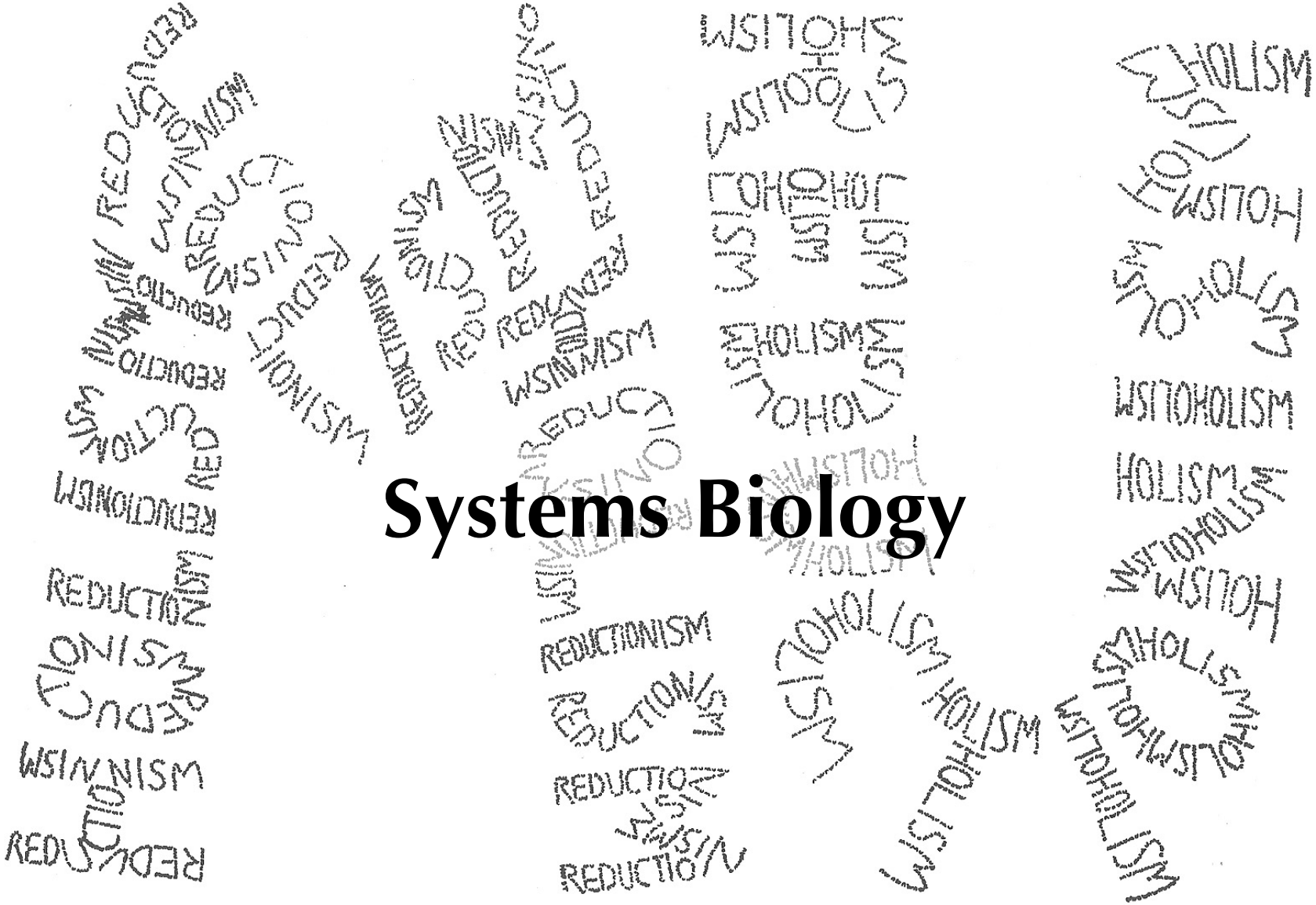


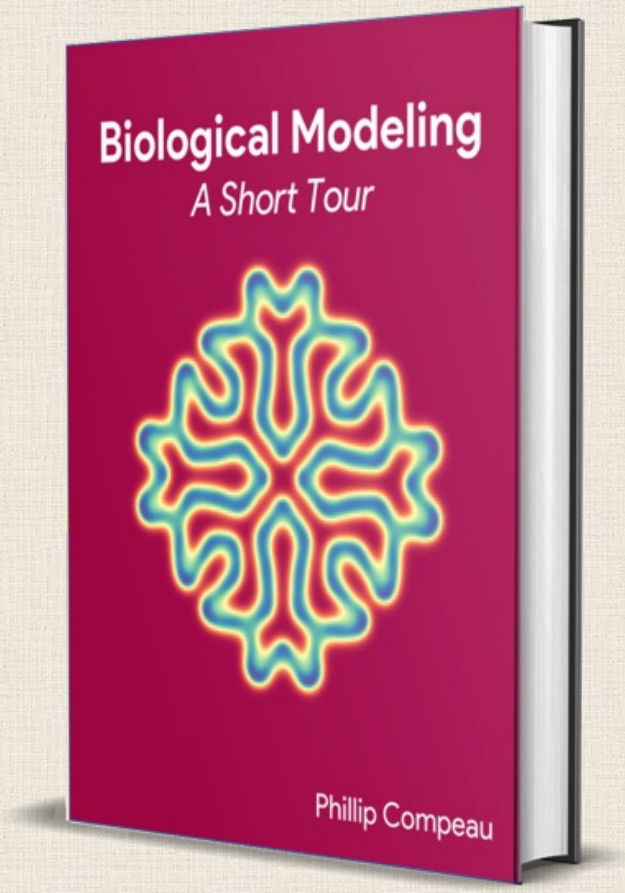
Source: *Godel, Escher, Bach* (Hofstadter).

This lecture comes from the *Biological Modeling* project

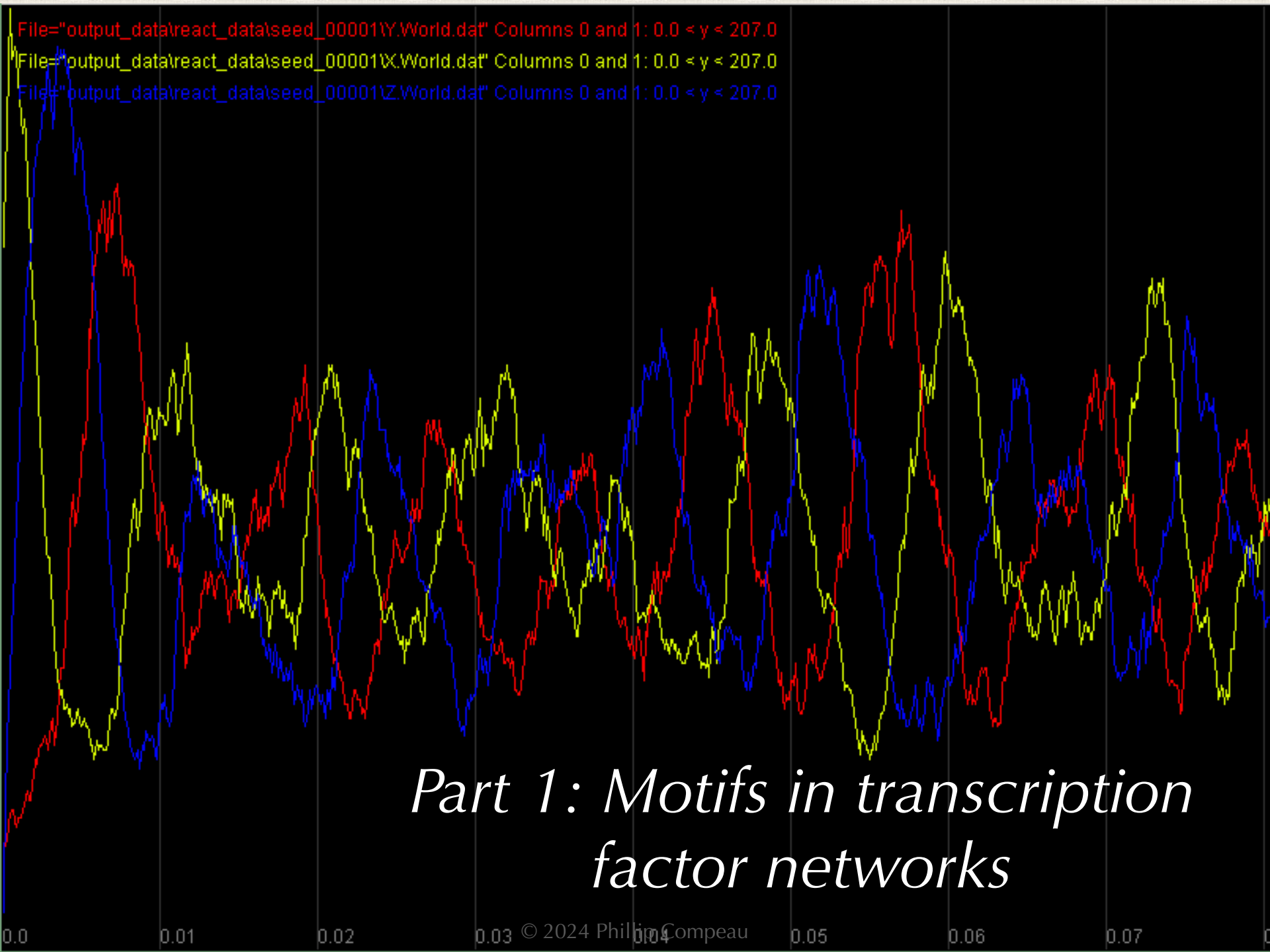
Special thanks to the two students who helped me build this content ☺

Noah Lee
(BSCB 2021)

Shuanger Li
(MSCB 2021)

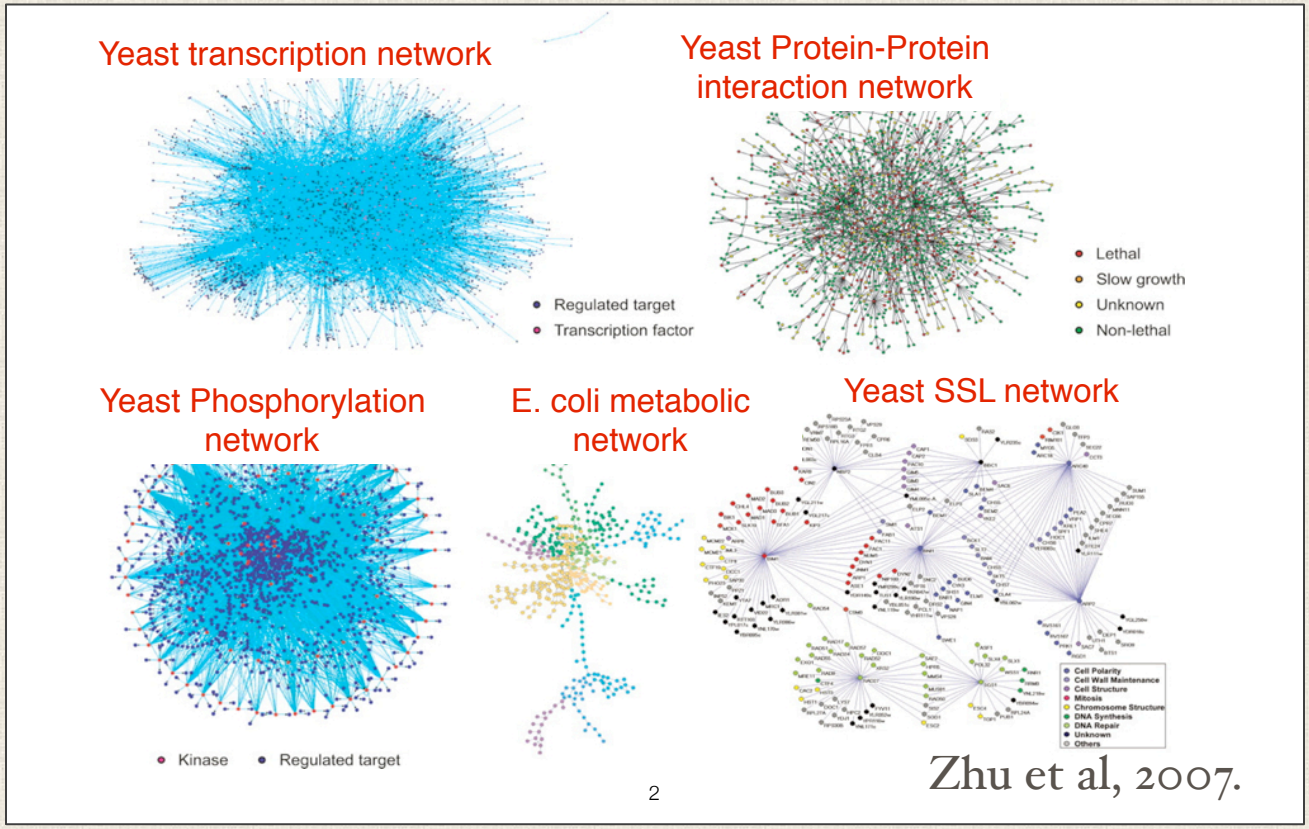


https://biologicalmodeling.org



Part 1: Motifs in transcription factor networks

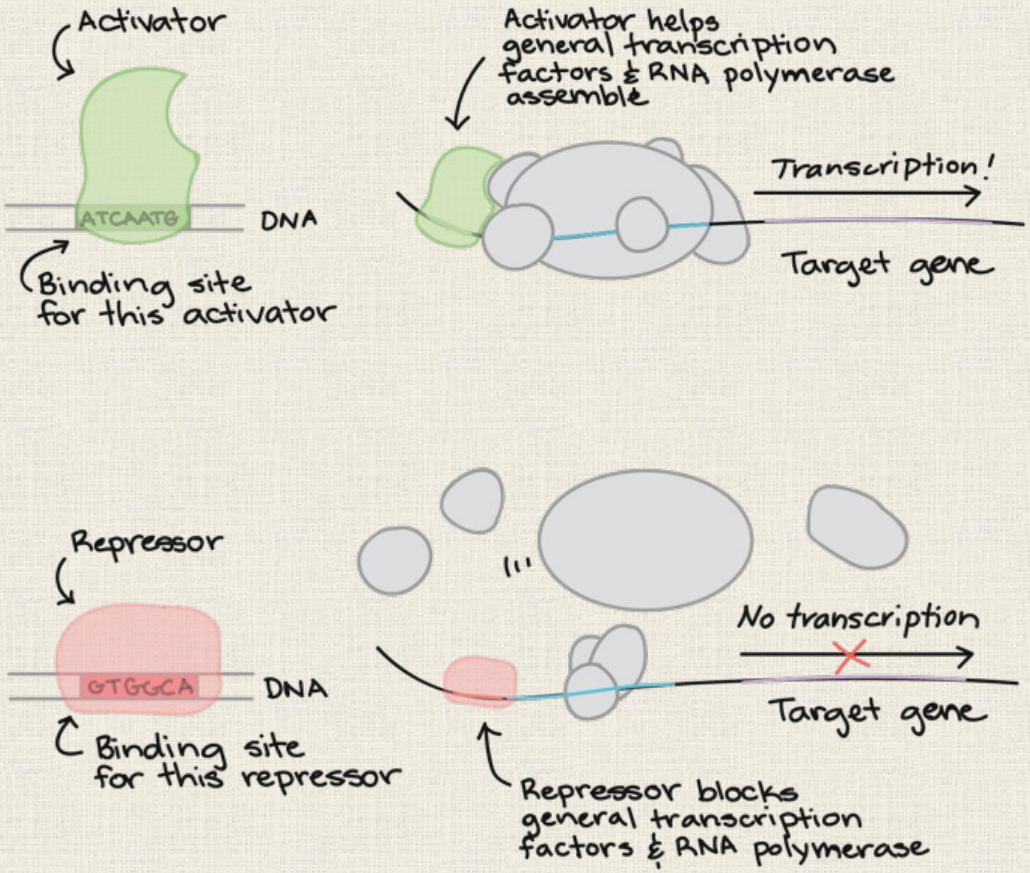
Biological Networks are Everywhere



Are there any recurring "subgraphs", or **network motifs**, that occur surprisingly often?

Recall: Transcription Factors

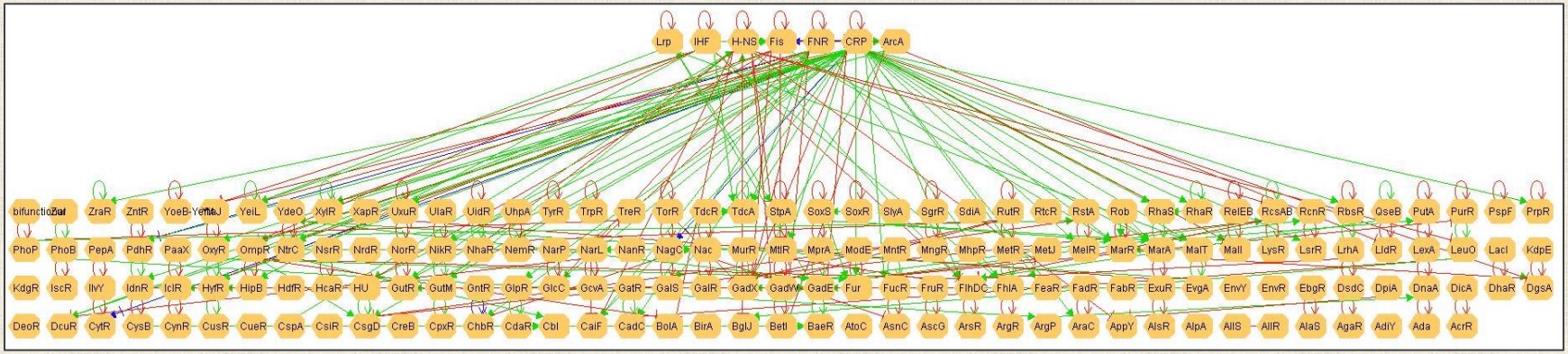
A **transcription factor** can either cause the cell to increase (activate) or decrease (repress) the production of RNA/protein corresponding to a given gene.



Transcription Factor Networks

Transcription factor network:

- Nodes: genes
- Edges: *x* is connected to *y* if *x* is a transcription factor that regulates the expression of *y*.



E. coli transcription factor network

Transcription Factor Networks

Transcription factor network:

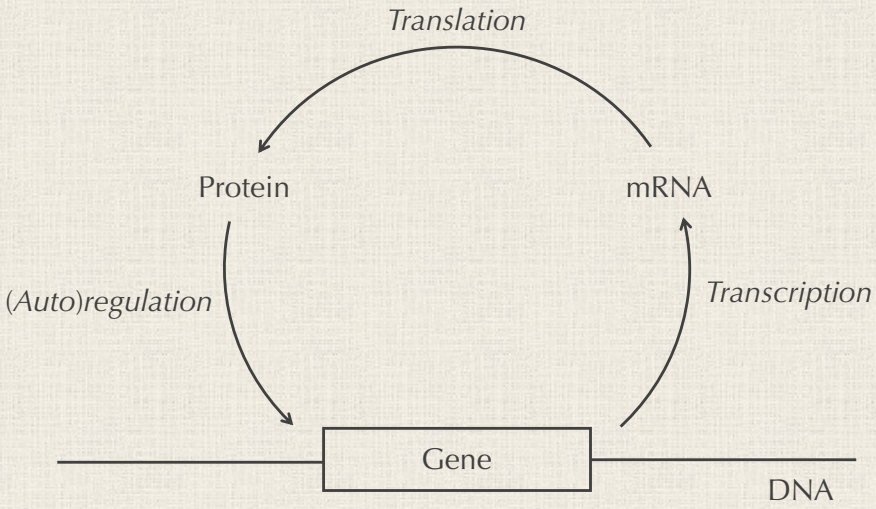
- Nodes: genes
- Edges: *x* is connected to *y* if *x* is a transcription factor that regulates the expression of *y*.

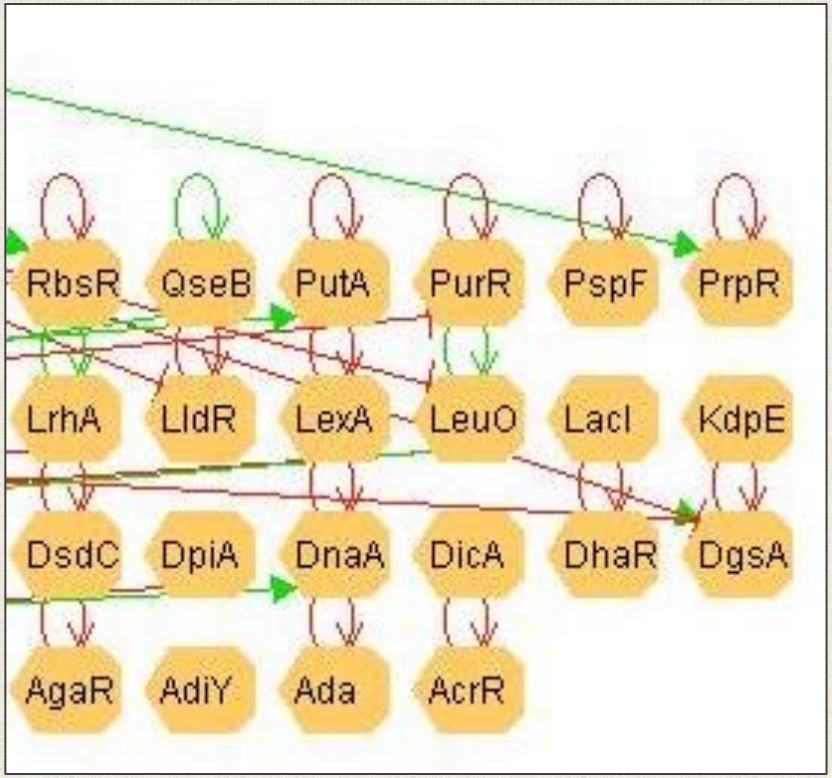
Notes:

1. Some of the nodes are transcription factors; others aren't.
2. Edge *x* → *y* is also labeled +/- according to whether *x* activates/represses *y*.

There sure are a lot of (feedback) loops!

Autoregulation: a transcription factor *Y* regulates *its own* transcription.





Source: https://journals.aps.org/pre/abstract/10.1103/PhysRevE.97.062407

Two Questions

Question 1: How can we justify that the number of loops in a TF network is "surprisingly large"?

Question 2: If autoregulation is so common, then *why* did such a strange mechanism evolve?

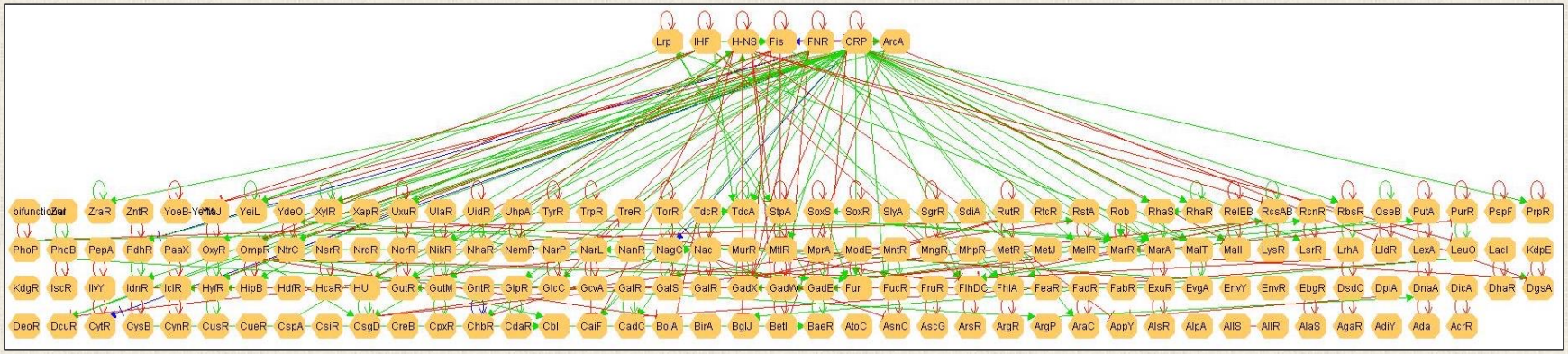
"Nothing in biology makes sense except in the light of evolution."

Theodosius Dobzhansky

AUTOREGULATION IS SURPRISINGLY FREQUENT

Transcription Factor Networks

STOP: What does it mean for a biological network motif to occur "surprisingly often"?

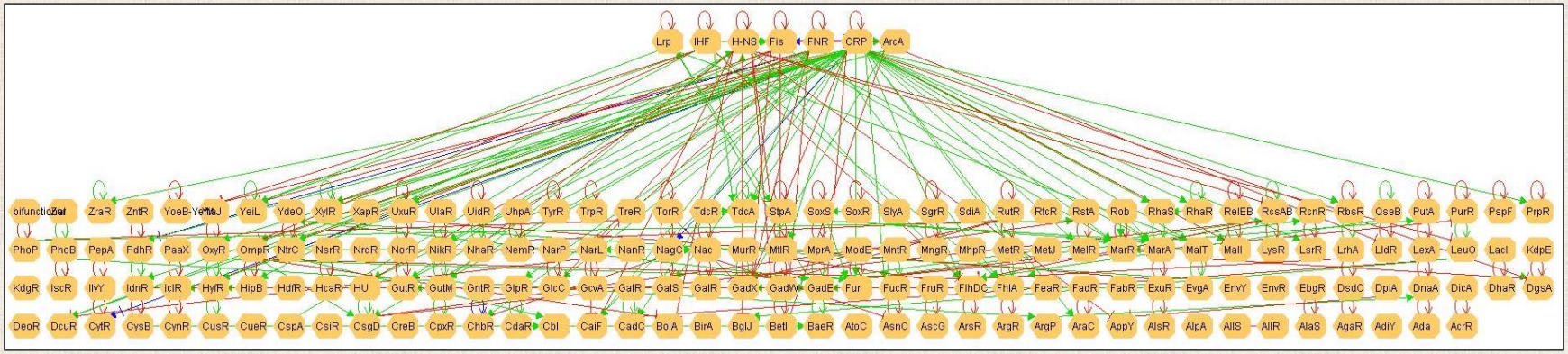


E. coli transcription factor network

Transcription Factor Networks

STOP: What does it mean for a biological network motif to occur "surprisingly often"?

Answer: It occurs more often than it would *if the network were random*! (Nothing new here ☺.)



E. coli transcription factor network

Constructing Random Networks

Gilbert model for random graphs: given an integer n and a number p between 0 and 1, define $G(n, p)$:

- Form n nodes.
- For all n^2 choices of starting node x and ending node y, connect x to y with probability p.

Random graphs

EN **Gilbert** - The **Annals** of **Mathematical Statistics**, **1959** - JSTOR

1. Introduction. Let N points, numbered 1, 2,'-, N, be given. There are N (N-1)/2 lines which can be drawn joining pairs of these points. Choosing a subset of these lines to draw, one obtains a graph; there are 2N (N-1) 12 possible graphs in total. Pick one of these graphs by the following random process. For all pairs of points make random choices, independent of each other, whether or not to join the points of the pair by a line. Let the common probability of join-ing be p. Equivalently, one may erase lines, with common probability q= 1-p from the …

☆ 99 Cited by 1590 Related articles All 5 versions

Constructing Random Networks

Gilbert model for random graphs: given an integer n and a number p between 0 and 1, define $G(n, p)$:

- Form n nodes.
- For all n^2 choices of starting node x and ending node y, connect x to y with probability p.

We limit ourselves to the TF network comprising only transcription factors that regulate each other. This network has 197 nodes and 477 edges.

Constructing Random Networks

Gilbert model for random graphs: given an integer n and a number p between 0 and 1, define $G(n, p)$:

- Form n nodes.
- For all n^2 choices of starting node x and ending node y, connect x to y with probability p.

We limit ourselves to the TF network comprising only transcription factors that regulate each other. This network has 197 nodes and 477 edges.

STOP: What should n be in our decoy network?

Constructing Random Networks

Gilbert model for random graphs: given an integer n and a number p between 0 and 1, define $G(n, p)$:

- Form n nodes.
- For all n^2 choices of starting node x and ending node y, connect x to y with probability p.

We limit ourselves to the TF network comprising only transcription factors that regulate each other. This network has 197 nodes and 477 edges.

Answer: $n = 197$ (the number of TFs).

Constructing Random Networks

Gilbert model for random graphs: given an integer n and a number p between 0 and 1, define $G(n, p)$:

- Form n nodes.
- For all n^2 choices of starting node x and ending node y, connect x to y with probability p.

We limit ourselves to the TF network comprising only transcription factors that regulate each other. This network has 197 nodes and 477 edges.

STOP: OK, but what should p be?

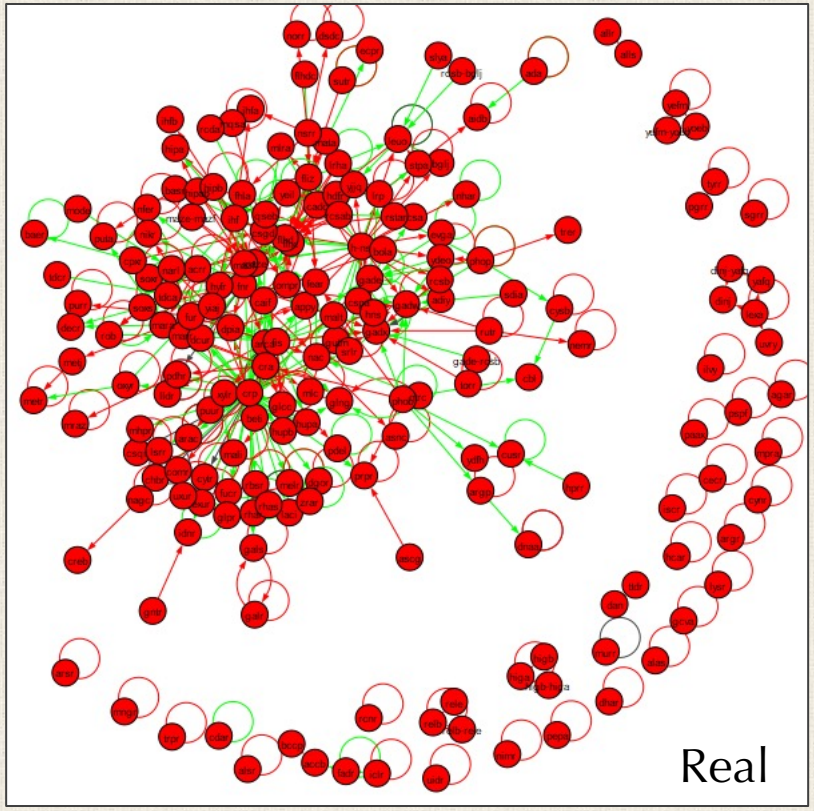
Constructing Random Networks

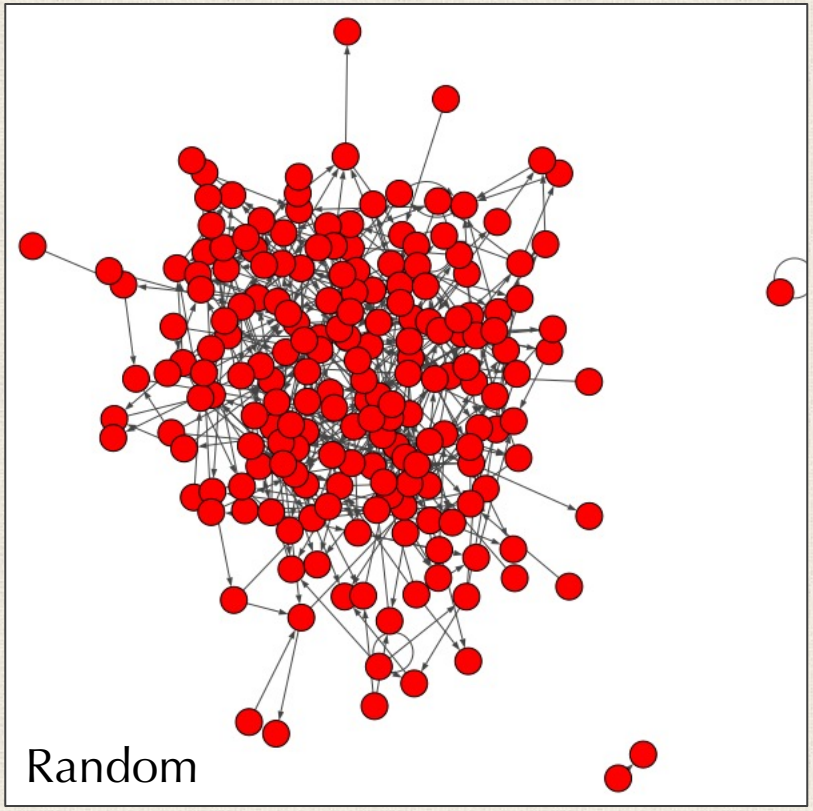
Gilbert model for random graphs: given an integer n and a number p between 0 and 1, define $G(n, p)$:

- Form n nodes.
- For all n^2 choices of starting node x and ending node y, connect x to y with probability p.

Answer: If we were to set p equal to $1/n^2$, then we would on average only see a single edge in the random network. To get 477 edges on average, we set p equal to $477/n^2 = 477/197^2 \approx 0.0123$.

