CCBS. Lecture 08 Stochastic Kinetics of Markov Chains and Computation. Biomachines.

From last time — Equilibrium physics of bioregulation.

- why equilibrium?

(1) From experiment / measurement perspective:

in dynamics, lots of rate constants. may not be measurable.

Experimentally. equilibrium quantities are easily measured.

e.g. eq. binding constants. $E+S \rightleftarrows C$. $K_d = \frac{ES}{C}$.

ΔG is easier to measure. vs $K_d = e^{-\Delta G/k_BT}$

(2) From bio system design perspective:

equilibrium mechanism have low cost.

(Low cost in energy and mechanism complexity).

e.g. a desired function/process is implementable just by binding versus by catalysis/gene expression.

(3) From system analysis/understanding perspective:

equilibrium steady states are easier to analyze.

- satisfies detailed balance. exist energy function.

no cycle fluxes. etc

Today (Briefly): What about Kinetics? Response speed etc.
Beyond Equilibrium?

(3) Kinetics and Beyond equilibrium: Markov chains for molecular state transitions.

(4) Steady state. distr. and hitting times

Then: Computation Biomachines.

— Biological systems need to process information.
just like computational machines we have engineered.

How do they do complex computation in comparison?

① Logic-gate computers in gene regulation.

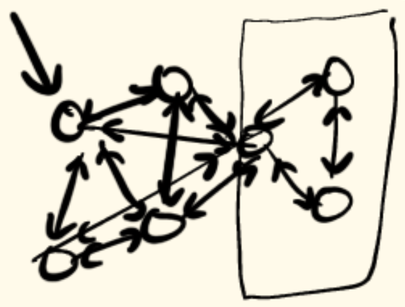
② Neural network computers in gene regulation.

③ Computational task of biology.

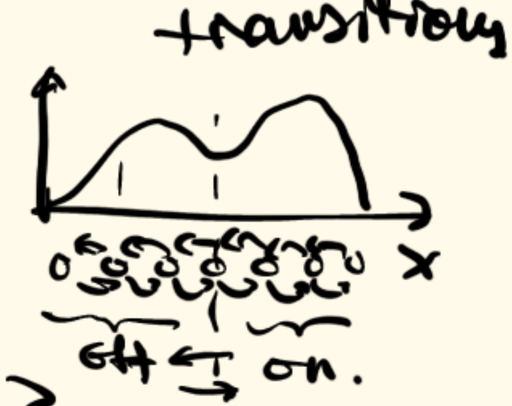
④ Complexity through binding at equilibrium.

③ Kinetics and Non-equilibrium steady states — Markov chains.

- Equilibrium or not. what about stochastic kinetics?



e.g. in response to environment.
cells or molecules "turn on" after state transition



e.g. experimental obs. of population.

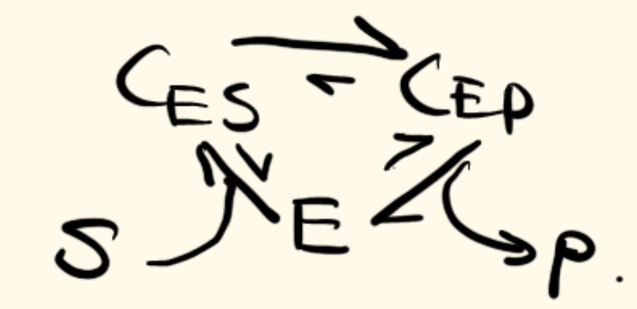
- Also, what if I encounter a behavior that is for sure not at equilibrium?

presence of strong driving force s.t. behavior can't be explained by equilibrium.
— my behavior of interest can only be achieved out of equilibrium...

— e.g. Kinetic proofreading ... (super precise)

e.g. enzymatic reaction

$$S \xrightarrow{E} P$$



e.g. phosphorylation cascade.

$$A \rightleftharpoons A_p$$

How to analyze that?

- For the special case of finite # 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.)

Also a Markov chain. but often infinite state.

- Continuous-time Markov Chain
  (from CME of chemical reactions).

  - $A \underset{k_2}{\overset{k_1}{\rightleftharpoons}} B$ chemical reactions.

