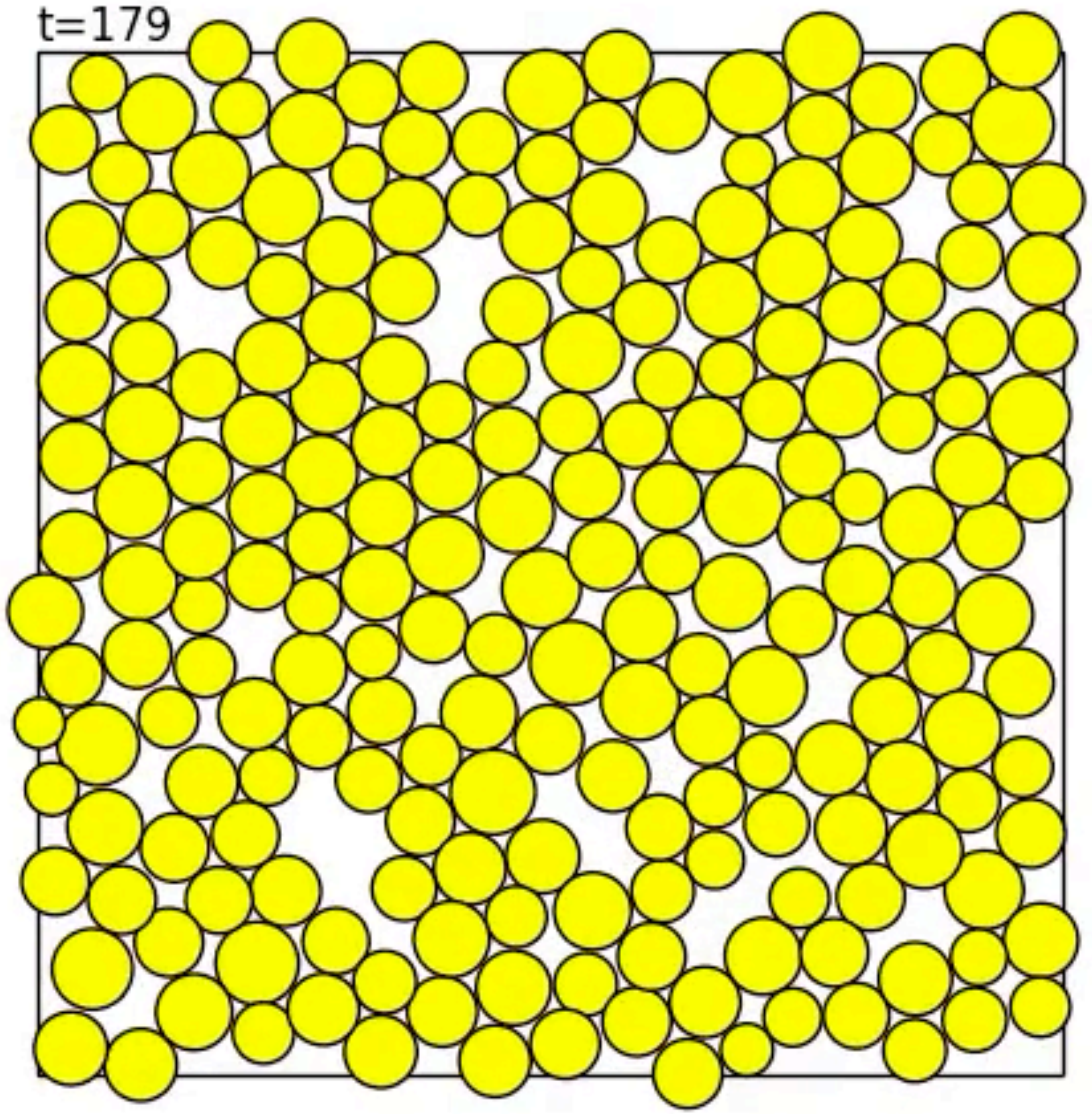
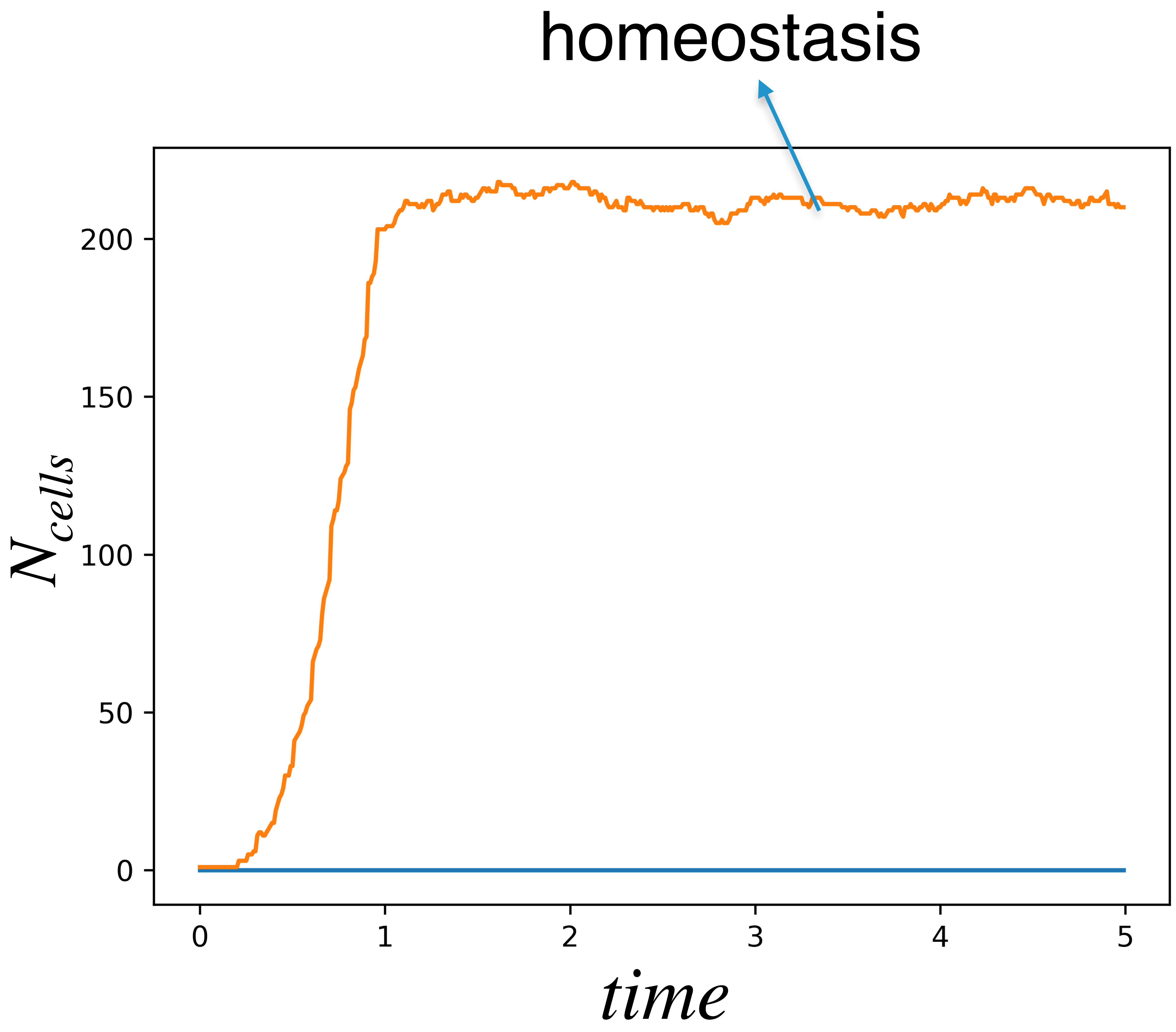
