Linear Regression Analysis of *comR* genes on natural competence of *Streptococcus sobrinus*

Presenter: Zong Fan

Email: zongfan2@illinois.edu

Research background

- Natural competence: bacteria take up DNA from environment

- 2 *comR* genes (*comR1*, *comR2*) regulate the natural competence of *Streptococcus sobrinus*.

- There are 3 genotypes for comR genes in this bacterium:
  - **wildtype** (wt): number of gene is unmodified.
  - **knockout** (ko): gene is deleted from the genome.
  - **over-expression** (oe): multiple copies of genes are complemented.

- Natural competence could be assessed by transformation assays.

Purpose of study

1. Quantify the effect of *comR1* and *comR2* in regulating the natural competence

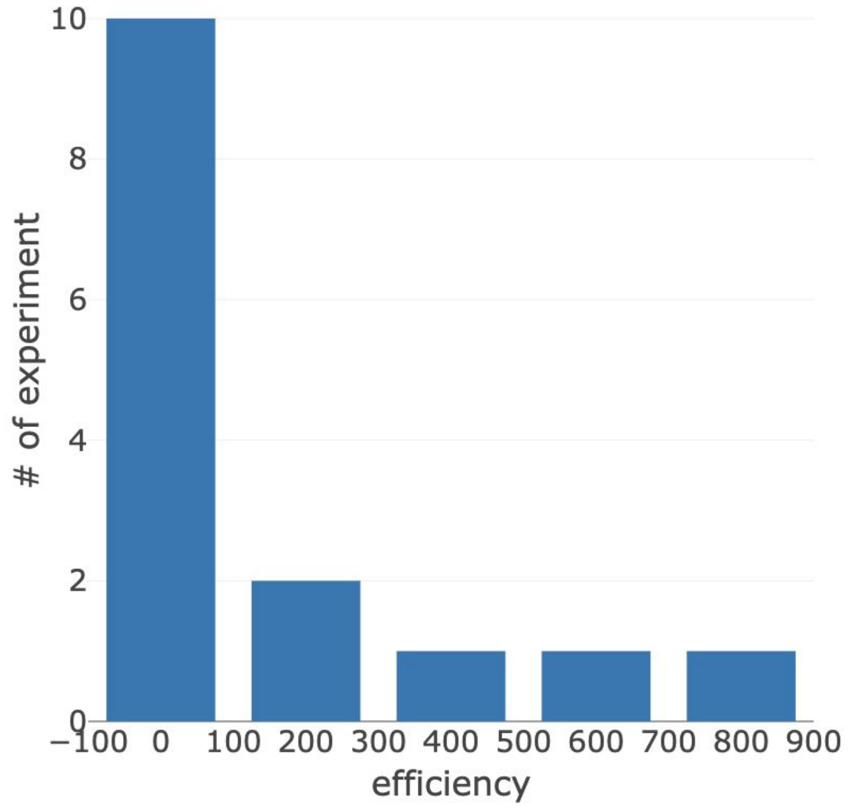
2. Explore whether different genotypes of *comR1* and *comR2* changes the ability of natural competence.

Data

45 observations with 3 duplicates of each genotype combination.
The data is incomplete since genotype *comR1*:oe *comR2*:oe is absent.

Genotype	# of obs	# of rep
comR1:wt *comR2*:wt	12	4
comR1:wt *comR2*:ko	3	1
comR1:wt *comR2*:oe	6	2
comR1:ko *comR2*:wt	3	1
comR1:ko *comR2*:ko	3	1
comR1:ko *comR2*:oe	6	2
comR1:oe *comR2*:wt	6	2
comR1:oe *comR2*:ko	6	2
comR1:oe *comR2*:oe	0	0

Statistic of genotypes. # of obs: number of observation; # of rep: number of replicate



Statistic of efficiency values (bin width=200). The values are averaged across 3 duplicates.

Methods

We apply **linear model with interaction and blocking factors** to predict transformation efficiency with *comR* genotypes.

$$efficiency = \beta_0 + \beta_b + \beta_1 comR1 + \beta_2 comR2 + \beta_{12} comR1 comR2$$

The task is to find the coefficient of each term.

