Modeling

overview

In this part, modeling group created three models to describe or predict some essential fact of our project. The first model is created to predict the influence of different RPA temperature and target DNA concentration in the experiment. This modeling is based on Michaelis-Menten equation and show consistence with our experiment results. The second model created a Neural Network (NN) based on Tensorflow to predict the binding capacity of protein-Aptamer. This model is an innovative way to find the influence factor during the binding between protein and Aptamer. The Neural Network (NN) helped us to role out unimportant factors among hundreds of parameters. The prediction results also show consistence with the existing data. The last model is a physical model. This model provides theoretical evidence to our hardware part. This modeling revealed the process how a liquid membrane can carry particles and focused on the physical theory itself, which shows interdisciplinary feature in iGEM.

1 CESAR-II *in sillico* verification based on Michaelis-Menten equation

For CESAR-II, as mentioned in our design part, RPA is needed before detection step. Why? Though AsCas12a is reported to be sensitive (5e8 aM)[1], we want CESAR-II to perform fast in clinical or POCT (point-of-

[1] Gootenberg JS, Abudayyeh OO, Kellner MJ, Joung J, Collins JJ, Zhang F. Multiplexed and portable nucleic acid detection platform with Cas13, Cas12a, and Csm6. Science. 2018 Apr 27;360(6387):439-444. doi: 10.1126/science.aaq0179. Epub 2018 Feb 15. PMID:

care testing) situations. Thus, the key point to answer for the necessity of RPA lies not in whether CESAR-II is able to detect target genes such as ARGs in samples, but lies in how much faster the whole detection process with RPA will be compared with that without RPA. Our assumption is that RPA is indeed necessary. More specifically, even with the extra RPA process (approximately 20 min), the time consumption of CESAR-II procedure will be significantly reduced. To support this assumption, we carry out an *in silico* simulation of CESAR-II as following two parts: RPA, Cas12a trans cleavage of fluorescence reporter.

1.1 RPA process

The RPA procedure can be described as:

$$N_t = N_0(1+p)^t(0) \tag{1}$$

in which N_t is the template concentration after time t and N_0 is the initial concentration of the template. Based on experimental observations that the rpa efficiency at 39^{o}C is the best among 37 42 ^{o}C, p is estimated to be 0.38, 0.40, 0.42 at 37, 39, 41^{o}C. Since the detection threshold of Cas12a is reported to be $0.5nM$, the initial condition N_0 is set to be this value. The concentration plot of RPA product is shown as below:

29449508; PMCID: PMC5961727.

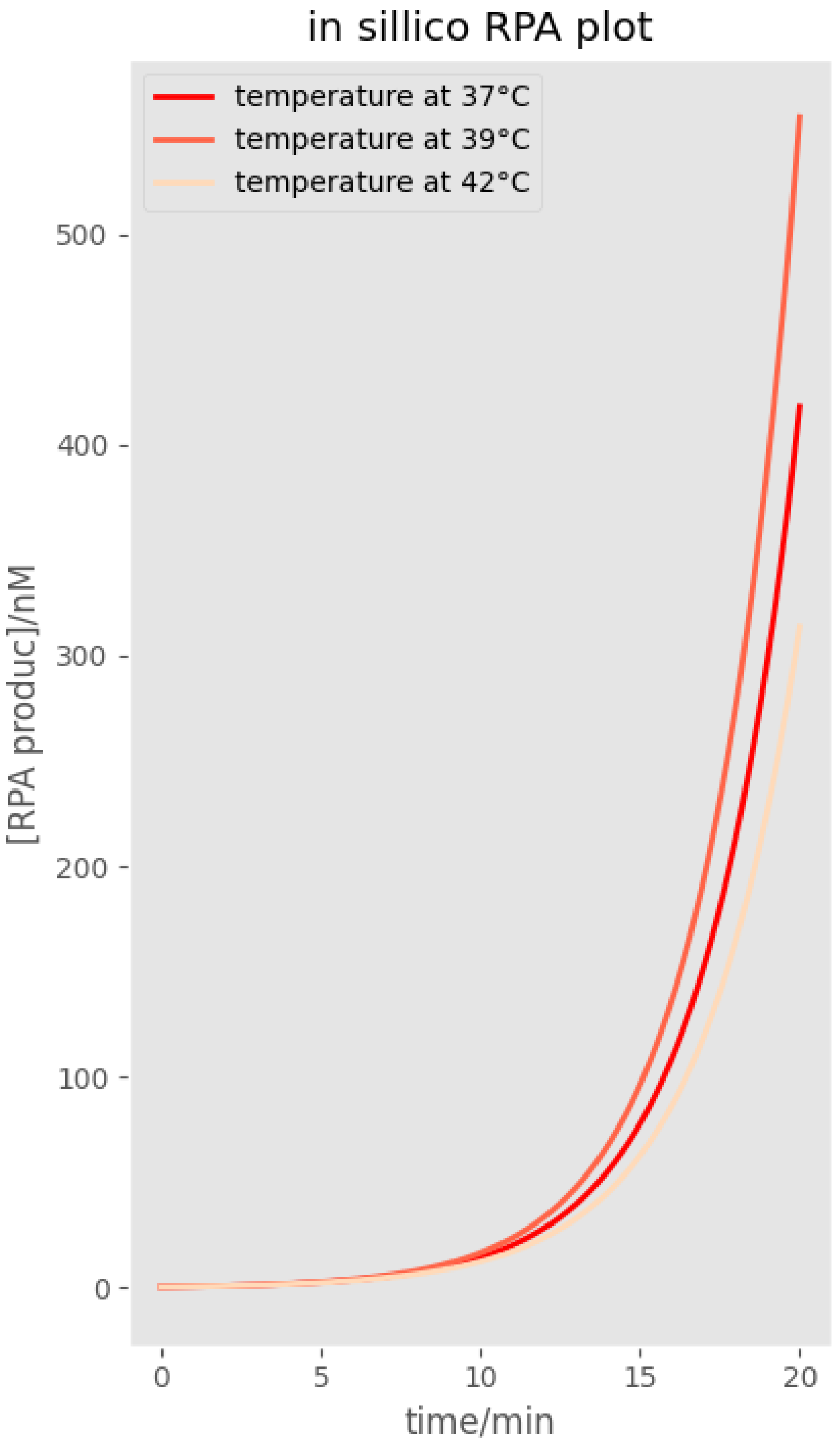


Fig 1: *in sillico* RPA plot

1.2 Michaelis-Menten equation

Using our kinetic data, we estimated the rate constants for the different reactions to create a simple ODE model.

