COBB2041: Cell and Systems Modeling 5/3/2021

Kinetic Modeling of Novel SARS-CoV-2 Yeast Agglutination Assay

Name: Josiah Kratz + Kamaru-Deen Lawal

Email: jkratz@andrew.cmu.edu + kal246@pitt.edu

Introduction

We are over a year into the COVID-19 pandemic and still have limited access to rapid point-of-need testing. As a result, the Bruchez lab at CMU has started development of a novel type of SARS-CoV-2 assay which utilizes yeast agglutination. In this assay, saliva from a potentially infected person will be added to a solution containing yeast cells which have been engineered to express high numbers of anti-SARS-CoV-2 nanobodies on their surfaces. As a result, if SARS-CoV-2 virions are present in saliva, the multivalent virions will link multiple yeast cells, resulting in the formation of large complexes which can be assayed spectrophotometrically.

Many types of agglutination assays have been used in the past to detect viruses and other multivalent analytes. Red blood cells, E. coli, and antibody-bound beads have all been used to create similar systems. However, using yeast for the agglutination assay is advantageous because it can be grown quickly and cheaply. In addition it can be transported easily. We are motivated to model the kinetics of this agglutination process in order to accelerate the development of this technology. Our goal is to (1) model agglutination as a function of the amount of virus present to determine the lowest concentration of virus that can be detected using this system and (2) investigate how the sensitivity range can be changed because the expected viral loads of infected patients is hard to estimate.

Related Work

There were two primary papers that we made use of to work on our project, (1) Whole-Cell Biosensor with Tunable Limit of Detection Enables Low- Cost Agglutination Assays for Medical Diagnostic Applications (Kyllis et. al) and (2) A Kinetic Model of the Agglutination Process (Dolgosheina et. al). Kyllis et. al helped to lay down the foundations for how biosensors can form the basis of easy-to-use diagnostic tests that can be readily deployed for point of care testing. The paper specifically develops a biosensing platform based on cell agglutination using an E-coli whole-cell biosensor surface-displaying nanobodies which bind selectively to target protein analytes. Dolgosheina et. al proposes a kinetic model of the aggregation process in a system consisting of a bivalent antibody and a polyvalent bacterial cell.

Methods

To model our yeast agglutination assay system, we have adapted the mathematical model utilized by Kylilis et al. in their E. coli agglutination system. The system can be described by two ordinary differential equations (ODEs). The time-dependent dynamics of normalized cell agglutination, $\tilde{N}$, is given by,

$$\frac{d\tilde{N}(t)}{dt} = C_{1n}a(t)[1 - a(t)](-\frac{1}{2}\tilde{N}^2(t))$$

where C_{1n} is the normalized agglutination rate constant and $a(t)$ is the time-dependent bound nanobody fraction. The normalized rate constant is given by $C_{1n} = C_1 N_0$, where N_0 is the concentration of yeast cells, and C_1 is the kinetic agglutination rate constant representing the interactions between cell clusters. A full derivation of this ODE is given by Kylilis et al., but the final result is also somewhat intuitive. For an agglutinate to form, it requires that a single cell or cell cluster must have a nanobodies already bound to virus, and then must come in contact with a cell or cell cluster with free nanobodies. As a result, the rate of agglutination must depend on both of these concentrations. The normalized number of cells or agglutinates with bound nanobodies is given by multiplying the bound nanobody fraction, $a(t)$, by the total number of cells and agglutinates, $\tilde{N}$. Similarly, the number of cells or agglutinates with unbound nanobodies is given by $[1 - a(t)]\tilde{N}$. The factor of $\frac{1}{2}$ that is also present in the rate equation comes out of the derivation, and is included to prevent double-counting when summing over all of the possible ways to construct agglutinates of different sizes. It is important to note that normalized cell agglutination, , represents the total concentration of agglutinating clusters and cells. As a result, at $t = 0$, $\tilde{N}(t) = 1$, and $\tilde{N}$ decreases over time as more and more cells become incorporated into agglutinating clusters. Consequently, the fraction of cells that are part of the agglutinate population is given by $1 - \tilde{N}$.

The fraction of nanobodies bound to analyte is also time-dependent, and is given by,

$$\frac{da(t)}{dt} = C_2 N_l [sa^2(t) - (1 + s + \alpha)a(t) + 1]$$

, where C_2i s the binding rate constant of the free virus to the nanobody, and N_l is the initial concentration of analyte in the sample. The constant is the ratio of total nanobody density to total analyte, and is given by $s = \frac{N_0 f}{N_l}$, where f is the number of nanobodies per cell. The constant α is the nondimensionalized equilibrium dissociation constant and is given by $\alpha = \frac{C_3}{C_2 N_l}$, where C_3 is the dissociation rate constant.

It is important to note that this model makes two assumptions. The first is that agglutinates diffuse at the same speed as single cells, and thus have the same probability of interacting and binding with each other as single cells do with other single cells. This assumption is robust but becomes increasingly inaccurate if extremely large agglutinates are formed. Second, even though SARS-CoV-2 virions have on average 24 spikes per virion and thus have multiple nanobody binding sites per virion (**Ke, Z., Oton, J., Qu, K. et al. Structures and distributions of SARS-CoV-2 spike proteins on intact virions. Nature 588, 498-502 (2020).**), it is assumed they can only bind two yeast cells concurrently. This constraint is simply due to geometry. A single yeast cell can be approximated as a sphere with a 6 m diameter (**”Physical Biology of the Cell”, Rob Phillips, Jane Kondev and Julie Theriot (2009)**.), while SARS-CoV-2 virions are on average spheres with a diameter of 91 nm. As shown by the schematic in Figure 1, because participating yeast cells orders of magnitude larger than virions, it is unlikely that more than two yeast cells could be bound by a single virion.

