Biophysical Journal, Volume *116*

Supplemental Information

Quantitative Analysis of the Correlation between Cell Size and Cellular Uptake of Particles

Jawahar Khetan, Md Shahinuzzaman, Sutapa Barua, and Dipak Barua

Quantitative analysis of the correlation between cell size and cellular uptake of particles

Jawahar Khetan[1], Md Shahinuzzaman[1], Sutapa Barua[1], and Dipak Barua[1,*]

[1]Department of Chemical and Biochemical Engineering, Missouri University of Science and Technology, Rolla, Missouri, USA
[*]Correspondence: baruad@mst.edu

DERIVATION OF THE REACTION-DIFFUSION MODEL

In the model, particles are reversibly captured by a cell-surface transporter. The transporter represents a generic molecule accounting for all different endocytic structures in the cell plasma membrane. Each transporter can handle one particle at a time. A particle captured by a transporter may dissociate and return to the solution, or it may be taken inside the cell through endocytosis. These steps are described by the following reaction scheme:

$$\text{Particle (solution)} + \text{Transporter} \underset{k_r}{\overset{k_f}{\rightleftharpoons}} \text{Complex} \xrightarrow{k_1} \text{Transporter} + \text{Particle (internalized)}$$

In the above scheme, the three reactions are associated with the following three rate constants: k_f is associated with the forward reaction that leads to the formation of the particle-transporter complex, k_r is associated with the reverse reaction that leads to dissociation of the complex, and k_1 is associated with the reaction that leads to particle endocytosis and regeneration of the transporter. The constant K_m is the Michaelis-Menten constant and is given by Eq. 1:

$$K_m = \frac{k_r + k_1}{k_f} \tag{1}$$

At steady-state condition, the flux of nanoparticles across the cell membrane can be described by the Michaelis-Menten rate law:

$$J = J_m \frac{C_0}{K_m + C_0} \tag{2}$$

Here, C_0 represents nanoparticle concentration at the solution-cell membrane interface, and $J_m = k_1 n$ represents maximum flux when there are n transporter molecules per unit area of the cell membrane.

For convenience, we rewrite Eq. 2 in dimensionless form:

$$J^* = \frac{C_0^*}{1 + C_0^*} \tag{3}$$

where $C_0^* = \frac{C_0}{K_m}$ and $J^* = \frac{J}{J_m}$. Further, we consider a spherical cell of radius r_0 and define dimensionless distance $r^* = \frac{r}{r_0}$ such that $r^* = 1$ at the cell surface.

The steady state concentration profile around a cell can be given by:

$$\nabla \cdot (D \nabla C^*) = 0 \tag{4}$$

where $C^* = \frac{C}{K_m}$ is the dimensionless nanoparticle concentration at $r^* > 1$. For spherical coordinates Eq. 4 becomes:

$$\frac{d}{dr^*}\left(r^{*2}\frac{dC^*}{dr^*}\right) = 0 \tag{5}$$

where r^* is the dimensionless radius $r^* = \frac{r}{r_0}$. Due to the symmetry of the spherical geometry, we assume no gradient in C^* in the θ and ϕ directions. Applying the two boundary conditions for the system:

$$C^* = C_b^* \;\; at \;\; r^* \to \infty \tag{6}$$

$$C^* = C_0^* \;\; at \;\; r^* = 1 \tag{7}$$

where $C_b^* = C_b/K_m$ is the dimensionless bulk nanoparticle concentration, we get the following solution:

$$\frac{C_b^* - C^*}{C_b^* - C_0^*} = \frac{1}{r^*} \tag{8}$$

The particle flux is given Fick's first law:

$$J = -D\nabla C \tag{9}$$

Thus,

$$J = -D\frac{\partial C}{\partial r}\Big|_{r=r_0} = \frac{D}{r_0}(C_b - C_0) \tag{10}$$

which can be brought back to the dimensionless form:

$$J^* = \frac{DK_m}{J_m r_0}(C_b^* - C_0^*) = \Psi(C_b^* - C_0^*) \tag{11}$$

where,

$$\Psi = \frac{DK_m}{J_m r_0} \tag{12}$$

From Eq.11 we get the total particle uptake rate by multiplying flux with cell surface area:

$$\dot{m} = 4\pi r_0^2 J_m \Psi(C_b^* - C_0^*) = 4\pi r_0 DK_m(C_b^* - C_0^*) = k(C_b^* - C_0^*), \tag{13}$$

where $k = 4\pi r_0 DK_m$.

Mass conservation requires that the two fluxes in Eq. 3 and Eq. 11 be equal. Thus by equating the two, we obtain the following quadratic equation:

$$C_0^{*2} + \left(\frac{1}{\Psi} + 1 - C_b^*\right)C_0^* - C_b^* = 0 \tag{14}$$

Solving for the nanoparticle concentration at the cell boundary we get:

$$C_0^* = -\frac{1}{2}\left(\frac{1}{\Psi} + 1 - C_b^*\right) + \frac{1}{2}\sqrt{\left(\frac{1}{\Psi} + 1 - C_b^*\right)^2 + 4C_b^*} \tag{15}$$

We then substitute C_0^* from Eq.15 in Eq.13 to get the total particle uptake rate by the entire cell,

$$\boldsymbol{\dot{m} = k\left(C_b^* + \frac{1}{2}\left(\frac{1}{\Psi} + 1 - C_b^*\right) - \frac{1}{2}\sqrt{\left(\frac{1}{\Psi} + 1 - C_b^*\right)^2 + 4C_b^*}\right)} \tag{16}$$

