Katedra buněčné biologie

Matematické modelování v bioinformatice

Název úlohy: Elaborát se simulací

Jméno: Anna Agafonova

Datum : 13. 12. 2022

Připomínky opravujícího:

	Možný počet bodů	Udělený počet bodů
Teoretická část		
Výsledky a zpracování měření		
Diskuse výsledků		
Závěr		
Použitá literatura		
Celkem		

Posuzoval: dne:

Contents

1 Problem definition **2**
1.1 Chemotaxis . . . 2

2 Practical considerations **2**
2.1 Continuous vs discrete space . . . 2
2.2 Maze generation . . . 2
2.3 Programming language . . . 3

3 Model description **3**
3.1 Diffusion model . . . 3
3.1.1 Theoretical background . . . 3
3.1.2 Practical aspects . . . 3
3.1.3 Mass diffusivity coefficient . . . 4
3.2 Cells simulation . . . 4
3.2.1 Self-generated chemotaxis . . . 4

4 Experiments **5**
4.1 Comparison between classical and self generated model of diffusion . . . 5
4.1.1 Classical vs self-generated model on the simplest map . . . 5
4.1.2 Classical vs self-generated model on the complex map . . . 5
4.1.3 The solution to the complex maze . . . 6
4.2 Parameters analysis . . . 6
4.2.1 Michaelis-Menten kinetics parameters V_m and k_{max} . . . 6
4.2.2 Experiment settings: cell number, concentration, and diffusivity parameters. . . . 7

5 Conclusion **7**

1 Problem definition

1.1 Chemotaxis

This work is based on the article [1]. The mentioned paper studied computational models of long-range movements of cells. The authors proposed a model of self-generated chemotaxis; they studied it in a series of computational experiments and validated the results of simulations with real-life experiments of real cell movements in constructed mazes.

In the classical model of chemotaxis, a cell is able to sense differences in the concentration gradient of a attractant molecule and move in direction of the increasing gradient. In contrast to such a simple model, in the model of self-generated gradient, a cell is additionally breaking down a molecule of attractant, thus, influencing her local gradient.

The aim of our experiments is to show whether this model of chemotaxis better describes the observed cell movements in real-life in a range of different settings. And to test several hypotheses regarding the model of chemotaxis, in particular, if cells that move according to a simple model are unable to resolve complex mazes architectures and move over long distances, whereas if they move according to the complex self-generated model they are successfully solving these issues.

2 Practical considerations

Before I am going to discuss model construction and experiments I would like to briefly mention the several practical aspects of this work.

2.1 Continuous vs discrete space

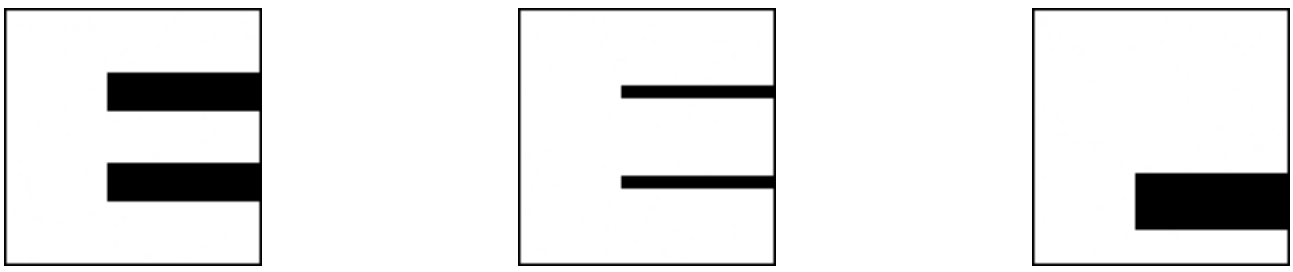
The model of chemotaxis assumes that entities, which are cells and molecules, are moving in a space according to internal laws. So, first of all, there was a consideration on how to represent a space and the movements of entities in this space. There were two possible alternatives:

1. Continuous space with cells moving according to functions output that predicts their location in space in the next time step
2. Discrete space where cells would move on the grid from one unit to another

The continuous space has the advantage of easier computations, for example, of diffusion gradient, while discrete space offers easier work with cell movements. In my model, I decided to work with discrete space and model everything as movements on the grid in time. Such a decision made the modeling of attractant gradient distribution slightly more complex, as in such a case we would like to describe the behavior of continuous differential equations discretely. Nevertheless, it simplifies the work with cells that are now simply entities moving on the grid.

2.2 Maze generation

For some experiments it is essential to have some kind of maze to explore cell chemotaxis in a complex environment. For the sake of simplicity I decided to create a maze in the a simple graphical editor and use it as True/False mask for the grid to prohibit the cell and molecular movements in some areas. Examples of the mazes are present in the Figure 1.



(a) Maze example 1 (b) Maze example 2 (c) Maze example 3

Figure 1: Examples of grid maps with constraints used in the experiments

Note that such an approach is associated with several drawbacks. One of them is the behavior of a diffusion submodel on the barrier regions (defined as False on our binary image). In this model, we are solving this issue by first allowing the "molecule" to leak through the barrier but later we recompute the whole material that is present according to the new distribution. What is meant by this is that we model a closed system from which the matter can not escape. If there wouldn't be such a step of energy redistribution after several steps the whole matter would leak away from the maze thus creating a natural zero state in our model. Alternatively, this could possibly be modeled with a more complex derivation of the diffusion model but for the sake of simplicity, I model it this way. Though I accept that such a simplification is diverging our model from the real-life diffusion process in which the matter would rather bounce off the wall.

2.3 Programming language

The work is selected to by done in Python since Matlab Simbiology does not allow to manipulate with space and movements. If necessary the work can be also done in Julia.

3 Model description

The model described in the paper would contain basically two main entities:

1. The population of cells moving through the maze. There should be certain parameters of this population like population size, initial position, an average speed of a cell, number of receptors for gradient measurement, the threshold for the sensitivity of gradient differences, speed of enzymatic breakage of attractant molecules. Somehow, each receptor should hold its identity in the sense of polarity for gradient differences calculations. Enzymatic breakage will be guided by Michaelis-Menten kinetics.
2. Attractant source with an initial population of attractant molecules. This source would have defined size in terms of the number of molecules and initial position. Additionally, each attractant molecule should have a specified rate of diffusion.

3.1 Diffusion model

3.1.1 Theoretical background

The diffusion model is simulated as close as possible to its physical description, e.g. mostly ruled by Fick's law of diffusion [2]. This law can be summarised for 2-dimensional space as following

$$\mathbf{J} = -D\nabla\varphi \tag{1}$$

In this law, D is a diffusion coefficient, and it will be one of the possible parameters for manipulation during the experiments. In its physical meaning, this term accounts for all the conditions of the environment such as temperature, pressure, and chemical characteristics of the substance to diffuse. While the second term φ is the concentration, of which the dimension is the amount of substance per unit area, which gradient we are going to follow to predict how the substance is going to spread in space.