β0: intercept;

βb: coefficient of effect representing the differences between dates of experiment runs;

β1: coefficient of effect of *comR1*;

β2: coefficient of effect of *comR2*;

β12: coefficient of effect depending on both *comR1* and *comR2*

Fit model with categorical variables

The genotypes (3 classes of each gene) and blocks (4 factors) are treated as **one-hot encoded** variables.

To find unique coefficient for each term, drop one in each set of variable to make coefficient matrix full-rank: block1, *comR1*:wt, *comR2*:wt.

The full format of model equation is:

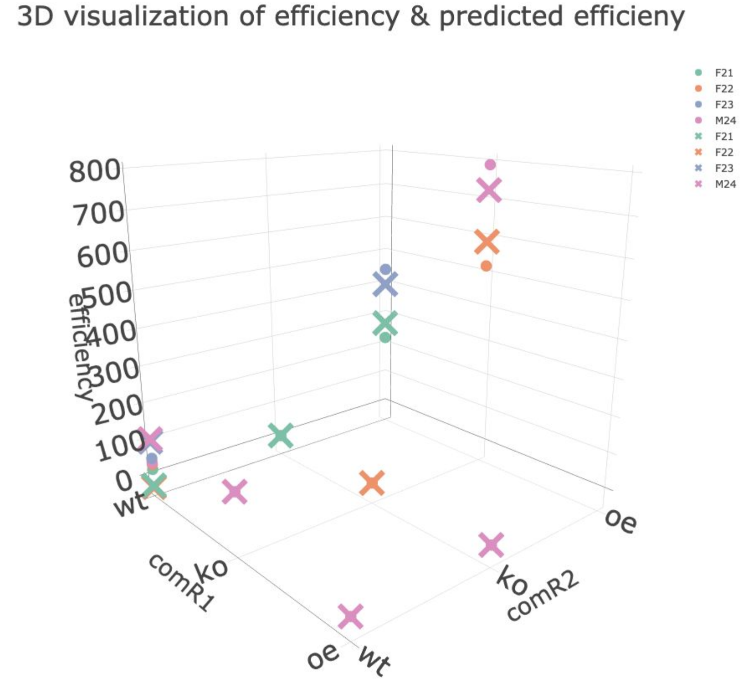
$$\begin{aligned} efficiency = &\beta_0 + \beta_{b2} block_2 + \beta_{b3} block_3 + \beta_{b4} block_4 \\ &+ \beta_2 comR1:ko + \beta_3 comR1:oe + \beta_5 comR2:ko + \beta_6 comR2:oe \\ &+ \beta_{25} comR1:ko\ comR2:kn + \beta_{26} comR1:ko\ comR3:oe \\ &+ \beta_{35} comR1:oe\ comR2:kn + \beta_{36} comR1:oe\ comR3:oe \end{aligned} \quad (1)$$

Results

1. linear model fitting raw efficiency data

comR2:oe dominates the efficiency: application of *comR2*:oe increase efficiency by 296.4 (see Appendix LM results).

Ground-Truth vs Predicted efficiency with LM



circle represents GT efficiency values; cross represents predicted efficiency value

GT vs predicted efficiency in 15 experiments

2. Linear model fits log transformed efficiency data

- Because efficiency data show skewed distribution that most of the values condense within [0, 100], logarithm transformation of efficiency is employed to alleviate the impact of skewness.
- The equation format is formatted as:

$$
\begin{aligned}
log(efficiency) = & \beta_0 + \beta_{b2} block_2 + \beta_{b3} block_3 + \beta_{b4} block_4 \\
& + \beta_2 comR1 : ko + \beta_3 comR1 : oe + \beta_5 comR2 : ko + \beta_6 comR2 : oe \\
& + \beta_{25} comR1 : ko\ comR2 : kn + \beta_{26} comR1 : ko\ comR3 : oe \\
& + \beta_{35} comR1 : oe\ comR2 : kn + \beta_{36} comR1 : oe\ comR3 : oe
\end{aligned}
\tag{2}
$$

For evaluation and prediction, the LM prediction output should be restored to original scale via **exp(y).**