The first reaction is the combination between Cas12a and crRNA as below :

$$Cas12a + crRNA \longrightarrow Cas12a_{crRNA} \tag{2}$$

$$\frac{d[Cas12a]}{dt} = -k_{cr} \cdot [Cas12a] \cdot [crRNA] \tag{3}$$

$$\frac{d[crRNA]}{dt} = -k_{cr} \cdot [Cas12a] \cdot [crRNA] \tag{4}$$

The second step is the complex $Cas12a_{crRNA}$combines with the target DNA which consists of two process: the PAM recognition and the R-loop formation. Both processes are describe as a equation as below:

$$Cas12a_{crRNA} + DNA_{target} \longrightarrow Cas12a_{crRNA/target} \tag{5}$$

$$\frac{d[Cas12a_{crRNA}]}{dt} = k_{cr} \cdot [Cas12a] \cdot [crRNA] - k_{combine} \cdot [DNA_{target}] \cdot [Cas12a_{crRNA}] \tag{6}$$

$$\frac{d[DNA_{target}]}{dt} = -k_{combine} \cdot [DNA_{target}] \cdot [Cas12a_{crRNA}] \tag{7}$$

Then the activated Cas12a have the ability to cleave the single-stranded DNA. This process can be described as below:

$$Cas12a_{crRNA/target} + ssDNA \overset{tans-cleavage}{\longrightarrow} Cas12a_{crRNA/target} + ssDNA_{Deg} \tag{8}$$

$$\frac{d[Cas12a_{crRNA/target}]}{dt} = k_{combine} \cdot [DNA_{target}] \cdot [Cas12a_{crRNA}] \tag{9}$$

According to Michaelis-Menten equation which is

$$V_0 = \frac{V_{max}[S]}{K_M + [S]} \tag{10}$$

$$\frac{d[DNA_{single-stranded}]}{dt} = \frac{k_{cat} \cdot [Cas12a_{crRNA/target}] \cdot [DNA_{single-stranded}]}{K_M + [DNA_{single-stranded}]} \tag{11}$$

The constants in these equations are shown below:

Rate constant	Value	Reference or Rationale
k_{cr}	$1[min^{-1}]$	Estimated from Mekler et al.(2016) Nucleic Acids Resarch[2]
$k_{combine}$	$0.00183[nM^{-1}min^{-1}]$	Estimated from experiment
k_M	$500[nM]$	Estimated from Weitz et al.(2014) Nature Chemistry[3]
k_{cat}/k_M	10	Estimated from experiment

Table 1: Rate constant used when modeling

The initial condition in these equations are based on experimental system and existing system reported:

$$[Cas12a_{t_0}] = 20nM[crRNA_{t_0}] = 250nM[ssDNA_{t_0}] = 200nM \tag{12}$$

Note that $[DNA_{target}]$ is based on results in RPA simulation:

$$[DNA_{target}](310.15K) = 418.34nM[DNA_{target}](312.15K) = 555.57nM[DNA_{target}](314.15K) = 313.73n \tag{13}$$

The equation solved in different conditions is shown below:

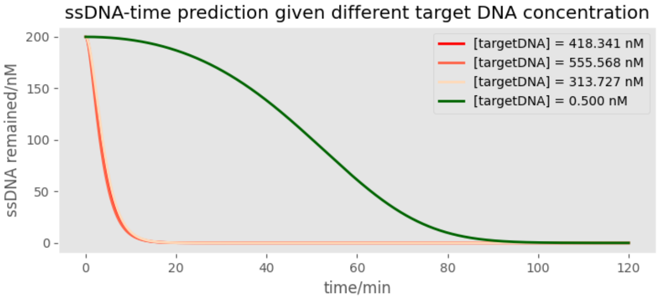


Fig 2: ssDNA-time prediction given different target DNA concentration

The fluorescence is given by:

$$fluorescence(t) = (1 - DNA_{single-stranded}(t)/[ssDNA_{t_0}]) \cdot 100\% \tag{14}$$

The fluorescence in different conditions is shown below:

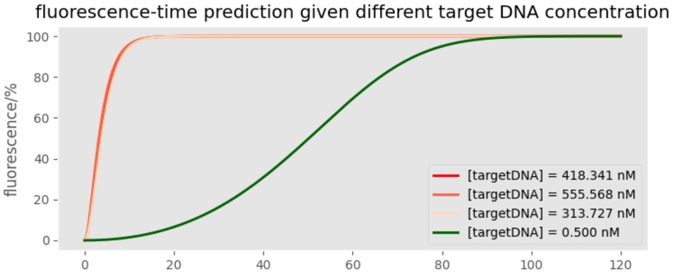


Fig 3: fluorescence-time prediction given different target DNA concentration

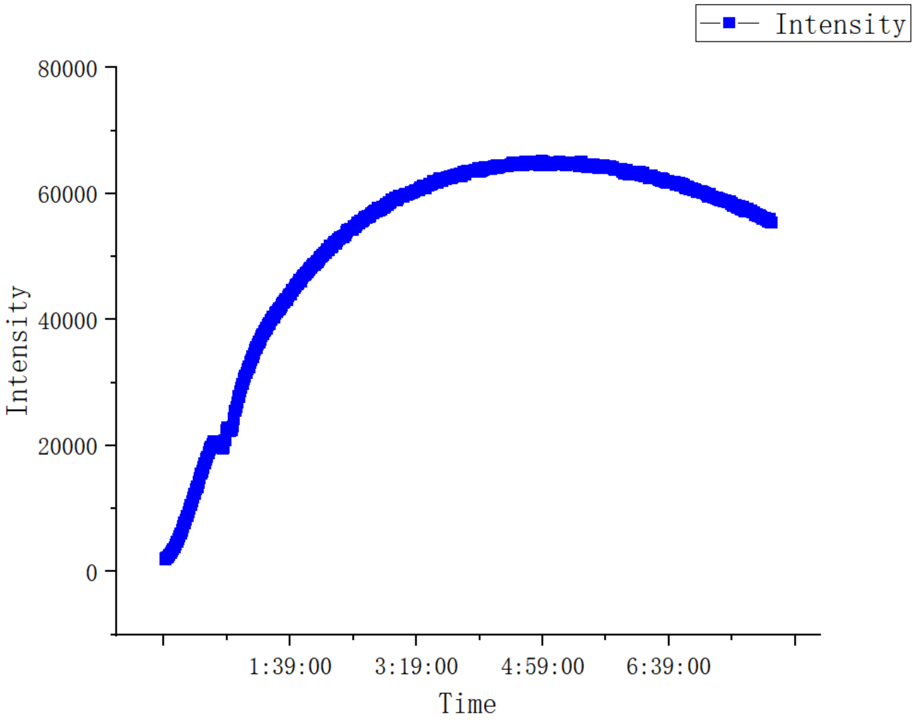


Fig 4: Wet experiment data

Above is our prediction and wet experiment data, which shows consistency in fluorescence trend. See more details in repository(W6)

1.3 Meaning of the model

This model based on Michaelis-Menten equation and aims to find the influences of different RPA temperature and target DNA concentration. The fluorescence curve shows a sigmoid behavior, which partly agrees with wet lab observations. It is illustrated that without pre-detection RPA process, the detection process will be significantly more time-consuming. Controling $[DNA_{target}]$ in sample the same, the detection with RPA reaches 60% fluorescence within 5 minutes, whereas the detection without RPA reaches 60% fluorescence at more than 50 minues, revealing a ten-fold difference. The model supports our previous assumption that it is indeed necessary to introduce RPA process. What's more, the prediction shows that the influences of RPA temperature does not effect the detection significantly in terms of time consumption. It can be also inferred from RPA simulation that Cas12a could be even more sensitive (5e6 aM) since RPA amplifies $[DNA_{target}]$ in sample at a scale of approximately 10e3 (from 0.5nM to 555nM at 39^{o}C). This model is created to give some predictions of our wet experiments and helps to verify CESAR-II design.

2 The Neural Network Built Based on Tensorflow to Predict the binding capacity of Protein-Aptamer

2.1 Why Neural Network(NN)?

When predicting protein-Aptamer binding, there are too many factors to consider, which means we need to use hundreds of parameters to describe the binding. In previous machine learning, people need to exact some important features by themselves and send them to computer to learn. With the development of algorithm, neural net work help us to leap over the feature extraction stage. This will benefit us to reduce the error caused by man and this is why we build this model.

2.2 Principle of neural network

2.2.1 Neuron

Neural Network(NN) is a kind of new algorithm which simulates the way how neuron works. Generally, neuron is the basic computing unit of neural network, also known as node or unit. It can take input from other neurons or external data, and then calculate an output. Each input value has a weight, which depends on the importance of this input compared to other input values. Then a specific function f is executed on the neuron, defined as shown in the figure below. This function performs an operation on all input values of the neuron and its weight.

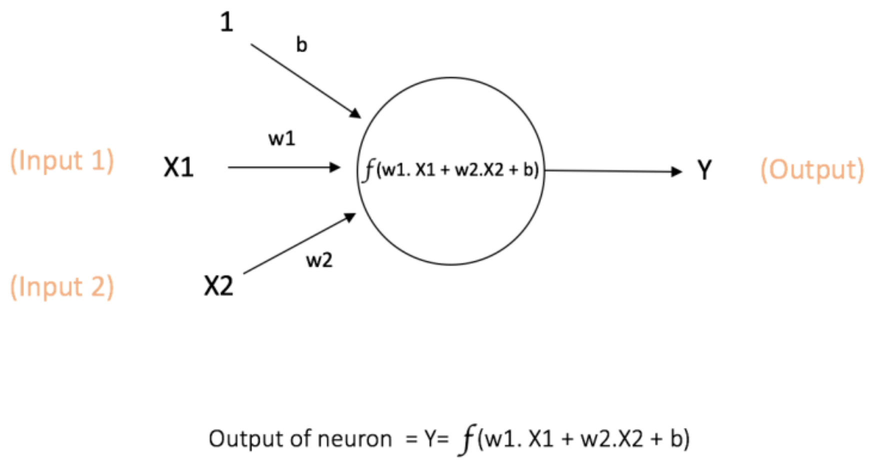


Fig 5: How a neuron works

As can be seen from the above figure, in addition to the weight, there is also a bias value bias with an input value of 1. The function f here is a nonlinear function called activation function. Its purpose is to introduce nonlinearity into the output of neurons. Because the data in the real world are nonlinear, we hope that neurons can learn these nonlinear representations.

2.2.2 Neural Network(NN)

Below is a very simple NN:

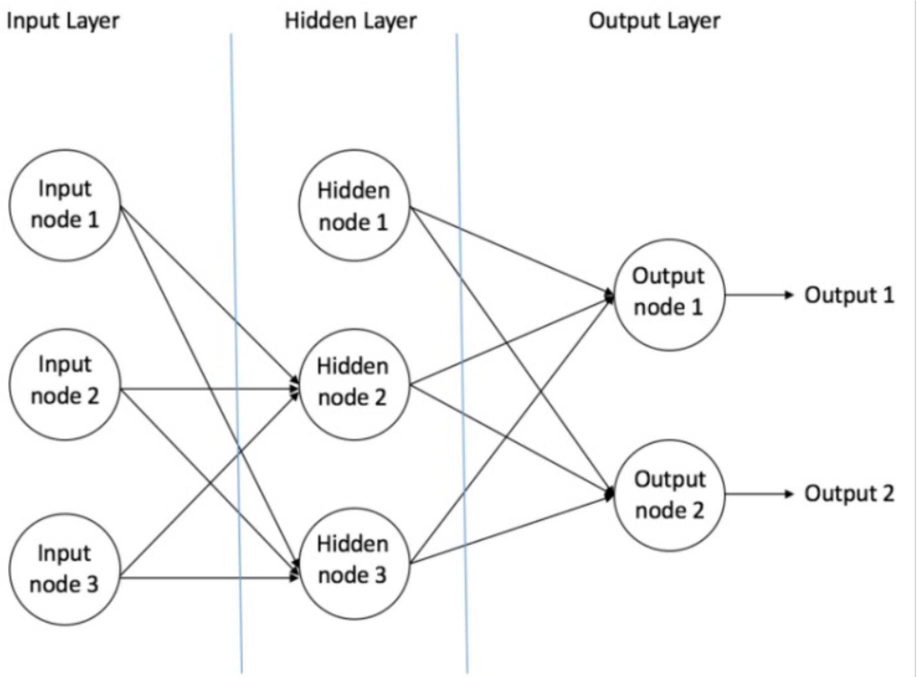


Fig 6: How a NN works

As shown in the figure above, the neural network is divided into three network layers, namely input layer, hidden layer and output layer. Each network layer contains multiple neurons, and each neuron will be connected with the neurons in the previous layer adjacent to each other. These connections are actually the input of the neuron. According to the different layers of neurons, the neurons of neural network can also be divided into three types:

Input neurons: located in the input layer, they mainly transfer information from the outside into the neural network, such as picture information, text information, etc. these neurons do not need to perform any calculation, but just as information transmission, or data into the hidden layer.

Hidden neurons: located in the hidden layer, the neurons in the hidden layer do not have direct connection with the outside world. They are all indirectly connected with the outside world through the input layer in front and the output layer at the back, so it is called hidden layer. The above figure only has one network layer, but in fact, there can be many hidden layers, far more than one, of course, there can be no, that is, only input Layer and output layer. The neurons in the hidden layer will perform the calculation, convert the input information of the input layer through calculation, and then output it to the output layer.

Output neuron: located in the output layer, the output neuron is to output the information from the hidden layer to the outside world, that is to output the final results, such as classification results.

2.3 Multilayer perceptron(MLP)

Single layer perceptron only has input layer and output layer, so it can only learn linear function, while multilayer perceptron has one or more hidden layers, so it can learn nonlinear function. Here is an example of a MLP with a hidden layer:

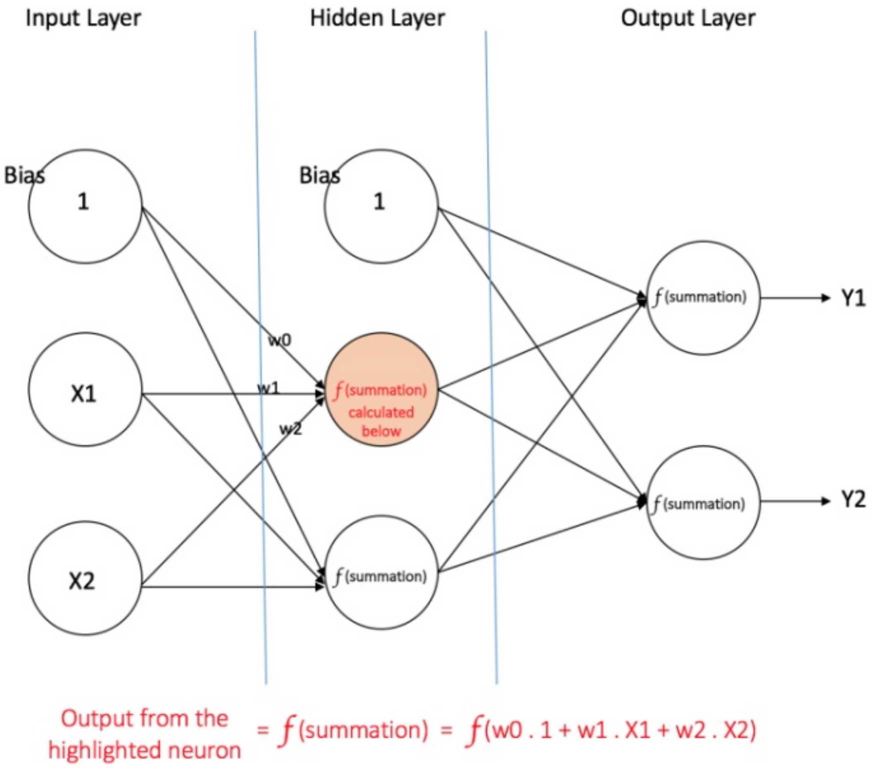

Input Layer
Hidden Layer
Output Layer
Bias
1
Bias
1
X1
X2
w0
w1
w2
f (summation) calculated below
f (summation)
f (summation)
f (summation)
Y1
Y2
Output from the highlighted neuron = f (summation) = f(w0 . 1 + w1 . X1 + w2 . X2)

Fig 7: A NN with a MLP

The above figure shows that the MLP has two input values X1 and X2, two neurons in the hidden layer, and then two output values Y1 and Y2. The calculation of one of the hidden neurons is also shown as:

$$f(summation) = f(w_0 * 1 + w_1 * X_1 + w_2 * X_2) \tag{15}$$

2.4 Details on how our NN works

2.4.1 Feature extraction

For a pair of protein and aptamer sequences, it is obviously not a good choice to characterize their sequences directly for that many proteins and aptamer properties will be lost during this process.