3.1.2 Practical aspects

Given the gradient of the substance in space to be able to evolve it in time we have to predict how the situation would look in the next moment.

In such a discrete case we want to approximate the concentration in each possible unit on the grid after doing the time step. This can be done according to the following equation. Given that

$$\mathbf{J} = -D\nabla\varphi = \frac{\partial\Phi}{\partial t} = D(\frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2}) \tag{2}$$

$$\frac{\varphi_{i,j}^{t+1} - \varphi_{i,j}^{t}}{\Delta t} = D\left[\frac{\varphi_{i+1,j}^{t} - 2\varphi_{i,j}^{t} + \varphi_{i-1,j}^{t}}{(\Delta x)^2} + \frac{\varphi_{i,j+1}^{t} - 2\varphi_{i,j}^{t} + \varphi_{i,j-1}^{t}}{(\Delta y)^2}\right] \tag{3}$$

$$\varphi_{i,j}^{t+1} = \varphi_{i,j}^{t} + D\Delta t\left[\frac{\varphi_{i+1,j}^{t} - 2\varphi_{i,j}^{t} + \varphi_{i-1,j}^{t}}{(\Delta x)^2} + \frac{\varphi_{i,j+1}^{t} - 2\varphi_{i,j}^{t} + \varphi_{i,j-1}^{t}}{(\Delta y)^2}\right] \tag{4}$$

It is essential to also specify the stable time step Δt

$$\Delta t = \frac{1}{2D}\frac{(\Delta x\Delta y)^2}{(\Delta x)^{2+}(\Delta y)^2} \tag{5}$$

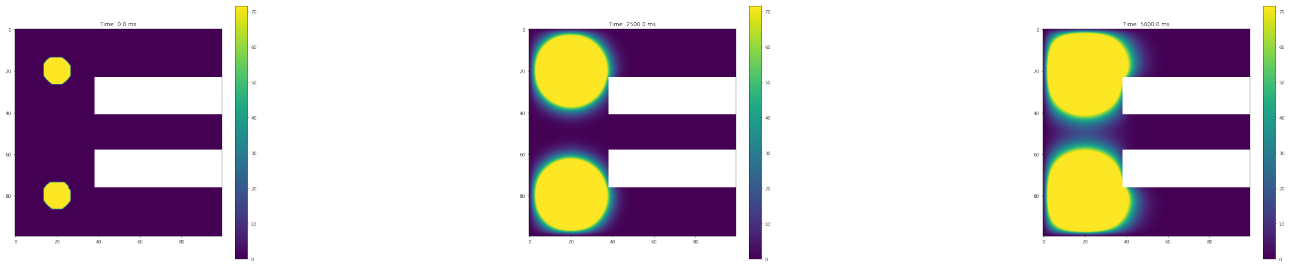
The discrete example of gradient computation was adapted from "Learning Scientific Programming with Python" online book.

Figure 2 shows an example of diffusion in the maze example 1 starting from 2 points at the concentration 5000 units/mm.

3.1.3 Mass diffusivity coefficient

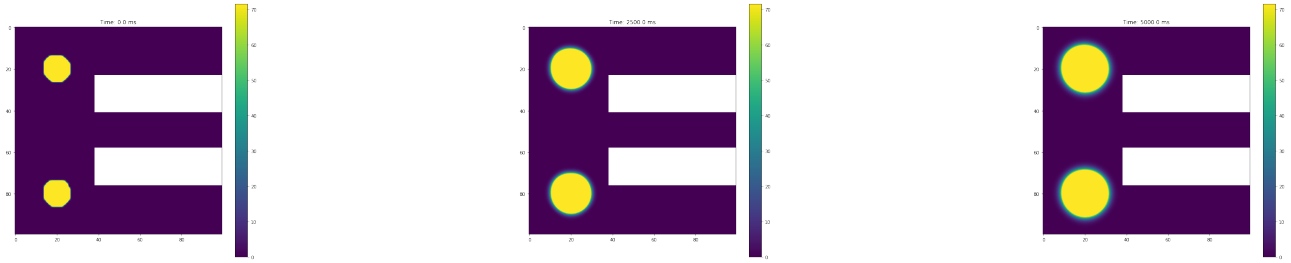
The major coefficient that defines the diffusion rate in the model is the mass diffusivity coefficient D. Theoretically, this coefficient characterizes the physical parameters (such as temperature and viscosity) of the given substance that affect the rate with which it is spread in space. The higher this parameter, the faster the substance is supposed to diffuse.

To ensure that it actually holds we can set an experiment (Figure 3) with different values of D in the same time frames that shows that D affects the simulation in an expected way.



(a) time = 0 ms (b) time = 2500 ms (c) time = 5000 ms

Figure 2: The simulation of simple diffusion, $D = 0.1$



(a) time = 0 ms (b) time = 2500 ms (c) time = 5000 ms

Figure 3: The simulation of simple diffusion, $D = 0.01$

For practical reason the time step in the simulation takes the D value into account as is defined in the equation (5).

3.2 Cells simulation

The simulation of cell movements and chemotaxis is inspired by lecture 6 from the course on biological modeling of the University of Tuft. Cells are represented as instances of a class "Cell" to initialize multiple instances at the same time. The entity of a cell has only several internal functions: **sense** and **step**, while the second is always dependent on the first. The function **sense** "observe" the concentrations in the nearby grid units, while function **step** does a stochastic move by computing the random choice from the selected neighbors according to the probabilities of such a move.

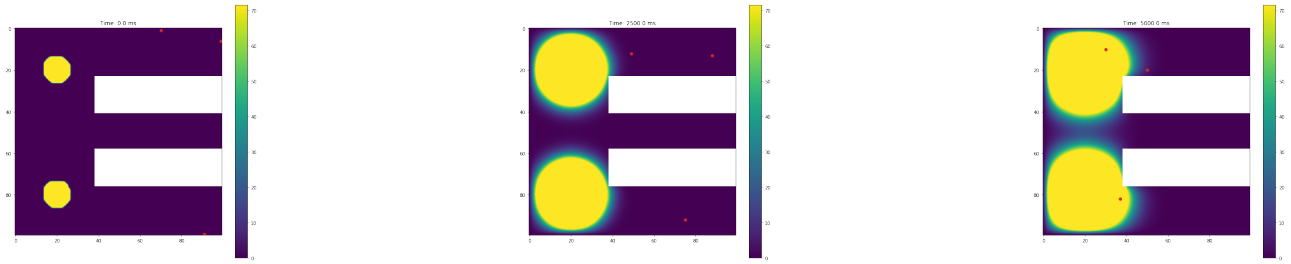
An example of classical chemotaxis performance is shown below.

