Lab 5 – Cell Based Models

Week 5: In this last assignment we will take a sneak peak into a completely different domain of computational Biology; Cell Based Modeling. This assignment is more of an open investigation and exploration of the Cellular-Potts model.

Background

Agent based modeling concerns the simulation of relatively simple agents. These agents behave according to certain rules, interacting with their environment and surrounding agents. Agent based models where each agent only senses its direct neighbours are called Cellular Automata (CA). Conway's Game of Life is an example of a two dimensional CA that can produce very complex patterns and these patterns can even be used for computation (and emulating itself) since it is Turing complete.

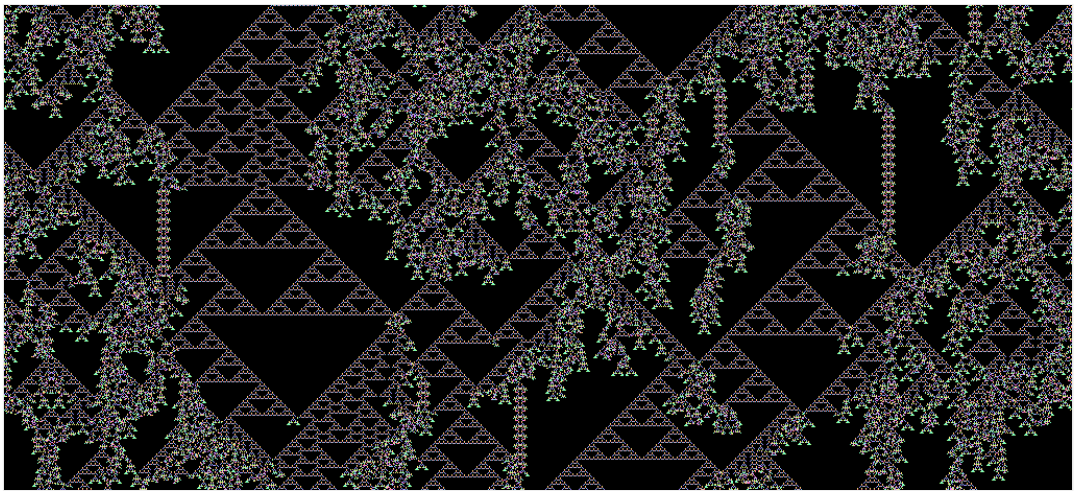


Figure 1: One of the simplest agent based models is the 1D Cellular Automaton, there is only one spatial (horizontal) coordinate and subsequent simulation steps are shown from top to bottom. In the CA shown above, each pixel can be in one of six states and the state it will take on next depends only on its current state and those of its two direct neighbours.

Biology is full of complex systems that consist of many parts that interact with each other. Often the whole system behaves in a very complicated way while the parts of the system seem relatively simple and are well understood. The systems that we can model range from whole ecosystems to single cells or even parts of cells. Depending on the system and the behaviour that we wish to investigate, we define the agents to be a whole species, groups of organ-

isms, single organisms, a single organ, a piece of tissue, a single cell, parts of a cell or even single molecules. When we simulate single cells to study the collective emergent behaviour of the resulting tissue or cell culture we use the term cell-based model. Cell based models exist in many forms and a wide range of complexity. From simple extensions of the above mentioned CA to simulate specific concepts or parts of tissue, to three dimensional lattice-free simulation of whole organisms.