Therefore, in this model, $pse - ACC$ and $pse - KNC$, which is widely applied in bioinformatics, convert the protein sequences and the aptamer sequences into 70 and 20 dimension vectors according to the physical and chemical properties of amino acids and nucleotides, respectively. Then, they are spliced together to generate a 90 - dimensional vectors as inputs of fixed length, and the protein - aptamer pair is marked as 'can be combined 'or 'cannot be combined' in one - hot category coding. Positive class is marked as [1, 0], while negative class is marked as [0, 1].

In the process of using the pse-KNC processing aptamer sequence, we choose the open source tools pseKNC-general. The home page is (website 1) , a relevant procedure is enclosed in the attachment. Both for RNA and DNA, shift, slide, rise, tilt, twist, roll six kinds of physical and chemical properties are selected as indicators of generating sequence, $\lambda$is set to 4, and K is set to 2 (i.e., only the nature of the thinking of the binary sequence group).

In the process of using the pse - AAC processing protein, we chose five kinds of physical and chemical properties of amino acids: polarity, Secondary structure, Molecular volume, Codon diversity, Electrostatic charge as indicators, λ is set to 50 (that is, considering most 50 together the sequence of amino acids)

The knowledge of Pse-AAC and Pse-KNC representing protein and amino acid of knowledge comes from (website 2 and 3). Relevant formulas are also available on the two sites.

2.4.2 Neural network construction

Since a complete feature project has been carried out, the traditional feedforward double hidden layer neural network is selected here. The code is shown in the annex Network.py, which is built using TensorFlow and the operating environment is Python3, TensorFlow 1.14.0. The trained model stored in the Model folder contains the following functions:

2.4.3 Inference

In the framework of the double hidden layer neural network, the two hidden layers are nonlinear processed with relu function, and both hidden layers have 50 neurons.

2.4.4 Training

During the training process of neural network, the BATCH_SIZE was 256, with a total of 1024 training samples. Another 351 instances that did not intersect with the training set were used as the verification set to demonstrate the generalization error of neural network. The final generalization accuracy converged at 0.703704. Considering a sample balance has been made, the accuracy of random guesses is 0.5.

2.4.5 Prediction

Input a 90-dimensional vector to predict whether it can be combined through the existing neural network model, and output 1 or 0 (1 means it can be combined, 0 means it cannot be combined).

2.4.6 Experimental data

Experiment required protein-aptamer to come from(website 4).the remaining 1375 after processing (mainly because of the category imbalance problem, more negative cases were deleted), pse-AAC, PSE -KNC processing see the file feature_balance. npy and Label_balance. npy.

2.5 Evaluation of neural networks

There are two important parameters to evaluate a NN, the train_loss and validate_accuracy.

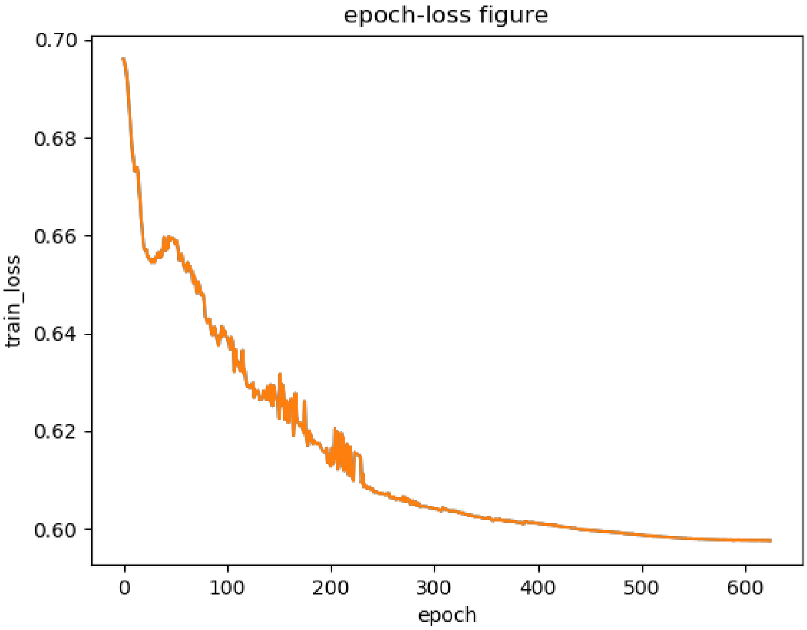


Fig 8: epoch-loss figure

Above is the epoch-loss figure, which is convergent. It means the NN have a low-loss when training, which is significant during deep learning.

