Diffusion and cellular-level simulation

CS/CME/BioE/Biophys/BMI 279
Nov. 3 and 10, 2022
Ron Dror
Outline

• How do molecules move around in a cell?
• Diffusion as a random walk (particle-based perspective)
• Continuum view of diffusion
• Simulating diffusion
How do molecules move around in a cell?
• The interior of the cell is crowded, and all the molecules jiggle about.
• Note that lots of molecules (e.g., water) aren’t even shown in this movie.
Molecules jiggle about because other molecules keep bumping into them

https://www.youtube.com/watch?v=1jYabtziQZo
Diffusion

• This “jiggling about” by lots of molecules leads to diffusion
• Individual molecules follow a random walk, due to collisions with surrounding molecules
• Diffusion = many random walks by many molecules
  – Substance goes from region of high concentration to region of lower concentration
• We will focus on the basic case of random, unconfined, undirected motion. Certain molecules move around in more complicated ways within cells.

Can’t predict the specific path of an individual molecule. But you can predict how quickly substances will move from high concentration to low concentration.
Diffusion as a random walk (particle-based perspective)
Random walk

• We can model the motion of a molecule as a random walk
  – At each time step, randomly pick a direction, and move one unit in that direction
  – This type of motion (when caused by random collisions with other molecules) is called “Brownian motion”

In the movie, only cardinal directions are chosen, but we could pick diagonal directions as well and still get Brownian motion
1, 2, or 3 dimensions

- In biological systems, a random walk can take place in:
  - 3 dimensions: a protein moving freely within the interior of a cell
  - 2 dimensions: a protein moving within a cell membrane
  - 1 dimension: a protein (e.g., transcription factor) moving along a strand of DNA
Consider the 1D case (for simplicity)

- A particle starts at $x_0 = 0$
- At each time step, it has 50% probability of moving one unit forward, and 50% probability of moving one unit backward
- Denote the sequence of positions as $x_0, x_1, x_2, x_3, \ldots$
- Question: if you repeat this process many times and make a histogram of the position $x_3$, what will it look like? How about and $x_{10}$ or $x_{100}$?

The resulting histogram will show that it is more likely to end up at the positions closer to 0
Position after 3 time steps ($x_3$)

Position ($x_3$) | $(x_3)^2$
---|---
+3 | +9
+1 | +1
+1 | +1
−1 | +1
+1 | +1
−1 | +1
−3 | +9

$E[x_3] = 0$  
$E[x_3^2] = 3$

After $N$ steps:  
$E[x_N] = 0$  
$E[x_N^2] = N$

Histogram will be symmetric around position 0

As the number of time steps increases, the further you could be. This intuition matches what we see here where the expected value grows with the number of time steps.
Position after 3 time steps ($x_3$)

- **Probabilities:**
  - $P(x_3 = -3) = 1/8$
  - $P(x_3 = -1) = 3/8$
  - $P(x_3 = 1) = 3/8$
  - $P(x_3 = 3) = 1/8$

- **Mean displacement:**
  $E[x_3] = 0$

- **Mean-squared displacement:**
  $E[x_3^2] = 3$
Position after 10 time steps ($x_{10}$)

- Mean displacement: $E[x_{10}] = 0$
- Mean-squared displacement: $E[x_{10}^2] = 10$
Properties of 1D Brownian motion

• After $N$ steps:
  – Mean displacement: $E[x_N] = 0$
  – Mean-squared displacement: $E[x_N^2] = N$

• More generally, if the particle moves a distance $L$ at each time step, $E[x_N^2] = NL^2$

• As $N$ grows large, the distribution approaches a Gaussian (with mean 0 and variance $NL^2$)
Diffusion as a function of time

• Instead of thinking of position as a function of \( N \), we might think of it as a function of time.
  – Let \( t \) denote total elapsed time and \( \Delta t \) denote length of each time step. Then:

\[
E \left[ x(t)^2 \right] = E \left[ x_N^2 \right] = N L^2 = \frac{t}{\Delta t} L^2
\]