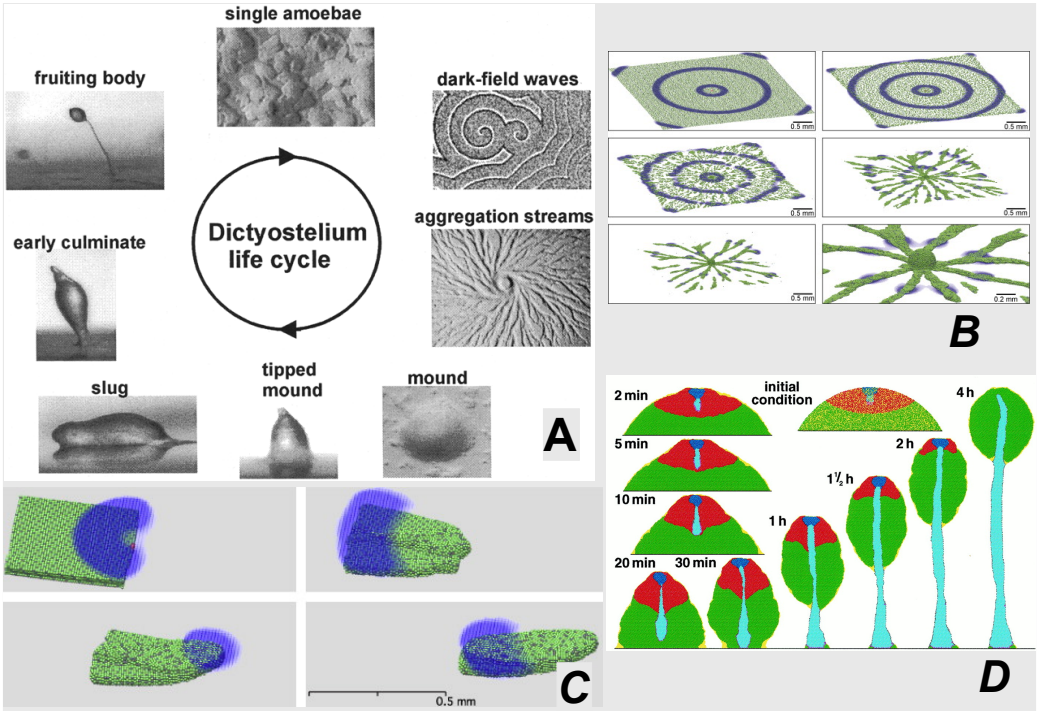


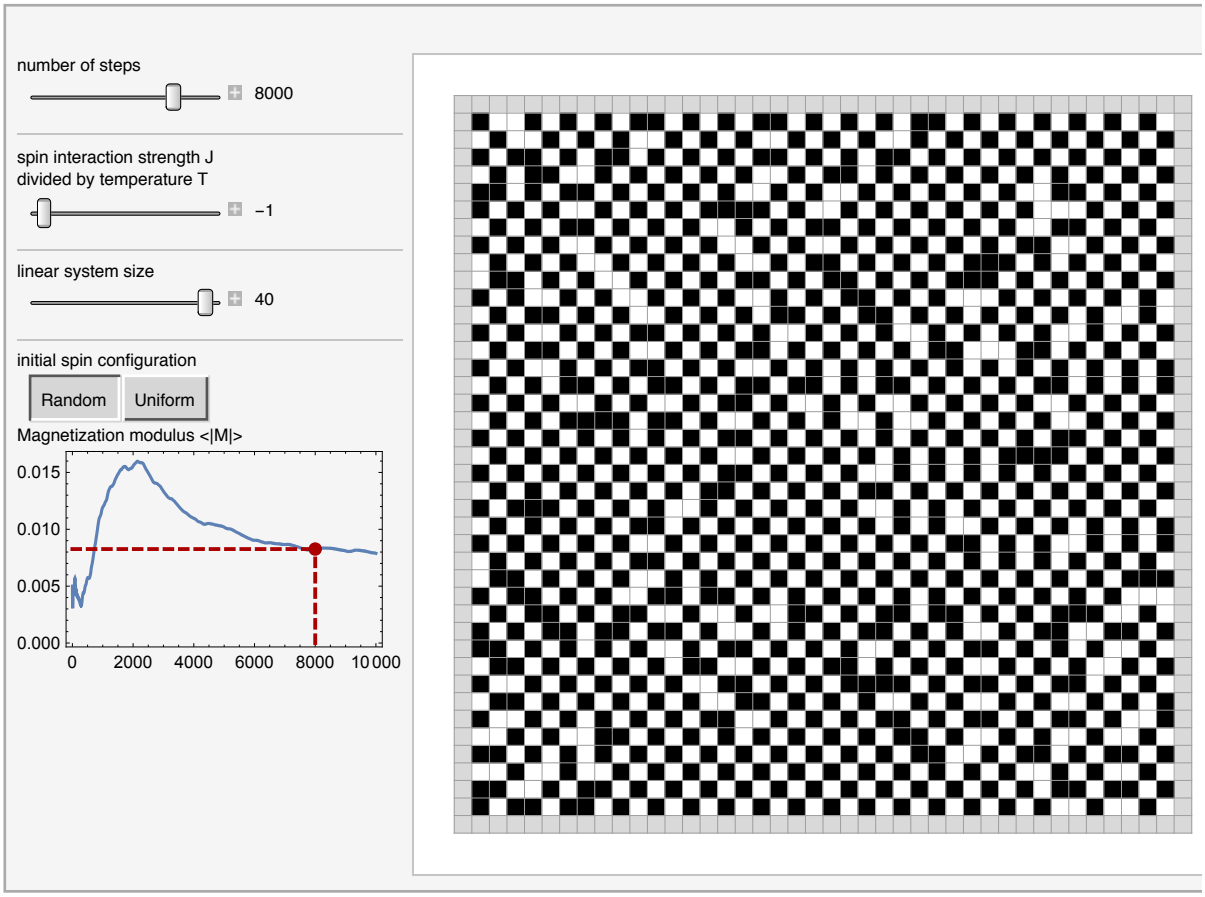
Figure 2: . The complete lifecycle (**A**) of the Dictyostelium organism has been modeled using extended Cellular-Potts models. Dictyostelium is both a single- and multi-celled organism as it can survive and reproduce asexually in single cell state but it will form larger bodies that can move to survive starvation and to reproduce sexually. Modelling the single Amoebae, the dark-field waves and mound phase (**B**), crawling slug phase (**C**) and the culminate and fruiting body phase (**D**) can be simulated using only local rules and a few longer ranged mechanisms.

