Examples

Loading packages

R-pakken *bmd* ligger på GitHub i depotet "doseResponse" sammen med nyeste version af *drc* og pakken *drcData* der indeholder eksempel-datasæts til dosis-respons modellering.

Ved første brug af en R-pakke skal den installeres. Det gøres for de fleste pakker gennem **Tools -> Install** packages fra CRAN, hvorefter de loades i R med library(). For at installere pakkerne *bmd*, *drc* og *drcData* der ikke ligger på CRAN men GitHub skal først pakken devtools installeres og loades. Derefter kan pakkerne installeres med install_github() som herunder. Foruden de tre allerede nævnte pakker skal vi også bruge *ggplot2*, *plyr*, *sandwich* og *metafor*, samt *mmmVcov* der kan findes i GitHub depotet "SigneMJensen".

```
library(devtools)
install_github("doseResponse/drc")
install_github("doseResponse/drcData")
install_github("doseResponse/bmd")
library(drc)
library(bmd)
library(drcData)
library(ggplot2)
library(plyr)
library(sandwich)
library(metafor)
install_github("SigneMJensen/mmmVcov")
library(mmmVcov)
```

Eksempel 1. Binomiale data: Regnorme test på chloroacetamide

Data til dette eksempel ligger i R-pakken drcData under navnet "clorac".

De første rækker af datasættet kan ses ved at bruge funktionen head().

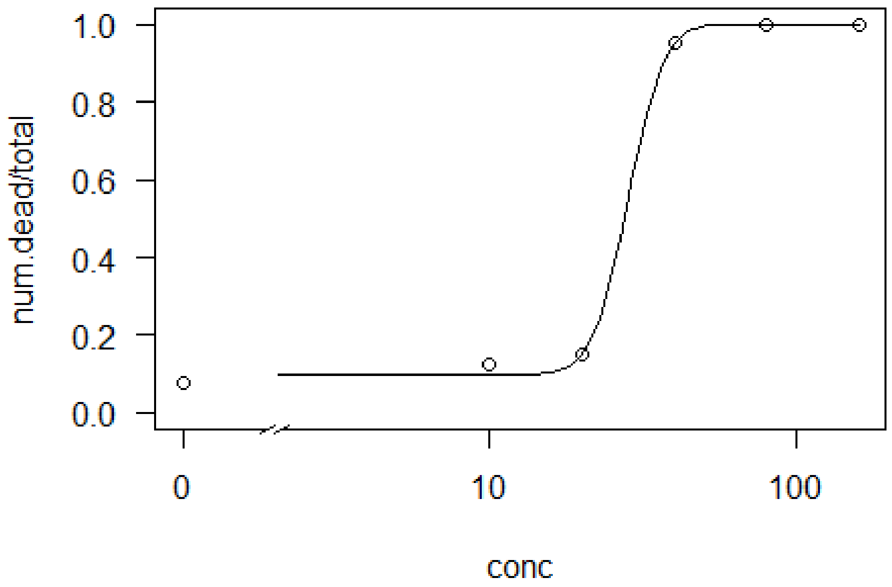
```
head(chlorac)

##   conc total num.dead
## 1    0    40        3
## 2   10    40        5
## 3   20    40        6
## 4   40    40       38
## 5   80    40       40
## 6  160    40       40
```

En 3-parameter log-normal dosis-respons model med øvre grænse 1 fittes til de binomiale data som følger.

```
chlorac.LN.3 <- drm(num.dead/total ~ conc, weights = total,
                    data = chlorac, fct = LN.3u(), type = "binomial")
```

