Assignment CPM

Making a Cellular Potts Model of Collective Migration

Deadline April 15th, 2020

Handout for the *Natural Computing* lecture, March 25th, 2020

Johannes Textor, Inge Wortel

In this assignment, you are going to use the Cellular Potts modeling framework to create cells *in silico*. You will first learn how to use the model's many parameters to create cells of realistic shapes and motility patterns. Then you will go a step further and model several migrating cells that interact with each other.

This assignment considers a slightly more advanced version of the Cellular Potts model than what was discussed in the lecture. The assignment therefore consists of two parts. First, there are three self-study exercises that walk you through the CPM using an interactive online version that we made. There is no need to write answers to these questions, these are simply meant for you to test your own understanding. Some questions are marked "Optional" which means that we don't consider these to be totally essential for completing this assignment, but you can use these to test how far your understanding of the model goes.

Second, we ask you to implement a simulation yourself in the CPM that is a small extension of the model considered in the third self-study exercise.

Objectives of This Exercise

1. Understand how the different parameters of a Cellular Potts Model (CPM) interact with each other, and how this influences cell behavior.
2. Realize that it can be difficult to tune the parameters of a CPM.
3. Apply this knowledge to create different modes of cell migration in the CPM, and explain how this helps understand those migration modes.
4. Apply the CPM to investigate the dynamics of collective cell migration *in silico*.

Please go to computational-immunology.org/cpm/collective.html. This is an implementation of a special version of the Cellular Potts Model in which cells can migrate [1]. Note that *this web page does not work properly in Internet Explorer*. It does work in either Firefox, Chrome, or Safari – so we recommend using any of those browsers for this exercise.

Notes on the CPM

Remember that in a CPM, copy attempts have a higher chance of succeeding if they lower the total energy H of the system. In other words, $\Delta H = H_{\text{after copy}} - H_{\text{before copy}}$ should be negative to guarantee that the copy attempt will work. Copy attempts with positive ΔH can still succeed, but do so with a lower probability that depends on the temperature of the system. In general, the formula for ΔH of a CPM looks something like this:

$$\Delta H = \Delta H_{\text{adhesion}} + \Delta H_{\text{Volume}} + \Delta H_{\dots} + \dots \tag{1}$$

Thus, we build up ΔH from all the different energetic factors we want the cell to consider – and we can always add stuff to make the model more complex. But what parameters do we need to tune to get realistic cells? In the following exercises, you will get a feel for how you can model cell behavior with a CPM by tuning the parameters that control the energy ΔH.

Exercises

Exercise CPM.1 A very basic CPM

In this exercise, we will first examine a very basic CPM in which cells do not (yet) migrate. Make sure that the field is empty (hit refresh or "remove all cells"), and that the parameters have the following values:

Adhesion$_{\text{cell-matrix}}$	Adhesion$_{\text{cell-cell}}$	Volume	λ_{Volume}	Perimeter	λ_{P}	Max$_{\text{Act}}$	λ_{Act}	T	Framerate
20	0	500	50	340	0	0	0	20	1

We will now investigate how the basic CPM parameters – controlling adhesion, cell volume, and cell perimeter (circumference) – influence behavior (this means you can ignore the λ_{Act} and Max$_{\text{Act}}$ parameters for now). This exercise is meant mostly to give you an idea of what the CPM parameters do, and the questions are to guide your thinking – so you don't have to write everything down. Try to spend no longer than 30-40 minutes on this exercise before continuing to the next.

