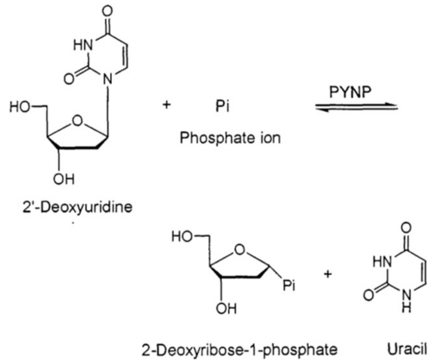
DEOXYURIDINE METABOLISM: URACIL PRODUCTION SPEED

INTRODUCTION

In an experiment, we constructed samples with differing concentrations of deoxyuridine in addition to a constant amount of enzyme ($E_0 = 0.1\ \mu M$) and the appropriate buffers. After a set period, we stopped the reaction (with sodium hydroxide in this case) and measured the amount of uracil produced via spectrophotometry.

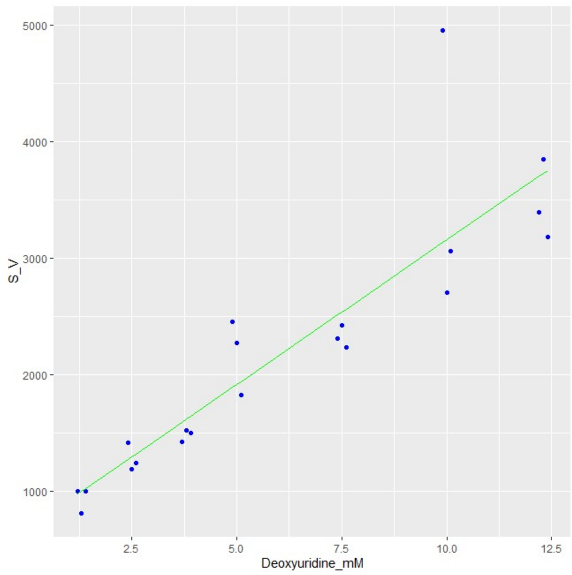


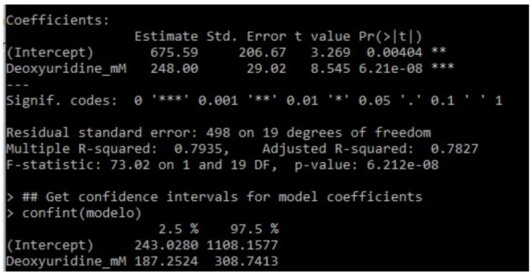
RESULTS AND CONCLUSIONS

Applying the Michaelis-Menten model, and, using linear models in R, inference of the Michaelis-Menten constant K_m for the system, as well as the maximum uracil production Vmax.

The starting point of the procedure is the Michaelis-Menten equation:

$$\frac{S}{V(uracil)} = \frac{K_m}{Vmax} + \frac{S}{Vmax}$$

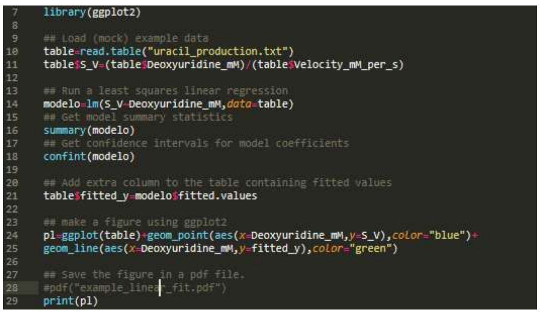




```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)       675.59     206.67   3.269  0.00404 **
Deoxyuridine_mM   248.00      29.02   8.545 6.21e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 498 on 19 degrees of freedom
Multiple R-squared:  0.7935,    Adjusted R-squared:  0.7827
F-statistic: 73.02 on 1 and 19 DF,  p-value: 6.212e-08

> ## Get confidence intervals for model coefficients
> confint(modelo)
                   2.5 %    97.5 %
(Intercept)     243.0280 1108.1577
Deoxyuridine_mM 187.2524  308.7413
```



```
library(ggplot2)

## Load (mock) example data
table=read.table("uracil_production.txt")
table$S_V=(table$Deoxyuridine_mM)/(table$Velocity_mM_per_s)

## Run a least squares linear regression
modelo=lm(S_V~Deoxyuridine_mM,data=table)
## Get model summary statistics
summary(modelo)
## Get confidence intervals for model coefficients
confint(modelo)

## Add extra column to the table containing fitted values
table$fitted_y=modelo$fitted.values

## make a figure using ggplot2
pl=ggplot(table)+geom_point(aes(x=Deoxyuridine_mM,y=S_V),color="blue")+
geom_line(aes(x=Deoxyuridine_mM,y=fitted_y),color="green")

## Save the figure in a pdf file.
#pdf("example_linear_fit.pdf")
print(pl)
```

$$Slope\ =\ 248.0\ s^{-1} \rightarrow Vmax = Slope^{-1} = 0{,}00403225\ s$$

$$indep.term =\ 675.6\ s^{-1} \rightarrow K_M = i.term * Vmax = 0.00403225 * 675.6 = 2.72\ mM$$