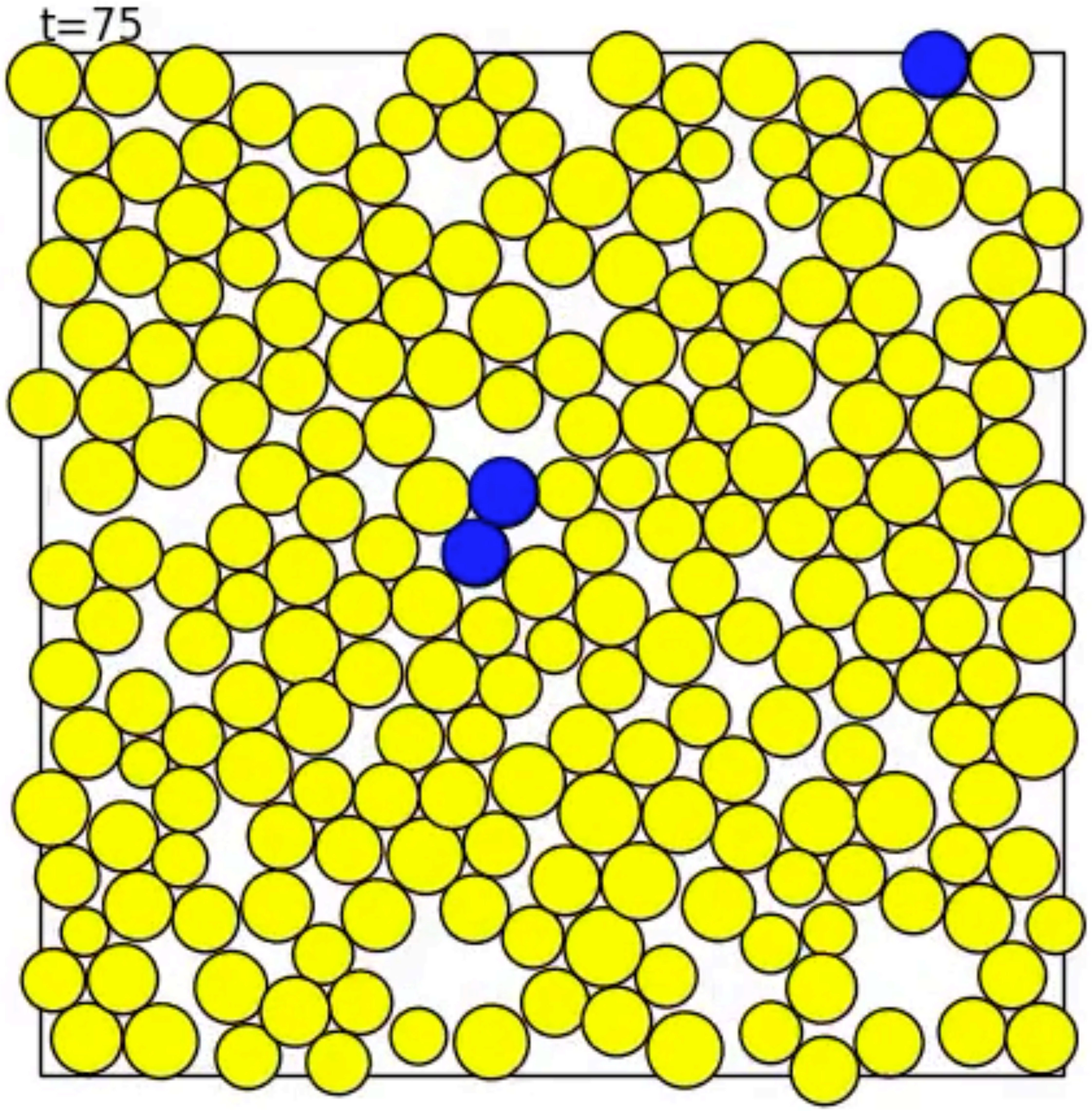
