Metabolic Modelling

Tianyi Shi

2020-11-26

Contents

- **Model 112 was used.**

1 The Haldane Relationship

Using the "Parameter Scan" utility in Copasi (Figure 1), I collected steady state concentrations of 'A' and 'B' in $5^4 = 625$ combinations of $V_{max(f)}$, $K_{m(p)}$, $V_{max(r)}$, and $K_{m(s)}$, where each parameter vary from 1×10^{-2} to 1×10^2 in a logarithmic scale.
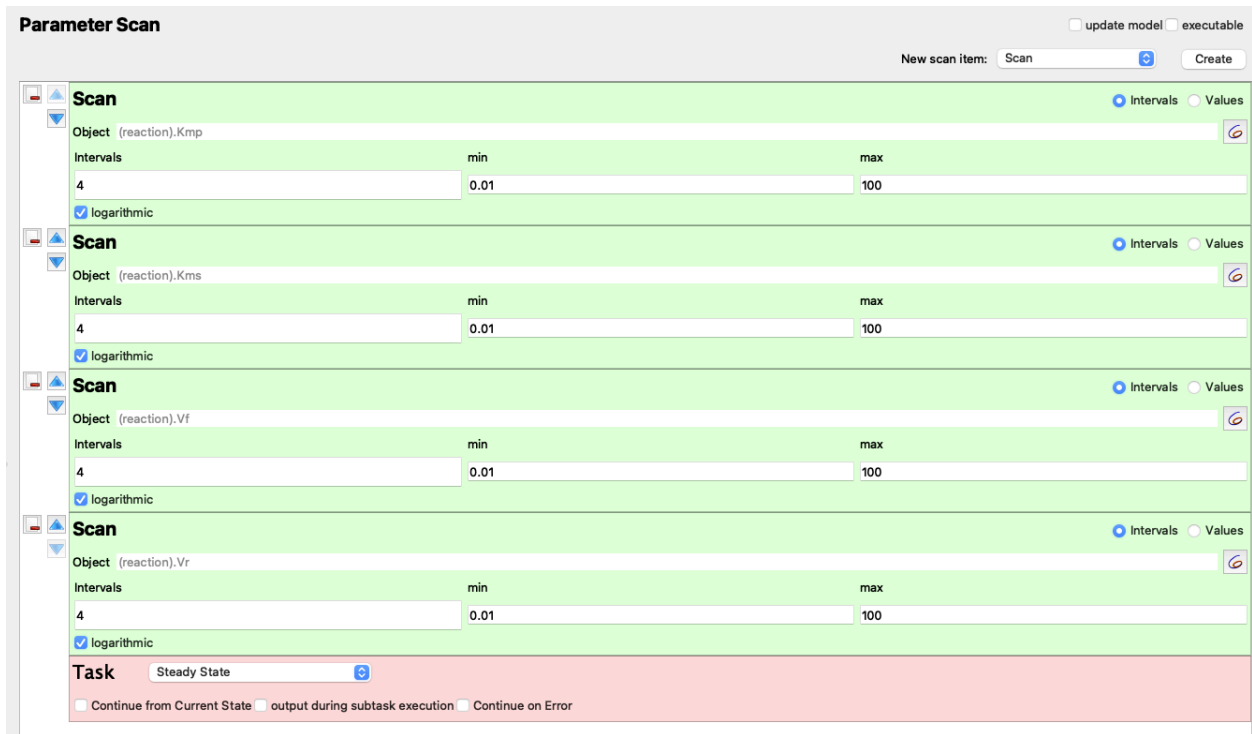

Figure 1: "Parameter Scan" setup

Here I read the data, and add two columns, where `keq` represents the experimental K_{eq}:

$$K_{eq} = \frac{[B]}{[A]}$$

and `keq_calc` represents the K_{eq} calculated from the Haldane relationship:

$$K_{eq} = \frac{V_{max(f)} \cdot K_{m(p)}}{V_{max(r)} \cdot K_{m(s)}}$$

```
q1 <- read_tsv('metabolic modelling/q1.txt', col_names = c('vf', 'kmp', 'vr', 'kms', 'A', 'B'))
q1 <- q1 %>% mutate(
  keq = B/A,
  keq_calc = (vf * kmp) / (vr * kms),
  keq_diff = keq - keq_calc
)
q1
```

```
## # A tibble: 625 x 9
##        vf   kmp     vr   kms     A       B     keq keq_calc
##     <dbl> <dbl>  <dbl> <dbl> <dbl>   <dbl>   <dbl>    <dbl>
## 1   0.01  0.01   0.01  0.01 1     1.00e+0 1.00e+0  1
## 2   0.01  0.01   0.1   0.01 1.82  1.82e-1 1.00e-1  0.1
## 3   0.01  0.01   1     0.01 1.98  1.98e-2 1.00e-2  0.01
## 4   0.01  0.01  10     0.01 2.00  2.00e-3 1.00e-3  0.001
## 5   0.01  0.01 100     0.01 2.00  2.00e-4 1.00e-4  0.0001
## 6   0.1   0.01   0.01  0.01 0.182 1.82e+0 1.00e+1 10
## 7   0.1   0.01   0.1   0.01 1     1.00e+0 1.00e+0  1
## 8   0.1   0.01   1     0.01 1.82  1.82e-1 1.00e-1  0.1
## 9   0.1   0.01  10     0.01 1.98  1.98e-2 1.00e-2  0.01
## 10  0.1   0.01 100     0.01 2.00  2.00e-3 1.00e-3  0.001
## # ... with 615 more rows, and 1 more variable: keq_diff <dbl>
```

