Statistics of Minicell Production

Humboldt Universität zu Berlin iGem-Team

October 22, 2021

Abstract

In this work we analyse properties of generated *Salmonella* Typhimurium minicells from the "Target Taxi" project of the iGEM team of the Humboldt Universität zu Berlin 2021. We evaluate experimental data in conjunction with already established literature to estimate the lifetime of these minicells after extraction. This lifetime is important because it focuses on the time frame in which the minicells are able to deliver anti cancer agents to infected tissue.

1 Methods

1.1 Production process of the minicells

Can be found in the iGEM wiki section of our web lab team. The python code can be found in 4

1.2 Size Distribution

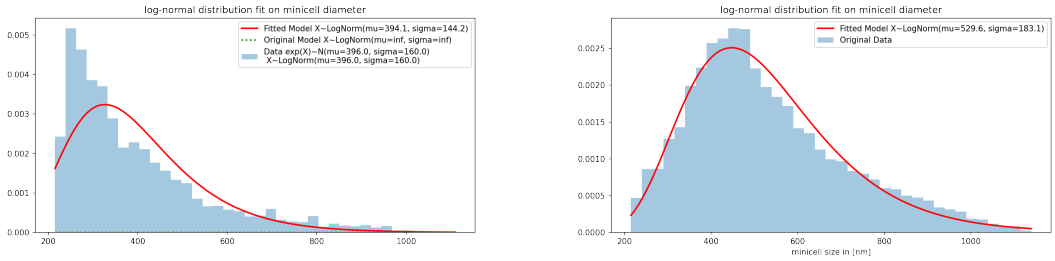
We consider the diameter d of a cell in a population as a random variable. We make the assumption that the probability density $P(d)$ of these measures in the produced cell population corresponds to a log normal distribution

$$P(d) = \frac{1}{d\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln(d) - \mu)^2}{2\sigma^2}\right). \tag{1}$$

We plot the distributions for filtered 1a and unfiltered 1b minicells. The probability of obtaining a minicell in asymmetric cell division can be calculated by an integral over the accepted interval $[a, b]$:

$$p(a \leq d \leq b) = \int_a^b P(d)\ \mathrm{d}d \tag{2}$$

The accepted interval for the diameter we set between 0.4 μm and 0.6 μm. The probability of finding a minicell in the filtered distribution was estimated to be 0.39. For the unfiltered distribution we obtain a probability of 0.28.



(a) (b)

Figure 1: Size distribution of (a) filtered and (b) unfiltered minicells, including logarithmic normal distribution fit

In the next step, we want to quantify how many functional minicells are produced in the process on statistical average. We model this with a binomial distribution, where n is the total number of cells after the asymmetric cell division and k is the number of functional minicells. The probability function is given by

$$B(n|p,N) = \binom{N}{n}(1-p)^{N-n}p^n. \tag{3}$$

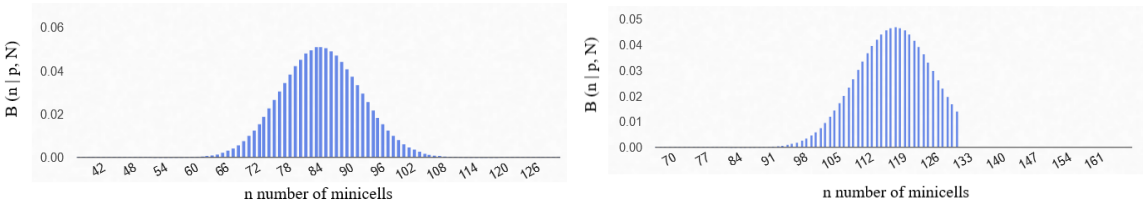
The expected value $\langle n \rangle$ of produced minicells increases linearly with population size:

$$\langle n \rangle = N \cdot p \tag{4}$$

For large N and $n \ll N$ the probability can be approximated by the Poisson distribution

$$B(n|p,N) \approx \frac{\langle n \rangle^n}{n!} \mathrm{e}^{-\langle n \rangle}. \tag{5}$$

The number of minicells produced as a statistical mean provides an information of the cell population required for production of the anti-cancer peptid in sufficient amounts at the site of action, i.e. at cancer cells that are aimed to be treated.



(a) (b)

Figure 2: binomial distribution for (a) unfiltered distribution and (b) filtered distribution with $N = 300$

1.3 Life span

We consider the metabolic rate β for biological systems, where thermal equilibrium is assumed. The metabolic rate has the physical dimension of a power, that is energy per time. Gillooly et al. [4][5] derived a general model based on the principles of biochemical kinetics and allometry, which describe the influence of absolute temperature T and organism mass m. They give the following equation for the mass-specific metabolic rate β_m, summing over all biochemical reactions of metabolism:

$$\beta_m = \frac{\beta}{m} = \left(\beta_0 \sum_{i=1}^{n} \exp\left(-\frac{E_i}{k_\mathrm{B}T}\right)\right) m^{-\theta} \tag{6}$$

Here E_i denotes the average activation energy for the respective rate-limiting enzyme-catalyzed biochemical reaction of metabolism, T the absolute temperature in Kelvin, k_B the Boltzmann constant, m the mass and θ a . β_0 is a constant to be determined of dimension $\mathrm{J} \cdot \sec \cdot \mathrm{kg}^{-\theta}$. In the following, these equations will be applied to minicells derived from *Salmonella* Typhimurium. We make the assumption that the geometry of the minicell can be approximated by a sphere surface with radius r respective diameter $d = 2r$. Furthermore, we assume a homogeneous density distribution, which results in the following equations for the minicell mass, where ρ_S denotes the density:

$$m = \frac{4}{3}\pi\rho_S r^3 = \frac{\pi}{6}\rho_S d^3 \tag{7}$$

The density of *Salmonella* $\rho_S = 1,005\ \mathrm{kg} \cdot \mathrm{m}^{-3}$ we approximate by the density of *Escherichia coli* [1], since *Salmonella* and *Escherichia coli* share similar cell morphology. The parameter θ is estimated to be $\frac{1}{4}$ for unicellular organisms. The lifespan of a biological system is inversely proportional to its mass-specific metabolic rate. Here, we standardize the prefactor to the Boltzmann factor. For notational reasons, we denote the whole prefactor by η.

$$\tau(m) = \underbrace{\left(\frac{\widetilde{\beta}_0}{k_\mathrm{B}T} \sum_{i=1}^{n} \exp\left(\frac{E_i}{k_\mathrm{B}T}\right)\right)^{-1}}_{\eta} m^{\theta} \tag{8}$$

However, it is not possible for us to track the lifespan of individual cells, so we use Proton Motive Force (short: PMF) to determine the average lifespan of a minicell population, where the accepted range of the diameter and the associated mass are $a \leq d \leq b$ and $m_a \leq m \leq m_b$. The average lifespan τ_0 determined by PMF can be obtained from the Gillooly equation by averaging over the accepted range. This allows us to calculate the constant factor η:

$$\tau_0 = \frac{\eta}{m_b - m_a} \int_{m_a}^{m_b} m^{\theta}\ \mathrm{d}m \tag{9}$$

$$\tau_0 = \frac{\eta}{1+\theta} \cdot \frac{m_b^{1+\theta} - m_a^{1+\theta}}{m_b - m_a} \tag{10}$$

$$\eta = \tau_0(1+\theta)\frac{m_b - m_a}{m_b^{1+\theta} - m_a^{1+\theta}} \tag{11}$$

For unicells ($\theta = \frac{1}{4}$) we obtain

$$\eta = \frac{5}{4}\tau_0 \sqrt[4]{\frac{3}{4\pi\varrho_0}} \cdot \frac{b^3 - a^3}{b^{\frac{15}{4}} - a^{\frac{15}{4}}}. \tag{12}$$

Through this we can establish a functional relationship between the lifespan τ and the diameter d of the minicell:

$$\tau(d) = \eta \sqrt[4]{\frac{\pi \rho_0 d^3}{6}} \tag{13}$$

1.3.1 PMF

We estimate the lifetime by fitting an exponential decay of the form (14) to experimentally measured data.

$$I = I_0 \exp\left(-\frac{t}{\tau_0}\right) \tag{14}$$

exponential fit

$$\ln(I) = -\lambda_0 t + \ln(I_0) \tag{15}$$

2 Simulation

We made sure that all of the simulated experiments were conform with the the measurements from the lab team as well as published literature. The python code for the simulation can be found below.

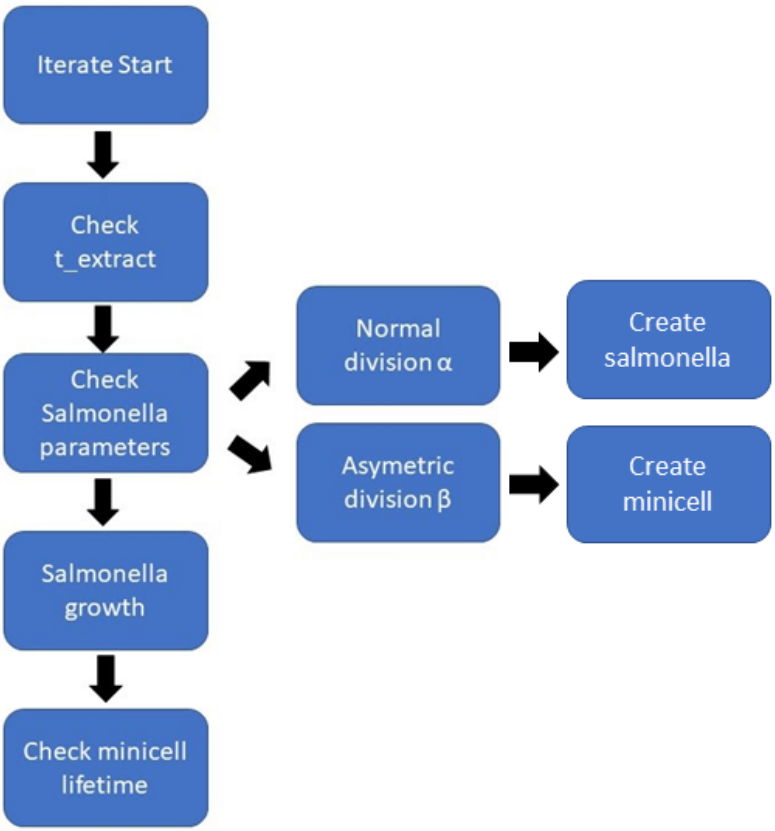


Figure 3: Simplified simulation architecture.

3 Results

This easy to extend model creates dividing *Salmonella* Typhimurium (*Salmonella*) cells that generate minicells who possess a size and lifetime. Thus, this model could be utilized in the future as a platform to answer many exciting questions in the context of minicell therapy.

For instance the question, whether less, but longer living and in size increased cells can destroy more or less tumor then many smaller ones. Further and most importantly, these insights will then be observable in the context of time.

By working closely with our wet lab team, we present a model that is based on already established literature and supported by additional experiments (such as proton motive force measurements (PMF)) to generate size and lifetime distributions of minicells.

Here, we compare the influence of two different size distributions of minicells on population and remaining life expectancy over time.

To simplify the simulation, we assume that the minicells generated during growth equal the final distribution after purification.

Next, we evaluate the influence of purification on the cell size. For this, we compare the size distribution of minicells, as they are present in the growing *Salmonella* culture, to the purified and filtered counterpart. For simplicity sake the generated data associated with the different size distributions will be called the filtered dataset and the unfiltered dataset. Filtration was done with a mesh size of 0.45μm.

While asymmetric division of *Salmonella* into minicell is the only cross interaction of *Salmonella* with the minicells is currently implemented, it was important to us that this feature can be implemented empathized on if needed.

3.1 minicell populations

First, we have a look at the number of minicells during the simulation. Population size differences during the stationary phase is due to the statistical nature of our model, although the influence of minicell size on *Salmonella* growth and therefore the minicell population after purification is really difficult to quantify. Still, we observe that the set of unfiltered minicells survives longer than the unfiltered one.

However, the behaviour of the population can be misleading when thinking about size and life expectancy as the next subchapters will show.

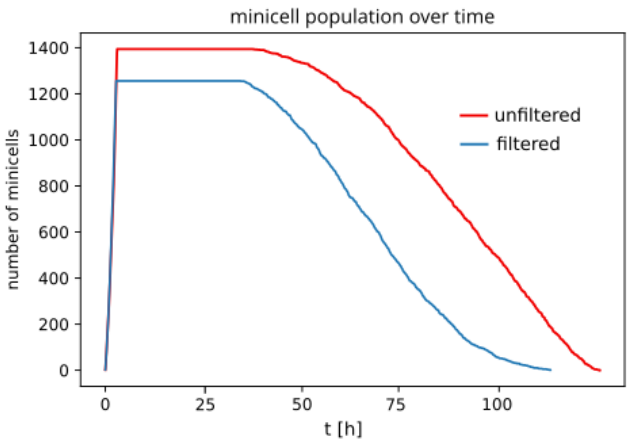


Figure 4: minicell population overtime

3.2 size heat map

Here we can observe that due to a minimum minicell lifetime of 37 h (minimal cell size = 200 nm), the minicell size stays stable for the first third of the runtime. After this time period the stability is lost. Thus, filtered minicells slowly degrade while small (<400 nm) and large (>600 nm) minicells dominate in later time periods.

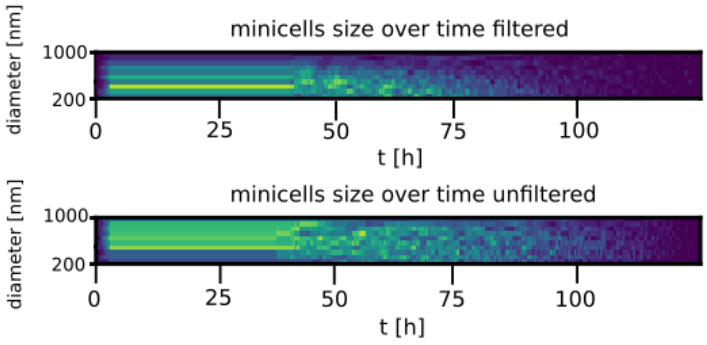


Figure 5: Minicell size distributions during the simulation for each timepoint. Y-axis is binned into 10 nm groups. The brighter the image is the higher the amount of minicells with a diameter within the range of the bin.

Strikingly, we observe a huge amount of abnormal large minicells (600 nm and more) in the later time stages for the unfiltered dataset.

In other words, the filtered minicells degrade in a specific pattern which results in increased amounts of small (late birth) and large (stable ones) minicells surviving at late stages. In con-

trast, the unfiltered minicells degrade continuously.

3.3 remaining lifetime heat map

Looking into the age distribution, the observation from the size simulation gets strengthened. This can be traced back to the approximately linear relation for minicell sizes between 100 - 200 nm diameter and the age distribution.

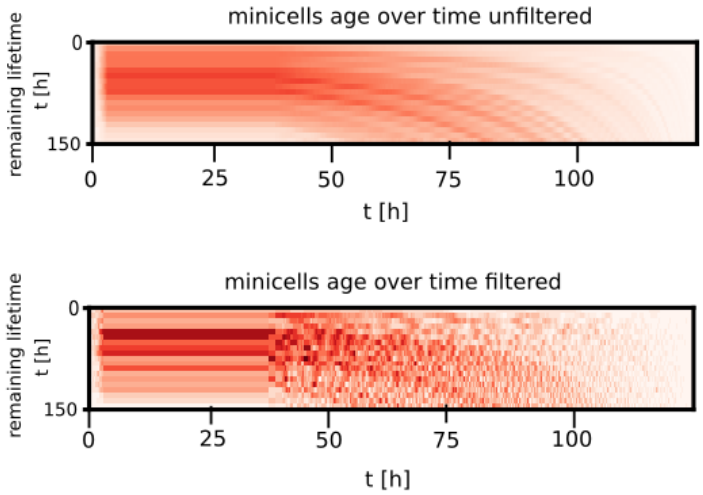


Figure 6: minicell population overtime

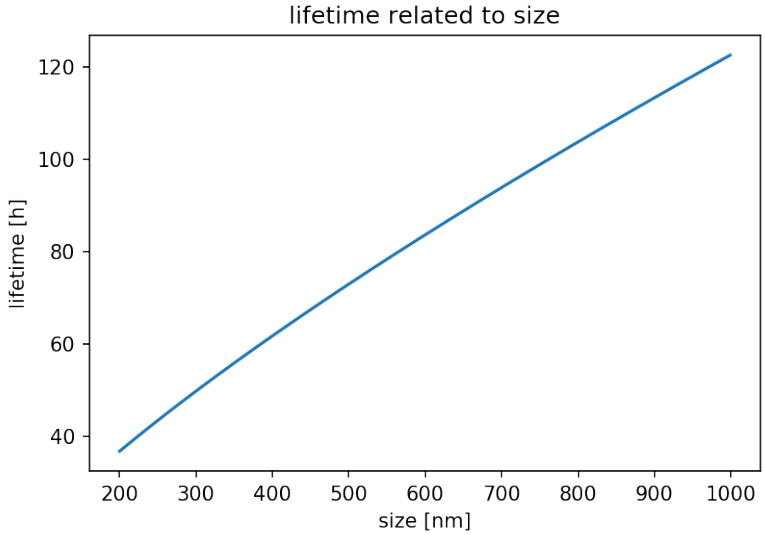


Figure 7: Fig. 5 Lifetime - size relationship for the allowed range of 200 to 1000 nm minicell diameter.

3.4 lifetime expectancy

When comparing the mean remaining lifetime of minicells for both datasets (filtered and unfiltered), the overall lifetime expectancy is higher in the unfiltered one, but draws after 110 hours. This allows us to observe that filtration has no influence on how long the average population survives after purification, although it does influence how many minicells are present in early and mid time periods.

The lifetime of minicells are the same for both datasets at the end of the simulation, indicating that the size distribution has no influence on total survival duration of total minicells. On the other hand, we see that size distribution has an influence on the amount of minicells that is present up to 80 hours.

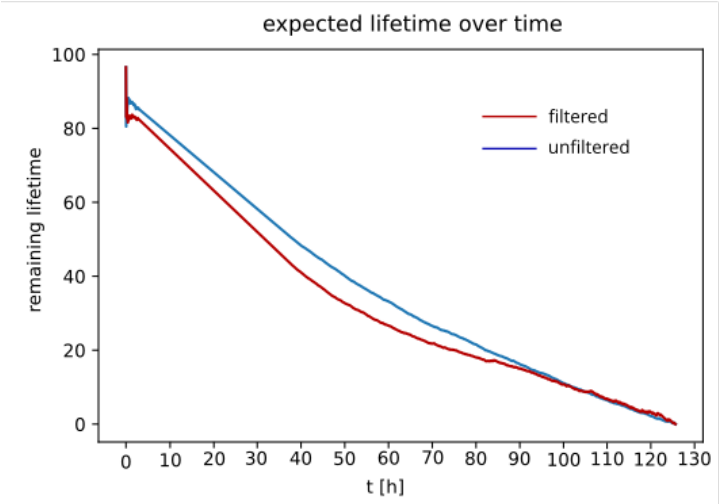


Figure 8: Remaining lifetime distribution for every minute. Bin size in y-axis is 1h. The brighter the image the higher the amount of minicells with a remaining life expectancy within the range of the bin.

3.5 theoretical background

Briefly we base our model mainly the findings of Gillooly et al [4,5], who allows us to directly link the size of the minicell to its lifetime. Furthermore we use other literature sources to choose the parameters for division frequency of modella [2] and death rate [3]. Additionally we approximate the density of salmonella to be the same of Eugela Coli [1].

3.6 possible further development

We see great potential in Model P to act as a platform to answer arising questions in the future, especially when combining expression efficiency, purification, drug delivery and application optimization for bacterial minicells.

Explicit idea for further development:

Approach: Use experimental measure of expression capacity and directly like it to size via surface area

Potential Insight: allows for model of protein density on minicell surface and quantification of incubation time of drug agents.