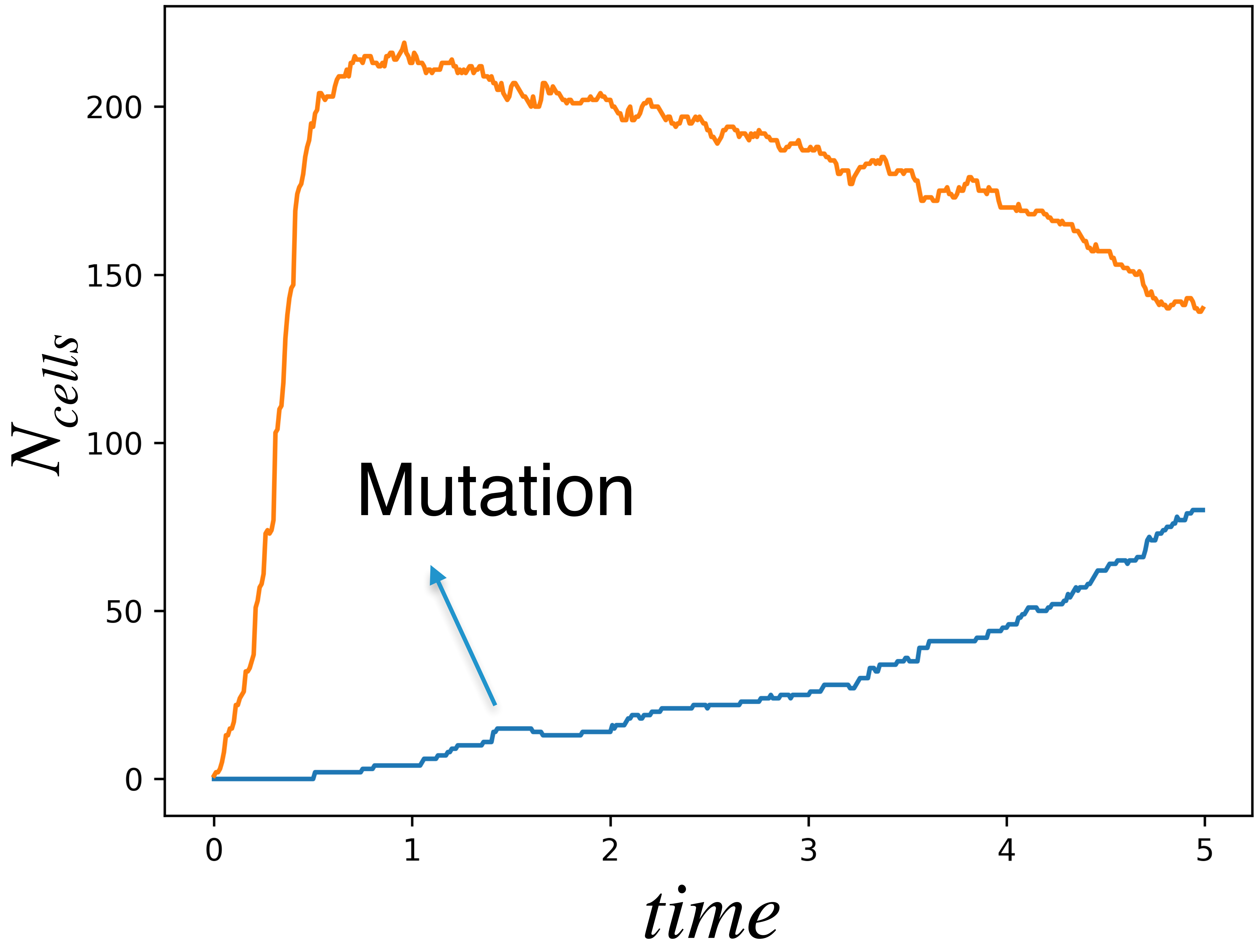
LAB: Tissue sorting and competition