Plotting `keq_calc` against `keq` shows that K_{eq} calculated in these two ways are equal, as the linear regression line has a gradient of 1 and passes through the origin.

```
q1 %>% ggplot(aes(keq, keq_calc)) +
  geom_point() +
  geom_smooth()+
  scale_x_log10()+
  scale_y_log10()
```
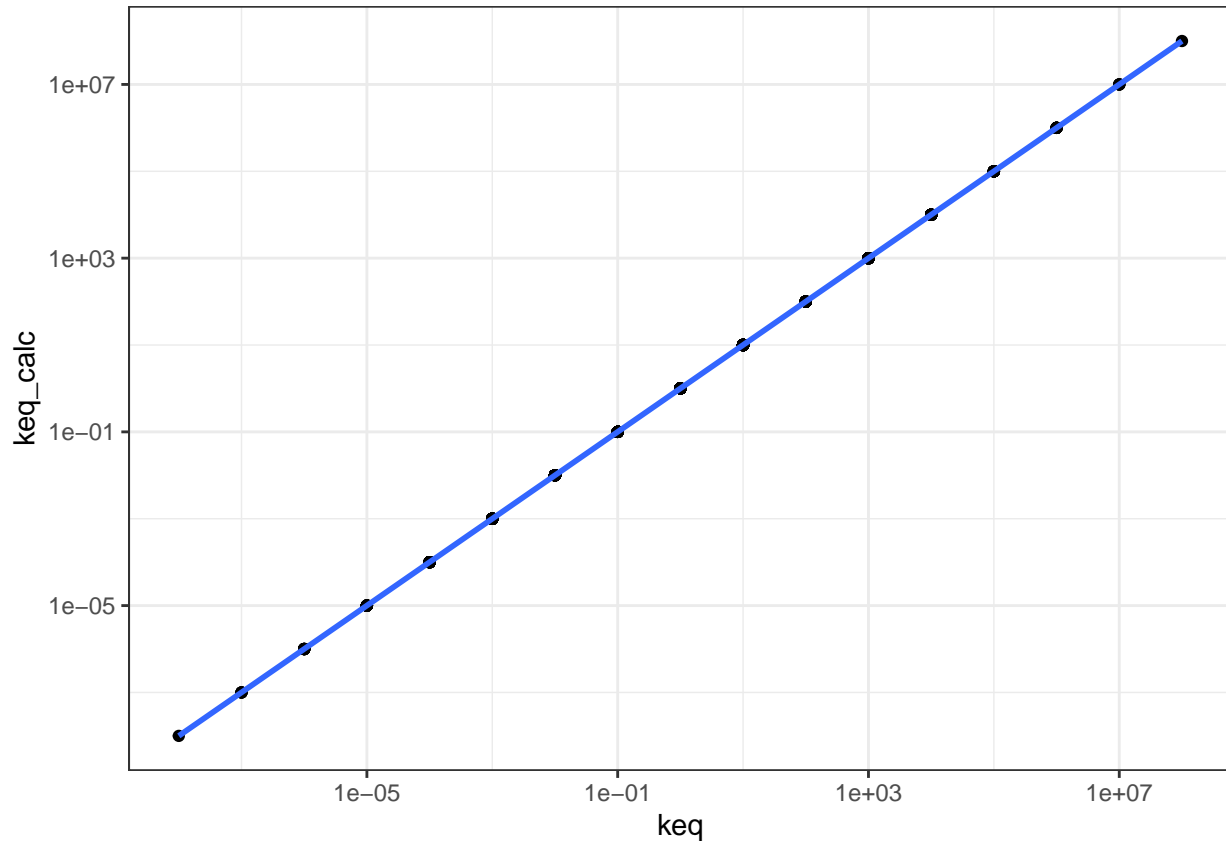
```r
mod <- lm(q1$keq ~ q1$keq_calc)
summary(mod)
```

```
##
## Call:
## lm(formula = q1$keq ~ q1$keq_calc)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.148e-08 -4.480e-10 -4.480e-10 -4.190e-10  2.880e-07
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept) 7.078e-10  4.630e-10 1.529e+00    0.127
## q1$keq_calc 1.000e+00  1.135e-16 8.814e+15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.156e-08 on 623 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 7.769e+31 on 1 and 623 DF,  p-value: < 2.2e-16
```

The Haldane relationship shows that K_{eq} is proportional to $V_{max(f)}$ and $K_{m(p)}$, and inversely proportional to $V_{max(r)}$ and $K_{m(s)}$, and this can be illustrated using a facetted heatmap:

```r
q1 %>% ggplot(aes(x = vf, y = kmp, fill = log10(keq))) +
  geom_tile()+
  facet_grid((1/vr)~(1/kms), as.table = FALSE, labeller = label_both)+
```

```
  scale_x_log10(labels = plain, expand = c(0, 0))+
  scale_y_log10(labels = plain, expand = c(0, 0))+
  scale_fill_gradient2(low = 'blue', mid = 'yellow', high = 'red')+
  labs(title = "Variation of Keq with vf, kmp, vr, and kms")+
  theme(axis.text.x = element_text(angle = 90))
```
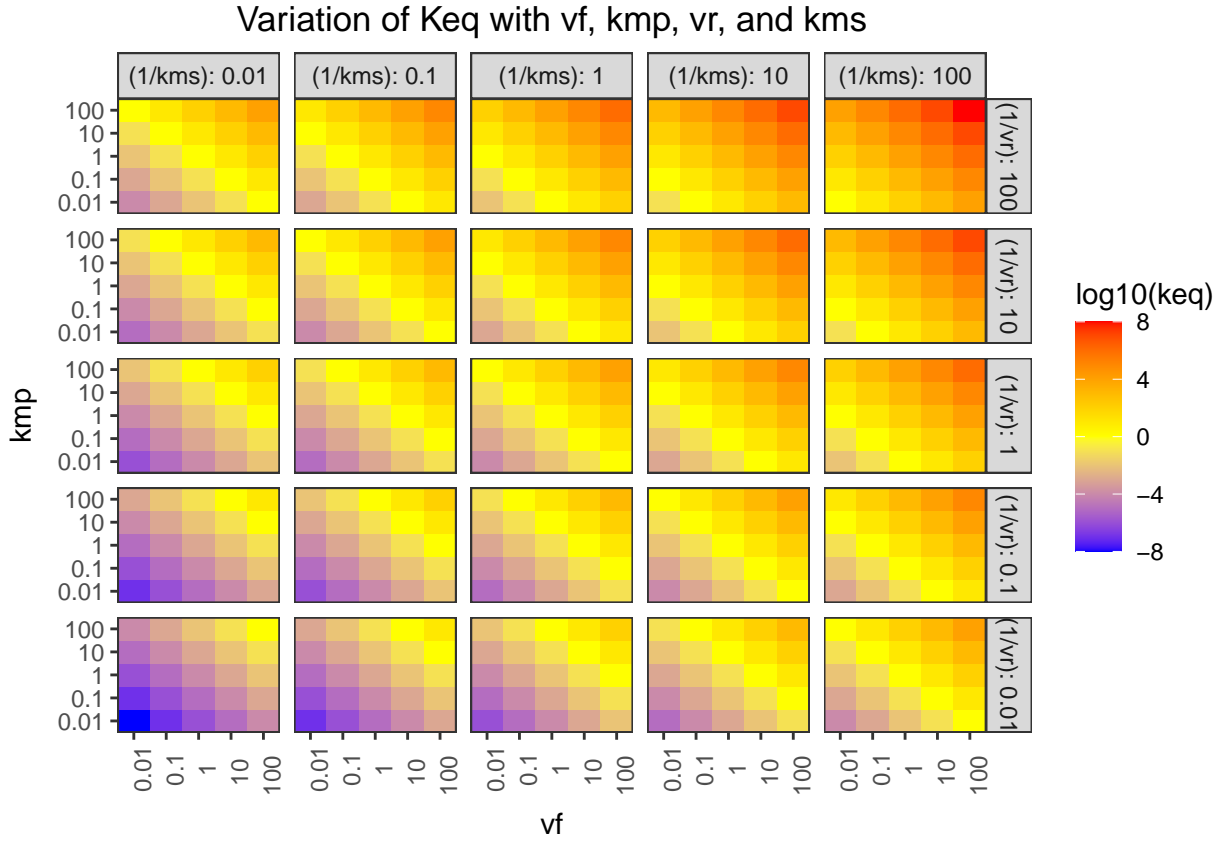

Figure 2: Variation of K_{eq} with $V_{max(f)}$, $K_{m(p)}$, $V_{max(r)}$, and $K_{m(s)}$

The pattern in each grid shows that K_{eq} is proportional to $V_{max(f)}$ and $K_{m(p)}$, and the pattern across the grids shows that it is inversely proportional to $V_{max(r)}$ and $K_{m(s)}$.

2 Control Points in A Simple Linear Pathway

I use the "Parameter Scan" function to vary the K_{eq} of reaction 3 (that catalyses the reversible conversion between C and D) in the range 10^{-6} to 10^{6} in a logarithmic scale (13 samples). The variables being recorded are K_{eq} and J (flux control coefficient) of reaction 3, and the concentrations of C and D. Then, the reaction quotient, Q (a.k.a. mass action ratio), of each row is calculated as:

$$Q = \frac{[\mathrm{D}]}{[\mathrm{C}]}$$

```
q2_scan_r3 <- read_tsv('metabolic modelling/Q2.txt')
q2_scan_r3 <- q2_scan_r3 %>% mutate(q = d/c)
q2_scan_r3
```

```
## # A tibble: 13 x 5
```

```
##             j      keq     c          d           q
##         <dbl>    <dbl> <dbl>      <dbl>      <dbl>
##  1 0.0000489 1.00e-6 100.   0.000100 0.00000100
##  2 0.0000489 1.00e-5 100    0.001     0.00001
##  3 0.00470   1.00e-4  99.1  0.00987   0.0000996
##  4 0.0353    1.00e-3  92.2  0.0885    0.000960
##  5 0.119     1.00e-2  69.0  0.556     0.00806
##  6 0.210     1.00e-1  44.5  2.35      0.0528
##  7 0.309     1.00e+0  32.0  6.77      0.212
##  8 0.369     1.00e+1  28.6 10.0       0.350
##  9 0.380     1.00e+2  28.2 10.6       0.378
## 10 0.381     1.00e+3  28.1 10.7       0.381
## 11 0.381     1.00e+4  28.1 10.7       0.381
## 12 0.381     1.00e+5  28.1 10.7       0.381
## 13 0.381     1.00e+6  28.1 10.7       0.381
```

