

2018

Brownian Motion in Cellular Receptors

Ouri Maler

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Maler, Ouri, "Brownian Motion in Cellular Receptors" (2018). *Graduate Theses, Dissertations, and Problem Reports*. 7212.

<https://researchrepository.wvu.edu/etd/7212>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Brownian Motion in Cellular Receptors

Ouri Maler

**Thesis submitted
to the Eberly College of Arts and Sciences
at West Virginia University**

in partial fulfillment of the requirements for the degree of

**Master of Science in
Applied Mathematics**

**Ádám M. Halász, Ph.D., Chair
Harvey Diamond, Ph.D.
Adrian Tudoraşcu, Ph.D.**

Department of Mathematics

**Morgantown, West Virginia
2018**

Keywords: Brownian motion, obstacles, cellular receptors

Copyright 2018 Ouri Maler

ABSTRACT

Brownian Motion in Cellular Receptors

Ouri Maler

In microbiology, chemical receptors on cellular membranes play a key role in a number of important reactions, defining when cells perform given roles and switch behaviors. These receptors are not static; their movements across the cellular membrane have been noted for their similarity to random Brownian motion. At the same time, some of these receptors only fire when clustered. We wish to establish a mathematical model for this phenomenon and check the model's validity by comparing its results to experimental data.

In part 1, we will be taking a closer look at the phenomenon being studied. In part 2, we will examine the proposed mathematical model, involving a combination of Brownian motion and asymmetrical obstacles. In part 3, we will explain in greater detail the goals we are striving toward with this project. In part 4, we will present the programs and methods used to simulate the model and analyze the resulting data. In part 5, we will look at our results to date. Finally, in part 6, we will explain our preliminary conclusions and our outlook from here on out.

Table of Contents

1. Background: The Biochemical Matter	1
1.1 Cells as chemical factories	1
1.2 Cell Signaling	2
2. Background: Mathematical Aspects of Brownian Motion	5
2.1 Brownian Motion (BM) and Diffusion	5
2.2 Brownian Motion: displacements statistics	7
3. Goals of the project	10
4. Methods	11
4.1 Brownian Motion Simulator	11
4.2 Analysis Modules	14
5. Results	21
5.1 Standard Analysis of Simulation Results	21
5.2 Kolmogorov-Smirnov Fitting of Square Displacement Distributions	23
5.3 Exploration of Localization Density with Random Obstacles	24
6. Preliminary Conclusions and Outlook	29
Bibliography	30

List of Figures

1. Brownian trajectories generated by our simulator	14
2. Mean square displacement (MSD) over different time interval lengths	15
3. Distribution of square displacements from our BM simulations	17
4. Illustration of the Kolmogorov-Smirnov (KS) statistic	19
5. Diffusion coefficient from BM simulations by minimizing the KS statistic	20
6. Diffusion coefficients estimated for different obstacle sets by KS minimization	21
7. Distribution of square displacements	22
8. Effective diffusion coefficients for the same runs as the previous figure	23
9. Trajectories and localization density for two biased obstacles	25
10. Trajectories and localization density for two bilateral obstacles	25
11. Trajectories and localization density for a random landscape of bilateral obstacles	26
12. Effective diffusion coefficients for different sampling times	27
13. Trajectories and localization density for a random landscape of biased obstacles	28

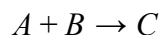
1. Background: The Biochemical Matter

At their root, the mathematical questions studied in this paper originate from attempts to comprehend certain aspects of cellular biochemistry. Before we examine the math, let us look at the real-world issue behind it.

1.1 Cells as chemical factories

Processes in living cells can be characterized in terms of chemical processes involving complex molecular entities. Those processes, called "reactions", revolve around the conversion of types of molecules. Reactions can take the form of 1)Association, wherein two or more molecules of given types merge together to form a molecule of another type, 2)Dissociation, wherein one molecule breaks down into several smaller ones, 3)Conformation changes, wherein a molecule retains its component atoms, but changes the geometric configuration in which they are held together, potentially changing the molecule's properties in radical ways, and 4)Translocation, wherein a molecule moves from one part of the cell to another (for instance, an RNA molecule leaving the cellular nucleus).

If cells are to be viewed as chemical factories, then the set of reactions that regiment them are traditionally modelled as Chemical Reaction Networks (CRN) [3]. Under the CRN model, we have 1)A number of *species*, each corresponding a type of molecule (where two identical molecules can be considered as different species if location or another factor causes them to behave differently), 2)A *state*, which is the amount (or in most instances the density) of each species at a given time, and 3)A set of reactions, each one corresponding to a state change. For instance, the following reaction:



In this reaction, a molecule of species A and a molecule of species B merge together to form a molecule of species C. This reaction affects the state by reducing the density of A and B while raising the density of C.

Under the CRN model, the cell is treated as an Ordinary Differential Equation system (ODE) [2][6]. The rate at which each reaction happens is equal to a constant multiplied by the densities of the component species, and the density of each species changes at a rate proportional to the rate of the reactions involving it. The resulting ODE tracks the rate of change of all species.

The CRN model is very commonly used, including by the pharmaceutical industry. It is a handy vehicle for summarizing knowledge about the biochemistry of organisms, ranging from humans to the (relatively) simpler viruses. However, the attempt of the CRN model to understand the cell's working using ODEs runs into a number of problems:

The first problem is the sheer, overwhelming complexity of the cell. Any given cell may contain, at a conservative estimate, thousands of different molecule species - there are, after all, around 20,000 protein-coding genes in the DNA present in each and every cellular nucleus in humans. An ODE with thousands of dimensions can be de facto impervious to attempts to resolve it through calculation.

The second problem is that, as useful as the ODE model can be, it is also an oversimplification. It does not address the spatial structure of the cell, when a molecule's behavior may be radically altered by its location. It does not account for the fact that the molecules of a given species are not uniformly distributed within the cell, or even within the part of the cell where they reside. Last but not least, while some molecule species are present in massive numbers in the cell, others might only exist as a small handful of copies; if too few molecules of a species exist, then the CRN approach of treating their behavior as a deterministic ODE breaks down; a stochastic, probabilistic approach becomes more appropriate.

1.2 Cell Signaling

The cells that form the human body and other such multicellular organisms must, naturally, coordinate. Now, every cell with a nucleus contains within itself the entirety of the human DNA, which describes all possible cellular behavior - however, each given cell will only perform the behavior required by its position and the current needs of the organism as a whole. What the cell does at any given moment is defined in no small part by its reactions to outside information received via **signaling**.

In biochemical signaling, the "message" is often going to be a type of molecule that circulates between cells. Any given cell type may produce and receive many different signal types; once messenger molecules are received, *cellular signal processing* is needed to translate the received signals into a *response*. This involves complex signaling pathways within the cells that are not, to date, completely understood.

It should be noted here that many illnesses - diabetes, autoimmune diseases, certain cancers - are believed to result from errors in signaling. Type 2 diabetes, for instance, is a condition in which cells fail to respond to the insulin (the signal) as they are supposed to [12].

Cellular signals and receptors are a vast field. One particular family of such is of interest to us here: The **receptor-ligand**. Ligand molecules act as the signal, binding to a specialized counterpart in the cell - the receptor. The receptors are confined to the cellular membrane, binding to ligands from the extracellular space.

One example of ligand is the Vascular Endothelial Growth Factor (VEGF) [5][9] and its receptor (VEGFR), a protein that stimulates the growth of blood vessels. VEGF is of interest to cancer research: A cancerous tumor that grows wildly will starve itself of oxygen and nutrients unless it gets irrigated by blood circulation. As such, whether cancer cells generate sufficient amounts of VEGF can make the difference between a benign tumor and a malignant one.

Ligand receptors, far from being static, "float" across the cellular membrane. One key aspect of their behavior is that they are also able to bond with each other when they get sufficiently close; those aggregates of receptors are called oligomers. Notably, the binding between two receptors is significantly stronger in the presence of ligands. Just as importantly, it is only when

large oligomers form that their intracellular domains interact, cross-activate, and induce processes inside the cell, continuing the signaling machinery.

What this means, then, is that the ligand-based signalling process is affected by the movement of the receptors on the cellular membrane - by anything that affects the likelihood and frequency of multiple receptors aggregating. We wish, then, to take a closer look at factors affecting receptor mobility on the cellular membrane - anything facilitating or limiting receptor-receptor binding.

Single Particle Tracking (SPT) [8] is a technique that allows for the observation of individual receptors over time, and has provided large amounts of experimental data on their movement. While they generally seem to perform typical Brownian motion, their clustering patterns suggest deviations from that model, which are thought to be the result of obstacles to free movement. There are two types of obstacles that have been theorized: Actin filaments, which would act as linear barriers [1], and trapping regions (possibly composed of "lipid rafts") [4], which would be easy for receptors to enter but difficult to leave.

Our goal here is to help understand how movement and clustering patterns emerge from barriers and trapping regions. Before that, however, we need to take a quick look at the mathematics behind Brownian motion.