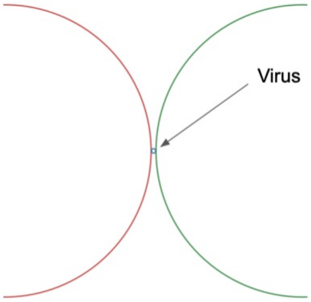

Figure 1. Schematic showing the relative size of two yeast cells (red and green circles) compared to a SARS-CoV-2 virion (labeled). Due to geometrical constraints, no more than two yeast cells are assumed to be able to bind a single virion.

To solve the system of ODEs that describe this system, the odeint function from SciPy was used. Because the time-dependent dynamics of normalized cell agglutination, $\frac{d\tilde{N}(t)}{dt}$, depends not only on $\tilde{N}$, but also on the fraction of bound nanobodies, $a(t)$, the two ODEs were solved simultaneously by odeint. A representative timecourse for both normalized agglutination and fraction of bound nanobody are shown in Figure 2 below. As can be seen from the figure, the bound nanobody fraction equilibrates within several minutes, while agglutination occurs on a timescale of hours.

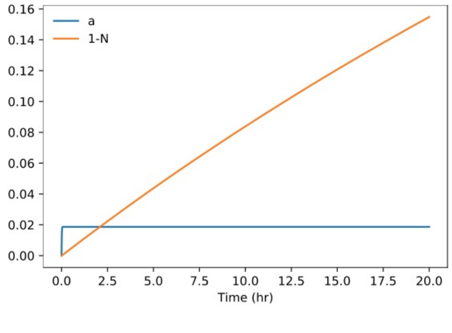

Figure 2. A representative timecourse of both agglutination (1-N) and the fraction of bound nanobody (a). Agglutination and bound nanobody fraction values were simulated for 20 hours for a yeast cell concentration of

$8.31 \cdot 10^8$ cells/mL and a virus concentration of 10^{-9} M. The bound nanobody fraction equilibrates much faster than agglutination.

To assess how agglutination and bound nanobody fraction were affected by viral load, agglutination profiles of $1 - \tilde{N}$ and a were simulated over a range of initial virus concentrations for a given timepoint and initial virus concentration. To confirm that this process was successfully implemented in python, a plot of both agglutination $(1 - \tilde{N})$ and bound nanobody density (a) at 18 hours as a function of initial analyte concentration from Kylilis et al. was reproduced successfully and is shown below in Figure 3.

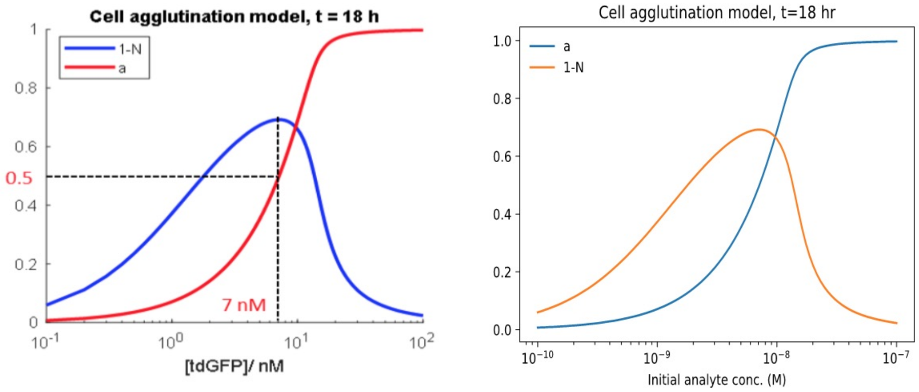

Figure 3. Successful implementation of model in Python. To validate that the model was successfully implemented in python, a plot of both agglutination $(1-\tilde{N})$ and bound nanobody density (a) from Kylilis et al. was reproduced using constants from their system.

With the implementation of the model in python validated, the parameters of the model were then modified to reflect the new system, and are given in Table 1. Because the specific nanobody that will be used for this assay was unknown, we assumed the use of the H11-D4 nanobody, which binds the Receptor Binding Domain (RBD) of the spike protein with an equilibrium dissociation constant, K_D, of 39 nM (**Huo, J., Le Bas, A., Ruza, R.R. et al. Neutralizing nanobodies bind SARS-CoV-2 spike RBD and block interaction with ACE2. Nat Struct Mol Biol 27, 846-854 (2020)**.). To obtain the binding and dissociation rate constants, C_2 and C_3, respectively, it was assumed that the nanobody-protein association rate constant was diffusion limited and that no long-range forces, such as electrostatics, played a large role in binding. Diffusion limited interactions are found in the range of 10^5 to 10^6 M^{-1} s^{-1} (**”Fundamental Aspects of Protein-Protein Association Kinetics” G. Schreiber, G. Haran, and H.-X. Zhou Chemical Reviews 2009 109 (3), 839-860**), thus $5 \cdot 10^5$ $M^{-1}s^{-1}$ was chosen for C_2. As a result, the dissociation rate constant, C_3, was set to $70.2h^{-1}$, because the equilibrium dissociation constant is given by the ratio of the dissociation and binding rate constants ($C_3 = K_D C_2$). The normalized agglutination rate constant, C_{1n}, was kept at $1h^{-1}$ because it has not been well characterized for this system. Lastly, the number of nanobodies displayed per cell was assumed to be 10^4, which is a standard amount of copies obtained when using yeast surface display technology (**Uchanski, T., Zgg, T., Yin, J. et al. An improved yeast surface display platform for the screening of nanobody immune libraries. Sci Rep 9, 382 (2019)**.)