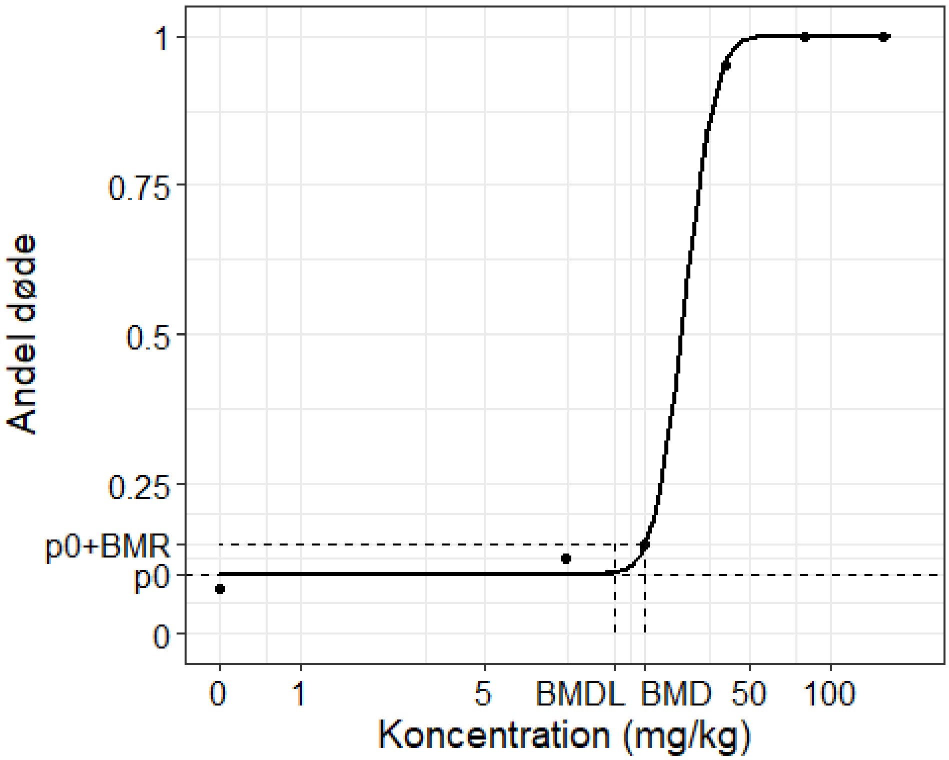
Det fittede model beskriver data fint.

```
plot(chlorac.LN.3, broken = TRUE, ylim = c(0,1))
```



For et pænere plot som præsenteret i rapporten kan følgende kode bruges:

```
newdata <- expand.grid(conc = exp(seq(log(0.5), log(170),
length = 100)))
pm <- predict(chlorac.LN.3, newdata = newdata,
interval = "confidence")
newdata$p <- pm[, 1]
newdata$pmin <- pm[, 2]
newdata$pmax <- pm[, 3]
chlorac$conc0 <- chlorac$conc
chlorac$conc0[chlorac$conc0 == 0] <- 0.5
```



De estimerede parametre ses med summary(). c-parameteren er nedre grænse, dvs. den naturlige dødelighed i dette eksempel.

```
summary(chlorac.LN.3)

##
## Model fitted: Log-normal with upper limit at 1 (3 parms)
##
## Parameter estimates:
##
##                 Estimate Std. Error t-value   p-value
## b:(Intercept)  4.603773   1.043813  4.4105 1.031e-05 ***
## c:(Intercept)  0.099988   0.033573  2.9783  0.002899 **
## e:(Intercept) 28.291922   2.271962 12.4526 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Konfidensintervaller fås med confint()

```
confint(chlorac.LN.3)

##                      2.5 %     97.5 %
## b:(Intercept)  2.55793762  6.6496084
## c:(Intercept)  0.03418668  0.1657886
## e:(Intercept) 23.83895875 32.7448861
```

BMD hørende til BMR = 0.05 med “excess risk” definitionen er:

```
bmd(chlorac.LN.3, bmr = 0.05, backgType = "modelBased", def = "excess")

##       BMD     BMDL
##  19.79229 15.15075
```

BMD hørende til BMR = 0.05 med “additional risk” definitionen er:

```
bmd(chlorac.LN.3, bmr = 0.05, backgType = "modelBased", def = "additional")