Log transformation fits data better

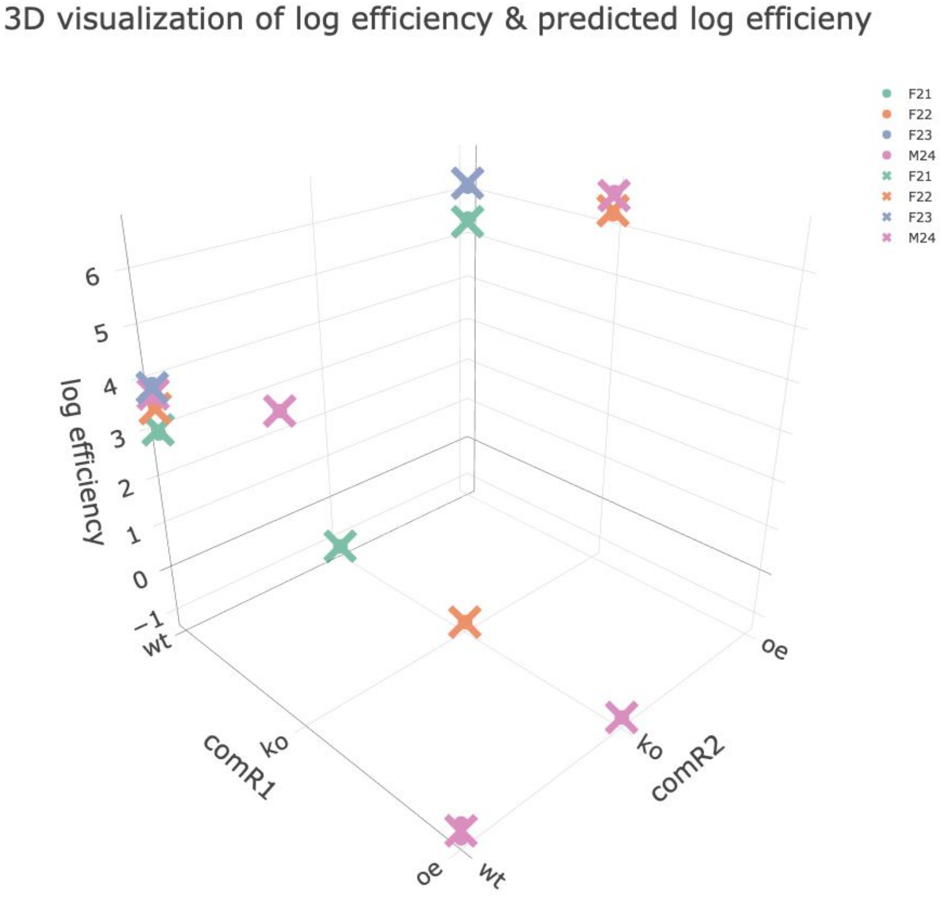
Main effect:

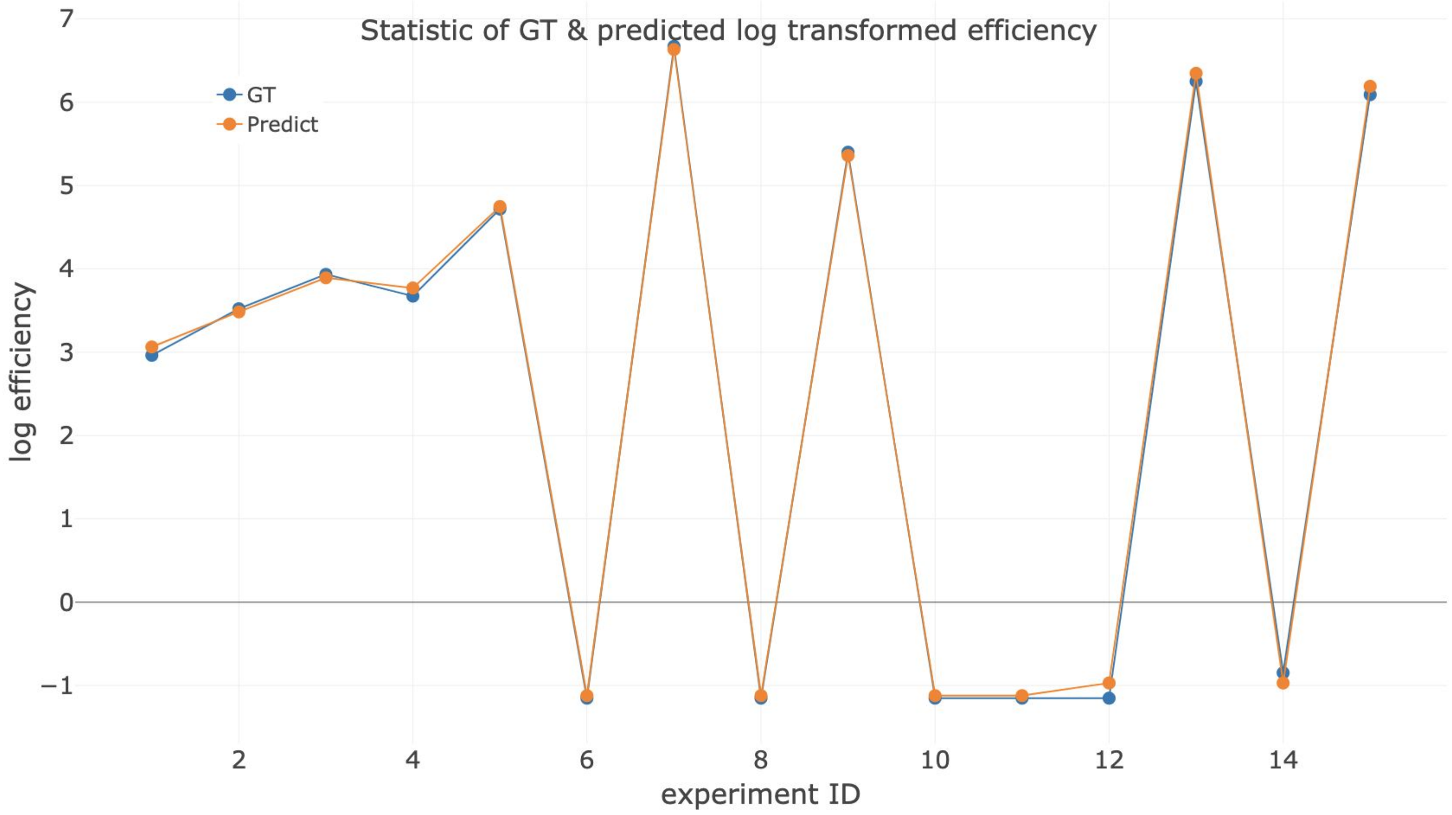
- Intercept representing wt genotype is 3.03, indicating the baseline transformation efficiency.
- *comR1*:ko increase log efficiency by 0.98, while *comR1*:oe decreases by 4.74, indicating *comR1* may function as a repressor.
- *comR2*:ko decreases log efficiency by 4.18, while *comR2*:oe increases by 2.30, indicating *comR2* may function as an activator.

Interaction effect:

- *comR1*:ko *comR2*:ko decreases log efficiency by 1.40.
- *comR1*:oe *comR2*:ko increases log efficiency by 4.03.
- Interestingly, *comR1*:ko *comR2*:oe should increase efficiency significantly if *comR1* is repressor and *comR2* is activator, but it doesn’t.