2. Background: Mathematical Aspects of Brownian Motion

2.1 Brownian Motion (BM) and Diffusion

The position of a Brownian particle is a random variable. If we know for certain the position $\vec{r}_0 = (x_0, y_0)$ at time $t = 0$, then the position $\vec{r}(t) = (x, y)$ at a later time $t > 0$ is distributed according to the *probability density function* (PDF):

$$f(x, y; t) = \frac{1}{4\pi Dt} \exp\left(-\frac{(x-x_0)^2 + (y-y_0)^2}{4Dt}\right)$$

where D is the diffusion coefficient. Note that the above PDF is simply the product of two normal distributions, one for each direction, with variance $\sigma^2 = 2Dt$. Thus, the displacement in the X direction, $\Delta x = x - x_0$, is distributed normally around zero, and the expectation of Δx^2 is equal to the variance $\sigma^2 = 2Dt$. In other words, while the value of the X coordinate at any future time $t > 0$ is not known precisely, it will be distributed symmetrically around the initial value x_0 following a normal distribution with variance proportional to the time t . The same considerations apply to the Y direction.

Note that the PDF is *normalized* so that the total probability of finding the particle *anywhere* in the XY plane is always 1:

$$\int_0^\infty \int_0^\infty f(x, y; t) dx dy = \frac{1}{4\pi Dt} \int_{-\infty}^\infty \int_{-\infty}^\infty dx dy e^{-\frac{(x^2+y^2)}{4Dt}} = \dots = 1$$

More generally, we may think of the cumulative PDF for the position of any number of particles, as the sum of individual PDFs:

$$\rho(x, y; t) = \sum_{k=0}^N f_j(x, y; t) = \frac{1}{4\pi Dt} \sum_{j=0}^N \exp\left(-\frac{(x-x_{0j})^2 + (y-y_{0j})^2}{4Dt}\right)$$

where (x_j, y_j) represent the initial coordinates of particle j . In this case, the cumulative PDF $\rho(x, y; t)$ acts as a joint probability density; its integral over any finite area $W \subset R^2$ gives the expected number of particles inside W .

A little further reasoning leads us to the following property:

$$\rho(x, y; t) = \frac{1}{4\pi D(t-t')} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp\left(-\frac{(x-x')^2 + (y-y')^2}{4D(t-t')}\right) \rho(x', y'; t') dx' dy'$$

If we know the cumulative density of a set of such particles at a given time, we can use the above to compute the PDF at some future time as the convolution of the past density with the displacement PDF.

This formula also provides an important connection to the *continuum picture*. The quantity $\rho(x, y; t)$ is normalized to the total number of particles in the system:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \rho(x, y; t) dx dy = N$$

As the number of particles increases, it is reasonable to interpret $\rho(x, y; t)$ as the “number density”, the physical density of particles expressed in units of particle count per unit area. Importantly, one can easily show (by direct computation) that $\rho(x, y; t)$ as defined by the convolution formula verifies the *diffusion equation* in two dimensions:

$$\frac{\partial \rho}{\partial t} = \frac{\partial^2 \rho}{\partial x^2} + \frac{\partial^2 \rho}{\partial y^2}$$

Thus, Brownian motion is equivalent to classical diffusion in the sense that the PDF of a set of Brownian particles verifies the classical (Fickian) diffusion equation for the same diffusion coefficient. In other words, the *localization density* of a set of discrete, point-like Brownian particles behaves *exactly as we imagine the continuum* representing the density of a diffusing

substance. This is a remarkable (if very old) result, and is the basis for our simulation based approach to the diffusion of biomolecules.

2.2 Brownian Motion: displacements statistics

The PDF for the next position of a Brownian particle is the mathematical basis of our simulation algorithm. It is also the starting point in the derivation of statistical measures that are commonly used to compare the movement patterns of particles to Brownian motion. These measures typically involve the *displacements* connecting two consecutive positions of the same particle.

By contrast with classical (also called Fickian or “normal”) diffusion, consistent with Brownian motion, random movements that deviate from BM are called “anomalous”. As with all things “normal”, it turns out that true Brownian motion is quite uncommon in Nature, and most biologically relevant molecules perform one form or another of anomalous diffusion. Still, the Brownian motion approach is very useful in providing the basic context of the discussion and also as a reference to quantify the actual behaviors.

Brownian Motion: Displacement Statistics

In the following we focus on particles that move more or less randomly in two dimensions. The position vector at some specific time t_0 consists of the corresponding X and Y coordinates, $\vec{r}(t_0) = (x(t_0), y(t_0))$. The change in the position vector at some later time $t_1 = t_0 + \Delta t$ is the *displacement*,

$$\vec{R}(\Delta t, t_0) = \vec{r}(t_1) - \vec{r}(t_0) = (\Delta x(\Delta t, t_0), \Delta y(\Delta t, t_0)) \dots$$

Since we are dealing with stochastic behavior, we want to be able to collect large samples. This is why it is useful to look at the displacement over a fixed length of time (the Δt above), and collect samples for the same Δt , starting from many initial times t_0 , *for the same particle*. Of course, when we are dealing with several particles (whose movement is not *a priori* known to follow different dynamics), we may collect displacements from different particles / trajectories.

To simplify the notation, we drop the second argument (the initial time) and use $\vec{R}(\Delta t)$ to represent the displacement vector over some time interval of length Δt , during the motion of some particle from the group of interest. In this context, we will rewrite the PDF and also derive (or state) probability density functions for the displacement vector and its magnitude. The joint PDF for the components of the displacement vector $\Delta x, \Delta y$ is the same as stated in the beginning of this section:

$$f_{XY}(\Delta x, \Delta y) = \frac{1}{4\pi D\Delta t} \cdot e^{-(\Delta x^2 + \Delta y^2)/(4D\Delta t)}$$

The corresponding expectation of $\Delta x, \Delta y$ is zero, and that of the squares is $\sigma^2 = 2D\Delta t$.

The **PDF for the scalar displacement** $R = |\vec{R}| = \sqrt{\Delta x^2 + \Delta y^2}$ can be obtained by changing to polar coordinates in the normalization integral (we drop the Δ to simplify the notation):

$$1 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{XY}(x, y) dx dy = \int_0^{\infty} R dR \int_0^{2\pi} d\phi \frac{1}{4\pi D\Delta t} \cdot e^{-R^2/(4D\Delta t)} = \frac{1}{2D\Delta t} \cdot \int_0^{\infty} R dR e^{-\frac{R^2}{4D\Delta t}}$$

In summary:

$$R(\Delta t) \equiv |\vec{R}(\Delta t)| \rightarrow f_R(R) = \frac{1}{2D\Delta t} \cdot R \cdot e^{-R^2/(4D\Delta t)}$$

The **PDF for the square displacement** $R^2 = \Delta x^2 + \Delta y^2$ follows either from integrating over the joint PDF for $\Delta x, \Delta y$, or more easily, by changing variables in the normalization integral to $S \equiv R^2 \rightarrow dS = 2RdR$

$$1 = \frac{1}{2D\Delta t} \cdot \int_0^{\infty} R dR e^{-\frac{R^2}{4D\Delta t}} = \frac{1}{4D\Delta t} \cdot \int_0^{\infty} dS e^{-S/(4D\Delta t)}$$

In summary:

$$S(\Delta t) \equiv R(\Delta t)^2 = |\vec{R}(\Delta t)|^2 \rightarrow f_S(S) = \frac{1}{4D\Delta t} \cdot e^{-S/(4D\Delta t)}$$

Comparing Experimental Movement to Brownian Statistics

Typically, we have access to experimentally derived *trajectories*, which are time-ordered lists of the positions (coordinates) of a particle at different moments in time. The positions are obtained from recordings (movies) that consists of images taken at fixed time intervals. Thus, in two spatial dimensions, we have an array or table of entries of the form $\{t_k, x_k, y_k\}_{k=1 \dots N}$ where the recording times are increasing $t_0 < t_1 < \dots < t_k < t_{k+1} \dots$; more often than not, the recordings are made at a fixed time interval, $t_{k+1} = t_k + \Delta t$.

Displacements can be obtained from a given trajectory by selecting ordered pairs $\{k, j\}$, $t_k > t_j$. It is useful to note that, given a trajectory $\{t_0, \vec{r}_0\}, \dots, \{t_N, \vec{r}_N\}$ obtained using a fixed time step, we may extract sets of displacements corresponding to one or more time steps:

$$\Delta t : \{1, 0\}, \{2, 1\}, \{3, 2\}, \dots \{N, N-1\}$$

$$2\Delta t : \{2, 0\}, \{3, 1\}, \{4, 2\}, \dots \{N, N-2\}$$

$$k\Delta t : \{k, 0\}, \{k+1, 1\}, \{k+2, 2\}, \dots \{N, N-k\}$$

The following discussion refers to sets of displacements corresponding to the same *time difference*.

The Brownian displacement and square displacement follow different distributions, as we have seen. Both can be used to analyze experimental data, by comparing a sample of displacement values (corresponding to the same time) to the analytical results.

Importantly, the distribution of the SD starts from zero (at $r = 0$), has a peak at () and a Gaussian ‘tail’ as $r \rightarrow \infty$; the square displacement $S = r^2$ is distributed exponentially. It is more practical because of its simple functional form, and the fact that, plotted on a semi-logarithmic scale ($\ln(f_s(r^2))$ vs. r^2) it is a straight line with slope $\ln(f_s(S))' = -4Dt$.

The most common measure is the *mean square displacement* (MSD). From the PDF, we have

$$\langle r^2 \rangle = \int_0^{\infty} S f_s(S) dS = \int_0^{\infty} \frac{1}{4Dt} \cdot S \cdot e^{-S/4Dt} dS = 4Dt$$

Here one would compare the mean square displacement computed for different time values and the dependence should be linear.

