Fiche TD avec le logiciel R : tdr47

Le modèle de Michaelis-Menten

Pr Jean R. Lobry

Ajustement par régression non-linéaire du modèle de Michaelis-Menten aux données de Michaelis et Menten.

Table des matières

Leonor Micahelis (1875-1949)

Maud Menten (1879-1960). Source : http://cwx.prenhall.com/horton/medialib/media_portfolio/05.html

1 Les données

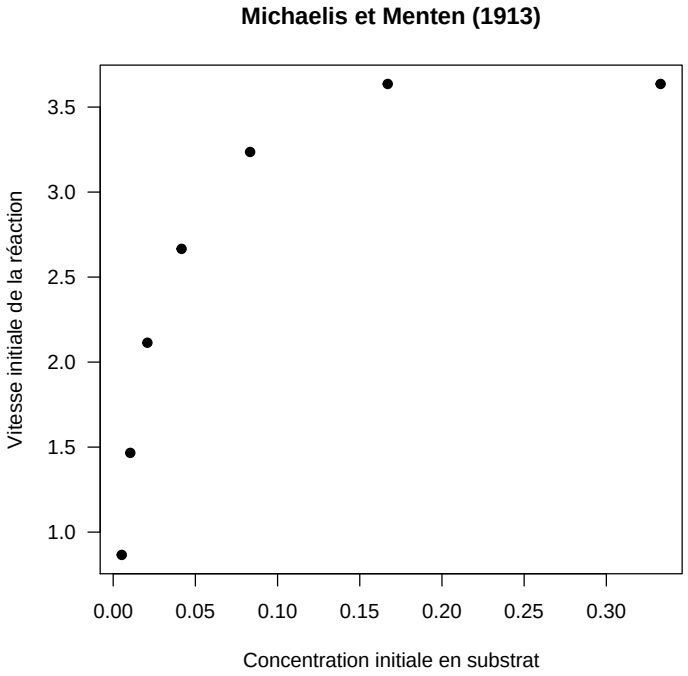
L'article de Michaelis et Menten a été publié en 1913 [10]. Les données sont extraites de la table I page 338. Cette table donne la concentration initiale en substrat, ici le saccharose (*Anfangskonzentration der Saccharose*) et la vitesse initiale de la réaction (*Anfangsgeschwindigkeit*). Les unités ne sont pas précisées dans la table. Importer les données dans R :

```
mm1913 <- read.table("http://pbil.univ-lyon1.fr/R/donnees/mm1913.txt",
                     sep = "\t", header = TRUE)
mm1913
```

```
      s     v
1 0.3330 3.636
2 0.1670 3.636
3 0.0833 3.236
4 0.0416 2.666
5 0.0208 2.114
6 0.0104 1.466
7 0.0052 0.866
```

Représentez les données :

```
plot(x = mm1913$s, y = mm1913$v, las = 1,
xlab = "Concentration initiale en substrat",
ylab = "Vitesse initiale de la réaction",
main = "Michaelis et Menten (1913)",
pch = 19)
```



2 Le modèle

Le modèle de Michaelis-Menten donne une relation entre la vitesse initiale de la réaction, v, et la concentration initiale en substrat, s,

$$v(s) = \frac{\mathrm{V_m} s}{\mathrm{K_s} + s}$$

où $\mathrm{V_m}$ représente la vitesse maximale et $\mathrm{K_s}$ la concentration en substrat telle que la vitesse soit demi-maximale.