Real vs. Random *E. coli* TF Network

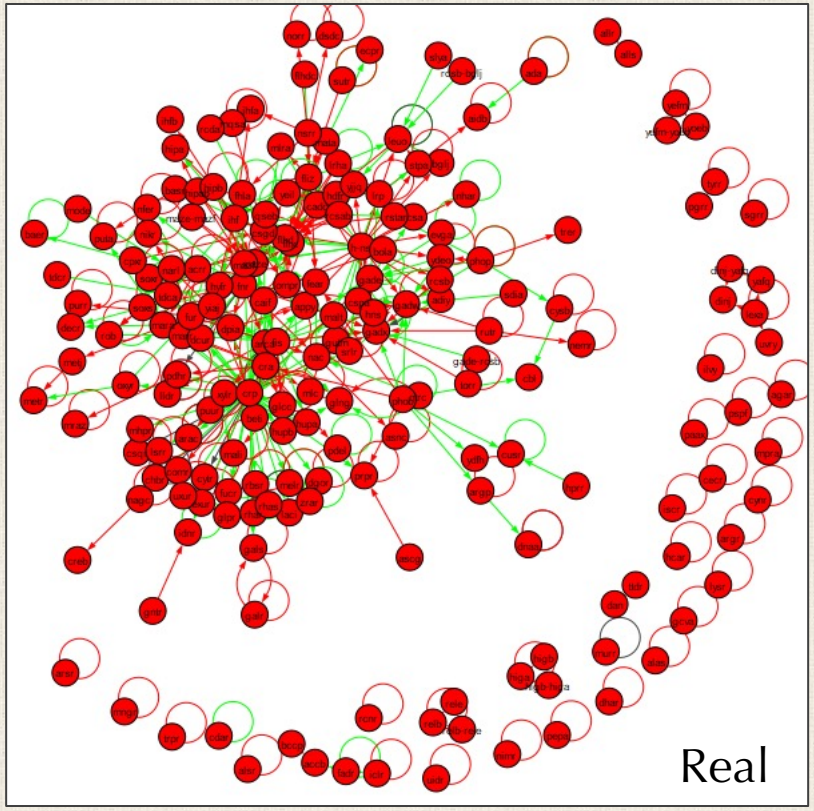


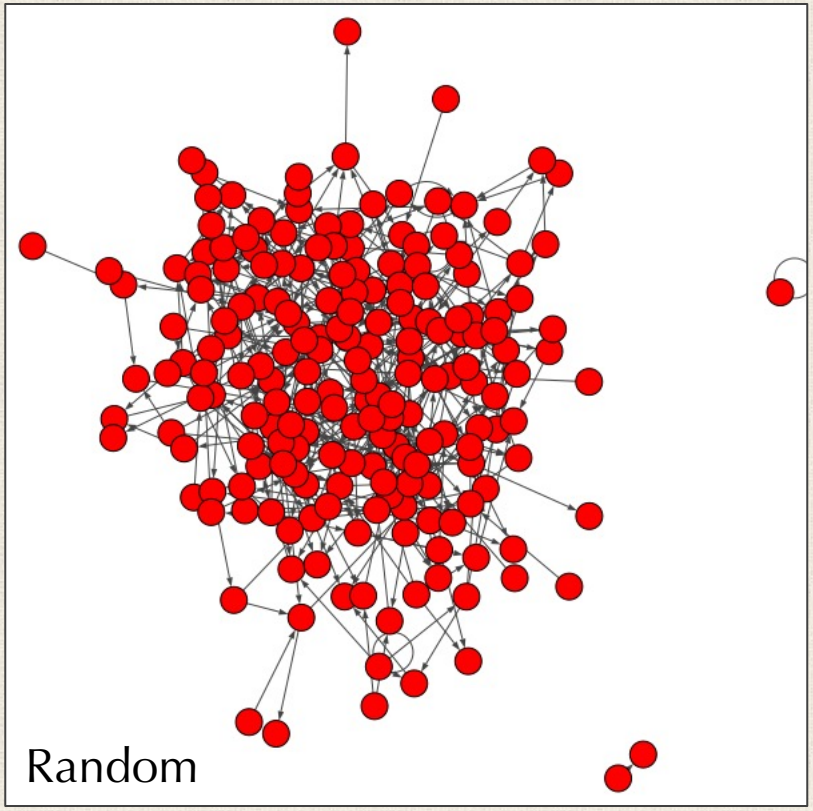


130 loops!

5 loops

Real vs. Random *E. coli* TF Network





130 loops!

95 are repression!

5 loops

Negative Autoregulation: The Simplest Network Motif

Negative autoregulation: a transcription factor *Y* represses its own expression.

www.nature.com › letters › article

Engineering stability in gene networks by autoregulation ...

by A Becskei - 2000 - Cited by 1559 - Related articles

Jun 1, 2000 - The **genetic** and biochemical **networks** which underlie such things as homeostasis in metabolism and the developmental programs of living ...

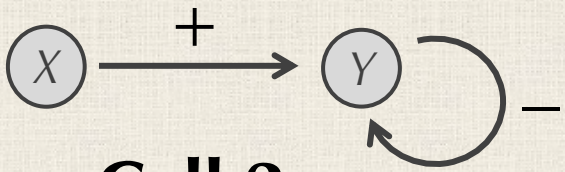
Question 2: If autoregulation is so common, then *why* did such a strange mechanism evolve?

AN EVOLUTIONARY BASIS FOR NEGATIVE AUTOREGULATION

Simulating a Race to Steady State

Say that a TF *X* regulates another transcription factor *Y*, and consider two cells. In both cells, *X* upregulates the transcription of *Y*, but in the second cell, *Y* also negatively autoregulates.

Cell 1

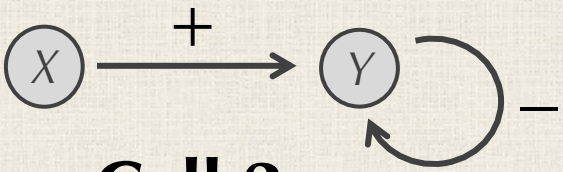


Cell 2

Simulating a Race to Steady State

We will simulate a "race" to the steady-state concentration of *Y* in the two cells. The cell that reaches this steady state faster can respond more quickly to its environment and is therefore more fit for survival.

Cell 1



Cell 2

Simulating Particle Level Simulations

MCell: A software program that simulates **reaction-diffusion** models, in which particles interact with each other as they diffuse randomly.

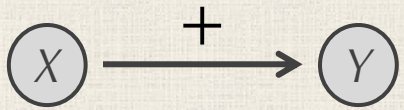
Monte Carlo methods for simulating realistic synaptic microphysiology using **MCell**

JR Stiles, TM Bartol - Computational neuroscience: realistic …, 2001 - books.google.com

… The MDL and **MCell's program** flow are summarized in Figure 4.1 and Boxes 4.1 and 4.2. Specific examples follow in Sections 4.5 and 4.6 … Page 115. **Monte Carlo** Synaptic Models 93 FIGURE 4.1 General Overview of **MCell** Simulations …

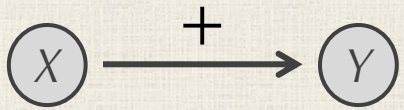
☆ 99 Cited by 307 Related articles All 4 versions

Simulating Cell 1



Initialization: Start with a constant number of *X* particles and no *Y*.

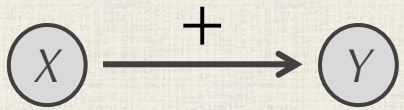
Simulating Cell 1



Initialization: Start with a constant number of *X* particles and no *Y*.

Diffusion: Both *X* and *Y* diffuse at the same rate.

Simulating Cell 1

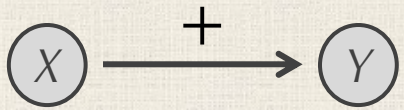


Initialization: Start with a constant number of *X* particles and no *Y*.

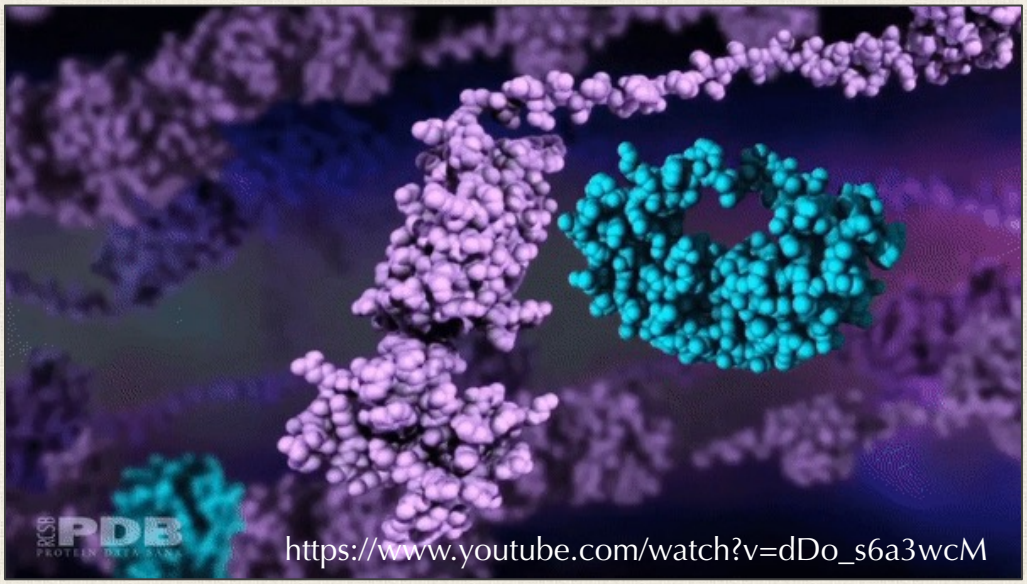
Diffusion: Both *X* and *Y* diffuse at the same rate.

***X* activating *Y*:** we add the reaction $X \rightarrow X + Y$. In any interval of time, there is some probability that an *X* particle will produce a *Y*.

Simulating Cell 1

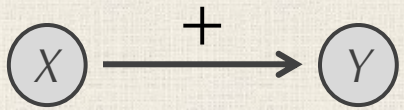


Over time, proteins are *degraded* by proteases so that proteins at high concentration can be removed.



Recall: viruses like HIV and SARS-CoV-2 use proteases to cut their translated RNA genome into protein fragments.

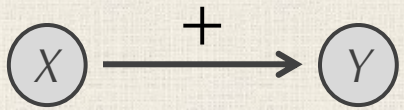
Simulating Cell 1



Over time, proteins are *degraded* by proteases so that proteins at high concentration can be removed.

Kill reaction: *Y* are removed at some rate.

Simulating Cell 1

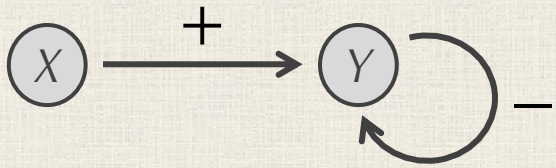


Over time, proteins are *degraded* by proteases so that proteins at high concentration can be removed.

Kill reaction: *Y* are removed at some rate.

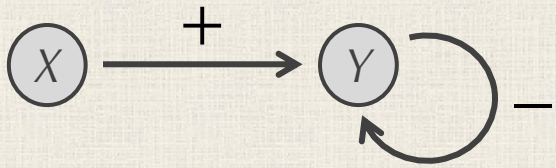
Note: we will assume that *X* is at steady-state, so the rate of production of *X* balances its removal and we do not need such reactions for *X*.

Simulating Cell 2



STOP: What reaction could be used to add a simulation of the negative autoregulation of *Y*?

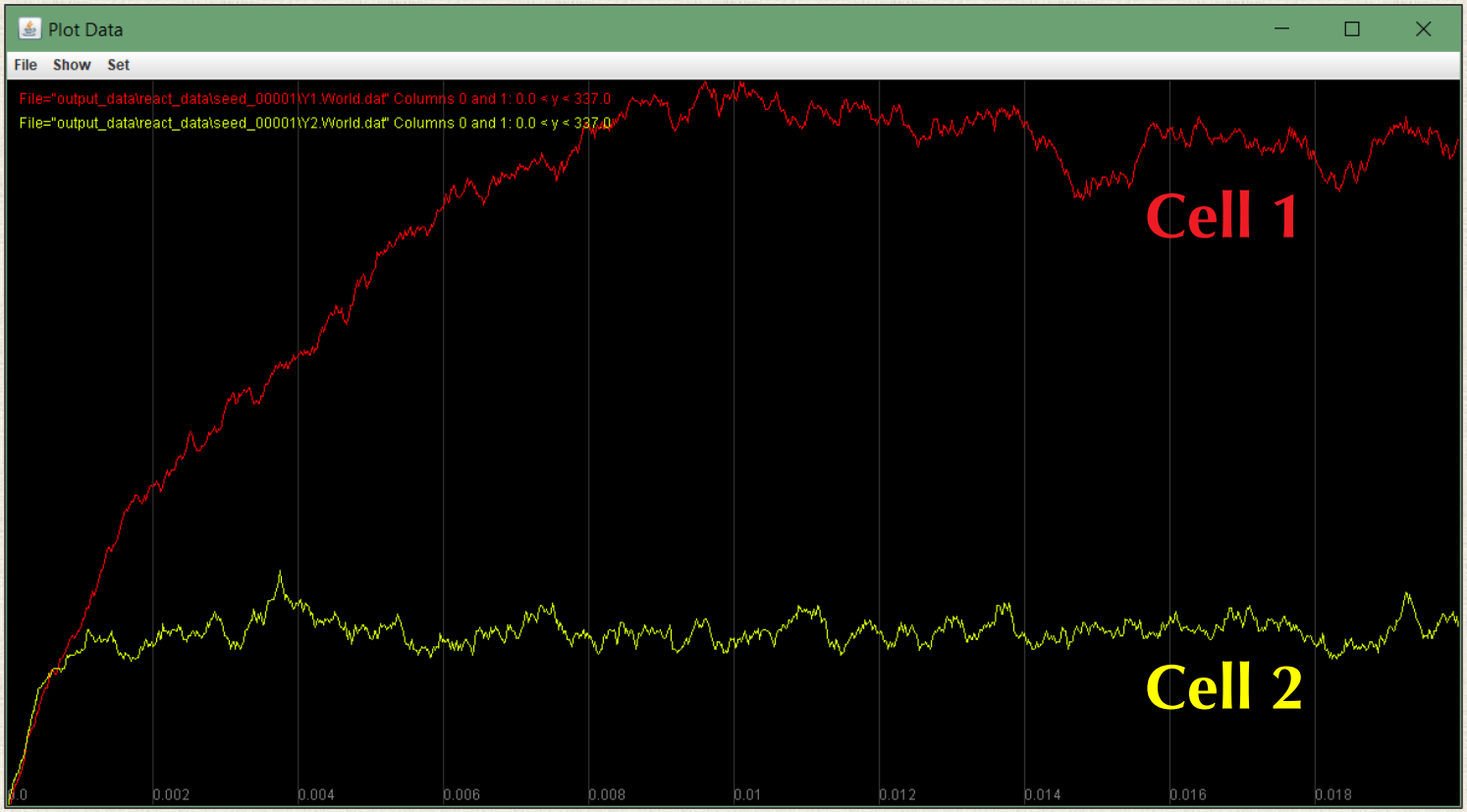
Simulating Cell 2



STOP: What reaction could be used to add a simulation of the negative autoregulation of *Y*?

Answer: We will use $Y + Y \rightarrow Y$. When two *Y* particles encounter each other, there is some probability that one of the particles is removed, which mimics the process of a transcription factor turning off another copy of itself during negative autoregulation.

Comparing concentration of *Y* in the two simulated cells



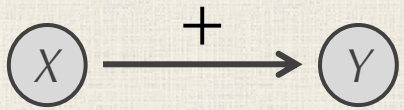
It seems like Cell 1 is winning because its **response time** to the external stimulus is shorter…

Our Comparison Isn't Currently Fair

Mathematically controlled comparison (Savageau 1976): we can only compare models on a mathematically level playing field.

Key Point: if we are comparing the two cells, then the steady-state concentration of *Y* in the two cells should be approximately the same.

Recall Cell 1's Model

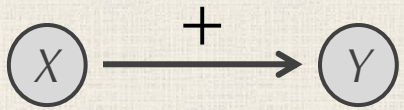


Initialization: Start with a constant number of *X* particles and no *Y*.

Diffusion: Both *X* and *Y* diffuse at the same rate.

***X* activating *Y*:** we add the reaction $X \rightarrow X + Y$. In any interval of time, there is some probability that an *X* particle will produce a *Y*.

Recall Cell 1's Model



Initialization: Start with a constant number of *X* particles and no *Y*.

Diffusion: Both *X* and *Y* diffuse at the same rate.

***X* activating *Y*:** we add the reaction $X \rightarrow X + Y$. In any interval of time, there is some probability that an *X* particle will produce a *Y*.

STOP: How can we ensure that Cell 2 has a higher steady-state concentration of *Y*?

Recall Cell 1's Model

Initialization: Start with a constant number of *X* particles and no *Y*.

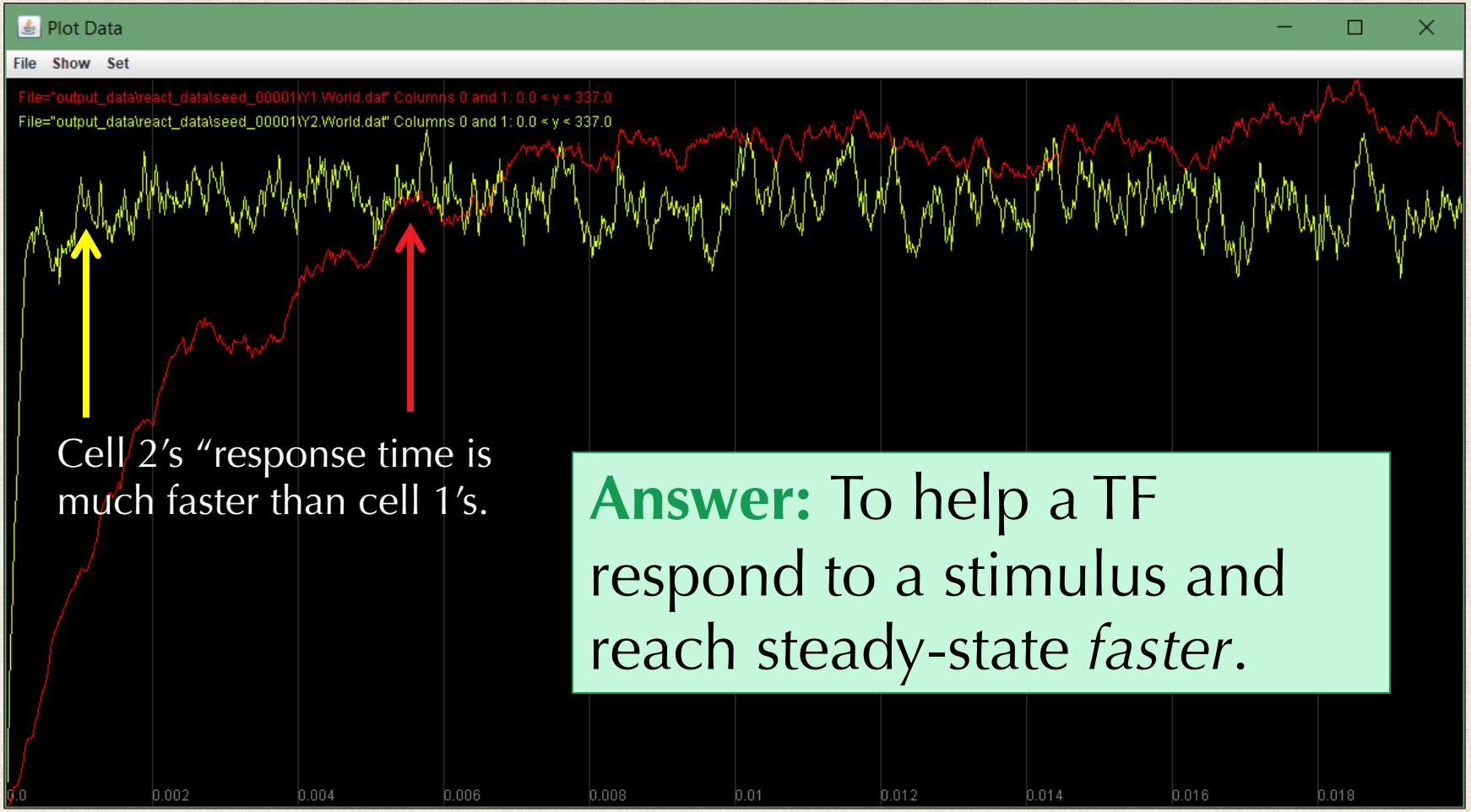
Diffusion: Both *X* and *Y* diffuse at the same rate.

***X* activating *Y*:** we add the reaction $X \rightarrow X + Y$. In any interval of time, there is some probability that an *X* particle will produce a *Y*.

Answer: The only thing that we can change is increasing the *rate* of $X \rightarrow X + Y$ in Cell 2.

Running a Fair Comparison

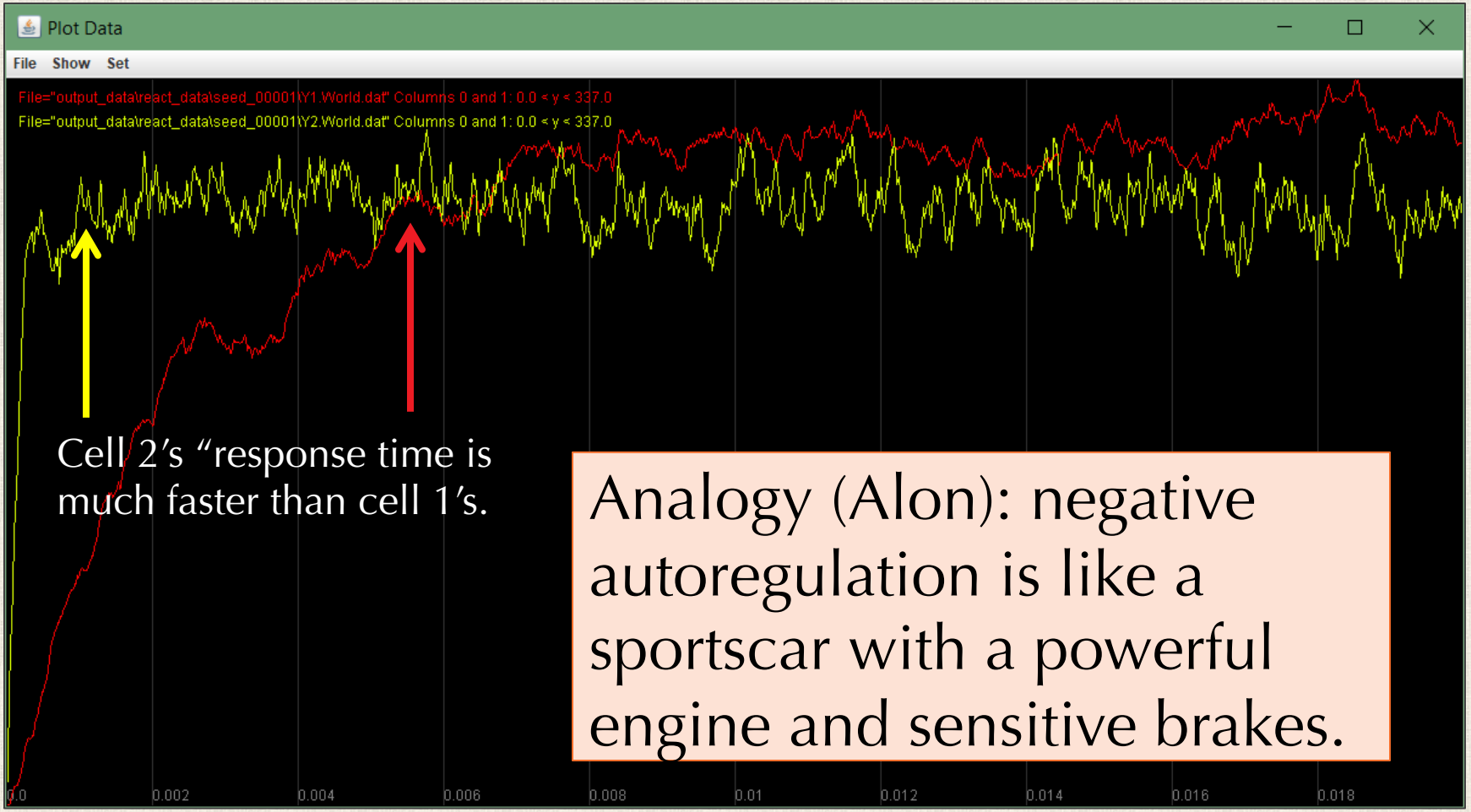
Ensuring that $Y_{st} = X_{st}$



Cell 2's "response time is much faster than cell 1's.

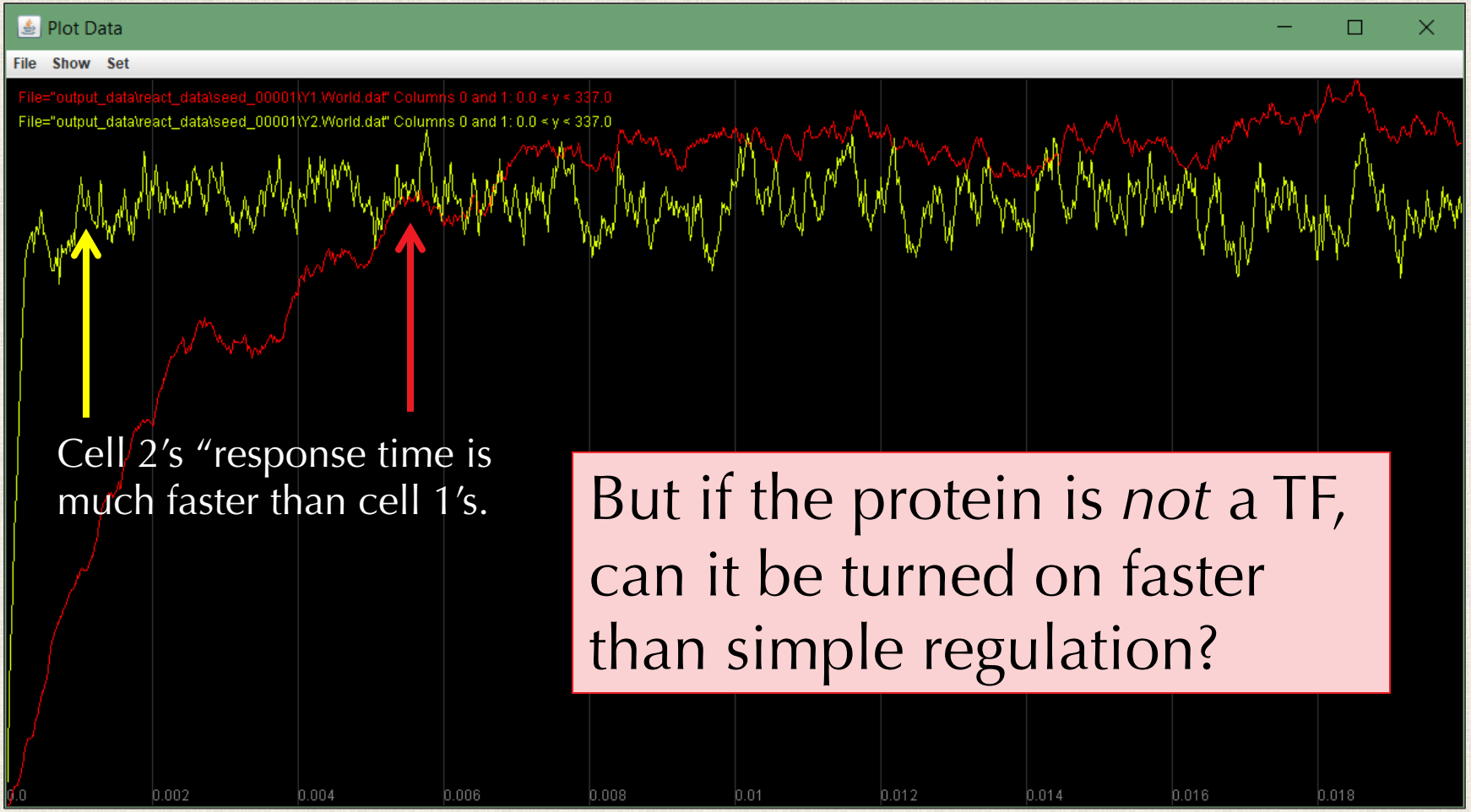
Answer: To help a TF respond to a stimulus and reach steady-state *faster*.

Ensuring that $Y_{st} = X_{st}$



Analogy (Alon): negative autoregulation is like a sportscar with a powerful engine and sensitive brakes.

Ensuring that $Y_{st} = X_{st}$

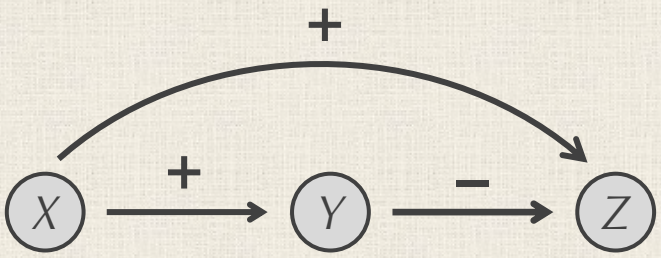


THE FEEDFORWARD LOOP MOTIF

Speeding Up Protein Manufacture for Non-Transcription Factors

Of the 4,400 total *E. coli* proteins, fewer than 10% (about 300) are transcription factors.

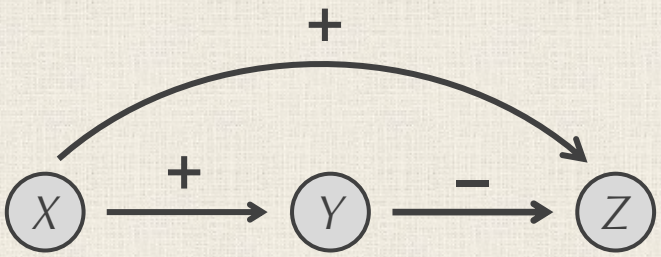
Feed-forward loop (FFL): a network motif connecting *X* → *Y*, *X* → *Z*, and *Y* → *Z*. (It's not a loop.) *E. coli* has 42 total FFLs.



One common labeling of FFL edges

"Type 1" incoherent feed-forward loop: an FFL with the activation/repression pattern shown below.

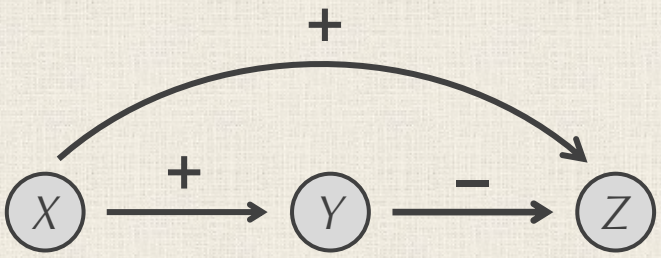
STOP: Why might this motif allow *Z* to be turned on faster than under simple regulation?



One common labeling of FFL edges

"Type 1" incoherent feed-forward loop: an FFL with the activation/repression pattern shown below.

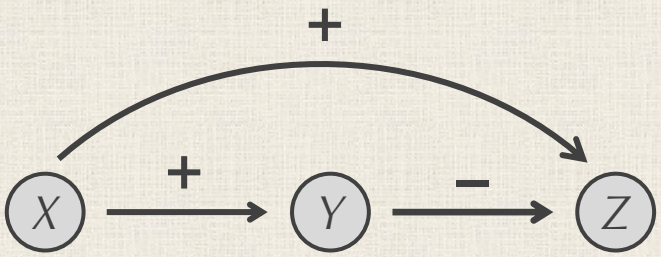
Answer: *X* serves to "ramp up" *Z* quickly, and once *X* builds up *Y*, *Y* serves as a delayed-action "brakes" for *Z*.



One common labeling of FFL edges

"Type 1" incoherent feed-forward loop: an FFL with the activation/repression pattern shown below.

STOP: We are going to compare this FFL against simple activation $X \rightarrow Z$. How can we model this motif with reactions, like what we did previously?



Cell 1 is the same system from before

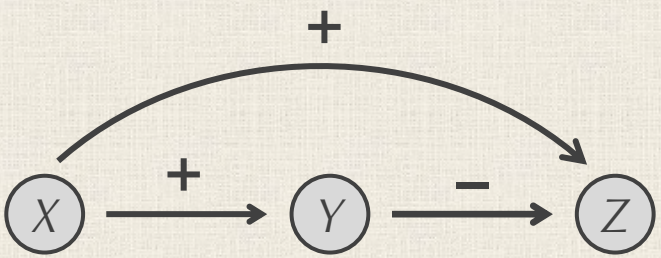
Initialization: Start with a constant number of *X* particles and no *Z*.

Diffusion: Both *X* and *Z* diffuse at the same rate.

***X* activating *Z*:** we add the reaction $X \rightarrow X + Z$. In any interval of time, there is some probability that an *X* particle will produce a *Z*.

Kill reaction: *Z* are removed at some rate.

Cell 2 requires a few more reactions



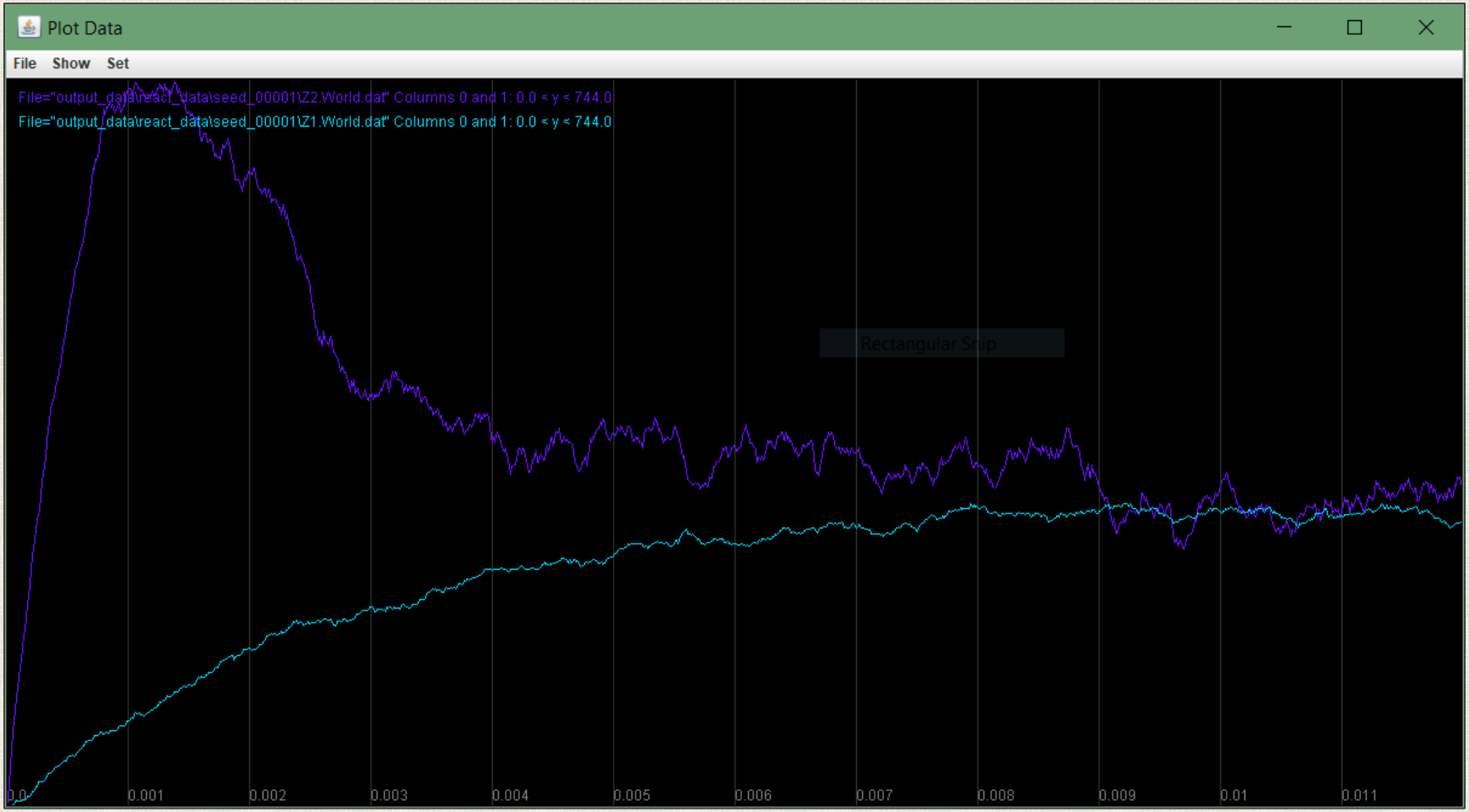
To simulate Cell 2, we will have all of these reactions, with a *higher* rate of activation $X \rightarrow X + Z$ to obtain a mathematically controlled simulation..

***X* activating *Y*:** $X \rightarrow X + Y$.

***Y* repressing *Z* :** $Y + Z \rightarrow Y$.

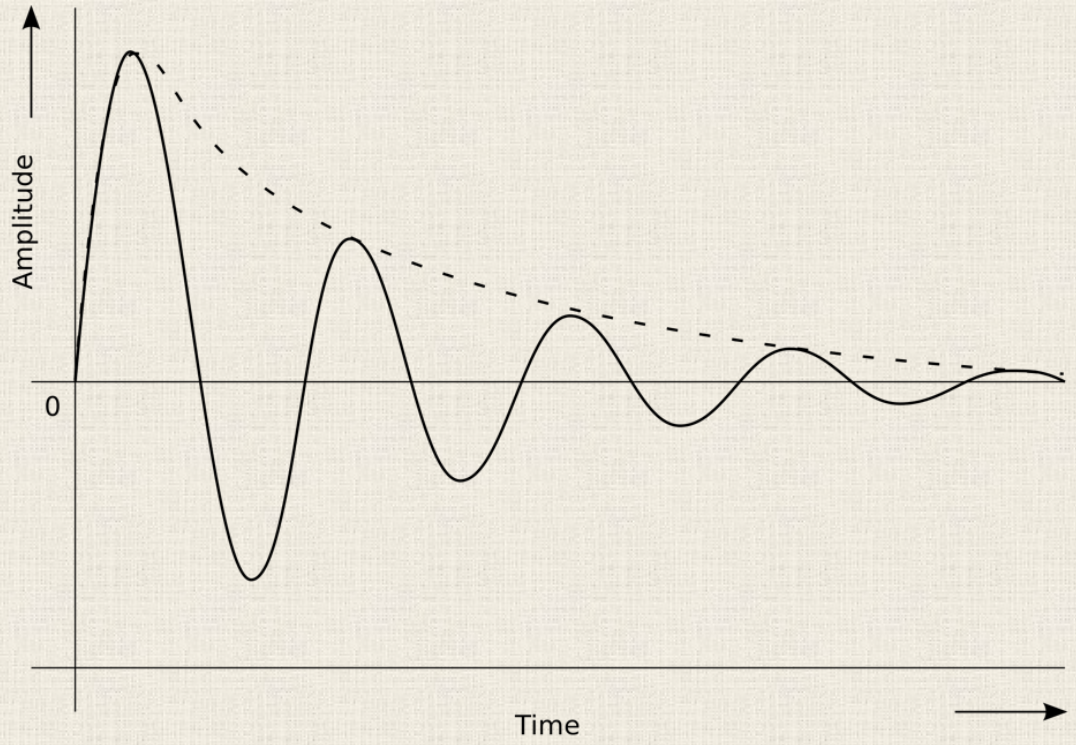
Kill reaction: *Y* and *Z* are removed at some rate.

Plotting a FFL-regulated protein *Z* against one regulated by *X* → *Z*

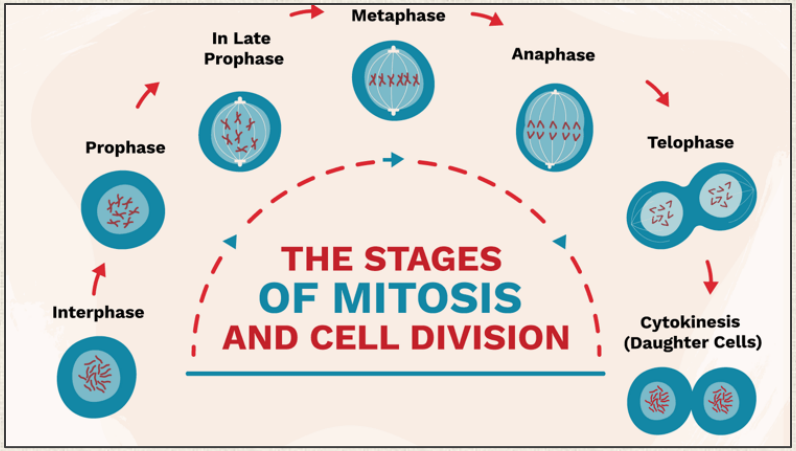


Damped Oscillations

Note: The shape of the FFL-regulated protein concentration is similar to a "damped" oscillation.

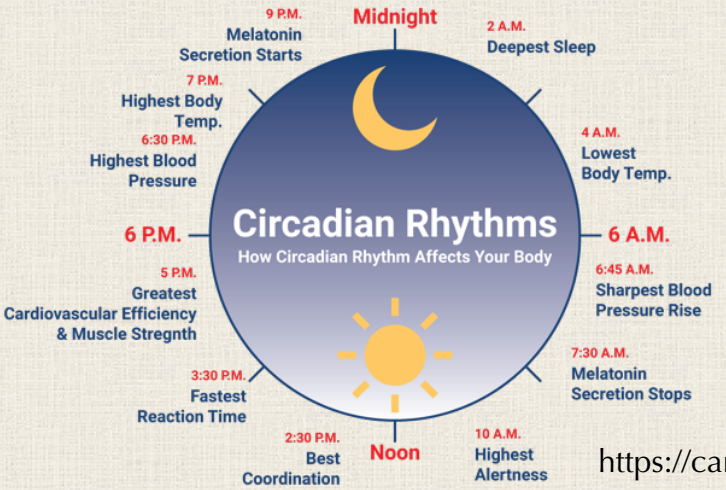


Real oscillations are everywhere in biology



https://www.thoughtco.com/stages-of-mitosis-373534

https://www.narayanahealth.org/blog/what-is-abnormal-heartbeat/



https://carex.com/blogs/resources/circadian-rhythm

The first simple synthetic oscillator

Repressilator: a three-element synthetic "cycle" motif of repression that produces oscillations.

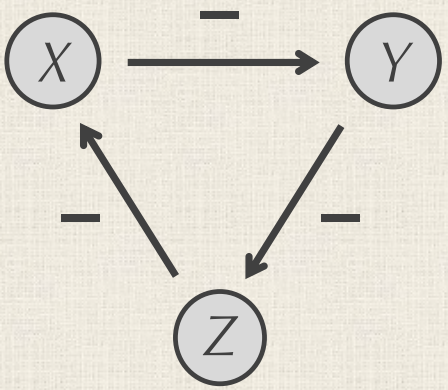
A synthetic oscillatory network of transcriptional regulators

MB Elowitz, S Leibler - Nature, 2000 - nature.com

Networks of interacting biomolecules carry out many essential functions in living cells 1, but the 'design principles' underlying the functioning of such intracellular networks remain poorly understood, despite intensive efforts including quantitative analysis of relatively simple …

☆ 99 Cited by 4909 Related articles All 82 versions

Oscillator motifs in nature are much more complicated than this!



The first simple synthetic oscillator

Repressilator: a three-element synthetic "cycle" motif of repression that produces oscillations.

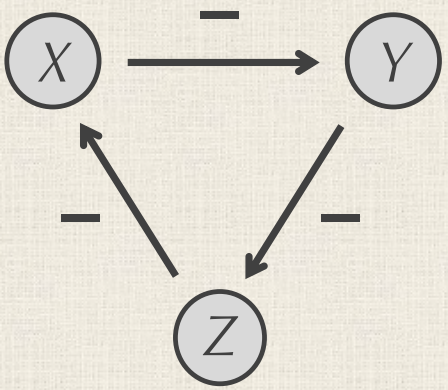
A synthetic oscillatory network of transcriptional regulators

MB Elowitz, S Leibler - Nature, 2000 - nature.com

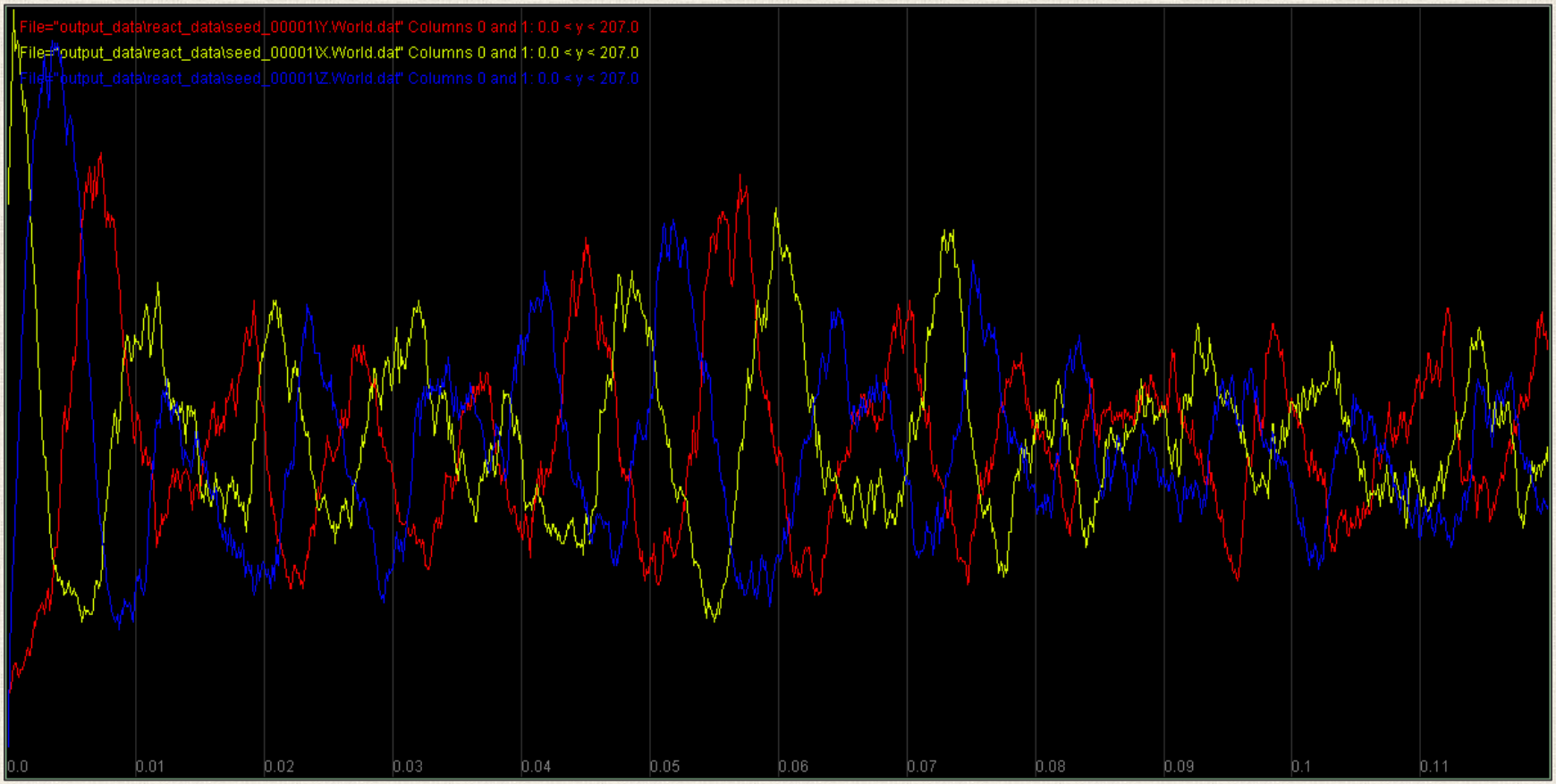
Networks of interacting biomolecules carry out many essential functions in living cells 1, but the 'design principles' underlying the functioning of such intracellular networks remain poorly understood, despite intensive efforts including quantitative analysis of relatively simple …

☆ 99 Cited by 4909 Related articles All 82 versions

STOP: Why might this motif produce oscillatory behavior?



Isn’t the repressilator neat?



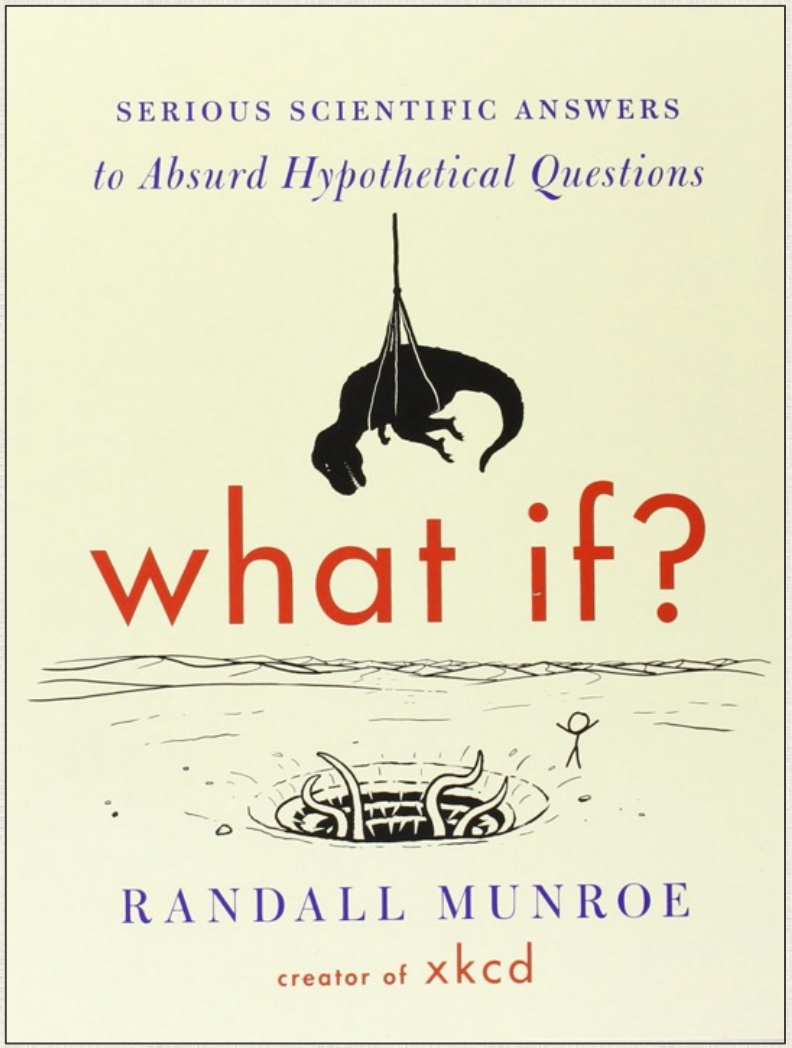
Part 2: Modeling Bacterial Chemotaxis

THE LOST IMMORTALS

An Absurd Hypothetical Question

If two immortal people were placed on opposite sides of an uninhabited Earth-like planet, how long would it take them to find each other?
100,000 years?
1,000,000 years?

STOP: Any thoughts?



Munroe's Answer: "Be an Ant"

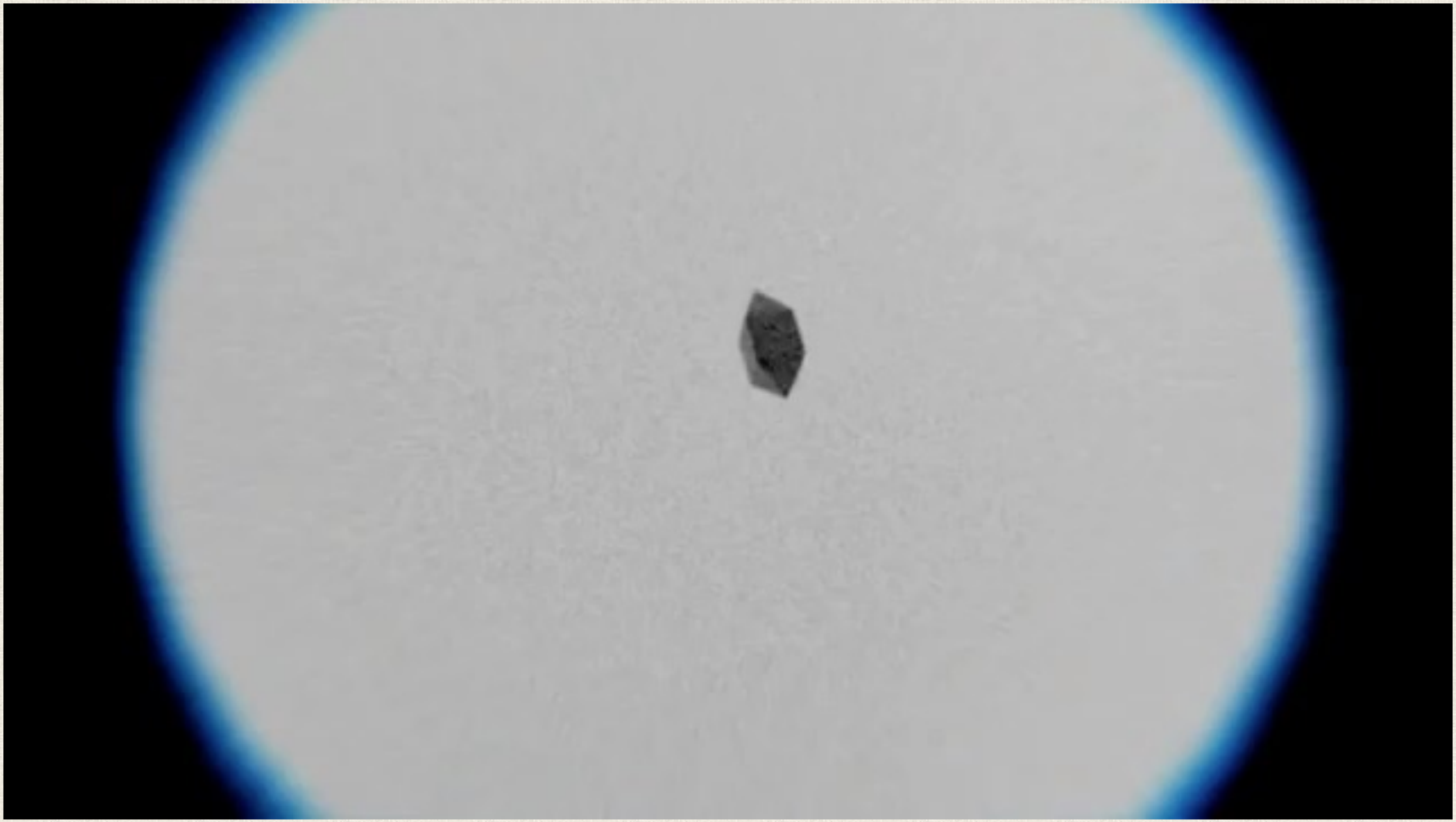
- *If you have no information, walk at random, leaving a trail of stone markers, each one pointing to the next. For every day that you walk, rest for three. Periodically mark the date alongside the cairn. It doesn't matter how you do this, as long as it's consistent. You could chisel the number of days into a rock, or lay out rocks to plot the number.*

- *If you come across a trail that's newer than any you've seen before, start following it as fast as you can. If you lose the trail and can't recover it, resume leaving your own trail.*

- *You don't have to come across the other player's current location; you simply have to come across a location where they've been. You can still chase one another in circles, but as long as you move more quickly when you're following a trail than when you're leaving one, you'll find each other in a matter of years or decades.*

- *And if your partner isn't cooperating—perhaps they're just sitting where they started and waiting for you—then you'll get to see some neat stuff.*

Bacteria employ a similar randomized algorithm to find food



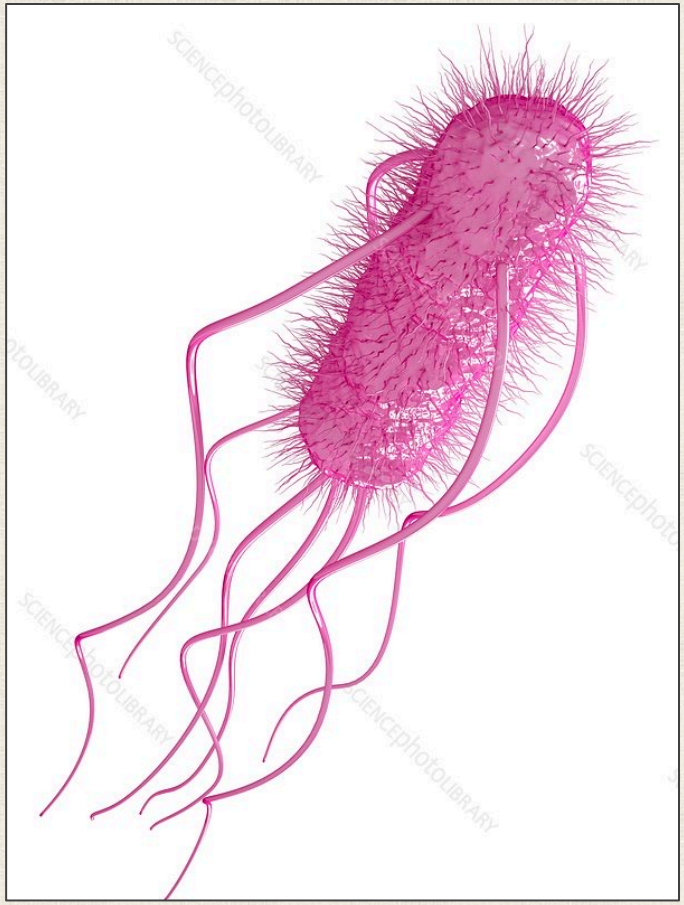
https://www.youtube.com/watch?v=F6QMU3KD7zw

E. COLI AND ITS RANDOM WALK EXPLORATION

Bacterial runs and tumbles

An *E. coli* cell has 5-12 flagella on its surface, which can rotate both clockwise and counter-clockwise.

Chemotaxis: The movement of an organism in response to a chemical stimulus.

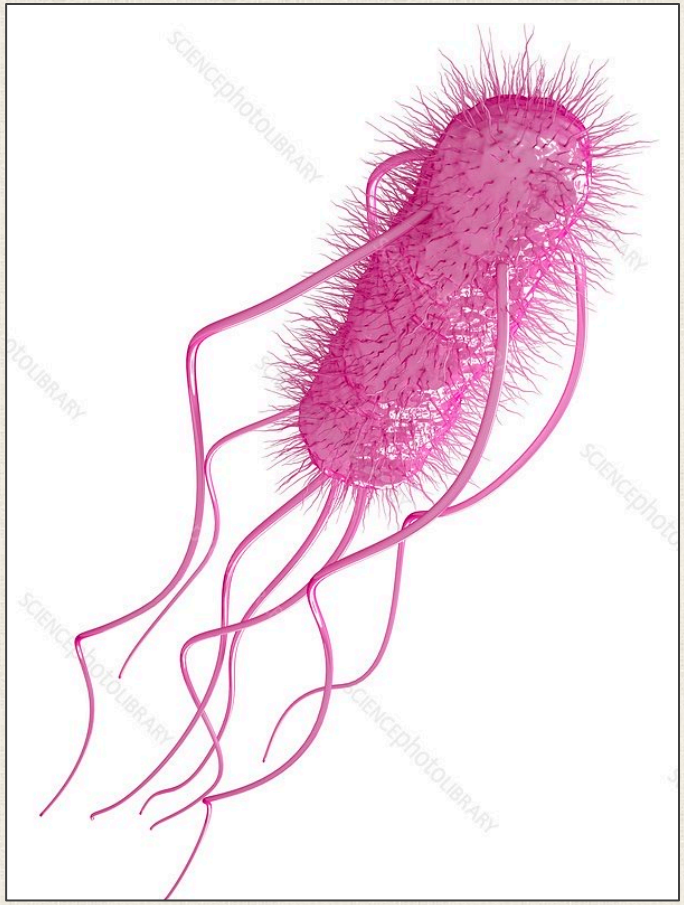


https://www.sciencephoto.com/media/659604/view/e-coli-bacterium-illustration

Bacterial runs and tumbles

An *E. coli* cell has 5-12 flagella on its surface, which can rotate both clockwise and counter-clockwise.

When the flagella are all rotating CCW, they form a bundle and propel the cell forward at 20 μm/s.

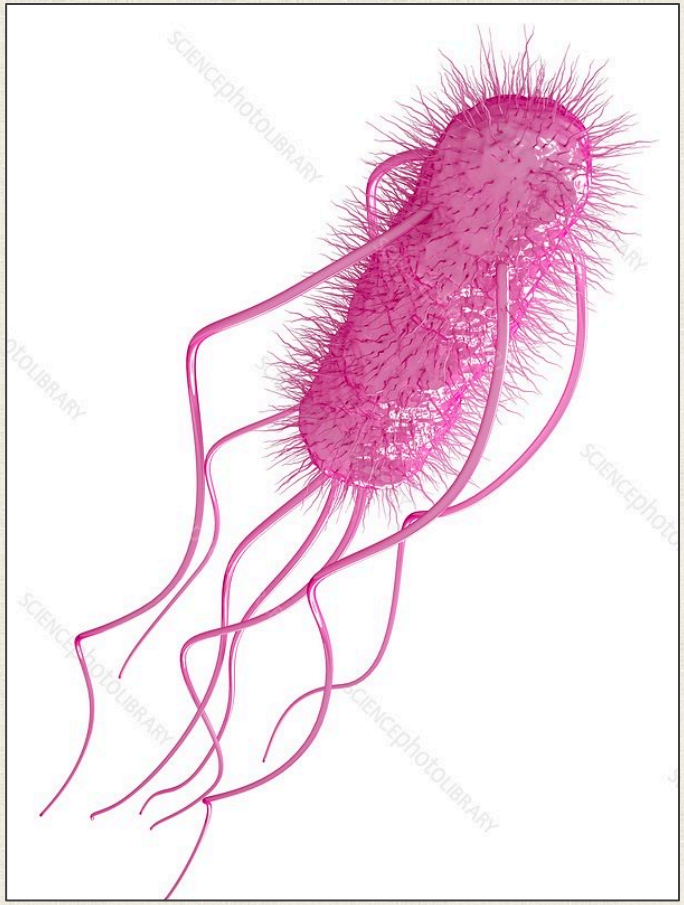


https://www.sciencephoto.com/media/659604/view/e-coli-bacterium-illustration

Bacterial runs and tumbles

Note: This is about 10x the length of the cell per second, like a car traveling at 160 kph (100 mph).

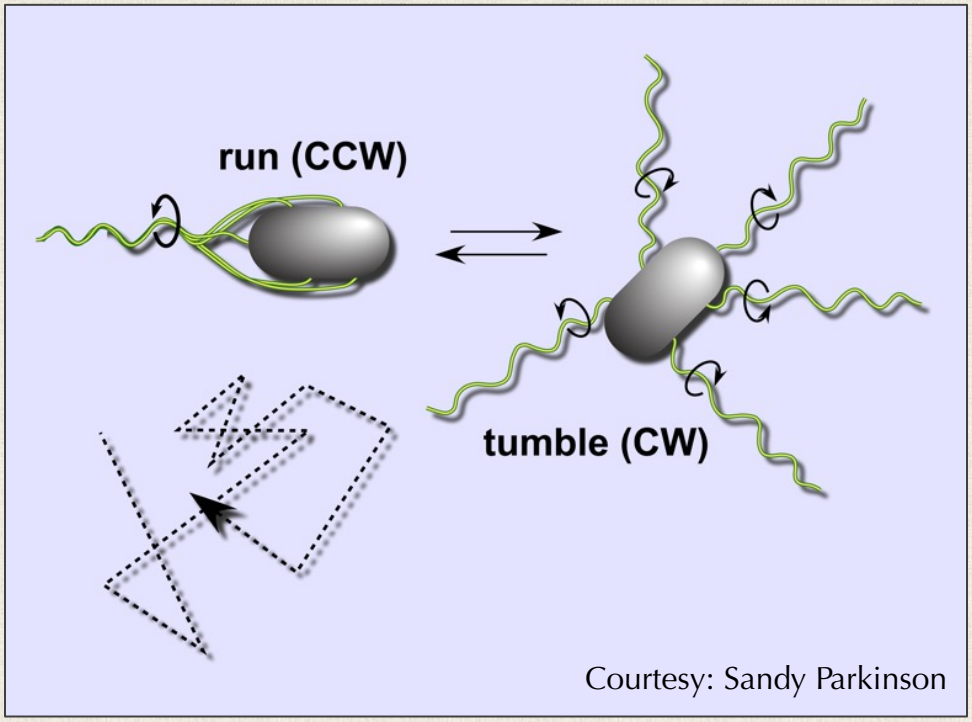
When the flagella are all rotating CCW, they form a bundle and propel the cell forward at 20 μm/s.



https://www.sciencephoto.com/media/659604/view/e-coli-bacterium-illustration

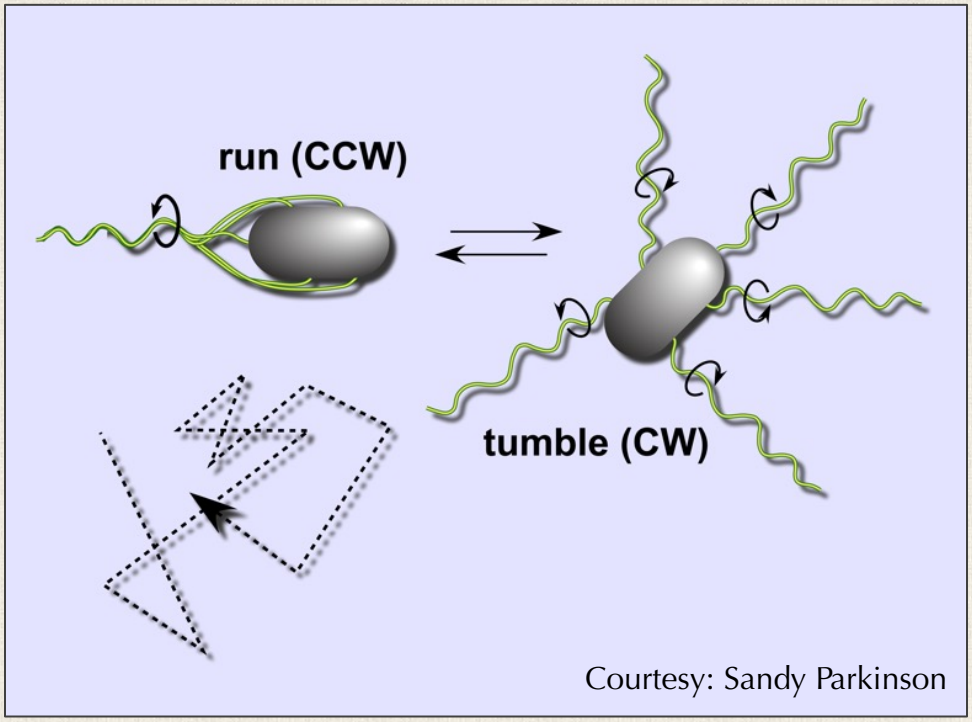
Bacterial runs and tumbles

When any flagellum rotates CW, the flagella are uncoordinated, and the bacterium stops and rotates.