Table 1. The parameters used for the yeast agglutination assay simulations, unless stated otherwise.

Parameter	Description	Value
C_{1n}	Normalized agglutination rate constant	1 h^{-1}
C_2	Binding rate constant of virus and nanobody	3600 x 5E+05 M^{-1} h^{-1}
K_D	Equilibrium dissociation constant	39E-09 M
C_3	Dissociation rate constant	70.2 h^{-1}
f	Number of nanobodies displayed per cell	1.00E+04

The shape of the resulting agglutination curve, which can be seen in the Figure 3 above, follows a general trend characteristic of most agglutination assays. The extent of agglutination is a function of the virus/nanobody ratio. At low ratios of virus/yeast, there is not enough virus present to link multiple yeast cells together to form agglutinates. At high ratios of virus/yeast, the virus is able to bind nearly all of the nanobodies, again preventing agglutination because there is a lack of unbound nanobodies able to bind virus already bound to another yeast cell. Maximum agglutination is achieved in between these two extreme ratios.

Results

To our knowledge, this assay will be the first of its kind in that it detects intact virions as opposed to viral products such as viral RNA or plaque-forming units, or immune responses such as measuring production of antibodies (**Ravi et al., ”Diagnostics for SARS-CoV-2 detection: A comprehensive review of the FDA-EUA COVID-19 testing landscape.”**). As a result, although the range of expected values is well characterized for other SARS-CoV-2 tests, there is little information on the expected range of concentrations of virus found in an infected person?s saliva. As a result, instead of attempting to determine the optimal yeast concentration to use for this assay, we focused on determined how the sensitivity range and minimum detectable concentration of the assay was affected by the system parameters. We focused our analysis on three parameters: yeast concentration, time of agglutination, and kinetic rate constant of agglutination.

The minimum detectable concentration for a given time and yeast concentration was defined as the virus concentration required to agglutinate 5% of the total yeast cells (shown in Figure 4). This value was chosen as such because it was the approximate percent of cells that needed to agglutinate for Kylilis et al. to be able to detect agglutinates after visual inspection. As a result, this is a slightly conservative estimate because the yeast used in our assay are fluorescent, and thus will be able to be detected at lower quantities, assuming that they can be successfully isolated from other cells that did not agglutinate. The sensitivity range was defined as the range of viral load concentrations between the minimum detectable concentration (when the fraction agglutinated is 0.05) and the value when agglutination is at a maximum (Figure 4).

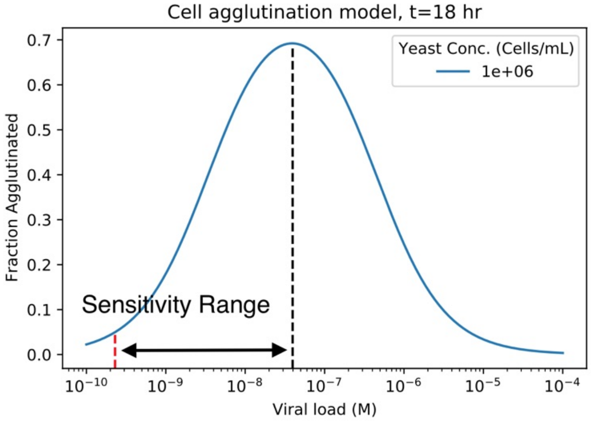

Figure 4. The sensitivity range is shown above. It was defined as the range of viral load concentrations between the minimum detectable concentration (when the fraction agglutinated is 0.05, red dashed line) and the value when agglutination is at a maximum (black dashed line).

We first investigated the effects of agglutination time on minimum detectable concentration. As can be seen below in Figure 5, as time progresses the viral load which yields the maximum agglutination remains unchanged. However, the minimum detectable virus concentration significantly decreases as more and more cells are incorporated into the agglutinate population. We simulated the minimum detectable concentration over time and found that in the first 10^{-15} hours the minimum concentration drops significantly and can decrease the sensitivity limit by an order of magnitude, before leveling off at approximately $10^{-10}M$ virions. Although the minimum detectable concentration does continue to decrease after the first 15 hours, it does so at a very slow rate.

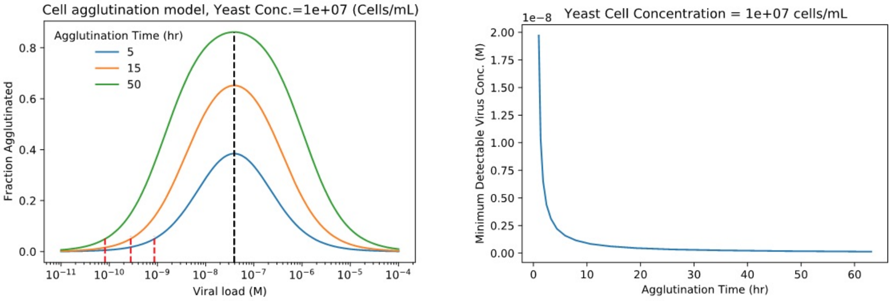

Figure 5. Left: Agglutination profiles are shown for three different time points. The minimum detectable concentration (red dashed line) decreases over time. Right: Minimum detectable concentration is plotted as a function of agglutination time. The minimum concentration drops significantly in the first 10 hours before leveling off. Simulations were run with a yeast concentration of 10^7 cells/mL.