3. Goals of the project

The purpose of our endeavour here is to gain a clearer understanding of how receptor movement and clustering patterns on the cellular membrane emerge from barriers and trapping regions. SPT already provides us with raw experimental information on the movement of tagged receptors, but does not provide a map of the underlying membrane landscape; it is our hope that we may provide a framework through which the SPT data may be interpreted.

With that goal in mind, we intend to simulate Brownian motion in the presence of various landscape configurations; those configurations are meant to simulate both actin filaments that block movement, and trapping domains that result in asymmetric blocking (that is, where moving in one direction is harder than in the opposite). Once the simulator works, we wish to develop methods via which landscape features may be identified through analysis of observed movement; applying those methods to SPT data will hopefully tell us more about the cellular membrane and how ligand receptors on it behave.

4. Methods

In this section, we will go over the software and mathematical tools that we have been using in this project. Our software has two major functions: (1) simulation of trajectories under various scenarios (2) analysis of trajectory data.

4.1 Brownian Motion Simulator

Algorithm

We use a first-principles approach [10], in that our motion simulation algorithm emulates Brownian motion using the exact expression for displacements. We follow a fixed number of particles, N_p , and generate their positions at consecutive times t_0, t_1, \dots using a fixed time step, Δt .

Each particle is represented by a two dimensional position vector, which evolves over time reflecting the movements of the particle. For instance, the position vector of the k -th particle is $\vec{r}^{(k)}(t) = (x^{(k)}(t), y^{(k)}(t))$. It is helpful to point out the distinction between integers used to identify the discrete times (when the updates occur) and the labeling of the different particles. The time at time step j is $t_j = t_0 + j \cdot \Delta t$.

The **state** of the (simulated) system at a given time t_j consists of the XY coordinates of the set of N_p particles:

$$\{\vec{r}^{(k)}(t_j)\}_{k=1 \dots N} \ ; \ \vec{r}^{(k)}(t_j) = (x^{(k)}(t_j), y^{(k)}(t_j)) \ ; \ k = 1, 2, \dots, N$$

When there is no risk of confusion, one may use double indices: $\vec{r}^{(k)}(t_j) = (x_j^{(k)}, y_j^{(k)})$.

Each **update** consists of adding displacements $\Delta x, \Delta y$ to the positions of each particle. Each displacement value is a random number generated consistent with the PDF

$$f_1(\xi) = \frac{1}{\sqrt{4D\Delta t}} \cdot \exp\left(-(\xi^2)/(4D\Delta t)\right)$$

which is equivalent to using the joint PDF

$$f(\Delta x, \Delta y) = \frac{1}{4D\Delta t} \cdot \exp\left(-\frac{\Delta x^2 + \Delta y^2}{4D\Delta t}\right).$$

At every update, the system time is advanced by the amount of one step; for each of the N particles, the X and Y coordinates are changed by amounts Δx , respectively Δy (generated as described above):

$$t \rightarrow t + \Delta t; (x^{(k)}, y^{(k)}) \rightarrow (x^{(k)} + \Delta x, y^{(k)} + \Delta y), \quad \forall k = 1, 2, \dots, N_p$$

Simulation boundaries are necessary in order to have a finite simulation area and avoid the loss of particles. We use a rectangular simulation space defined by intervals in the X and Y directions, $[X_A, X_B] \times [Y_A, Y_B]$ (where $X_A < X_B$, $Y_A < Y_B$). We require that the coordinates of all simulated particles satisfy $(x^{(k)}, y^{(k)}) \in [X_A, X_B] \times [Y_A, Y_B]$ that is

$$X_A \leq x^{(k)}(t_j) \leq X_B, \quad Y_A \leq y^{(k)}(t_j) \leq Y_B, \quad \forall j, k.$$

Assuming that the condition is satisfied at a given time t_j , we ensure that it stays that way at the next time step, by *rejecting* position updates that would take the particle out of the box (i.e. violate the inequalities). During each update, after generating the new positions for each particle, we identify the particles whose “new” position would be outside the box; the positions of these particles are then re-set to the previous one.

This approach is equivalent to *reflective boundary conditions*. It is slightly more imprecise than alternatives where the displacement vector is reflected by the boundary; the discrepancy vanishes in the limit $\Delta t \rightarrow 0$. The geometry of the boundaries is such that all points inside or on the rectangular boundary have at least a $1/4$ chance of moving to an acceptable position.

Obstacles are treated similarly to the system boundaries. They are represented by directed line segments \vec{AB} , which partially block the movement of simulated particles. A proposed move

of a particle **may** be rejected if the line segment going from the original (“before”) to the newly generated (“after”) position of a particle intersects the line segment representing the obstacle. We found it useful to make the obstacles block in only one direction, and have a (possibly) non zero crossing probability. First, only moves (proposed displacement vectors) that cross from left to right are blocked. Second, even from among these, moves are accepted with a probability $p_{cross} \geq 0$ as follows: for each proposed move that would cross the obstacle, we generate a random number $r \in [0, 1]$, and *accept* the move if and only if $r \leq p_{cross}$.

Implementation

Using the Python programming language, we designed a program that simulates Brownian motion by making a number of objects perform a random movement at regular time intervals.

The program is given a number of parameters. The first is the diffusion coefficient D , representing distance-per-time. Another one is the time length of the simulation. Another one is the number of time steps taken during the simulation (each time step has a length Δt equal to the total length of the simulation divided by the total number of steps). One is the number of objects (representing the receptors).

The last parameter is a file describing the landscape in which the simulation takes place by listing obstacles. Each obstacle is treated as a segment in the landscape that has a given probability of nullifying movement that goes through it. Obstacles can be bilateral (meaning that they block movement regardless of direction) or one-way (meaning that they only stop movement going in one direction). Another program can be used to randomly generate landscape files; once given a number of obstacles to generate, it will assign them random sizes and positions in the landscape, making them bilateral or one-way as requested.

During runtime, the simulator places each sample object in a random position within the landscape. After that, for each time step, each individual sample object is given a random direction and a distance (which is equal to a random number $r \in [0, 1]$, multiplied by D and the

length Δt of the time step); if none of the obstacles in the landscape blocks its movement, then it moves that distance in the selected direction.

For every sample object, the program generates a trajectory file that lists its successive positions at each time step. It is those trajectory files that we will perform our analysis on.

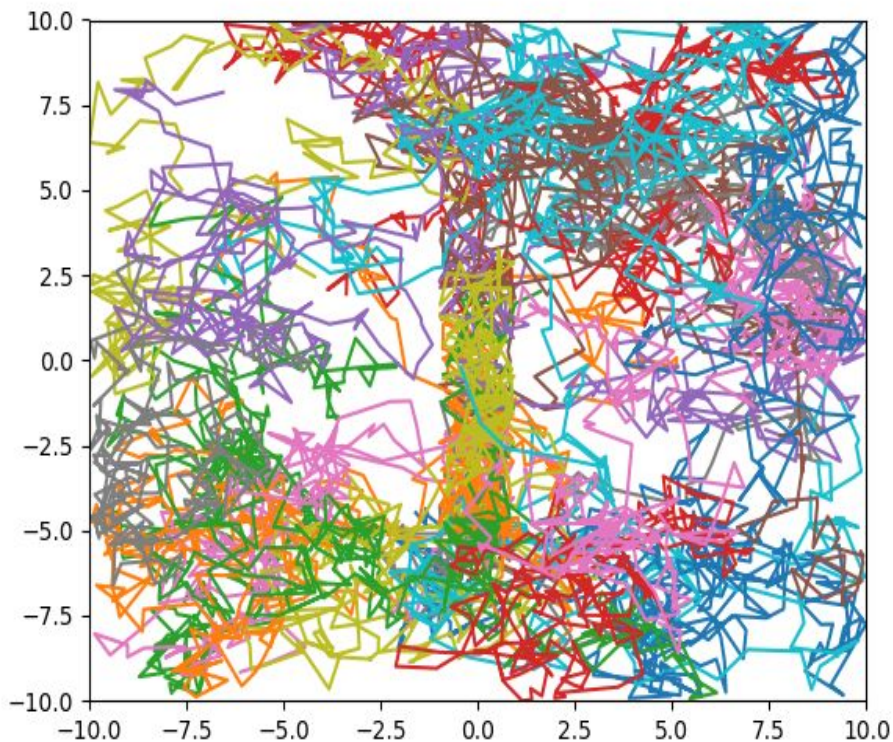


Figure 1: Brownian trajectories generated by our simulator. The simulation box has reflective boundary conditions and ranges from -10 to 10 in both directions. The landscape in this instance includes two obstacles positioned in parallel to each other.

4.2 Analysis Modules

The goal of the project is to develop analysis protocols that can be used to extract useful information from experimentally obtained trajectories. The trajectory files generated by the simulation module are proxies for the “real” data to be analyzed. Once we have generated the trajectory files, we can use several different programs to analyze them.

Traditional MSD estimation

The first of these programs performs an analysis of **mean square displacement**. As discussed in part 2, under ideal Brownian motion, the mean square displacement (MSD) is

$$\langle R^2 \rangle = 4D\Delta t$$

Here, R represents the displacements (distance traveled by particles over a given time); D is the diffusion coefficient and Δt is the length of the time used to sample the displacements.