  – In other words, expected mean squared displacement grows linearly with time
Diffusion coefficient

Greater diffusion constant means molecule is moving faster

• To quantify speed of diffusion, we define the diffusion coefficient $D$:

$$ D = \frac{L^2}{2\Delta t} $$

Note: $L$ is average displacement per time step for each coordinate (x, y, or z)

• Then $E\left[ x(t)^2 \right] = 2Dt$ Grows linearly and rate is defined by diffusion constant

• In 2D, the diffusion coefficient is defined such that

$$ E\left[ r(t)^2 \right] = E\left[ x(t)^2 \right] + E\left[ y(t)^2 \right] = 4Dt $$

$r(t)$ is displacement from initial position at time $t$

• In 3D, $E\left[ r(t)^2 \right] = E\left[ x(t)^2 \right] + E\left[ y(t)^2 \right] + E\left[ z(t)^2 \right] = 6Dt$

• Larger molecules generally diffuse more slowly than small ones

Diffusion constant is affected by the type of surrounding solvent and the size of the molecule
Example values

• Diffusion coefficient ($D$):
  – Sugar: $500 \, \mu m^2/s$
  – Typical protein: $5 \, \mu m^2/s$

• Cell size:
  – Bacterium (E. coli): 1 \, \mu m radius
  – Human neutrophil (white blood cell): 10 \, \mu m radius
  – A human neuron can be 100 \, \mu m wide and, in extreme cases, over 1 m in length

From Chris Burge
(see links on course website)
Continuum view of diffusion
Basic intuition

• Although we can’t predict the motion of one particle, we can predict the average motion of a large number of particles
  – Particles will move from regions of high concentration to regions of low concentration
Fick’s law (or Fick’s 1st law)

• Suppose that particles are uniformly distributed in the \( y \) and \( z \) dimensions, and vary only in \( x \)
• Let \( c \) represent concentration (a function of \( x \))
• Define the flux \( J \) as the rate at which particles diffuse across a boundary
• Then Fick’s 1st law states that: 
\[
J = -D \frac{\partial c}{\partial x}
\]

Note: Fick’s “laws” are approximations assuming frequent collisions between particles (https://doi.org/10.1002/aic.14926)
Fick’s law (or Fick’s 1st law)

• Suppose that particles are uniformly distributed in the $y$ and $z$ dimensions, and vary only in $x$
• Let $c$ represent concentration (a function of $x$)
• Define the flux $J$ as the rate at which particles diffuse across a boundary
• Then Fick’s 1st law states that: $J = -D \frac{\partial c}{\partial x}$

![Diagram showing concentration distribution and flux directions](image)
How does concentration change with time?

• Now think of concentration and flux as a function of position $x$ and time $t$

• The concentration at a particular position goes up with time if there is less flux away from that position than there is coming in to that position (in other words, if the flux at that position is decreasing as one moves in the positive $x$ direction)

$$\frac{\partial c}{\partial t} = - \frac{\partial J}{\partial x}$$

[Diagram showing concentration increasing with time, large positive (rightward) flux, and concentration decreasing with time, small positive (rightward) flux]
Diffusion Equation (or Fick’s 2nd law)

• Combining these formulae gives us:

\[
\frac{\partial c}{\partial t} = -\frac{\partial J}{\partial x} = -\frac{\partial}{\partial x}\left(-D\frac{\partial c}{\partial x}\right) = D\frac{\partial^2 c}{\partial x^2}
\]

\[
\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2}
\]

Concentration decreasing with time

Large positive (rightward) flux

Small positive (rightward) flux

Concentration increasing with time
Example

- Diffusion from a point:
  - Solution to the diffusion equation is a Gaussian whose variance grows linearly with time
In three dimensions …

- Now suppose concentration varies as a function of $x$, $y$, $z$, and $t$
- The diffusion equation generalizes to:

\[
\frac{\partial c}{\partial t} = D\nabla^2 c = D \left( \frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} + \frac{\partial^2 c}{\partial z^2} \right)
\]

$\nabla^2$ is called the Laplacian operator
Simulating diffusion
Reaction-diffusion simulation

- *Reaction-diffusion simulation* is a common way to model how molecules move within the cell.