The effect of changing the yeast concentration in the assay was investigated next. As can be seen in Figure 6 below, we found that the yeast concentration can have a dramatic impact on the sensitivity range and minimum detectable concentration. By simply changing the yeast concentration, the sensitivity range can be shifted several orders of magnitude. For example, with a yeast concentration of 10^{11} cells/mL, the sensitivity range is from approximately 10^{-8} to 10^{-6} M virions. However, when the yeast concentration is decreased to 10^9 cells/mL, the sensitivity range is shifted to approximately $10^{-9.5}$ to $10^{-7.5}$ M virions. Interestingly, as can be seen below, the minimum detectable virus concentration starts to level off at approximately 10^9 cells/mL, resulting in a minimum detectable concentration of 10^{-9} M virus. Further decreases in yeast concentration have little effect on the minimum detectable concentration after 15 hours of agglutination.

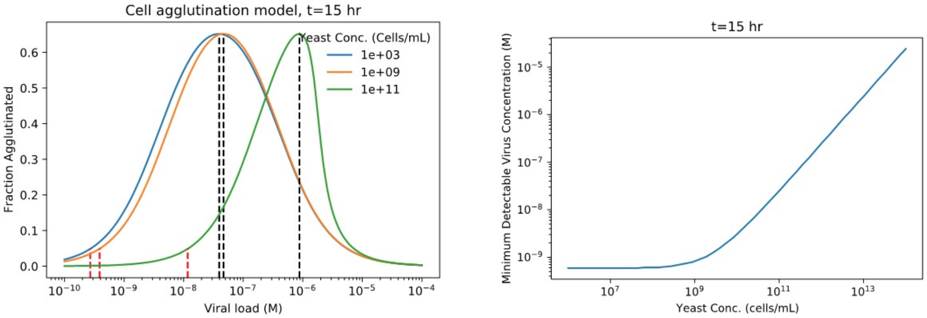

Figure 6. Left: Agglutination profiles are shown for varying yeast concentrations. Changing the yeast concentration can drastically change the assay's sensitivity range and minimum detectable concentration. Right: The minimum detectable virus concentration is plotted as a function of yeast concentration. Decreasing the yeast concentration results in a decrease in the minimum detectable concentration. Interestingly, at around 10^9 cells/mL the rate of decrease in the minimum detectable concentration significantly decreases.

The last parameter we investigated was the kinetic agglutination constant, C_{1n}. Although this parameter cannot be easily tuned experimentally, we were interested in its effect because it has not been well characterized for this system. For the previous simulations shown above, a default value of $1h^{-1}$ was used for C_{1n}, as this was the value used in the Kylilis et al. report which used a similar system. However, there is insufficient experimental data to properly estimate the kinetic constant of agglutination for both the Kylilis et al. system and for this system, and so we were interested in how sensitive the minimum detectable virus concentration and sensitivity range is to changes to this value. As can be seen in Figure 7 below, the agglutination constant has a large impact on the minimum detectable concentration, with the ability to decrease the minimum detectable concentration over several orders of magnitude. Additionally, the minimum detectable concentration decreases rapidly as the rate constant increases over the first 20 units. It is important to note however, that the sensitivity range does decrease as the agglutination constant increases, because the system can become fully agglutinated at a much faster rate.

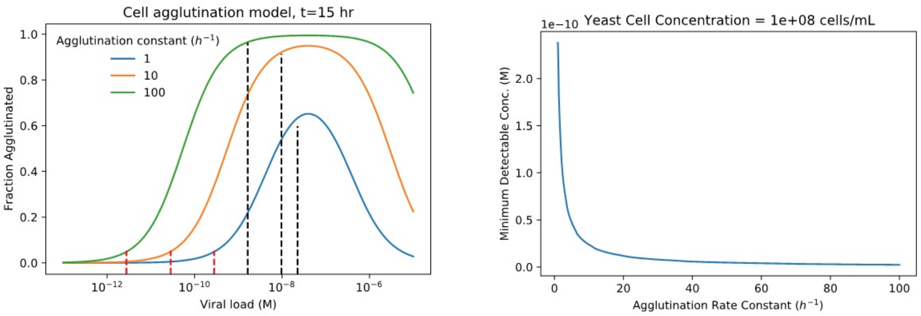

Figure 7. Left: Agglutination profiles are shown for different values of the normalized agglutination rate constant, C_{1n}, with a yeast concentration of 10^8 cells/mL. Increasing the rate constant can drastically decrease the minimum detectable virus concentration, and also shifts the sensitivity range. The upper bound of the sensitivity range was set as the virus concentration which results in 97% of the maximum agglutination value. Right: The minimum detectable concentration as a function of the normalized agglutination rate constant. The minimum detectable concentration decreases rapidly as the rate constant increases over the first 20 units.

Discussion

We modeled the kinetics of a novel SARS-CoV-2 assay which utilizes yeast agglutination. In an effort to better understand the assay, and hopefully accelerate development of the test, we investigated how three parameters impacted the agglutination. Initial time course simulations indicated that virions quickly reach a binding equilibrium with the displayed nanobodies, while agglutination takes significantly longer. Not surprisingly, we found that the time the system is allowed to agglutinate can significantly impact both the sensitivity range and minimum detectable virus concentration of the assay. This is because the yeast cells diffuse much slower than the virion, and the additional time allows for more productive collisions between agglutinating clusters and free cells. We found that in the first 10^{-15} hours of agglutination the minimum concentration drops significantly and can decrease the sensitivity limit by an order of magnitude, before leveling off at approximately 10^{-10} M virions.