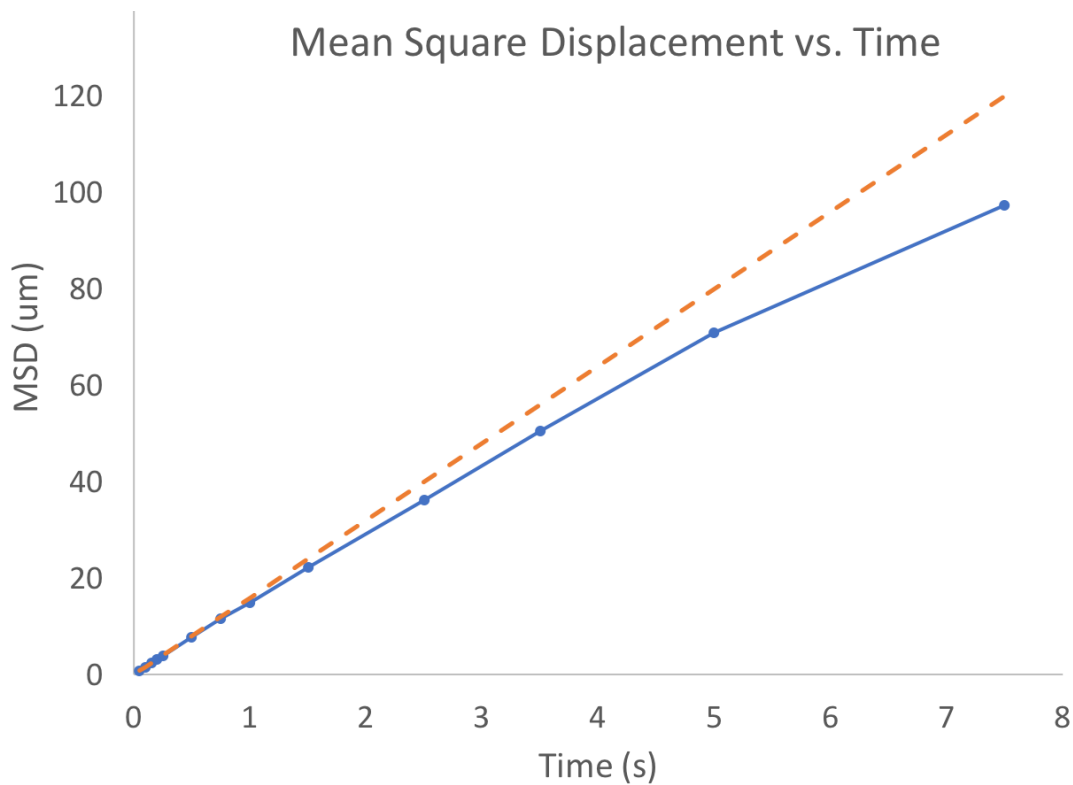


Figure 2: Mean square displacement (MSD) over different time interval lengths. The simulation result (blue) is close to the theoretical dependence (red), but the discrepancy increases with time.

An actual simulation naturally gives results that will not precisely match the mathematical theory, but serves as an approximation. Multiple simulations were run with a diffusion coefficient of $D = 4.0$ and a varying set of obstacles, and otherwise similar conditions. The simulation time step was $\Delta t^{(sim)} = 0.05$ s. Displacements were collected for several numbers of

time steps. The resulting MSD is shown in Figure 2 (blue line / points) and it is reasonably close to the theoretical dependence. As the time used to estimate the displacement increases, the simulation result falls behind. This most likely reflects the finite size of the simulation box.

Square Displacement Statistics

The second program used to analyze the trajectory data compares the **distribution of the square displacements** to the theoretical distribution it ought to have under ideal Brownian motion, creating two corresponding histograms and superposing them. This is similar to the mean square displacement analysis, but it is much more detailed since it looks at the entire distribution, not only its expectation (mean).

As discussed previously, the expected distribution of **square displacements** is exponential:

$$f_S(R^2) = \frac{1}{4D\Delta t} \exp\left(-\frac{R^2}{4D\Delta t}\right)$$

The analysis program collects displacements from a set of trajectories and plots a histogram that is compared to the theoretical distribution.

The histogram of actual square displacement is generated by recovering and ordering by magnitude the list of square displacements from the trajectory files, partitioning them into magnitude categories (for example, the group of square displacements between 0 and 0.3, the group between 0.3 and 0.6...), and then counting how many individual displacements fit into each category.

The histogram of theoretical displacement uses the above distribution formula. It partitions square displacement by magnitude, then, for each category in the partition, integrating the above PDF to get the odds of any one movement falling into any interval $[S_{min}, S_{max}]$ in the partition:

$$P(S_{min} \leq R^2 \leq S_{max}) = \exp\left(\frac{-S_{min}}{4D\Delta t}\right) - \exp\left(\frac{-S_{max}}{4D\Delta t}\right)$$

Once the program has thus calculated the probability of any square displacement falling under a given magnitude category of the partition, it multiplies that probability by the total

number of displacements in all the trajectory files. The result is then the number of displacements that should, under theoretical, ideal Brownian motion, fall under that category. The program then performs these calculations for the entire partition, generating the histogram of MSD for ideal Brownian motion. Once the two histograms have been generated, they are superimposed to make for easy visual comparison.

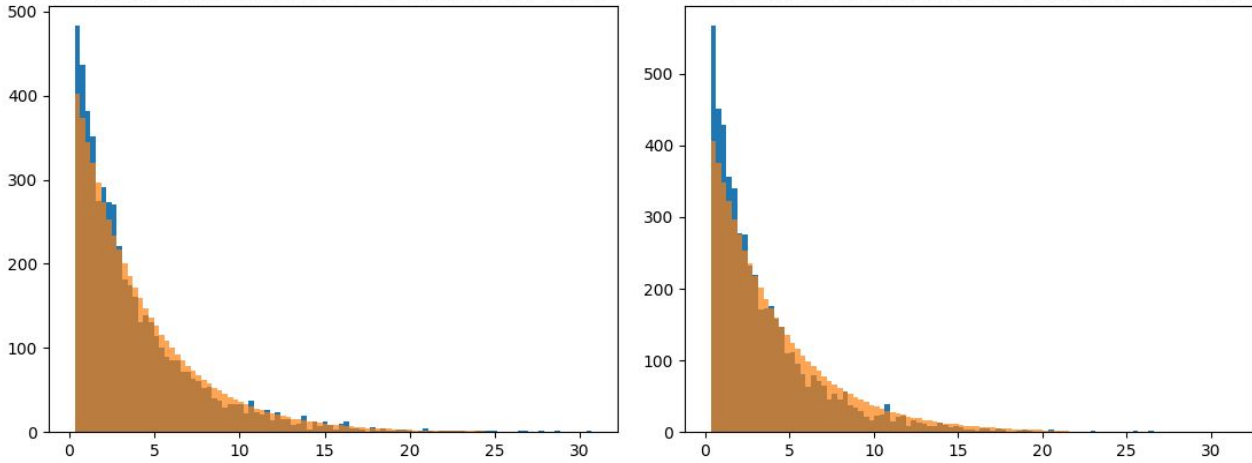


Figure 3: *Distribution of square displacements from our BM simulations (blue bars) compared to the expected distributions (yellow bars). The left panel is from a simulation with no obstacles, the right hand one has two linear obstacles; the X-axis represents square displacement in square microns, while the Y-axis represents the number of motions. The simulation is performed with 30 samples, 200 time steps of 50 milliseconds each, and a diffusion coefficient of 4.*

Fitting Distribution Parameters with Kolmogorov-Smirnov Statistic.

With the third data analysis program, we seek an actual method of quantifying the difference between the expected and actual distribution. For that purpose, we have chosen to employ the Kolmogorov-Smirnov (KS) [7][11] statistic, which is often used to test whether a given sample could come from a distribution or if two given samples could have come from the same distribution.

Kolmogorov-Smirnov Statistic

We want to compare a sample of values (X_1, X_2, \dots) with a known theoretical distribution function $f(x)$. The idea is to compare the cumulative distribution function $g(x)$ corresponding to $f(x)$ with a similar function constructed from the sample values.

The *cumulative distribution function* (or CDF) $g(x)$ is the probability that a number X randomly selected according to the true distribution function $f(x)$ is less than the value x where the CDF is evaluated:

$$g(x) \equiv P(X < x \mid X \text{ has PDF } f(x))$$

The CDF is simply the integral of the PDF, from a value m less than or equal to the smallest possible value of the random variable to the value of interest x . This identity follows from the definition of the PDF (if Ω is the set of all possible values of the random variable X , the probability that X is in a subset $W \subset \Omega$ is the integral of the PDF over W):

$$g(x) = P(X < x) = \int_m^x f(\xi) d\xi \quad \text{with } m \text{ s.t. } P(X \leq m) = 0$$

The theoretical CDF is to be compared to the step function $h(x \mid \{X_k\})$ which is the fraction of numbers X_k in the sample that are less than x :

$$h(x \mid \{X_k\}) = \frac{1}{N} \sum_{k=1}^N H(x - X_k)$$

Here, N denotes the size of the sample, and $H(\xi)$ stands for the Heaviside function (defined as $H(\xi) = 1$ if $\xi \geq 0$ and $H(\xi) = 0$ otherwise). A practical way to define $h(x \mid \{X_k\})$ is to sort the sample in increasing order: $X_1 \leq X_2 \leq \dots \leq X_N$ and identify the largest value of the index k such that $X_k \leq x$:

$$h(x \mid \{X_k\}) = \max_{X_k \leq x} (k) / N$$

Given $g(x)$ and $h(x \mid \{X_k\})$ as described, we want to characterize the discrepancy by a measure $M(\{X_k\})$. There are many ways to do this, as long as $M(X_1, X_2, \dots) \geq 0$, $M = 0$ if and only if $g(x) \equiv h(x)$, and it increases as the discrepancy between the sample and the theoretical PDF gets worse.