Additionally cells can perform in two possible modes:

- According to classical model of chemotaxis
- According to self-generated chemotaxis

3.2.1 Self-generated chemotaxis

The self-generated model of chemotaxis assumes that the cell entity is not only able to observe the concentration gradient and to move in the direction of a higher gradient but that it actively interacts with the gradient by



(a) time = 0 ms (b) time = 2500 ms (c) time = 5000 ms

Figure 4: The simulation of classical chemotaxis

consuming it. The consumption is defined by the Michaelis-Menten equation (6) as an underlying mechanism that probably involves the enzyme that breaks the given substance down.

$$v = V_{\max} \frac{c}{k_{\mathrm{m}} + c} \tag{6}$$

The graph that shows the behavior of the Michaelis-Menten equation generated from the simulation using concentration values from the simulated concentration map φ is shown in Figure 6.

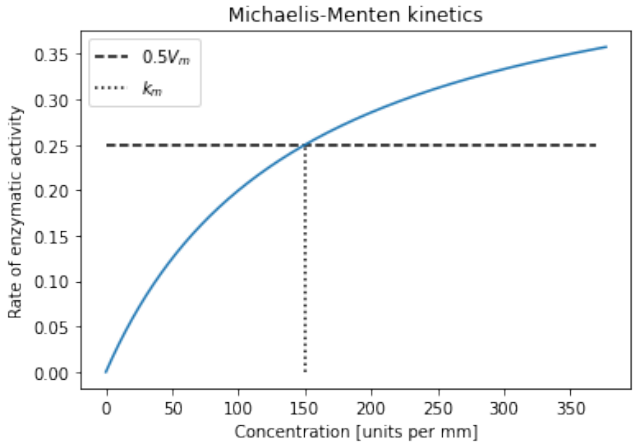


Figure 5: Saturation curve for an enzyme reaction showing the relation between the substrate concentration and reaction rate, generated from simulation

4 Experiments

4.1 Comparison between classical and self generated model of diffusion

4.1.1 Classical vs self-generated model on the simplest map

First, the most simple maze was tested. Such a maze had a single path with a single bifurcation and two subpaths. The experiment was conducted with one or two substance sources, in both cases, cells were successful to find the source and there was no observable difference between the self-generated and classical models. The experiment is summarized in Figures 6 and 7.

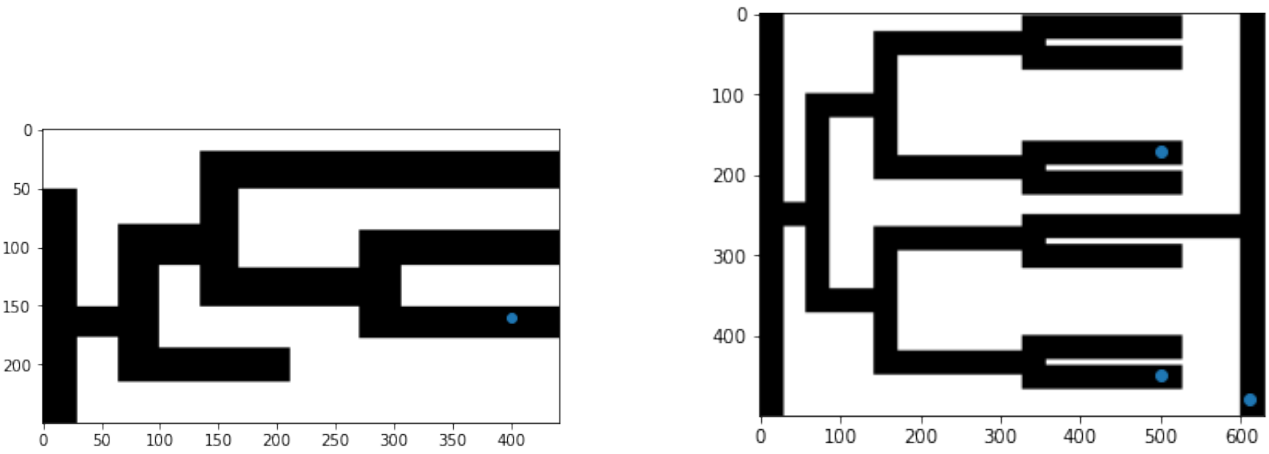
If there are two sources the initial population of cells will be separated into two populations and both will continue in a similar manner. Again, there is no difference if the model is self-generated or classical.

In the case of a classical model, after reaching the source, cells got trapped in it. In the case of a self-generated model, the cells will start slowly to consume all the available resources and then continue with exploratory behavior.

4.1.2 Classical vs self-generated model on the complex map

As we have seen the complex model has no advantage over a simpler one in the case of a simple maze. We can explore the more complex maps to find out if there are any advantages there for the complex model. Such

a complex maze is shown in Figure 9. It can be seen that it contains multiple dead-ends and can potentially contain several sources.



(a) Complex maze 1

(b) Complex maze 2

Figure 8: Example of a complex maze

In the case of one source, the experiment is clear, both simple and complex model cells will find the source and get to it. In the case, however, of multiple sources, the simple model is capable of finding the closest and it gets stuck there forever as can be seen in Figure 9. The complex model shows a much more interesting behavior, in the complex model cells rapidly find the closest source, then consume it and move to the next one as shown in Figure 10. This, however, is only true if the second source is not too far away, otherwise, cells are likely to get stuck in the first source location until all of the molecules are not consumed and are misguided by any small fluctuations.

In comparison with the original publication, [1] my self-generated model did not show any type of population separations at the nodes of the maze. The full population always travels as a whole, guided by the closest source. Probably it can be explained by the fact that for one side the source was too close so it had already a much higher gradient in the beginning.

4.1.3 The solution to the complex maze

What if the cells are given a task to find the largest source in the large maze? In this case the distribution of sources are not equal. In our particular example as shown in Figure 11 and 12 there are two sources of the size equal to $\frac{1}{4}$ of a given concentration and one source that is one $\frac{1}{2}$ of a concentration. Will cells find the biggest sources? Unfortunately, on the short time distance both models performed poorly. Both models find the closest source, even if it is smaller and get stuck in it, only in the case of self-generated model this situation is probably resolved after cells consume all available to them source.