    Two interpretations:

    Concentration change ↔ single molecule state transition.

    Rate equation: $\frac{dA}{dt} = k_2 B - k_1 A$

    Define. $p_A = \frac{A}{A_{tot}}$, $p_B = \frac{B}{A_{tot}}$. $A_{tot} = A + B$.

    ⇒ $\frac{d\,p_A}{dt} = k_2 p_B - k_1 p_A$. State transition!

    Same for CME. (check.)

  - This holds because linearity / first order is special.
    It's the same as scaling with size / number.
    So. 1st order = per molecule

  - All CRNs with 1st and 0th order reactions can be interpreted as state transitions.

    0th order? $\emptyset \to A$ $A \to \emptyset$.
    Can be viewed as $B_0 \to A$, $A \to B_0$
    and B_0 is held constant by external means.
    This also happens when studying certain dynamics of Markov chains.

  - Master equation for continuous-time Markov Chain.

    p. — vector of state probabilities.

    $q_{ij} > 0$ — rate of transition $j \to i$

    $q_{ii} < 0$ — rate of leaving state i.

$$\frac{dp_i}{dt} = \sum_{j \neq i} q_{ij} p_j - q_{ii} p_i$$

$$\frac{dp}{dt} = Q p. \qquad Q - \text{transition rate matrix.}$$

- Conservation of prob.

$1^T p = 1$ always.

So. $\frac{d\, 1^T p}{dt} = 1^T Q p = 0 \quad \forall p.$

So $\underline{1^T Q = 0}$. $\left(\text{So. } q_{ii} = -\sum_{j \neq i} q_{ji} \right)$.

- Steady state distribution.

$\underline{\pi. \text{ s.t. } Q\pi = 0}$

(Solve : Take. Q. ask for "Null space".)

Irreducible $\Leftrightarrow$ Ergodic $\Leftrightarrow$ steady state is unique

State transition graph is strongly connected.

Prob starting in i, reach j. at time t. is positive. $\forall i, j$

Q's nullspace has dimension 1.
All eigenvalues has real part < 0.
except one. has 0.

- Equilibrium? π is an eq. steady state. then

Detailed balance: $q_{ij} \pi_j = q_{ji} \pi_i$.

This can only happen if the <u>rates</u> satisfy conditions.

(cycle flux = 0.).

Interpretation of "Markovian"

Q: But not all systems are memoryless/Markovian...

- The idea of state
  - Everything is a state space system. / Markovian.
  - Markov chains are just stochastic state space systems.

State: all info about the system you need to know to describe its internal dynamics evolving into the future.

i.e. if isolated. its future only depends on its state. not on its past.

- $\dot{x} = f(x)$. state-space Deterministic system
- $\frac{d}{dt} p(x) = Q\, p(x)$. Stochastic state-space system. !
- All systems can be written as state space systems.

  i.e. All systems can be written as Markovian

  (all systems can be made Memoryless)

  ↳ if the state is large/representative enough!

e.g. $X(t)$ depends on its past 5 time points.

Then make $\hat{X}$ s.t. $X_1 = X(t)$ $X_2 = X(t-1)$...
$X_5 = X(t-5)$.

e.g. Newtonian mechanics. position x's future depends on velocity (past). $\Rightarrow$ $\hat{X}_1 = x$. $\hat{X}_2 = \dot{x}$.

- Kinetics from a Markov chain.

— How long does it take to transition to an active state?

e.g. gene expression. $G_{off} \rightleftharpoons G_{exposed} \rightleftharpoons G_{active}$.

enzymatic reactions.

