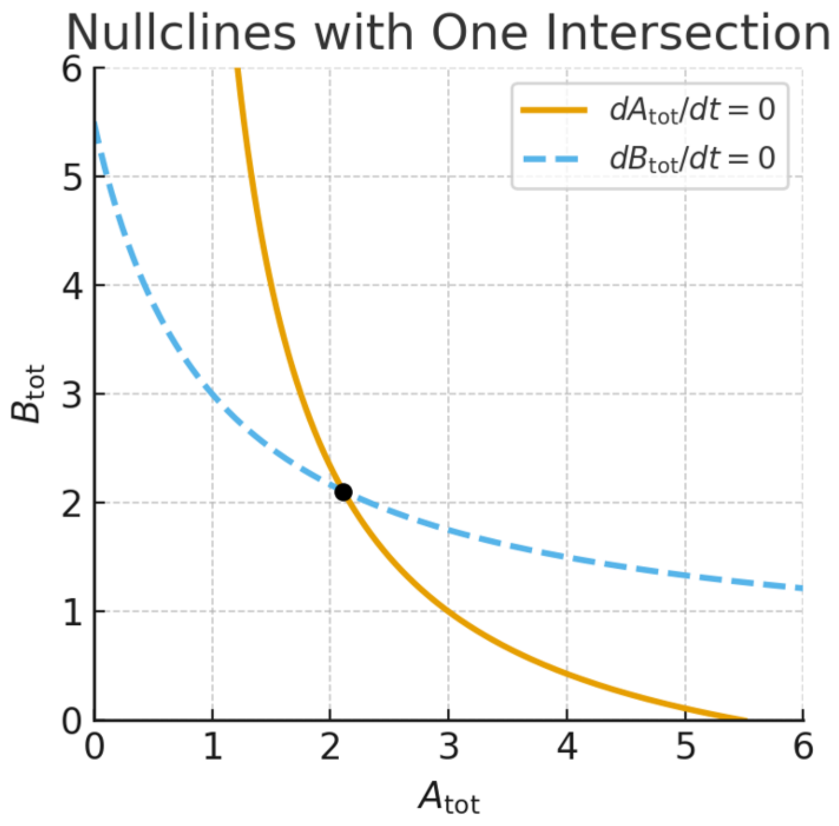


Figure 8.10 Nullclines for the mutual repression toggle in a simple parameter regime with a single intersection. More curvature is needed to obtain multiple fixed points.

8.4.3 Developmental Motifs and SAMI

Motifs from developmental biology often combine *self-activation* with *mutual inhibition* (sometimes referred to as SAMI). It is easy in such motifs to make one gene high and the other low, yielding two alternative stable states.

However, competitive binding for inhibition and higher-order binding such as dimerization can further sharpen response curves and introduce ultrasensitivity. This raises the question: can two genes produce more than two stable states (e.g. three stable states)?

Binding Reactions and Nullcline Analysis

We enrich the model by introducing explicit binding between A and B:

$$G_A + A \rightleftharpoons G_A^*, \tag{8.7}$$
$$G_B + B \rightleftharpoons G_B^*, \tag{8.8}$$
$$A + B \rightleftharpoons C_{AB}. \tag{8.9}$$

The dynamics become

$$\frac{dA_{tot}}{dt} = \alpha + \beta \frac{G_A^*}{G_{tot}} - \delta A_{tot}, \tag{8.10}$$
$$\frac{dB_{tot}}{dt} = \alpha + \beta \frac{G_B^*}{G_{tot}} - \delta B_{tot}. \tag{8.11}$$