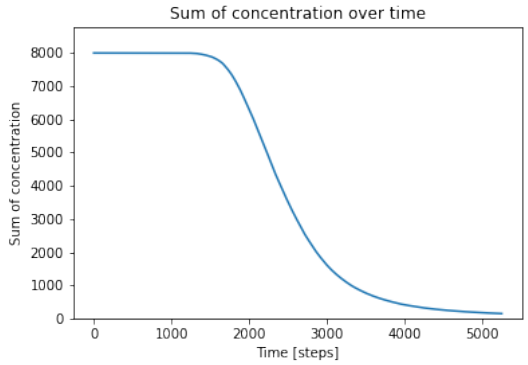
4.2 Parameters analysis

4.2.1 Michaelis-Menten kinetics parameters V_m and k_{max}

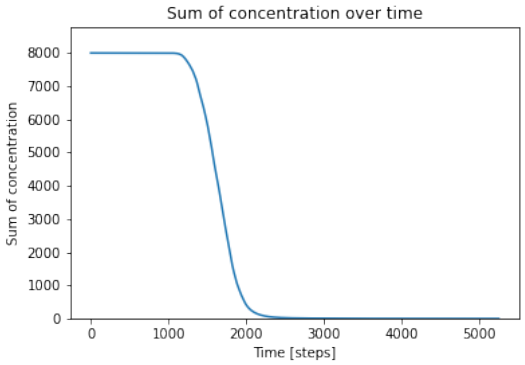
We would like to understand the Michaelis-Menten kinetics since it is a crucial parameter for the functioning of our self-generated model.

Theoretically, V_{max} is the parameter that describes the maximum possible rate of reaction. Biologically, this would mean the availability of an enzyme and how fast this enzyme is capable to work. The k_{max} is a concentration of a substrate at which rate of reaction is equal to $\frac{1}{2}V_{max}$.

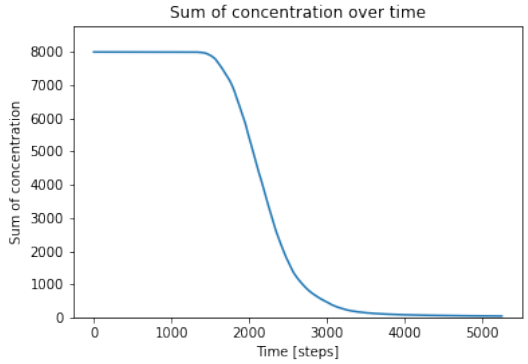
Here, we can show that with a higher V_{max} value, the rate with which cells can consume something grows. The parameter k_{max} seems to not increase the rate but rather shift the critical point in time at which the concentration starts to decrease. The results of these experiments are shown in Figure 8.



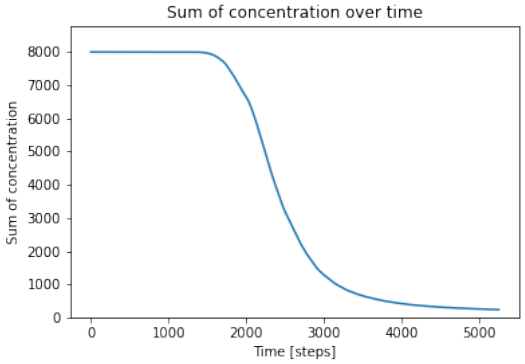
(a) Concentration decrease with parameter $V_{max} == 0.1$



(b) Concentration decrease with parameter $V_{max} == 1$



(c) Concentration decrease with parameter $k_{max} == 0.1$



(d) Concentration decrease with parameter $k_{max} == 1$

Figure 13: Graphs of concentration plotted vs time for different parameter of Michaelis-Menten kinetics

4.2.2 Experiment settings: cell number, concentration, and diffusivity parameters.

Other parameters such as diffusivity coefficient, initial concentration, and the number of cells that are present in an experiment are unlikely to change the type of behavior that is observed but rather change the timescale at which the behavior is displayed. For example, in the scenario of multiple sources and self-generated model cells are going to find both sources in either case but the diffusivity coefficient is going to influence how fast they are able to do that. A quite similar idea holds for initial concentration levels. Because the time step is computed with regard to the value of diffusivity practically this means that we always see similar behavior but it is played in a different timescale.

The number of cells acts in a pretty much similar manner. The more cells there are the faster they are going to consume all available molecules in one source and move to the next one. In the end, it only affects the timescale, but not the general behavior.

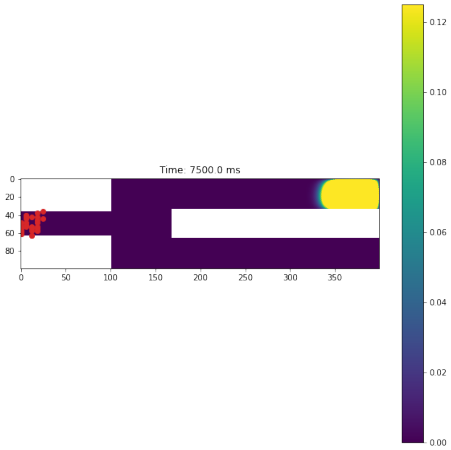
5 Conclusion

In the original publication, [1] authors state that a simple imposed model cannot resolve branches, nor travel on longer distances, thus they proposed the self-generated model that can overcome these difficulties. However, in my simulation, both models were able to travel any distances if the time given was long enough and concentration was sufficient. Both models performed well in all environments that were tested. Nevertheless, the more complex self-generated gradient model was capable of exhibiting more interesting exploratory behavior as could be expected. It was able to not only find the closest source but consume it and move to the next one. This was observed in most of the environments. Cells always moved towards the closest source disregarding the availability of the resource there. This observation could be further tested in different environments to prove that cells always select the optimal path.

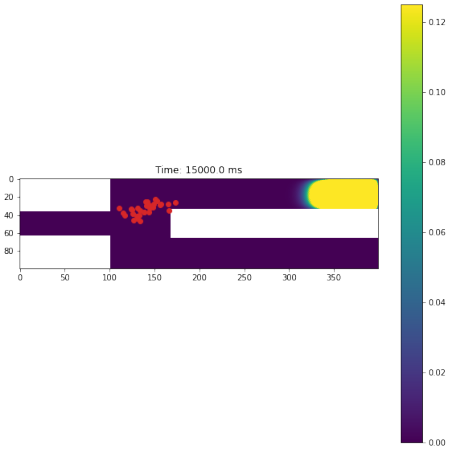
References

[1] L. Tweedy, P. A. Thomason, P. I. Paschke, K. Martin, L. M. Machesky, M. Zagnoni, and R. H. Insall, "Seeing around corners: Cells solve mazes and respond at a distance using attractant breakdown," *Science*, vol. 369, no. 6507, p. eaay9792, 2020.