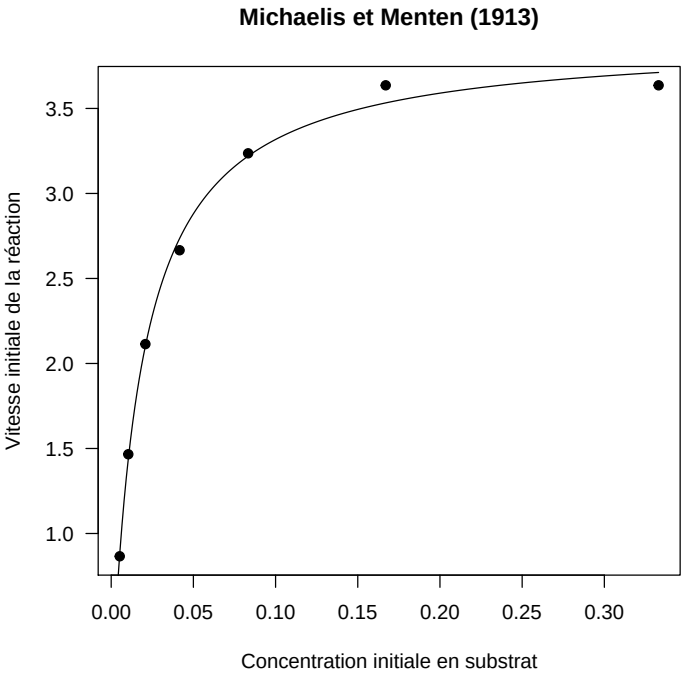
3 L'ajustement

Au vu du graphique précédent, on part d'une estimation grossière de 0.025 pour $\mathrm{K_s}$ et de 3.6 pour $\mathrm{V_m}$.

```
nlsfit <- nls(v~Vm*s/(Ks+s),data=mm1913, start=list(Ks=0.025, Vm=3.6))
nlsfit
```

```
Nonlinear regression model
  model: v ~ Vm * s/(Ks + s)
   data: mm1913
     Ks      Vm
0.01789 3.91094
 residual sum-of-squares: 0.02257
Number of iterations to convergence: 4
Achieved convergence tolerance: 5.781e-07
```

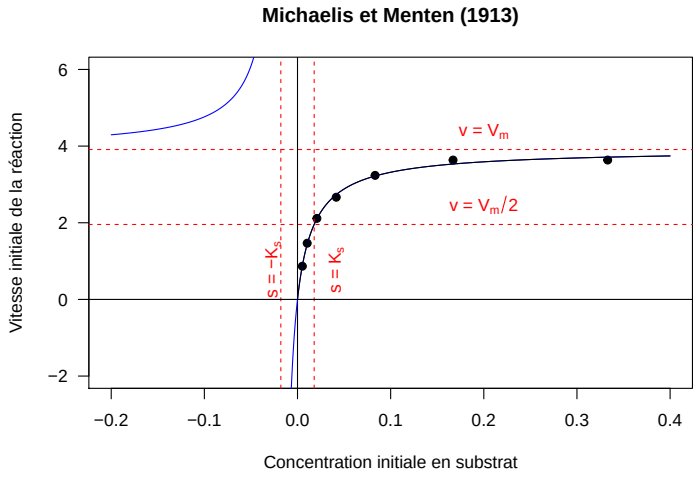
```
plot(x = mm1913$s, y = mm1913$v, las = 1,
xlab = "Concentration initiale en substrat",
ylab = "Vitesse initiale de la réaction",
main = "Michaelis et Menten (1913)",
pch = 19)
x <- seq(from = 0, to = max(mm1913$s), length = 255)
lines(x = x, y = predict(nlsfit, newdata =  list(s = x)))
```



Le modèle de Michaelis-Menten s'ajuste donc bien aux données de Michaelis et Menten. On s'en doutait un peu. Faire une représentation graphique pour mettre en valeur la signification des paramètres du modèle :

```
xmin <- -0.2
xmax <- 0.4
ymin <- -2
ymax <- 6
Ks <- nlsfit$m$getPars()["Ks"]
Vm <- nlsfit$m$getPars()["Vm"]
plot(x = mm1913$s, y = mm1913$v, las = 1,
xlab = "Concentration initiale en substrat",
ylab = "Vitesse initiale de la réaction",
main = "Michaelis et Menten (1913)",
pch = 19,
xlim = c(xmin, xmax),
ylim = c(ymin, ymax))
abline(h = Vm, col = "red", lty = 2)
text(xmax/2, Vm, expression(paste("v = ",V[m])), pos = 3, col = "red")
abline(h = Vm/2, col = "red", lty = 2)
text(xmax/2, Vm/2, expression(paste("v = ",V[m]/2)), pos = 3, col = "red")
abline(v = -Ks, col = "red", lty = 2)
text(-Ks, 0.5, expression(paste("s = -",K[s])), pos = 3, col = "red", srt = 90)
```

```
abline(v = Ks, col = "red", lty = 2)
text(0.05, 0.5, expression(paste("s = ",K[s])), pos = 3, col = "red", srt = 90)
abline(h=0)
abline(v=0)
x <- seq(from = xmin, to = xmax, length = 1000)
lines(x = x[x<(-Ks)], y = predict(nlsfit, newdata =  list(s = x[x<(-Ks)])), col = "blue")
lines(x = x[x>-Ks], y = predict(nlsfit, newdata =  list(s = x[x>-Ks])), col = "blue")
lines(x = x[x>=0], y = predict(nlsfit, newdata =  list(s = x[x>=0])))
```