The Kolmogorov-Smirnov (KS) statistic is defined as the *largest absolute value of the difference* between the CDF $g(x)$ and the step function $h(x)$, encountered over the entire range of possible values of x

$$K = \max_x |g(x) - h(x|\{X_k\})|$$

The diagram in Figure 4 illustrates the idea of the KS statistic.

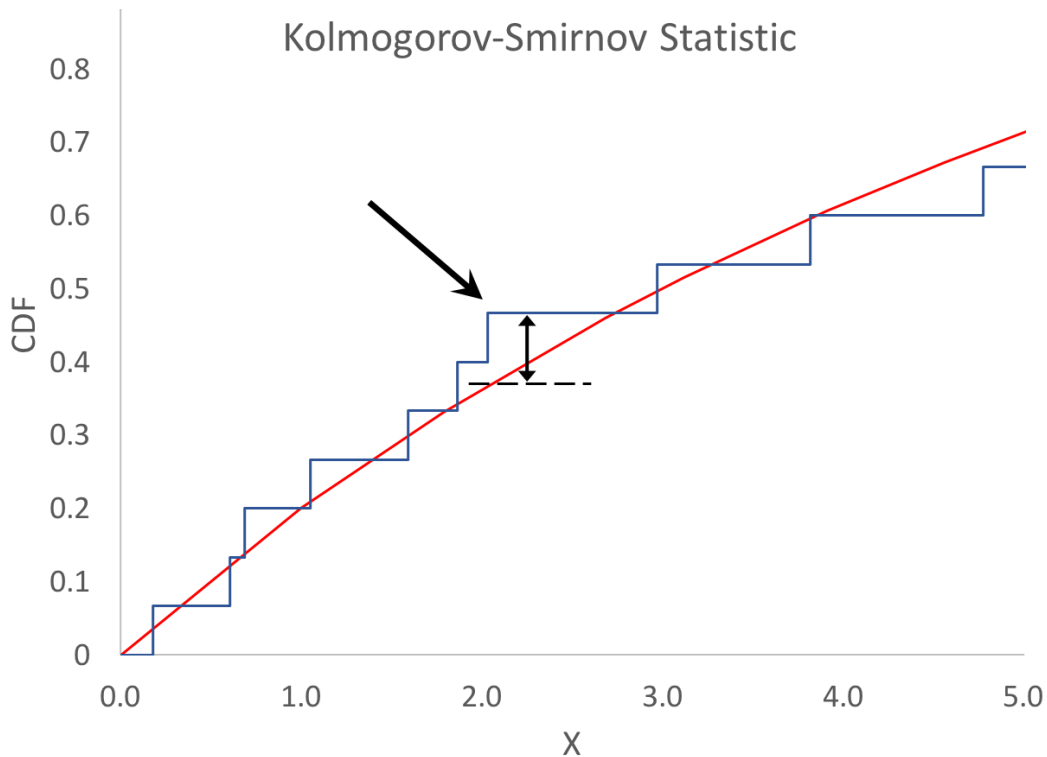


Figure 4: Illustration of the Kolmogorov-Smirnov (KS) statistic. The maximum difference between the step function (blue) and the theoretical CDF (red) is the K-S statistic.

Using the K-S Statistic for Parameter Estimation

The Kolmogorov-Smirnov statistic is more often than not used to evaluate the likelihood that a sample comes from a known distribution, or whether two samples come from the same distribution. We are not concerned with that here, at least not primarily. We **know** that our displacements are generated consistently with Brownian motion, with a value of the diffusion constant that we set. The K-S statistic may serve as a consistency check. However, as we

simulate Brownian motion in a rectangular “box”, and later, in the presence of a complex landscape of linear obstacles, we also **know** that our displacement distributions will deviate from Brownian motion.

Our goal is to investigate **how** the movement of particles deviates from ideal in the presence of a barrier landscape, and we plan to characterize it with various empirically derived distribution functions that we hope reflect the underlying landscape. The K-S statistic is a single positive number that characterizes the match between such a proposed analytical function and experimentally obtained samples; we plan to use it to estimate the “best-fit” parameters by minimising the K-S statistic. As a proof of principle, we did this to derive an “effective” diffusion coefficient. Our fitting program minimizes the K-S statistic corresponding to a set (sample) of displacements and the analytical form of the CDF for Brownian motion; it returns the value of the diffusion coefficient D that corresponds to the best fit (i.e. minimal K-S statistic). As a consistency check, we show the results of this fit for two sets of Brownian motion simulations (in the $[-10, 10] \times [-10, 10]$ simulation box), performed with diffusion coefficient $D_0 = 4.0 \mu\text{m}^2/\text{s}$, without and with obstacles:

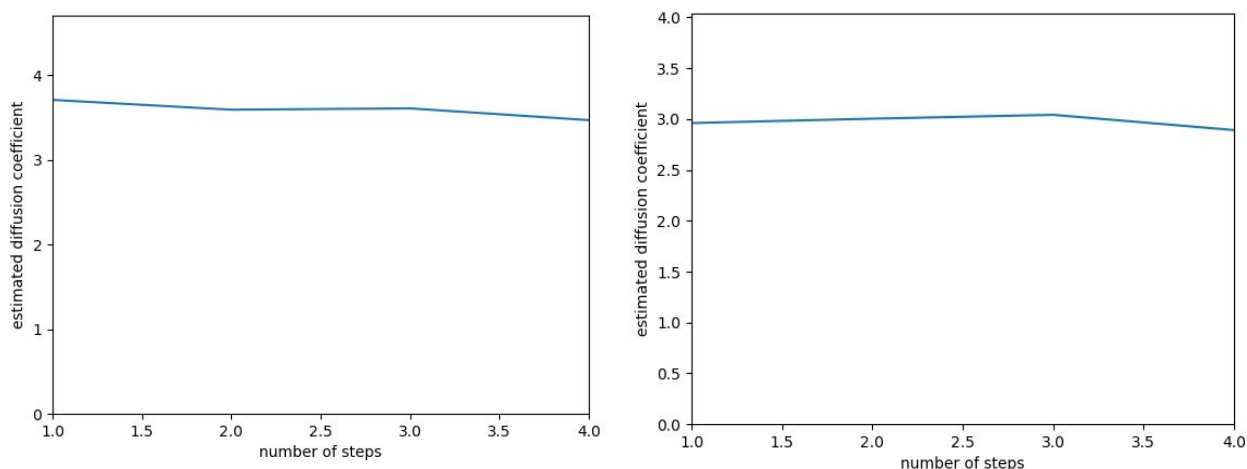


Figure 5: Diffusion coefficient from Brownian motion simulations estimated by minimizing the K-S statistic. Simulations in a $20 \times 20 \mu\text{m}^2$ box with reflective boundary conditions, with no additional obstacles (left) and with obstacles (right). The simulation diffusion coefficient was $D_0 = 4.0 \mu\text{m}^2/\text{s}$ and the simulation step length was $\Delta t = 0.05 \text{ s}$ in both simulations. The sampling time is a multiple of $\tau = l \cdot \Delta t$ with the step number l as indicated.

5. Results

In this section, we are going to look at the results yielded by our simulator and the analysis programs to date.

5.1 Standard Analysis of Simulation Results

MSD for different obstacle configurations

Multiple simulations were run with a diffusion coefficient of $D = 4.0 \mu\text{m}^2/\text{s}$, varying obstacles, and otherwise similar conditions (using 30 samples and 200 time steps of length $\Delta t = 0.05 \text{ s}$); the resulting MSDs were as shown in Figure 6.