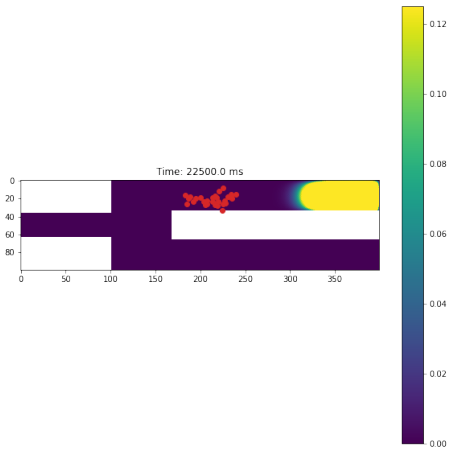
[2] J. Philibert, "One and a half century of diffusion: Fick, einstein, before and beyond," *Diffusion Fundamentals*, vol. 2, 11 2004.



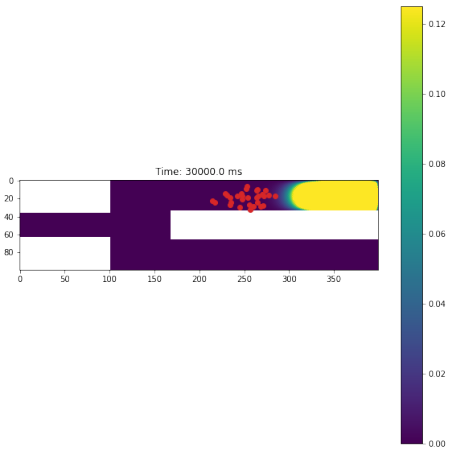
(a) Simulation run at time 7500 ms



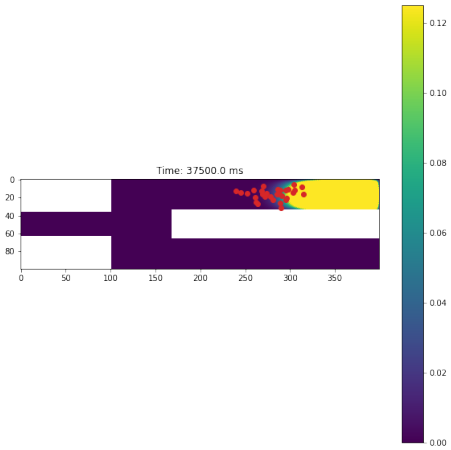
(b) Simulation run at time 15000 ms



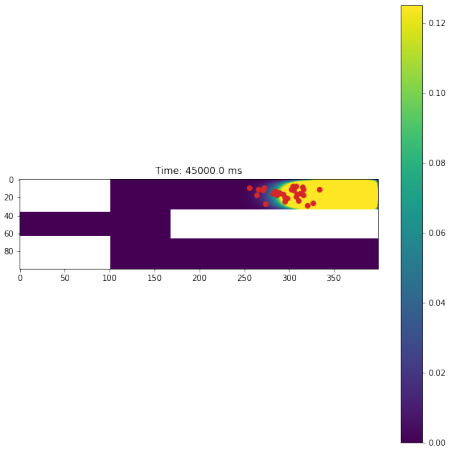
(c) Simulation run at time 22500 ms



(d) Simulation run at time 30000 ms

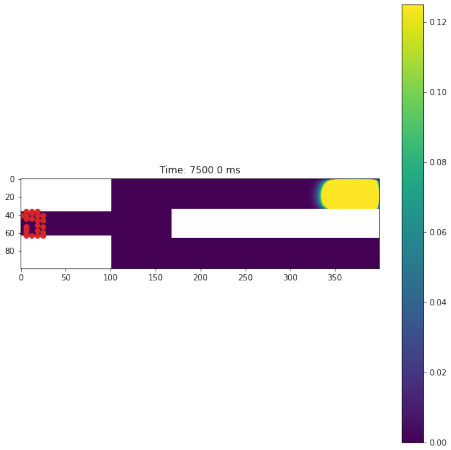


(e) Simulation run at time 37500 ms

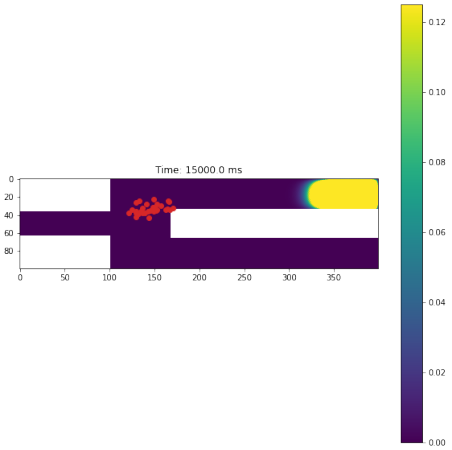


(f) Simulation run at time 45000 ms

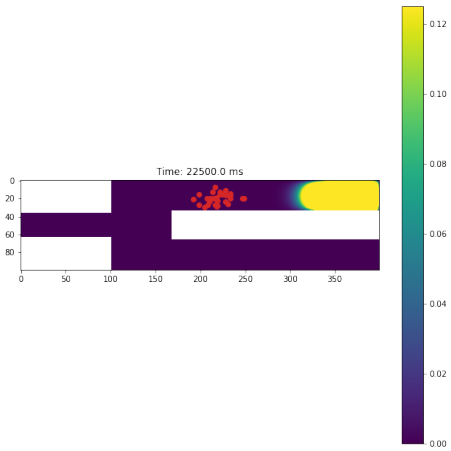
Figure 6: Simulation in a simple maze of a classical model of chemotaxis