Daniel Matoz-Fernandez

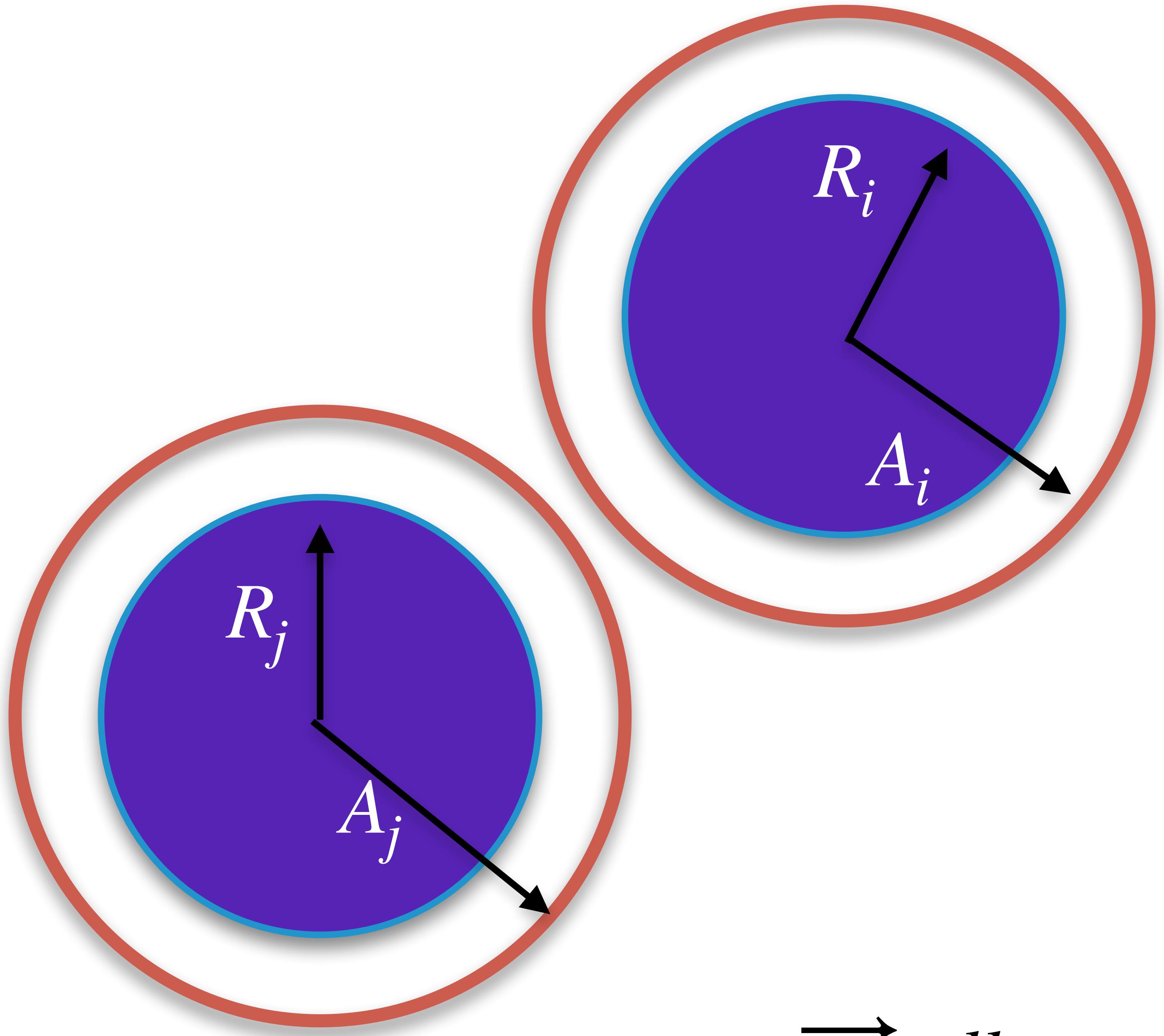

t=179


homeostasis
200
150
100
50
0
N_{cells}
0
1
2
3
4
5
time


t=75


200
150
100
50
0
0
1
2
3
4
5
N_{cells}
time
Mutation

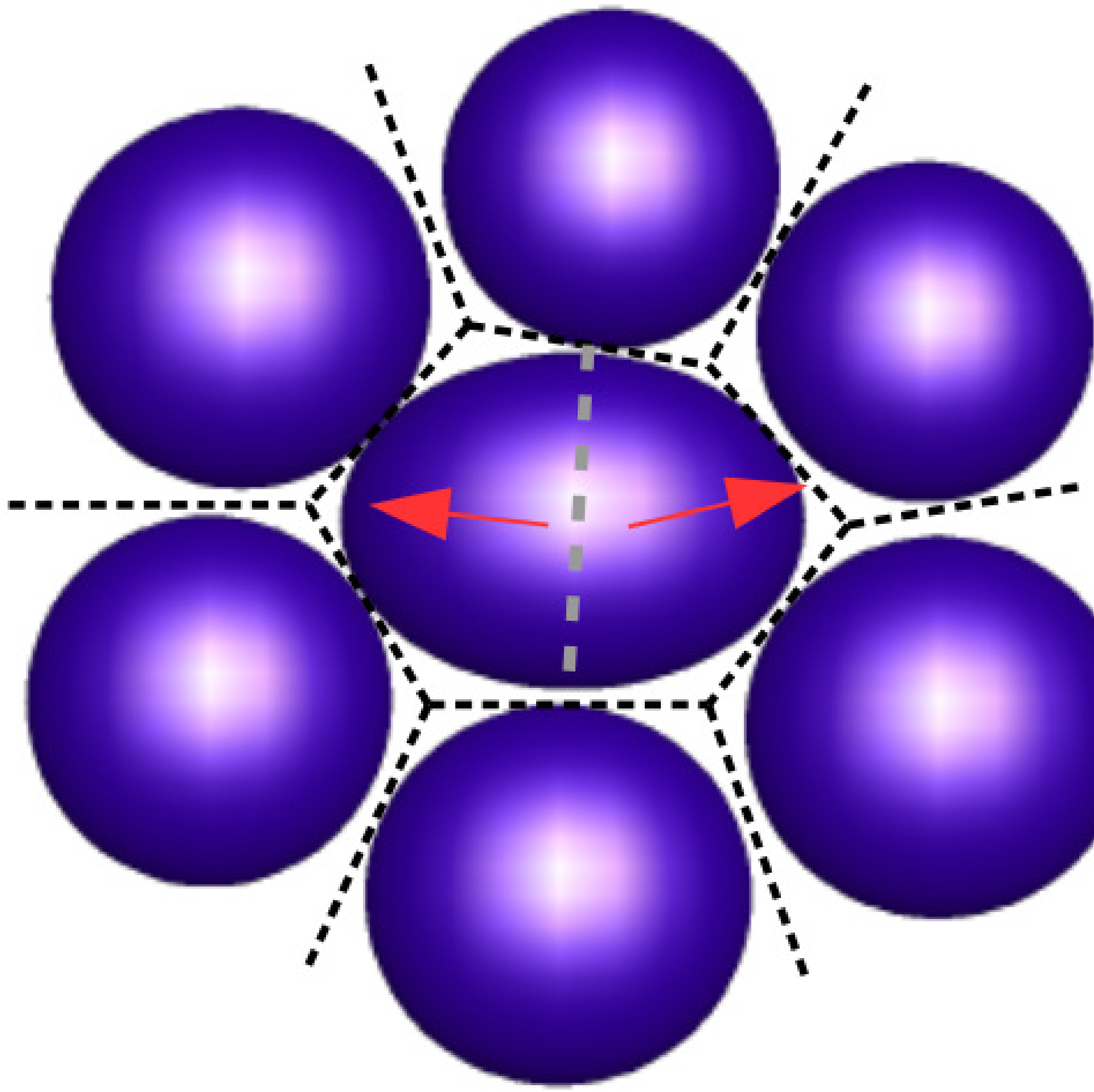
Presentation of the model



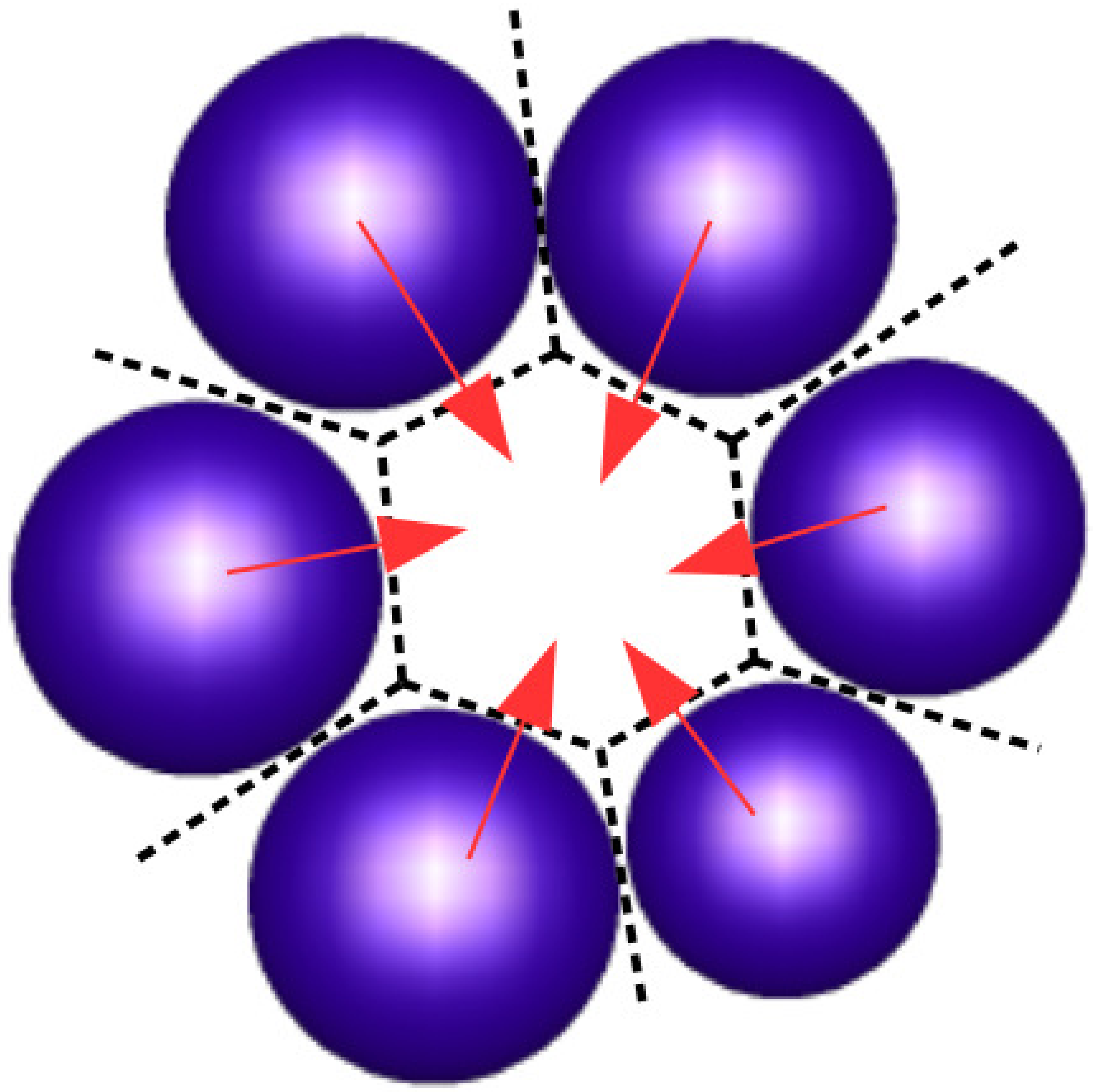
$$\overrightarrow{F}_i^{cell} = \overrightarrow{F}_{\textbf{EV}} + \overrightarrow{F}_{\textbf{atractive}}$$

$$\overrightarrow{F}_i^{cell} = \begin{cases} -k_{repulsive}\left((R_i + R_j) - r_{ij}\right)\vec{\hat{r}}_{ij} & \textbf{if } r_{ij} < (R_i + R_j) \\ k_{attractive}\vec{\hat{r}}_{ij} & \textbf{if } (R_i + R_j) \leq r_{ij} \leq 1.1 * (R_i + R_j) \end{cases}$$