4 Python Code

```
#!/usr/bin/env python
# coding: utf-8

# # Simulation

# In[13]:


import pprint as pp
import os
import matplotlib.pylab as plt
import random as rd
import numpy as np
import matplotlib.pyplot as plt
import scipy.ndimage as ndimage
import math


# ### Evaluation with salmonella as number

# In[14]:


#Calculate Values

c = 0.03432 * 60 #  to per min measured experimentally
#c = 0.015 / 360 # old measure
roh  = 1105 # in kg / m^3 Literature
r_min = 75*10**-9 # in m
r_max = 500*10**-9 # in m

eta = 1/c * 5/3 * roh * ( r_max**3 - r_min**3 ) * ( r_max**(15/4) - r_min**(15/4) )**

delta = ( math.pi * 1/6 * roh ) ** ( 1/4 )

def get_tau(diameter): # return in minuite steps
    d = diameter
    return  eta * delta * d ** (3/4)

# time
t_max = 150 * 60  # Simulation will run for 150 h
t_extract = 180   # culture is done for 3 h according to Lab protocoll

# initial values and rates:
alpha = 0.01 # Literature, salmonella pop. grows with 1% per min [1%-10%]
beta = alpha*1/2 # Assumption from counting from the experiment "time series"
gamma = 0.0086 /1000 # Measured from time_lapse recording with delta t = 15min
```

```
deathrate= 0.0045 # Assumption, salmonella dye with 0.45% per timestep
n_salmonella = 1000 # Assumption we start with 1mio salmonella cells



# Print a sainity check for a few sizes
l = [200, 300, 400, 500, 600]
for d in l:
    print('diameter',d,'nm\n','\u03C4: ', round(get_tau(d*10**-9)/60),'h\n')


# In[15]:


# Define Minicells and Salmonella
class Minicell:
      def __init__(self, size, age):
        self.age = age
        self.size = size

class Salmonella:
      def __init__(self, _size, _age):
        self.age = _age
        self.size = _size

# Define functions

def get_size():
    size = 0

    while size < 200 or size >1000: # keep size between 200 and 1000

        size = (np.random.lognormal(0.3941, 0.1442) -1 )*1000 # Use this for filtered
        #size = (np.random.lognormal(0.5296,0.1831) -1 ) * 1000 # Use this for unfilt

    return round(size)

def salmonella_growth(_size):
    return _size * gamma/1000 * ( 5000 -_size ) +_size

def get_salmonella_size():    # create salmonella that is between 1 and 5 um
    x  = rd.randrange(1000,5000)
    #print(x)
    return x


# In[16]:
```

```
#### import pandas as pd

# Utility definitions
phase = 'growth'
mini_plot = []
parent_plot = []

mini_list = []
mini_series = []
salmonella_list = []
salmonella_series = []


# Add starting salmonellas
for i in range(n_salmonella):
    salmonella_list.append(Salmonella(get_salmonella_size(), 0))


# Let s Iterate
for t in range(t_max):


    # Check if keep simulation running is a good idea
    if t > t_extract and mini_list == []: #only run as long as minicells remain
        break

    # Define extraction
    if t >= t_extract:
        phase = 'extraction'

    spontainious_kill = []

    # salmonella grow and create minicells and other salmonella in growth phase:
    if phase == 'growth':

        for salmo in salmonella_list:

            # salmonella divides into salmonella

            # Only those above 2 um length may divide into normal cells
            if salmo.size > 2000:

                # Probability of division beta per timestep
                if rd.random() < alpha:

                    # Create new salmonella
                    salmonella_list.append(Salmonella(salmo.size/2, t))

                    # update old salmonella
```

```
                    salmo.size = ( salmo.size / 2 )
                    salmo.age = t

                # salmonella divides into minicell
                # only that are 2 um or bigger
                if rd.random() < beta:
                    mini_size = round(get_size())
                    tau = get_tau(mini_size*10**-9)
                    mini_list.append(Minicell(mini_size,t+tau))

                    # Salmonella cell does not refresh
                    salmo.size = salmo.size - mini_size