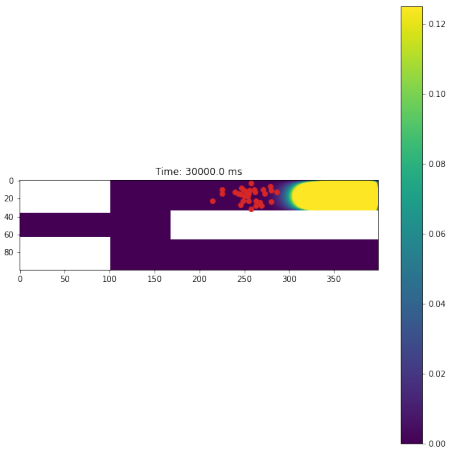
(a) Simulation run at time 7500 ms



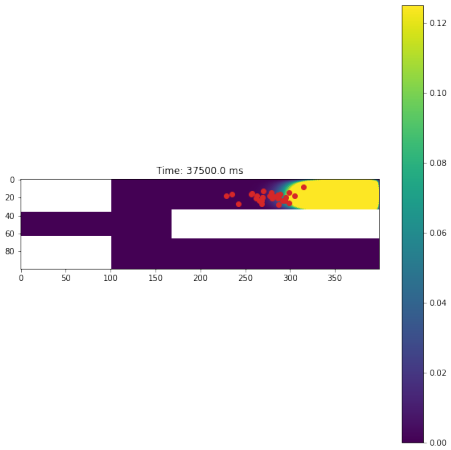
(b) Simulation run at time 15000 ms



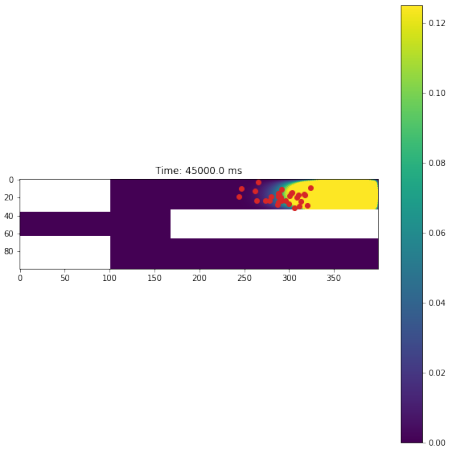
(c) Simulation run at time 22500 ms



(d) Simulation run at time 30000 ms

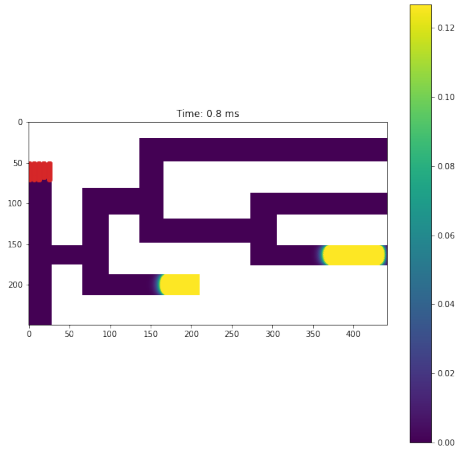


(e) Simulation run at time 37500 ms

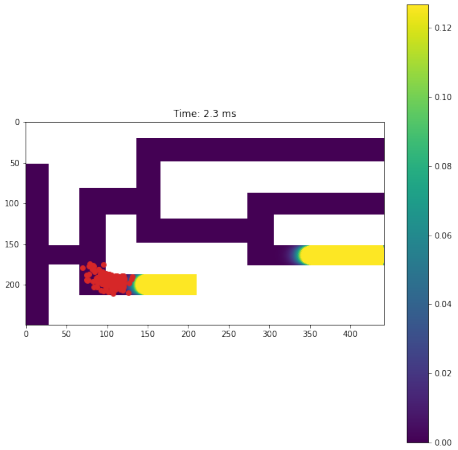


(f) Simulation run at time 45000 ms

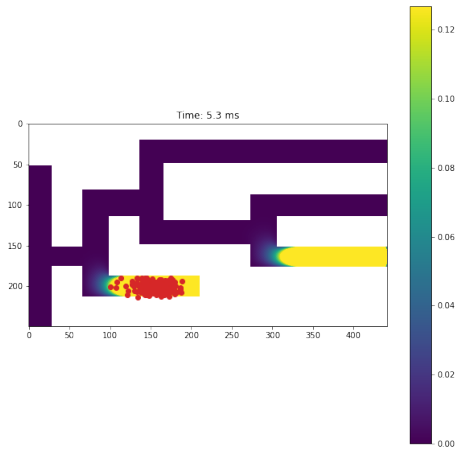
Figure 7: Simulation in a simple maze of a self-generated model of chemotaxis