Figure 3 shows the variation of the flux control coefficient, J, of reaction 3, with its K_{eq}. The plot shows that a high K_{eq}, i.e. high irreversibility, is correlated with a high J, and in the intermediate range J varies linearly with $\ln(K_{eq})$, i.e. varies linearly with $\Delta G = -RT \ln(K_{eq})$

```
q2_scan_r3 %>% ggplot(aes(log(keq), j)) +
  geom_point()
```
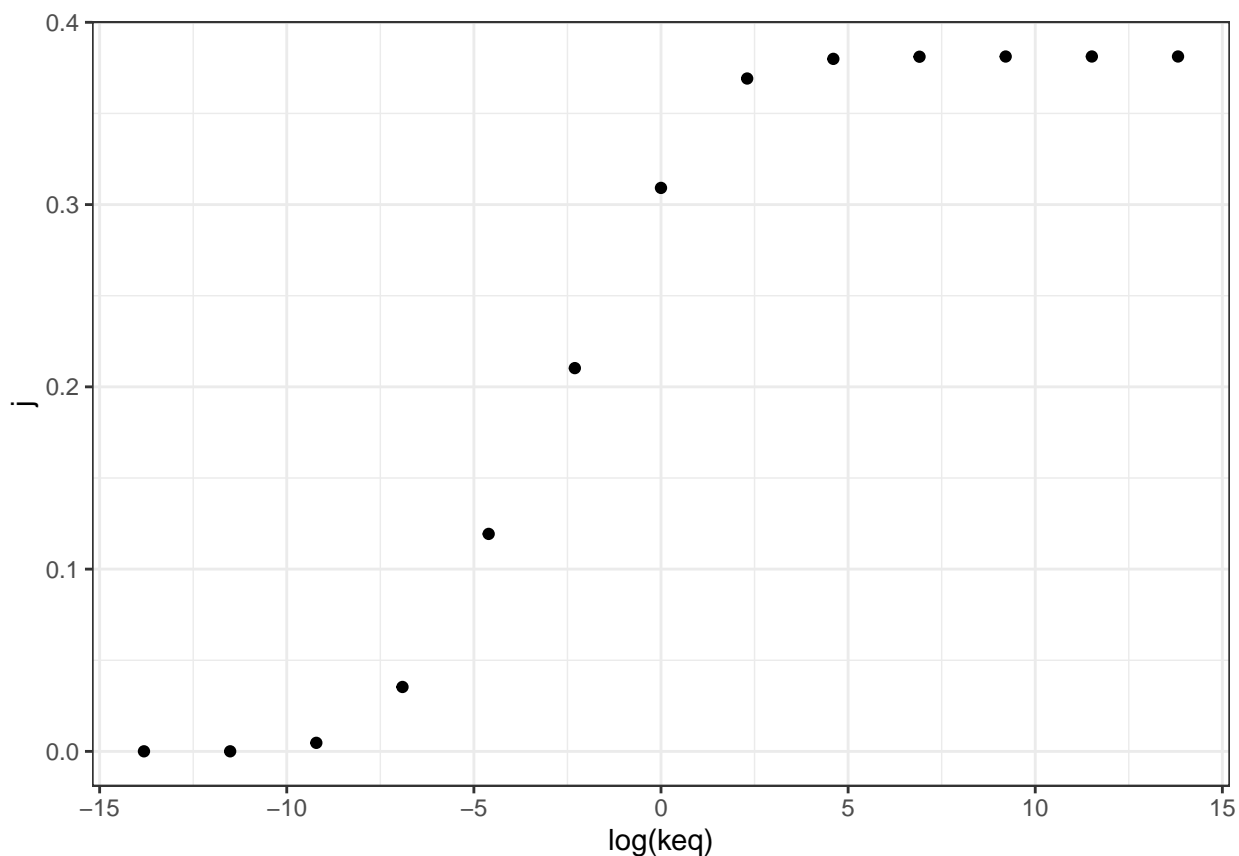

Figure 3: Variation of the flux control coefficient, J, of reaction 3, with its K_{eq}

Figure 4 shows the variation of the flux control coefficient, J, of reaction 3 with Q/K_{eq}, which is a measure of displacement of the reaction from the equilibrium. A Q/K_{eq} close to 1 indicates the reaction is close to

equilibrium. The plot shows that when reaction is further displaced from the equilibrium, the its flux control coefficient is higher.

```
q2_scan_r3 %>% ggplot(aes(q/keq, j)) +
  geom_point()+
  geom_smooth(method = 'lm', size = 0.5)
```
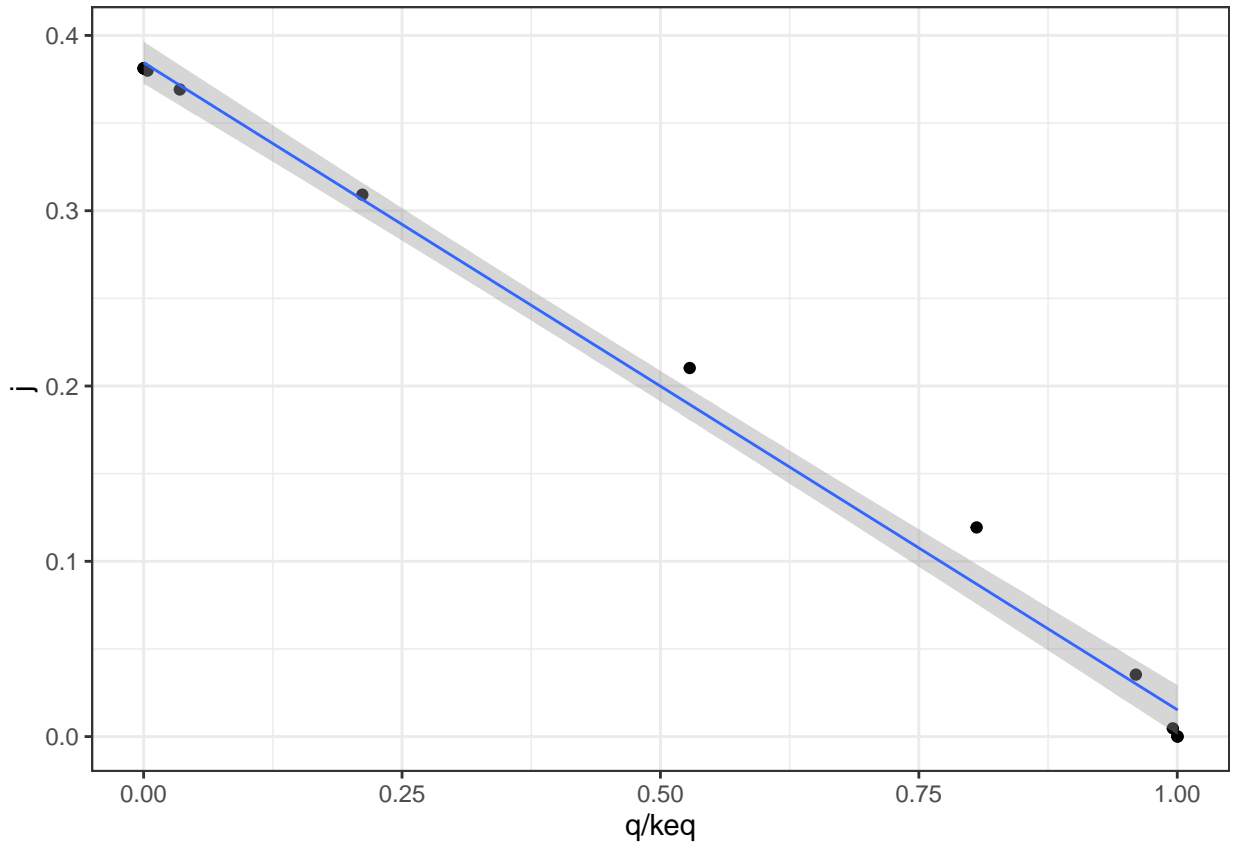