##      BMD     BMDL
##  20.0155 15.39552
```

Eksempel 2. Tælle data: Toksicitet test med kobber under varierende temperatur

Data ligger på GitHub sammen med denne kode under navnet VarTemp.csv

Når datasættet er gemt på egen computer kan det hentes ind med følgende kommando der giver dig åbner et nyt vindue hvor du kan lede efter filen.

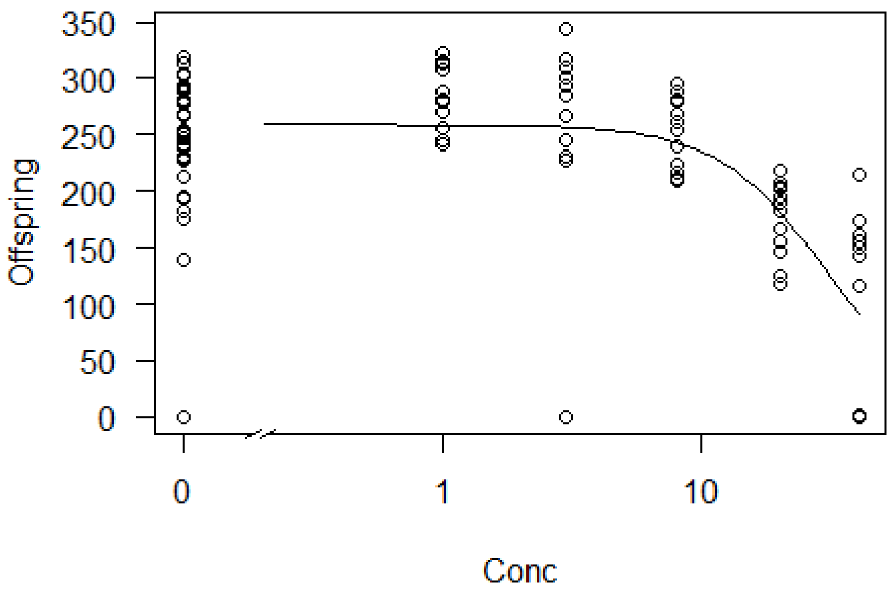
```
CopperTemp <- read.csv2(file.choose())
```

En tre-parameter log-logistisk model med nedre grænse 0 fittes til data under antagelse af Poisson fordelte data

```
CopperTemp.m1<-drm(Offspring ~ Conc,
                   data = CopperTemp,
                   type = "Poisson",
                   fct = LL.3())
```

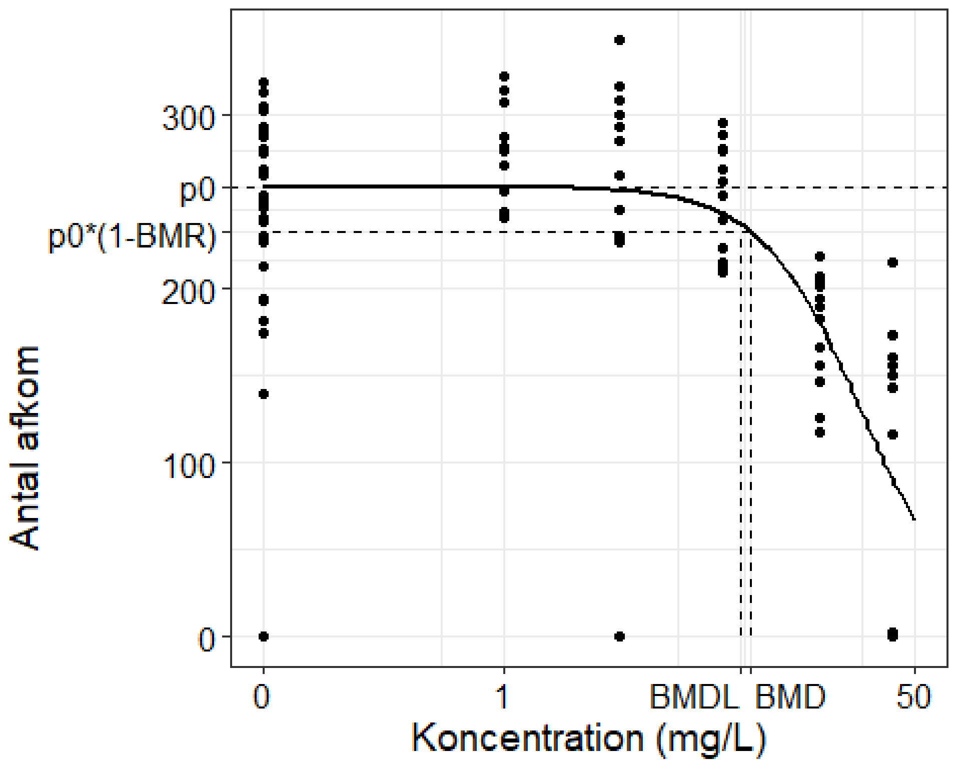
Et hurtigt visuelt tjek af modelfittet viser at modellen beskriver data fint.