— This is captured by "Hitting times" or "First Passage Times"

τ_{ji} = expected time it takes to hit i when starting at j

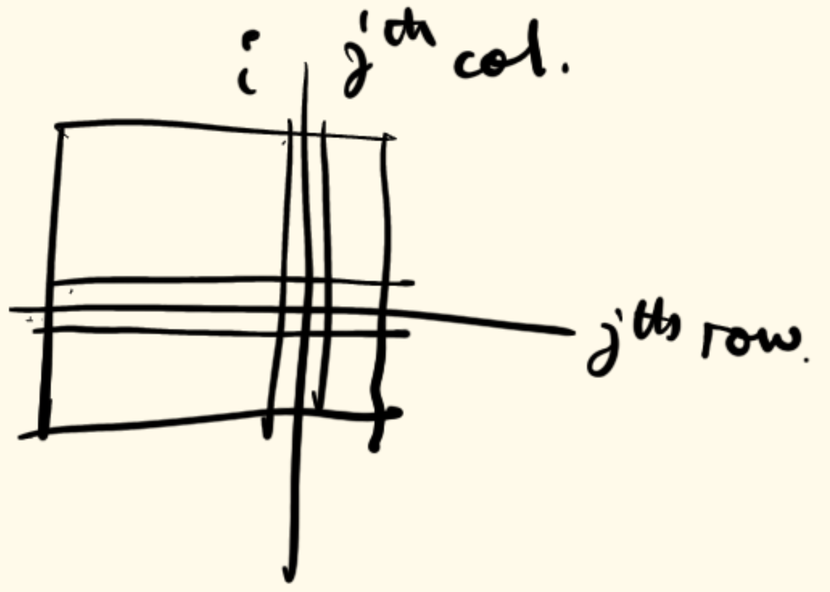
— e.g. $ON \underset{k_+}{\overset{k_-}{\rightleftharpoons}} off$ $\quad \tau_{on \leftarrow off} = \frac{1}{k_+}$.

— In general.

$$\tau_{ji} = \underbrace{\frac{1}{|q_{ii}|}}_{\text{wait}} + \frac{q_{ji}}{|q_{ii}|} \cdot \underbrace{0}_{\tau_{ji}} + \sum_{i' \neq i,j} \frac{q_{i'i}}{|q_{ii}|} \tau_{ji'}$$

$$- q_{ii} \tau_{ji} = 1 + \sum_{i' \neq i,j} q_{i'i} \tau_{ji'}$$

$$\Rightarrow \sum_{i' \neq j} q_{i'i} \tau_{ji'} = -1$$



Hitting time vector $\tau^{(j)}$ — time to hit state j.

$$Q^{(j)} \tau^{(j)} = -1$$

$Q^{(j)}$ is Q after deleting the j^{th} row and col.

- Other topics of stochastic kinetics.
  - Entropy production in Nonequilibrium steady states (NESS)
  - Markov chains for statistical inference (Markov chain Monte Carlo. MCMC. etc.).
  - Finite state projection. (Solving CME by Markov chain methods).

Bioreactive II: Computation Bioreactive.

- Again. like in adaptation bioreactive.
  - 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 <u>same tasks</u>. the structures and design principles of engineered machines could be used to better understand why biological systems are built in this way.
  - Same function (hopefully).
    Different implementations.

- Today. Computation bioreactive.
  - Biological systems need to <u>process information</u> to respond to environments.

e.g. 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.

– How is it done? How complex a computation the cell achieve in this way?

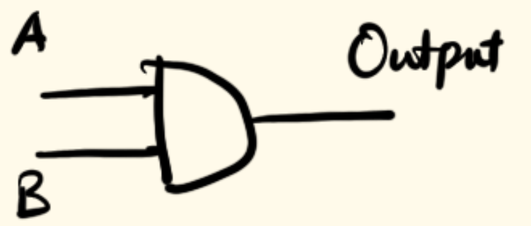
In engineered machines. we build computational units:
then link them up for larger and larger computations.

⇒ In search of the biological computational unit

(1st) Logic gates

– We built computers consisting of binary logic gates to perform computation.
Maybe cells do the same?

— e.g. AND gate.



Truth table.

A	B	output
0	0	0
0	1	0
1	0	0
1	1	1

OR gate 

NOT gate 

– Cells have chemical reactions inside. (catalysis).
maybe implement gates via chemical reactions?

Input / output become species' concentrations.

0 — low conc.
1 — high conc.
relative to a "total" conc.

e.g. a variable y is represented in two forms. Y_0, Y_1

$Y_0 \rightleftharpoons Y_1$ (e.g. like phosphorylated vs not.)

Then "transmission" of a signal can be done

by catalysis of production / degradation.

e.g. $X_1 + Y_0 \longrightarrow X_1 + Y_1$.

and. $Y_1 \longrightarrow Y_0$.

$\Rightarrow \dot{Y}_1 = x_1 Y_0 - Y_1 \Rightarrow \frac{Y_1}{Y_0} = x_1$.

So. high $X_1 \Rightarrow$ high Y_1

$x = 1 \Rightarrow y = 1$.

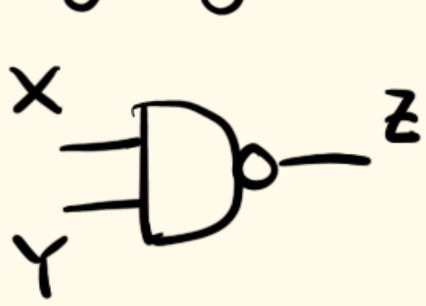
— How to build a <u>universal Turing machine</u> out of logic gates?

(i.e. You can do any computation.)

- You just need <u>NAND</u> gate.

since all logic gates can be built from it.

— NAND gate.



X	y	Z
0	0	1
0	1	1
1	0	1
1	1	0

$X_0 + Y_0 + Z_0 \rightarrow X_0 + Y_0 + Z_1$

$X_0 + Y_1 + Z_0 \rightarrow X_0 + Y_1 + Z_1$

$X_1 + Y_0 + Z_0 \rightarrow X_1 + Y_0 + Z_1$

$X_1 + Y_1 + Z_1 \rightarrow X_1 + Y_1 + Z_0$.

(catalysts)

So. <u>Chemical reaction networks are Turing universal!</u>

— Cells can perform logic gate computations via catalysis chemical reactions!

In fact. <u>arbitrarily complex</u> computations!

- Points to worry:
  - signal may get corrupted in cascade
    (repeat / restore)
  - such reactions maybe hard to implement
    (e.g. not tri molecular. make it 1st / 2nd order.).
  - reactions are stochastic / discrete. / noise
    (in fact that's even better...)
  - maybe (catalysis) reactions are more powerful.
    (use mass-action catalysis reaction themselves as computational units. similar. a bit better)

WAIT. - But. How complex can a cell do in this way?

# of catalysis reactions ≈ # types of enzymes.

This is 10^3. (5×10^3 in database for all known
10^3 in human cell...)

2-8 reactions / gate. ⇒ $\approx 3 \times 10^2$ logic gates.

Compare: 1972. first pocket calculators. 3×10^3 transistors.
2-8 transistors / gate. (e.g. 4 for NAND).
⇒ 4 bit data bandwidth. (4 bit / one decimal at a time.

Moon landing. 1970s - 5×10^3 gates. (20× cell).

Now a days. microcontroller chips. → 10^6 gates.
Intel Laptop chips. typically 10^9 gates (10^3 × cell).

- Can't do much Complexity! (10^{10} transistors).

Maybe cells do computations in smarter ways?

↳ More efficient computing than logic gates?

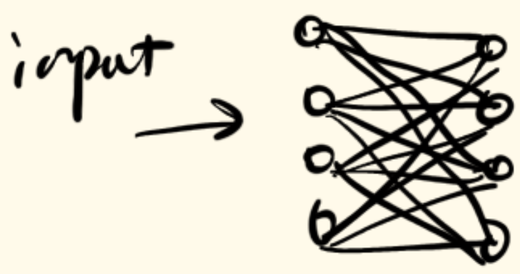
(2') Neural networks.

- Recent years (2010 onwards). Artificial neural networks (ANN) have taken over in complexity of certain computational tasks, e.g. computer vision (image recognition), language models/translation ... 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

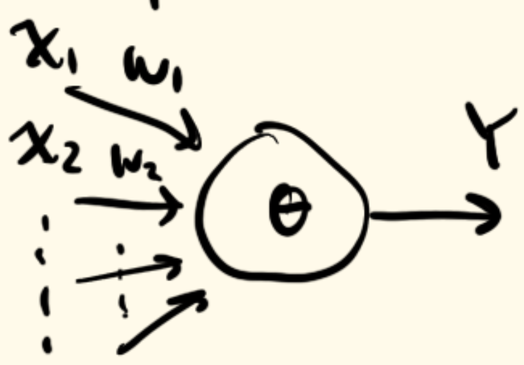


Idea: use layers to build a very complex input-output map that represents the solution of a computational task.

then learn/represent this map by ANN.