Ground-Truth vs Predicted efficiency with log LM

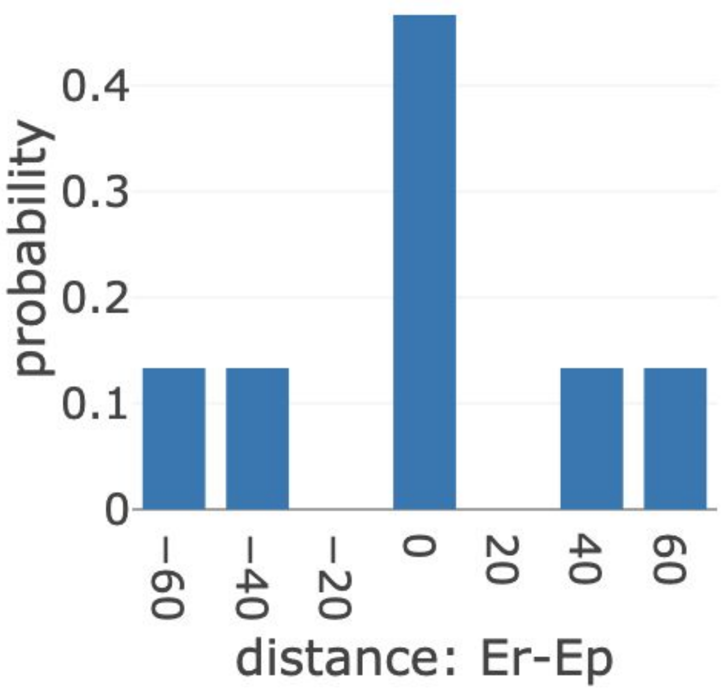




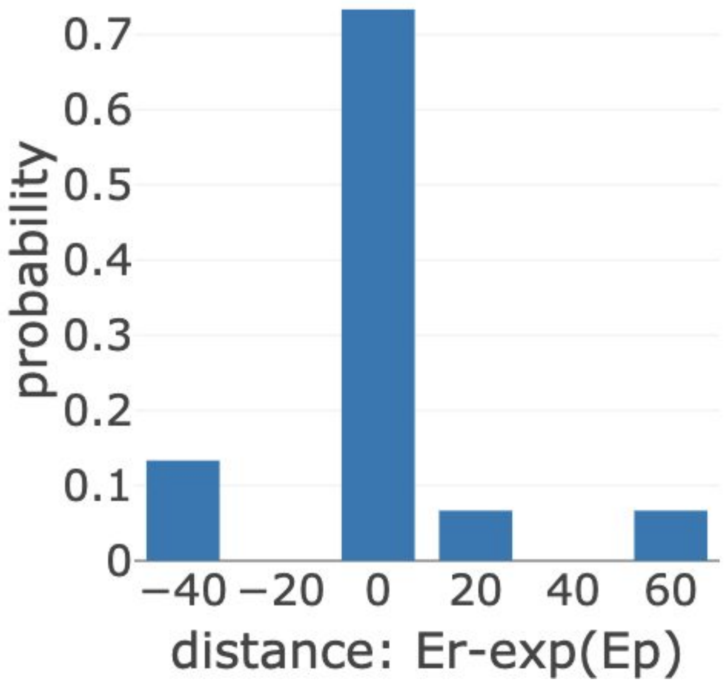
circle represents GT efficiency values; cross represents predicted efficiency value

GT vs predicted efficiency in 45 experiments

3. Log transformation improves model performance



Probability histogram of discrepancy between GT vs predicted values using LM (bin size=20).



Probability histogram of discrepancy between GT vs predicted values using log transformed LM (bin size=20)

The discrepancy between GT efficiency and predicted efficiency are more condensed within [-10, 10] by use of log transformation (0.71 vs 0.44). Their Root mean square deviations (RMSD, as equation below) are 42.01 vs 18.36.

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} \left(x_i - \hat{x}_i\right)^2}{N}}$$

4. Comparison of genotype oe vs wt

To test whether genotype oe has different level of natural competence compared with wt, contrast analysis is applied.

The null hypothesis is

$$H_0 : \beta_{oe} - \beta_{wt} = 0$$

comR1:

```
                      Estimate Std. Error   t value   Pr(>|t|)
comR1 c=( -1 0 1 ) -4.565685   1.532398 -2.979438 0.01149775
attr(,"class")
[1] "fit_contrast"
```

comR2:

```
                      Estimate Std. Error  t value    Pr(>|t|)
comR2 c=( -1 0 1 ) 3.701371   1.069244 3.461672 0.004702628
attr(,"class")
[1] "fit_contrast"
```

For both *comR1* and *comR2*, genotype oe has significantly different level of natural competence with wt. *comR1*:oe decreases efficiency while *comR2*:oe increases efficiency.

