CCBS. Lecture 10. Computation Biomachine

From last time :- Biomachine perspective.

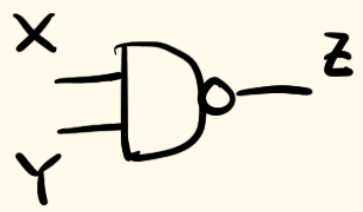
- Computation / information processing

① - Logic gate computation in cells.

- Encode 0/1 as low/high concentration. X_0, X_1

- Implementation via catalysts reactions.

- NAND gate.



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

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

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

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

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

- NAND gate is enough to build <u>universal Turing machines</u>.

(Intro: the concept of <u>universality</u> in Computation)

- But numbers don't make sense!

A cell has 10^3 types of enzymes

$\Rightarrow$ 3×10^2 logic gates.

Compare to 3×10^3 transistors in 1972 pocket calculator

5×10^3 gates for Moonlanding. 1970s.

Today. microcontroller. 10^6 gates.

intel laptop chips. 10^9 gates.

NOT ENOUGH COMPLEXITY!

Maybe cells do computations in smarter ways?

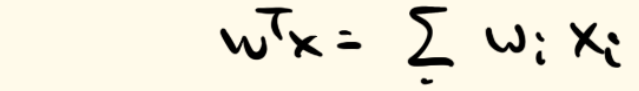
↳ More efficient computing than logic gates?

(2) Neural networks.

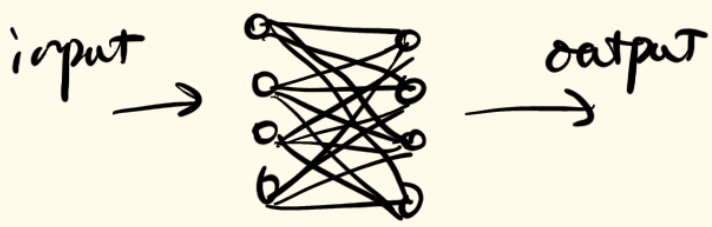
- Recent years (2010 onwards). Artificial neural networks (ANN) have taken over in complexity of certain computational tasks, e.g. computer vision (image recognition), language models/translation ..., robotics ...

- 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

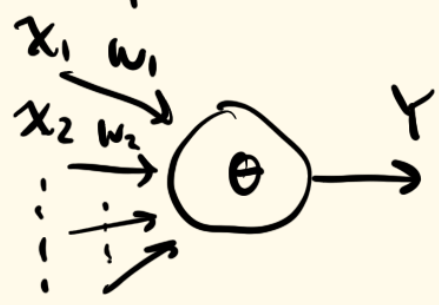


Idea: use layers to build a very complex input-output map that represents the solution of a computational task.

then learn/represent this map by ANN.

(perceptron).

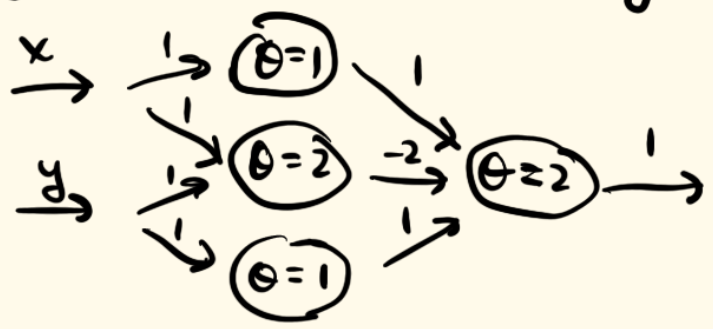
- Computational units of ANN are linear threshold units (LTU).



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

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

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

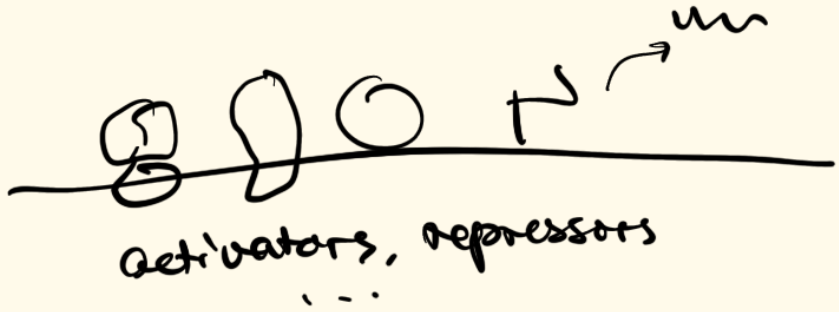


x	y	z
0	0	0
1	0	1
0	1	1
1	1	0

- How to implement a LTU in cells?