4 Région de confiance pour les paramètres du modèle

Une région de confiance pour la valeur des paramètres avec un risque de première espèce α est donnée [3] par l'ensemble des valeurs des paramètres telles que la somme des carrés des résidus n'excède pas un seuil donné,

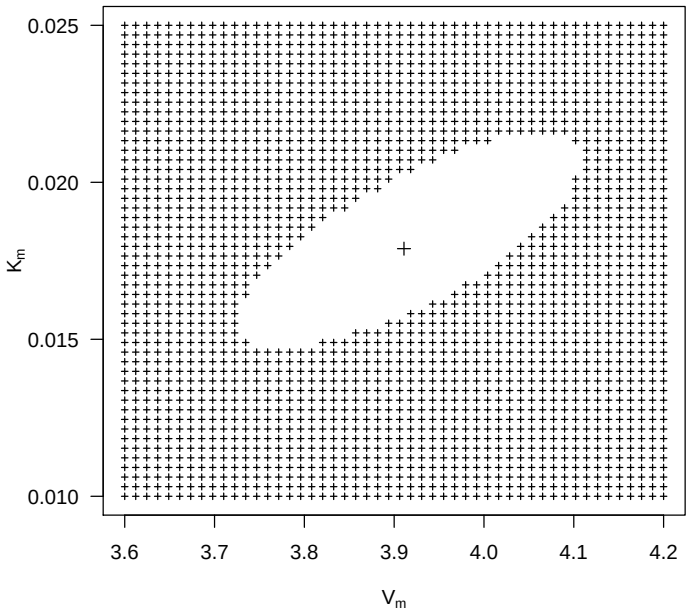
$$\theta / S(\theta) \leq S(\hat{\theta})(1 + \frac{p}{n-p} F^{\alpha}_{p;n-p})$$

où p est le nombre de paramètres du modèle, n le nombre de points disponibles dans le jeu de données, et $\hat{\theta}$ le vecteur des valeurs des paramètres tel que le critère soit minimal.

On peut faire une exploration systématique dans le plan des paramètres :

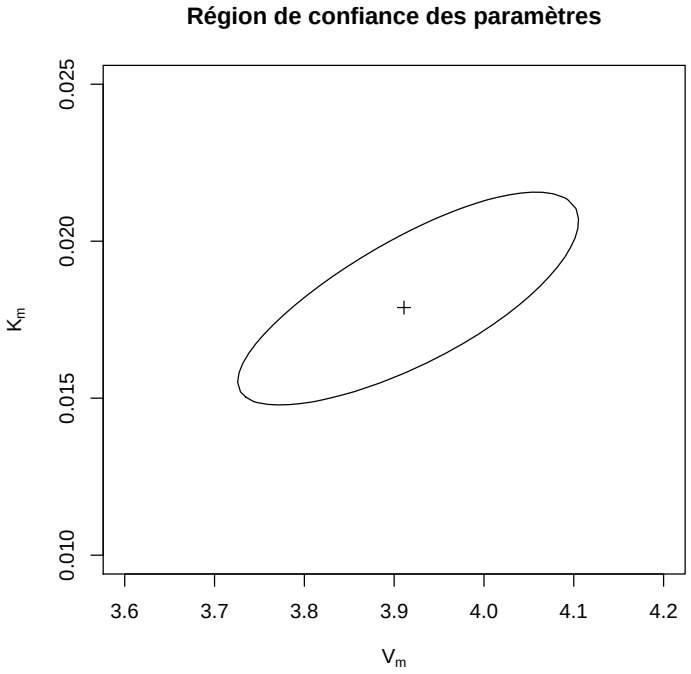
```
scemin <-  nlsfit$m$deviance()
n <- nrow(mm1913)
p <- 2
alpha = 0.05
seuil <- scemin*( 1 + (p*qf(p = 1 - alpha, df1 = p, df2 = n - p))/(n - p) )
Vmseq <- seq(from = 3.6, to  = 4.2, length = 50)
Kmseq <- seq(from = 0.01, to = 0.025, length = 50 )
scegrid <- matrix(nrow = length(Vmseq), ncol = length(Kmseq))
plot(nlsfit$m$getPars()["Vm"], nlsfit$m$getPars()["Ks"], pch = 3,
xlim = c(3.6, 4.2), ylim = c(0.01, 0.025), las = 1,
xlab = expression(V[m]),
ylab = expression(K[m]))
sce <- function(Vm, Km){
 obs <- mm1913$v
 theo <- (Vm*mm1913$s)/(Km + mm1913$s)
  sum((obs - theo)^2)
```

```
}
for( i in 1:length(Vmseq) ){
  for( j in 1:length(Kmseq)){
    scegrid[i,j] <- sce(Vm = Vmseq[i], Km = Kmseq[j])
    if(scegrid[i,j] > seuil){
      points(Vmseq[i],Kmseq[j], pch = 3, cex = 0.5)
    }
  }
}
```