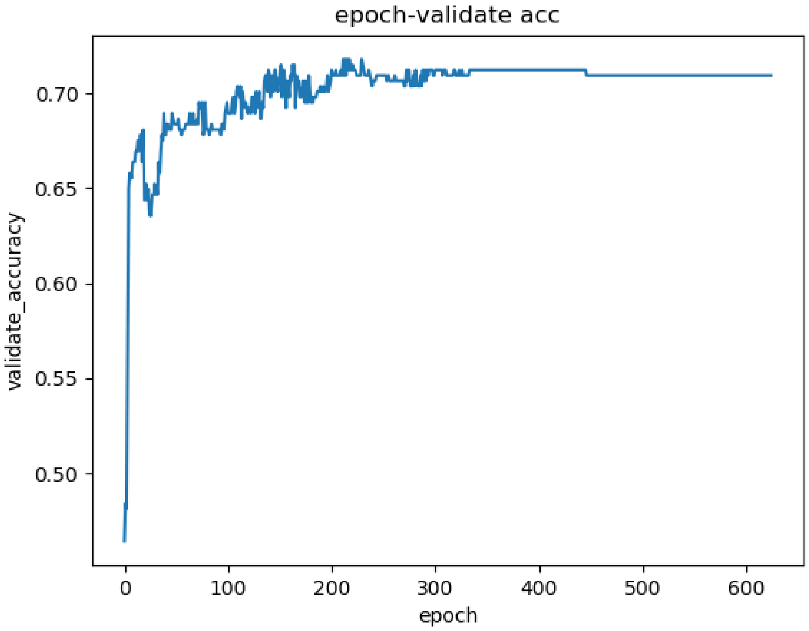


Fig 9: epoch-validate_accuracy figure

Above is the epoch-validate_accuracy figure. This shows our NN's validate_accuracy is good enough(more than 70%), and does not need too many epoches(when epoches are greater than 200, the increment of the validate is not obvious). It shows our NN have the competence for the prediction task.

2.6 Meaning of the Neural Network

Binding between the protein and Aptamer is influenced by a lot of factors, which means we need a great number of parameters to describe the process and do predictions. But it is really hard for us to extract the determining feature. NN helps us to skip this stage and get the results directly, which is significant and innovative in the field.

See the Python file for details(website 5)

Acknowledge: This part of modeling is the result of cooperation with team $NJU - China$.

3 liquid membrane model

3.1 Why we need such a model?

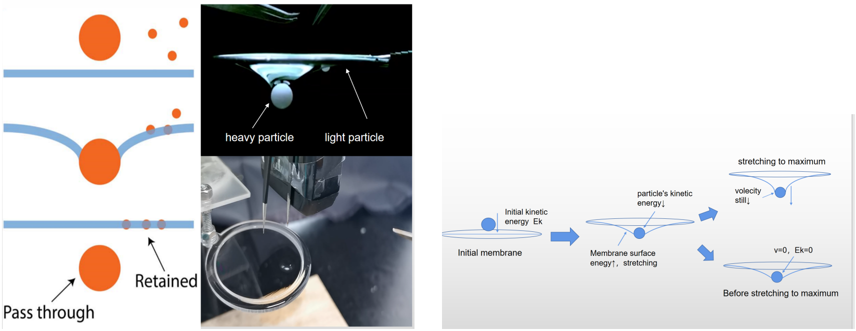
We have designed a special and simple tool to make it easy for our customers to utilize the achievements of 2020 iGEM ShanghaiTech China . As we have presented in our hardware part, the key to the device is the ring carrying the reaction liquid. It is very simple for us to understand the principle of the device, but if it is possible for our device to carry the liquid and if so how much liquid the device can take. To such problems, we should analyze some complex physical process and show why a liquid membrane is a good choice.

3.2 Why a membrane?

The reason why our team choose a liquid membrane is that the enzyme in the reaction solution needs frozen to make sure its activity. By this way, the liquid membrane that can also be frozen is good choice(Both reaction solution and membrane are frozen, which makes it is really easy to transport and use). But we should firstly claim that not every kind of liquid can be used to make such a membrane. The membrane should have enough surface tension to make it possible to carry enough liquid. But this kind of substance should not react with the sample and base. By this way, the membrane should also be easy to make and stable when carrying the reaction liquid. In physics, the surface tension coefficient γ is a good coefficient to describe such properties of the material we need. The problem then can be easy to solve. In our wet experiment, we make a series of membrane with different surface tension cofficient γ to find the ideal membrane, utilizing different concentrations of glycerol.

3.3 The simple description of the physics model we proposed

In general conditions, the membrane in a ring is like a filter: particle passes or stops. We can see it as two conditions. Different conditions are presented in the following figure:


heavy particle
light particle
Retained
Pass through
stretching to maximum
particle's kinetic energy↓
velocity still↓
Initial kinetic energy Ek
Initial membrane
Membrane surface enegy↑, stretching
v=0, Ek=0
Before stretching to maximum

(a) Different conditions (b) Model created

Fig 10. Different conditions when particles pass a liquid membrane

We can conclude that the particle which is heavy or with a high velocity can not be carried by the liquid membrane, that is because they have high energy. When the membrane is not tough enough to tension the liquid sphere on it, the sphere will just pass through the membrane but do not destroy the membrane. Two tangents of sphere and the surface will intersect at the same point as follows:

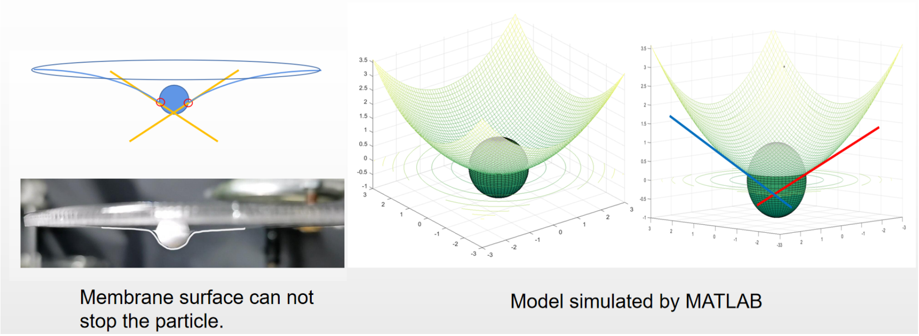

Membrane surface can not stop the particle.
Model simulated by MATLAB

Fig 11. Model developed by MATLAB to show when a particle passes through the membrane

But there are some kinds of liquid membranes that are tough enough. This special property of the membrane can be described by the surface energy. The surface energy of the membrane can be described by the surface tension coefficient γ which is:

$$E_{surface} = \gamma S_{surface} \tag{16}$$

In which $E_{surface}$ is the surface energy of the membrane and $S_{surface}$ is the square of the membrane. This equation gives the surface energy of the membrane which we will use to measure the capacity of membrane.