Although the sensitivity limit can be decreased with time, the virus concentration which results in maximum agglutination does not change with time. As a result, although agglutination time is important in decreasing the sensitivity limit, it cannot significantly influence the sensitivity range. In contrast, changing the yeast concentration not only can influence the minimum detectable virus concentration, but it can also drastically shift the sensitivity range. Excitingly, this indicates that the assay's sensitivity range is extremely tunable simply by adjusting the yeast concentration, which is extremely easy to do experimentally. This is advantageous considering that the expected range of viral loads from infected persons is currently hard to estimate. Interestingly, we found that the sensitivity limit reaches a minimum at approximately 10^9 cells/mL, resulting in a minimum detectable concentration of 10^{-9} M virus, after 15 hours of agglutination. The minimum sensitivity limit occurs when both the yeast and virus concentrations are low. This is because the probability of having productive collisions is low due to the low concentrations, and thus under these conditions a significant amount of additional time is required for agglutinates to form.

The last parameter we investigated was the kinetic agglutination constant, C1n. Although this parameter cannot easily be tuned experimentally, we were interested in its effect because it has not been well characterized for this system. We found that increasing the agglutination constant could significantly decrease the minimum detectable virus concentration, allowing for the sensitivity limit to go below 10^{-11} M virions. However, it did also result in a shift of the upper bound of the sensitivity range because the system was able to reach near complete agglutination at lower virus/yeast ratios. This decrease in range is not expected to be a major issue however, because the assay is still sensitive over a range of two orders of magnitude. To increase the accuracy of the simulations in the future, the kinetic agglutination constant should be determined by fitting the model to experimental data. Additionally, it is important to note that in the Kylilis et al. workflow, the agglutinates were formed in the absence of any mixing. However, the kinetic agglutination constant could be increased significantly by continuously mixing the solution as it agglutinates. Importantly, this would not only decrease the agglutination time, but also decrease the sensitivity limit, both of which are advantageous properties of a clinical assay. It must be noted however that mixing at high speeds could potentially have other effects not accounted for in this model, such as breaking apart large agglutinates.

Conclusion

In conclusion longer agglutination times and decreased yeast concentrations lower the minimum detectable concentration of virus. Optimal results take place around 10 hours of agglutination and 10^9 yeast cells/mL. In addition, we noticed that increasing the agglutination rate constant significantly decreased the minimum detectable concentration. Further work could involve fitting a model to real world data, and testing the predictive power of the model on data. In addition, we could see how mixing the yeast assays impacts the results. One final area of future work could be attempting to model the distribution of the number of cells in each individual agglutinate. This is a particularly interesting problem because it would require a completely different set of methodologies to effectively determine the distribution of agglutinate size.

Contributions

The initial plan was for Josiah and Kamaru-deen to each write as much code as we could individually to ensure that we both had a solid understanding of the project at all given points. We started off by seeking to replicate the results in Kyllis et. al. Kamaru-deen managed to replicate some of the plots from the Kyllis et. al paper using the odeint package, as did Josiah. From there it became fairly obvious that we could leverage Josiah's background in the natural sciences to complete the remainder of the project. The code written to model and analyze the yeast agglutination system, which generated the plots shown in the results section, was completed by Josiah. Both Josiah and Kamaru-deen worked together to design and execute the presentation. With regard to typing up the final report, Josiah was responsible for writing the Methods, Results, and Discussion sections. Kamaru-deen was responsible for writing the Introduction, Related Work, and Conclusions section. In addition, Kamaru-deen was responsible for typesetting the document in LaTeX.

References

- Nicolas Kylilis, Pinpunya Riangrungroj, Hung-En Lai, Valencio Salema, Luis ngel Fernndez, Guy-Bart V. Stan, Paul S. Freemont, and Karen M. Polizzi ACS Sensors 2019 4 (2), 370-378 DOI: 10.1021/acssensors.8b01163

- Dolgosheina EB, Karulin AYu, Bobylev AV. A kinetic model of the agglutination process. Math Biosci. 1992 Apr;109(1):1-10. doi: 10.1016/0025-5564(92)90048-2. PMID: 1591446.

- Ke, Z., Oton, J., Qu, K. et al. Structures and distributions of SARS-CoV-2 spike proteins on intact virions. Nature 588, 498-502 (2020).

- ?Physical Biology of the Cell?, Rob Phillips, Jane Kondev and Julie Theriot (2009).

- Huo, J., Le Bas, A., Ruza, R.R. et al. Neutralizing nanobodies bind SARS-CoV-2 spike RBD and block interaction with ACE2. Nat Struct Mol Biol 27, 846-854 (2020).

- "Fundamental Aspects of Protein-Protein Association Kinetics? G. Schreiber, G. Haran, and H.-X. Zhou Chemical Reviews 2009 109 (3), 839-860

- Uchanski, T., Zgg, T., Yin, J. et al. An improved yeast surface display platform for the screening of nanobody immune libraries. Sci Rep 9, 382 (2019)

- Ravi et al., ?Diagnostics for SARS-CoV-2 detection: A comprehensive review of the FDA-EUA COVID-19 testing landscape.?