Puis utiliser la fonction `contour()` pour couper la surface au niveau du seuil :

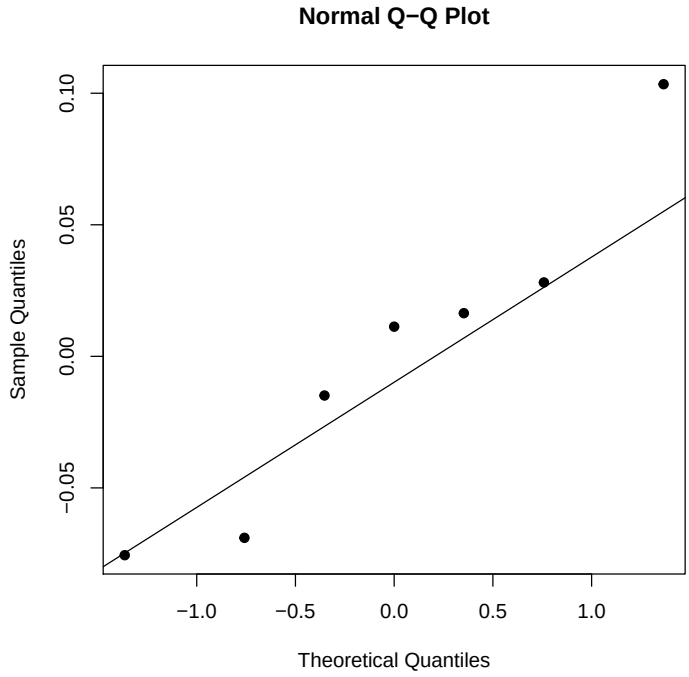
```
contour(x = Vmseq,y=Kmseq,z=scegrid, levels= seuil, drawlabels = FALSE,
xlab = expression(V[m]),
ylab = expression(K[m]),
main ="Région de confiance des paramètres")
points(nlsfit$m$getPars()["Vm"], nlsfit$m$getPars()["Ks"], pch =3)
```



5 Normalité des résidus

La détermination de la région de confiance pour les paramètres suppose que les erreurs soient *iid* gaussiennes. Il est donc intéressant de vérifier si les résidus respectent cette hypothèse. Le nombre de points est très faible ici :

```
qqnorm(nlsfit$m$resid(), pch = 19)
qqline(nlsfit$m$resid())
```



Le test de normalité de Shapiro et Wilk nous donne le résultat suivant :

```
shapiro.test(nlsfit$m$resid())
```

```
        Shapiro-Wilk normality test
data:  nlsfit$m$resid()
W = 0.93611, p-value = 0.604
```

Avec un risque de première espèce α de 5 %, les données expérimentales ne nous permettent pas de rejeter l'hypothèse de la normalité des résidus. On accepte donc faute de mieux l"hypothèse nulle avec un risque de deuxième espèce β inconnu, et sans doute très élevé ici à cause de la faible puissance du test avec aussi peu de points.