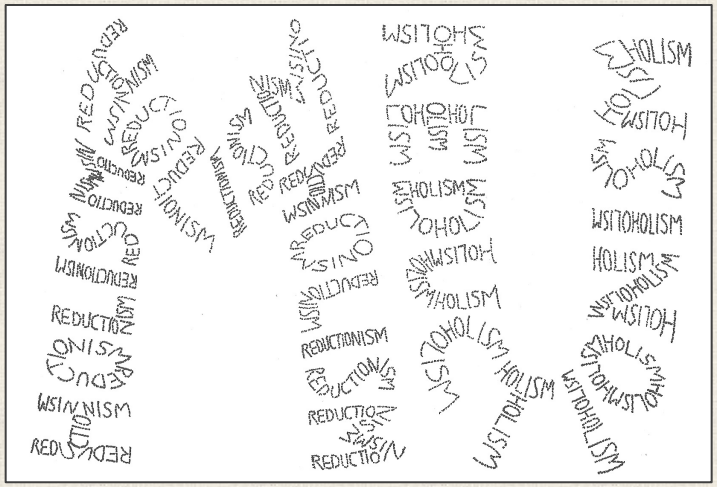
Bacterial runs and tumbles

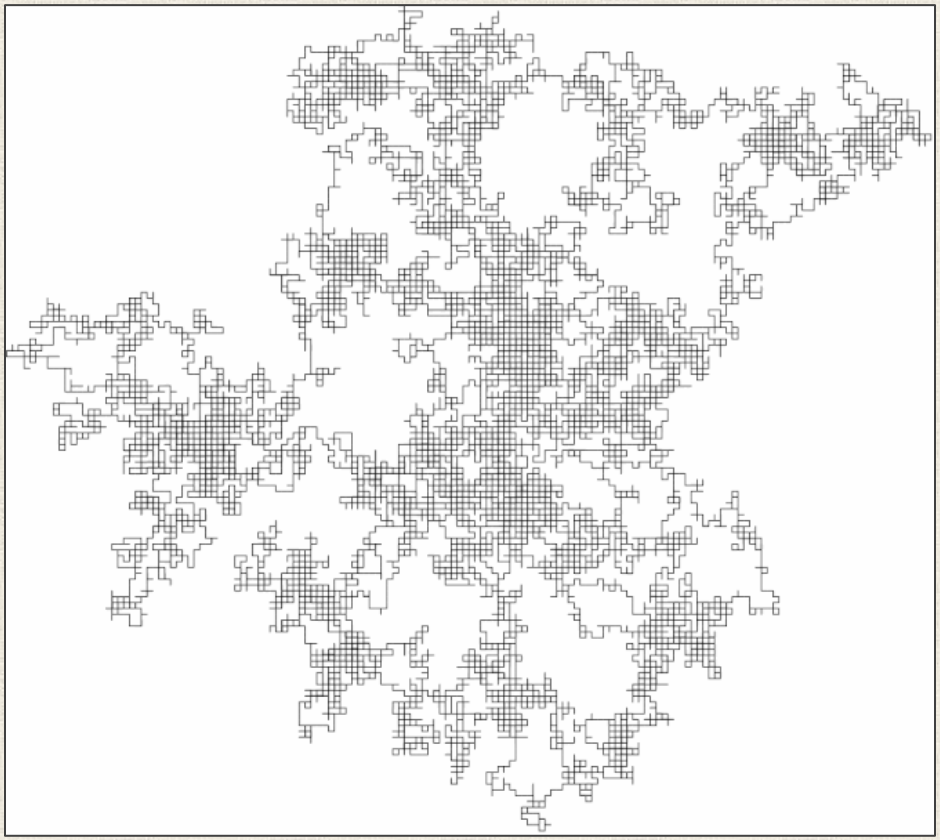
Run and tumble model: when we zoom out, *E. coli* alternates between running and tumbling in place.



E. coli"s movement looks like a random walk!

Question: what is the *molecular* basis for the random walk movement?





https://commons.wikimedia.org/wiki/File:Random_walk_25000.gif

Bacterial tumbling is constant

Key point: almost *all* bacteria, in the absence of attractant/repellent, tumble every 1-1.5 secs. Why?

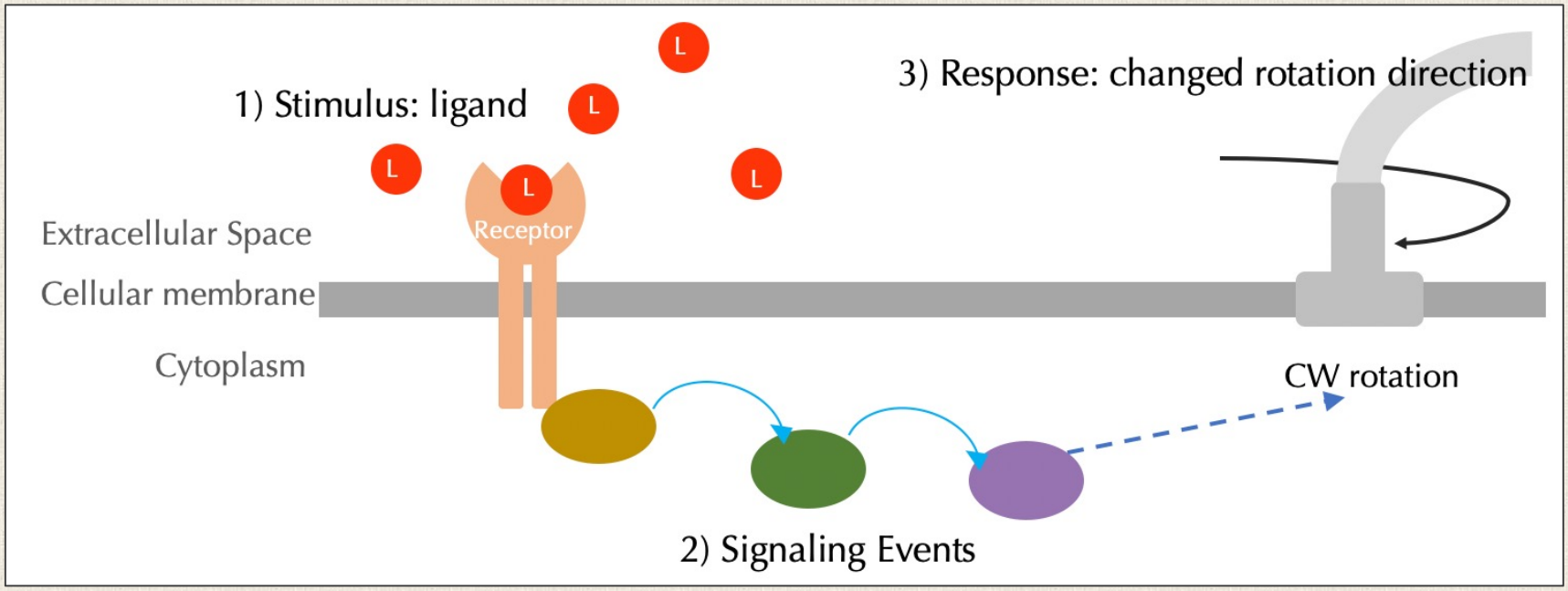
"Nothing in biology makes sense except in the light of evolution."

Theodosius Dobzhansky

SIGNALING AND LIGAND-RECEPTOR DYNAMICS

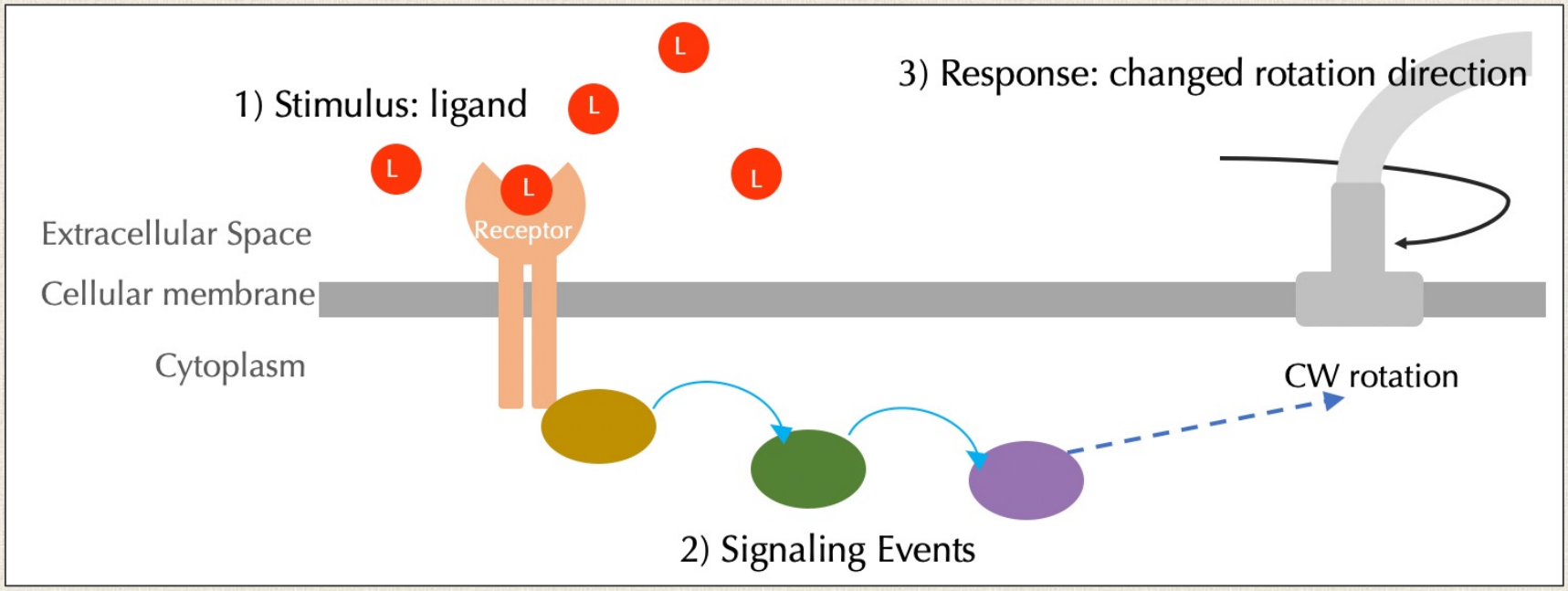
How *E. coli* detects attractants

E. coli has **receptor proteins** that detect attractants such as glucose by binding to and forming a complex with these attractant **ligands**.



How *E. coli* detects attractants

The "signal" of the binding is then "transduced" via a series of internal chemical processes that leads to a change in the flagellar rotation.

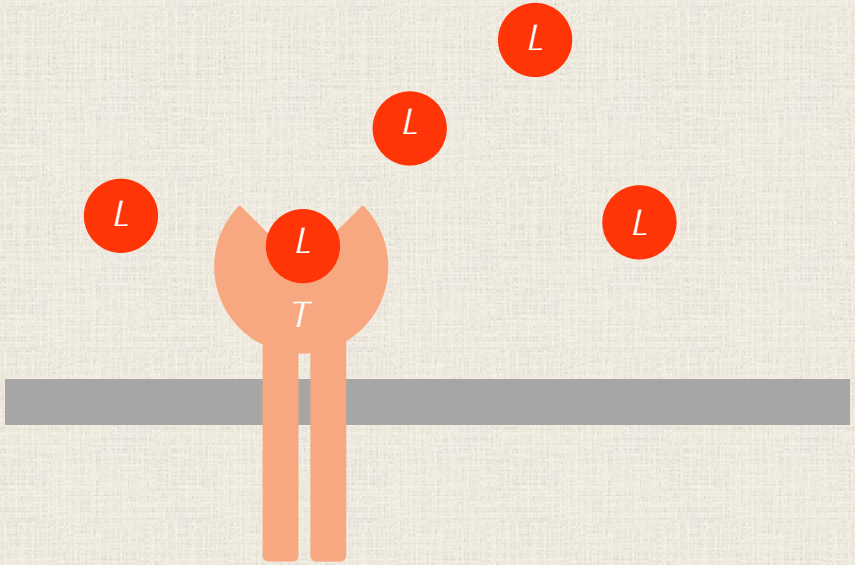


Modeling a single ligand-receptor reaction

To model ligand-receptor dynamics, we will use a **reversible reaction** in which a ligand *L* and receptor *T bind* and *dissociate* at different rates.

$$T + L \leftrightarrows LT$$

STOP: Why would a ligand and receptor need to dissociate?

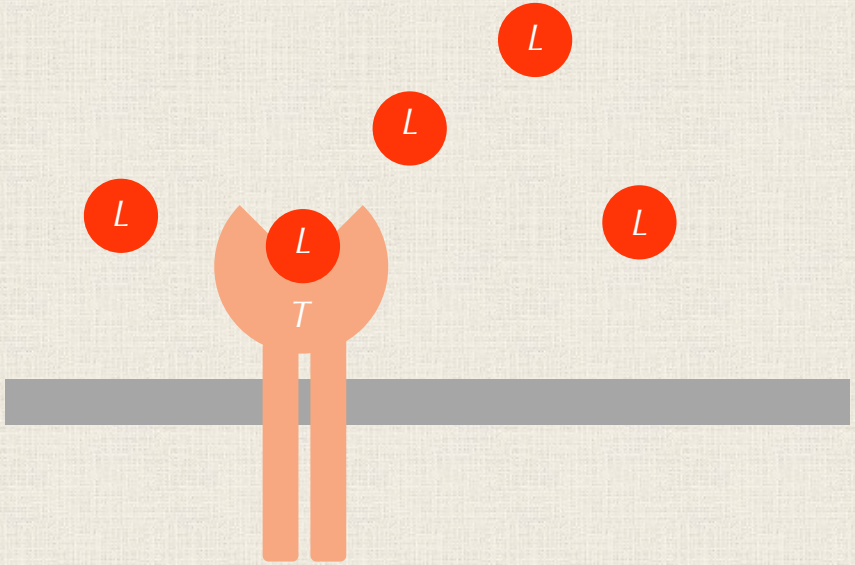


Modeling a single ligand-receptor reaction

To model ligand-receptor dynamics, we will use a **reversible reaction** in which a ligand *L* and receptor *T bind* and *dissociate* at different rates.

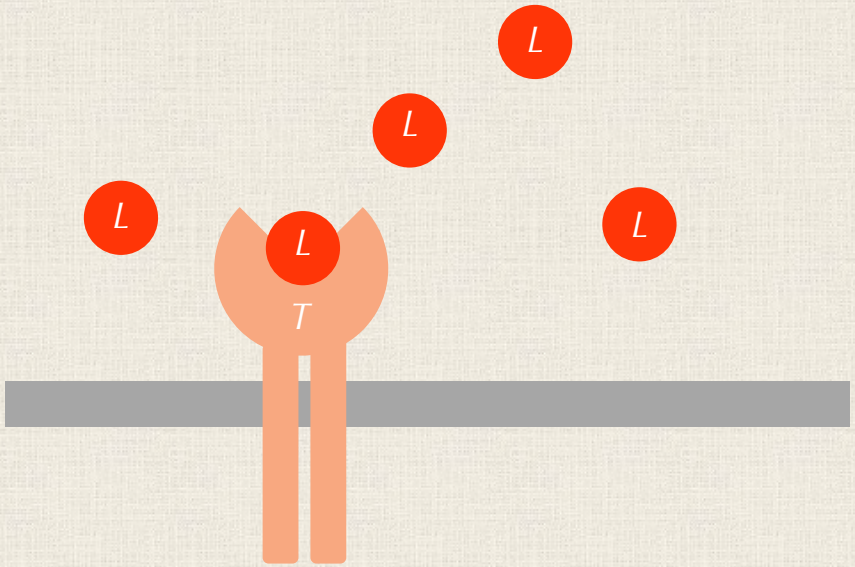
$$T + L \leftrightarrows LT$$

Answer: We don't want to detect temporary changes permanently.



Modeling a single ligand-receptor reaction

The rates k_{bind} and $k_{dissociate}$ of the forward/reverse reactions determine the **equilibrium**, or **steady state**, of the system. Let's see how…



Calculation of Equilibrium in a Reversible Ligand-Receptor Reaction

Suppose that:

- initial concentrations of L and T are l_0 and t_0;
- $[L]$, $[T]$, and $[LT]$ denote the concentrations of the three molecule types;
- the reaction rates k_{bind} and $k_{\text{dissociate}}$ are fixed.

Calculation of Equilibrium in a Reversible Ligand-Receptor Reaction

Suppose that:

- initial concentrations of L and T are l_0 and t_0;
- $[L]$, $[T]$, and $[LT]$ denote the concentrations of the three molecule types;
- the reaction rates k_{bind} and $k_{\text{dissociate}}$ are fixed.

At steady-state, binding is equal to dissociation:

$$k_{\text{bind}} \cdot [L] \cdot [T] = k_{\text{dissociate}} \cdot [LT].$$

Calculation of Equilibrium in a Reversible Ligand-Receptor Reaction

Suppose that:

- initial concentrations of L and T are l_0 and t_0;
- $[L]$, $[T]$, and $[LT]$ denote the concentrations of the three molecule types;
- the reaction rates k_{bind} and $k_{\text{dissociate}}$ are fixed.

At steady-state, binding is equal to dissociation:

$$k_{\text{bind}} \cdot [L] \cdot [T] = k_{\text{dissociate}} \cdot [LT].$$

By law of conservation of mass:

$$[L] + [LT] = l_0 \qquad [T] + [LT] = t_0.$$

Calculation of Equilibrium in a Reversible Ligand-Receptor Reaction

Solving the conservation of mass equations gives

$$[L] = l_0 - [LT] \qquad [T] = t_0 - [LT]$$

Substituting into our steady-state equation gives

$$k_{\text{bind}} \cdot (l_0 - [LT]) \cdot (t_0 - [LT]) = k_{\text{dissociate}} \cdot [LT]$$

At steady-state, binding is equal to dissociation:

$$k_{\text{bind}} \cdot [L] \cdot [T] = k_{\text{dissociate}} \cdot [LT].$$

By law of conservation of mass:

$$[L] + [LT] = l_0 \qquad [T] + [LT] = t_0.$$

Calculation of Equilibrium in a Reversible Ligand-Receptor Reaction

Solving the conservation of mass equations gives

$$[L] = l_0 - [LT] \qquad [T] = t_0 - [LT]$$

Substituting into our steady-state equation gives

$$k_{\text{bind}} \cdot (l_0 - [LT]) \cdot (t_0 - [LT]) = k_{\text{dissociate}} \cdot [LT]$$

Expansion of this equation gives

$$k_{\text{bind}} \cdot [LT]^2 - (k_{\text{bind}} \cdot l_0 + k_{\text{bind}} \cdot t_0) \cdot [LT] = k_{\text{dissociate}} \cdot [LT] + k_{\text{bind}} \cdot l_0 \cdot t_0$$

Calculation of Equilibrium in a Reversible Ligand-Receptor Reaction

Solving the conservation of mass equations gives

$$[L] = l_0 - [LT] \qquad [T] = t_0 - [LT]$$

Substituting into our steady-state equation gives

$$k_{\text{bind}} \cdot (l_0 - [LT]) \cdot (t_0 - [LT]) = k_{\text{dissociate}} \cdot [LT]$$

Expansion of this equation gives

$$k_{\text{bind}} \cdot [LT]^2 - (k_{\text{bind}} \cdot l_0 + k_{\text{bind}} \cdot t_0) \cdot [LT] = k_{\text{dissociate}} \cdot [LT] - k_{\text{bind}} \cdot l_0 \cdot t_0$$

Calculation of Equilibrium in a Reversible Ligand-Receptor Reaction

Solving the conservation of mass equations gives

$$[L] = l_0 - [LT] \qquad [T] = t_0 - [LT]$$

Substituting into our steady-state equation gives

$$k_{\text{bind}} \cdot (l_0 - [LT]) \cdot (t_0 - [LT]) = k_{\text{dissociate}} \cdot [LT]$$

Expansion of this equation gives

$$k_{\text{bind}} \cdot [LT]^2 - (k_{\text{bind}} \cdot l_0 + k_{\text{bind}} \cdot t_0) \cdot [LT] = k_{\text{dissociate}} \cdot [LT] - k_{\text{bind}} \cdot l_0 \cdot t_0$$

Subtract the right side from both sides:

$$k_{\text{bind}} \cdot [LT]^2 - (k_{\text{bind}} \cdot l_0 + k_{\text{bind}} \cdot t_0 + k_{\text{dissociate}}) \cdot [LT] + k_{\text{bind}} \cdot l_0 \cdot t_0 = 0$$

This is just a quadratic equation in $[LT]$!

Let's Look at a Real Example

$l_0 = 10{,}000$; $t_0 = 7{,}000$; $lt_0 = 0$

$k_{\text{bind}} = 0.0146((\text{molecules}/\mu\text{m}^3)^{-1})\text{s}^{-1}$; $k_{\text{dissociate}} = 35\text{s}^{-1}$

Let's Look at a Real Example

$l_0 = 10{,}000$; $t_0 = 7{,}000$; $lt_0 = 0$

$k_{\text{bind}} = 0.0146((\text{molecules}/\mu\text{m}^3)^{-1})\text{s}^{-1}$; $k_{\text{dissociate}} = 35\text{s}^{-1}$

Our previous quadratic equation was:

$$k_{\text{bind}} \cdot [LT]^2 - (k_{\text{bind}} \cdot l_0 + k_{\text{bind}} \cdot t_0 + k_{\text{dissociate}}) \cdot [LT] + k_{\text{bind}} \cdot l_0 \cdot t_0 = 0$$

Let's Look at a Real Example

$l_0 = 10{,}000$; $t_0 = 7{,}000$; $lt_0 = 0$

$k_{\text{bind}} = 0.0146((\text{molecules}/\mu\text{m}^3)^{-1})\text{s}^{-1}$; $k_{\text{dissociate}} = 35\text{s}^{-1}$

Our previous quadratic equation was:

$$k_{\text{bind}} \cdot [LT]^2 - (k_{\text{bind}} \cdot l_0 + k_{\text{bind}} \cdot t_0 + k_{\text{dissociate}}) \cdot [LT] + k_{\text{bind}} \cdot l_0 \cdot t_0 = 0$$

Solving this for $[LT]$ gives 😮 that

$$[LT] = 4{,}793 \text{ molecules}/\ \mu\text{m}^3$$

Let's Look at a Real Example

$l_0 = 10{,}000$; $t_0 = 7{,}000$; $lt_0 = 0$

$k_{\text{bind}} = 0.0146((\text{molecules}/\mu\text{m}^3)^{-1})\text{s}^{-1}$; $k_{\text{dissociate}} = 35\text{s}^{-1}$

Our previous quadratic equation was:

$$k_{\text{bind}} \cdot [LT]^2 - (k_{\text{bind}} \cdot l_0 + k_{\text{bind}} \cdot t_0 + k_{\text{dissociate}}) \cdot [LT] + k_{\text{bind}} \cdot l_0 \cdot t_0 = 0$$

Solving this for $[LT]$ gives 😮‍💨 that

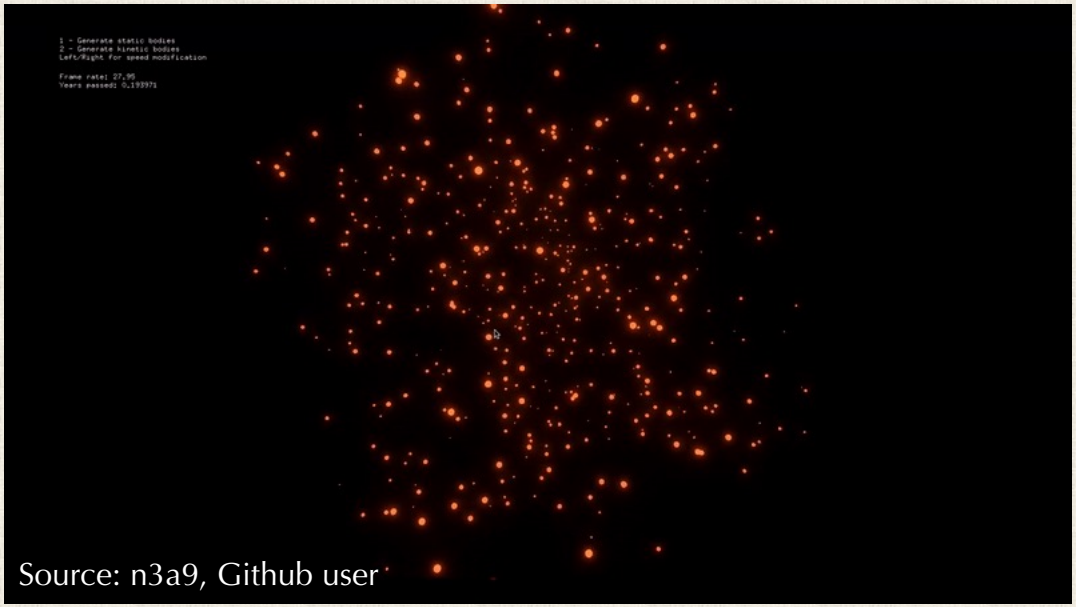
$$[LT] = 4{,}793 \text{ molecules}/\ \mu\text{m}^3$$

By law of conservation of mass,

$$[L] = l_0 - [LT] = 5{,}207 \qquad [T] = t_0 - [LT] = 2{,}207$$

Scaling this will be a disaster

Key point: This is not too bad for one reversible equation, but real biological systems have many reactions, and this will not scale. As with the *n*-body problem in physics, we need a simulation.



Source: n3a9, Github user

The Need for a "Particle-Free" Model

The *E. coli* cell is so small that we will assume that the concentration of any particle in its immediate surroundings is **well-mixed** (i.e., uniform).

The Need for a "Particle-Free" Model

The *E. coli* cell is so small that we will assume that the concentration of any particle in its immediate surroundings is **well-mixed** (i.e., uniform).

Our model of chemotaxis will have many particles and reactions that depend on each other, and so a "particle-free" model that does not track the diffusion of individual particles will greatly increase efficiency.

STOCHASTIC SIMULATION OF CHEMICAL REACTIONS WITH THE GILLESPIE ALGORITHM

The Poisson Distribution

Say that you own a store and have noticed that on average, there are λ customers entering your store in a single hour. Let X be a random variable denoting the number of customers that enter the store in the next hour.

The Poisson Distribution

Say that you own a store and have noticed that on average, there are λ customers entering your store in a single hour. Let X be a random variable denoting the number of customers that enter the store in the next hour.

X follows a **Poisson distribution**; it can be shown that the probability that exactly n customers arrive in the next hour is

$$\Pr(X = n) = \frac{\lambda^n e^{-\lambda}}{n!}$$

The Poisson Distribution

Furthermore, the probability of observing exactly n customers in t hours is

$$\frac{(\lambda t)^n e^{-\lambda t}}{n!}$$

X follows a **Poisson distribution**; it can be shown that the probability that exactly n customers arrive in the next hour is

$$\Pr(X = n) = \frac{\lambda^n e^{-\lambda}}{n!}$$

From Poisson to Exponential

If we let T be the random variable corresponding to the wait time on the next customer, then the probability of waiting at least t hours is the probability of seeing zero customers in t hours:

$$\Pr(T > t) = \Pr(X = 0) = \frac{(\lambda t)^0 e^{-\lambda t}}{0!} = e^{-\lambda t}$$

From Poisson to Exponential

If we let T be the random variable corresponding to the wait time on the next customer, then the probability of waiting at least t hours is the probability of seeing zero customers in t hours:

$$\Pr(T > t) = \Pr(X = 0) = \frac{(\lambda t)^0 e^{-\lambda t}}{0!} = e^{-\lambda t}$$

That is, $\Pr(T > t)$ decays exponentially as t increases; thus, random variable T follows an **exponential distribution**. (Mean wait time: $1/\lambda$).

From Poisson to Exponential

If we let T be the random variable corresponding to the wait time on the next customer, then the probability of waiting at least t hours is the probability of seeing zero customers in t hours:

$$\Pr(T > t) = \Pr(X = 0) = \frac{(\lambda t)^0 e^{-\lambda t}}{0!} = e^{-\lambda t}$$

STOP: What is the probability $\Pr(T < t)$?

From Poisson to Exponential

If we let T be the random variable corresponding to the wait time on the next customer, then the probability of waiting at least t hours is the probability of seeing zero customers in t hours:

$$\Pr(T > t) = \Pr(X = 0) = \frac{(\lambda t)^0 e^{-\lambda t}}{0!} = e^{-\lambda t}$$

STOP: What is the probability $\Pr(T < t)$?

Answer: $1 - e^{-\lambda t}$.

An overview of the Gillespie algorithm

Given a well-mixed environment and a reaction taking place at some known average rate, we would like to know how long we expect to wait before this reaction occurs somewhere in the environment.

An overview of the Gillespie algorithm

Given a well-mixed environment and a reaction taking place at some known average rate, we would like to know how long we expect to wait before this reaction occurs somewhere in the environment.

STOP: Remind you of anything?

An overview of the Gillespie algorithm

Given a well-mixed environment and a reaction taking place at some known average rate, we would like to know how long we expect to wait before this reaction occurs somewhere in the environment.

STOP: Remind you of anything?

Answer: We will model each of our reactions using an exponential distribution!

An overview of the Gillespie algorithm

Given a well-mixed environment and a reaction taking place at some known average rate, we would like to know how long we expect to wait before this reaction occurs somewhere in the environment.

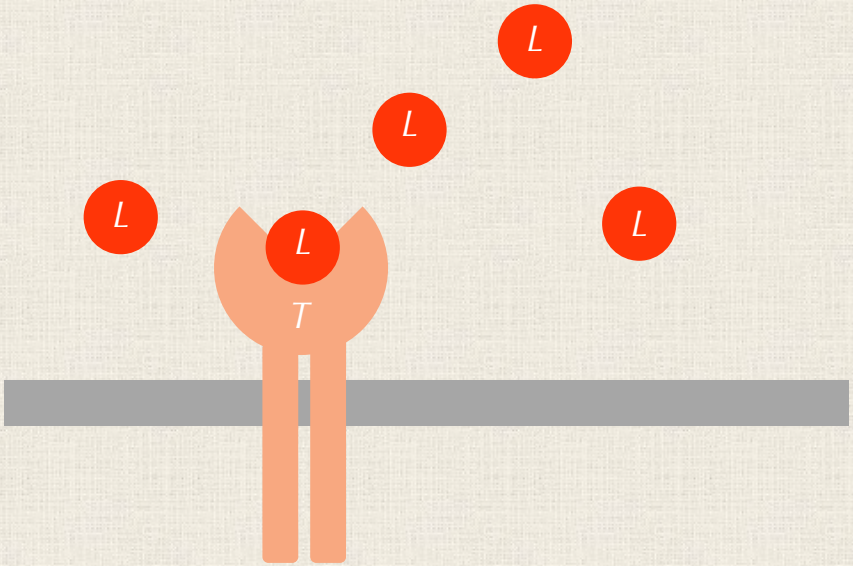
STOP: Remind you of anything?

Answer: We will model each of our reactions using an exponential distribution!

This idea is the engine of **Gillespie's stochastic simulation algorithm (SSA)**.

Modeling a single ligand-receptor reaction with Gillespie

Recall that the rate of $T + L \rightarrow LT$ is k_{bind} and the rate of $LT \rightarrow T + L$ is $k_{\text{dissociate}}$.



Modeling a single ligand-receptor reaction with Gillespie

Recall that the rate of $T + L \rightarrow LT$ is k_{bind} and the rate of $LT \rightarrow T + L$ is $k_{\text{dissociate}}$.

Repeat the following steps for the entire simulation.

1. Define $r_{\text{bind}} = k_{\text{bind}} \cdot [L] \cdot [T]$ and $r_{\text{dissociate}} = k_{\text{dissociate}} \cdot [LT]$.

Modeling a single ligand-receptor reaction with Gillespie

Recall that the rate of $T + L \rightarrow LT$ is k_{bind} and the rate of $LT \rightarrow T + L$ is $k_{\text{dissociate}}$.

Repeat the following steps for the entire simulation.

1. Define $r_{\text{bind}} = k_{\text{bind}} \cdot [L] \cdot [T]$ and $r_{\text{dissociate}} = k_{\text{dissociate}} \cdot [LT]$.
2. Pick a wait time according to an exponential distribution with $\lambda = r_{\text{bind}} + r_{\text{dissociate}}$.

Modeling a single ligand-receptor reaction with Gillespie

Recall that the rate of $T + L \rightarrow LT$ is k_{bind} and the rate of $LT \rightarrow T + L$ is $k_{\text{dissociate}}$.

Repeat the following steps for the entire simulation.

1. Define $r_{\text{bind}} = k_{\text{bind}} \cdot [L] \cdot [T]$ and $r_{\text{dissociate}} = k_{\text{dissociate}} \cdot [LT]$.
2. Pick a wait time according to an exponential distribution with $\lambda = r_{\text{bind}} + r_{\text{dissociate}}$.
3. The probability that the reaction is the forward reaction is $\Pr(L + T \rightarrow LT) = r_{\text{bind}}/(r_{\text{bind}} + r_{\text{dissociate}})$.

Modeling a single ligand-receptor reaction with Gillespie

STOP: What is the probability that the reaction is the reverse reaction?

Repeat the following steps for the entire simulation.