```
plot(CopperTemp.m1, broken = TRUE, type = "all")
```



Plottet som det er præsenteret i rapporten

```
newdata <- expand.grid(conc = exp(seq(log(0.1), log(50),
length = 100)))
pm <- predict(CopperTemp.m1, newdata = newdata,
interval = "confidence")
newdata$p <- pm[, 1]
newdata$pmin <- pm[, 2]
newdata$pmax <- pm[, 3]
CopperTemp$conc0 <- CopperTemp$Conc
CopperTemp$conc0[CopperTemp$conc0 == 0] <- 0.1
```



BMD hørense til BMR = 0.1 og den relative definition

```
bmd(CopperTemp.m1, bmr = 0.1, backgType = "modelBased", def = "relative")

##        BMD     BMDL
##  10.47006 9.403057
```

En model der tager højde for levetiden af den enkelte nematode fås ved at udnytte argumentet weights.

```
CopperTemp.m2 <- drm(Offspring ~ Conc,
                     data = CopperTemp,
                     weights = Lifespan,
                     type = "Poisson",
                     fct = LL.3())
```

BMD for denne model

```
bmd(CopperTemp.m2,
    bmr = 0.1,
    backgType = "modelBased",
    def = "relative")

##        BMD     BMDL
##  31.16192 28.89062
```

BMD med brug af sandwich estimatorer til at tage højde for overdispersion

```
bmd(CopperTemp.m2,
    bmr = 0.1,
    backgType = "modelBased",
    def = "relative",
    sandwich.vcov = TRUE)
```

```
##        BMD     BMDL
##   31.16192 18.62904
```

Dosis-repsons model med Poisson antagelse skiftet ud med en antagelse om negativt binomiale data

```
CopperTemp.m3 <- drm(Offspring ~ Conc,
                     data = CopperTemp,
                     weights = Lifespan,
                     type = "negbin2",
                     fct = LL.3())
```

BMD for denne model

```
bmd(CopperTemp.m3,
    bmr = 0.1,
    backgType = "modelBased",
    def = "relative")
```

```
##        BMD     BMDL
##   27.98813 15.99371
```

Eksempel 3. Kontinuerte data og model-gennemsnit: Fiske test

Data kan findes i R-pakken drcData under navnet O.mykiss

Alle missing data smides væk.

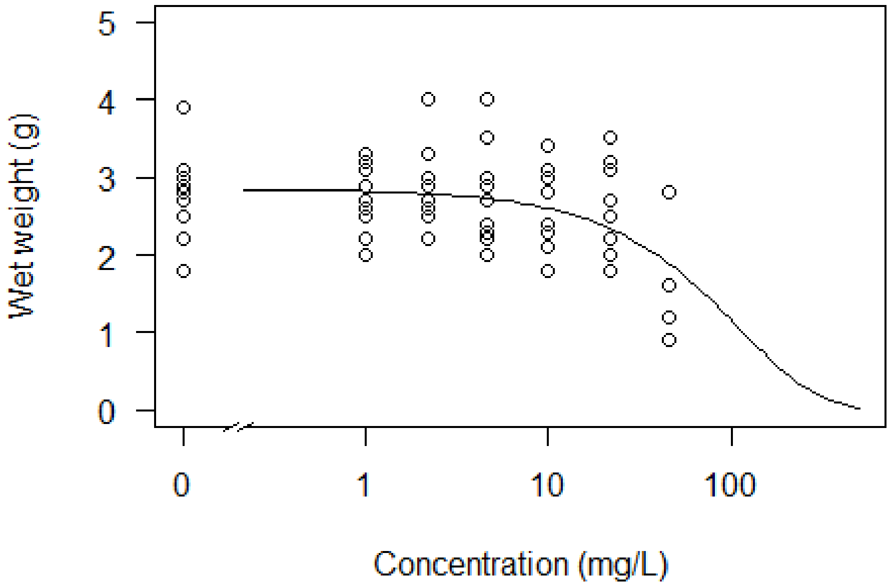
```
O.mykiss.c <- na.omit(O.mykiss)
```

En to-parameter exponential henfaldsmodel fittes til data.