Conclusion

1. Logarithm transformation of efficiency could improve fitness of linear model in experiment data.

2. Both *comR1* and *comR2* has significant impact on natural competence of the bacterium. Also, these 2 genes have interaction effect.

3. For both *comR1* and *comR2*, genotype over-expression has significantly different level with widetype.

Discussion

- Effect of comR1:oe comR2:oe is unable to be estimated with given data. More experiments of this genotype should be conducted.

- The comR1 seems to have negative effect on natural competence, while comR2 seems to be positive. However, in this case, interactive comR1:oe comR:ko should be negative theoretically, whereas it is positive in our model. Deeper analytics should be employed to explore the relation of the 2 genes.

LM Results

```
Call:
lm(formula = efficiency ~ block + comR1 * comR2, data = ave_data)

Residuals:
        1          2          3          4          5          6          7 
 4.710e+01  6.633e+01 -4.710e+01 -6.633e+01 -1.243e-14 -5.329e-15  6.633e+01 
        8          9         10         11         12         13         14 
 2.132e-14 -4.710e+01  1.954e-14 -8.882e-15 -5.622e-02 -6.633e+01  5.622e-02 
       15 
 4.710e+01 

Coefficients: (1 not defined because of singularities)
               Estimate Std. Error t value Pr(>|t|)  
(Intercept)     -27.743     70.456  -0.394   0.7138  
blockF22         -4.743     99.639  -0.048   0.9643  
blockF23        125.965     81.355   1.548   0.1965  
blockM24        133.477     99.639   1.340   0.2514  
comR1ko           6.311    107.623   0.059   0.9561  
comR1oe        -105.362     90.958  -1.158   0.3112  
comR2ko          28.059    107.623   0.261   0.8072  
comR2oe         296.423     81.355   3.644   0.0219 *
comR1ko:comR2ko  -1.567    177.310  -0.009   0.9934  
comR1oe:comR2ko -28.115    134.912  -0.208   0.8451  
comR1ko:comR2oe 314.417    134.912   2.331   0.0802 .
comR1oe:comR2oe      NA         NA      NA       NA  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 81.36 on 4 degrees of freedom
Multiple R-squared:  0.9678,	Adjusted R-squared:  0.8874 
F-statistic: 12.03 on 10 and 4 DF,  p-value: 0.01424
```

LM fitting result in R using raw data

```
Call:
lm(formula = efficiency ~ block + comR1 * comR2, data = data2)

Residuals:
        1          2          3          4          5          6          7 
-7.030e-02  6.713e-02  7.030e-02 -6.713e-02  3.816e-17  6.349e-16  6.713e-02 
        8          9         10         11         12         13         14 
 2.429e-17  7.030e-02 -6.210e-16  6.939e-18 -1.521e-01 -6.713e-02  1.521e-01 
       15 
-7.030e-02 

Coefficients: (1 not defined because of singularities)
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)       3.0334     0.1255  24.161 1.74e-05 ***
blockF22          0.4214     0.1776   2.373  0.07655 .  
blockF23          0.8305     0.1450   5.729  0.00460 ** 
blockM24          0.7075     0.1776   3.985  0.01633 *  
comR1ko           0.9780     0.1918   5.100  0.00698 ** 
comR1oe          -4.7401     0.1621 -29.245 8.14e-06 ***
comR2ko          -4.1847     0.1918 -21.820 2.61e-05 ***
comR2oe           2.2971     0.1450  15.845 9.27e-05 ***
comR1ko:comR2ko  -1.3994     0.3160  -4.429  0.01143 *  
comR1oe:comR2ko   4.0325     0.2404  16.774 7.40e-05 ***
comR1ko:comR2oe  -0.4121     0.2404  -1.714  0.16164    
comR1oe:comR2oe       NA         NA      NA       NA    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.145 on 4 degrees of freedom
Multiple R-squared:  0.9994,	Adjusted R-squared:  0.9979 
F-statistic: 666.7 on 10 and 4 DF,  p-value: 5.385e-06
```