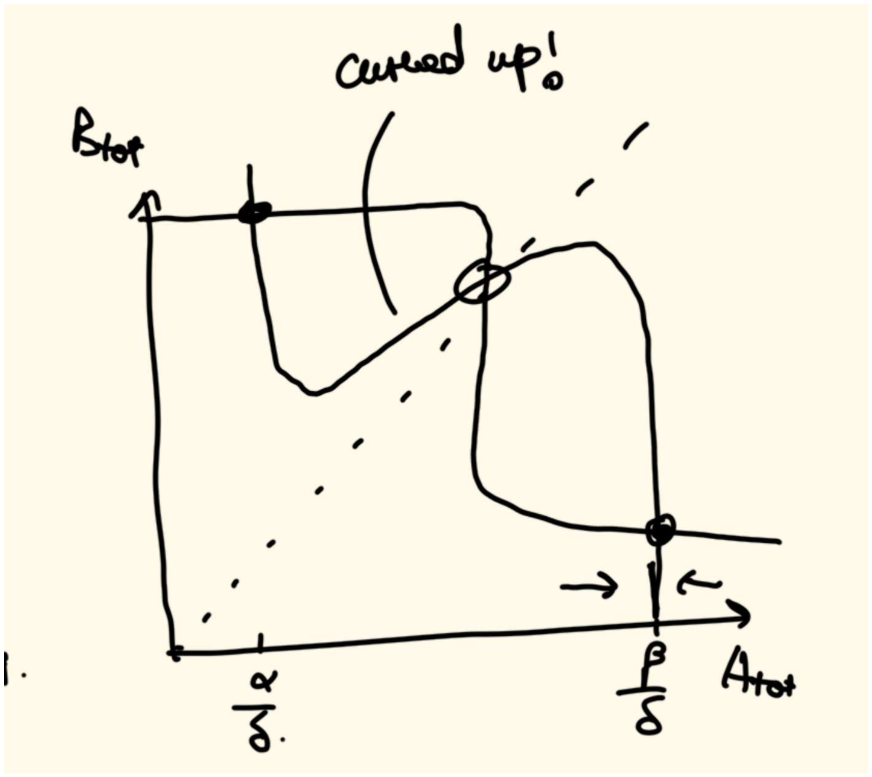


Figure 8.11 With stronger nonlinearity, the nullclines can intersect three times; typically the middle fixed point is unstable and the outer ones are stable, giving bistability.

Using a quasi–steady-state approximation,

$$G_A^* \approx \frac{G_{tot}A}{A+K_A}, \tag{8.12}$$

$$G_B^* \approx \frac{G_{tot}B}{B+K_B}, \tag{8.13}$$

and assuming $C_{AB} \ll K_A, K_B$ so that

$$A_{tot} \approx A + C_{AB}, \qquad B_{tot} \approx B + C_{AB}.$$

For the A_{tot} nullcline,

$$\alpha + \beta\frac{A}{A+K_A} = \delta A_{tot}.$$

We consider different regimes:

1. A_{tot} very low:

$$\alpha \gg \beta\frac{A}{A+K_A} \quad \Rightarrow \quad A_{tot} \approx \frac{\alpha}{\delta}.$$

2. A_{tot} very large:

$$\alpha \ll \beta\frac{A}{A+K_A} \quad \Rightarrow \quad \beta\frac{A}{A+K_A} = \delta A_{tot}.$$

(a) If B_{tot} is small, then $A \approx A_{tot}$ and

$$A_{tot} \sim \frac{\beta}{\delta}.$$

(b) If B_{tot} is comparable with A_{tot}, then

$$A \approx A_{tot} - B_{tot} \ll K_A,$$

giving

$$\frac{\beta}{K_A}(A_{tot} - B_{tot}) = \delta A_{tot},$$

hence

$$B_{tot} = \left(1 - \frac{\delta K_A}{\beta}\right) A_{tot}.$$

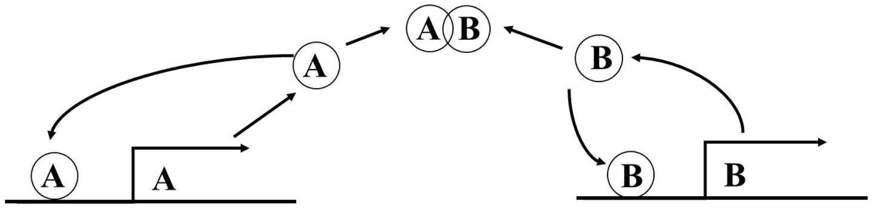


Figure 8.12 Schematic nullcline of A_{tot} in the presence of A–B binding; in the relevant region the slope can be tuned to be less than 1.

Despite the additional binding, the system may still be only bistable unless we introduce stronger ultrasensitivity in the activation step.