When considering variation in mean transporter density, $\tilde{n}$, with cell size, we substitute the following equation:

$$\frac{\tilde{n}}{\langle n\rangle} = \left(\frac{r_0}{\langle r_0\rangle}\right)^{\alpha} \tag{17}$$

into Ψ to get:

$$\Psi = \frac{DK_m}{J_m r_0} = \frac{DK_m}{k_1 \tilde{n} r_0} = \frac{DK_m}{k_1\langle n\rangle\left(\frac{r_0}{\langle r_0\rangle}\right)^{\alpha} r_0} \tag{18}$$

PYTHON CODE IMPLEMENTING THE REACTION-DIFFUSION MODEL

Use the following Python code to create a Python file, such as model.py. Execute the Python file, which will generate Fig. 5B of the paper.

```
#!/usr/bin/python
import matplotlib
matplotlib.use("TkAgg")
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import matplotlib.ticker as mtick
import numpy as np
import random
import math

from pylab import genfromtxt;
font = {'family' : 'serif',
        'weight' : 'normal',
        'size'   : 20}

matplotlib.rc('font', **font)
plt.rc('axes', labelsize=22)

#mat0 = genfromtxt("L_molecule_avg_dist7.dat");
#mat1 = genfromtxt("L_molecule_avg_dist8.dat");
#mat2 = genfromtxt("L_molecule_avg_dist9.dat");



Fs = 10000
```

```
f = 1

sample = Fs


a = 0.05 # particle radius in micron

w = 10.0 # ug/mL solution; nanoparticle solution on weight-basis

spg = 1.06 # Polystyrene (nanoparticle material) specific gravity.

m_particle = (4.0/3)*(np.pi)*((a/10000)**3)*1.00*(10**6) # Mass of a
nanoparticle in microggram

Cb = (w/(10**(12)))/m_particle # Bulk particle concentration; number
of particles per um^3 of the bulk solution



r0 = 10.0 # Mean cell size (radius)

mur = np.log(r0) # Mean cell size (radius) in log scale

sigmar = 0.5 # Standard deviation for cell size distribution



KB = 1.38064852e-23 # Boltzmann constant

T = 298.15          # Temperature

nu = 1e-3           # Water viscosity, Pa.s


D = (KB * T / (6 * np.pi * nu * a * 1e-6))*1e12  # particle
diffusivity micron^2/s



kf = 0.1 # Goldstein, intrinsic on rate

nmu0 = 0.119 # Number of coated pits per unit surface area (150 in a
cell of 10 micron radius)

nstd = 0.4 # standard deviation - cell -to cell variability in surface
density of pit

kr = 0.1 # Mean residence time of a particle in a pit is 10 second
```

```
k1 = 0.02  # Mean lifetime of a pit is 50 seconds; this is inverse of
the pit lifetime


#npit = np.exp(np.random.normal(np.log(nmu), nstd, sample))


#y = 4 * np.pi * r * D * r * km / (D + r * km)
#y = 4 * np.pi * r * r * km




def evaluate_f2(*vartuple):
    kf = vartuple[0]
    kr = vartuple[1]
    k1 = vartuple[2]
    n =  vartuple[3]
    r =  vartuple[4]
    Cb = vartuple[5]
    D = vartuple[6]

    Jm = k1 * n

    Km = (kr + k1)/kf
    Cb = Cb/Km # dimensionless bulk concentration

    Psi = D*Km / (Jm * r)
    Zet = ((1 / Psi) + 1 - Cb)
    C0 = -Zet/2 + (math.sqrt(Zet*Zet + 4*Cb))/2

    k = 4 * np.pi * r * D * Km

    m = k * (Cb - C0)
```

```
    return(m);



n_elem = 6;
#z1 = [[0 for x in range(sample)] for y in range(n_elem)]

#z = [[0 for xx in range (sample)] for yy in range(5)]

y = np.empty([n_elem, sample])

factor = np.array([1, 0.1, 0.03, 0.01, 0.003, 0.001])
colors = np.array(['k', 'orange', 'g', 'r', 'c', 'b'])
plots  = np.empty([n_elem])
alpha = np.array([1, 0.5, 0, -0.5, -1, -2])

marker_size = 0.5

for j in range(n_elem):
    csize = np.random.normal(mur, sigmar, sample)
    x = np.exp(csize)
    for i in range(sample):
        r = x[i]
        nmu = nmu0*((r/r0)**(alpha[2]))
        ntot = np.exp(np.random.normal(np.log(nmu), nstd))
        val1 = evaluate_f2(kf, kr, k1, ntot, r, Cb, factor[j]*D)


        Dmean = D
        rmean = r0
        kmeanCb = 4*(np.pi)*rmean*Dmean*Cb # k*(Cb*) =
4*(pi)*r_0D*Km*(Cb/Km)  = 4*(pi)*r_0*D*Cb
```

```
        y[j,i] = val1/kmeanCb # Normalized by k*Cb*


    plt.scatter(x, y[j,:], color= colors[j], marker= ".", s =
marker_size)
    #plt.plot(x, y[j,:], color= colors[j])


# x-axis label
plt.xlabel("Cell radius ($\mu$m)")
# frequency label
#plt.ylabel("Uptake ($m/\~{k}C_b$)")
plt.ylabel("Uptake")
# plot title
#plt.title('My scatter plot!')
# showing legend
leg = plt.legend()
leg.get_frame().set_alpha(0.0)
#plt.xscale("log");
plt.xlim(1,20);
plt.ylim(0,0.012);


plt.tick_params(direction='in', length=6, width=2, colors='k',
               grid_color='r', grid_alpha=0.5, pad=10)


plt.gca().yaxis.set_major_formatter(mtick.FormatStrFormatter('%0.1g'))
plt.tight_layout()
plt.savefig('Fig5B.png', format='png', dpi=1500)
# function to show the plot
plt.show()
```