  Need lots of inputs combined, weighted.

  ⇒. Gene regulation?



- Gene regulatory network (GRN).

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

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

T_i: transcripts

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

$T_i \xrightarrow{k_{rd}} \phi$

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

R_i: regulatory species. (e.g. transcription factors).

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

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

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

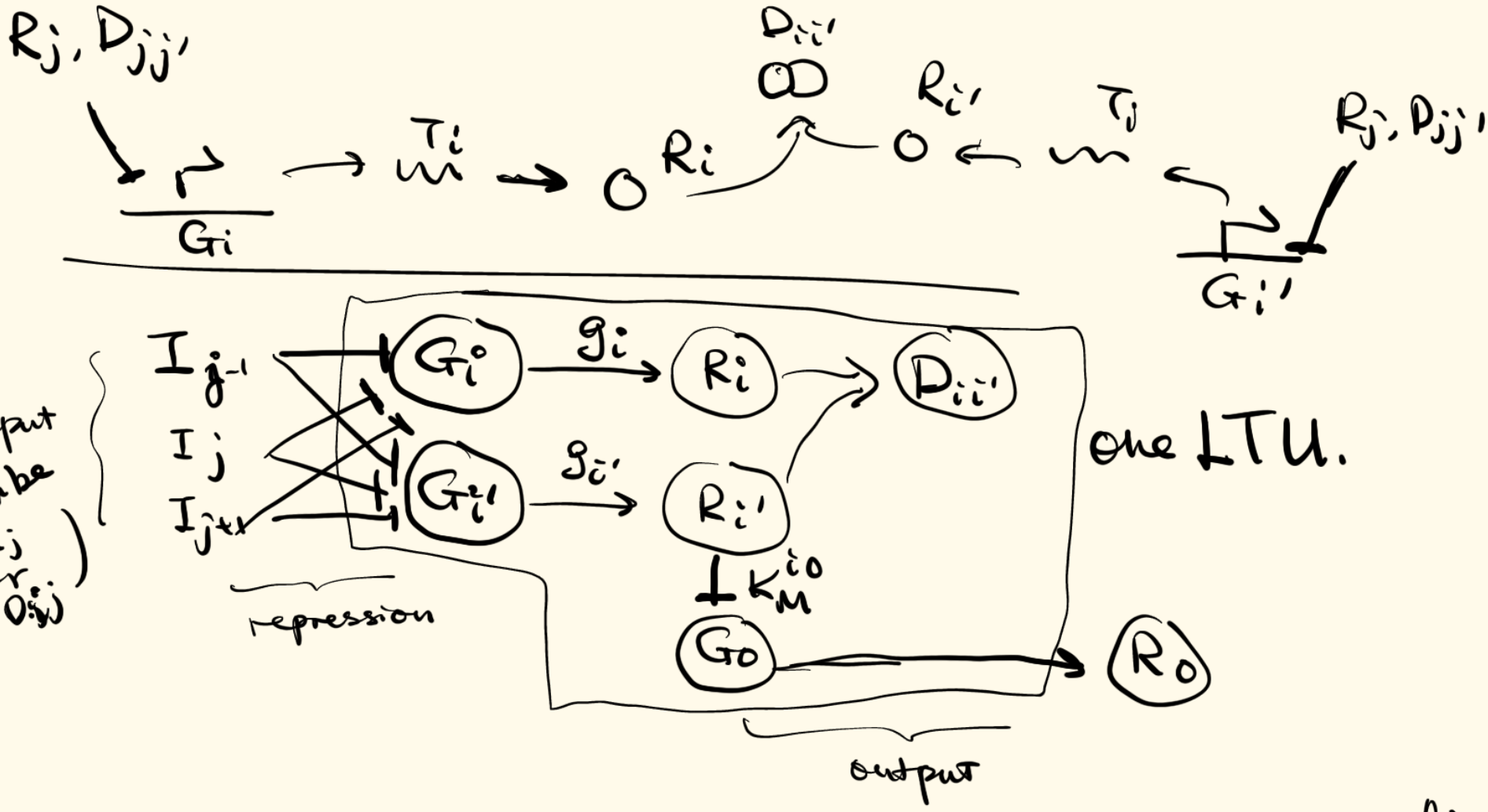
$$D_{ij} \xrightarrow{k_{pd}} \emptyset$$