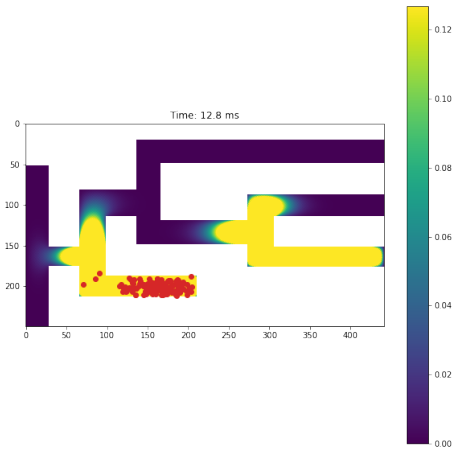
(a) Simulation run at time 0.8 ms



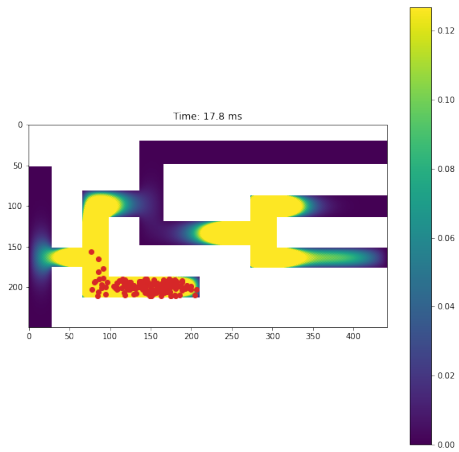
(b) Simulation run at time 2.3 ms



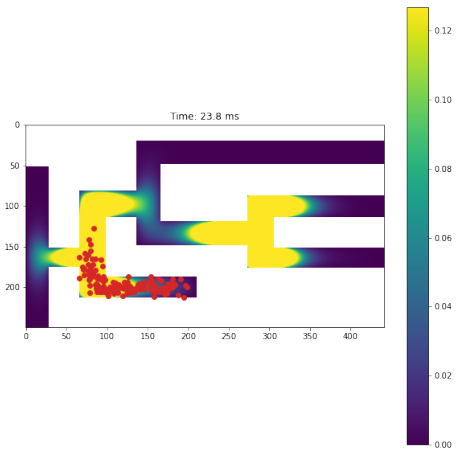
(c) Simulation run at time 5.3 ms



(d) Simulation run at time 12.8 ms

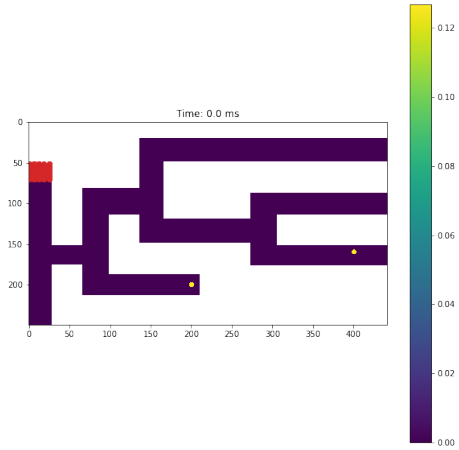


(e) Simulation run at time 17.8 ms

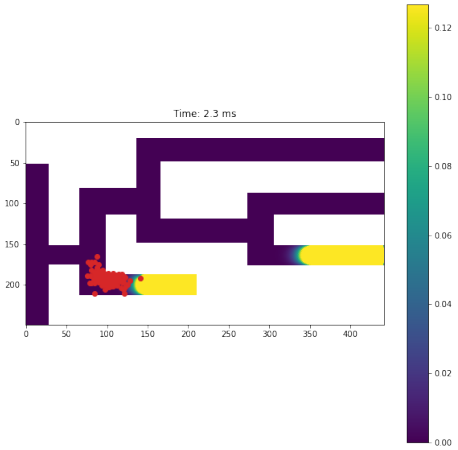


(f) Simulation run at time 23.8 ms

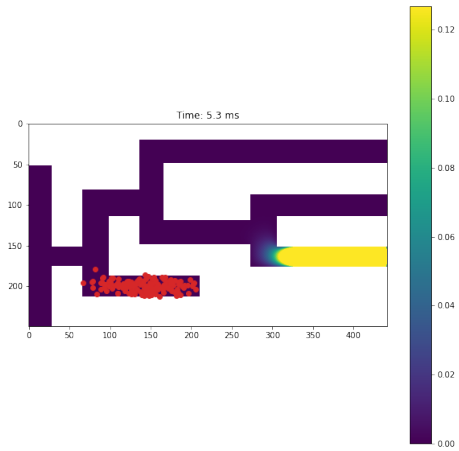
Figure 9: Simulation in a complex maze with two sources of a classical model of chemotaxis



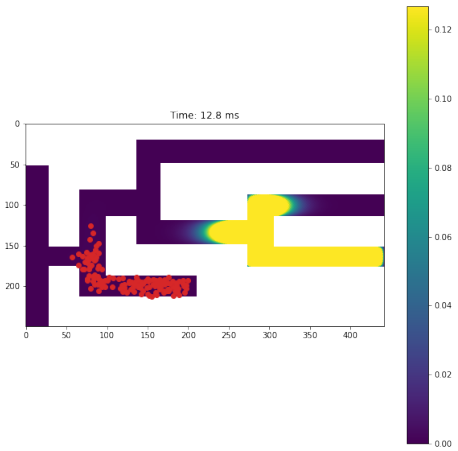
(a) Simulation run at time 0.0 ms



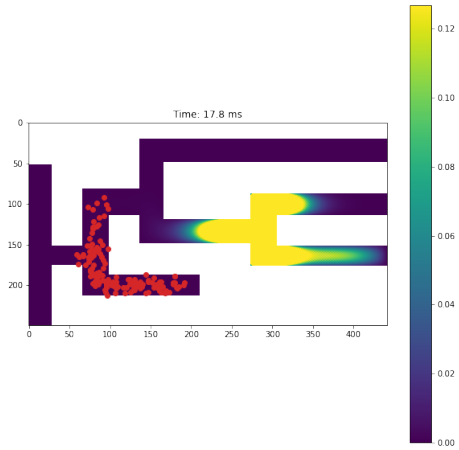
(b) Simulation run at time 2.3 ms



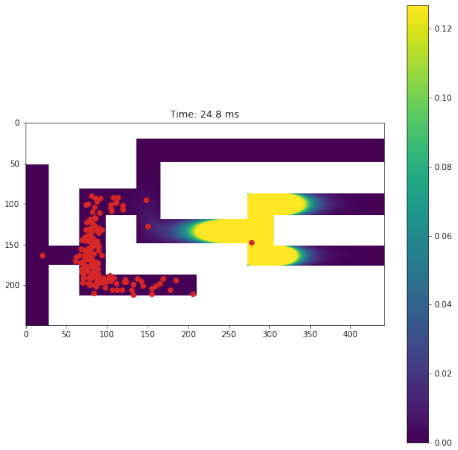
(c) Simulation run at time 5.3 ms



(d) Simulation run at time 12.8 ms

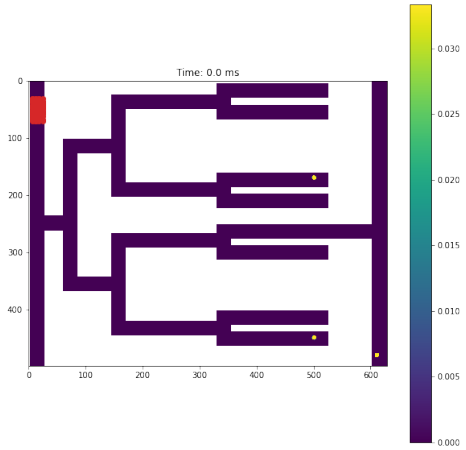


(e) Simulation run at time 17.8 ms

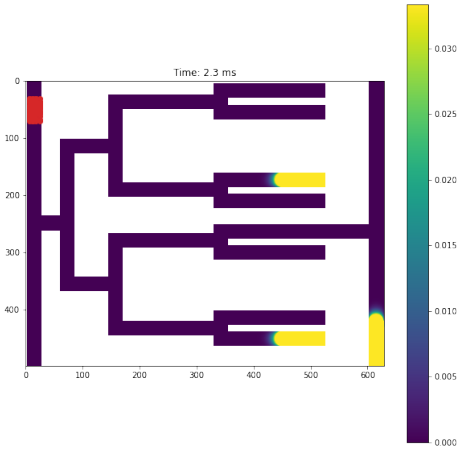


(f) Simulation run at time 24.8 ms

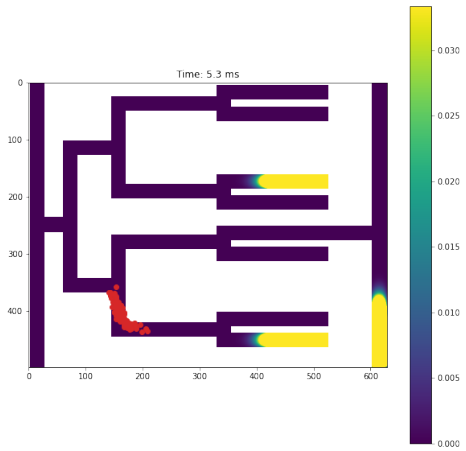
Figure 10: Simulation in a complex maze with two sources of a self-generated model of chemotaxis