1. Define $r_{\text{bind}} = k_{\text{bind}} \cdot [L] \cdot [T]$ and $r_{\text{dissociate}} = k_{\text{dissociate}} \cdot [LT]$.
2. Pick a wait time according to an exponential distribution with $\lambda = r_{\text{bind}} + r_{\text{dissociate}}$.
3. The probability that the reaction is the forward reaction is $\Pr(L + T \rightarrow LT) = r_{\text{bind}}/(r_{\text{bind}} + r_{\text{dissociate}})$.

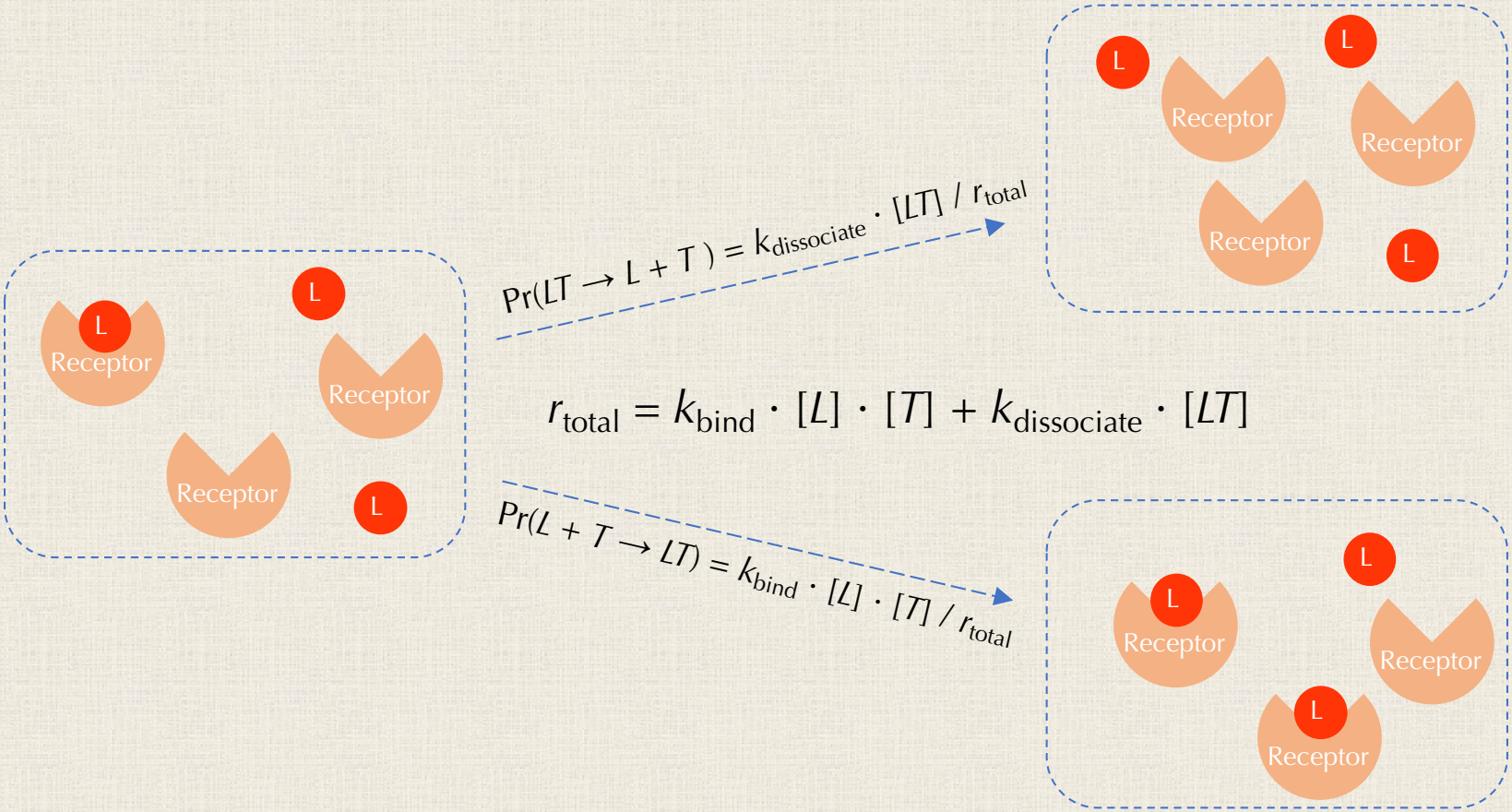
Modeling a single ligand-receptor reaction with Gillespie

Answer: $\Pr(LT \rightarrow L + T) = r_{\text{dissociate}}/(r_{\text{bind}} + r_{\text{dissociate}})$.

Repeat the following steps for the entire simulation.

1. Define $r_{\text{bind}} = k_{\text{bind}} \cdot [L] \cdot [T]$ and $r_{\text{dissociate}} = k_{\text{dissociate}} \cdot [LT]$.
2. Pick a wait time according to an exponential distribution with $\lambda = r_{\text{bind}} + r_{\text{dissociate}}$.
3. The probability that the reaction is the forward reaction is $\Pr(L + T \rightarrow LT) = r_{\text{bind}}/(r_{\text{bind}} + r_{\text{dissociate}})$.

An illustration of Gillespie



Can Gillespie Replicate Our Example?

$l_0 = 10{,}000$; $t_0 = 7{,}000$; $lt_0 = 0$

$k_{\text{bind}} = 0.0146((\text{molecules}/\mu\text{m}^3)^{-1})\text{s}^{-1}$; $k_{\text{dissociate}} = 35\text{s}^{-1}$

Our previous quadratic equation was:

$$k_{\text{bind}} \cdot [LT]^2 - (k_{\text{bind}} \cdot l_0 + k_{\text{bind}} \cdot t_0 + k_{\text{dissociate}}) \cdot [LT] + k_{\text{bind}} \cdot l_0 \cdot t_0 = 0$$

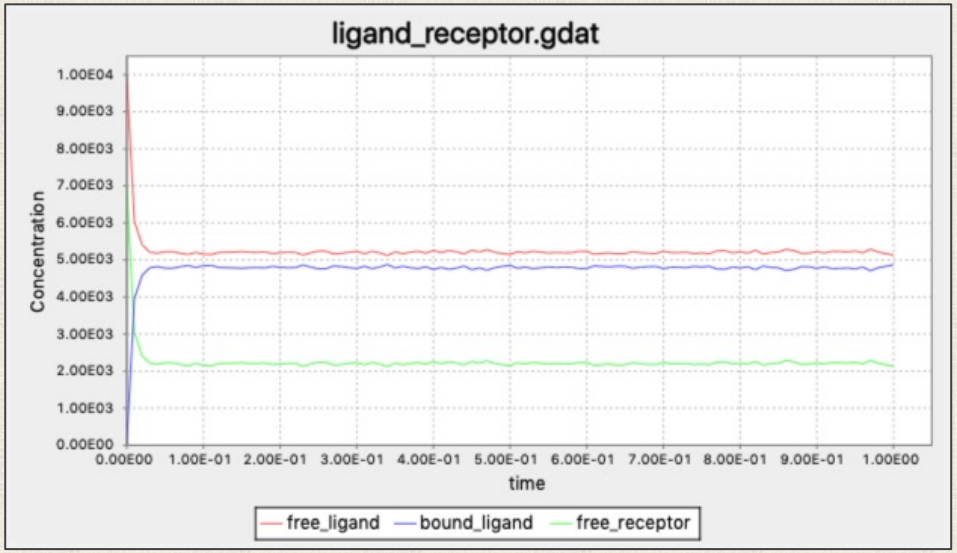
Solving this for $[LT]$ gives 🤭 that

$$[LT] = 4{,}793 \text{ molecules}/\ \mu\text{m}^3$$

By law of conservation of mass,

$$[L] = l_0 - [LT] = 5{,}207 \qquad [T] = t_0 - [LT] = 2{,}207$$

It Can! ☺



Solving this for $[LT]$ gives 🤭 that

$$[LT] = 4{,}793 \text{ molecules}/\ \mu m^3$$

By law of conservation of mass,

$$[L] = l_0 - [LT] = 5{,}207 \qquad [T] = t_0 - [LT] = 2{,}207$$

A note on Gillespie with multiple different reactions

STOP: The question is how to generalize this idea to n reactions, having rates k_1, k_2, …, k_n. Ideas?

Repeat the following steps for the entire simulation.

1. Define $r_{\text{bind}} = k_{\text{bind}} \cdot [L] \cdot [T]$ and $r_{\text{dissociate}} = k_{\text{dissociate}} \cdot [LT]$.
2. Pick a wait time according to an exponential distribution with $\lambda = r_{\text{bind}} + r_{\text{dissociate}}$.
3. The probability that the reaction is the forward reaction is $\Pr(L + T \rightarrow LT) = r_{\text{bind}}/(r_{\text{bind}} + r_{\text{dissociate}})$.

A note on Gillespie with multiple different reactions

Answer: It's easier than you might imagine! We just extend the definitions and sum over n terms.

Repeat the following steps for the entire simulation.

1. For each i, define $r_i = k_i \cdot$ (product of reactant concentrations in equation i).
2. Pick a wait time according to an exponential distribution with $\lambda = r_1 + r_2 + \ldots + r_n$.
3. The probability that the reaction is the i-th reaction is $\Pr(L + T \rightarrow LT) = r_i / \lambda$.

Again, great ideas don't have to be complicated

Exact stochastic simulation of coupled chemical reactions
DT Gillespie - The journal of physical **chemistry**, 1977 - ACS Publications
There are two formalisms for mathematically describing the time behavior of a spatially homogeneous chemical system: The deterministic approach regards thetime evolution as a continuous, wholly predictable process which is governedby a set of coupled, ordinary …
☆ 99 Cited by 10346 Related articles All 47 versions

Repeat the following steps for the entire simulation.

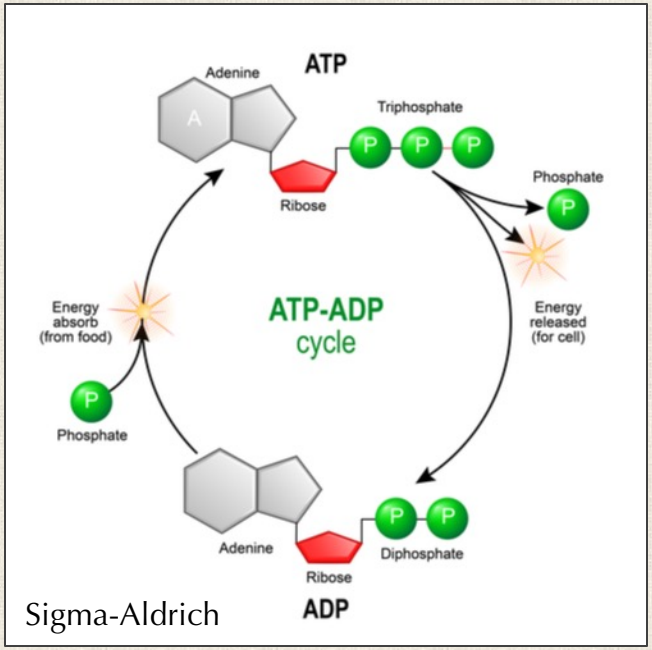
1. For each i, define $r_i = k_i \cdot$ (product of reactant concentrations in equation i).
2. Pick a wait time according to an exponential distribution with $\lambda = r_1 + r_2 + \ldots + r_n$.
3. The probability that the reaction is the i-th reaction is $\Pr(L + T \longrightarrow LT) = r_i / \lambda$.

BUILDING AN ACCURATE MODEL OF CHEMOTAXIS WITH RULE-BASED MODELING

A "phosphorylation cascade" is the engine of signal transduction

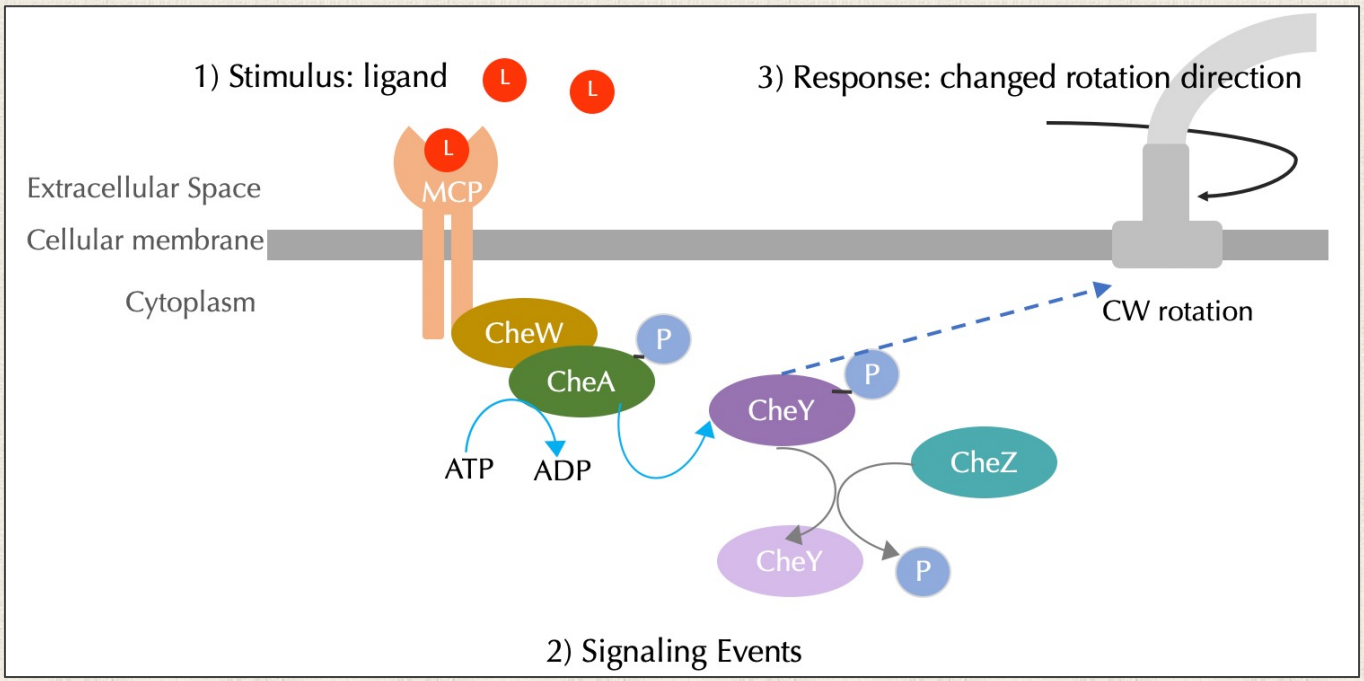
In a **phosphorylation** event, a phosphoryl group (PO_3^-) is attached to an organic molecule.

Phosphoryl can be broken off an adenosine triphosphate (ATP) molecule, or exchanged as part of *dephosphorylation* of a phosphorylated molecule.



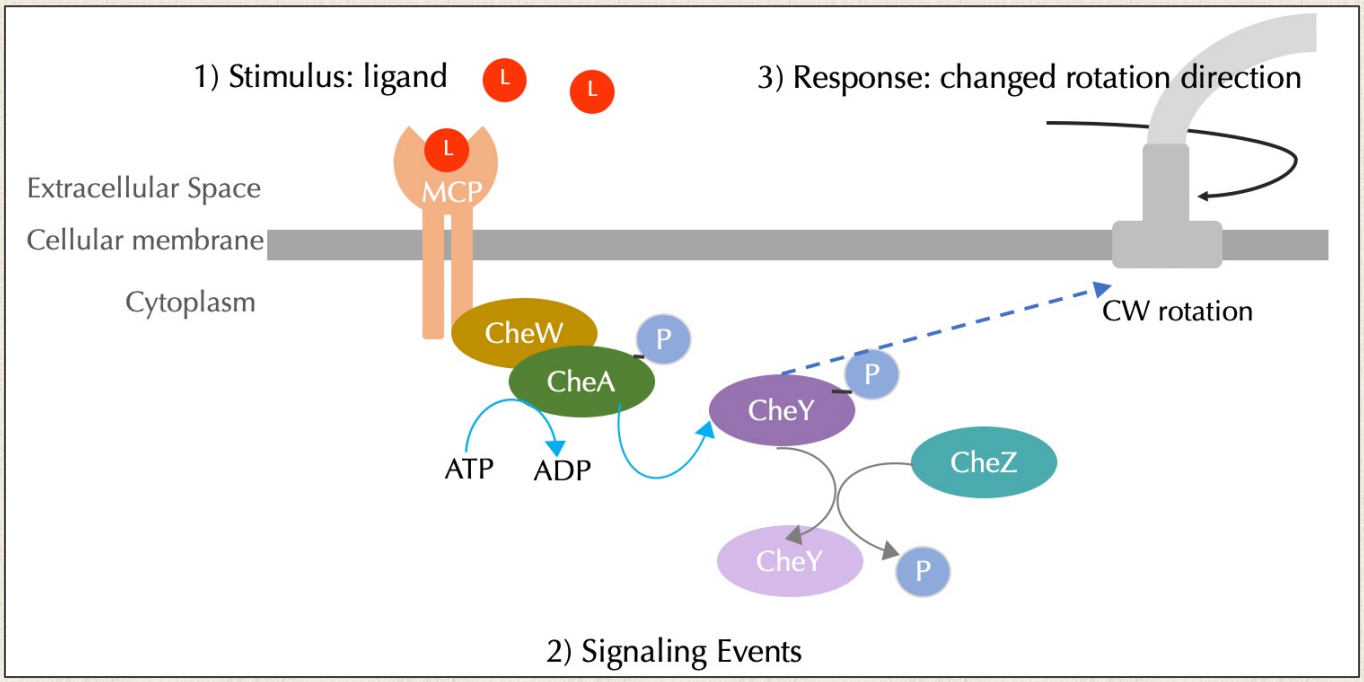
Explaining the molecular basis for signaling

Receptors form a complex on the inside of the cell with CheA and CheW proteins, which is more stable without ligand binding. Remember this fact!



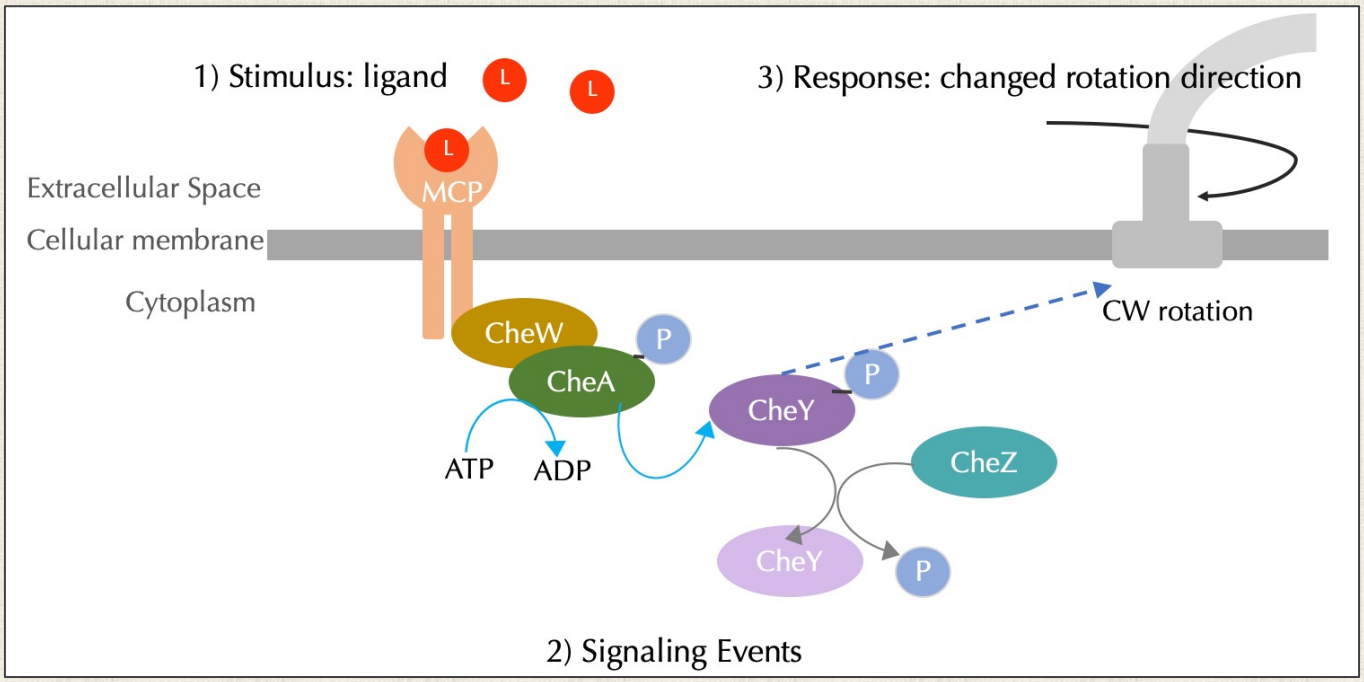
Explaining the molecular basis for signaling

When bound, CheA **autophosphorylates**, adding a phosphoryl group to itself – not a strange concept after autoregulation ☺



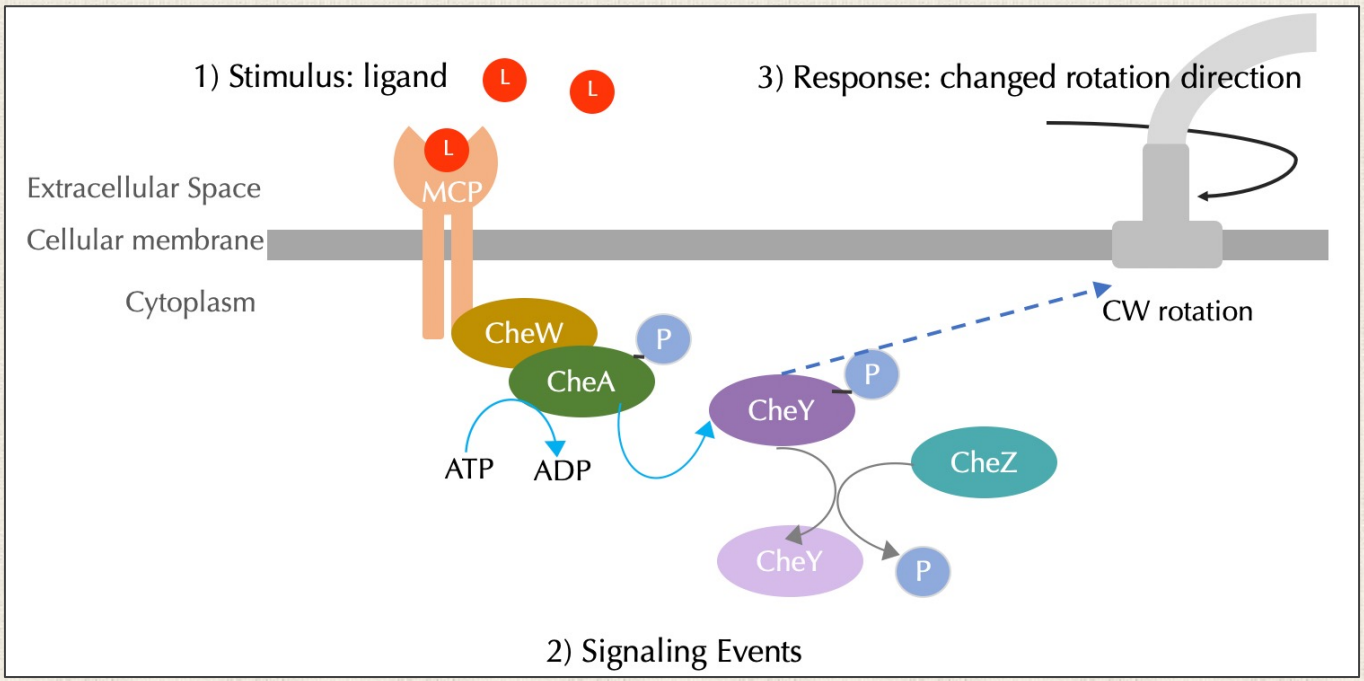
Explaining the molecular basis for signaling

When phosphorylated, CheA can pass on the phosphoryl group to a molecule called CheY.



Explaining the molecular basis for signaling

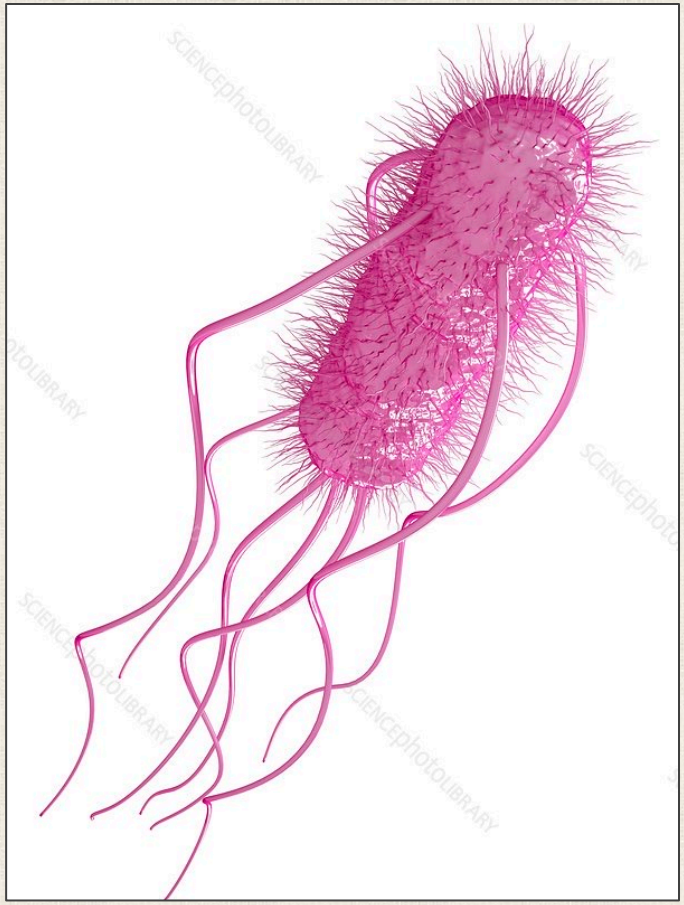
When phosphorylated CheY interacts with the **flagellar motor switch** protein complex on the flagellum, it changes rotation from CCW to CW.



Bacterial runs and tumbles

Recall: when the flagella are all rotating CCW, they form a bundle and propel the cell forward at 20 μm/s.

STOP: What happens when the flagellum rotates CW instead?



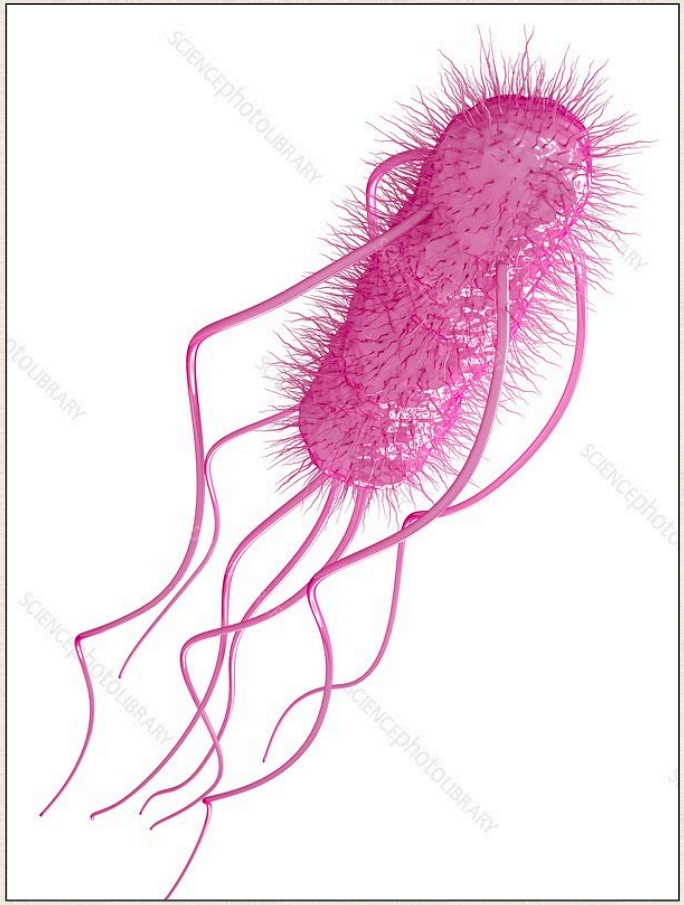
https://www.sciencephoto.com/media/659604/view/e-coli-bacterium-illustration

Bacterial runs and tumbles

Recall: when the flagella are all rotating CCW, they form a bundle and propel the cell forward at 20 μm/s.

STOP: What happens when the flagellum rotates CW instead?

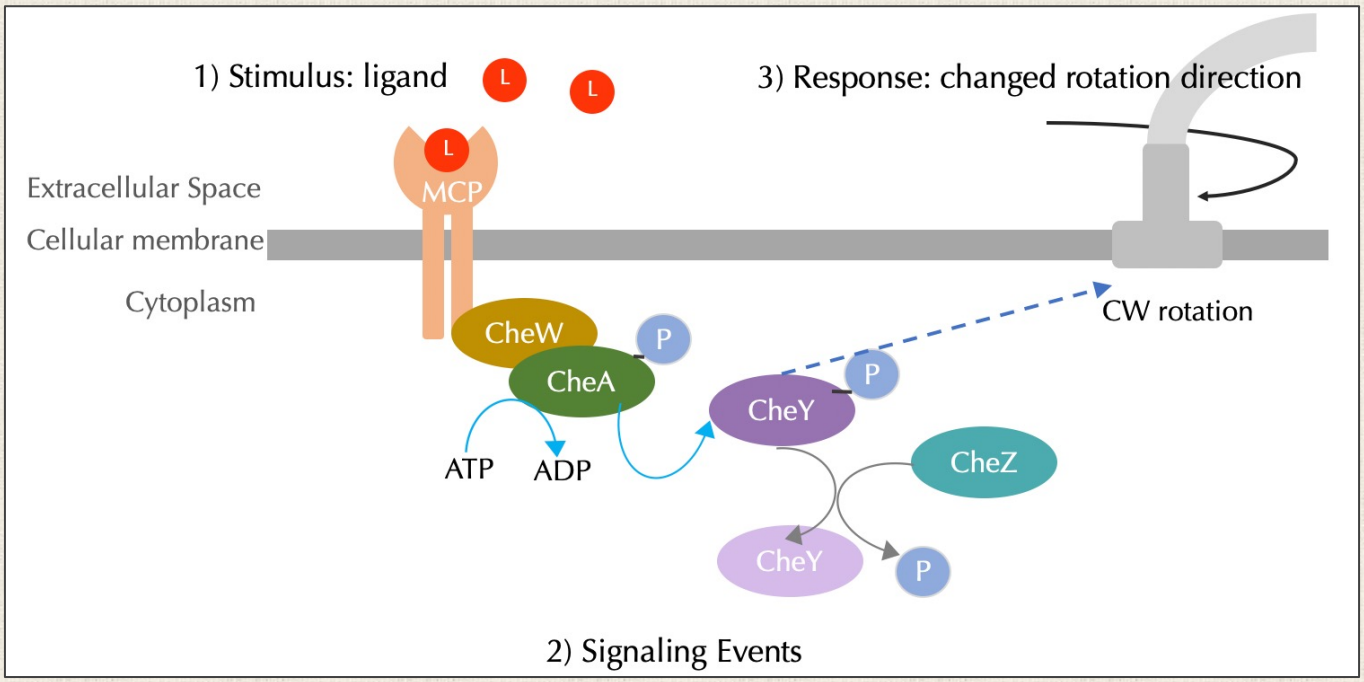
Answer: Tumble!



https://www.sciencephoto.com/media/659604/view/e-coli-bacterium-illustration

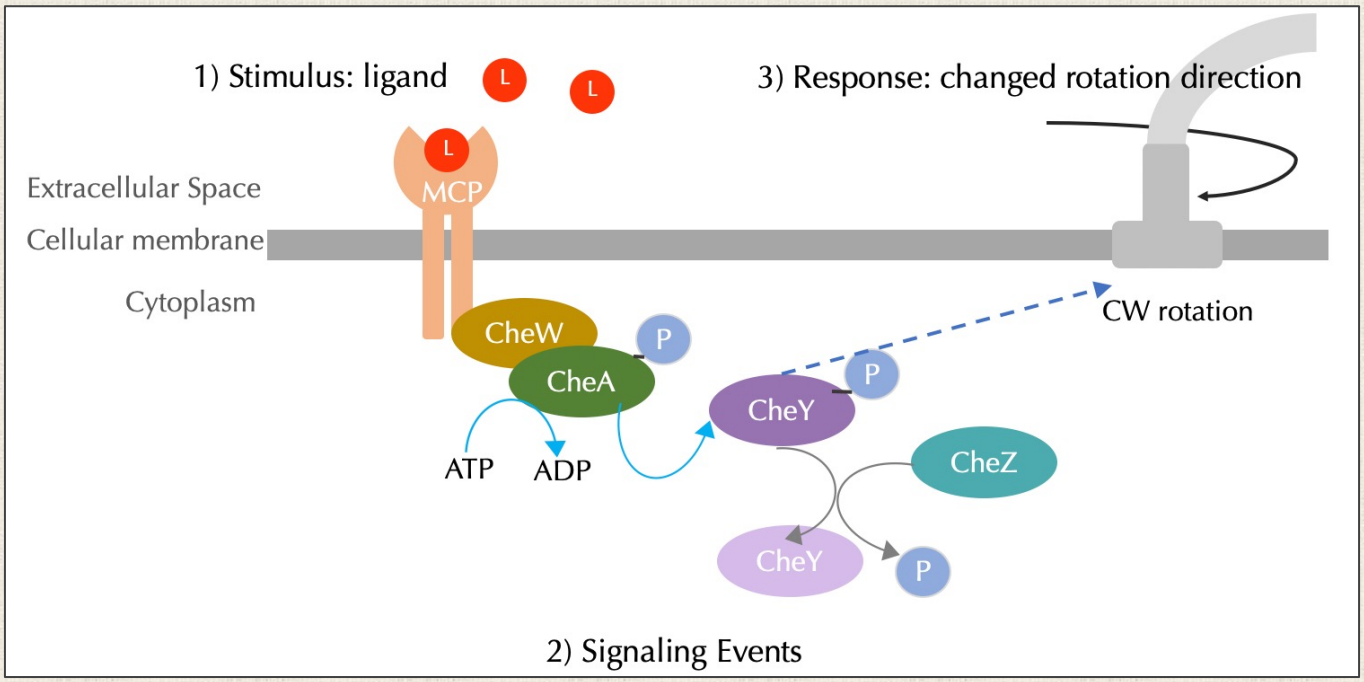
Explaining the molecular basis for signaling

If a ligand is detected, then the cell needs to decrease CheY concentration to reduce tumbling, dephosphorylating CheY with the **CheZ** enzyme.