(perceptron).

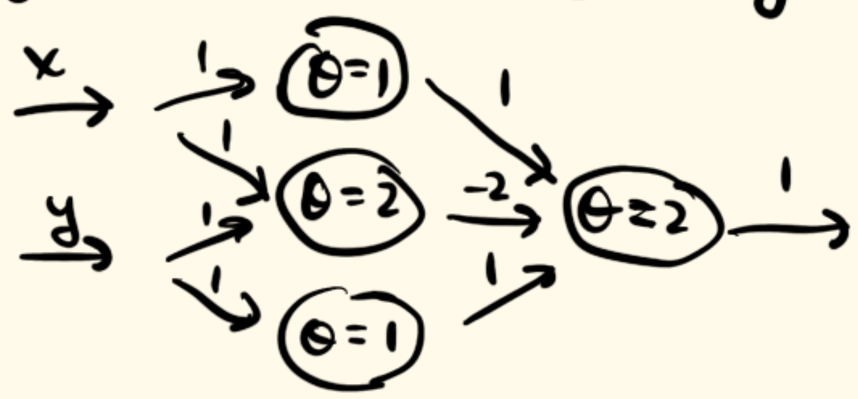
- Computational units of ANN are linear threshold units (LTU).



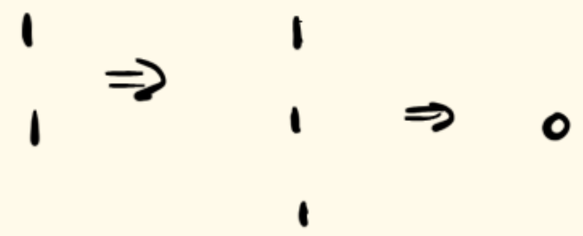
$$w^T x = \sum_i w_i x_i > \theta : \quad y = 1$$

$$w^T x = \sum_i w_i x_i < \theta : \quad y = 0.$$

e.g. $Z = XOR(x,y)$ by



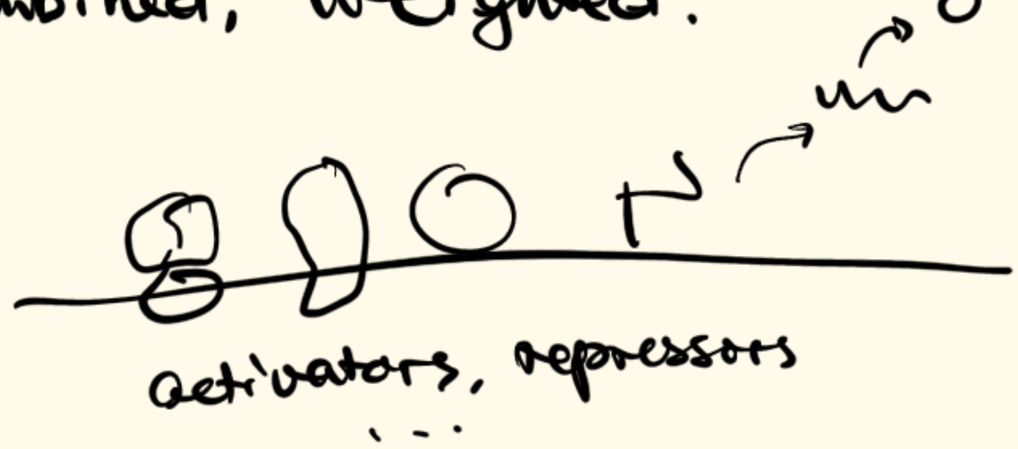
x	y	z
0	0	0
1	0	1
0	1	1
1	1	0



- How to implement a LTU in cells?

  Need lots of inputs combined, weighted.

  ⇒. Gene regulation?