The Ising-model and the Metropolis algorithm

In this assignment you will be given a relatively simple model and you will work towards a more complex lattice-based model of cells. In this we follow the actual historic development of the Cellular-Potts model. The Ising model was originally constructed for the modelling of spins of single atoms and the emergent behaviour of ferro-magnetism. However it can be used to study a wide variety of systems in biology concerning binding behaviour of atoms, molecules, or even cells when it is slightly altered. Here we have an implementation of the Ising model. Since this model calculates the complete evolution

until the last time step before showing the result, the sliders may respond slowly.

```
Manipulate[Module[{ic, TL, TLT}, SeedRandom[1234];
  ic = If[ch, ArrayPad[RandomInteger[{1, 2}, {nx, nx}], 1] /. 2 -> -1,
   ArrayPad[Table[1, {i, 1, nx}, {j, 1, nx}], 1]];
  TL = NestList[accept[#1, jt] &, ic, nsteps];
  TLT = Accumulate[Abs[Total /@ Total /@ TL]] /
    (nx * nx * Range[nsteps + 1]); LPD = ListLinePlot[TLT,
   Epilog -> {Darker[Red], PointSize -> Large, Point[{k, TLT[[k]]}],
    Thick, Darker[Red], Dashed, Line[{{k, 0}, {k, TLT[[k]]}}],
    Line[{{0, TLT[[k]]}, {k, TLT[[k]]}}]}, PlotRange -> Full,
   Frame -> True, Axes -> False, ImageSize -> 200];
  ArrayPlot[TL[[k]], Mesh -> All, ColorRules ->
   {0 -> LightGray, 1 -> Black, -1 -> White}, ImageSize -> 400]],
 Item["number of steps"], {{k, 8000, ""}, 1, 10 000, 1,
 Appearance -> "Labeled", ImageSize -> Small}, Delimiter,
 Item["spin interaction strength J\ndivided by temperature T"],
 {{jt, -1, ""}, -1, 3, Appearance -> "Labeled", ImageSize -> Small},
 Delimiter, Item["linear system size"], {{nx, 40, ""}, 10, 40, 1,
 Appearance -> "Labeled", ImageSize -> Small}, Delimiter,
 Item["initial spin configuration"], {{ch, True, ""},
 {True -> "Random", False -> "Uniform"}},
 {{LPD, Graphics[]}, ControlType -> None}, Delimiter,
 Item["Magnetization modulus <|M|>"], Dynamic[Item[LPD]],
 ControlPlacement -> Left, SynchronousUpdating -> False,
 AutorunSequencing -> {1}, TrackedSymbols :> {k, jt, nx, ch},
 Initialization :> (nsteps = 10 000; accept[m_, z_] :=
  Module[{ech, x1, y1}, {x1, y1} = RandomInteger[{2, Length[m] - 1},
    2]; ech = m[[x1 - 1, y1]] + m[[x1 + 1, y1]] + m[[x1, y1 - 1]] +
    m[[x1, y1 + 1]]; If[RandomReal[] < Exp[(-ech) * z * 2. * m[[x1, y1]]],
   ReplacePart[m, {x1, y1} -> -m[[x1, y1]]], m]])]
```