Explaining the molecular basis for signaling

Remember: this whole process is *more likely* when ligand is *not* present. So, less ligand means more tumbling, and more ligand means more running.



The model is starting to get pretty unwieldy and won't fit on one slide …

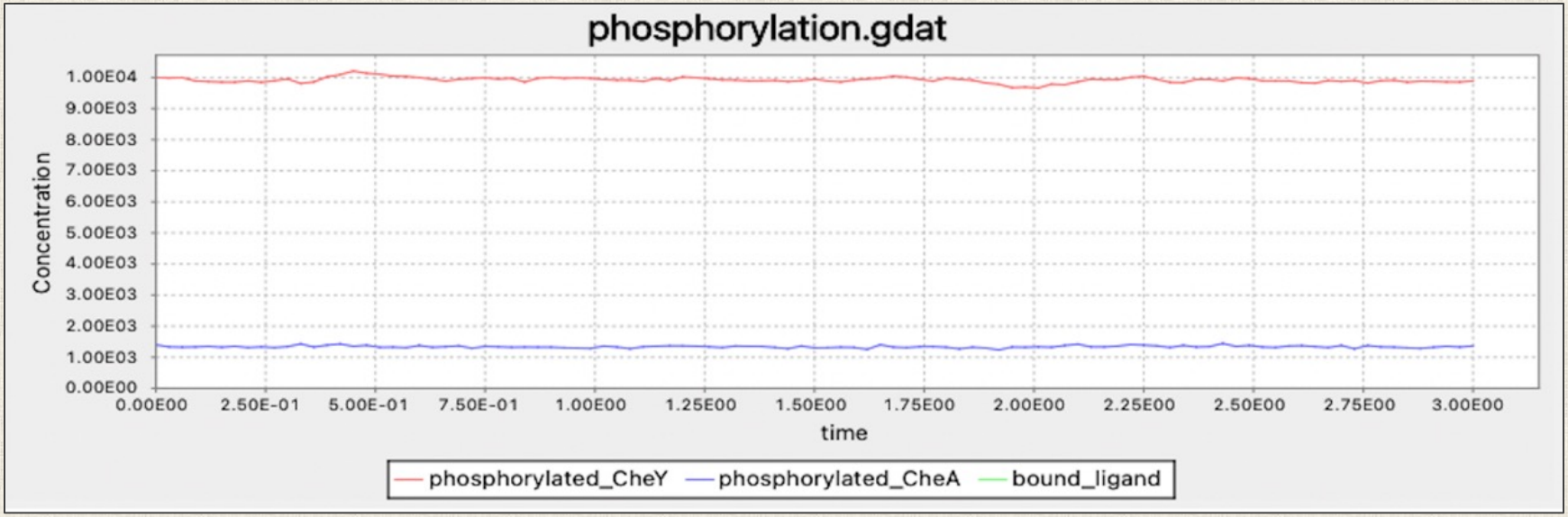
We need three particle types corresponding to MCP molecules, ligands, and bound complexes.

A bound complex molecule binds with CheA and CheW and can be either phosphorylated or unphosphorylated.

And CheY can be phosphorylated or unphosphorylated too …

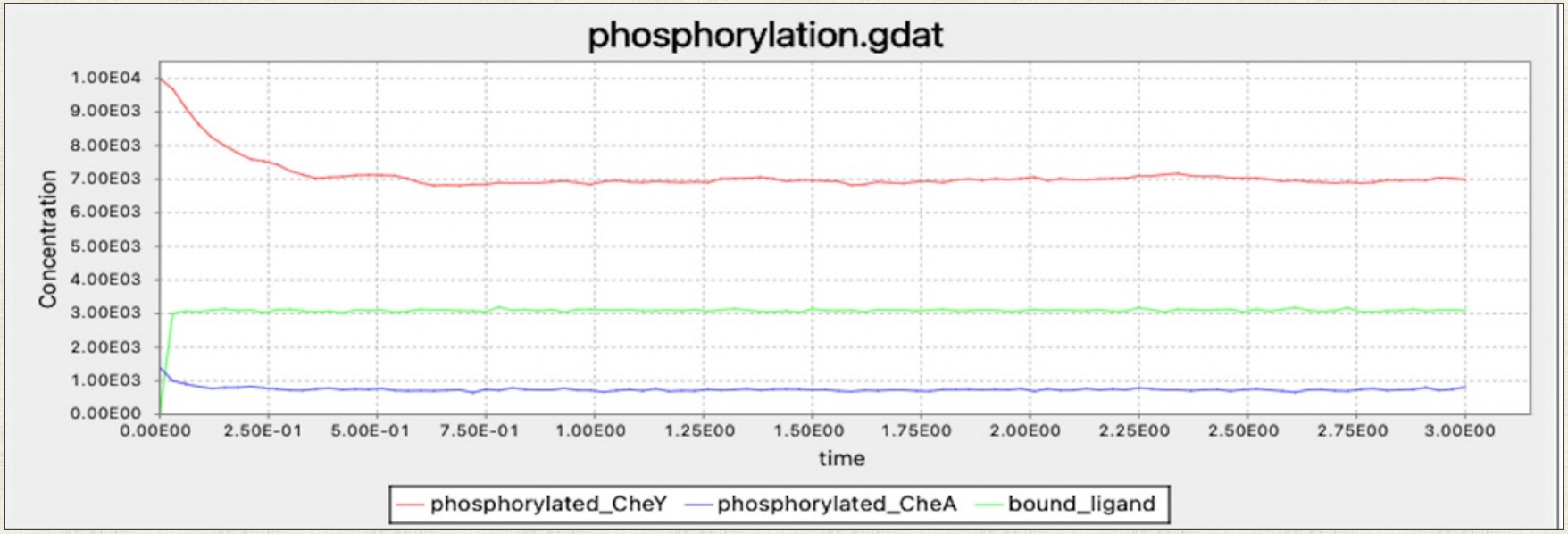
Changing ligand concentrations leads to a change in molecular concentrations

The figure below plots phosphorylated CheA and CheY at equilibrium in the absence of ligand.



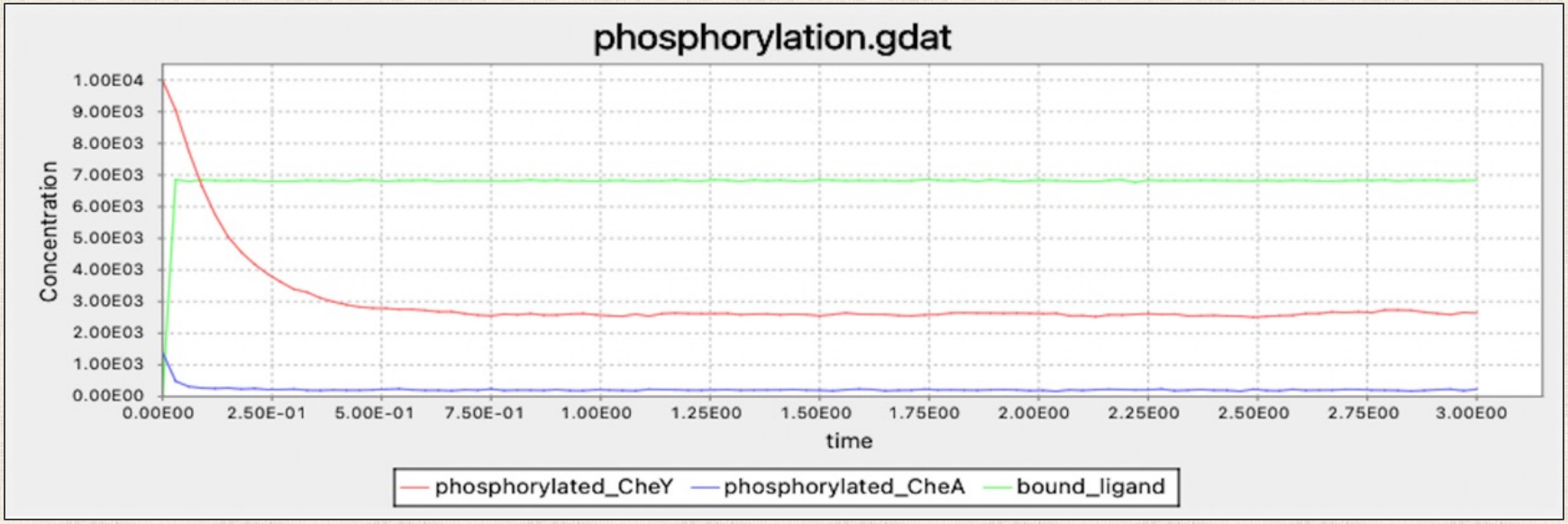
Changing ligand concentrations leads to a change in molecular concentrations

The addition of 5,000 attractant ligand molecules increases bound receptors, leading to less CheA autophosphorylation, and less phosphorylated CheY.



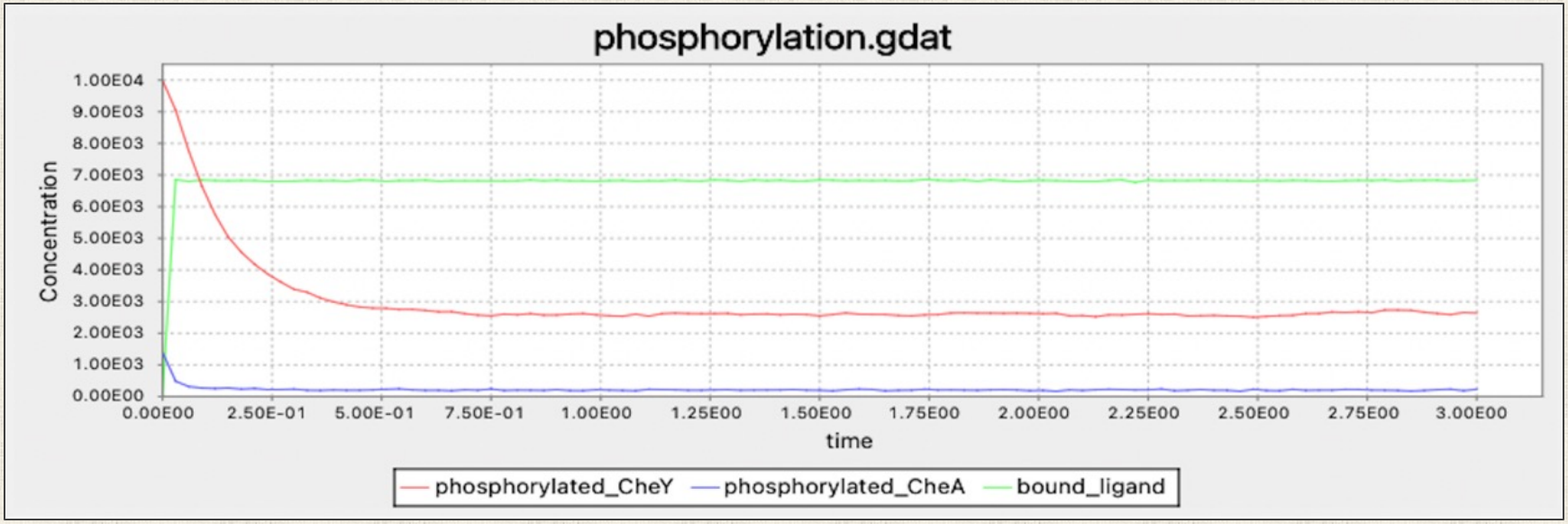
Changing ligand concentrations leads to a change in molecular concentrations

If we instead add 100,000 attractant molecules, then we see an even more drastic decrease in phosphorylated CheA and CheY.



Changing ligand concentrations leads to a change in molecular concentrations

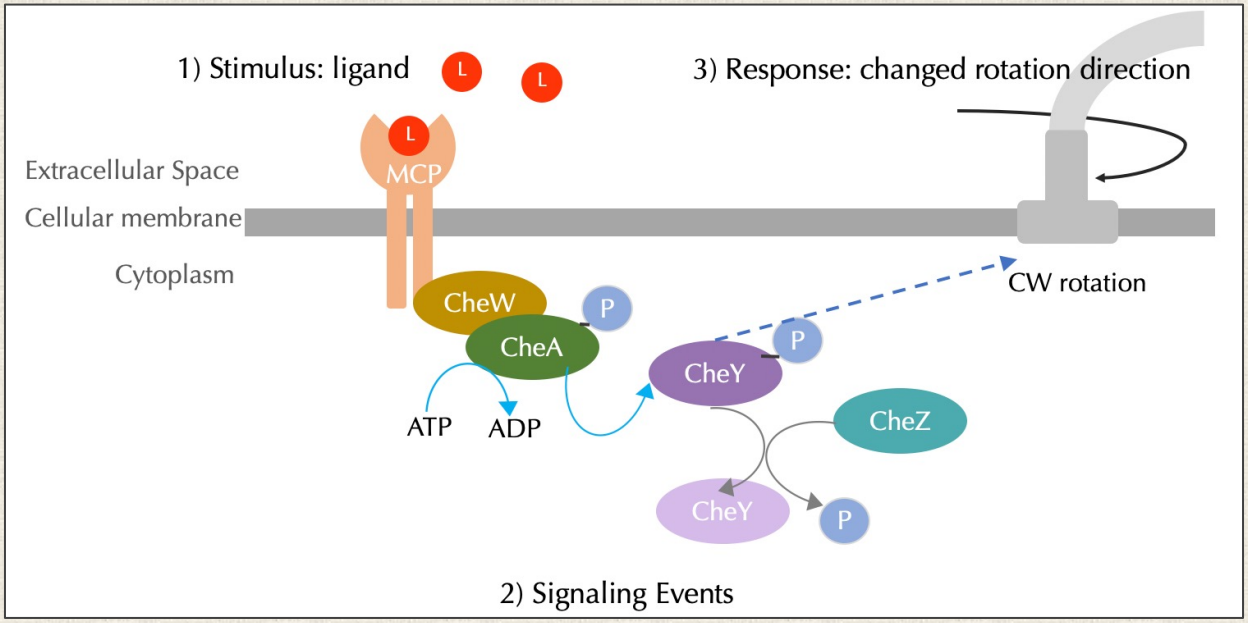
So far, none of this is surprising, other than how fast the cell can react. *But what we have shown is just part of the story …*



METHYLATION HELPS A BACTERIUM ADAPT TO DIFFERING CONCENTRATIONS

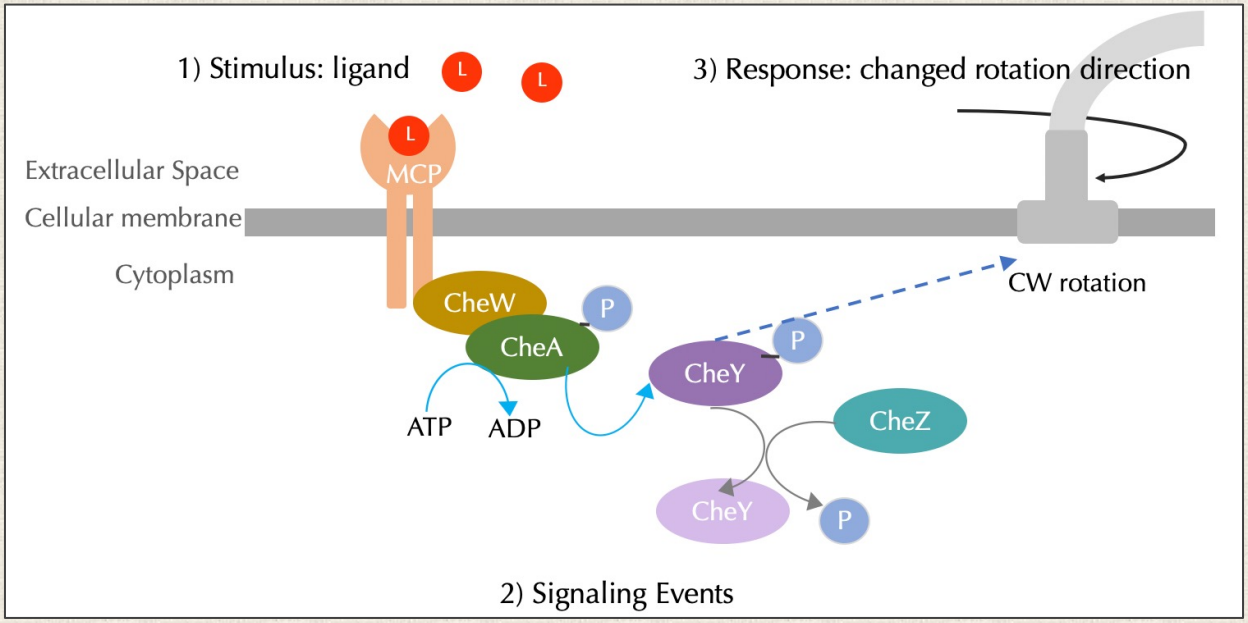
E. coli is smarter than you think

Recall that in the absence of an attractant, CheW and CheA readily bind to an MCP, leading to greater autophosphorylation of CheA, which phosphorylates CheY, increasing tumbling.



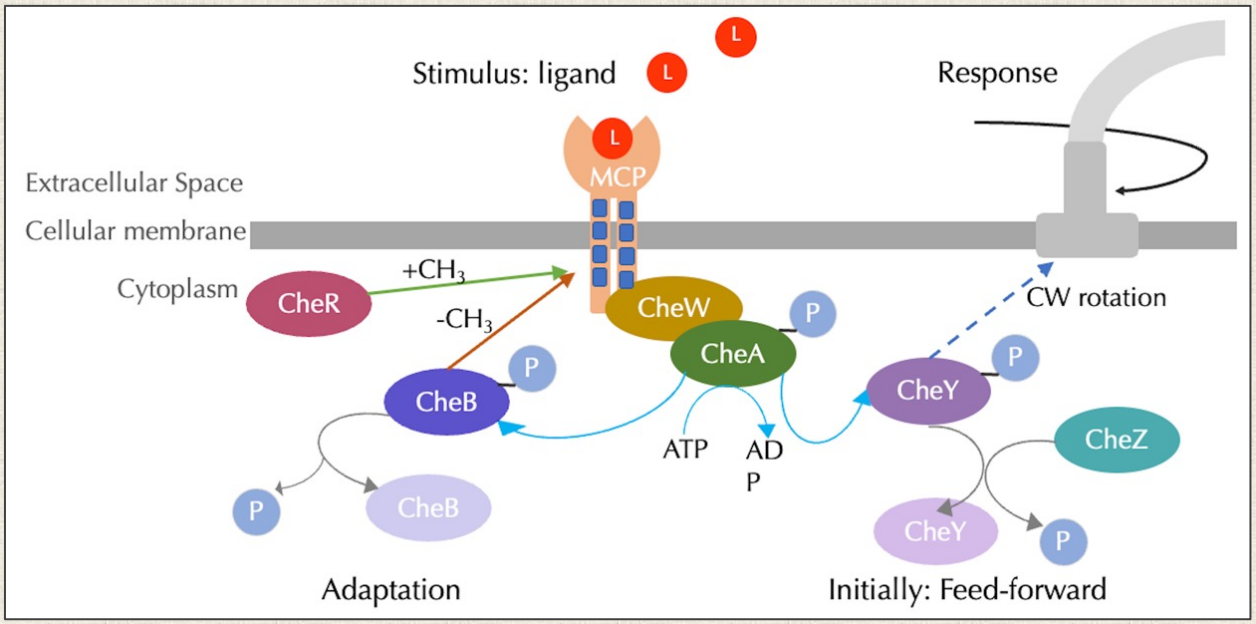
E. coli is smarter than you think

E. coli also has a "memory" of past concentrations through a chemical process called **methylation**, in which ($-CH_3$) is added to a molecule.



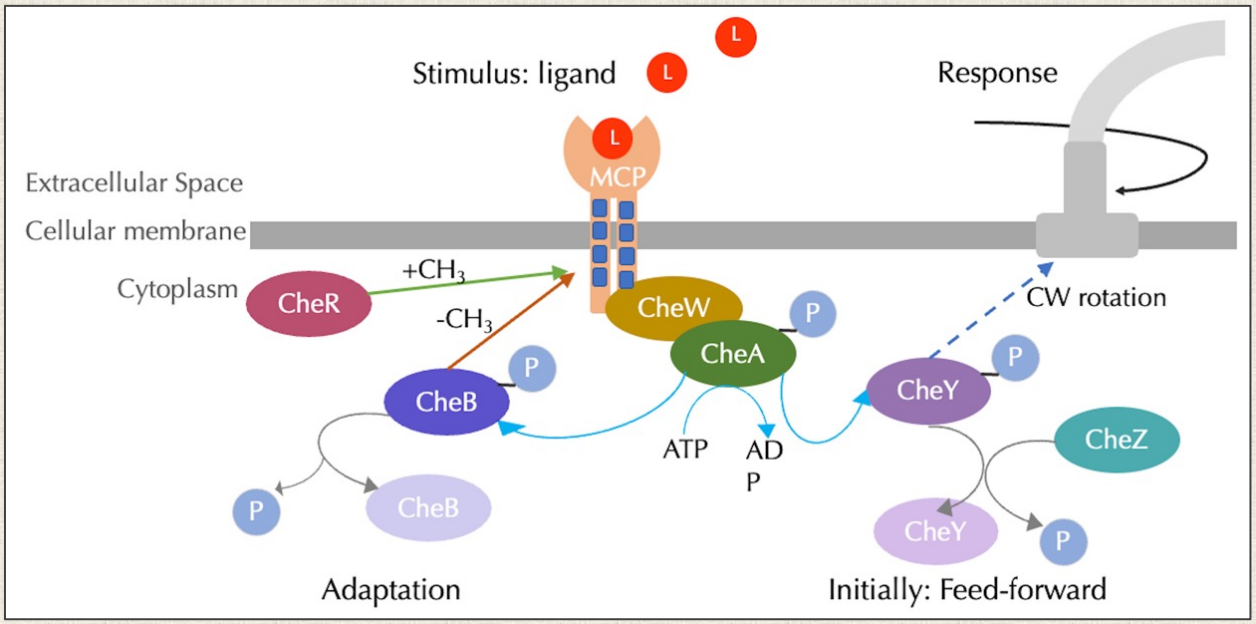
E. coli is smarter than you think

Every MCP receptor has four methylation sites. The more sites that are methylated, the higher the phosphorylation rate of CheA, therefore the higher the phosphorylation of CheY, and higher tumbling.



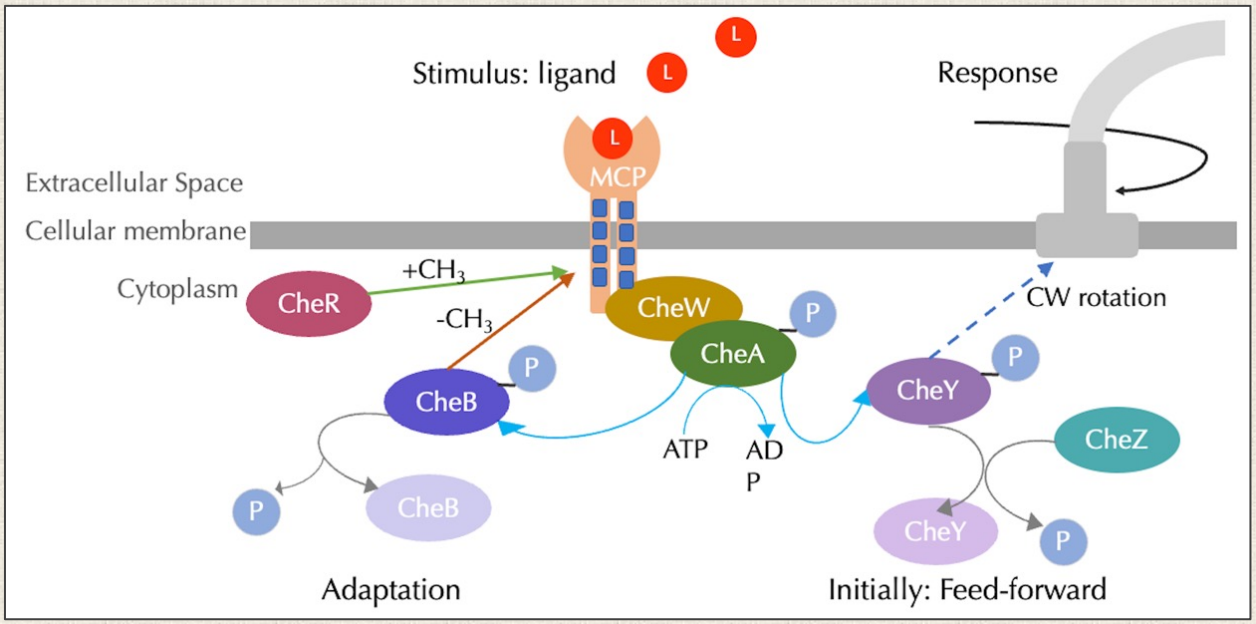
E. coli is smarter than you think

CheR methylates ligand-MCP complexes, so that if the attractant concentration is high but stable, this methylation will boost CheA autophosphorylation, raising tumbling frequency to default levels.



E. coli is smarter than you think

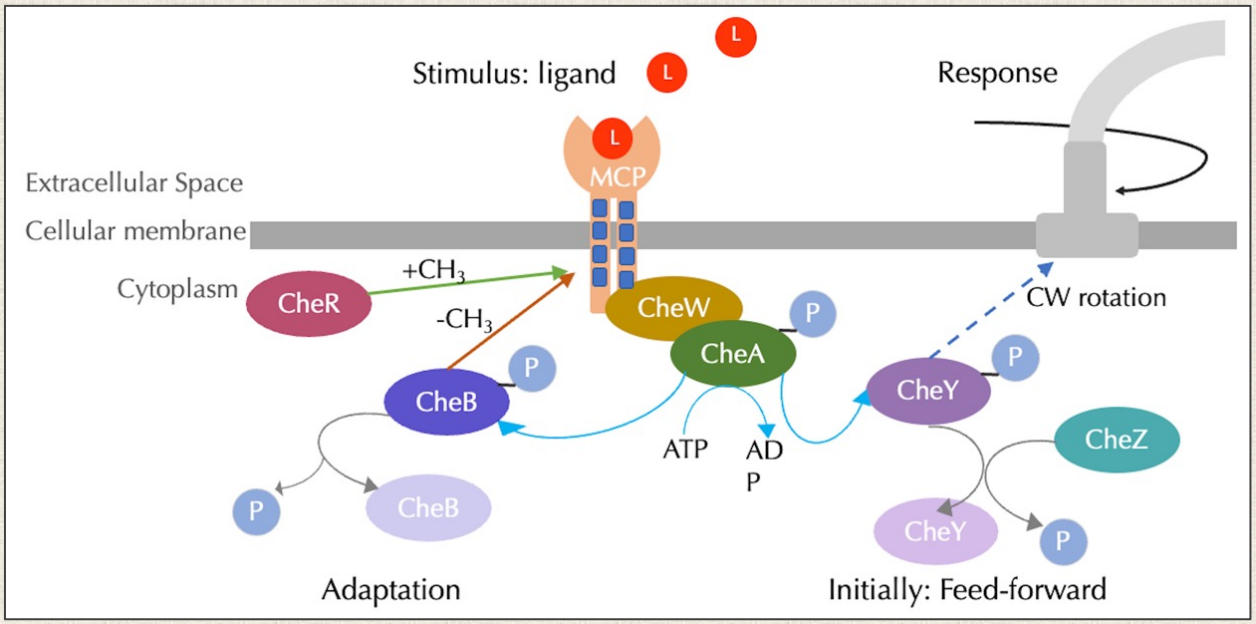
Methylation should be temporary and can be undone with **CheB**, which works faster when methylation is high.



E. coli is smarter than you think

Tumbling frequency can be increased in two ways:

1. Phosphorylation cascade (low attractant)
2. Increased MCP methylation (when attractant stabilizes).



E. coli is smarter than you think

Tumbling frequency can be increased in two ways:

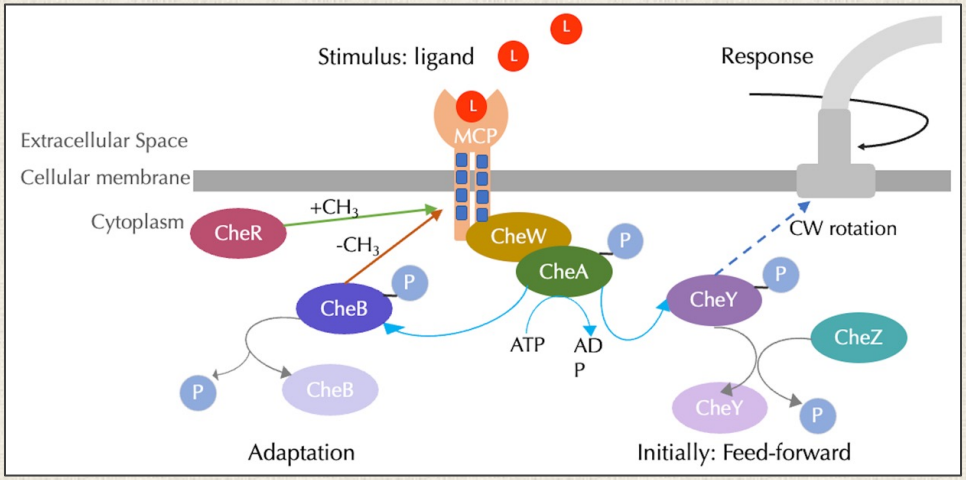
1. Phosphorylation cascade (low attractant)
2. Increased MCP methylation (when attractant stabilizes).

#1 allows the cell to respond to *absolute* changes in attractant concentration, and #2 means that the cell can adapt based on *relative* changes in this concentration.

Building this model from scratch would be a nightmare!

Each MCP could have one of 40 states:

- is it bound to a ligand? (2 states)
- is it bound to CheR? (2 states)
- is it phosphorylated? (2 states)
- which methylation state is it in? (5 states)



Building this model from scratch would be a nightmare!

Each MCP could have one of 40 states:

- is it bound to a ligand? (2 states)
- is it bound to CheR? (2 states)
- is it phosphorylated? (2 states)
- which methylation state is it in? (5 states)

STOP: Say that we only want to model the MCP-ligand binding reaction, $T + L \rightarrow TL$. How many of the 40 possible states does this affect?

Building this model from scratch would be a nightmare!

Each MCP could have one of 40 states:

- is it bound to a ligand? (2 states)
- is it bound to CheR? (2 states)
- is it phosphorylated? (2 states)
- which methylation state is it in? (5 states)

Answer: 20, since it corresponds to the half of the MCP states in which the MCP is unbound to ligand. And so we will need 20 *different* reactions!

Building this model from scratch would be a nightmare!

Each MCP could have one of 40 states:

- is it bound to a ligand? (2 states)
- is it bound to CheR? (2 states)
- is it phosphorylated? (2 states)
- which methylation state is it in? (5 states)

The number of reactions needed to represent a complex system grows very fast. This principle is called **combinatorial explosion**.

Building this model from scratch would be a nightmare!

Each MCP could have one of 40 states:

- is it bound to a ligand? (2 states)
- is it bound to CheR? (2 states)
- is it phosphorylated? (2 states)
- which methylation state is it in? (5 states)

And yet, all 20 reactions can be summarized by one rule: “regardless of the other states, allow unbound MCPs to bind to ligand at some rate”.

Rule-based modeling fixes combinatorial explosion

Rule-based modeling: a modeling approach that uses a small set of "rules" to generate a potentially huge number of different reactions automatically.