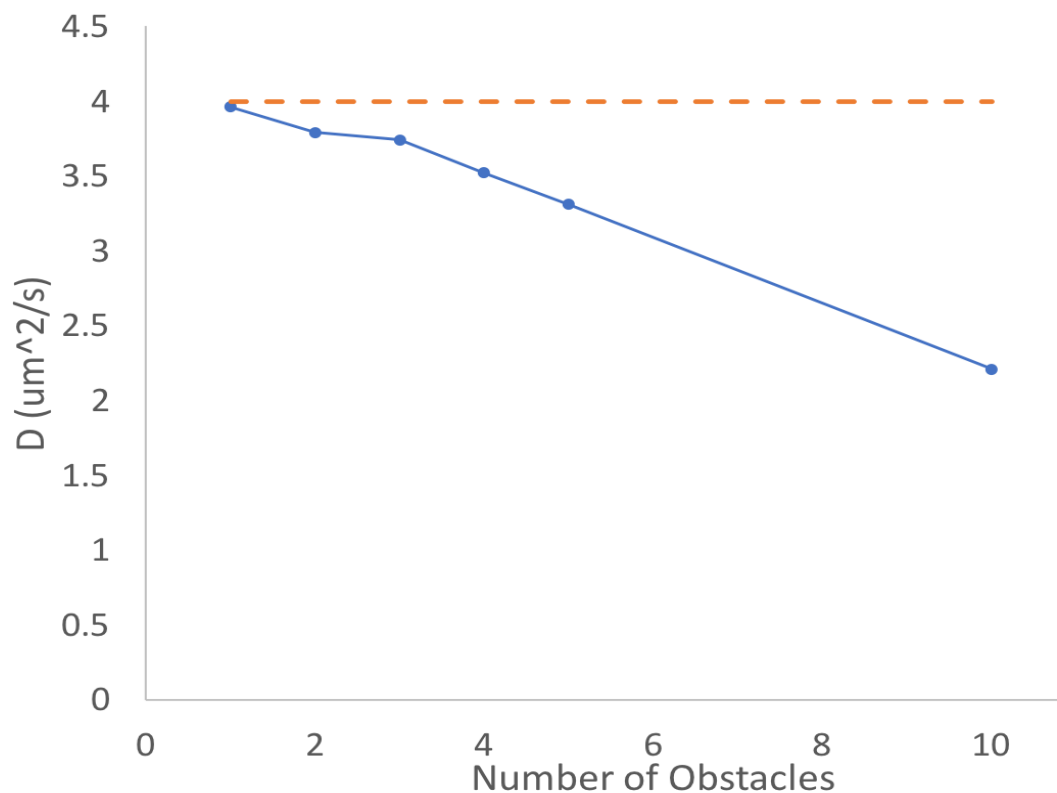


Figure 6: Diffusion coefficients estimated for different obstacle sets by KS minimization. See text for details.

In other words, without obstacles, the MSD is almost exactly what the theory says it should be under ideal Brownian motion. As the number of obstacles goes up, MSD goes down, since more and more movements get blocked (thus giving a displacement of zero). This is all to be expected; this test is largely intended to confirm that the program is working as intended.

Distribution of Square Displacements for Different Obstacle Configurations

The second analysis program, comparing histograms of actual and expected displacements, was tested on the trajectories of 30 samples, using 200 time steps of 50 milliseconds each, with a diffusion coefficient of 4. This was done successively with landscapes containing no obstacles, then 2, then 6, then 10 obstacles. The results can be seen in Figure 7:

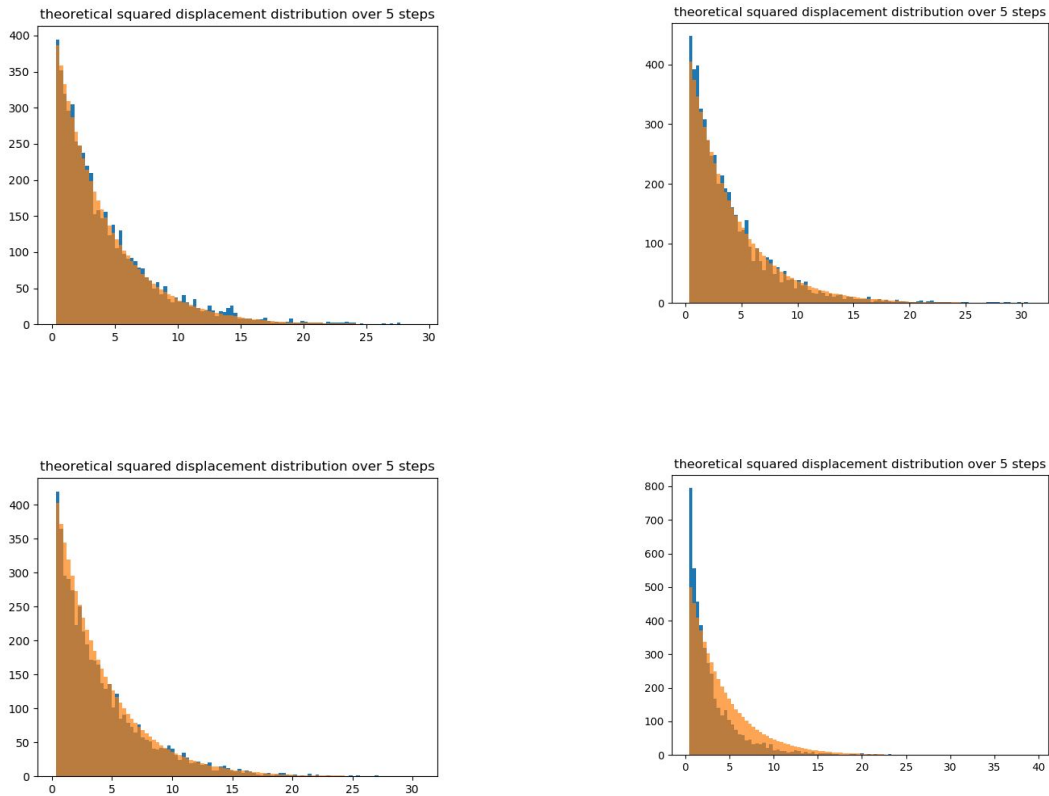


Figure 7: *Distribution of square displacements. Runs were performed under identical conditions with a varying number of obstacles (clockwise from top left: 0,2,10,6). Blue bars represent histograms of displacements collected from the simulations; yellow bars are the theoretical expectation for ideal Brownian motion. See text for more details.*

As may be seen from these histograms, the more obstacles are added, the more the actual square displacement distribution differs from the theoretical one: long displacements become increasingly rare, null displacements rise in number. This matches our natural expectations: the longer a displacement is, the greater the chance it will run into an obstacle and be set to zero.

5.2 Kolmogorov-Smirnov Fitting of Square Displacement Distributions

The Kolmogorov-Smirnov analysis program, similarly, was run with 30 samples, 200 time steps of 50 milliseconds each, a diffusion coefficient of 4, and successive landscapes with 0, 2, 6, and finally 10 obstacles. The KS statistic was calculated for square displacements over 1, 2, 3 and 4 steps, as seen in Figure 8.

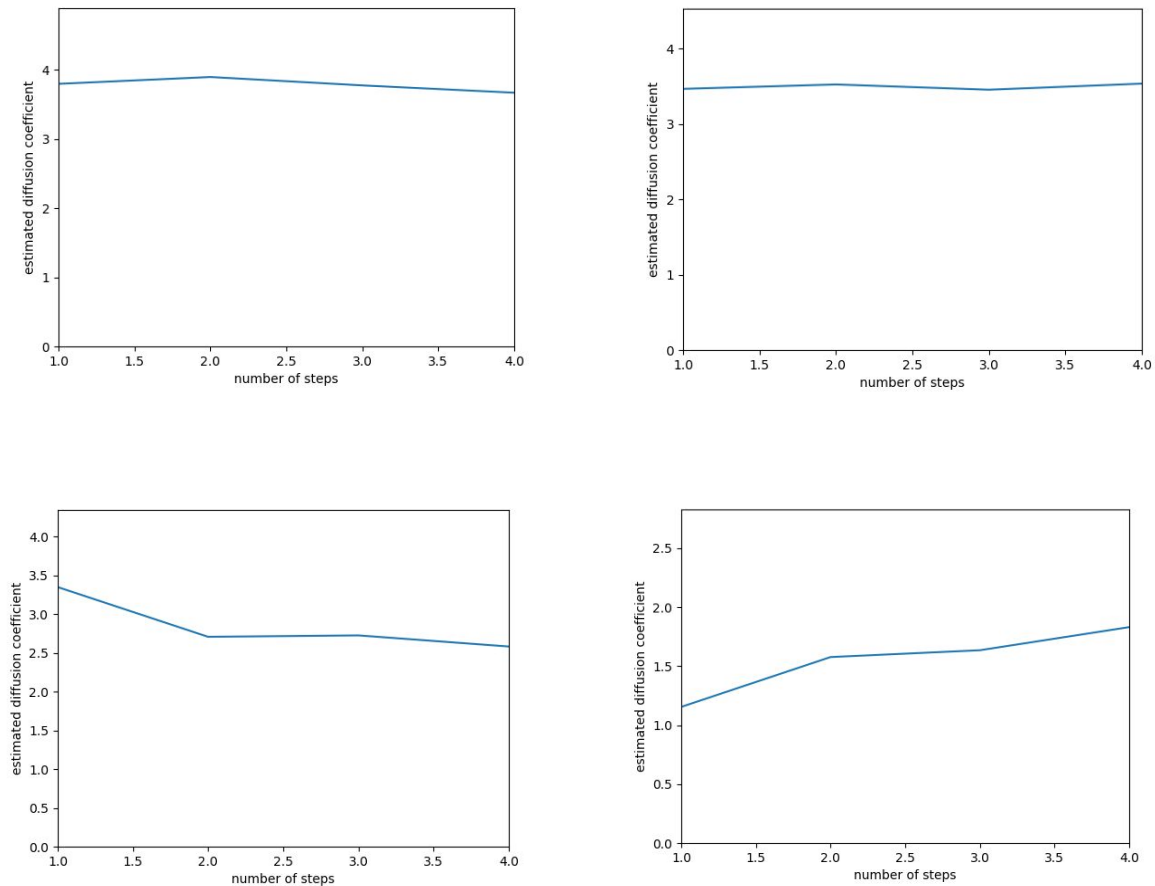


Figure 8: Effective diffusion coefficients for the runs shown in the previous figure. Runs were performed under identical conditions with a varying number of obstacles (clockwise from top left: 0, 2, 10, 6). See text for more details.