1. Make sure all the parameters are set as in the table above, click "seed cell" and then "start". What do you see? What kind of motion does this cell have?
2. Now set the Adhesion$_{\text{cell-matrix}}$ to 0. What happens to the cell? Why do you think that happens? (Hint: look back to the description of the CPM and adhesion energy in the lecture...) Also try a negative value for Adhesion$_{\text{cell-matrix}}$. What is the meaning of positive or negative adhesion values here?
3. *(Optional)* Instead of setting the Adhesion$_{\text{cell-matrix}}$ back to 20, try setting the Adhesion$_{\text{cell-cell}}$ to -20 while having the Adhesion $_{\text{cell-matrix}}$ still at 0. Does that have the effect that you expected? Why/why not do you think that is? Hint: try drawing a grid like you saw in the lecture for a copy attempt you are interested in. Do the two adhesion energies change in the same way for that copy attempt?
4. Return to the parameters in the table above. With these parameters, the cell is given an ideal volume (500 pixels), and a "level of importance" of this volume for the energy (λ_{Volume}). Try making the cell bigger or smaller (what parameter should you change?). How can you make the volume unimportant for the energy – and what happens then? What happens when you make λ_{Volume} really large (say, 1000)?
5. So far, we have considered ΔH with only terms for adhesion energy and cell volume. We will now investigate the effect of the cell perimeter (circumference). The cell already has a target perimeter (340), but this is currently not taken into account in the calculation of ΔH. For that, we need to make λ_{P} non-zero. Try setting it to 2. What happens to the cell? Try making the cell "membrane" more or less ruffled. How would you do that?
6. Set the perimeter to 340 and λ_{P} to 2. Now change the Adhesion$_{\text{cell-matrix}}$ to 0 again. Does this have the same effect as it did in question 2? Why do you think that is?
7. *(Optional)* Play around with the volume and perimeter parameters for a while (using the adhesion parameters from the table, or try your own). How can you change the cell? Can you change the parameters independently of each other? And what happens if you change the temperature?

Exercise CPM.2 Cell migration: the Act model

We will now investigate the Act model, an extension of the CPM that allows the cells to migrate [1]. This model adds an extra term to the system energy, so that:

$$\Delta H = \Delta H_{\text{adhesion}} + \Delta H_{\text{Volume}} + \Delta H_{\text{Perimeter}} + \Delta H_{\text{Act}} \tag{2}$$

In this model, pixels that were recently added to the cell remember their recent "protrusive activity". This makes them more likely to protrude again. This positive feedback is controlled by the energy term ΔH_{Act}, which is negative (favourable!) when a recently active pixel tries to copy itself into a less active pixel. The Act model has two extra parameters: λ_{Act}, which controls how important the positive feedback is relative to the other ΔH energies, and Max$_{\text{Act}}$, which determines how long pixels "remember" that they were active. In this exercise, we will see what happens when we vary those two parameters of the model. In particular, we will see that we can reproduce two very different "modes" of migration: amoeboid and keratocyte-like (see lecture).

Before starting this exercise, please refresh the page and ensure that the parameters of the CPM are set as follows:

Adhesion$_{\text{cell-matrix}}$	Adhesion$_{\text{cell-cell}}$	Volume	λ_{Volume}	Perimeter	λ_{P}	Max$_{\text{Act}}$	λ_{Act}	T	Framerate
20	0	500	50	340	2	20	0	20	1

1. Seed a cell and click "start". You should now see colored pixels at the border of the cell, which indicate the "activity" that pixels remember (because we have set $\mathrm{Max}_{\mathrm{Act}}$ to 20). Other than the color of the pixels, does the cell behave in a different way than with $\mathrm{Max}_{\mathrm{Act}} = 0$? Why/why not?
2. Set λ_{Act} to 100. Would you describe this movement as random or persistent?
3. *(Optional)* What happens when you set λ_{P} to 0 now? Why do you think that is? (Reset it to 2 before going to the next question)
4. What happens when you increase λ_{Act} further? (Try steps of 100).
5. *(Optional)* If you increase λ_{Act} to very high values (eg 1000), the cell is prone to breaking in pieces. Can you fix that by altering some other CPM parameter again? (Note: you may have to increase λ_{Act} further when you have done this... Does that make sense to you?)
6. Reset λ_{Act} to 0, change $\mathrm{Max}_{\mathrm{Act}}$ to 80, and repeat questions 1,2, and 4 above. What do you see?
7. *(Optional)* If you have time, play around with different combinations of λ_{Act} and $\mathrm{Max}_{\mathrm{Act}}$. Can you get a clue of what they are doing – beyond the mathematical description given above?
8. *(Optional)* Try halving or doubling the cell's target volume. That won't work. What do you need to change to get the same behavior as before? What does that mean for your model (in other words: to what extent are your choices of parameters important for the behaviour you see? How worried should you be about getting parameters "wrong" and drawing the wrong conclusions?)?
9. The nice thing about computer models is that you know exactly which "rules" you put in, and therefore that those are the only rules that the cells will follow. However, the behavior that follows from those "rules" is not always obvious (that's why we run the simulation in the first place!). In fact, the most interesting computer models can sometimes produce behavior that you did not explicitly put in there. We call this "emergent" behavior. Can you think of an example of emergent behavior in the Act model? What does that tell you about migration in a living cell?