Figure 4: Variation of the flux control coefficient, J, of reaction 3, with Q/K_{eq}

A linear regression analysis shows that there is a strong linear correlation between J and Q/K_{eq}, with $p = 4.02 \times 10^{-6} < 10^{-5}$

```
mod <- with(q2_scan_r3, lm(j ~ q/keq))
summary(mod)
```

```
##
## Call:
## lm(formula = j ~ q/keq)
##
## Residuals:
##       Min       1Q    Median       3Q      Max
## -0.054438 -0.019991 -0.012931  0.001452  0.108459
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.448e-02  2.221e-02   2.453   0.0341 *
## q            8.960e-01  9.010e-02   9.945 1.67e-06 ***
## q:keq       -4.196e-08  1.569e-07  -0.267   0.7946
```

6

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05374 on 10 degrees of freedom
## Multiple R-squared:  0.9167, Adjusted R-squared:    0.9
## F-statistic:    55 on 2 and 10 DF,  p-value: 4.02e-06
```

In order to vary Q/K_{eq} without directly varying K_{eq}, and to see the effect not only on reaction 3 but also on all other reactions, I vary V_f, not K_{eq}, of reaction 3 from 10^{-6} to 10^6 with 1000 intervals, and collected the flux control coefficients of all 6 reactions, the K_{eq} of reaction 3, as well as concentrations of all species, which are then used to calculate the mass action ratio of each reaction. Then, the displacement of each reaction is calculated. The resulting dataframe contains three columns: 1) the reaction number; 2) the flux control coefficient and 3) displacement from equilibrium of this reaction

```r
q2_scan_r3_vf <- read_tsv('metabolic modelling/Q2-1.txt')
q2_scan_r3_vf <- q2_scan_r3_vf %>%
  mutate(
    d1 = (B/A) / 10,
    d2 = (C/B) / 1,
    d3 = (D/C) / 0.5,
    d4 = (E/D) / 5,
    d5 = (F/E) / 2,
    d6 = (G/F) / 10,
    c1 = A + B,
    c2 = B + C,
    c3 = C + D,
    c4 = D + E,
    c5 = E + F,
    c6 = F + G,
  ) %>% select(!(1:7))
q2_scan_r3_vf_cleaned <- tibble(
  reaction = integer(),
  displacement = double(),
  j = double()
  )
for (i in 1:6) {
  q2_scan_r3_vf_cleaned <- add_row(
    q2_scan_r3_vf_cleaned,
    reaction = i, j = q2_scan_r3_vf[[paste0('j', i)]], displacement = q2_scan_r3_vf[[paste0('d', i)]])
}
q2_scan_r3_vf_cleaned
```

```
## # A tibble: 6,006 x 3
##    reaction displacement            j
##       <int>        <dbl>        <dbl>
## 1        1         1.00 0.0000000343
## 2        1         1.00 0.0000000353
## 3        1         1.00 0.0000000363
## 4        1         1.00 0.0000000373
## 5        1         1.00 0.0000000383
## 6        1         1.00 0.0000000394
## 7        1         1.00 0.0000000405
## 8        1         1.00 0.0000000416
## 9        1         1.00 0.0000000428
## 10       1         1.00 0.0000000440
```

```
## # ... with 5,996 more rows
```

Then, for all reactions, J versus displacement from equilibrium is plotted:

```
q2_scan_r3_vf_cleaned %>% ggplot(aes(displacement, j)) +
  geom_point()+
  facet_wrap(~reaction, scales = 'free', labeller = label_both)+
  xlab("MAR/Keq")
```
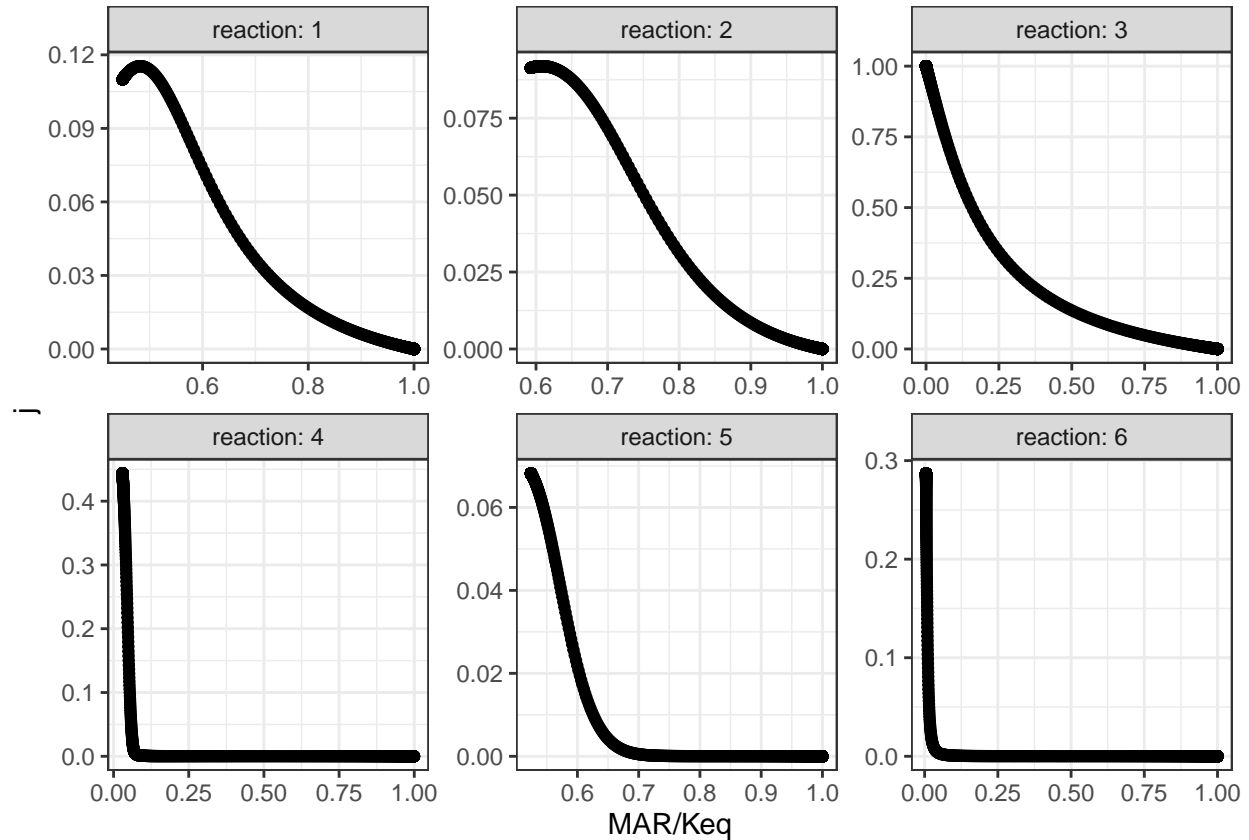

The plots shows that, when disturbing $V_{max(f)}$ of reaction 3, the disequilibrium ratio of all reactions are also altered. Not only reaction 3 but also all other reactions follow the rule that, the flux control coefficent of a reaction increases with the extent of displacement from the equilibrium. However, it is hard to predict J given only the value of Q/K_{eq}, as different reactions show different patterns of variations, so Q/K_{eq} (displacement from equilibrium) is not a robust indicator of J.

3 Linear Pathway with Negative Feedback

I chose reaction 2 to be the one sensitive to the inhibitor, and varied K_i from 10^{-18} to 10^{18} with 1000 intervals, and recorded it along with flux control coefficients of all reactions. The results are shown in Figure 5.