As hoped, when there are no obstacles, the program estimates a diffusion coefficient close to its actual value. As more obstacles are added and the MSD goes down, however, the program's estimated diffusion coefficient becomes smaller and smaller.

5.3 Exploration of Localization Density with Random Obstacles

At this stage, we generate landscape configurations that interest us and look at the induced density patterns. The localization density is an indication of possible accumulation or depletion of receptors from an area of the simulation space.

There are many indications, including SPT observations, that receptors tend to accumulate - to cluster - in relatively small areas; we know it is this clustering that allows ligand receptors to form oligomers and coss-activate. It is plausible that clustering is, at least in part, induced by a network of actin filaments running along the cellular membrane: Filaments will impede the movement of receptors, but it is not clear how this "corral" structure can lead to clustering.

An alternative explanation is "attractive domains" - areas that attract receptors through some mechanism. *One of the major objectives of this project is to help elucidate the origin of clusters.* To that end, we run simulations in various landscape models. Corrals are simulated by partitioning the surface with bilateral obstacles. Attractive domains, on the other hand, are simulated using one-way barriers, creating areas that let receptors in more easily than out.

Once those simulations have been performed, we need to compare them with experimental data. This involves looking for trajectories similar to what we generate, finding ways to measure "confinement", and, eventually, identifying movement patterns that reflect features of the underlying landscape.

Symmetric (bilateral) vs. biased (one sided) blocking

We first wish to compare biased and symmetric blocking. For that purpose, we use a landscape with only two large, parallel barriers. The barriers are asymmetric; they hinder

movement leaving the space between them, but not movements heading into it. The trajectories, Fig. 9 (left), are shown in yellow; they are concentrated on the space between the barriers.

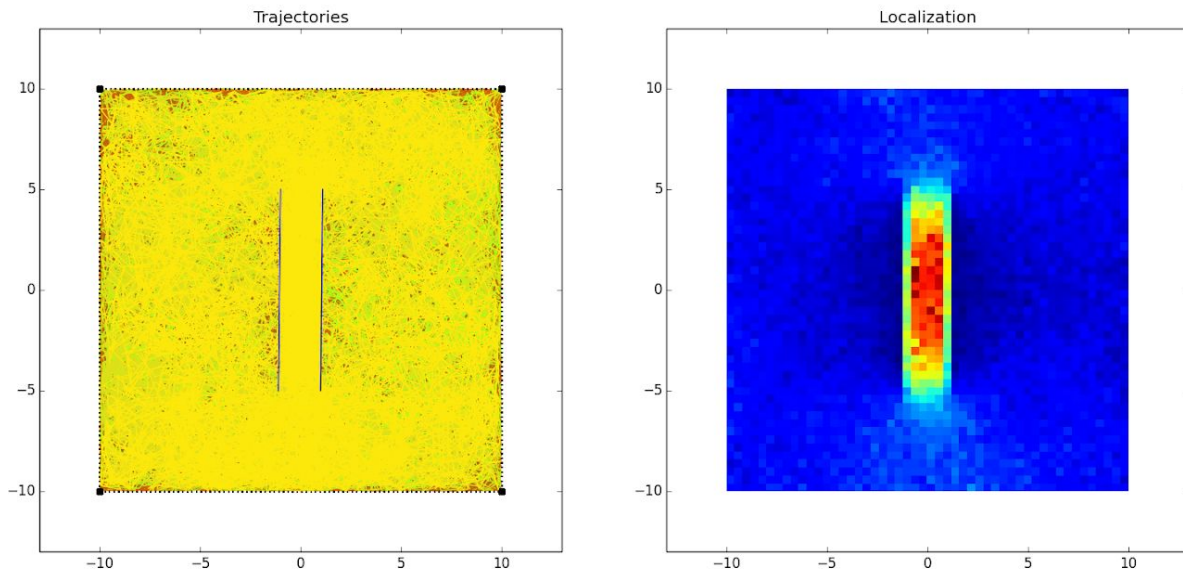


Figure 9: Trajectories and localization density in a landscape with two parallel, biased obstacles that hinder movement leaving the space between them.

Next, we try an identical simulation, but with bilateral barriers, as seen in Figure 10.

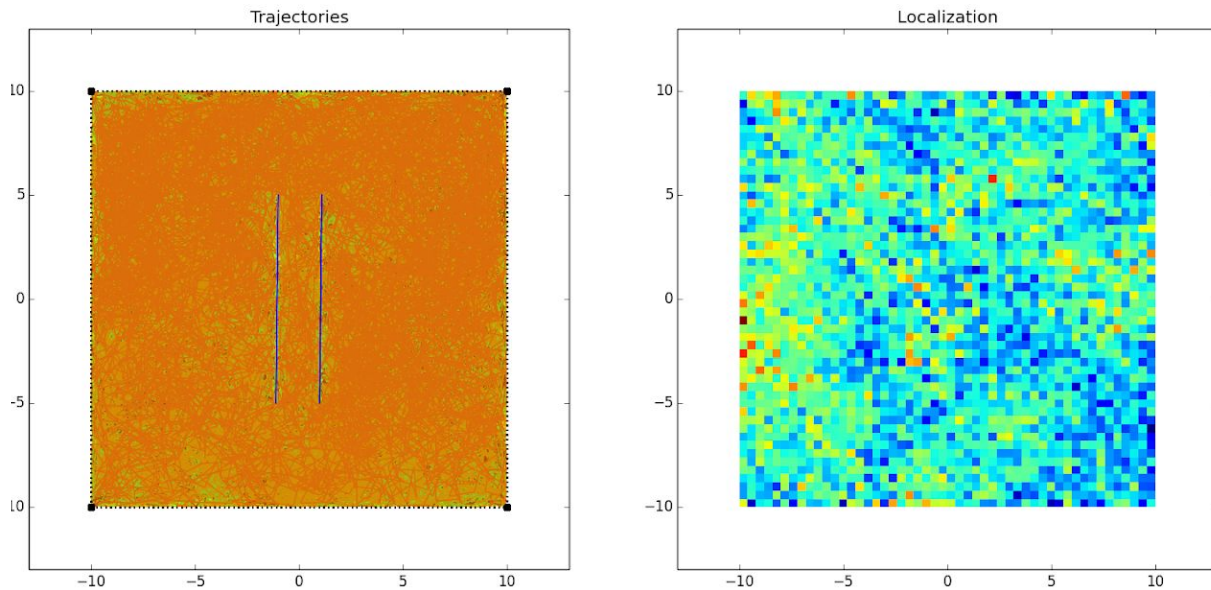


Figure 10: Trajectories and localization density with two parallel, bilateral obstacles.

This time, there is no discernible density effect.

Random barrier landscapes: results of simulations

For this part, we have run simulations to observe the clustering patterns. The blue lines indicate barriers (which are randomly generated). The density plot indicates some correlation between barriers and local density.

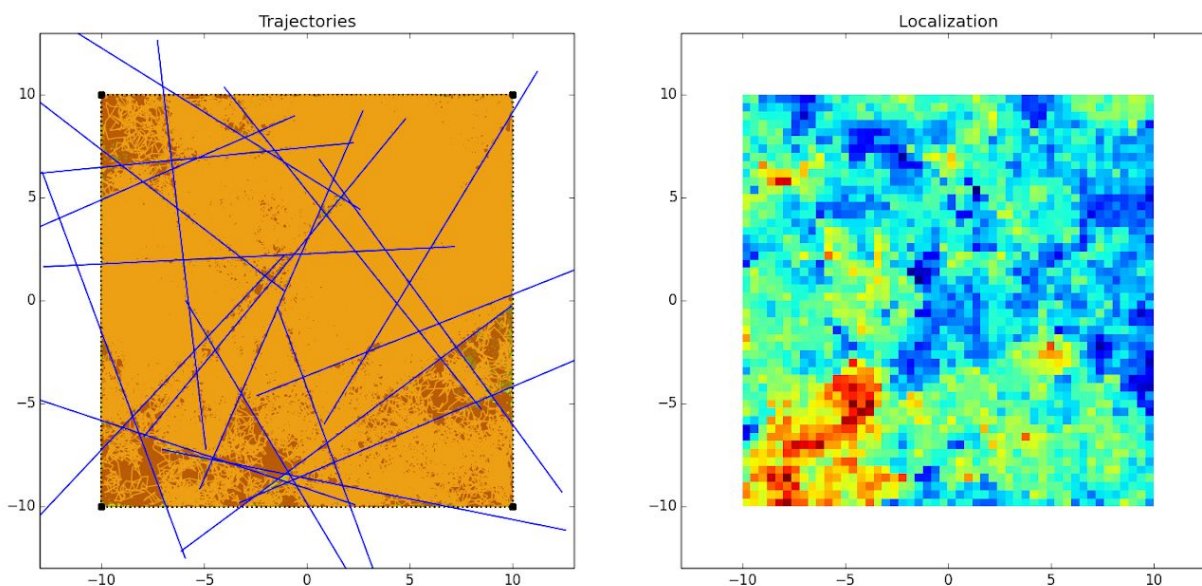


Figure 11: *Trajectories and localization density in a random landscape with bilateral obstacles. The left panel shows the trajectories; the right panel is a density plot obtained by performing a two dimensional histogram of all points in all trajectories. There are indications that the high and low density regions follow the corral boundaries.*

When looking at the effective diffusion coefficient (estimated by minimizing the K-S statistic), we find that the effective values we obtain are smaller than the one used for the simulations ($4.0 \mu\text{m}^2/\text{s}^2$). We performed this analysis for the simulations presented in Figure 11, using several different values for the time over which the displacements occur. In Figure 12 we plot the estimated diffusion coefficient against the expected MSD that corresponds to the sampling time, $\sigma = \sqrt{4D_0\tau}$. After an initial increase that is possibly an artefact of the sampling method, the plot in Figure 12 indicates that the effective diffusion coefficient decreases as the sampling time (and the equivalent expected jump size) increase. This is to be expected, and the decrease in mobility should be stronger as the expected jump size becomes comparable to the typical distance between barriers. These preliminary results are promising but inconclusive.