- <u>Gene regulatory network (GRN)</u>.

$G_i \xrightarrow{k^i_{tx}} G_i + T_i$ Transcription. $k^i_{tx} \simeq g_i : k^0_{tx}$

T_i: transcripts; g_i: factor for each gene; k^0_{tx}: same base rate

$T_i \xrightarrow{k_{tl}} T_i + R_i$ Translation

$T_i \xrightarrow{k_{rd}} \phi$ R_i: regulatory species. (e.g. transcription factors).

$R_i \xrightarrow{k_{pd}} \phi$

Typically. $k_{tx}, k_{tl} \sim 10^{-3}/s$ per molecule.

$k_{rd}, k_{pd} \sim 10^{-3}/s$ (e.g. 20 min / hour timescale).

$R_j + G_i \underset{k^-_{ji}}{\overset{k^+_{ji}}{\rightleftharpoons}} G_{ji}$ Regulation of transcription

Typically. $k^+_{ji} \sim 1/nM \cdot s$.

$k^-_{ji} = K_M k^+_{ji}$.

K_M: binding constant. 0.1 to 10^5 nM.

$$R_i + R_j \underset{k^-_{ij}}{\overset{k^+_{ij}}{\rightleftharpoons}} D_{ij}$$

Regulation among transcription factors.

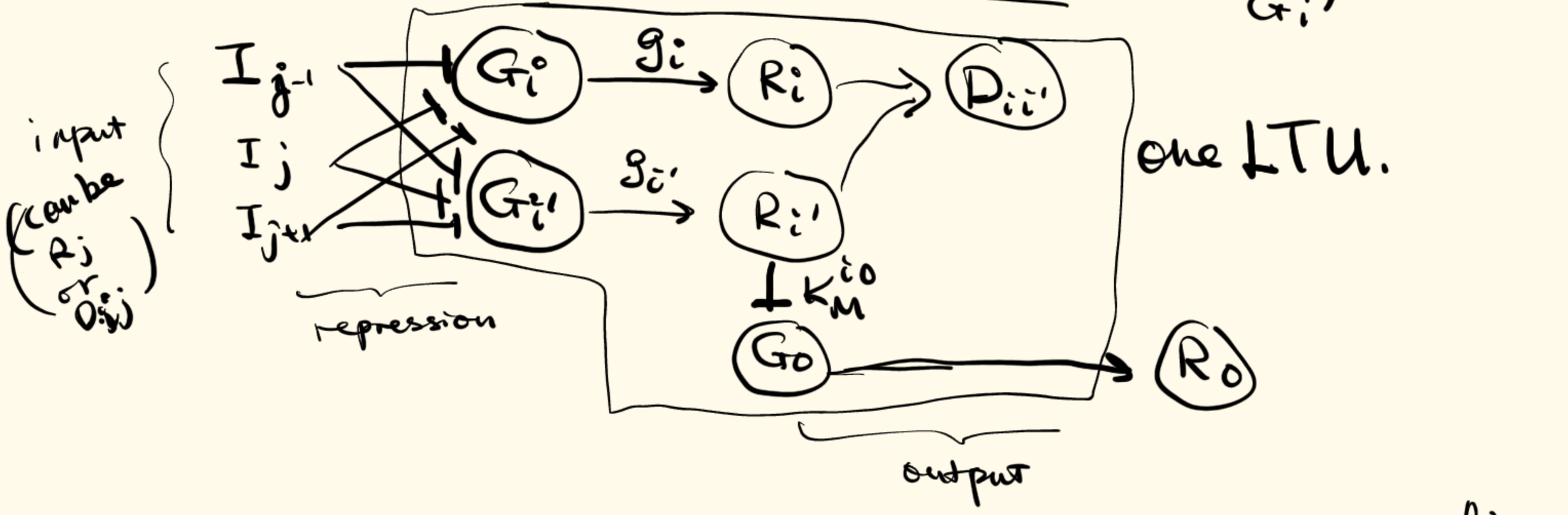
$$D_{ij} \xrightarrow{k_{pd}} \emptyset$$

$$D_{ij} + G_k \underset{k^-_{ijk}}{\overset{k^+_{ijk}}{\rightleftharpoons}} G_{ijk}$$

$$G_{j:} \xrightarrow{k^{j:}_{tx}} G_{j:} + T_i$$

$$G_{ijk} \xrightarrow{k^{ijk}_{tx}} G_{ijk} + T_k$$

Result of regulation (rates could change)



– LTU in Titration GRN ← could be any R_j...