```
q3 <- read_tsv('metabolic modelling/Q3.txt')
q3 <- q3 %>% gather(reaction, "j", -ki)
q3 %>% ggplot(aes(ki, j)) +
  geom_point()+
  facet_wrap(~reaction, labeller = label_both)+
  scale_x_log10()
```
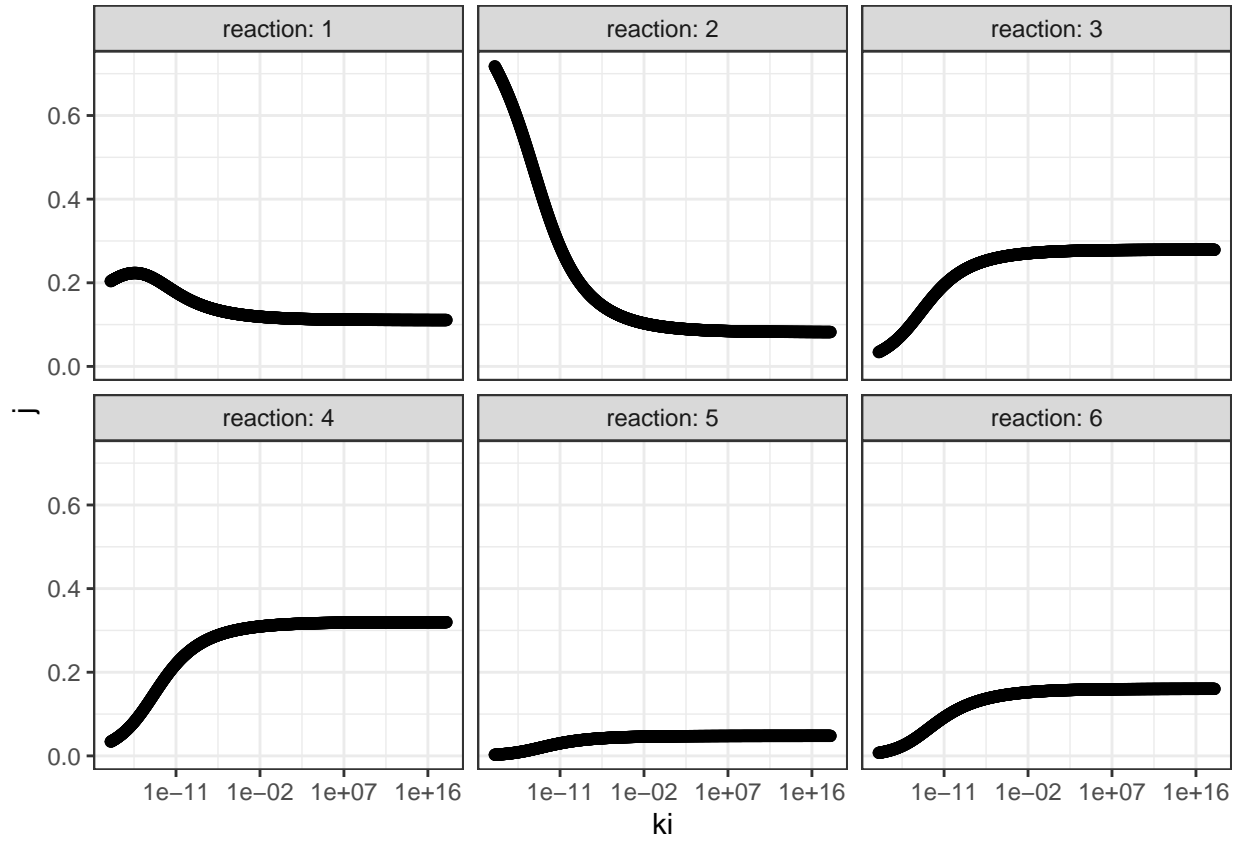
Figure 5: Effect of changing K_i of reaction 2 on the flux control coefficient of all reactions. Lower K_i means higher binding affinity to the inhibitor.

The plots show that, the flux control coefficient (J) of reaction 2 increases as the binding affinity (i.e. sensitivity) to the inhibitor of the enzyme involved in this step increases (i.e. as K_i decreases). J of the upsteam reaction 1 also inceases slightly. For all downstream reactions, J decreases.

3.1 Re-analyse the effect of K_i with constant flux

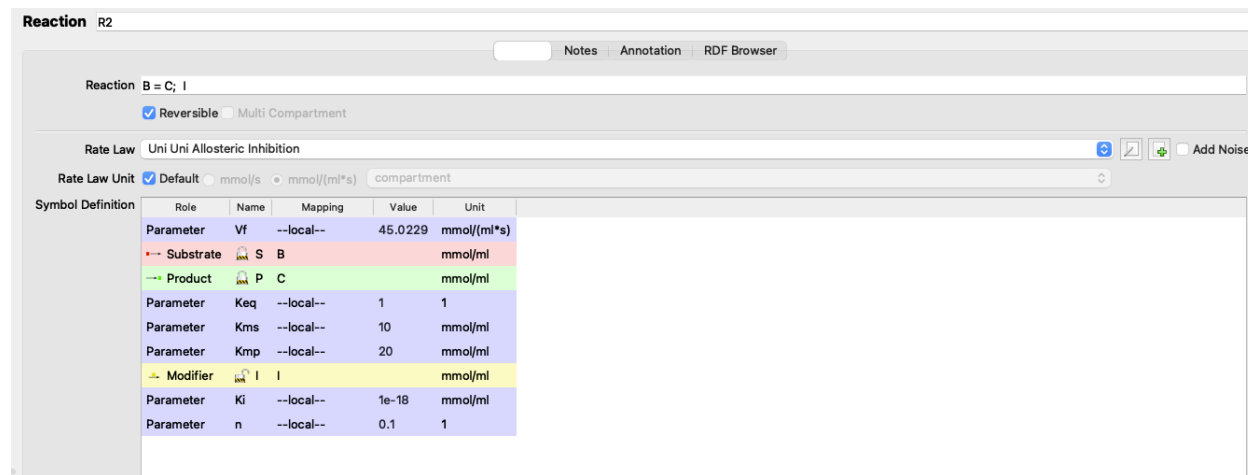The flux and values of J when $K_i = 10^{-18}$ and when $K_i = 10^18$ are shown below:

```
ki_low = 1e-18
flux_low = 0.476572
j_low = c(0.203839, 0.717872,   0.0342001,  0.0340061,  0.00278022, 0.007302)
ki_high = 1e18
flux_high = 1.07393
j_high = c(0.110937,    0.0822319,  0.278976,   0.319753,   0.0478304,  0.160271)
```

V_f of reaction 2 are optimised so that the flux when $K_i = 10^{-18}$ is 1.07393 (the same as when $K_i = 10^{18}$)

```
Optimization Result:

    Objective Function Value:   1.07393
    Function Evaluations:    248
    CPU Time [s]:   0.061
    Evaluations/Second [1/s]:   4065.57


    (R2).Vf: 45.0229
```


Figure 6: Adjusting Vf

I adjusted the V_f of reaction 2 from 5 to 45.0229 (6), verified that the flux is 1.07393 (the same as in the state with negligible inhibition), and the values of J are:

```
j_low <-    c(0.144048, 0.0858949,  0.266261,   0.30518,    0.0456503,  0.152966)
```

which can be directly compared to the J values in the uninhibited state:

```
tibble(
  ki =c(rep("low (1e-18)", 6), rep("high (1e18)", 6)),
  reaction = rep(as.character(1:6), 2),
  j = c(j_low, j_high)
) %>% ggplot(aes(reaction, j, fill = ki)) +
  geom_col(position = 'dodge')
```
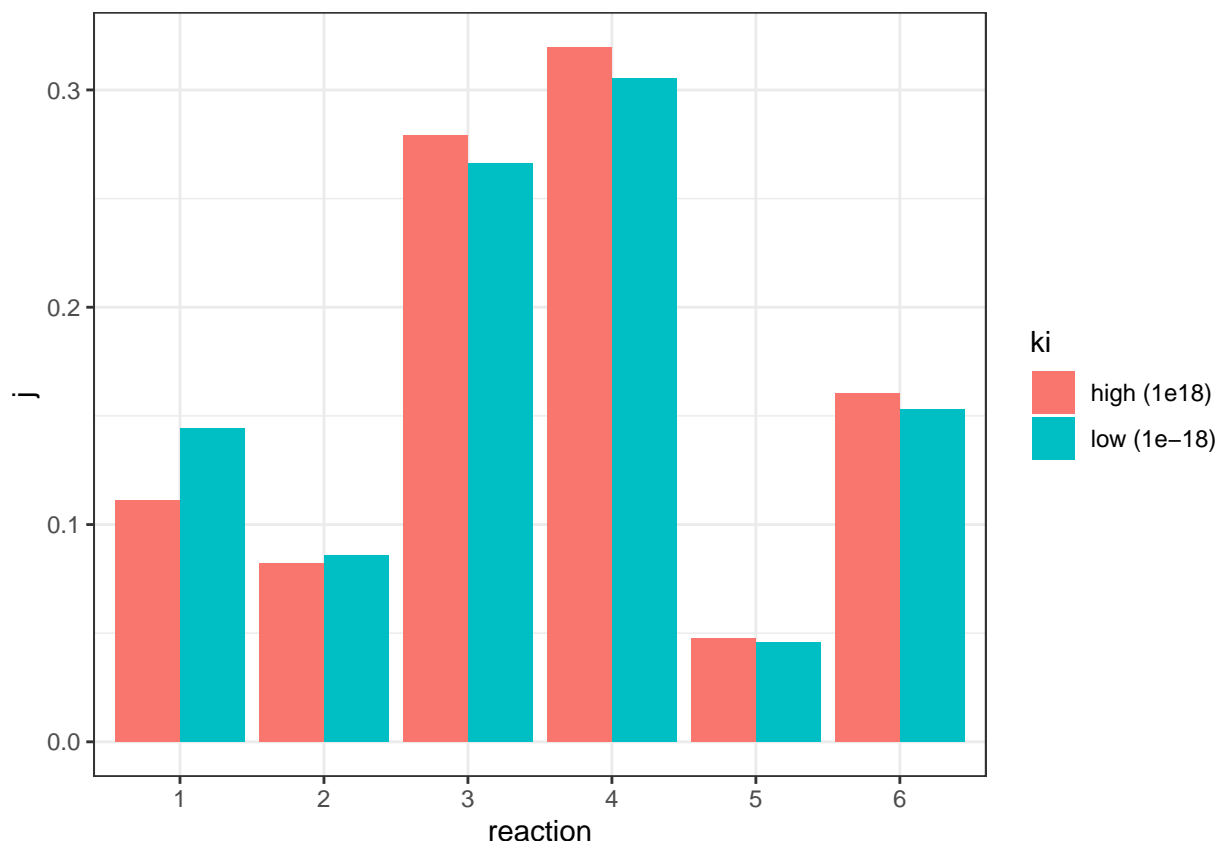
The plot shows that, when the flux is made constant, a lower K_i (higher affinity of inhibitor binding) increases the flux control coefficent of reaction 2 and the upstream reaction 1, and decreases that of downstream reactions, which is consistent with the previous experiment. However, the amount of change is not as much as previously modelled.

3.1.1 Repeat with Model 212

I repeat the last analysis with model 212, this time choosing reaction 1 as the one to be affected by the inhibitor.