We can obviously know that every kind of substance have trend to be the lowest energy state. In this model, γ is a constant which depends on the material we choose, so the lowest energy state represents the state with the smallest square S_{min}. We find that this lowest energy state is just the Catenoid Surface Model. The derivation is as follows:

$$S_{surface} = \int 2\pi r ds = 2\pi \int_{-L}^{+L} r\sqrt{1+r'^2}dz \tag{17}$$

We assume that the minimum square can be obtained when the area element change rate is zero. Based on this assumption, we can get the equation:

$$\delta S_{surface} = S_{surface}(r+\delta r) - S_{surface}(r) = 0 \tag{18}$$

Accroding to equation (1) and (2) we can have a new equation:

$$\delta S = 2\pi \int_{-L}^{+L} \delta(r\sqrt{1+r'^2})dz = 0 \tag{19}$$

Solving this equation, we can get a differential equation which is :

$$rr'' = 1 + r'^2 \tag{20}$$

The differential equation can be solved as:

$$r = Kcosh(\frac{z-k}{K}) \tag{21}$$

But according to the symmetry, we can easily realize that k=0.

So we have proven that the lowest energy state is the Catenoid Surface Model(CSM):

$$r(z) = Kcosh(\frac{z}{K}) \tag{22}$$

As the 3D model shows:

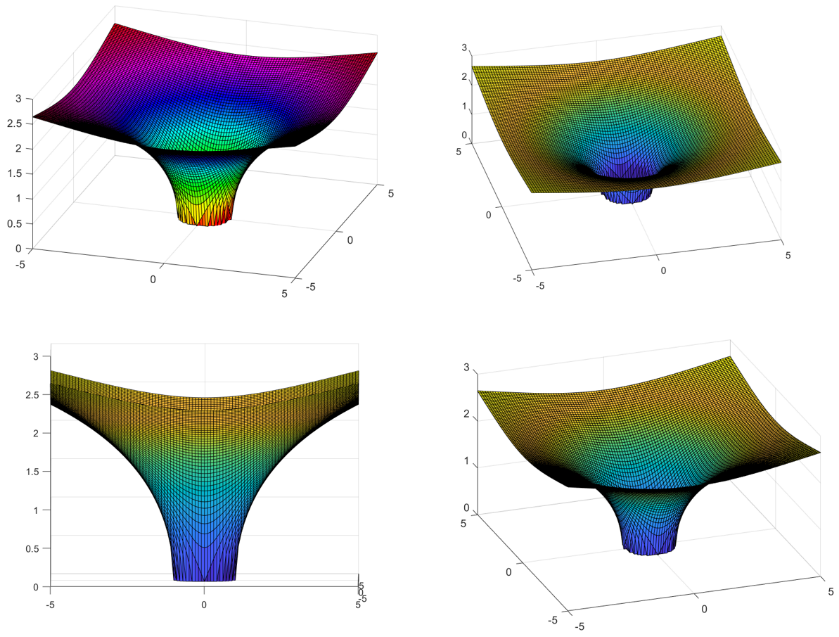


Fig 12. 3D model of the CSM

And our experiment also show the same result with our CSM (In this experiment, we use small solid bobbles to simulate the liquid sphere):

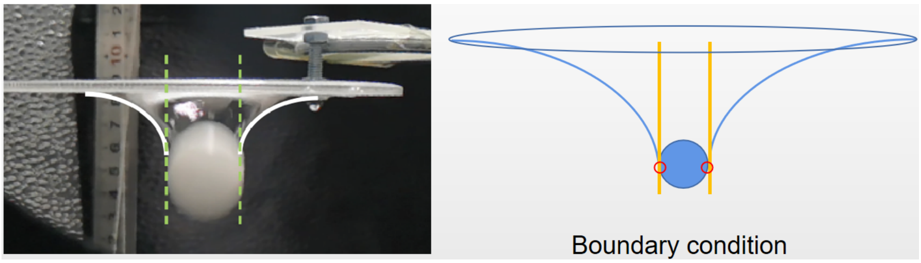


Fig 13. Picture captured by high-speed photography

Now we have had the model of the membrane, then what we should calculate is the energy of the upper surface of the catenoid surface which we can get by calculating the upper surface square of the catenoid surface.

The boundary condition can be concluded by such ways:

$$r(0) = R_{sphere} \tag{23}$$

$$r(z_1) = R_{ring} \tag{24}$$

In order to make the expression easy, we let $R_{sphere} = R_b$ and $R_{ring} = R_f$ According to the boundary condition, we can slve that $K = R_b$ and the integral upper bound $z_1 = R_b \cdot cosh^{-1}(\frac{R_f}{R_b})$.