- **Basic rules:**
  - Molecules move around by diffusion.
  - When two molecules come close together, they have some probability of reacting to combine or modify one another.

- **Two implementation strategies:**
  - Particle-based
  - Continuum models
MCell: one of several particle-based simulation software packages

Other similar software packages: Smoldyn, Chemcell
How MCell works

• Particles representing molecules move according to a random walk, and react with one another probabilistically when they come into contact
  – MCell uses Monte Carlo algorithms

• Morphology of cell membranes (and other cellular structures) represented by a mesh

http://www.mcell.cnl.salk.edu/
MCell applications

• MCell has been widely used in neuroscience, to model phenomena such as synaptic transmission
• A common approach is to perform simulations under various assumptions and see which ones best match experimental data
  – See, for example, Coggan et al., Evidence for Ectopic Neurotransmission at a Neuronal Synapse, *Science* 309:446-451 (2005)
Continuum approach

• Divide space into finite “voxels”
• Instead of tracking positions of molecules, track concentrations of each type of molecule in each voxel
• At each time step, update concentrations based on reactions of molecules within a voxel, and diffusion between neighboring voxels based on concentration differences (i.e., the diffusion equation)
Continuum approach

- Advantage: faster
- Disadvantage: less accurate for small numbers of molecules
- Unlike the particle-based approach, the continuum approach is deterministic
- Example software: Simmune
Example: Gray-Scott model

Chemical A is added at a given "feed" rate.

Reaction: two Bs convert an A into B, as if B reproduces using A as food.

Chemical B is removed at a given "kill" rate.

Diffusion: both chemicals diffuse so uneven concentrations spread out across the grid, but A diffuses faster than B.

The system is approximated by using two numbers at each grid cell for the local concentrations of A and B.

You're not responsible for these details

http://www.karlsims.com/rd.html
Gray-Scott model

The grid is repeatedly updated using the following equations to update the concentrations of A and B in each cell, and model the behaviors described above.

\[
A' = A + (D_A \nabla^2 A - AB^2 + f(1-A)) \Delta t
\]
\[
B' = B + (D_B \nabla^2 B + AB^2 - (k+f)B) \Delta t
\]

**Feed:** at rate \( f \), scaled by \( 1-A \) so \( A \) doesn't exceed 1.0

"Delta t" is the change in time for each iteration. All the terms get scaled by this.

**Diffusion:** rates for A and B

These are 2D Laplacian functions, which give the difference between the average of nearby grid cells and this cell. This simulates diffusion because A and B become more like their neighbors.

**Kill:** this term is subtracted to remove B and scaled by B so it doesn't go below 0. \( f \) is added to \( k \) here so the resulting kill rate is never less than the feed rate.

**Reaction:** the chance that one A and two B will come together is \( A \times B \times B \). A is converted to B so this amount is subtracted from A and added to B.

You're not responsible for these details

http://www.karlsims.com/rd.html
Gray-Scott model

All sorts of interesting patterns emerge as one varies the parameters

Try it out at https://pmneila.github.io/jsexp/grayscott/

http://www.karlsims.com/rd.html
Alan Turing on morphogenesis

• Alan Turing proposed this as a model for pattern formation in animals

THE CHEMICAL BASIS OF MORPHOGENESIS

By A. M. TURING, F.R.S. University of Manchester

*(Received 9 November 1951—Revised 15 March 1952)*

It is suggested that a system of chemical substances, called morphogens, reacting together and diffusing through a tissue, is adequate to account for the main phenomena of morphogenesis. Such a system, although it may originally be quite homogeneous, may later develop a pattern or structure due to an instability of the homogeneous equilibrium, which is triggered off by random disturbances. Such reaction-diffusion systems are considered in some detail in the case of an isolated ring of cells, a mathematically convenient, though biologically unusual system.

You’re not responsible for this
Gray-Scott model

- Demo: