Stochastic simulation

For these slides see:
www.physics.upenn.edu/~pcn
Activated hopping

Can we get a computer to draw from an exponential distribution?

% make it nicer by plotting log freqs:
N=histc(-log(rand(1,100000)),0:20);
bar(.5+(0:20),log(N))

hist(-log(rand(1,50000)),100);

% make it nicer by plotting log freqs:
N=histc(-log(rand(1,100000)),0:20);
bar(.5+(0:20),log(N));
flipper

pcn 8/2017
simulate symmetric double-well hopping

```python
In [ ]: import numpy as np
from vpython import *
from numpy.random import random as rng

In [ ]: framerate = 10  # frame rate in Hz, so 10 means 100ms per frame

In [ ]: dts = -.5*np.log(rng(100))  # waiting times [s] with mean (0.5)s

In [ ]: ts = np.cumsum(dts)  # transition times [s]
lasttime = ts[-1]
print(lasttime)

In [ ]: scene = canvas()
myobject = sphere(pos=vector(1,0,0))
j = 0  # which transition we're on now
for timenow in np.arange(0, lasttime-1, 1/framerate):  # which video frame
    while ts[j] < timenow:  # decide whether a new event has happened
        j+=1  # since last frame
        myobject.pos = vector((-1)**j,0,0)
        rate(framerate)  # pause till it's time to show the frame
```
Ok, that genuinely was easy and fun. But it was more than just fun.

Sure, you can show data in tables and graphs. Sure you can apply lots of sophisticated statistical tools to it. But don’t forget to also present data in a way that looks just your experiment—to generate a time series and present it to your neural wetware as a time series.

That approach can be a great source of intuition.
What about more complicated processes? E.g. *two* kinds of transition?

betaA = .5; % mean rate of process A
betaB = 2; % type B comes faster than A
Nsteps = 50; % how many steps

betatot = betaA + betaB;
deltats = -log(rand(Nsteps,1))/betatot; % waiting times
ts = cumsum(deltats); % event times
types = rand(Nsteps,1) < (betaA/betatot); % event types
Nf=figure; colormap([1 0 0; 0 0 1]);
scatter(ts,zeros(Nsteps,1),[],2-types)

simulated compound Poisson process

 We just invented the “Gillespie algorithm” for “stochastic modeling.”
From there, it’s a short “step” to random walks, made a bit more realistic:
Onward to gene expression

Rob made a hugely important comment: Often simulation is unilluminating, doesn't give general results, can't evaluate its validity, etc. Generally speaking, that happens when we have to introduce extra ad hoc assumptions to get the system into simulatable form.

However, if your model's primary definition is in terms of transitions among discrete states, then simulation is its most direct implementation; here it can be the case that analytic results require suspect approximations, not the other way round.

One abstract representation:       Even more abstract representation:

What’s new here is that the propensities are not constants. Doesn’t matter!
function [ts ls] = transcrip2rxn(lzero, T)
% Gillespie simulation of birth/death process
%% inputs
% lzero = initial number of mRNA
% T = total time to run
%% outputs
% ts = times at which x changed
% ls = running values of x just after those times
%% Parameters
ks = [0.15, 0.014]; % rate constants in 1/minute
stoich = [0, 1]; % reaction orders
%% initialize
k = 0; % current time
x = lzero; % current mRNA population
ts(1) = t; % histories
ls(1) = x;
Here’s an example of two time series (red, blue), for the case in which the molecule count starts out higher than its steady value. Black trace is the continuous deterministic approximation.

Hmm, seems like a lot of work just to recover exponential decay! Was it worth the effort?

P Nelson, *Physical models of living systems* (W. H. Freeman and Co.)
Something much more interesting happens when the total numbers are not large. We see that the “steady” state can actually be pretty lively (big fluctuations). And interesting—those fluctuations follow a very famous distribution.

Um... Why not just see this via the Master equation? Well, for one thing there’s no substitute for watching individual time courses, which after all is what single-molecule experiments see. Also, often the moment you add one little extra bit of realism, then the analytic solutions are lost. Let’s look at some of that realism.
A gene of interest was controlled so that it could be turned on ("induced") at will:

\[ \text{P}_{\text{lac/ara}} \rightarrow \text{mRFP1} \rightarrow \text{96x MS2-bs} \]

Golding et al 2005
Golding et al 2005:

“An MS2-GFP fusion protein was used to tag transcripts as they were made. The transcript target, produced from a single-copy F plasmid, consists of the coding region for a red fluorescence protein, mRFP1, followed by a tandem array of 96 MS2 binding sites. The two components were under the control of inducible promoters. RNA transcripts were then induced, and samples were taken at different time points and imaged by fluorescence microscopy.”

In the image, Green foci are each one or more mRNA. Red color indicates gene product (RFP1).

(B) Detection of mRNA and protein in living cells. The picture is a false-colored overlay of the green and red channels. Scale bar, 1 µm.
mRNA dynamics appears at first to be Poisson, rising and saturating as usual.