Value of the catalysis constant k_{cat}:

$$Vmax = k_{cat} * E_0 \rightarrow k_{cat} = \frac{0{,}00403225\ s}{0{,}0001\ M} = 40{,}32\ s/mM$$

For the case in which each molecule of complex has exactly the same probability of disintegrating originating a molecule of product or doing so giving back one molecule of substrate, values of k_f and k_r.

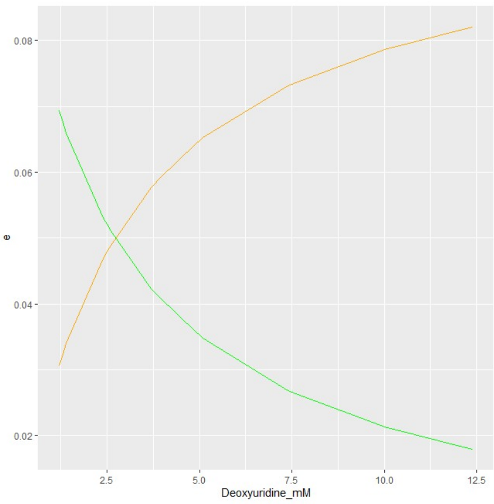
We assume k_r=k_f, then if:

$$K_M = \frac{(k_r + k_{cat})}{k_f} \rightarrow K_M = \frac{(k_f + k_{cat})}{k_f} \rightarrow k_f = \frac{k_{cat}}{K_M + 1} = \frac{40{,}32 \frac{s}{M}}{2.72M + 1} = 10.84\ mM$$

$$= k_r$$

Graphical representation of the predicted concentrations of complex and enzyme as a function of the remaining substrate: C(t) and E(t) in the range of deoxyuridine concentrations present in the input data.

$$E(t) = \frac{K_M E_0}{(K_M + S)};\ C(t) = (E_0 * S)/(K_M + S)$$




```
library(ggplot2)

## Load (mock) example data
table=read.table("uracil_production.txt")
table$S_V=(table$Deoxyuridine_mM)/(table$Velocity_mM_per_s)

## Run a least squares linear regression
modelo=lm(S_V~Deoxyuridine_mM,data=table)
## Get model summary statistics
summary(modelo)
## Get confidence intervals for model coefficients
confint(modelo)

## Add extra column to the table containing fitted values
table$fitted_y=modelo$fitted.values

## make a figure using ggplot2
pl=ggplot(table)+geom_point(aes(x=Deoxyuridine_mM,y=S_V),color="blue")+
geom_line(aes(x=Deoxyuridine_mM,y=fitted_y),color="green")

## Save the figure in a pdf file.
#pdf("example_linear_fit.pdf")
print(pl)
#dev.off()

table$c=(table$Deoxyuridine_mM*0.1)/(table$Deoxyuridine_mM+2.720)
table$e=(2.720*0.1)/(table$Deoxyuridine_mM+2.720)
#data=rbind(d,e,c)
print(table)


pl2=ggplot(table)+
geom_line(aes(x=Deoxyuridine_mM,y=e),color="green")+
geom_line(aes(x=Deoxyuridine_mM,y=c),color="orange")


print(pl2)
```

Algebraic representation of each of them:

$$N1 = \begin{pmatrix} -1 \\ -1 \\ 1 \\ 2 \end{pmatrix}$$

$$N2 = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix}$$

$$N3 = \begin{pmatrix} 1 & -1 & -1 \end{pmatrix}$$

$$N4 = \begin{pmatrix} 1 & -1 & 0 & -1 \\ 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$N5 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{pmatrix}$$

System 1:

$$\frac{dS1}{dt} = -V1; \frac{dS2}{dt} = -V1; \frac{dS3}{dt} = V1; \frac{dS4}{dt} = 2V1$$

System 2:

$$\frac{dS1}{dt} = V1 - V2; \frac{dS2}{dt} = V3 - V2; \frac{dS3}{dt} = V3 - V4; \frac{dS4}{dt} = V4 - V5$$