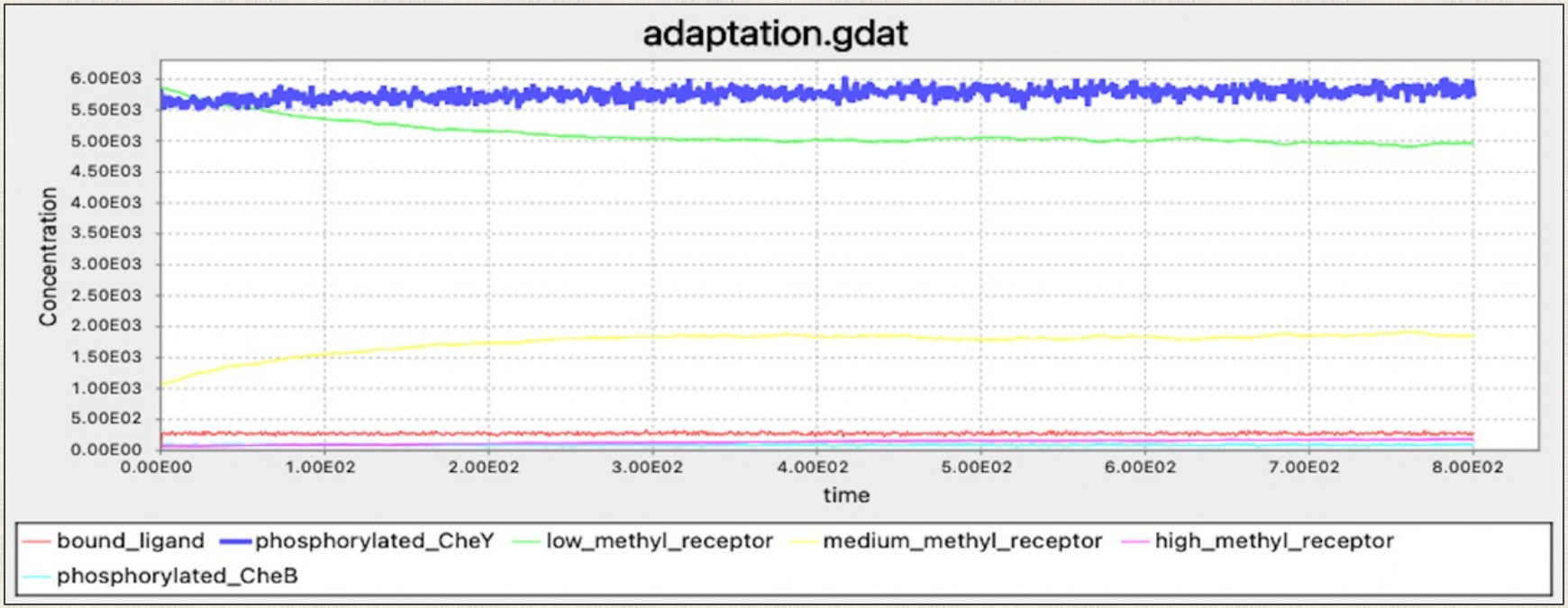
Rule-based modeling of biochemical systems with BioNetGen

by JR Faeder · 2009 · Cited by 491 — **Rule-based modeling** involves the representation of molecules as structured objects and molecular interactions as rules for transforming the…

If you're interested in seeing how to use rule-based modeling to build a complete model of chemotaxis, check out http://biologicalmodeling.org.

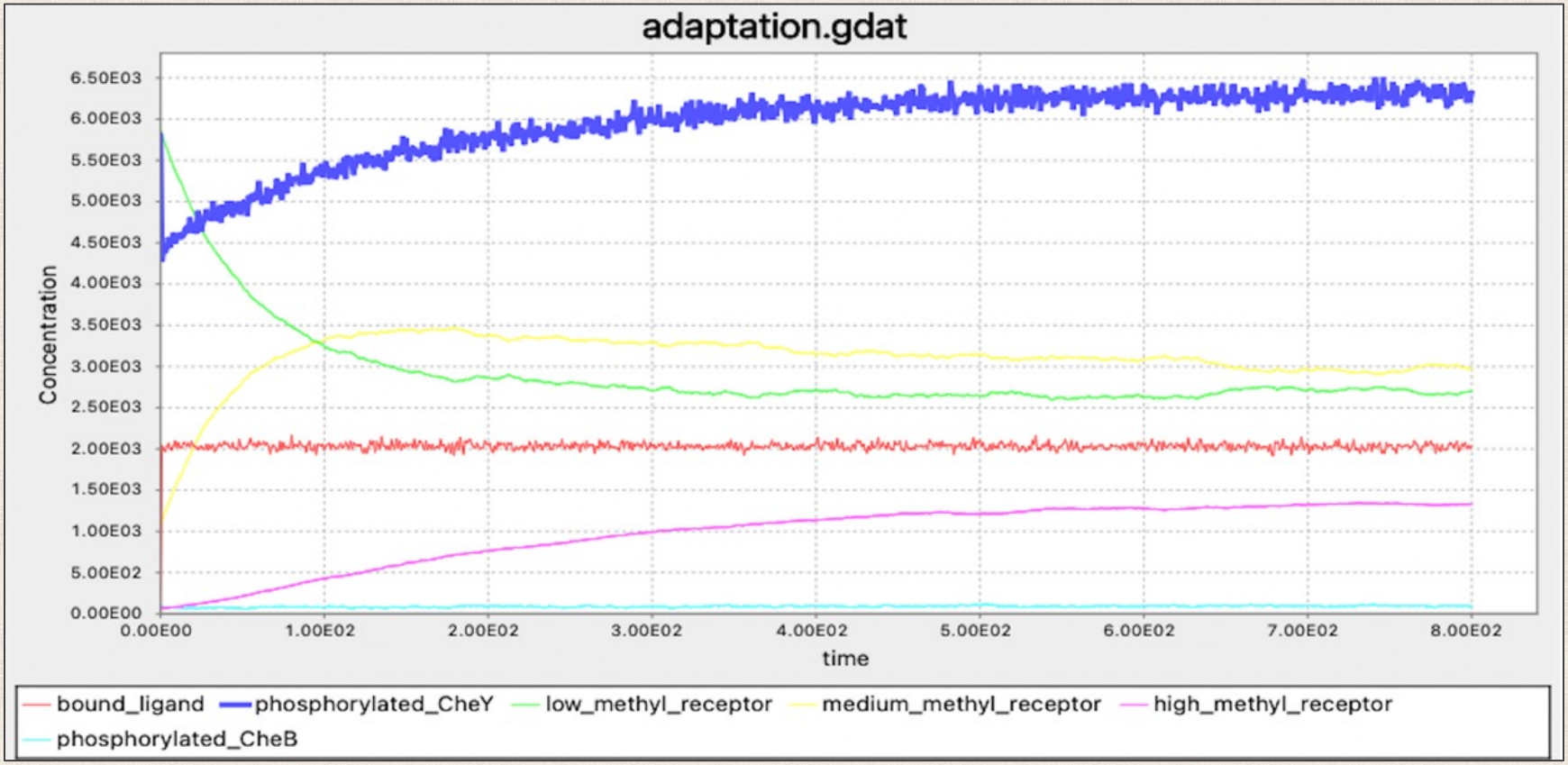
Our bacterium can return to default tumbling even with huge attractant boost

First, we add a relatively small amount of attractant, setting I_0 equal to 10,000, which has essentially no effect.



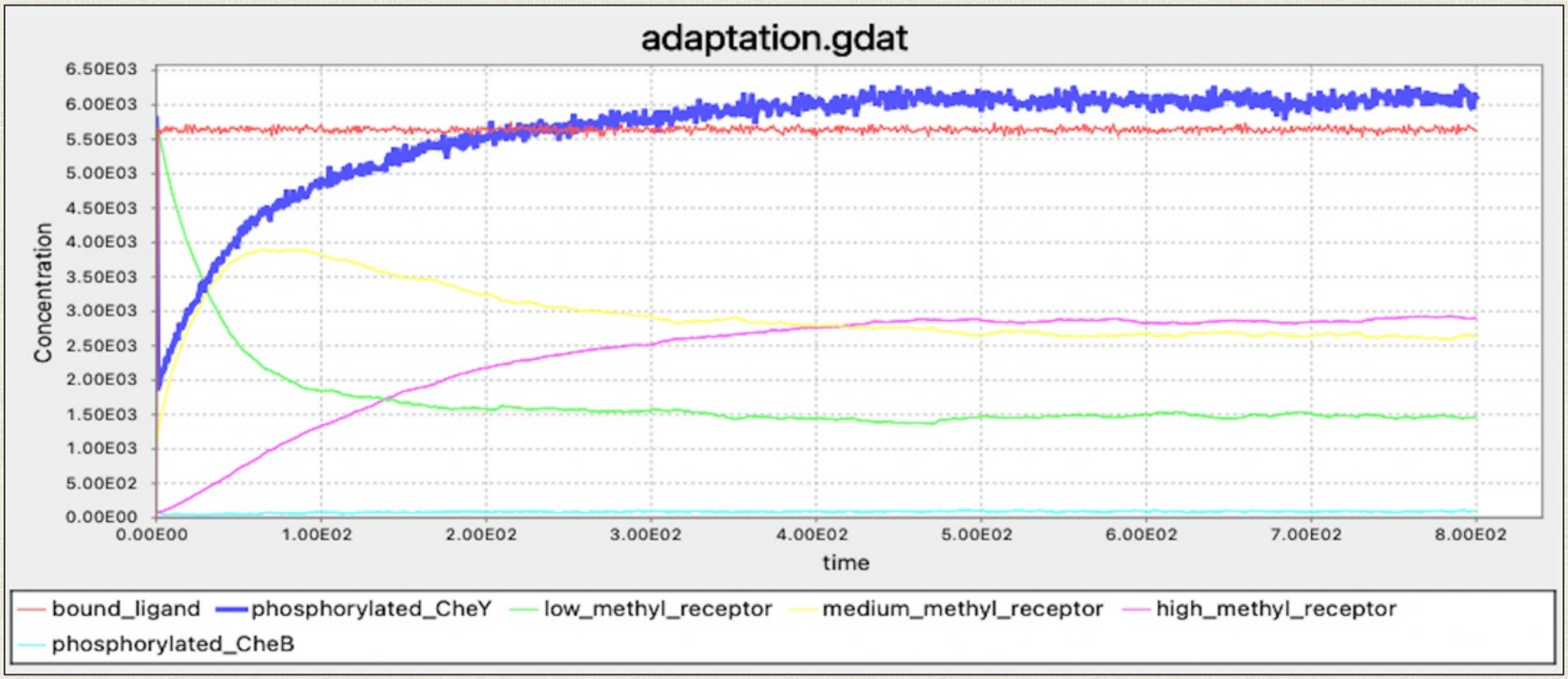
Our bacterium can return to default tumbling even with huge attractant boost

Setting l_0 equal to 100,000 causes a change, and equilibrium is achieved in a few minutes.



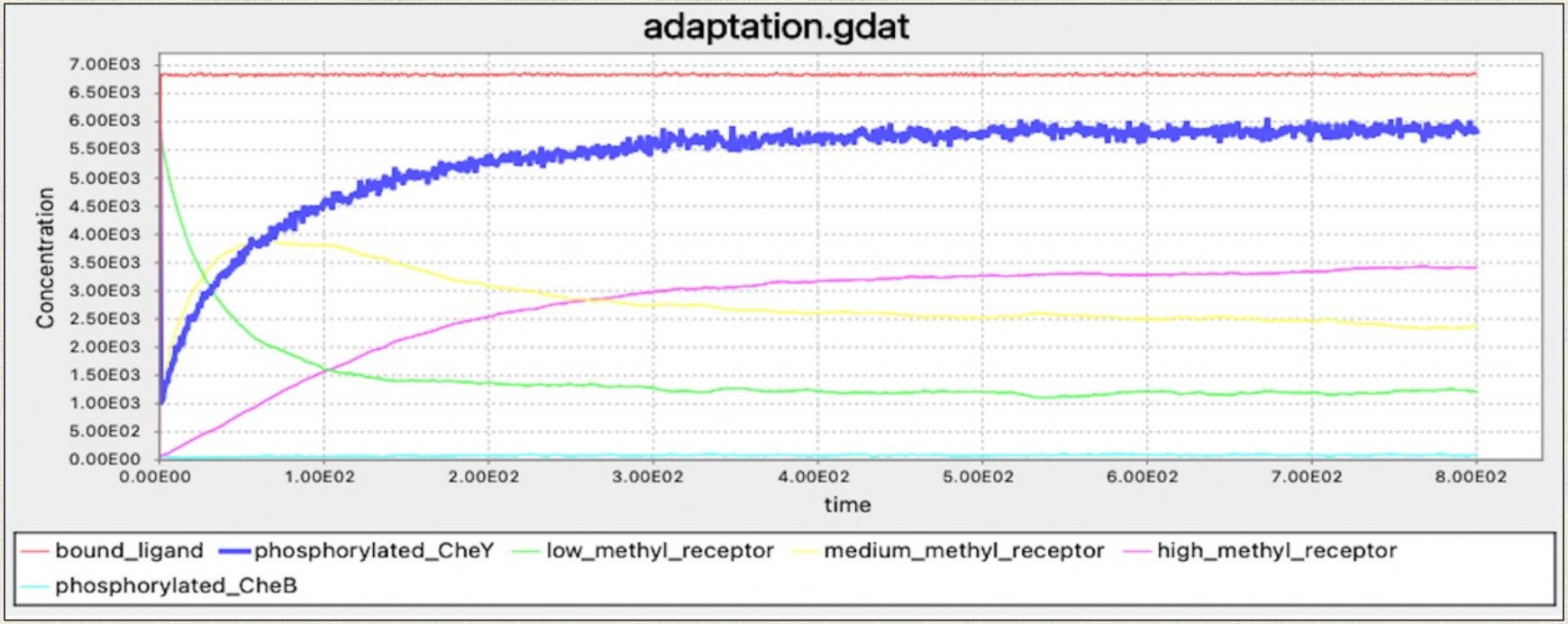
Our bacterium can return to default tumbling even with huge attractant boost

With l_0 =1,000,000, the initial drop is bigger, but the system still returns to equilibrium quickly.



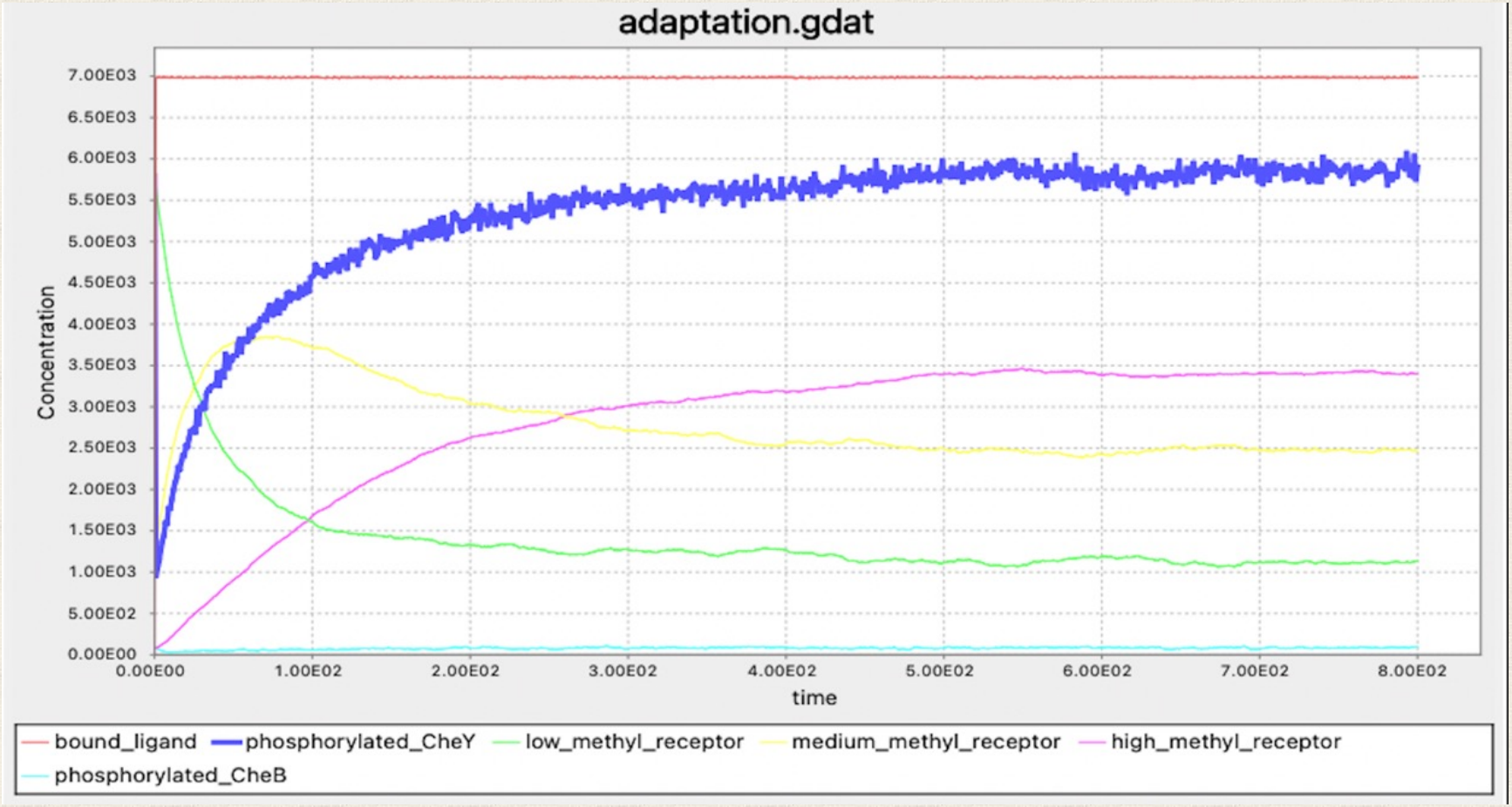
Our bacterium can return to default tumbling even with huge attractant boost

When I_0 = 10,000,000, we see an even bigger jolt.



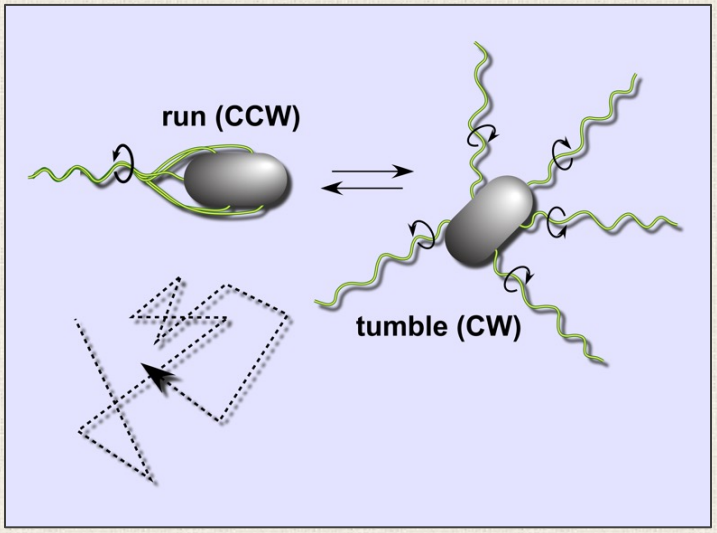
Our bacterium can return to default tumbling even with huge attractant boost

The system is still robust if I_0 =100,000,000.



Yet it seems like we are missing something

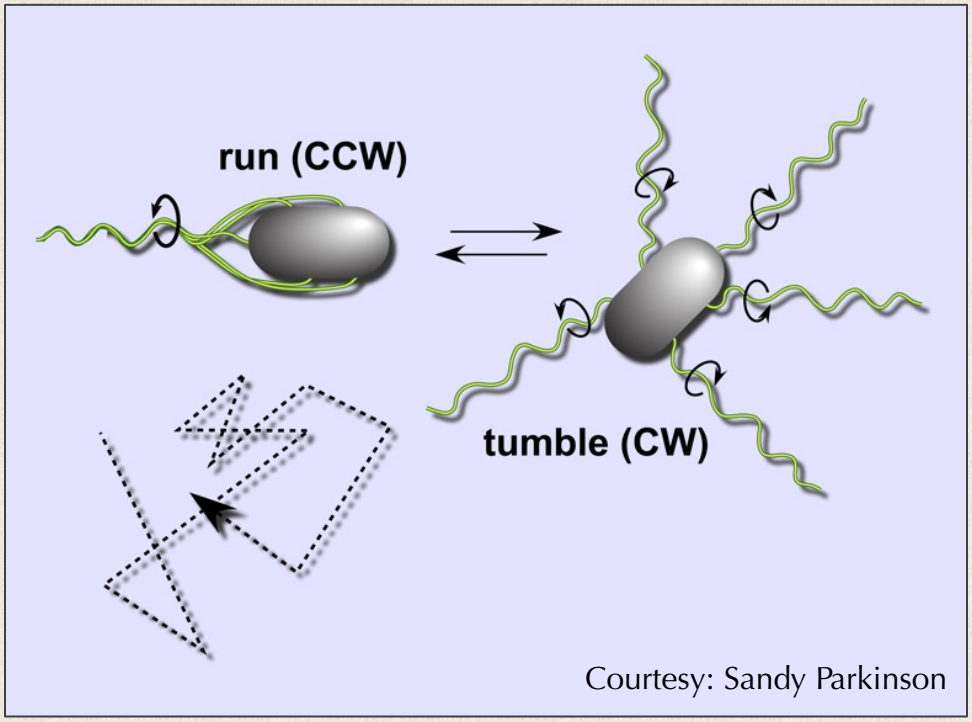
But according to the run and tumble model, the direction that a bacterium is moving at any point in time is random! So why would a decrease in tumbling frequency help *E. coli* move *toward* an attractant?



THE BEAUTY OF E. COLI'S ROBUST RANDOMIZED EXPLORATION ALGORITHM

Let's model the chemotaxis algorithm!

Recall the "run and tumble model": *E. coli* alternates between running and tumbling in place.



A simple chemotaxis algorithm

Take the following actions for a finite number *n* of steps.

- Select a random direction of movement (uniform random decimal between 0 and 360 degrees)
- Randomly select a duration of movement.
- Move the bacterium to a new position indicated by the direction and duration of movement.

A simple chemotaxis algorithm

Take the following actions for a finite number *n* of steps.

- Select a random direction of movement (uniform random decimal between 0 and 360 degrees)
- Randomly select a duration of movement.
- Move the bacterium to a new position indicated by the direction and duration of movement.

STOP: How should we select the duration of movement?

A simple chemotaxis algorithm

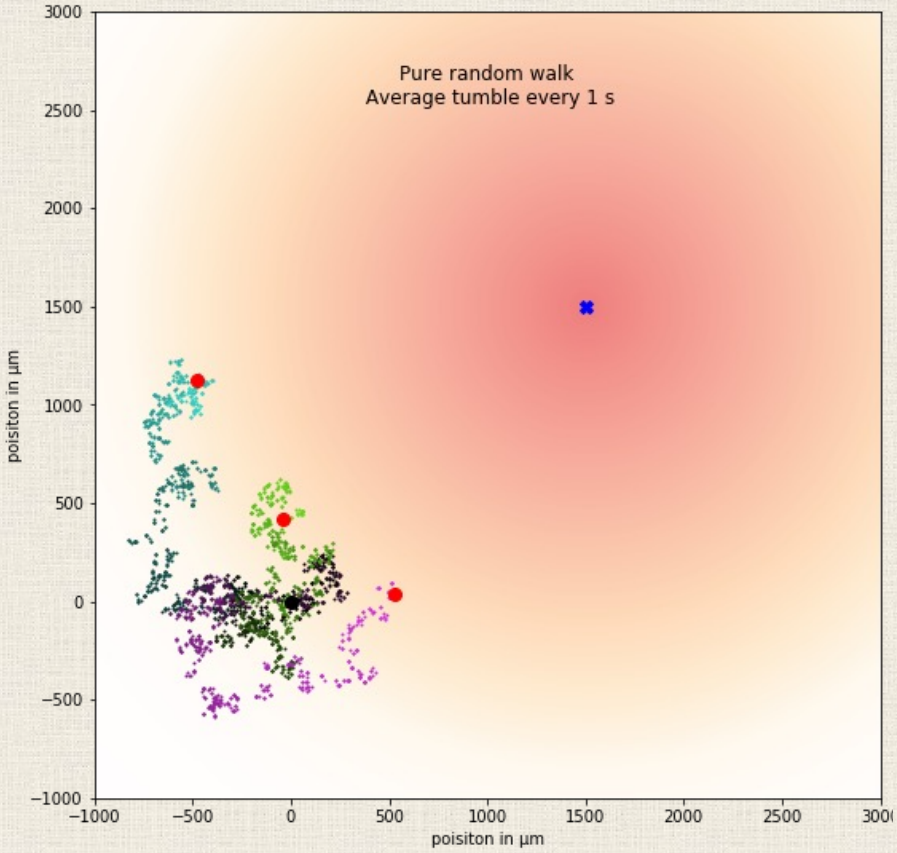
Take the following actions for a finite number *n* of steps.

- Select a random direction of movement (uniform random decimal between 0 and 360 degrees)
- Randomly select a duration of movement.
- Move the bacterium to a new position indicated by the direction and duration of movement.

Answer: We are *waiting* for the next change of direction, so we use an exponential distribution! (With mean = experimentally verified 1.0 second.)

Mimicking a real environment

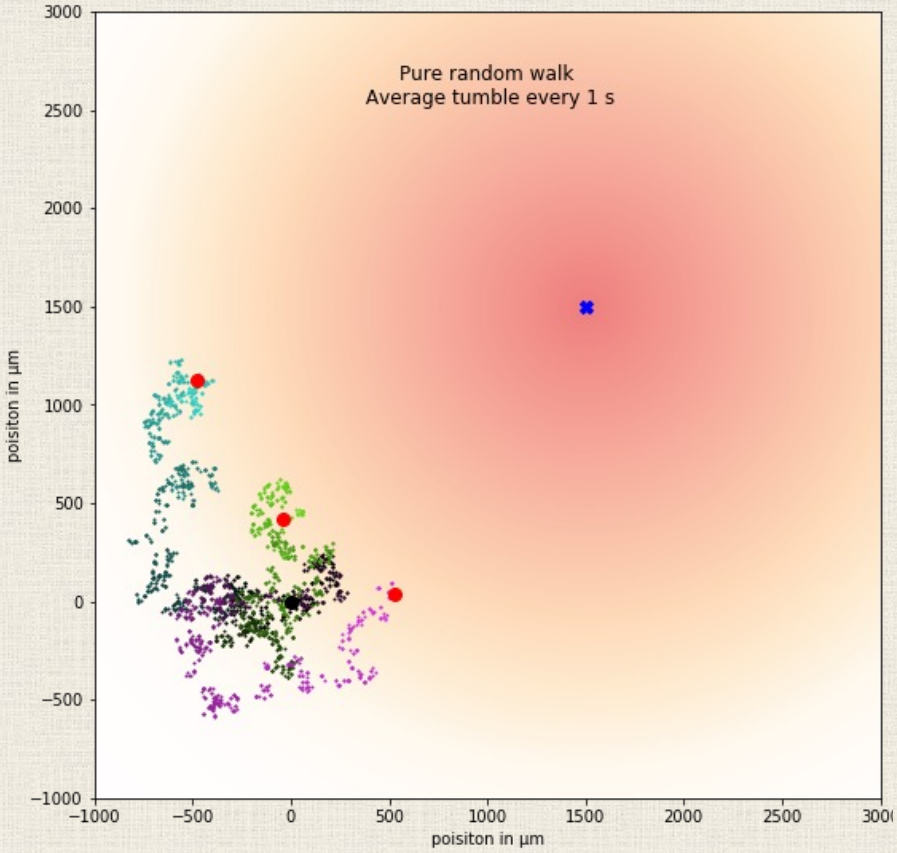
Let's start a bacterium at (0,0) in an attractant gradient that is maximized at a blue "goal" (1500, 1500).



Mimicking a real environment

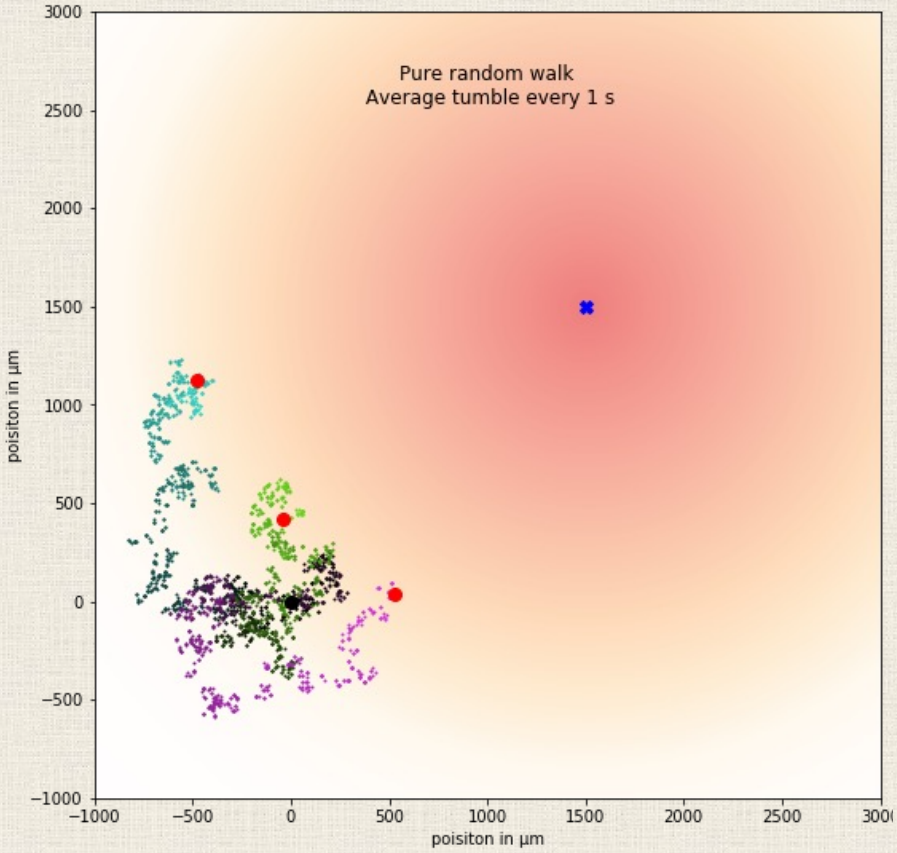
Let's start a bacterium at (0,0) in an attractant gradient that is maximized at a blue "goal" (1500, 1500).

This shows three different walks, ending at the red points.



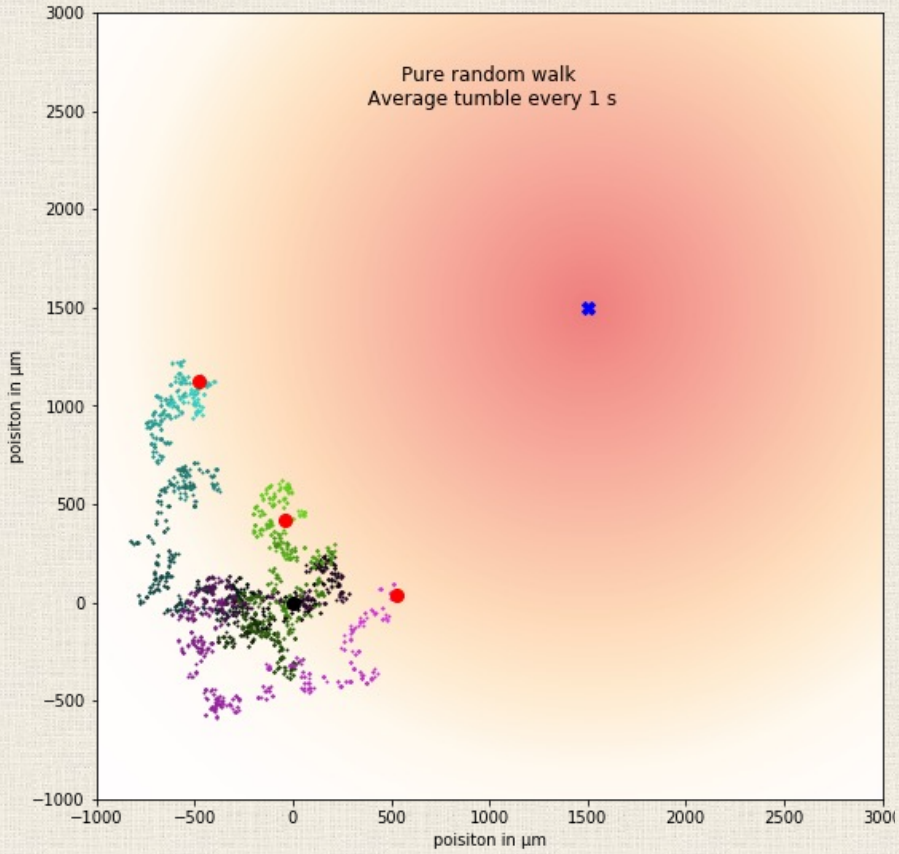
This algorithm doesn't really model chemotaxis

STOP: This is just a random walk with variable run times, so the bacterium won't get any closer to a goal. How does the real chemotaxis algorithm differ from it?



This doesn't really model chemotaxis

Answer: Recall that the duration of a bacterium's run depends on the *relative* change in attractant concentration that it detects.



A more realistic chemotactic algorithm

Take the following actions for a finite number *n* of steps.

- Select a random direction of movement (uniform random decimal between 0 and 360 degrees)
- Randomly select a duration of movement, such that the larger the difference $\Delta[L]$ between the concentration at the cell's current point andits previous point, the longer the walk.
- Move the bacterium to a new position indicated by the direction and duration of movement.