```
ki_low = 1e-16
j_low = c(0.982287, 0.0109131,  0.00433655, 0.00237021, 6.9396e-05, 2.40539e-05)
flux_low = 0.091902
ki_high = 1e16
flux_high = 0.807846
j_high = c(0.437942,    0.31832,    0.104533,   0.12481,    0.0105218,  0.00387277)
```

Optimization Result:

```
    Objective Function Value:   0.807846
    Function Evaluations:    82
    CPU Time [s]:   0.018
    Evaluations/Second [1/s]:   4555.56

    (R1).Vf: 25.7611
```

I adjusted the V_f of reaction 1 from 5 to 25.7611 (6), verified that the flux is 0.807846 (the same as in the state with negligible inhibition), and the values of J are:

```
j_low <-    c(0.479732, 0.294652,   0.0967611, 0.11553,   0.00973947, 0.00358483)
```

comparing to the J values in the uninhibited state:

```
tibble(
  ki =c(rep("low (1e-16)", 6), rep("high (1e16)", 6)),
  reaction = rep(as.character(1:6), 2),
  j = c(j_low, j_high)
) %>% ggplot(aes(reaction, j, fill = ki)) +
  geom_col(position = 'dodge')
```
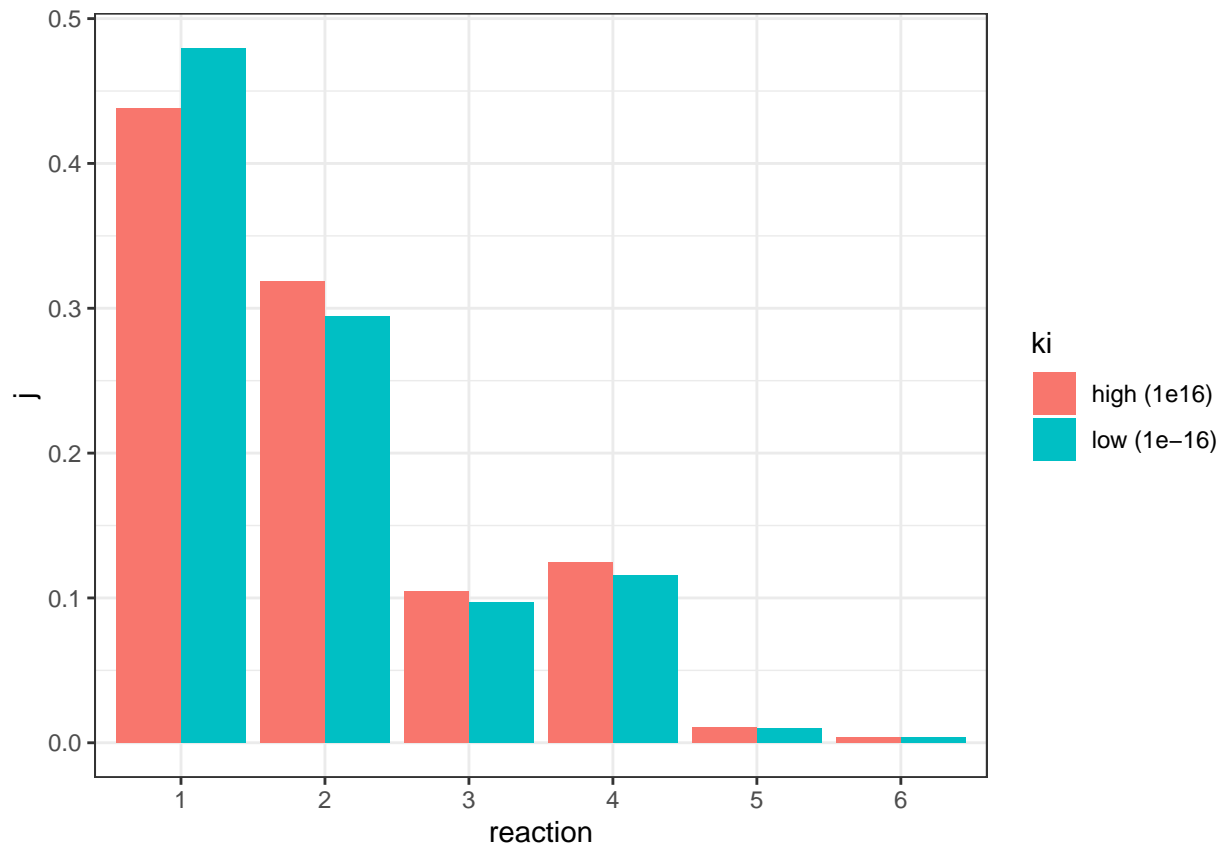

This time only reaction 1 has an increased J, and all other downstream reactions have an lowered J, which is consistent with previous observations.