                #Spontainious death
                if rd.random() < deathrate:
                    spontainious_kill.append(salmo)


                # let salmonella grow:
                for salmonella in salmonella_list:
                    if salmonella.size <= 5000:
                        salmonella.size = salmonella_growth(salmonella.size)

    # Execute spontanious death
    for i in spontainious_kill:
        salmonella_list.remove(i)

    # kill minicell, dependet of age:
    if mini_list != []: # there have to be minicells at the timepoint
        kill_list = []
        for i in range(len(mini_list)): # check every minicell that is currently aliv
            lifetime = mini_list[i].age
            #print(age_of_minicell)
            if t>= lifetime:
                kill_list.append(mini_list[i]) # add minicell object to list to be re

        #print(kill_list)
        for i in kill_list:
            mini_list.remove(i)
    #print('Cycle Complete')

    # Decide what variables to extract and plot
    mini_plot.append(len(mini_list)) # minicell count
    parent_plot.append(len(salmonella_list)) # salmonella count

    mini_series.append(mini_list.copy()) # all data for each timepoint
    salmonella_series.append(salmonella_list.copy())

print('SIMULATION_COMPLETE')
```

```
### Lets Add salmonella as objects that split and grow on their own
#

## Plot size and age distributios at the end of simulation

### Plot populations

# In[17]:


# -- plot population count of minicells and samonella --

plt.figure(dpi = 100)
plt.title('minicells')
plt.ylabel('time in min')
plt.plot(mini_plot)
plt.ylabel('number of minicells')
plt.savefig('graphics/minicell population over time.pdf')

plt.show()

plt.figure(dpi = 100)
plt.plot(parent_plot[:t_extract + 20])
plt.title('Salmonella in growth phase')
plt.xlabel('time in min')
plt.ylabel('number of salmonella')
plt.savefig('graphics/salmonella population over time.pdf')
plt.show()


# In[18]:


# -- plot size distribution for selected timepoints --


size = []
age = []
#print(mini_series[2000])
for i in mini_series[2800]:
    size.append(i.size)
    age.append(i.age)

plt.figure(dpi  = 100)
plt.violinplot(size)
plt.title('Size in silico minicells at 2800 min')
plt.ylabel('diameter in mm')
```

```
plt.xticks([])
plt.savefig('graphics/Size in silico minicells at 2800 min.pdf')
plt.show()

plt.figure(dpi = 100)
plt.violinplot(age)
plt.title('Death at t in silico minicells at 2800 min')
plt.ylabel('t in model')
plt.xticks([])
plt.savefig('graphics/Death at t in silico minicells at 2800 min.pdf')
plt.show()

for i in mini_series[4500]:
    size.append(i.size)
    age.append(i.age)

plt.figure(dpi = 100)
plt.violinplot(size)
plt.title('Size in silico minicells at 4500 min' )
plt.ylabel('diameter in nm')
plt.xticks([])
plt.savefig('graphics/Size in silico minicells at 4500 min.pdf')
plt.show()

plt.figure(dpi = 100)
plt.violinplot(age)
plt.xticks([])

plt.title('Death at t in silico minicells at 4500 min')
plt.savefig('graphics/Death at t in silico minicells at 4500 min.pdf')
plt.ylabel('t in model')
plt.show()

for i in mini_series[7000]:
    size.append(i.size)
    age.append(i.age)

plt.figure(dpi = 100)
plt.violinplot(size)
plt.title('Size in silico minicells at 7000 min' )
plt.ylabel('diameter in nm')
plt.xticks([])
plt.savefig('graphics/Size in silico minicells at 7000 min.pdf')
plt.show()

plt.figure(dpi = 100)
plt.violinplot(age)
plt.xticks([])
```

```
plt.title('Death at t in silico minicells at 7000 min')
plt.savefig('graphics/Death at t in silico minicells at 7000 min.pdf')
plt.ylabel('t in model')
plt.show()


#

# In[19]:


# generate a heatmap to observe distribution change over time

def generate_heatmap(mini_series,y):

        picture_size = []
        picture_age = []
        mean_size = []
        mean_age = []

        T = 0
        for t in mini_series:
            size_t = []
            age_t = []