These quantitative failures of the B-D model led to a bursting hypothesis:
* If each burst makes $k$ copies, but the burst-initiation rate is $k$ times slower than in the B-D model, then the first time course will look about the same and we retain that good agreement.
* But then the variance increases by a factor of $k^2$, so we fix the second graph.
* And the initial slope of the last graph also decreases by a factor of $k$, fixing it too.
Indeed, Golding et al. were able to observe “bursts” of mRNA synthesis directly:

Often we don’t remember the crucial role of statistical inference because it only served to motivate a more direct experiment. The new experiment gets all the glory, but often it *would never have been done* (or not till years later) without the kick in the butt from the indirect argument.
The starts and ends of bursts are exponentially distributed (Golding et al 2005):

\[ \ln(\text{estimated pdf} \times 1 \text{min}) \]

\[ t_{w,\text{stop}} \]

\[ t_{w,\text{start}} \]

\[ 0 \quad 50 \quad 100 \quad 150 \]

\[ t_w \quad [\text{min}] \]

And that led them to propose an almost-simple model:

... which required a stochastic simulation to confront with experiment... but which succeeded...

and whose underlying molecular mechanism wasn’t understood for several more years!
Life’s secret Secret

Everybody knows “the secret of Life is DNA,” right?

But it is less well appreciated that the stability of a molecule of DNA does not guarantee the accuracy of its replication and transcription. There is another big secret here, just as essential to Life as the well known ones.

I didn't mean this morning that this problem is "done"! I meant only that I had got a simulation of one particular approach, due to John Hopfield, which I thought delivers some insights into the model. Now it's time to see if that model captures the most important aspects of Nature's solution or not.
Imagine that you run an art museum and wish to find a mechanism that picks out Picasso lovers from among all your museum's visitors. You could open a door from the main hallway into a room with a Picasso painting. Visitors would wander in at random, but those who do not love Picasso would not remain as long as those who do. Thus, the concentration of Picasso lovers in the room would arrive at a steady value (with fluctuations, of course) that is enriched for the desired subpopulation.

To improve the enrichment factor further, you could hire an employee who occasionally closes the door to the main hallway, stopping the dilution of your enriched group by random visitors. Then open a new exit doorway onto an empty corridor. Some of the trapped visitors will gratefully escape, but die-hard Picasso lovers will still remain, leading to a second level of enrichment.

After an appropriate time has elapsed, you can then reward everyone still in the room with, say, tickets to visit the Picasso museum in Paris.

— Paraphrased from *An introduction to systems biology: Design principles of biological circuits* by Uri Alon

Let's try to apply this metaphor and see via simulation whether it really can explain (some of) the high fidelity of the ribosome.
the “second exit”!

Then shift right and go back to 0.

Then shift right and go back to 0.
The mechanism assumes, and there is evidence that,
- No GTPase activity until the complex is bound to a ribosome (the “clock doesn’t start” till then).
- After hydrolysis the complex can’t get back to 0 by swapping the GDP for an ambient GTP from solution.
- The last “commitment” step cannot happen until after GTP hydrolysis.

Correct binding can also speed up some forward steps, leading to further reduction in error rate. We will consider only Hopfield’s original idea, where the only selection was that incorrect binding speeds up the rejection steps. This has been confirmed in single-molecule experiments.

Recently some evidence of two sequential proofreading steps, but we will illustrate with just one.

We completely neglect the role of EF-G, not implicated in the high fidelity puzzle.
We want a stochastic simulation of these five states representing a single ribosome in a bath of W’s and C’s.

Isn’t that terribly inefficient?

Zuckerman,
http://physicallensonthecell.org/cell-biology-phenomena/active-kinetic-proofreading
Wrap

The ribosome is pretty complex, pretty evolved. I am not claiming that kinetic proofreading explains all (or even most of) its impressive accuracy. I am pointing out that:

• There are known steps corresponding to the ones I have imagined (as well as other steps I didn’t show).
• Those steps include initial selection, irreversible GTP hydrolysis, and secondary selection.
• They have now been seen in single-molecule studies. Their sequence is established in part by blocking one step and seeing that the later steps won’t happen either.
• All by themselves, without postulating anything more fancy (“induced fit” etc.), these simple steps yield a big enhancement in accuracy...
• As one can confirm by a simple simulation.
• The slowness of the final step (incorporation), crucial for this enhancement, has been observed in single-molecule studies.
Wrap the wrap

This is a whole *new kind* of modeling. It can go to places where analytic angels fear to tread. And it’s not (necessarily) hard.

But... um... what is physical modeling, anyway?
Don’t want to get all philosophical on you. I say, *It’s a Tetrahedron:*

```
Nf1=figure(1);
quiver(dm,theta,dmdot.*scaling,thetadot.*scaling,1);
% nullclines
hold on;
smalltg = [-1.2:.02:1.2];
```

“Yadda, yadda... feedback, yadda... bistability, hysteresis, yadda,... bifurcation...

From Nelson, *Physical models of living systems.*
Some of this material was taken from a recent textbook: 

**Physical models of living systems**

(www.physics.upenn.edu/biophys/PMLS).

Other bits are being written up: KY Chen, DM Zuckerman, PC Nelson (ask me).

Also see:


*A student’s guide to Python for physical modeling* by Jesse Kinder and PN (Princeton University Press, 2018).


For these slides see: [www.physics.upenn.edu/~pcn](http://www.physics.upenn.edu/~pcn)