LM fitting result in R using log transformed data

Appendix: LM fitting code

```
library("tidyverse")
library("gdata")
library("Metrics")
library("gmodels")
library("plotly")

data <- read.csv("../comR12_tx_data.csv")
data$comR1 <- gdata::reorder.factor(data$comR1, new.order=c("wt", "ko", "oe"))
data$comR2 <- gdata::reorder.factor(data$comR2, new.order=c("wt", "ko", "oe"))
# average duplicates
obs <- length(data$comR1)
comR1 <- data$comR1[seq(1, obs, 3)]
comR2 <- data$comR2[seq(1, obs, 3)]
block <- data$block[seq(1, obs, 3)]
efficiency <- (data$efficiency[seq(1, obs, 3)] + data$efficiency[seq(2, obs, 3)] + data$efficiency[seq(3, obs, 3)])/3
ave_data <- data.frame(comR1, comR2, efficiency, block)
# fit raw data
model <- lm(formula = efficiency ~ block + comR1 * comR2, data=ave_data)
# predicted data
pred_input <- ave_data %>% select(comR1, comR2, block)
pred_input$efficiency <- predict.lm(model, newdata=pred_input)
# rmse
r1 <- rmse(ave_data$efficiency, pred_input$efficiency)
# log transformed
data2 <- ave_data
data2$efficiency <- log(data2$efficiency)
model2 <- lm(formula = efficiency ~ block + comR1 * comR2, data=data2)
pred_input_trans <- data2 %>% select(comR1, comR2, block)
pred_input_trans$efficiency <- predict.lm(model2, newdata=pred_input_trans)
# rmse
r2 <- rmse(ave_data$efficiency, exp(pred_input_trans$efficiency))
```

Visualization code

```r
# visualize 3D data and predicted data
p<-plot_ly(data=ave_data, x=~comR1, y=~comR2, z=~efficiency, color=~block, type="scatter3d", mode="markers", marker=list(symbol='circle', sizemode='diameter', width=1280, height=1280)) %>%
    add_trace(data=pred_input, x=~comR1, y=~comR2, z=~efficiency, color=~block, type="scatter3d", mode="markers", marker=list(symbol='x', sizemode='diameter')) %>%
    layout(showlegend=TRUE, legend=list(x=0.7, y=0.9, font=(size=24)),
    scene=list(
        xaxis=list(title="comR1", tickfont=list(size=20), titlefont=list(size=28)),
        yaxis=list(title="comR2", tickfont=list(size=20), titlefont=list(size=28)),
        zaxis=list(title="efficiency", tickfont=list(size=20), titlefont=list(size=28))
        ),
    title=list(text="3D visualization of efficiency & predicted efficieny", font=list(size=32), y=0.95))

# visualize distance in 2D scatter
runs <- 1:length(ave_data$efficiency)
p_diff <- plot_ly(x=runs, y=ave_data$efficiency, type="scatter", mode='lines+markers', marker=list(size=15), name="GT", width=1680, height=960) %>%
    add_trace(x=runs, y=pred_input$efficiency+3, type="scatter", mode="lines+markers", marker=list(size=15), name="Predict") %>%
    layout(title=list(text="Statistic of GT & predicted efficiency", font=list(size=32), y=0.95)) %>%
    layout(xaxis=list(title="experiment ID", tickfont=list(size=28), titlefont=list(size=32)),
           yaxis=list(title="efficiency", tickfont=list(size=28), titlefont=list(size=32)),
           showlegend=TRUE,
           legend=list(x=0.9, y=0.9, font=list(size=24)))
        #    margin=list(pad=50, b=10, l=50,r=50))
# statistic distance in histogram
abs_dist <- ave_data$efficiency - pred_input$efficiency
p_hist <- plot_ly(x=abs_dist, type="histogram", histnorm="probability", width=480, height=480, xbins=list(size=20)) %>%
    layout(
        #title=list(text="Distribution of difference between GT efficiency & predicted efficiency", font=list(size=32), y=0.98),
    xaxis=list(title="distance: Er-Ep", tickfont=list(size=28), titlefont=list(size=32), dtick=20),
    yaxis=list(title="probability", tickfont=list(size=28), titlefont=list(size=32)),
    bargap=0.25)
```