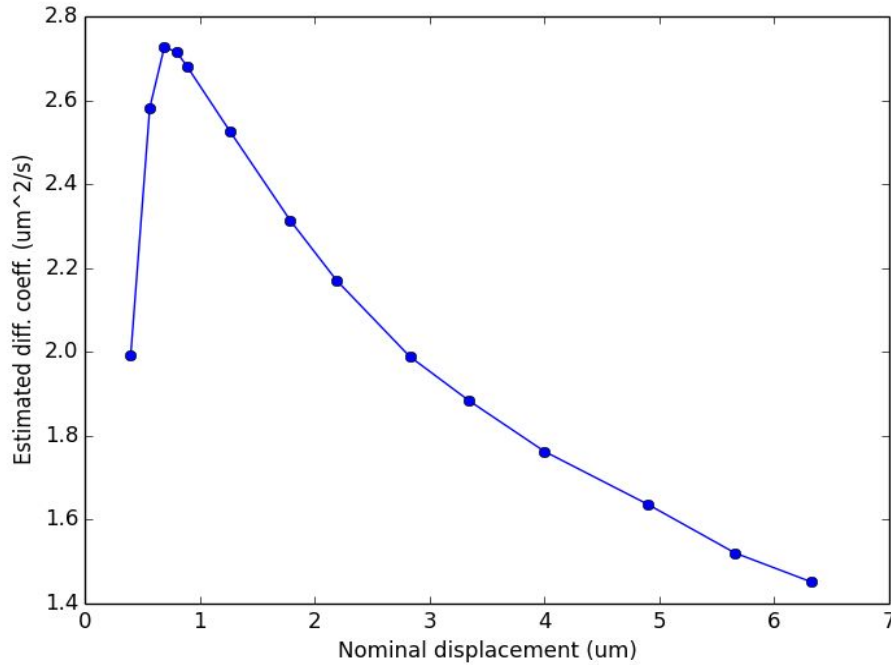


Figure 12: *Effective diffusion coefficients estimated using different sampling times. Results shown are for the runs shown in the previous figure, that used a random landscape with bilateral blocking. The estimated values are plotted against the expected mean square displacement $\sqrt{4D\tau}$ to compare to the size of the obstacles. See text for more details.*

Finally, we performed the same localization analysis with a version of the same landscape as in Figure 11, but with biased (one sided) blocking. The corresponding trajectories are shown in Figure 13. They are dramatically changed in that they tend to concentrate in small areas and avoid other altogether. This is also seen in the density plots, that indicate two small regions with very high density.

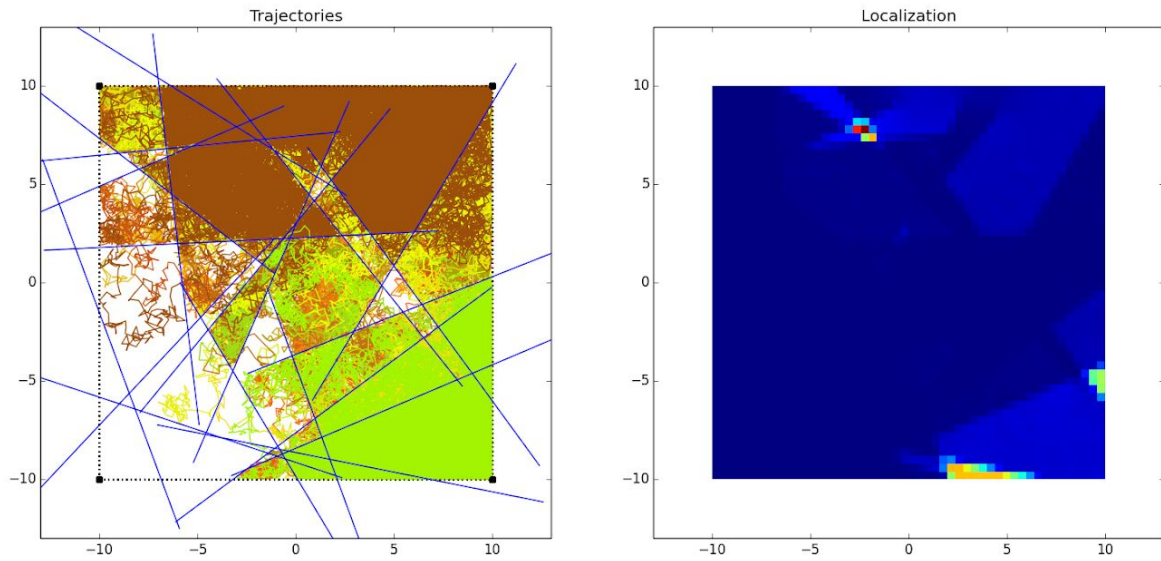


Figure 13: Trajectories and localization density in the same random landscape, but with obstacles that block in only one direction. The left panel shows the trajectories; the right panel is a density plot obtained by performing a two dimensional histogram of all points in all trajectories. The effect of the bias is such that the density is very high in two small areas, while others are devoid of particles.

6. Preliminary Conclusions and Outlook

Having run the above tests, we will in this section discuss the preliminary conclusions that may be taken from our data, and consider what work needs to be done in the future.

We have confirmed that, while the mean square displacement approximates its expected value in the absence of obstacles, it decreases as obstacles are added. We have obtained a tentative method of quantifying the behavioral changes that obstacles induce in the model. Preliminary comparisons of model behavior have indicated that clustering patterns are facilitated by biased barriers, they are unaffected by bilateral obstacles. This lends further credence to the hypothesis that cells regulate ligand receptor clustering via lipid raft trapping zones rather than actin corrals.

From here, future research needs to compare the simulated results to experimental observations. SPT gives us ample data on the trajectories of receptors; by fine-tuning the Brownian motion simulator to yield trajectories with similar properties, we may better understand the model under which receptor motion behaves.

Bibliography

- [1] Andrews, N.L., Lidke, K.A., Pfeiffer, J.R., Burns, A.R., Wilson, B.S., Oliver, J.M., Lidke, D.S.:1209 Actin restricts fc ϵ ri diffusion and facilitates antigen-induced receptors immobilization. *Nature Cell Biol.*10(8), 955–963 (2008). doi:10.1038/ncb1755
- [2] Chen, K.C., Csikasz-Nagy, A., Gyorffy, B., Val, J., Novák, B., Tyson, J.J.: Kinetic analysis of a molecular model of the budding yeast cell cycle. *Molecular Biology of the Cell* 11(1), 369–391 1217 (2000)
- [3] Craciun, G., Tang, Y., Feinberg, M.: Understanding bistability in complex enzyme-driven reaction networks. *Proc. Nat. Acad. Sci. USA* 103, 8697–8702 (2006)
- [4] Edidin, M.: Lipid microdomains in cell surface membranes. *Current opinion in structural biology* 7(4), 528–532 (2001)
- [5] Holmes K, Roberts OL, Thomas AM, Cross MJ (Oct 2007). Vascular endothelial growth factor receptor-2: structure, function, intracellular signalling and therapeutic inhibition. *Cell Signal.* **19** (10): 2003–2012.doi:10.1016/j.cellsig.2007.05.013. PMID 17658244.
- [6] Kholodenko, B.N., Demin, O.V., Moehren, G., Hoek, J.B.: Quantification of short term signaling by the epidermal growth factor receptor. *The Journal of Biological Chemistry* 274(42), 123730169–30181 (1999)
- [7] Kolmogorov A (1933). Sulla determinazione empirica di una legge di distribuzione. *G. Ist. Ital. Attuari.* **4**: 83–91.
- [8] Kusumi, A., Sako, Y.: Cell surface organization by the membrane skeleton. *Current opinion in cell biology* 8(4), 566–574 (1996)
- [9] MacGabhann, F., Popel, A.S.: Systems Biology of Vascular Endothelial Growth Factors. *Microcirculation* 15(8), 715–738 (2008)

[10] Pryor, M.M., Low-Nam, S.T., Halász, A.M., Lidke, D.S., Wilson, B.S., Edwards, J.S.: Dynamic transition states of erbb1 phosphorylation predicted by spatial-stochastic modeling. *Biophys. J.* 105(6), 1533–1543 (2013)

[11] Smirnov N (1948). Table for estimating the goodness of fit of empirical distributions. *Annals of Mathematical Statistics.* 19: 279–281. doi:10.1214/aoms/1177730256.

[12] Diabetes Fact sheet N°312. WHO. October 2013. Archived from the original on 26 August 2013. Retrieved 25 March 2014.