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

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

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

Result of regulation (rates could change)



- LTU in Titration GRN

Represent inputs by $x_j = \frac{I_j}{m_{ij}}$ ← could be any R_j... ← 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,

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

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

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

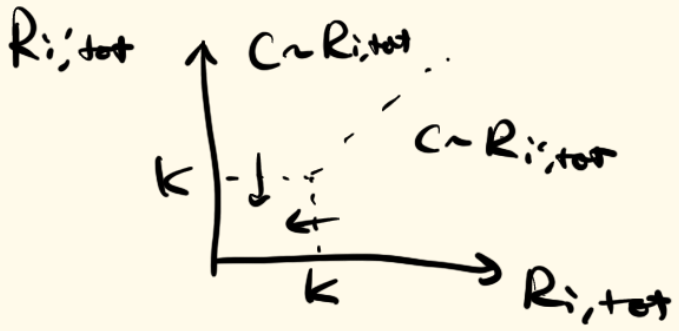
$$T_i = g_i k_{tx} G_i - k_{pd} T_i \quad \Rightarrow \quad T_i = \frac{k_{tx}}{k_{pd}} g_i G_i$$

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

- 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 → like Transformers.
	LeNet (LeCun) MLP. on MNIST. etc.	Deep Nets. AlexNet (2012) VGG-16 (2014) ResNet (2015) on ImageNet	GPT-1. 10^8 GPT-2 10^9 GPT-3 10^{11} 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...

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

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

We can implement $f(x,y) = \text{ReLU}(x-y)$

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 LTIs...

- # 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			
protein – RNA		$10^4 - 10^5$	
protein-lipid/ion..			
Total.	10^5 to 10^6	10^6 to 10^7	10^7 to 10^8

But still unsatisfactory...

Not all bindings 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?

(4) Computational tasks in Bio. as Engineered.

- Logic gates in computers.

  Task: Logical deductions.

  Do cells need to do logical deductions?

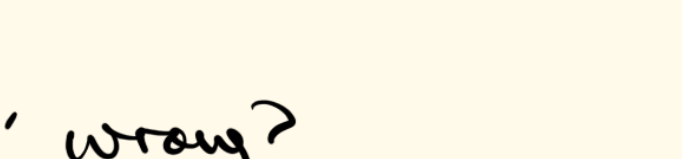
- LTUs in ANNs.

  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

Example: MultiFate. Zhu Ronghui. Elowitz lab. 2022. Science.

Computation task in cell differentiation? (Transcription factors)

10^2 to 10^3 cell types. ideally encoded by 10 genes,

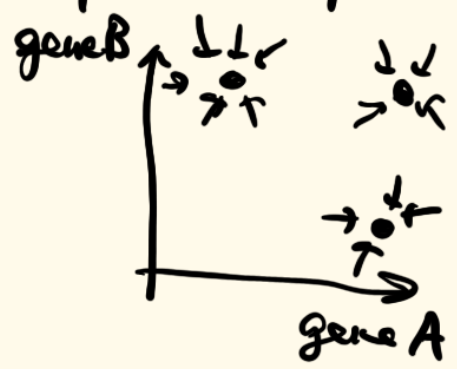
not 10^2 to 10^3.

How? Combinatorial encoding. (total TF in human: 10^3).

e.g. A, B, C. genes. each high/low. $\Rightarrow$ 2^N cell types.

How to implement? **Multistability** in gene expression space.

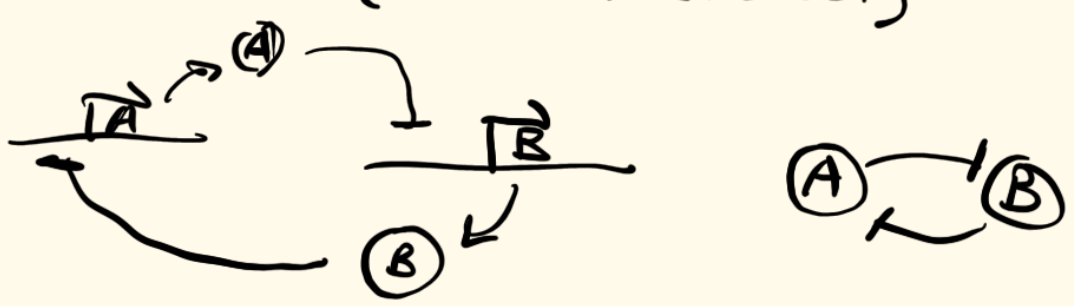
Each cell type is a distinct stable state.



- Example of multistability — Toggle Switch.

(Jim Collins. 2000.)

Mutual repression.



$G_A + B \rightleftharpoons G_A^B$

$G_B + A \rightleftharpoons G_B^A$.

$$\frac{dA_{tot}}{dt} = \alpha + \beta \frac{G_A}{G_{tot}} - \delta A_{tot}$$

$$\frac{dB_{tot}}{dt} = \alpha + \beta \frac{G_B}{G_{tot}} - \delta B_{tot}.$$

Assume $G_{tot} \ll K_A, K_B$.

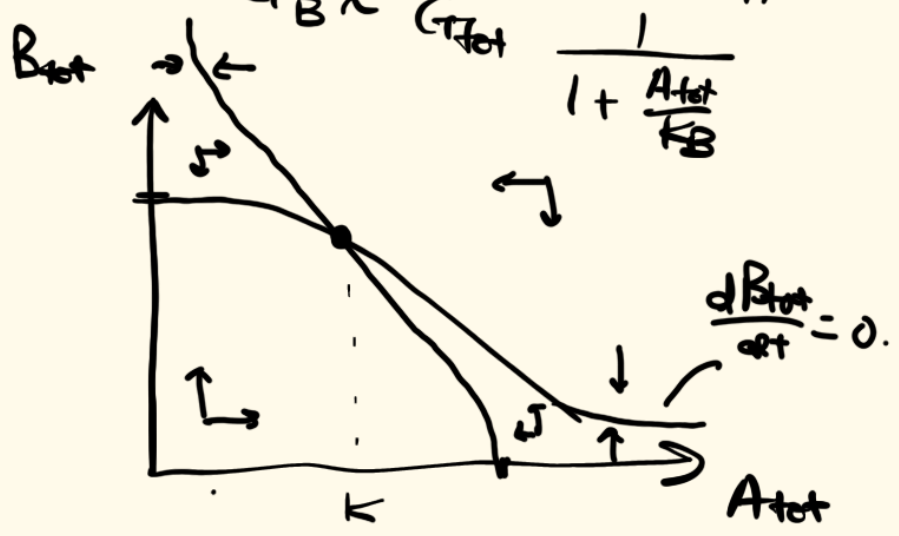
$\Rightarrow$. $G_A \sim G_{tot} \dfrac{1}{1 + \frac{B_{tot}}{K_A}}$

$G_B \sim G_{tot} \dfrac{1}{1 + \frac{A_{tot}}{K_B}}$

Draw nullcline.

$$0 = \frac{d}{dt} B_{tot} : \quad \alpha + \beta \frac{G_B}{G_{tot}} = \delta B_{tot}.$$

$$B_{tot} = \frac{1}{\delta}\left(\alpha + \beta \frac{1}{1 + \frac{A_{tot}}{K_A}}\right)$$



Only one intersection.

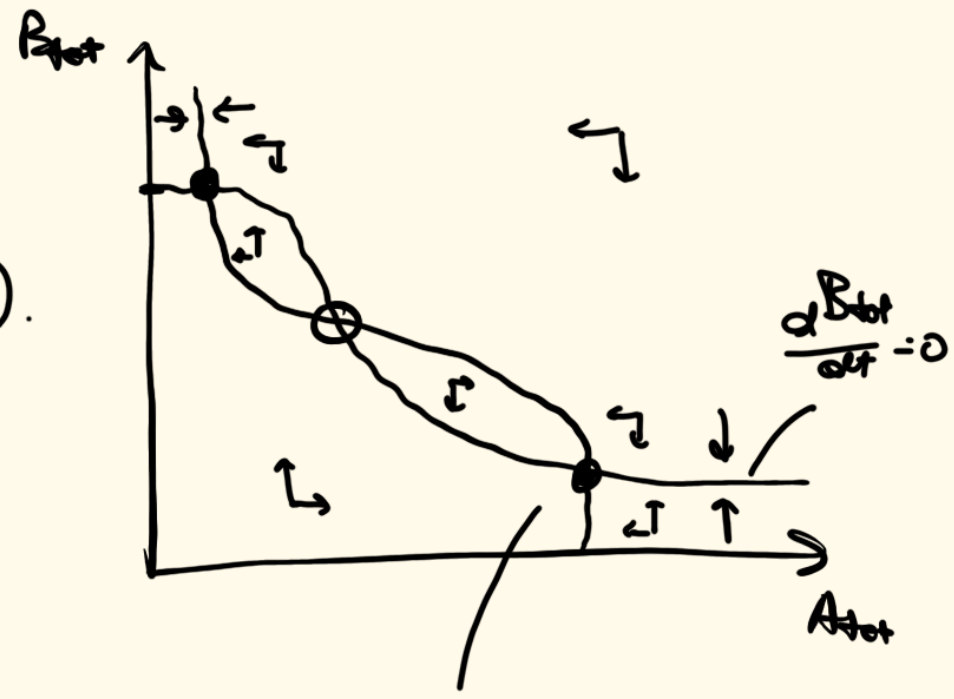
Needs some more curvature.

This can be done by

letting repressors be dimers (or multimers).

e.g. $G_A + 2B \rightleftharpoons G_A^B$

then. $G_A \sim G_{tot} \dfrac{1}{1+\left(\frac{B_{tot}}{K_A}\right)^2}$



- But only bistable

Because the middle intersection.

is at strong repression.

$$\frac{dA_{tot}}{dt} \sim \beta\left(\frac{B_{tot}}{K_A}\right)^{-2} - \delta A_{tot}$$

$$\frac{dB_{tot}}{dt} \sim \beta\left(\frac{A_{tot}}{K_B}\right)^{-2} - \delta B_{tot}$$

Mutual repression is too strong. de-stabilizing!

stable! because.

Just look at reaction order.

$$\begin{bmatrix} -1 & -2 \\ -2 & -1 \end{bmatrix}.$$

(Detail: draw phase portrait)

- Two genes. two stable states ...

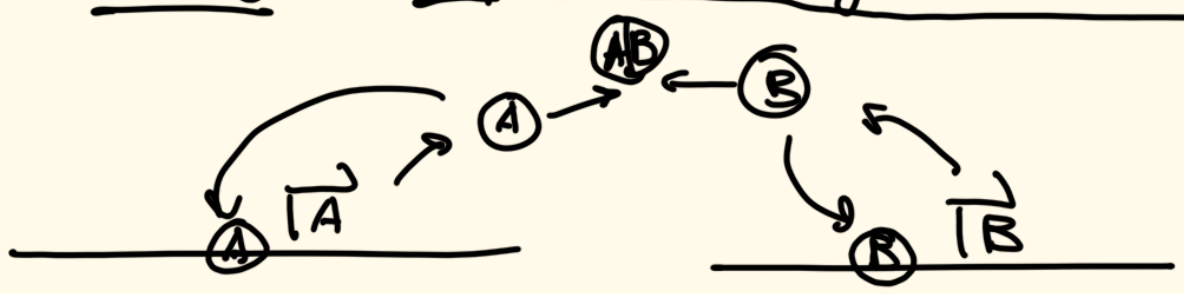
Can we do more? two genes. three states?

- Motif from developmental biology: self-activation
  mutual-inhibition
  (SAMI).

— Easy to make one high, the other low. stable.

— But with binding! Competitive binding for inhibition.

e.g.



$$G_A + A \rightleftharpoons G_A^* \qquad G_B + B \rightleftharpoons G_B^* \qquad A + B \rightleftharpoons C_{AB}$$

$$\frac{d A_{tot}}{dt} = \alpha + \beta \frac{G_A^*}{G_{tot}} - \delta A_{tot}$$

$$\frac{d B_{tot}}{dt} = \alpha + \beta \frac{G_B^*}{G_{tot}} - \delta B_{tot}$$

Assume. $G_{tot} \ll K_A, K_B$.

(So G_A^*, G_B^* are negligible in A_{tot}, B_{tot})

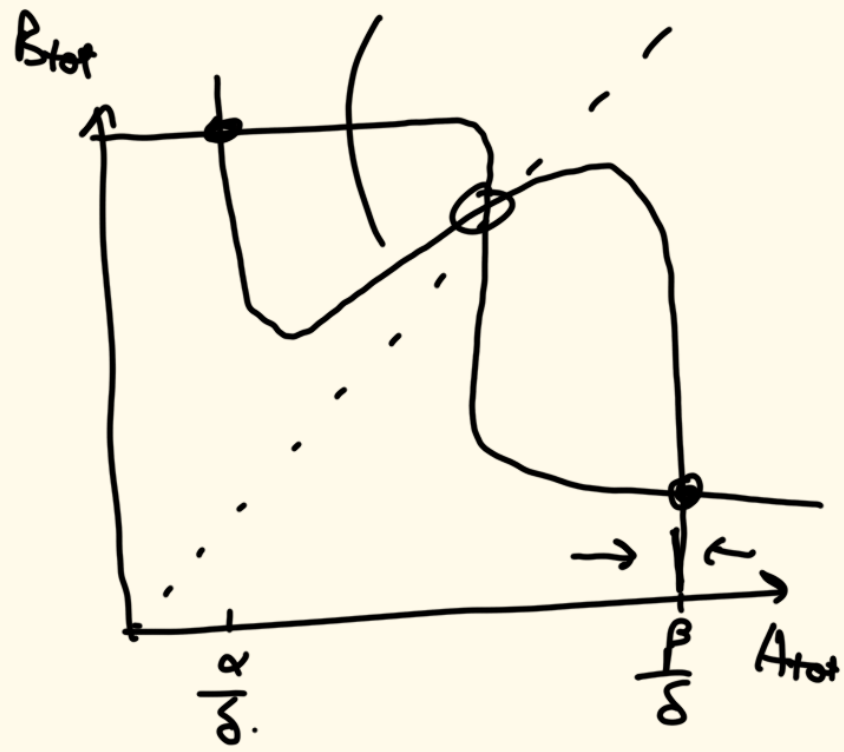
$\rightarrow$. $A_{tot} \sim A + C_{AB}$

$B_{tot} \sim B + C_{AB}$.

$$\Rightarrow. \quad G_A^* \sim G_{tot} \frac{A}{A + K_A}$$

$$G_B^* \sim G_{tot} \frac{B}{B + K_B}$$

curved up!



Draw nullcline of A_{tot}.

$$\alpha + \beta \frac{A}{A + K_A} = \delta A_{tot}$$

① . A_{tot} very low. $\alpha \gg \beta \frac{A}{A + K_A}$.

$A_{tot} \sim \frac{\alpha}{\delta}$. make α small so this is consistent.

② . A_{tot} very large. $\alpha \ll \beta \frac{A}{A + K_A}$.

$$\beta \frac{A}{A + K_A} = \delta A_{tot}.$$

2.1 . B_{tot} is small. so. $A \sim A_{tot}$. is large.

$\Rightarrow$. $A_{tot} \sim \frac{\beta}{\delta}$. say β is large to make this consistent.

2.2 . B_{tot} is comparable with A_{tot}.

So. $A \sim A_{tot} - B_{tot} \ll K_A$.

$$\Rightarrow. \quad \beta \frac{A}{A + K_A} \sim \frac{\beta}{K_A} A \sim \frac{\beta}{K_A} (A_{tot} - B_{tot}).$$

Nullcline is

$$\Rightarrow. \quad \frac{\beta}{K_A} (A_{tot} - B_{tot}) = \delta A_{tot}.$$

$$\Rightarrow. \quad \left(\frac{\beta}{\delta K_A} - 1 \right) A_{tot} = \frac{\beta}{\delta K_A} B_{tot}.$$

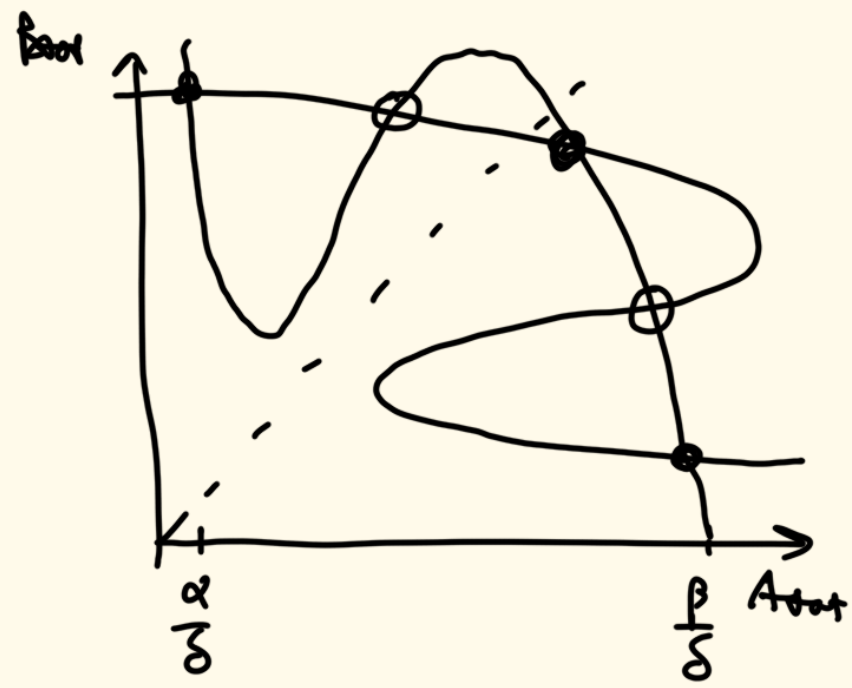
$$B_{tot} = \left(1 - \frac{\delta K_A}{\beta} \right) A_{tot}. \qquad \text{Slope} < 1.$$