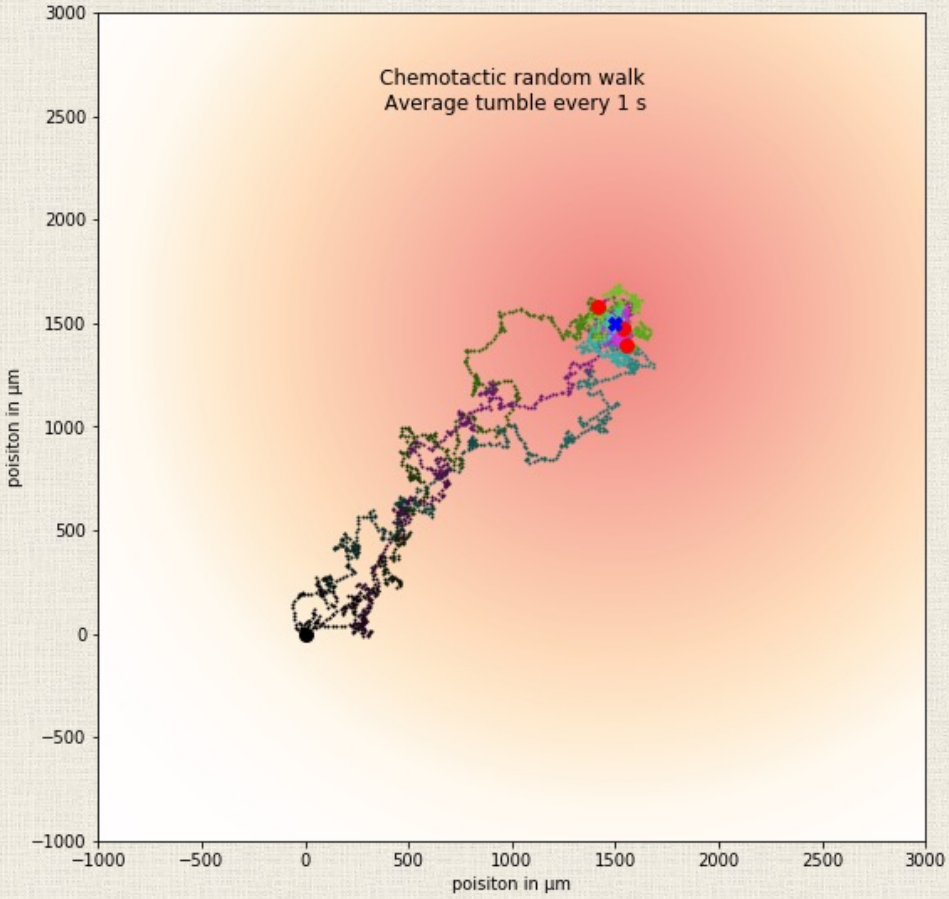
A more realistic chemotactic algorithm

Note: This step may seem tricky, but we still use an exponential distribution. The mean "wait time" to stop and tumble is simply proportional to $\Delta[L]$.

- Randomly select a duration of movement, such that the larger the difference $\Delta[L]$ between the concentration at the cell's current point andits previous point, the longer the walk.

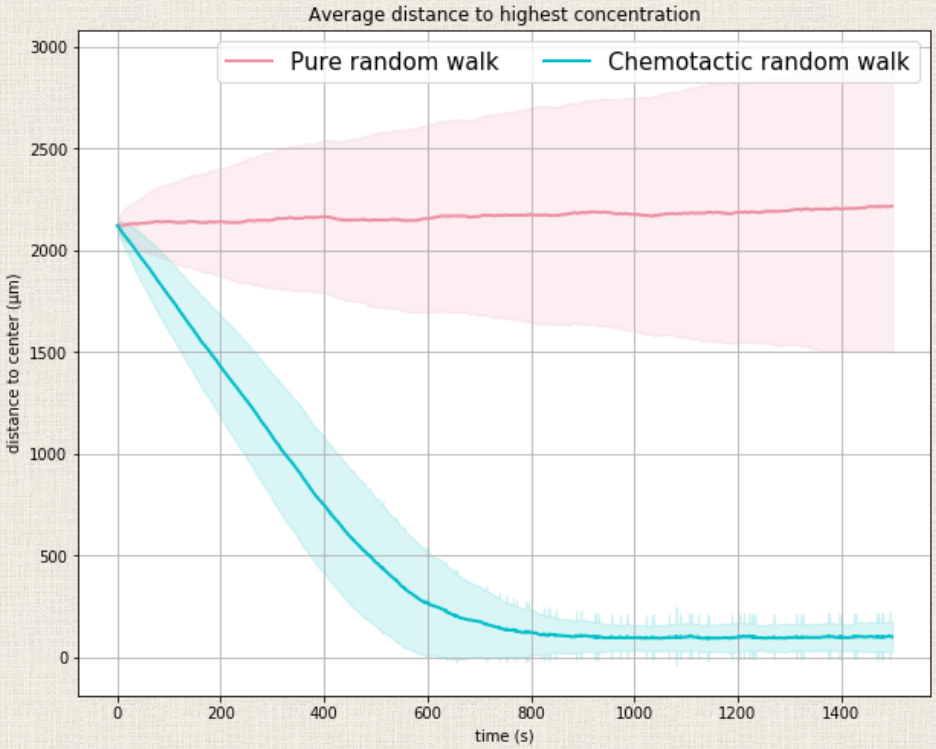
A tiny change made a big difference

The only change in our algorithm is “if the current concentration is getting larger, run for longer”. And yet we are able to hone in on the goal quickly.



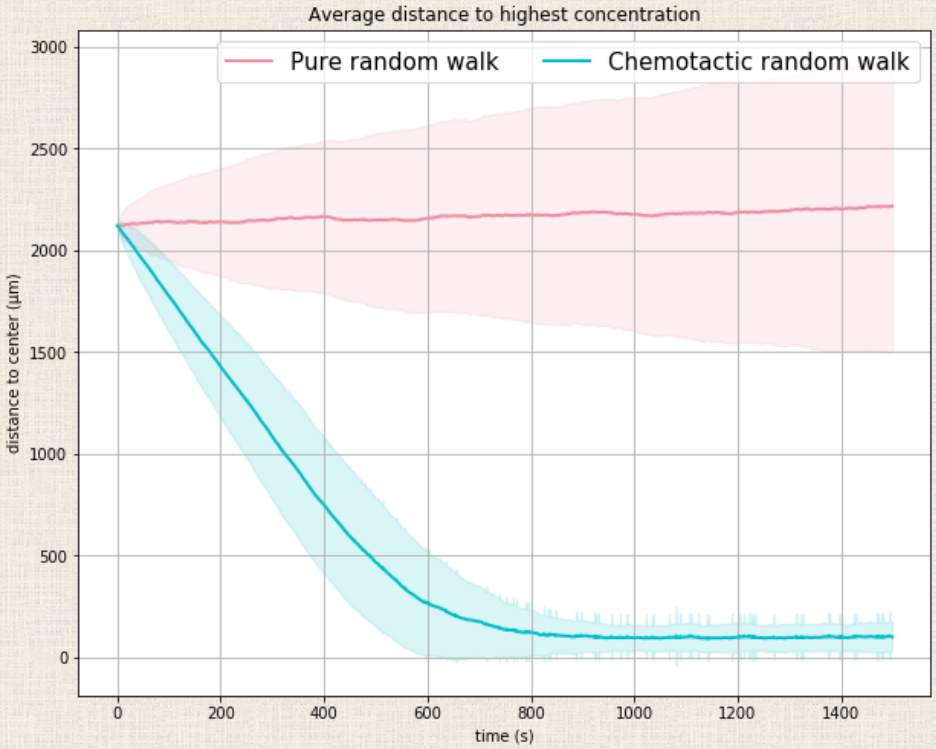
From 3 to 500 simulations

Average distance to the goal over time for 500 simulations following the two strategies. The shaded area for each strategy represents one standard deviation from the average.



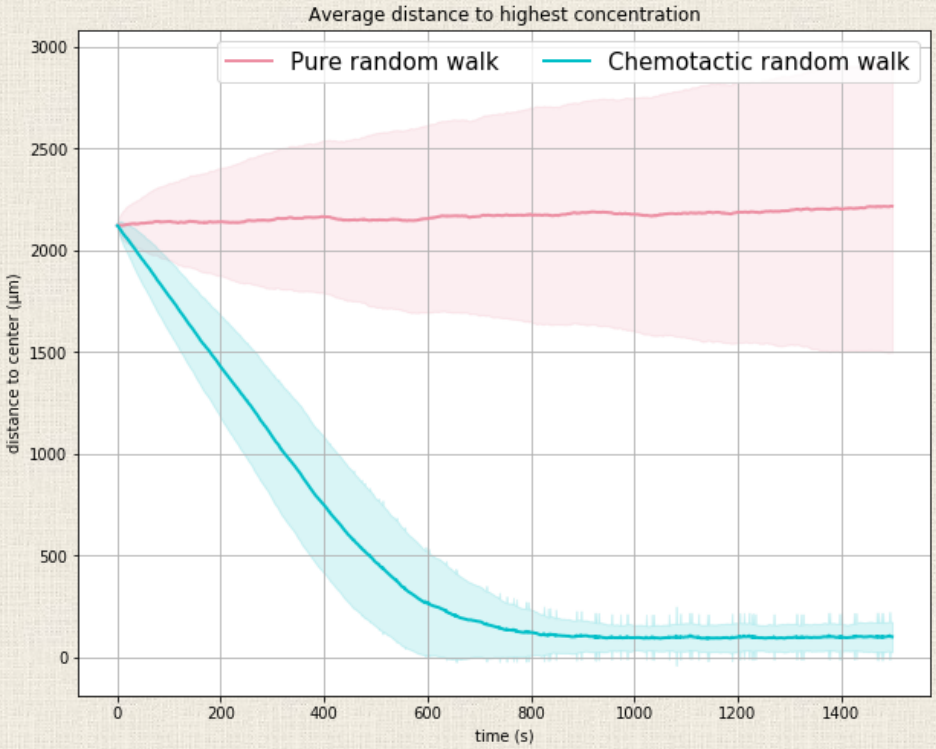
From 3 to 500 simulations

STOP: Why do you think that such a simple algorithm is able to reach the goal?



From 3 to 500 simulations

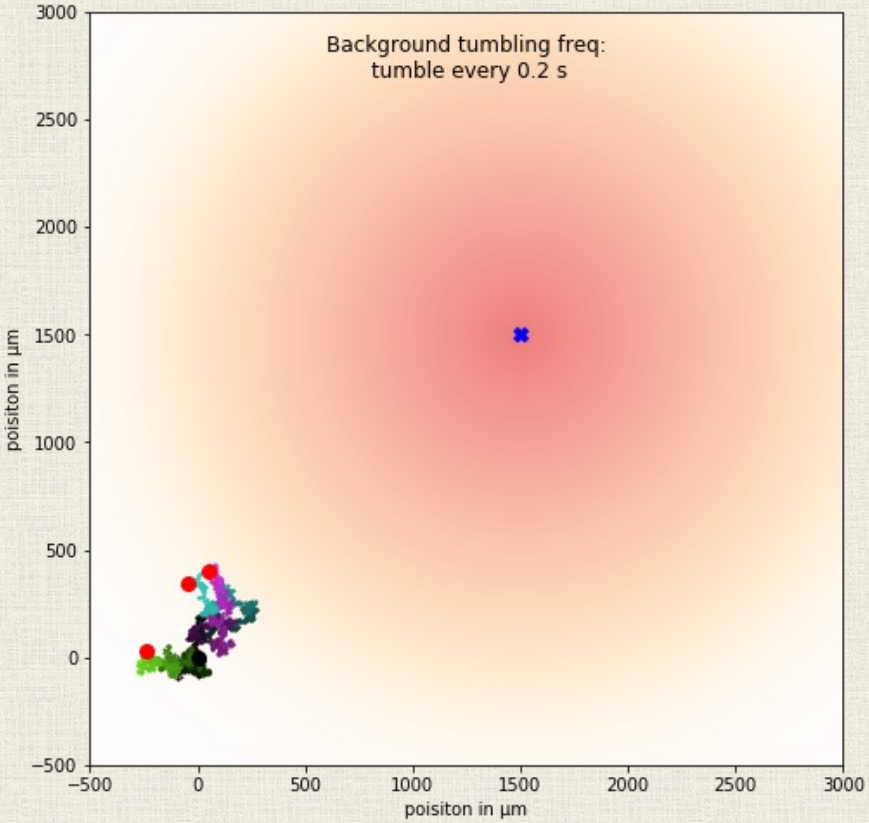
STOP: Why do you think that such a simple algorithm is able to reach the goal?



Answer: Attractant detection acts like a "rubber band" that prevents bacterium from running too far in wrong direction.

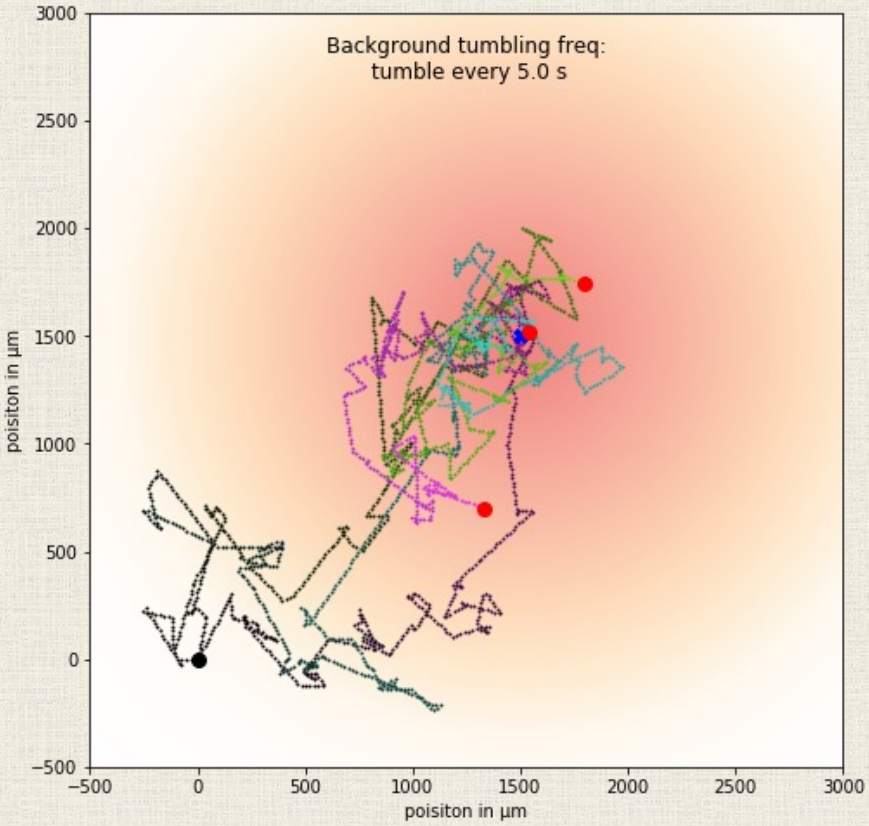
Tumbling frequency is robust to disturbance across species

If we change the default tumbling frequency (when $\Delta[L] = 0$) from 1.0 secs to 0.2 secs, the "rubber band" is too tight.



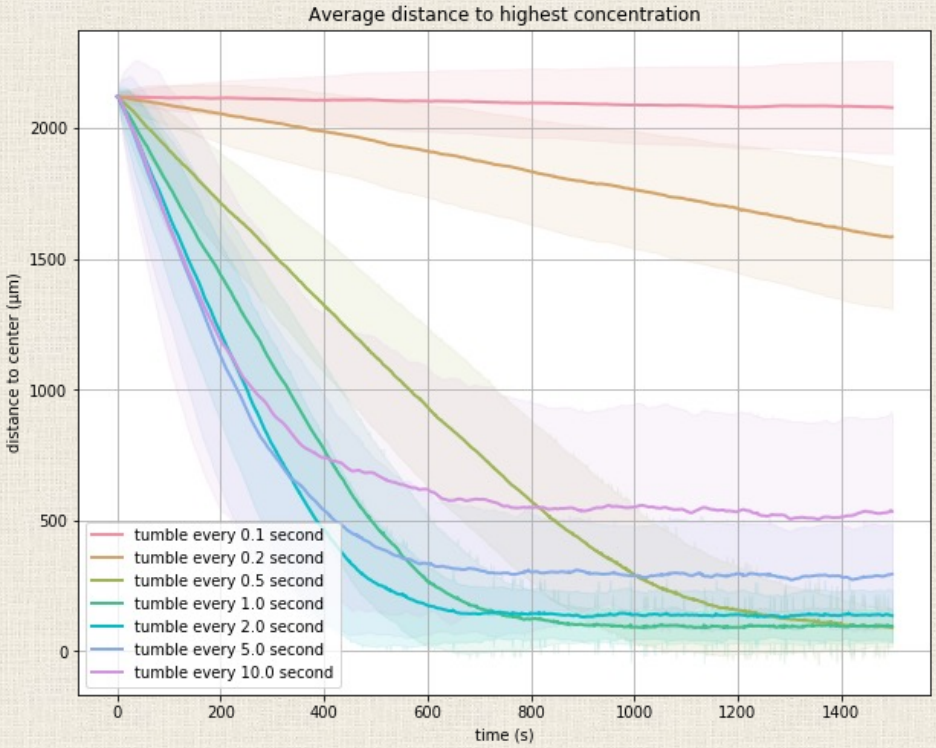
Tumbling frequency is robust to disturbance across species

And if we change it to every 5.0 secs, then the rubber band is too flexible, and cells run past the goal without being able to stop.



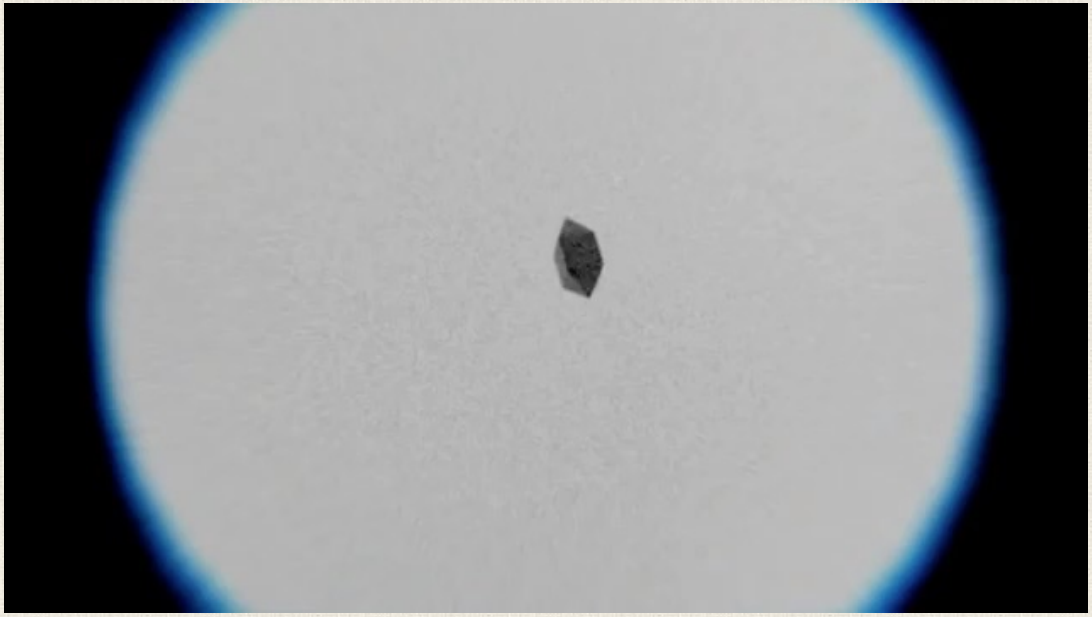
Tumbling frequency is robust to disturbance across species

Trying a collection of different frequency values shows that bacteria are able to quickly find and stay at food when they tumble every ~1.0 second.



Bacteria are even smarter than we thought

Saragosti et al. 2011: When moving toward an attractant, the bacterium makes only small directional changes.



https://www.youtube.com/watch?v=F6QMU3KD7zw

A remark on solving computational problems

Note: If our problem is to find a sugar cube, then there isn't a computational problem to solve – just move greedily in the direction of largest increase.

A remark on solving computational problems

Note: If our problem is to find a sugar cube, then there isn't a computational problem to solve – just move greedily in the direction of largest increase.

And yet what nature has evolved is somehow a more *robust* approach that relies on probability, despite having no real intelligence/communication.

A remark on solving computational problems

Note: If our problem is to find a sugar cube, then there isn't a computational problem to solve – just move greedily in the direction of largest increase.

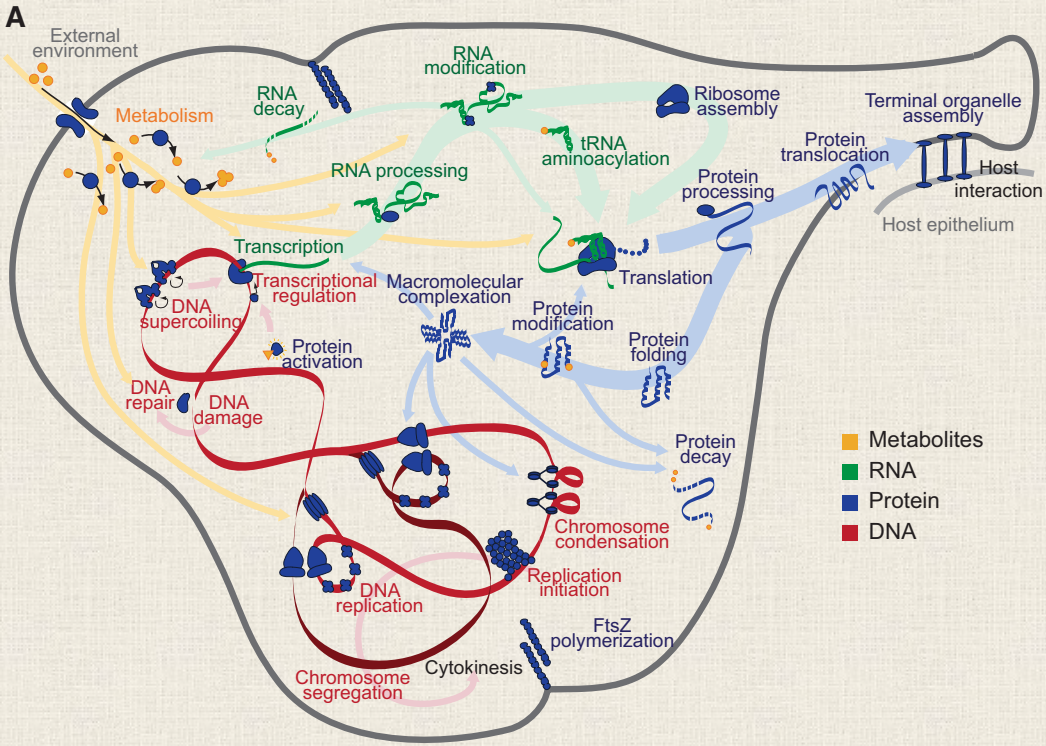
And yet what nature has evolved is somehow a more *robust* approach that relies on probability, despite having no real intelligence/communication.

This will be a theme of our work when we discuss more about *algorithms in nature*.

CONCLUSION: TOWARD A COMPLETE MODEL OF THE BACTERIAL CELL

Once we have a model of chemotaxis, what should we do with it?

We could model every process for a very simple bacterium (*M. genitalum*, only 525 genes).



Once we have a model of chemotaxis, what should we do with it?

We could model every process for a very simple bacterium (*M. genitalum*, only 525 genes).

Then build a "super-model" that links up these smaller models into a model of the cell.

[HTML] A whole-cell computational model predicts phenotype from genotype

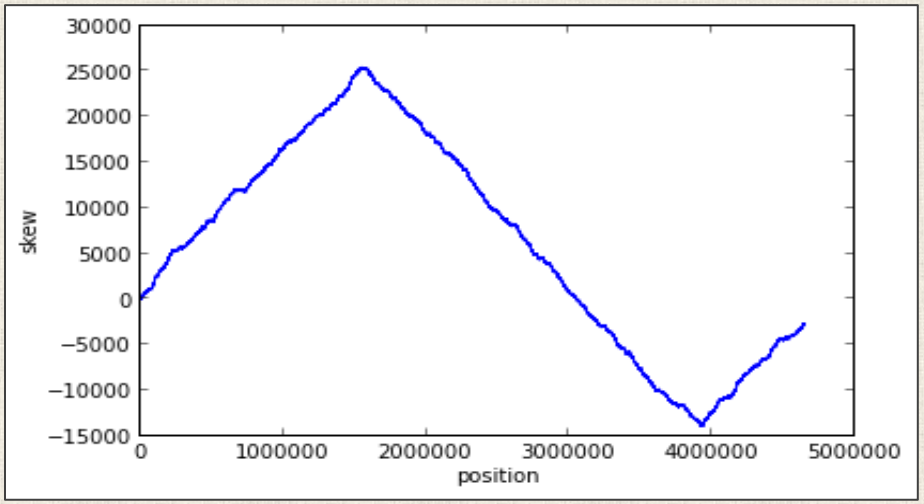
JR Karr, JC Sanghvi, DN Macklin, MV Gutschow… - Cell, 2012 - Elsevier

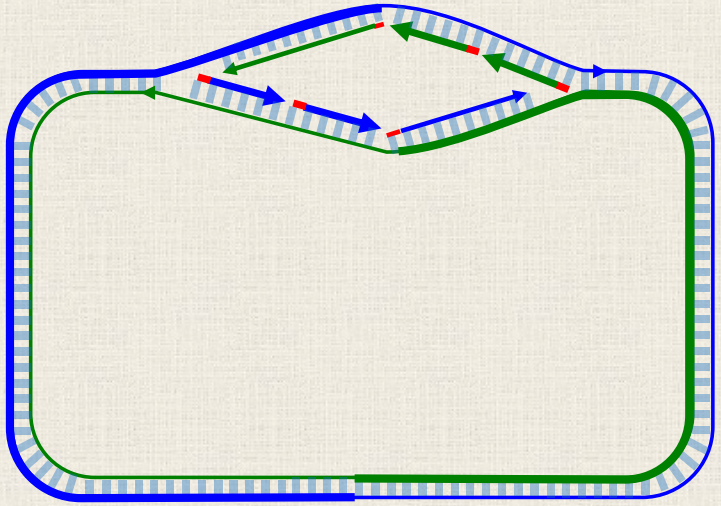
Understanding how complex phenotypes arise from individual molecules and their interactions is a primary challenge in biology that computational approaches are poised to tackle. We report a whole-cell computational model of the life cycle of the human pathogen Mycoplasma genitalium that includes all of its molecular components and their interactions. An integrative approach to modeling that combines diverse mathematics enabled the simultaneous inclusion of fundamentally different cellular processes and experimental …

☆ 99 Cited by 1249 Related articles All 39 versions

‘member replication?

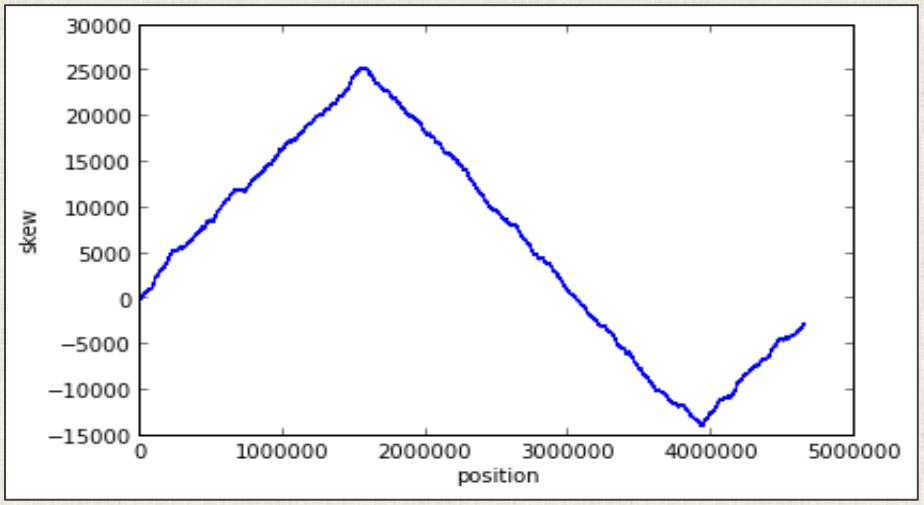
The whole cell model showed that the lengths of *initiation* and *replication* are inversely correlated. In other words, the length of replication is *robust* to small stochastic changes in the cell.

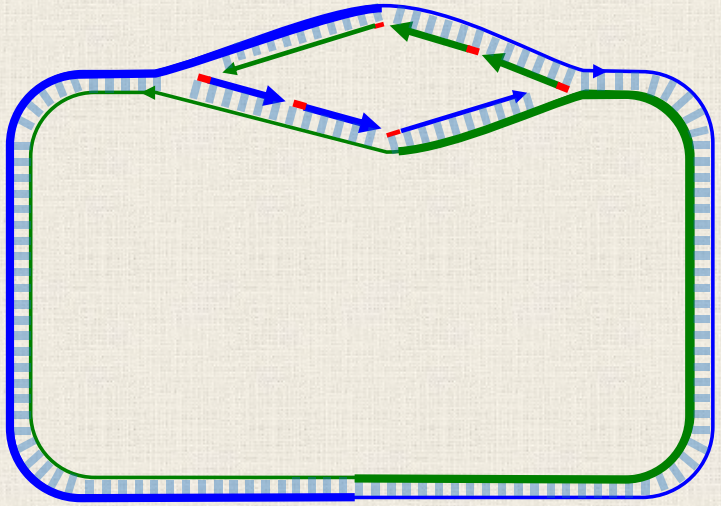




‘member replication?

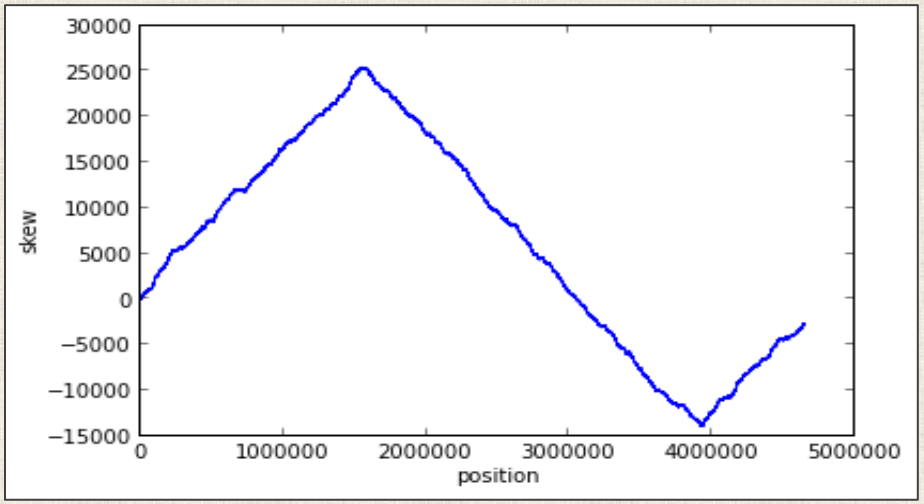
Why? If initiation of replication is slow, the cell builds up a larger surplus of dNTP molecule used by DNA polymerase during replication.

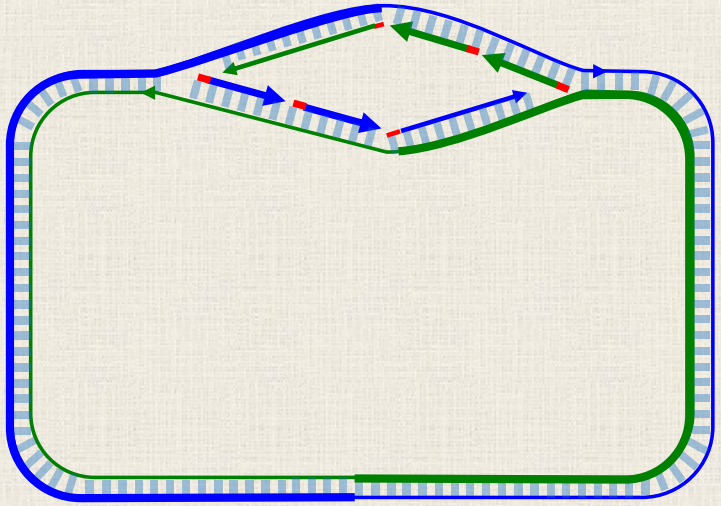




Doing biological research with a computational model

Key point: This was a *new biological* observation made by a purely *computational* model that was outside known research at the time.





And yet biology remains difficult …

Key point: This was a *new biological* observation made by a purely *computational* model that was outside known research at the time.

Unfortunately, still no model of this sophistication for *E. coli* has been published.

[HTML] A whole-cell computational model predicts phenotype from genotype

JR Karr, JC Sanghvi, DN Macklin, MV Gutschow… - Cell, 2012 - Elsevier

Understanding how complex phenotypes arise from individual molecules and their interactions is a primary challenge in biology that computational approaches are poised to tackle. We report a whole-cell computational model of the life cycle of the human pathogen Mycoplasma genitalium that includes all of its molecular components and their interactions. An integrative approach to modeling that combines diverse mathematics enabled the simultaneous inclusion of fundamentally different cellular processes and experimental …

☆ 99 Cited by 1249 Related articles All 39 versions

As of 2021, they were 43% finished

The E. coli Whole-Cell Modeling Project

Gwanggyu Sun [#1], Travis A Ahn-Horst [#1], Markus W Covert [1]

Affiliations + expand

PMID: 34242084 DOI: 10.1128/ecosalplus.ESP-0001-2020

Abstract

The Escherichia coli whole-cell modeling project seeks to create the most detailed computational model of an E. coli cell in order to better understand and predict the behavior of this model organism. Details about the approach, framework, and current version of the model are discussed. Currently, the model includes the functions of 43% of characterized genes, with ongoing efforts to include additional data and mechanisms. As additional information is incorporated in the model, its utility and predictive power will continue to increase, which means that discovery efforts can be accelerated by community involvement in the generation and inclusion of data. This project will be an invaluable resource to the E. coli community that could be used to verify expected physiological behavior, to predict new outcomes and testable hypotheses for more efficient experimental design iterations, and to evaluate heterogeneous data sets in the context of each other through deep curation.