Exercise CPM.3 Migration in a multicellular system

In this last exercise, we will investigate what happens when there are many cells. Before starting, please refresh the page to clear the grid, and then ensure that the parameters have the following values:

$\mathrm{Adhesion}_{\text{cell-matrix}}$	$\mathrm{Adhesion}_{\text{cell-cell}}$	Volume	$\lambda_{\mathrm{Volume}}$	Perimeter	λ_{P}	$\mathrm{Max}_{\mathrm{Act}}$	λ_{Act}	T	Framerate
20	0	200	50	180	2	20	200	20	5

Note that the framerate is not actually a parameter of the model, but it specifies how often the program draws the updated grid (eg framerate of 5 means visualize only 1 in every 5 "frames" of the movie). Setting it to 5 will speed up the animation.

1. To seed many cells at once, click "+10 cells" a few times (this will take a while...). What do you see? How does the behavior change as the grid becomes more densely packed with cells?
2. Now, refresh the page, reset the parameters, increase $\mathrm{Max}_{\mathrm{Act}}$ to 80, and repeat the previous exercise. What happens? How is this different from what you saw with $\mathrm{Max}_{\mathrm{Act}} = 20$? How full can you make the grid before the behavior really changes (if it ever does...)? If you're in doubt, change $\mathrm{Max}_{\mathrm{Act}}$ back to 20 and compare what happens with a full grid versus a single cell at this parameter value...

Assignment

Now that you have completed the self-study part of this assignment, you have all the necessary background information to implement your own model to investigate the "sensitivity" of your observation to changes in the model. Specifically, we will look what happens when obstacles are present.

For this part of the exercise, we recommend that you use our framework CPMjs, written in the JavaScript language. (Of course, feel free to try implementing a CPM yourself in any programming language you wish if you prefer to do so – you should just realize that this could take you quite some time...)

You can find the CPMjs framework at https://github.com/ingewortel/cpmjs. It contains many methods allowing you to build and visualize CPMs, for which you can find documentation and some tutorials at https://ingewortel.github.io/cpmjs.github.io/. See the tutorials "Getting started" and "Your first simulation" under the "Manual" tab to start using the framework. We also recommend having a look at the examples in the `examples/html/` folder of the code repository. If you encounter any problems or have any questions, please use the forum on Brightspace and we will try to get back to you as soon as possible.

In the first part of the exercise, we have seen that "crowding" (having a lot of cells close together) can impact movement under some conditions, but not others. We will now investigate how obstacles change the motion of cells on a densely packed grid.

1. How could you model an obstacle in the CPM? Hint: can you make a round obstacle by changing the CPM parameters in a certain way?
2. Build your own simulation to investigate the effect of obstacles. Hint: you can do this by building a simulation with two kinds of cells – see e.g. `Cellsorting.html`, `ManyCellsPrefDir.html`, or `examples/html/EpidermisWithTCells.html` in the `examples/html/` folder.
3. From the first part of the exercise, start from the cells that kept moving even at large densities. What happens when you place obstacles between them? Do they still move? What happens if you increase the number of obstacles? Please use obstacles with half the size of the cells for this exercise, and place them on the grid with regular spacing between them (Hint: You will have to modify the `initializeGrid` method of your simulation. Have a look at the `seedCellAt` method and the `examples/html/CancerInvasion.html` example).

Report

- Write a little report (at most ~ 4-5 pages) on the exercise on collective migration. Describe how you have implemented the obstacles in the CPM, and what effects these obstacles have on collective cell migration.
- Where necessary, you can add screenshots to illustrate the behavior you see, and one or two screen recordings of the simulations that you based your conclusions on. Think about visualizations - how can you best show your point? For example, you can make screenshots of simulations, you can attempt to measure the motion of the cells, or you could even make videos and upload them with your assignment. Please precisely specify which parameters you have used for your simulation, for example, by mentioning these in a table.
- The deadline for submission is April 15th (see Brightspace).

References

[1] I. Niculescu, J. Textor, and R. J. de Boer. Crawling and Gliding: A Computational Model for Shape-Driven Cell Migration. *PLOS Computational Biology*, 11(10):1–22, 2015.