The Ising model is a Monte-Carlo type simulation and uses the so called Metropolis algorithm to evolve the system in time. The Metropolis algorithm changes a random lattice site to a different state each step and calculates the difference caused by this change in some predefined energy or Hamiltonian. Then it is decided whether the change is accepted or rejected (model returns to previous state) based on the energy difference that was calculated. In the case of the Ising model the probability of accepting the change is $P = exp(-\Delta\mathcal{H}/kT)$ (where the change is always accepted when P>1). And the Hamiltonian is given by the sum of all spin-spin pairs the lattice site forms with its neighbours (being -1 if they have opposing spins and 1 if they have same spin), times some interaction strength J.

The Cellular-Potts model

In 1951 Renfrey Potts extended the Ising model to incorporate more than two spin states and it was called the Potts model. In 1992 Graner and Glazier used an version of the Potts model to model cellular patterns in soap, where a spin state $\sigma(i, j)$ is defined uniquely for every single cell. In this so called *large-Q Potts model*, each cell can consist of several lattice sites. Next to a

unique cell ID, each cell also belongs to a cell type $\tau(\sigma(i,j))$, in general the number of cell types is much smaller than the number of cells. The spin interaction Hamiltonian is now specified ta account for bonding between lattice site of the same cell type and a new term is added to the Hamiltonian next to the bonding energy to push each cell's volume to a specified equilibrium. This last term is the implementation of the physical tension that a cell builds up when it is compressed or pulled apart.

Model Description Summary

Agents

Each lattice site (i,j) belongs to a cell with a cell identity $\sigma(i,j)$ that uniquely identifies the cell. Additionally each cell has a cell type $\tau(\sigma(i,j))$. There are usually only a few cell types of which one is the special type M for the medium in which the cells are immersed.

Metropolis step

1. Randomly choose lattice site (i,j)
2. Randomly choose neighbour (i',j') within the Moore neighbourhood of (i,j)
3. Calculate the change in energy (Hamiltonian) to the system caused by setting $\sigma(i,j) \leftarrow \sigma(i',j')$
4. Apply the following transition rule:
   - if $\Delta\mathcal{H} \leq 0$ then set $\sigma(i,j) \leftarrow \sigma(i',j')$
   - if $\Delta\mathcal{H} > 0$ then set $\sigma(i,j) \leftarrow \sigma(i',j')$ with probability $P = exp(-\Delta\mathcal{H}/kT)$

   where k is the Boltzmann constant (which we can set to 1) and T > 0 is a simulation parameter called temperature (as it has somewhat analogous to the thermodynamic quantity)

Cellular-Potts Hamiltonian

$$\mathcal{H} = \mathcal{H}_{vol} + \mathcal{H}_{bond}$$

$$\mathcal{H}_{vol} = \lambda \sum_{\sigma \neq M} \left(a(\sigma) - A_{\tau(\sigma)}\right)$$

where $a(\sigma)$ is the area of a certain cell and $A_{\tau(\sigma)}$ is the equilibrium area of a cell type $\tau(\sigma(i,j))$.

$$\mathcal{H}_{bond} = \sum_{(i,j),(i',j')} J\left\{\tau\left(\sigma(i,j)\right),\tau\left(\sigma(i',j')\right)\right\}\left\{1-\delta_{\sigma(i,j),\sigma(i',j)}\right\}$$

where $\delta_{\sigma(i,j),\sigma(i',j)}$ is the Kronecker delta and J some interaction strength depending on the type of both lattice sites.