Represent inputs by $x_j = \frac{I_j}{n_{ij}}$ reference value.

Assume I_j are repressors, so $K^{ji'}_{tx} = 0$.

Assume tight binding between $R_i, R_{i'}$. so $K_M^{D_{ii'}} \to 0$.

Assume dimer $D_{ii'}$ don't bind G_i or $G_{i'}$. So $G_{ii'i}, G_{ii'i'}$ ignored

Idea: Competitive binding among transcription factors. $R_i, R_{i'}$ to form a decision

- Binding equilibrium with inputs $G_{ji} = \frac{I_j G_i}{K_M^{ji}}$

$$G_{tot} = G_i + \sum_j G_{ji} = G_i + \sum_j \frac{I_j G_i}{K_M^{ji}} = G_i \left(1 + \sum_j \frac{I_j}{K_M^{ji}}\right)$$

$$\Rightarrow \quad G_i = \frac{G_{tot}}{1 + \sum_j \frac{I_j}{K_M^{ji}}}$$

- Gene expression of R_i,

$$R_{i,tot} = R_i + D_{ii'}$$

$$\dot{R}_{i,tot} = k_{+l} T_i - k_{rd} R_{i,tot}$$

$$\Rightarrow \quad R_{i,tot} = \frac{k_{+l}}{k_{rd}} T_i$$

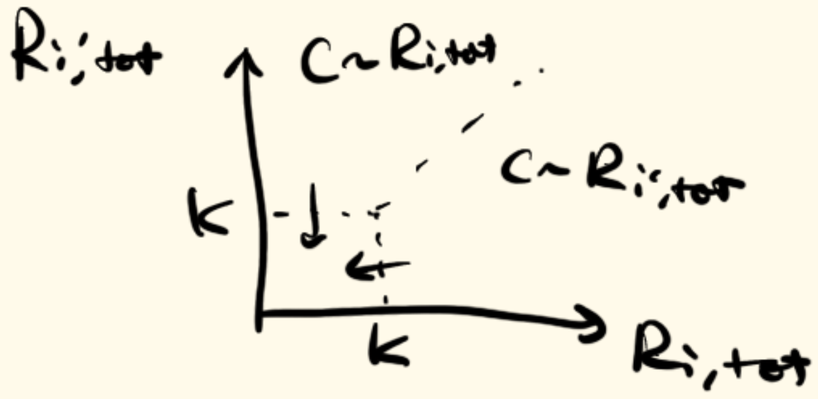
$$\dot{T}_i = g_i k_{tx} G_i - k_{pd} T_i \quad \Rightarrow \quad T_i = \frac{k_{tx}}{k_{pd}} g_i G_i$$

$$\Rightarrow \quad R_{i,tot} = \underbrace{\frac{k_{tx} k_{+l}}{k_{pd} k_{rd}}}_{\approx 1 \text{ per one copy of } G_{tot}} g_i G_i = g_i \frac{G_i}{G_{tot}} = \frac{g_i}{1 + \sum_j \frac{I_j}{K_M^{ji}}}$$

- Competitive binding / titration between $R_i, R_{i'}$.

Due to tight binding. $R_i = \begin{cases} 0 & \text{if } R_{i,tot} < R_{i',tot} \\ R_{i,tot} - R_{i',tot} & \text{otherwise,} \end{cases}$

$(D_{ii'} = \min\{R_{i,tot}, R_{i',tot}\})$



- Decision boundary.
  - There are $R_{i'} > 0$ to bind with G_0, output gene.
    if $R_{i',tot} > R_{i,tot}$. repressing G_0. (also binds tightly).

i.e. $$\frac{g_{i'}}{1+\sum_j \frac{I_j}{K_m^{ji'}}} > \frac{g_i}{1+\sum_j \frac{I_j}{K_m^{ji}}}$$ when output = 0.