System 3:

$$\frac{dS1}{dt} = V1 - V2 - V3$$

System 4:

$$\frac{dS1}{dt} = V1 - V2 - V4; \frac{dS2}{dt} = 2V3 - V2; \frac{dS3}{dt} = V4$$

System 5:

$$\frac{dS1}{dt} = V1; \frac{dS2}{dt} = V3 - V2; \frac{dS3}{dt} = V2 - V3$$

Kernel solutions

System 1:

$$0 = -V1; 0 = -V1; 0 = V1; \ 0 = 2V1$$

System 2:

$$0 = V1 - V2; \ 0 = V3 - V2; \ 0 = V3 - V4; \ 0 = V4 - V5$$

$$0 = V1 = V2 = V3 = V4 = V5$$

System 3:

$$0 = V1 - V2 - V3$$

System 4:

$$0 = V1 - V2 - V4; \ 0 = 2V3 - V2; \ 0 = V4$$

$$V1 = V2; \ V3 = V2/2; \ 0 = V4$$

System 5:

$$0 = V1; \ 0 = V3 - V2; \ 0 = V2 - V3$$

$$0 = V1; V3 = V2$$

The configuration, for all the velocities, associated to the optimization of the objective function.

$$N = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

```r
N = matrix(rep(0,45), nrow=9, ncol=5)

N[1 ,]=c(1 ,0 ,0 ,0 , -1 ,0 ,0 ,0 ,0 ,0)
N[2 ,]=c(0 ,1 ,0 ,0 , -1 , -1 ,0 ,0 ,0 ,0)
N[3 ,]=c(0 ,0 ,1 ,0 ,0 , -2 ,0 ,0 ,0 ,0)
N[4 ,]=c(0 ,0 ,0 ,1 ,0 , -1 ,0 ,0 ,0 ,0)
N[5 ,]=c(0 ,0 ,0 ,0 ,1 ,0 ,0 , -1 ,0 ,0)
N[6 ,]=c(0 ,0 ,0 ,0 ,1 ,0 , -4 ,0 ,0 ,0)
N[7 ,]=c(0 ,0 ,0 ,0 ,0 ,1 , -1 ,0 ,0 ,0)
N[8 ,]=c(0 ,0 ,0 ,0 ,0 ,1 ,0 ,0 , -1 ,0)
N[9 ,]=c(0 ,0 ,0 ,0 ,0 ,0 ,1 ,0 ,0 , -1)
N[10 ,]=c(0 ,1 ,1 ,0 ,0 ,0 , 0 ,0 ,0 ,0)
N[11 ,]=c(1 ,0 ,0 ,0 ,0 ,0 , 0 ,0 ,0 ,0)
N[11 ,]=c(0 ,1 ,0 ,0 ,0 ,0 , 0 ,0 ,0 ,0)
N[13 ,]=c(0 ,0 ,1 ,0 ,0 ,0 , 0 ,0 ,0 ,0)
N[14 ,]=c(0 ,0 ,0 ,1 ,0 ,0 , 0 ,0 ,0 ,0)
N[15 ,]=c(0 ,0 ,0 ,0 ,1 ,0 , 0 ,0 ,0 ,0)
N[16 ,]=c(0 ,0 ,0 ,0 ,0 ,1 , 0 ,0 ,0 ,0)
N[17 ,]=c(0 ,0 ,0 ,0 ,0 ,0 , 1 ,0 ,0 ,0)
N[18 ,]=c(0 ,0 ,0 ,0 ,0 ,0 , 0 ,1 ,0 ,0)
N[19 ,]=c(0 ,0 ,0 ,0 ,0 ,0 , 0 ,0 ,1 ,0)
N[20 ,]=c(0 ,0 ,0 ,0 ,0 ,0 , 0 ,0 ,0 ,1)

constraints_matrix=N
constraints_direction=c("=","=","=","=","=","=","=","=","=","<=",">=",">=",">=",">=",
">="," >=",">=",">=",">=",">=")
constraints_values=c(0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,56 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0)

objective=c(0,0,0,0,0,0,0,0,0,0)



optimum <-  lp(direction="max",
               objective.in = objective,
               const.mat = constraints_matrix,
               const.dir = constraints_direction,
               const.rhs = constraints_values)

optimum$objval
optimum$solution

table=data.frame(constraints_matrix)
table$signo=constraints_direction
table$res=constraints_values
```

```
> optimum$objval
[1] 8
> optimum$solution
 [1] 32 40 16  8 32  8  8 32  8  8
```

The output indicates the respective optimal velocities for maximum V10 which is 8 mM/s.