Then the upper surface square can be calculate as follows:

$$S_{upper} = 2\int_0^{z_1} S_{surface}(z)dz = 2\int_0^{z_1} \pi r^2(z)dz \tag{25}$$

$$S_{upper} = 2\pi\int_0^{z_1} R_b^2 \cdot cos^2h(\frac{z}{R_b})dz = \pi R_b^2[sinh(2cosh^{-1}(\frac{R_f}{R_b}))+(2cosh^{-1}(\frac{R_f}{R_b}))] \tag{26}$$

The maximum surface energy:

$$E_m = \gamma S_{total} = \gamma(S_{upper} + S_{hemisphere}) \tag{27}$$

$$S_{hemisphere} = \frac{1}{2}S_{surface} = 2\pi R_b^2 \tag{28}$$

The initial surface energy:

$$E_0 = \gamma S_{initialmembrane} = \gamma(S_{ring}+S_{innercircular}) = 2\pi\gamma(R_f^2-R_b^2+R_b^2) \quad (29)$$

Which remains constant. The maximum surface energy increment is as follows:

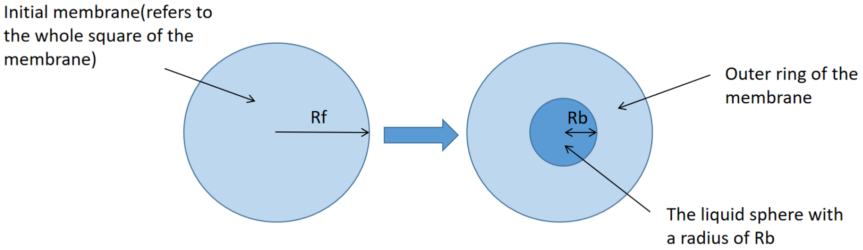

Initial membrane(refers to the whole square of the membrane)
Rf
Rb
Outer ring of the membrane
The liquid sphere with a radius of Rb

Fig 14. Model

$$E_s = E_m - E_0 = \gamma(S_{total} - S_{initialmembrane}) \quad (30)$$

$$E_s = \pi\gamma\{R_b^2[sinh(2cosh^{-1}(\frac{R_f}{R_b})) + (2cosh^{-1}(\frac{R_f}{R_b})) - 2(R_f^2 - R_b^2)]\} \quad (31)$$

Which is consistent with the reference paper equation:

$$E_s = \pi\gamma\{R_b^2[sinh\phi + \phi] - 2(R_f^2 - R_b^2)\} \quad (32)$$

In which $\phi = 2cosh^{-1}(\frac{R_f}{R_b})$.[4]

In conclusion, the CSM is a good model to describe the state when the membrane surface is carrying liquid sphere. In order to make it easy to view the model, we bulid a 3D model as follows:

[4] This is consistent with the $BirgittBoschitschStogin, LukeGockowski, HannahFeldstein, HoustonClaure, JingWang, Tak$ $SingWong'Free-standingliquidmembraneasunusualparticleseparators'$(2018)

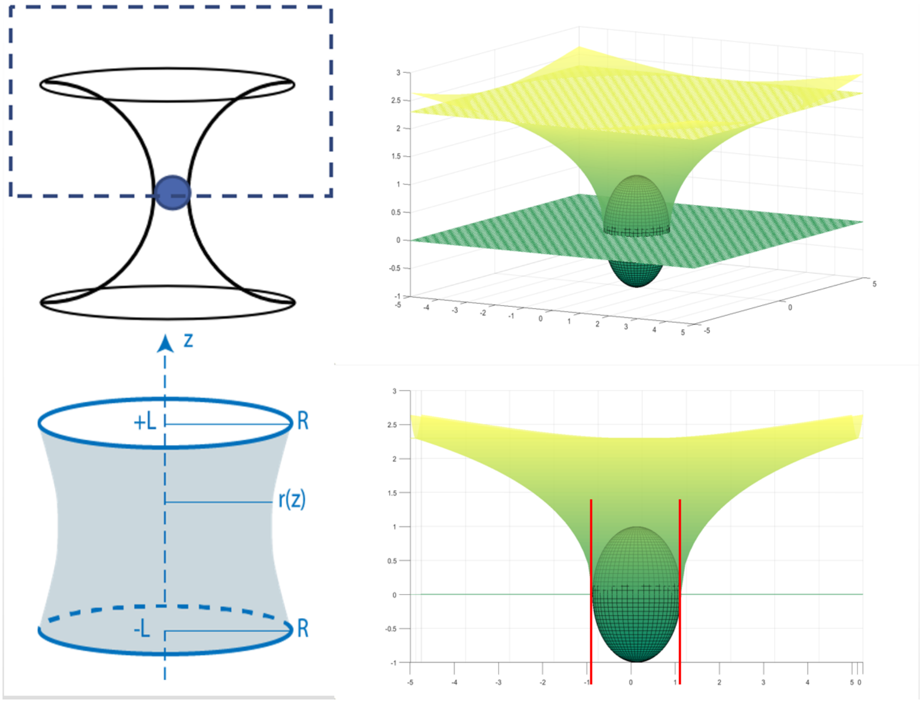


Fig 15. 3D CSM model created by MATLAB

3.4 The reason why our device is possible

After figured out the maximum surface energy increment, we can calculate the capacity of liquid membrane very easily. Given a specific R_f, we can figure out how much liquid it can carry by the following equation:

$$M_{liquid}gh = E_s \tag{33}$$

in which $M_{liquid}gh$ is the gravitational potential energy and E_s is the maximum surface energy increment. When $R_f = 1cm$ and $\gamma = 100mN/m(293.15K)$, its capacity is more than 10uL of reaction solution, which is far more than we need. In a word, this model shows our hardware is possible.

3.5 Meaning of the model

According to the results of the model showed before, we have calculated the maximum theoretical mass that the liquid membrane can suffer. This will help us in designing the radius of the ring and direct us to find the liquid membrane with the best γ. The best γ liquid can be obtained by adding glycerol. And of course the membrane can carry our reaction liquid. By this way, modeling benefits the design of our device and we believe it will make a difference to the world .
Citation list: Website1 Website2 Website3 Website4 Website5