Homodimer Activation, Heterodimer Inhibition and Scaling

We now consider homodimer activation together with heterodimer inhibition. Dimers A_2 and B_2 activate their own promoters, whereas the heterodimer C_{AB} is inactive:

$$A + A \rightleftharpoons A_2, \tag{8.14}$$
$$B + B \rightleftharpoons B_2, \tag{8.15}$$
$$A + B \rightleftharpoons C_{AB}, \tag{8.16}$$
$$A_2 + G_A \rightleftharpoons G_A^*, \tag{8.17}$$
$$B_2 + G_B \rightleftharpoons G_B^*. \tag{8.18}$$

$$A_{tot} = A + 2A_2 + C_{AB}, \qquad B_{tot} = B + 2B_2 + C_{AB}.$$

Assuming $G_{tot} \ll K_{GA}, K_{GB}$ and that A_2, B_2 never dominate, one can show that in the regime where B_{tot} is comparable to A_{tot},

$$A \approx A_{tot} - B_{tot}, \qquad A_2 \approx \frac{(A_{tot} - B_{tot})^2}{K_A},$$

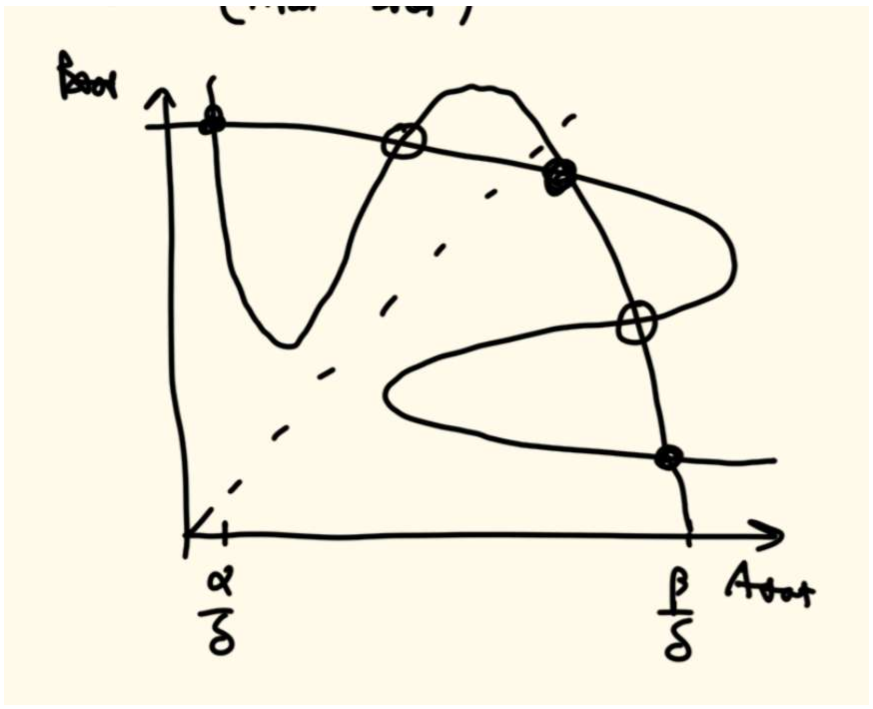


Figure 8.13 A^2-like activation bends the nullcline upward, allowing three intersections and thus three stable states for two genes.

and setting $dA_{tot}/dt = 0$ yields

$$\beta \frac{(A_{tot} - B_{tot})^2}{K_A K_{GA}} = \delta A_{tot}.$$

This corresponds to an effective A^2-type activation that can bend the nullcline upward and produce three intersections (three stable states) for two genes.

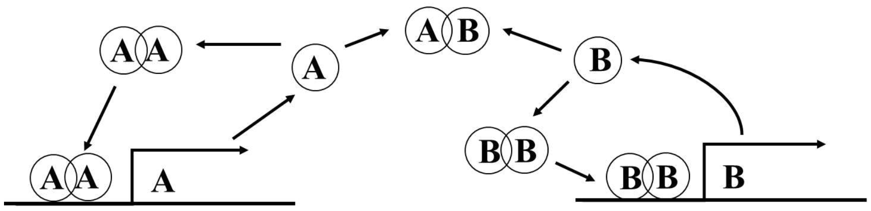


Figure 8.14 Homodimer activation (A_2, B_2) and heterodimer inhibition via C_{AB}.

Ultrasensitivity from binding and dimerization thus adds extra stable states. Roughly, if N transcription factors participate in such combinatorial binding networks, one can achieve on the order of 2^N stable expression patterns. For $N \approx 10$, this yields 10^2 to 10^3 stable cell types—comparable to the diversity of cell types in many multicellular organisms.

Cells therefore perform efficient and complex computation primarily through binding networks: many binding reactions and comparatively few catalytic reactions, a very different "hardware" from digital electronics but capable of similarly rich computational behavior.

Bibliography