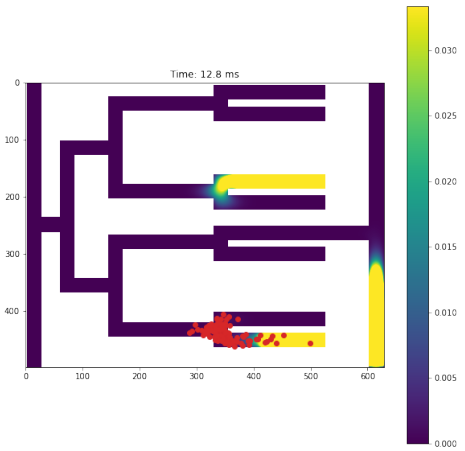
(a) Simulation run at time 0.8 ms



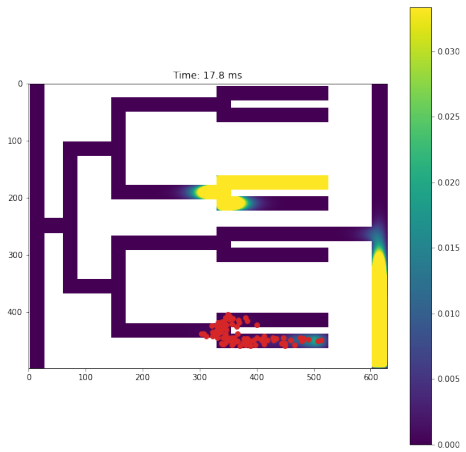
(b) Simulation run at time 2.3 ms



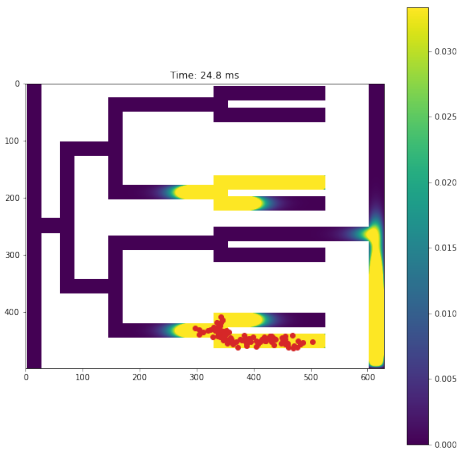
(c) Simulation run at time 13.3 ms



(d) Simulation run at time 32.3 ms

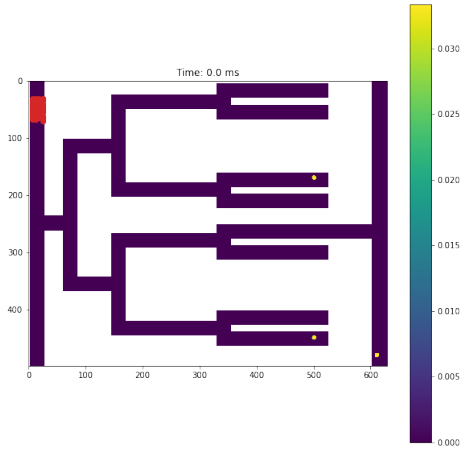


(e) Simulation run at time 44.9 ms

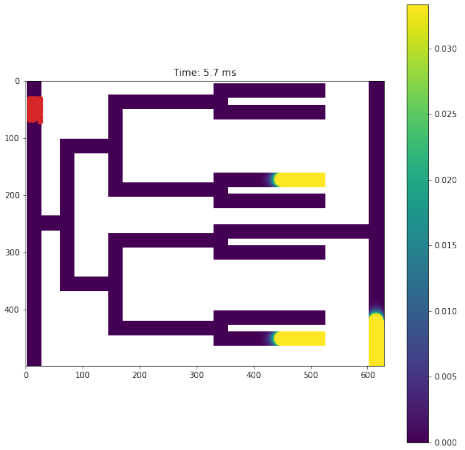


(f) Simulation run at time 62.6 ms

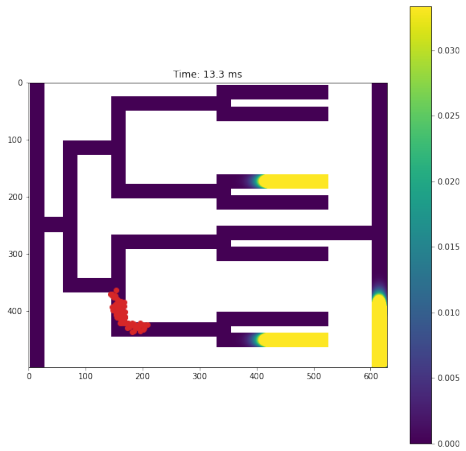
Figure 11: Simulation in a complex maze with three unequal sources of a classical model of chemotaxis



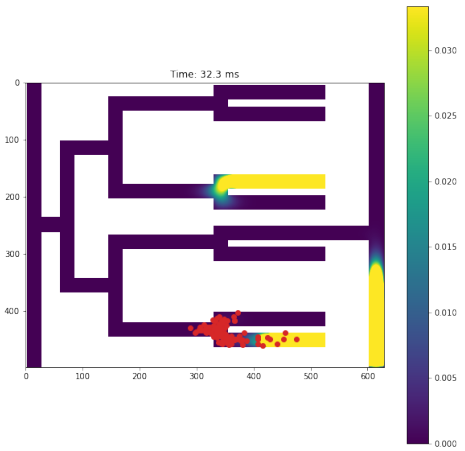
(a) Simulation run at time 0.0 ms



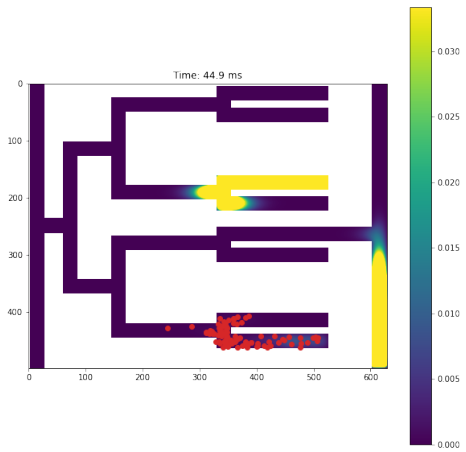
(b) Simulation run at time 5.7 ms



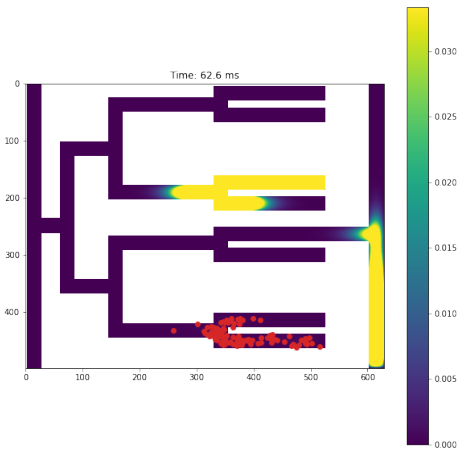
(c) Simulation run at time 5.3 ms



(d) Simulation run at time 12.8 ms



(e) Simulation run at time 17.8 ms



(f) Simulation run at time 24.8 ms

Figure 12: Simulation in a complex maze with three unequal of a self-generated model of chemotaxis