            for cell in t:
                size_t.append(cell.size)
                age_t.append((cell.age)-T)

            mean_size.append(np.mean(size_t))
            mean_age.append(np.mean(age_t))

            # adjust so heat can be seen:
            size = np.histogram(size_t, bins = y)
            age = np.histogram(age_t, bins = round(len(mini_series)/60))
            picture_size.append(size[0])
            picture_age.append(age[0])

            T=T+1
        return picture_size, picture_age, mean_size, mean_age



# In[20]:


# calculate heatmaps
pic_size, pic_age, mean_size, mean_age = generate_heatmap(mini_series,80)
salmo_size, salmo_age, mean_salmo_size, mean_salmo_age = generate_heatmap(salmonella
```

```
#print(len(salmonella_series))


# In[21]:


def plot(pic_size, pic_age, title):

    angle = 90 # in degrees

    rotated_pic_size = ndimage.rotate(pic_size, angle, reshape=True)

    plt.figure(dpi = 200)
    plt.imshow(rotated_pic_size)#, interpolation= 'nearest')
    plt.xlim(0,len(pic_size))
    plt.ylabel('diameter in nm')
    plt.title(title + ' Size over Time')

    #plt.yticks(ticks = ['200','450','1000'], labels = [0,400,800])
    plt.savefig('graphics/' + title + ' Size over Time.pdf')
    plt.show()

    plt.figure(dpi = 200)
    plt.xlim(0,len(pic_age))
    plt.title(title+ ' Age over Time')
    rotated_pic_age = ndimage.rotate(pic_age, angle, reshape=True)
    plt.imshow(rotated_pic_age, cmap = 'Reds')#, interpolation= 'nearest')
    plt.xlabel('time of simulation')
    plt.ylabel('expected lifetime ')
    plt.savefig('graphics/' + title +' Age over Time.pdf')

    plt.show()

# Show heatmaps
plot(pic_size, pic_age, 'Minicells')

plot(salmo_size[:500],salmo_age[:500], 'Salmonella')


# In[22]:


# plot mean of heatmaps

def plot_mean(mean_size, mean_age, title):
    plt.figure(dpi  = 150)
    plt.plot(mean_size)
    plt.title('mean size of '+ title)
    plt.xlabel('t in min')
```

```
    plt.ylabel('size in nm')
    plt.savefig('graphics/mean_size_of_' + title + '.pdf')
    plt.show()

    plt.figure(dpi  = 150)
    plt.title('mean_age_of_' + title)
    plt.xlabel('t_in_min')
    plt.ylabel('timepoint_of_death_for_' + title)
    plt.plot(mean_age)
    plt.savefig('graphics/timepoint_of_death_for_'+title+'.pdf')

    plt.show()

plot_mean(mean_size, mean_age, 'Minicells')
plot_mean(mean_salmo_size[:t_extract], mean_salmo_age, 'Salmonella')


# In[23]:


# plot lifetime size relationship

l = []
for i in range(200,1000):
    l.append(get_tau(i*10**-9)/60)

plt.figure(dpi = 150)
plt.plot(l)
plt.title('Tau_related_to_size')
plt.ylabel('tau_in_h')
plt.xlabel('size_in_nm')
plt.savefig('graphics/Tau_related_to_size.pdf')
plt.show()


# In[24]:


# sainity check salmonella grwoth

l = [1000,]

def salmonella_growth(_size):
    return _size * gamma * (5000-_size) +_size

for i in range(0,100):
    l.append(salmonella_growth(l[-1]))
print(l)
```

```
plt.figure(dpi = 150)
plt.plot(l)
plt.xlabel('time in min')
plt.ylabel('size in nm')
plt.title('single salmonella growth for 100 min.pdf')
plt.savefig('graphics/example Salmonella grwoth for 100min')
plt.show()


# In[ ]:




# In[ ]:
```

5 References

[1] S. Y. Lee, High cell-density culture of escherichia coli, TIBTECH, 14 (1996).

[2] D. B. Lowrie and V. R. Aber and M. E. W. Carrol, Division and Death Rates of Salmonella typhimurium Inside Macrophages: Use of Penicillin as a Probe, Journal of General Microbiology, 110/2 (1997)

[3] D. S. Chandler and J. A. Craven, Relationship of Soil Moisture to Survival of Escherichia coli and Salmonella typhimurium in Soils, Crop and Pasture Science, 31 (1980)

[4] Gillooly et al., Explaining differences in the lifespan and replicative capacity of cells: a general model and comparative analysis of vertebrates, The Royal Society, 297 (2012)

[5] Jamie Gillooly and V. m. Savage and James Brown, Effects of Size and Temperature on Metabolic Rate, sciencemag, 293 (2001)