Code

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.integrate import odeint

#Normalized agglutination rate constant
C1_n = 1 #h^-1, C_1n = C1*N_0
#Binding rate constant between free virus and nanobody
C2 = 3600*5*10**5 #M^-1 h^-1, estimation
#Equilibrium Constant, K_D
K_D = 39*10**-9 #M, for H11-D4 nanobody
#Dissociation rate constant
C3 = K_D*C2 #h^-1
#Number of nanobodies per cell
f = 1*10**4
#Initial concentration of cells
N_0 = 1.38*10**-12
#Initial concentration of nanobody bound
a0 = 0
#Initial normalized unbound cells, N
N0 = 1

#packing the ODEs together so can be solved together by odeint
def timecourse(Species, t, C1_n, C2, N_l, s, alpha):
    a,N = Species #Tuple upacking of each species
    a_ode = bound_nanobody(a, t, C2, N_l, s, alpha)
    N_ode = norm_cell_agglutination(N, t, a, C1_n)
    dSpeciesdt = [a_ode,N_ode]
    return dSpeciesdt

def PlotSpecies(Species, Time):
    #Unpacking the Species Matrix
    a = Species[:,0]
    N = Species[:,1]
    N = 1-N
    #Plotting S and P as functions of time
    plt.plot(Time,a, label=’a’)
    plt.plot(Time,N, label=’1-N’)
    plt.legend()
    plt.title(’Timecourse of Agglutination’)
    plt.xlabel(’Time (hr)’)

def RunAndPlot(a0,N0,C1_n,C2,N_l,s,alpha,simulation_max,n_points):
    #create time array
    t = np.linspace(0,simulation_max,n_points)
    species_0 = [a0,N0]
    argtuple = (C1_n,C2,N_l,s,alpha)
    #solving the ODE
    species = odeint(timecourse, species_0, t, args=argtuple)
    PlotSpecies(species,t)

simulation_max = 20 #in hrs
n_points = 10000 #number of simulation points
N_l = 10**-9 #initial analyte (virus) concentration

s = N_0*f/N_l
alpha = C3/(C2*N_l)

RunAndPlot(a0,N0,C1_n,C2,N_l,s,alpha,simulation_max,n_points)
#plt.savefig(’timecourse.pdf’)

#same function as RunAndPlot but does not plot
```

```
def Run_Sim(a0,N0,C1_n,C2,N_l,s,alpha,simulation_max):
    #create time array
    t = np.linspace(0,simulation_max,1000)
    species_0 = [a0,N0]
    argtuple = (C1_n,C2,N_l,s,alpha)
    #solving the ODE
    species = odeint(timecourse, species_0, t, args=argtuple)
    return species

#Defining normalized cell agglutination (1-N(t)) and bound nanobody (a) as a function of initial analyte concentration (N_l)

def agglutination_N_l_range(N_l, n_points_N_l, simulation_max, C1_n, N0):
    i = 0
    agglutination = np.zeros((n_points_N_l, 2))
    while i < n_points_N_l:
        s = N_0*f/N_l[i]
        alpha = C3/(C2*N_l[i])
        #simulating
        species = Run_Sim(a0,N0,C1_n,C2,N_l[i],s,alpha,simulation_max)
        a = species[:,0]
        cell_agg = 1-species[:,1]
        agglutination[i,0] = a[-1]
        agglutination[i,1] = cell_agg[-1] #storing last value of cell_agg and a
        i +=1
    return agglutination

C1_n = 1

t_end = 18 #in hrs, time of agglutination

#Initial concentration of cells
cells = 10**5 #cells per mL
N_0 = cells*1000/(6.022*10**23) #converting cells/mL to molarity

n_points_N_l = 600
min_N_l = -11 #concentrations are 10^(min/max_N_l)
max_N_l = -5

N_l = np.logspace(min_N_l, max_N_l, num=n_points_N_l) #concentration in M
agglutination = agglutination_N_l_range(N_l, n_points_N_l, t_end, C1_n, N0)
a = agglutination[:,0]
cell_agg = agglutination[:,1]

#plotting a and 1-N
plt.plot(N_l, a, label=’a’)
plt.plot(N_l, cell_agg, label=’1-N’)

plt.xscale(’log’)
plt.xlabel(’Viral load (M)’)
plt.title(’Cell agglutination model, t={:} hr’.format(t_end))
plt.legend()
plt.show()

#testing over a range of initial yeast cell concentrations
C1_n = 1

t_end = 15
#Initial concentration of cells
cells = np.array((10**3,10**9,10**11)) #cells per mL

i=0
```

```
while i < len(cells):
    N_0 = cells[i]*1000/(6.022*10**23)
    n_points_N_l = 500
    min_N_l = -10 #concentrations are 10^(min/max_N_l)
    max_N_l = -4
    N_l = np.logspace(min_N_l, max_N_l, num=n_points_N_l) #concentration in M
    agglutination = agglutination_N_l_range(N_l, n_points_N_l, t_end, C1_n, N0)
    #a = agglutination[:,0]
    cell_agg = agglutination[:,1]
    #plotting 1-N
    plt.plot(N_l, cell_agg, label=’{:0.0e}’.format(cells[i]))
    plt.xscale(’log’)
    plt.xlabel(’Viral load (M)’)
    plt.ylabel(’Fraction Agglutinated’)
    plt.title(’Cell agglutination model, t={:} hr’.format(t_end))
    plt.legend(title=’Yeast Conc. (Cells/mL)’)
    #finding N_l value where 1-N is at a maximum
    max_cell_agg = max(cell_agg)
    max_agg_location = np.where(cell_agg == max_cell_agg)[0] #index of max cell agglutination value
    max_agg_location = max_agg_location[0]
    N_l_max_agg = N_l[max_agg_location]
    #finding N_l value for given limit of detection
    LOD = 0.05 #limit of detection, given as fraction of complete agglutination
    LOD_location = np.where(cell_agg[0:max_agg_location] <= LOD)[0] #only considering values before the max
    LOD_location = LOD_location[-1]
    cell_agg_LOD = cell_agg[LOD_location] #cell_agg value for given LOD, used for vertical line max
    N_l_LOD = N_l[LOD_location]
    #plotting range of sensitivity of assay
    plt.axvline(N_l_max_agg, ymax=(max_cell_agg+0.3), linestyle=’--’, color=’k’)
    plt.axvline(N_l_LOD, ymax=(cell_agg_LOD+0.065), linestyle=’--’, color=’r’)
    i +=1
    #plt.savefig(’sensitivity_range.pdf’)