Presentation of the model

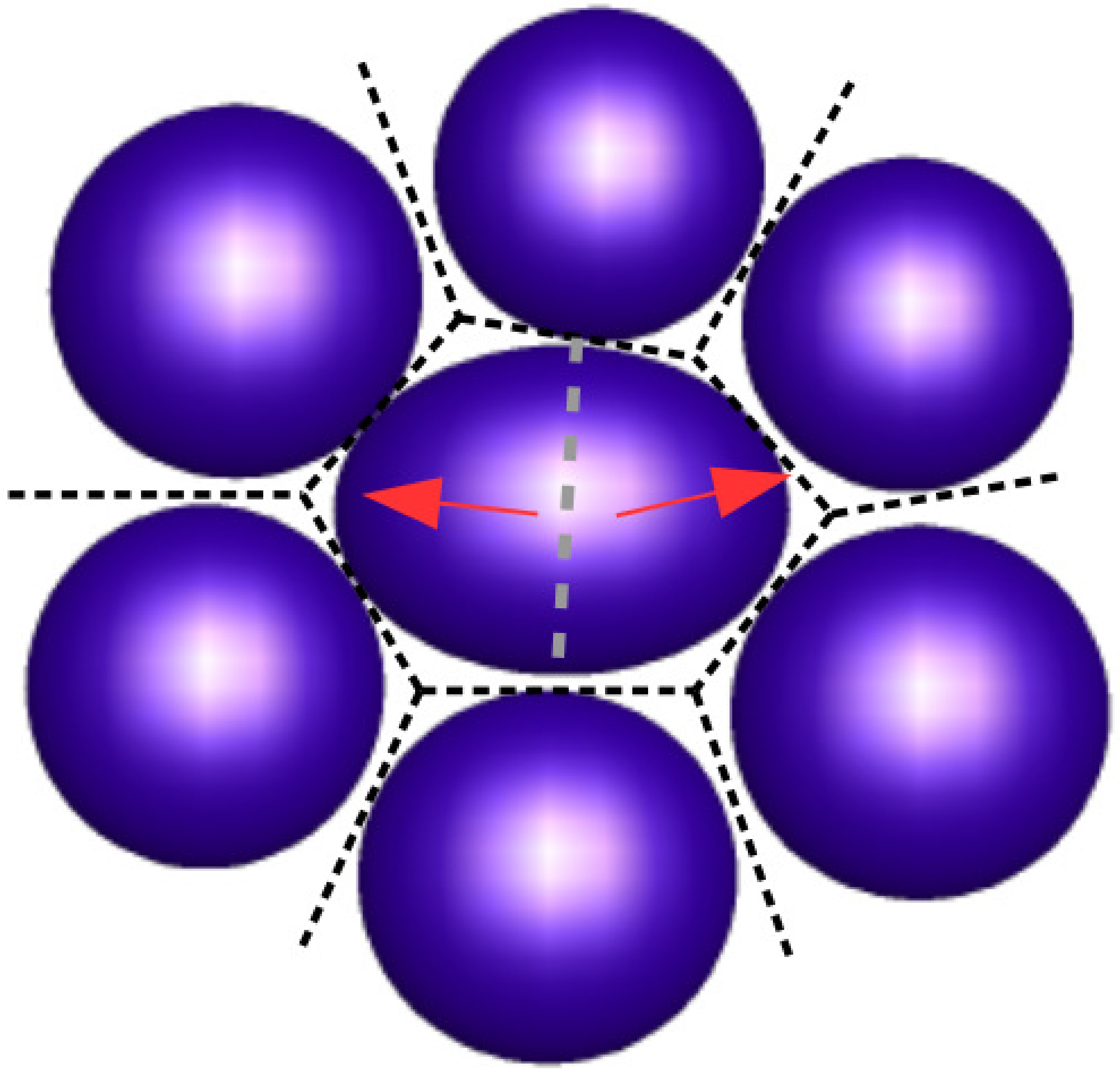


Division



Dead

Cell division



Division

Create a new cell as follows:

(1) The cell can divide at a rate given by:

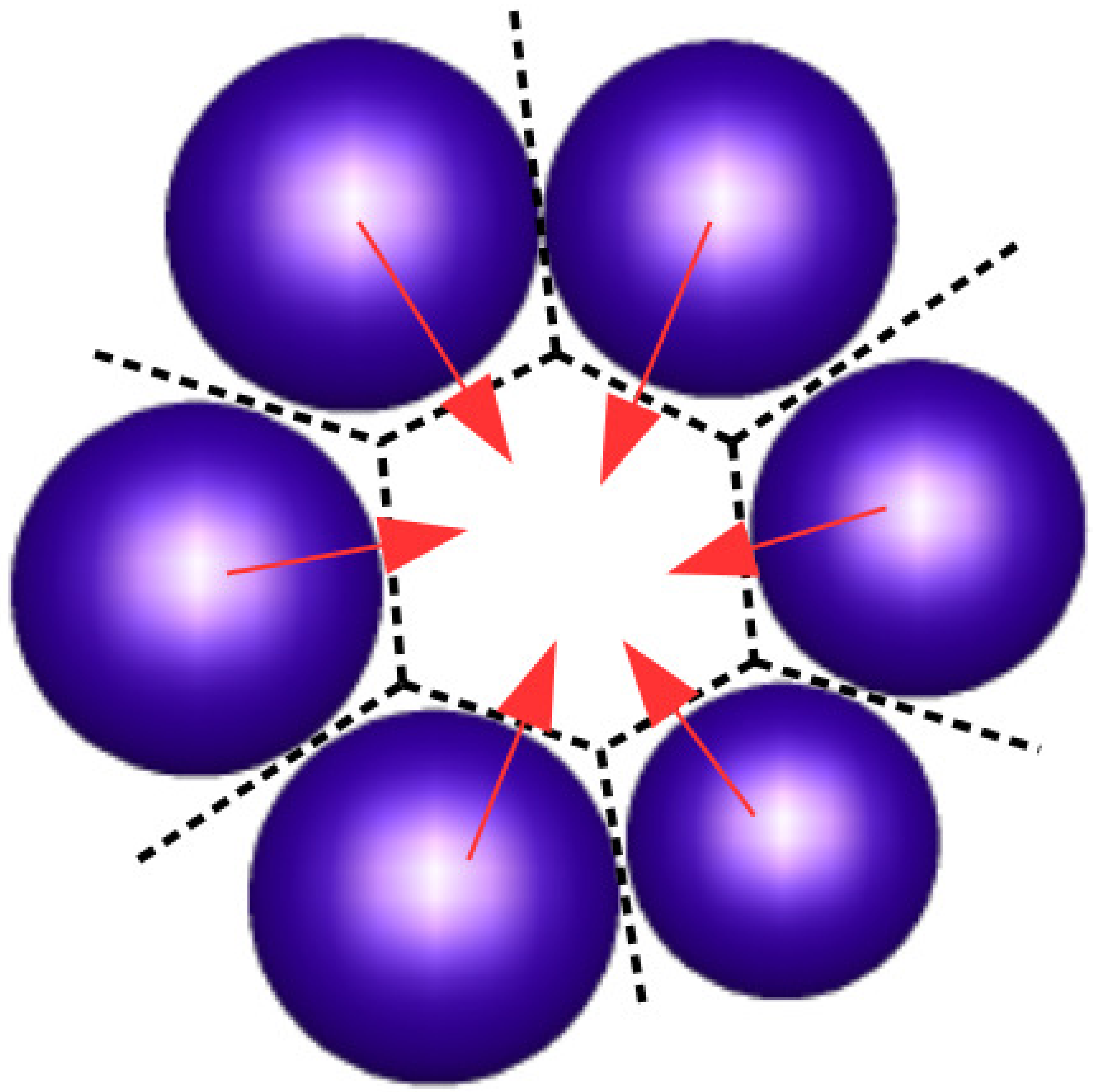
$$k_d = k_d^0\left(1.0 - \frac{Z_{ng}}{6}\right),$$