6 Transformations linéaires

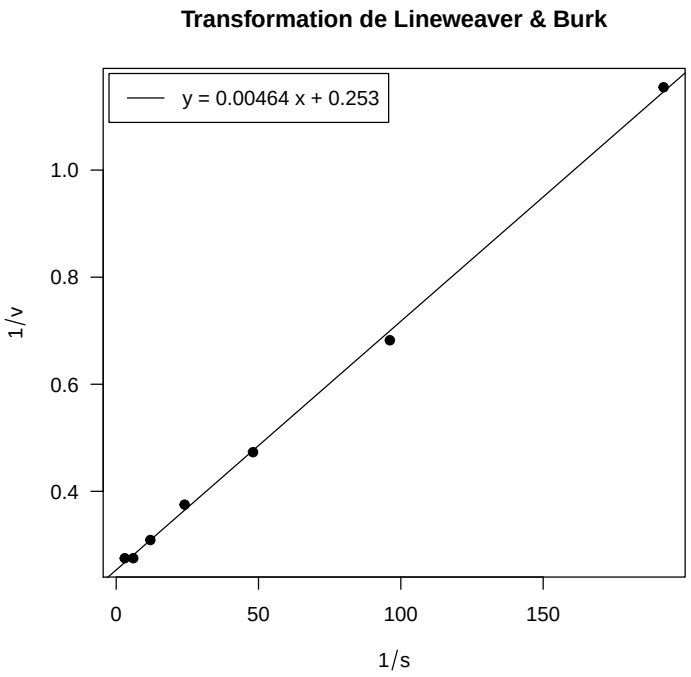
De nombreuses méthodes ont été publiées pour transformer les données de cinétique enzymatique de façon à produire une relation linéaire et faciliter ainsi l'estimation des paramètres [4, 1]. Avec les logiciels de statistique tels que R, ces transformations offrent surtout un intérêt historique.

6.1 Lineweaver et Burk

La transformation de Lineweaver et Burk [9] consiste à représenter $y = \frac{1}{v}$ en fonction de $x = \frac{1}{s}$:

$$y(x) = \frac{\mathrm{K_s}}{\mathrm{V_m}} x + \frac{1}{\mathrm{V_m}}$$

```
x <- 1/mm1913$s
y <- 1/mm1913$v
plot(x, y, pch = 19, main = "Transformation de Lineweaver & Burk",
las = 1, xlab = expression(1/s), ylab = expression(1/v))
lm1 <- lm(y~x)
abline(lm1$coef)
legend("topleft", inset = 0.01,
paste("y =", round(lm1$coef[2],5),"x +",round(lm1$coef[1],3)), lwd = 1)
```



L'ordonnée à l'origine et la pente sont donnés par :

```
lm1$coef
```
```
(Intercept)           x
0.253106355 0.004644795
```

On en déduit la valeur des estimateurs de V_m et K_s avec la transformation de Lineweaver et Burk :

```
(Vme1 <- 1/lm1$coef[1])
```
```
(Intercept)
   3.950908
```
```
(Kme1 <- Vme1*lm1$coef[2])
```
```
(Intercept)
 0.01835116
```

Les intervalles de confiance pour l'ordonnée à l'origine et la pente sont donnés par :

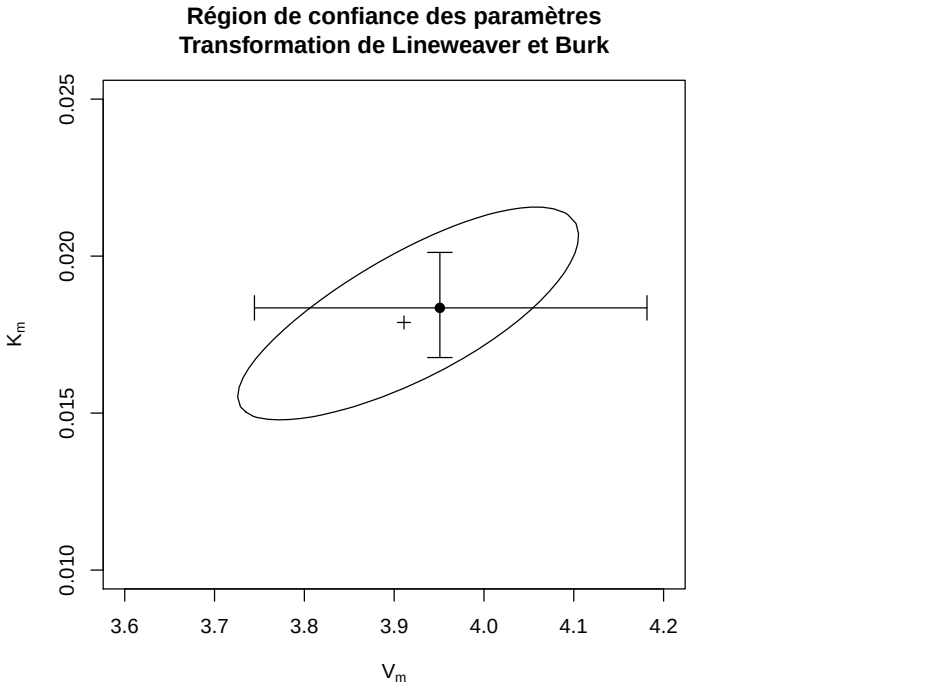
```
(conf1 <- confint(lm1))
```
```
                  2.5 %      97.5 %
(Intercept) 0.239149220 0.267063490
x           0.004478494 0.004811096
```