#testing over a range of agglutination kinetic constants
C1_n = np.array((1,10,100)) #different kinetic constants
t_end = 15
#Initial concentration of cells
cells = 10**8 #cells per mL
i=0
while i < len(C1_n):
    N_0 = cells*1000/(6.022*10**23)
    n_points_N_l = 500
    min_N_l = -13 #concentrations are 10^(min/max_N_l)
    max_N_l = -5

    N_l = np.logspace(min_N_l, max_N_l, num=n_points_N_l) #concentration in M
    agglutination = agglutination_N_l_range(N_l, n_points_N_l, t_end, C1_n[i], N0)
    #a = agglutination[:,0]
    cell_agg = agglutination[:,1]

    #plotting 1-N
    plt.plot(N_l, cell_agg, label=’{:}’.format(C1_n[i]))

    plt.xscale(’log’)
    plt.xlabel(’Viral load (M)’)
    plt.ylabel(’Fraction Agglutinated’)
    plt.title(’Cell agglutination model, t={:} hr’.format(t_end))
    plt.legend(title=’Agglutination constant ($h^{-1}$)’)

    #finding N_l value where 1-N is at a maximum
    max_cell_agg = max(cell_agg)*0.97 #setting sensitivity range max to 97% of max
    max_agg_location = np.where(cell_agg >= max_cell_agg)[0] #index of max cell agglutination value
    max_agg_location = max_agg_location[0]
    N_l_max_agg = N_l[max_agg_location]

    #finding N_l value for given limit of detection
    LOD = 0.05 #limit of detection, given as fraction of complete agglutination
    LOD_location = np.where(cell_agg[0:max_agg_location] <= LOD)[0] #only considering values before the max
    LOD_location = LOD_location[-1]
    cell_agg_LOD = cell_agg[LOD_location] #cell_agg value for given LOD, used for vertical line max
    N_l_LOD = N_l[LOD_location]

    #plotting range of sensitivity of assay
    plt.axvline(N_l_max_agg, ymax=(max_cell_agg-0.04), linestyle='--', color='k')
    plt.axvline(N_l_LOD, ymax=(cell_agg_LOD+0.045), linestyle='--', color='r')
    i +=1
    #plt.savefig('C1n.pdf')

#testing over a range of agglutination kinetic constants
C1_n = np.logspace(0,2,num=40) #different kinetic constants
t_end = 18
#Initial concentration of cells
cells = 10**8 #cells per mL
min_detectable_conc = np.zeros((len(C1_n),1))
i=0
while i < len(C1_n):
    N_0 = cells*1000/(6.022*10**23)
    n_points_N_l = 500
    min_N_l = -14 #concentrations are 10^(min/max_N_l)
    max_N_l = -6
    N_l = np.logspace(min_N_l, max_N_l, num=n_points_N_l) #concentration in M
    agglutination = agglutination_N_l_range(N_l, n_points_N_l, t_end, C1_n[i], N0)
    #a = agglutination[:,0]
    cell_agg = agglutination[:,1]

    #finding N_l value where 1-N is at a maximum
    max_cell_agg = max(cell_agg)
    max_agg_location = np.where(cell_agg == max_cell_agg)[0] #index of max cell agglutination value
    max_agg_location = max_agg_location[0]
    #N_l_max_agg = N_l[max_agg_location]

    #finding N_l value for given limit of detection
    LOD = 0.05 #limit of detection, given as fraction of complete agglutination
    LOD_location = np.where(cell_agg[0:max_agg_location] <= LOD)[0] #only considering values before the max
    LOD_location = LOD_location[-1]
    cell_agg_LOD = cell_agg[LOD_location] #cell_agg value for given LOD, used for vertical line max
    N_l_LOD = N_l[LOD_location]
    min_detectable_conc[i] = N_l_LOD
    i +=1

plt.plot(C1_n,min_detectable_conc)
plt.title('Yeast Cell Concentration = {:0.0e} cells/mL'.format(cells))
plt.xlabel('Agglutination Rate Constant ($h^{-1}$)')
plt.ylabel('Minimum Detectable Conc. (M)')
#plt.xscale('log')
#plt.yscale('log')
plt.show()
#plt.savefig('C1n_sim.pdf')

#testing agglutination profile over time
```

```
C1_n = 1

t_end = np.array((5,15,50))
#Initial concentration of cells
cells = 10**7 #cells per mL

i=0
while i < len(t_end):

    N_0 = cells*1000/(6.022*10**23)

    n_points_N_l = 500
    min_N_l = -11 #concentrations are 10^(min/max_N_l)
    max_N_l = -4

    N_l = np.logspace(min_N_l, max_N_l, num=n_points_N_l) #concentration in M
    agglutination = agglutination_N_l_range(N_l, n_points_N_l, t_end[i], C1_n, N0)
    #a = agglutination[:,0]
    cell_agg = agglutination[:,1]