— But still only bistable!

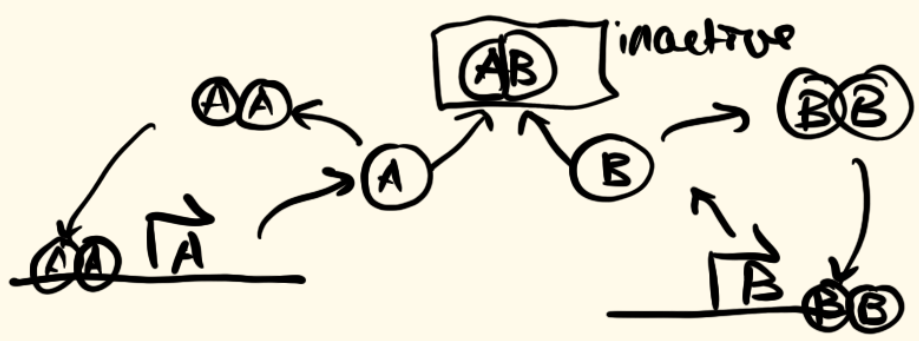
⇒. Increase the "slope upward". e.g. A^2 activation

↳ If (2.2) has nullcline. $A^2 = (A_{tot} - B_{tot})^2$.

Then it will work!



— Homodimer activation & heterodimer inhibition.



$A + A \rightleftharpoons A_2$ $\qquad$ $B + B \rightleftharpoons B_2$

$A + B \rightleftharpoons C_{AB}$

$A_2 + G_A \rightleftharpoons G_A^*$ $\qquad$ $B_2 + G_B \rightleftharpoons G_B^*$

$$\Rightarrow \quad A_{tot} = A + 2A_2 + C_{AB}$$
$$B_{tot} = B + 2B_2 + C_{AB}$$

$$\frac{d A_{tot}}{dt} = \alpha + \beta \frac{G_A^*}{G_{tot}} - \delta A_{tot}, \qquad \frac{d B_{tot}}{dt} = \alpha + \beta \frac{G_B^*}{G_{tot}} - \delta B_{tot}$$

Assume $G_{tot} \ll K_{GA}, K_{GB}$.

$$G_A^* \sim \frac{A_2}{A_2 + K_{GA}} G_{tot} \qquad G_B^* \sim \frac{B_2}{B_2 + K_{GB}} G_{tot}$$

Assume A_2, B_2 never dominates A_{tot}, B_{tot}.

So. $A_{tot} \sim A + C_{AB}$

$B_{tot} \sim B + C_{AB}$.

(2.2). $A \sim A_{tot} - B_{tot}$. $A_2 = \frac{A^2}{K_A} \sim \frac{(A_{tot} - B_{tot})^2}{K_A}$

So. $\frac{d A_{tot}}{dt} = 0$ is. $\beta \frac{A_2}{K_{GA}} = \delta A_{tot}$

$\Rightarrow$. $\beta \frac{(A_{tot} - B_{tot})^2}{K_A K_{GA}} = \delta A_{tot}$.

- So. ultrasensitivity enabled by binding could add extra stable states! Two genes → Three stable states.

How does it scale?

Roughly. 2^n. 10 genes → 100 to 1000 stable states.

- Cells perform Efficient and complex computation by binding!

(Like. 1000 binding reactions.
10 catalysis reactions.)

Next (Last!): Cell as growth machine!