$$\Leftrightarrow \quad g_{i'}\left(1+\sum_j \frac{I_j}{K_m^{ji}}\right) > g_i\left(1+\sum_j \frac{I_j}{K_m^{ji'}}\right)$$

$$\Leftrightarrow \quad (g_{i'} - g_i) > \sum_j I_j \left(\frac{g_i}{K_m^{ji'}} - \frac{g_{i'}}{K_m^{ji}}\right)$$

$x_j = \frac{I_j}{m}$

$$\Leftrightarrow \quad \underbrace{\frac{g_{i'} - g_i}{m}}_{\theta_i} > \sum_j x_j \underbrace{\left(\frac{g_i}{K_m^{ji'}} - \frac{g_{i'}}{K_m^{ji}}\right)}_{w_j}$$

This is the case for output = 0.

Recall. LTU. $H_i(x) = 0$ if $\sum_j w_{ji} x_j < \theta_i$

- That's Great! We can build a LTU out of just three genes!

Cells can achieve much more efficient computation using neural networks!

But WAIT--- Complexity of ANN scales with # weights.

connections in GRN ~ # transcription binding to a gene × # genes. (TF also).

		E.coli	Yeast	Human
	# TFs	300 (7%)	200 (3%)	1800
Chip-Seq. etc.	# TFs on a gene	1 ~ 2. (0 to 10)	5 - 12. (0 to 20)	10 - 12 (0 to 100)
	# weights.	10^2	10^3	10^4

weights for ANNs?

1990s. → 2010s → 2020s → time

1990s.	2010s	2020s
LeNet (LeCun)	Deep Nets.	Transformers.
MLP.	AlexNet (2012)	GPT-1. 10^8
on MNIST. etc.	VGG-16 (2014)	GPT-2 10^9
	ResNet (2015)	GPT-3 10^{11}
	on ImageNet	GPT-4 10^{12}
# weights 10^5	$10^7 - 10^8$	

So. even as complex as human cells ... only $\frac{1}{10}$ of LeNet doing hand-written digit recognition? (0 to 9)

Something is wrong... Cells must be doing information processing much more efficiently...

Side Note: Binding is much more efficient.! (Than catalysis or gene expression)

$$ReLU(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{else.} \end{cases}$$

We can implement $f(x,y) = ReLU(x-y)$

easily by tight binding. $X + Y \rightleftarrows D$.

then $X_{free} = \begin{cases} X_{tot} - Y_{tot} & \text{if } X_{tot} > Y_{tot} \\ 0 & \text{else.} \end{cases}$

So. $X_{free} = f(X_{tot}, Y_{tot})$.

- Then. Composition of such f can achieve arbitrarily complex behavior.

e.g. any piecewise-linear function.

any logic gates...

any LTUs...

- # binding reactions in cells?

	E. coli	Yeast	Mammalian
protein-protein	10^4 to 10^5	$10^5 - 10^6$	10^6 to 10^7
enzyme-substrate/cofactor			
protein – DNA		$10^4 - 10^5$	
protein – RNA			
protein – lipid/ion..			
Total.	10^5 to 10^6	10^6 to 10^7	10^7 to 10^8

But still unsatisfactory...

Not all binding are tight...

Most bindings are not for computation, but for production....

getting to 10^9. Complexity of Sophisticated Computing.

Maybe. we got the "function" wrong?

(3') Computational tasks in Bio. vs Engineered.

- Logic gates in computers.

  Task: Logical deductions.

  Do cells need to do logical deductions?

- LTUs in ANNs.

  Task: represent/fit/learn arbitrary complex input/output maps.

  Do cells need to do this?

- One task that we see cells do for sure.

In response to changing environments, perform different actions, often in the form of (distinct) gene expression.

So. at least. cell needs to

encode one complex input-output map.

(not arbitrary) (roughly binary?)

e.g. antibiotics / nutrients / morphogens ...

⇒ cell state transitions.

- This can be done efficiently. via binding networks

  such that N genes → $O(2^N)$ states.

Next: The power of holistic, combinatorial, multi-species binding