where k_d^0 is the division rate and z_{ng} is the number of close neighbors.

(2) The new daughter cell interacts with the same repulsive potential but attenuated at 50%.

(3) The new daughter separates when the distance between mother-daughter Is larger than $0.98(R_{mother} + R_{daughter})$. When they separate, they become "normal." cells and they can divide again by (1). Otherwise, see (4)

(4) Cell that are in the process of division because they do not fulfill (3) they cannot divide



Dead

Cell Dead

Create a new cell as follows:
(1) The cell died at a rate given by:

$$k_a,$$

where k_a is the dead rate.

(2) Cells that are dividing cannot die.

Numerical Integration

$$\dot{\vec{r}}_i = \vec{F}_i + \vec{F}_R \qquad \left\langle \vec{F}_R(t) \right\rangle = 0 \qquad \left\langle \vec{F}_R(t) \cdot \vec{F}_R(t') \right\rangle = \alpha\delta\,(t - t')$$

$$\vec{r}_i(t + \Delta t) = \vec{r}_i(t) + \vec{F}_i \Delta t + \sqrt{\alpha \Delta t}\,\vec{\eta}$$

Random Vector from a Gaussian Distribution

Remember to increase the time that a cell has been alive when you integrate the equation of motion!!

Code Scheme

```
Nsnaps # Number of Pictures
Nrun # Run Steps until a picture is taken
Cell_Processes_Check # how often we check if the cell can
                  # divide or died. Note we don’t have to check each time step.
                  # The rates k_d and k_a then change to k_a(t) = cell.time*k_a.
                  # Same for k_d(t) = cell.time*k_d.
                  # Here cell.time is the time that a given cell has been alive
                  # Each process is trigger at random so if ran() < k_a(t) then the cell died, same for k_d
for snap in range(Nsnaps):
    dump(cells, L, 't={}'.format(snap), False)


    for run in range(1, Nrun+1):
        #Reset Forces
        …
        #Compute Forces
       …
     #Integrate equations of motion
        …
     #Apply Periodic boundary condition
    …

     if (snap*Nrun+run) % Cell_Processes_Check == 0:
      #Check for cell division
      …
      #Check for cell dead
      …
```

Code Scheme: cell divide

```
def cell_divide(cell, new_id):
#use the position of the mother plus random to create the daugther
    new_pos = cell.r + (1e-1)*np.random.rand(2)
#apply periodic boundaries to the new daughter in case that the cell is at the edge
    apply_periodic(new_pos, L)
#select a new radius with a normal distribution for both daugther and mother
    new_radii = np.random.normal(1.0, 1e-1)
    cell.R = np.random.normal(1.0, 1e-1)
#finally create a new cell
    new_cell = CellClass(x=new_pos[0],
                         y=new_pos[1],
                         id=new_id,
                         kd=cell.kd,
                         ka=cell.ka,
                         R=new_radii,
                         type=cell.type)
#reset the time in each cell
    new_cell.time = 0.0
    cell.time = 0.0
#link the two cells, you will need to implement something like that
# to know what force calculate
    new_cell.daughter = cell.id
    cell.daughter = new_id
#return the new cell
    return new_cell
```

Code snippet, this is not a working example!

Parameters

$$L = 25 - 50$$
$$R = \textbf{gaussian}(1.0, 1e - 1)$$
$$k_{repulsive} = 5 \times 10^1$$
$$k_{attractive} = 10^0$$
$$k_d^0 = 10^{-1}$$
$$k_a = 10^{-3} - 10^{-1}$$
$$\alpha = 0.1$$
$$\Delta t = 10^{-2}$$

Task:
Explore how the number of cells changes when k_a it increases.

Extra:
Explore what happens when cell mutates and they can divide faster, i.e, 10x the "normal" cells