On en déduit la valeur des intervalles de confiance pour V_m et K_s avec la transformation de Lineweaver et Burk :

```
(Vmconf1 <- 1/conf1[1,])
   2.5 %   97.5 %
4.181490 3.744428
(Kmconf1 <- c(min(Vmconf1)*min(conf1[2,]),max(Vmconf1)*max(conf1[2,]) ))
[1] 0.01676940 0.02011755
```

Graphiquement :

```
contour(x = Vmseq,y=Kmseq,z=scegrid, levels= seuil, drawlabels = FALSE,
xlab = expression(V[m]),
ylab = expression(K[m]),
main ="Région de confiance des paramètres\nTransformation de Lineweaver et Burk")
points(nlsfit$m$getPars()["Vm"], nlsfit$m$getPars()["Ks"], pch =3)
points(Vme1, Kme1, pch = 19)
arrows(Vme1, Kmconf1[1], Vme1, Kmconf1[2], angle = 90, code = 3, le = 0.1)
arrows(Vmconf1[1], Kme1, Vmconf1[2], Kme1, angle = 90, code = 3, le = 0.1)
```



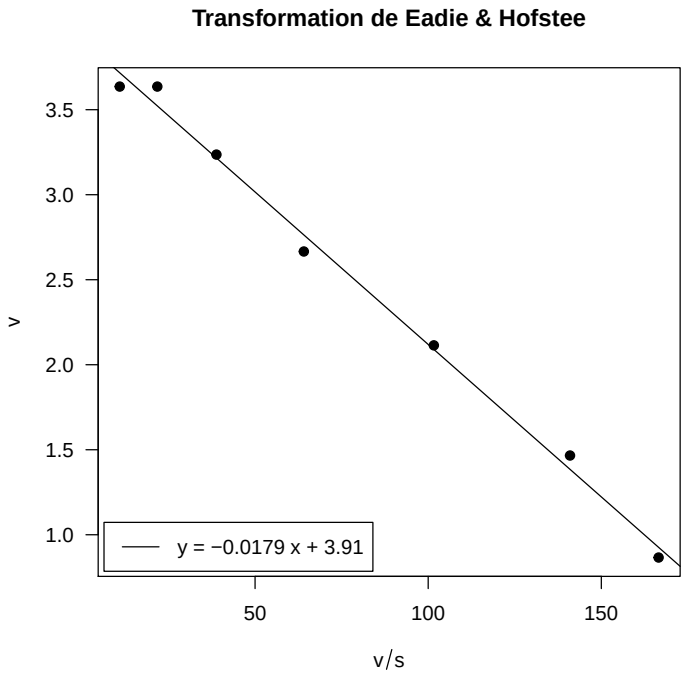
6.2 Eadie et Hofstee

La transformation de Eadie et Hofstee [5, 8] est également attribuée à Augustinsson [2]. J.B.S. Haldane en donne la paternité [6] à son ami le Dr. Barnett Woolf pour la traduction allemande en 1932 de son livre [7]. Cette transformation consiste à représenter $y = v$ en fonction de $x = \frac{v}{s}$:

$$y(x) = -\mathrm{K_s}x + \mathrm{V_m}$$

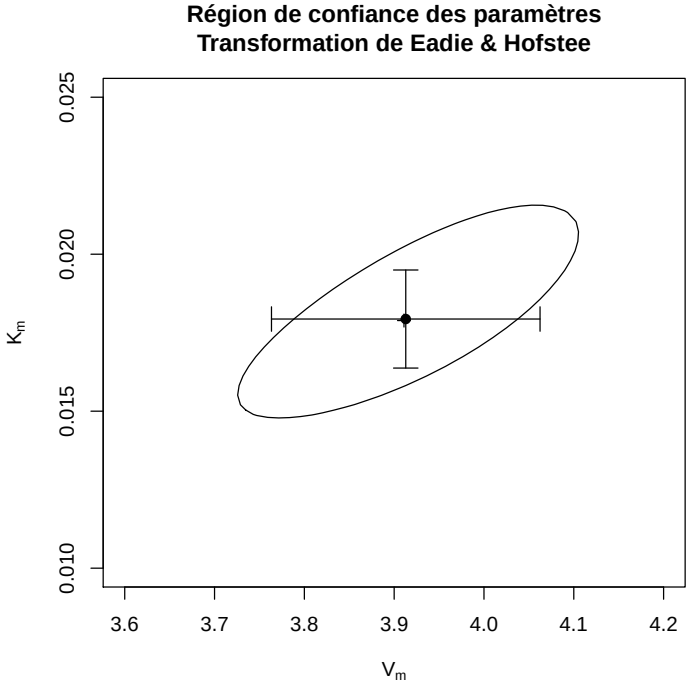
```
x <- mm1913$v/mm1913$s
y <- mm1913$v
plot(x, y, pch = 19, main = "Transformation de Eadie & Hofstee",
xlab = expression(v/s), ylab = expression(v),
las = 1)
```

```
lm2 <- lm(y~x)
abline(lm2$coef)
legend("bottomleft", inset = 0.01,
paste("y =", round(lm2$coef[2],4),"x +",round(lm2$coef[1],2)), lwd = 1)
```



Graphiquement :

```
Vme2 <- lm2$coef[1]
Kme2 <- -lm2$coef[2]
conf2 <- confint(lm2)
Vmconf2 <- conf2[1,]
Kmconf2 <- -conf2[2,]
contour(x = Vmseq,y=Kmseq,z=scegrid, levels= seuil, drawlabels = FALSE,
xlab = expression(V[m]),
ylab = expression(K[m]),
main ="Région de confiance des paramètres\nTransformation de Eadie & Hofstee")
points(nlsfit$m$getPars()["Vm"], nlsfit$m$getPars()["Ks"], pch =3)
points(Vme2, Kme2, pch = 19)
arrows(Vme2, Kmconf2[1], Vme2, Kmconf2[2], angle = 90, code = 3, le = 0.1)
arrows(Vmconf2[1], Kme2, Vmconf2[2], Kme2, angle = 90, code = 3, le = 0.1)
```

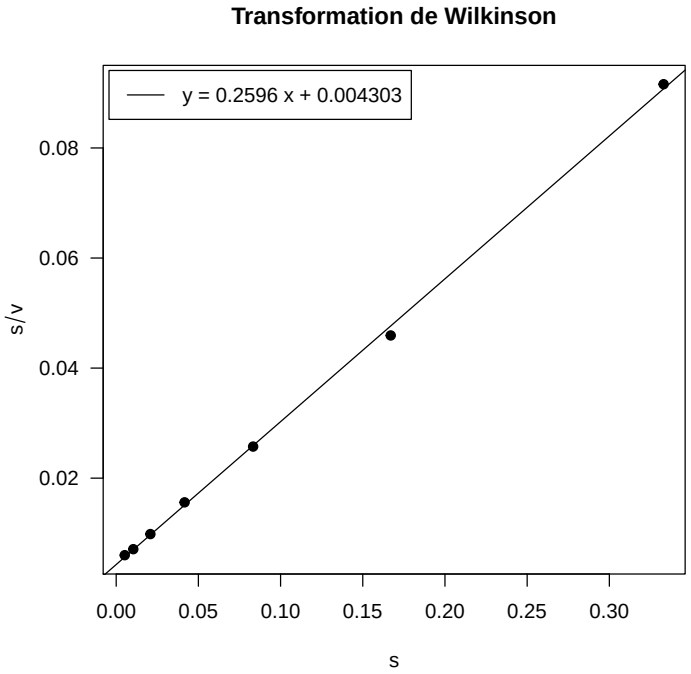


6.3 Wilkinson

La transformation de Wilkinson [11] consiste à représenter $y = \frac{s}{v}$ en fonction de $x = s$:

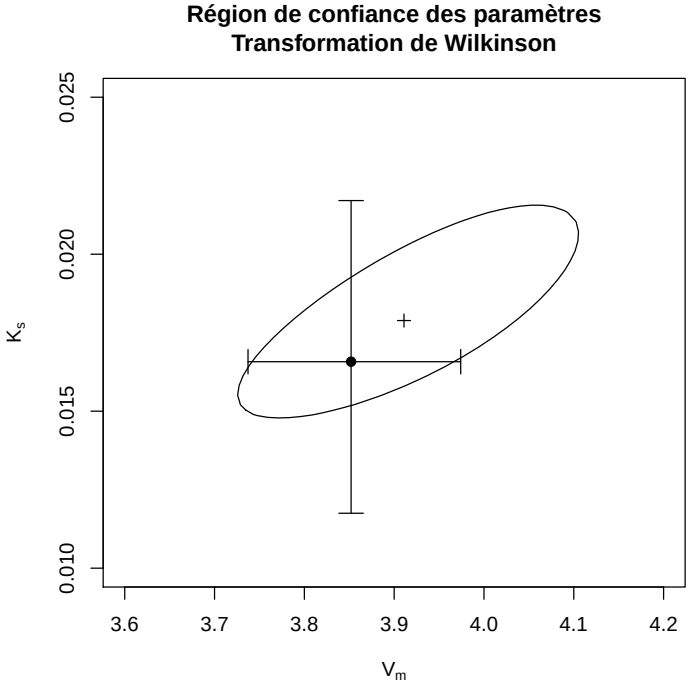
$$y(x) = \frac{1}{\mathrm{V_m}} x + \frac{\mathrm{K_s}}{\mathrm{V_m}}$$

```
x <- mm1913$s
y <- mm1913$s/mm1913$v
plot(x, y, pch = 19, main = "Transformation de Wilkinson",
xlab = expression(s), ylab = expression(s/v),
las = 1)
lm3 <- lm(y~x)
abline(lm3$coef)
legend("topleft", inset = 0.01,
paste("y =", round(lm3$coef[2],4),"x +",round(lm3$coef[1],6)), lwd = 1)
```



Graphiquement :

```
Vme3 <- 1/lm3$coef[2]
Kme3 <- Vme3*lm3$coef[1]
conf3 <- confint(lm3)
Vmconf3 <- 1/conf3[2,]
Kmconf3 <- c(min(Vmconf3)*min(conf3[1,]), max(Vmconf3)*max(conf3[1,]))
contour(x = Vmseq,y=Kmseq,z=scegrid, levels= seuil, drawlabels = FALSE,
xlab = expression(V[m]),
ylab = expression(K[s]),
main ="Région de confiance des paramètres\nTransformation de Wilkinson")
points(nlsfit$m$getPars()["Vm"], nlsfit$m$getPars()["Ks"], pch =3)
points(Vme3, Kme3, pch = 19)
arrows(Vme3, Kmconf3[1], Vme3, Kmconf3[2], angle = 90, code = 3, le = 0.1)
arrows(Vmconf3[1], Kme3, Vmconf3[2], Kme3, angle = 90, code = 3, le = 0.1)
```



Références

[1] G.L. Atkins and I.A. Nimmo. A comparison of seven methods for fitting the Michaelis-Menten equation. *Biochemical Journal*, 149 :775–777, 1975.

[2] K.-B. Augustinsson. Cholinesterase. A study in comparative enzymology. *Acta Physiologica Scandinavia*, 15 :S52, 1948.

[3] E.M.L. Beale. Confidence regions in non-linear estimation. *Journal of the Royal Statistical Society*, 22B :41–88, 1960.

[4] J.E. Dowd and D.S. Riggs. A comparison of estimates of Michaelis-Menten kinetic constants from various linear transformations. *The Journal of Biological Chemistry*, 240 :863–869, 1965.

[5] G.S. Eadie. The inhibition of cholinesterase by physostigmine and prostigmine. *Journal of Biological Chemistry*, 146 :85–93, 1942.

[6] J.B.S. Haldane. Graphical methods in enzyme chemistry. *Nature*, 179 :832–832, 1957.

[7] J.B.S. Haldane and K. Stern. *Allgemeine Chemie der Enzyme*. Steinkopf, Leipzig, Berlin, Germany, 1932.

[8] B.H.J. Hofstee. On the evaluation of the constants V_m and K_M in enzyme reactions. *Science*, 116 :329–331, 1952.

[9] H. Lineweaver and D. Burk. The determination of enzyme dissociation constants. *Journal of the American Chemical Society*, 56 :658–666, 1934.

[10] L. Michaelis and M.L. Menten. Die Kinetik der Invertinwirkung. *Biochemische Zeitschrift*, 49 :333–369, 1913.

[11] G.N. Wilkinson. Statistical estimations in enzyme kinetics. *Biochemical Journal*, 80 :324–332, 1961.