    #plotting 1-N
    plt.plot(N_l, cell_agg, label='{:}'.format(t_end[i]))

    plt.xscale('log')
    plt.xlabel('Viral load (M)')
    plt.ylabel('Fraction Agglutinated')
    plt.title('Cell agglutination model, Yeast Conc.={:0.0e} (Cells/mL)'.format(cells))
    plt.legend(title='Agglutination Time (hr)')

    #finding N_l value where 1-N is at a maximum
    max_cell_agg = max(cell_agg)
    max_agg_location = np.where(cell_agg == max_cell_agg)[0] #index of max cell agglutination value
    max_agg_location = max_agg_location[0]
    N_l_max_agg = N_l[max_agg_location]

    #finding N_l value for given limit of detection
    LOD = 0.05 #limit of detection, given as fraction of complete agglutination
    LOD_location = np.where(cell_agg[0:max_agg_location] <= LOD)[0] #only considering values before the max
    LOD_location = LOD_location[-1]
    cell_agg_LOD = cell_agg[LOD_location] #cell_agg value for given LOD, used for vertical line max
    N_l_LOD = N_l[LOD_location]

    #plotting range of sensitivity of assay
    plt.axvline(N_l_max_agg, ymax=(max_cell_agg+0.09), linestyle='--', color='k')
    plt.axvline(N_l_LOD, ymax=(cell_agg_LOD+0.06), linestyle='--', color='r')
    i +=1
    #plt.savefig('agg_time.pdf')

#calculating dynamic range as function of agglutination time

t_end = np.logspace(0,1.8,num=15)
#Initial concentration of cells
cells = 10**7 #cells per mL

min_detectable_conc = np.zeros((len(t_end),1))

i=0
while i < len(t_end):

    N_0 = cells*1000/(6.022*10**23)
    n_points_N_l = 500
    min_N_l = -10 #concentrations are 10^(min/max_N_l)
    max_N_l = -5

    N_l = np.logspace(min_N_l, max_N_l, num=n_points_N_l) #concentration in M
    agglutination = agglutination_N_l_range(N_l, n_points_N_l, t_end[i], C1_n, N0)
    #a = agglutination[:,0]
    cell_agg = agglutination[:,1]

    #finding N_l value where 1-N is at a maximum
    max_cell_agg = max(cell_agg) #also used for vertical line max
    max_agg_index = np.where(cell_agg == max_cell_agg)[0] #index of max cell agglutination value
    max_agg_index = max_agg_index[0]
    #N_l_max_agg = N_l[max_agg_index]

    #finding N_l value for given limit of detection
    LOD = 0.1 #limit of detection, given as fraction of complete agglutination
    LOD_index = np.where(cell_agg[0:max_agg_index] <= LOD)[0] #only considering values before the max
    LOD_index = LOD_index[-1]
    cell_agg_LOD = cell_agg[LOD_index] #cell_agg value for given LOD, used for vertical line max
    N_l_LOD = N_l[LOD_index]

    #saving sensitivity range value
    min_detectable_conc[i] = N_l_LOD

    i +=1

plt.plot(t_end,min_detectable_conc)
plt.title(’Yeast Cell Concentration = {:0.0e} cells/mL’.format(cells))
plt.xlabel(’Agglutination Time (hr)’)
plt.ylabel(’Minimum Detectable Virus Conc. (M)’)
#plt.yscale(’log’)
plt.show()
#plt.savefig(’min_detect_sim.pdf’)

#calculating limit of detection as function of yeast concentration

t_end = 15
#Initial concentration of cells
#cells = np.array((10**6,10**7,10**8,10**9,10**10,10**11,10**12)) #cells per mL
cells = np.logspace(6,14, num=50)

min_detectable_conc = np.zeros((len(cells),1))

i=0
while i < len(cells):

    N_0 = cells[i]*1000/(6.022*10**23)

    n_points_N_l = 500
    min_N_l = -10 #concentrations are 10^(min/max_N_l)
    max_N_l = -3

    N_l = np.logspace(min_N_l, max_N_l, num=n_points_N_l) #concentration in M
    agglutination = agglutination_N_l_range(N_l, n_points_N_l, t_end, C1_n, N0)
    #a = agglutination[:,0]
    cell_agg = agglutination[:,1]

    #finding N_l value where 1-N is at a maximum
    max_cell_agg = max(cell_agg) #also used for vertical line max
    max_agg_index = np.where(cell_agg == max_cell_agg)[0] #index of max cell agglutination value
    max_agg_index = max_agg_index[0]
    #N_l_max_agg = N_l[max_agg_index]

    #finding N_l value for given limit of detection
    LOD = 0.1 #limit of detection, given as fraction of complete agglutination
    LOD_index = np.where(cell_agg[0:max_agg_index] <= LOD)[0] #only considering values before the max
    LOD_index = LOD_index[-1]
    cell_agg_LOD = cell_agg[LOD_index] #cell_agg value for given LOD, used for vertical line max
    N_l_LOD = N_l[LOD_index]

    #saving sensitivity range value
    min_detectable_conc[i] = N_l_LOD

    i +=1

plt.plot(cells,min_detectable_conc)
plt.title('t={:} hr'.format(t_end))
plt.xlabel('Yeast Conc. (cells/mL)')
plt.ylabel('Minimum Detectable Virus Concentration (M)')
plt.xscale('log')
plt.yscale('log')
#plt.savefig('yeast_conc.pdf')
```