Many extensions can be made on the Cellular-Potts model, an extra term can be added to the Hamiltonian that depends on the size of the cell boundary. This is necessary when negative bonding strengths are explored. Also the model could be coupled with a set of differential equations describing chemotaxis. This is exactly what Savill and Hogeweg (1997) did when they modelled part of the lifecycle of Dictyostelium, shown in Figure 2.

Exercises/Guidelines

These exercise are formulated less like closed questions, they are more like guidelines. I just want you to explore this model and try explore its possibilities. Again make a small report on your findings.

1. Constructing the Cellular-Potts model

As said we start from a working Ising model and we will generalize this to the Cellular-Potts model. Below you find two blocks of code that form a simplified version of the Ising model above. These can be used as skeleton code to build the Cellular-Potts model as it is structurally very similar. It's you job to construct a Cellular Potts model from this. Mind that you need to take account of two states per cell now, the cell ID and the cell type. Updating two grids each step could work, but these two states are not completely unrelated so there may be smarter ways to implement this. You'll also have to find a way to plot both the cell type and the boundaries between individual cells. Cells can span more than one lattice site so think of removing the grid that is shown above and construct your own grid.

In[7]:=
```
nx = 40; (*Lattice size*)
nsteps = 10 000;
accept[m_, j_] := Module[{hamiltonian, x1, y1},
  (*We choose a random lattice site in the model*)
  {x1, y1} = RandomInteger[{2, Length[m] - 1}, 2];
  (*The Ising model Hamiltonian is calculated*)
  hamiltonian =
   j * (m[[x1 - 1, y1]] + m[[x1 + 1, y1]] + m[[x1, y1 - 1]] + m[[x1, y1 + 1]]);
  (*Then we apply the metropolis algorithm,
  get a random variable and decide wether the new state is accepted*)
  If[RandomReal[] < Exp[-hamiltonian * 2. * m[[x1, y1]]],
   ReplacePart[m, {x1, y1} → -m[[x1, y1]]],
   m]]
```

In[10]:=
```
Manipulate[
 (*A random seed is set so we get
  reproducable results for testing the model*)
 SeedRandom[1234];
 (*The grid is initialized and then
  a list of all subsequent grids is constructed*)
 typeGrid = ReplaceAll[ArrayPad[RandomInteger[{1, 2}, {nx, nx}], 1], 2 → -1];
 typeList = NestList[accept[#, jt] &, typeGrid, nsteps];
 (*From here it's all just plotting and manipulation features*)
 ArrayPlot[typeList[[k]], Mesh → All,
  ColorRules → {0 → LightGray, 1 → Black, -1 → White}, ImageSize → 400],
 Item["number of steps"],
 {{k, 1, ""}, 1, 10 000, 1, Appearance → "Labeled", ImageSize → Small},
 Delimiter,
 Item["spin interaction strength J/T"],
 {{jt, -1, ""}, -1, 3, Appearance → "Labeled", ImageSize → Small},
 {{LPD, Graphics[]}, ControlType → None},
 Delimiter,
 Item["Magnetization modulus <|M|>"],
 Dynamic@Item[LPD],
 ControlPlacement → Left,
 SynchronousUpdating → False
]
```

Out[10]=

2. Modeling Cell-sorting

Now we want to gather some statistics about cell sorting in the Cellular-Potts model. Think of interesting metrics that tell us something about the whole system and run some simulations with different parameter settings to explore the parameter space. See the magnetization plot in the example Ising model for inspiration on how to implement this. Mind that if you want to do real measurements, you should disable the random seed that was set at the start to make the model truly stochastic. See if you can also implement the extension to the Hamiltonian that is necessary for negative values of J.

References

- Various sources from wikipedia
- Tamulonis, C. (2013). Cell Based Models. Self published.
- Graner, F. and Glazier, J.A. (1992). Simulation of biological cell sorting using a two-dimensional extended Potts model. *Phys. Rev. Lett.* **69:** 2013-2016.
- Savill and Hogeweg (1997). modelling Morphogenesis: From Single Cells to Crawling Slugs. *Journal of Theoretical Biology* ***184***:229-235.