```
O.mykiss.EXD.2 <- drm(weight ~ conc,
                      data = O.mykiss.c,
                      fct = EXD.2())
```

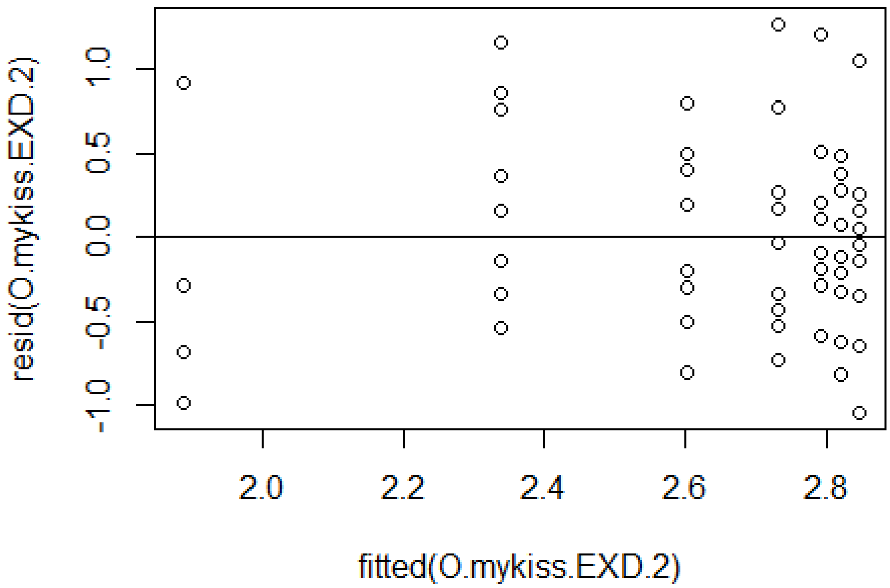
Et plot af den fittede model viser at modellen beskriver data fit.

```
plot(O.mykiss.EXD.2, broken = TRUE, type="all",
     xlim = c(0, 500), ylim = c(0, 5),
     xlab="Concentration (mg/L)", ylab="Wet weight (g)")
```



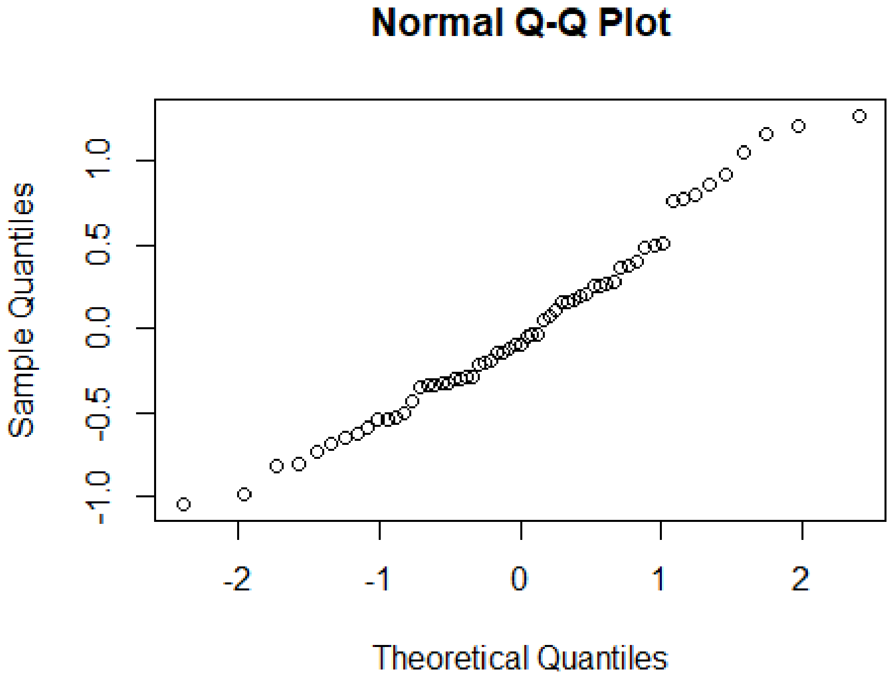
Residual plot

```
plot(resid(O.mykiss.EXD.2)  ~ fitted(O.mykiss.EXD.2))
abline(h=0)
```



QQ-plot

```
qqnorm(resid(O.mykiss.EXD.2))
```



BMD hørende til BMR = 0.05 med brug af hybrid metoden hvor 2 SD betragtes som cutoff.

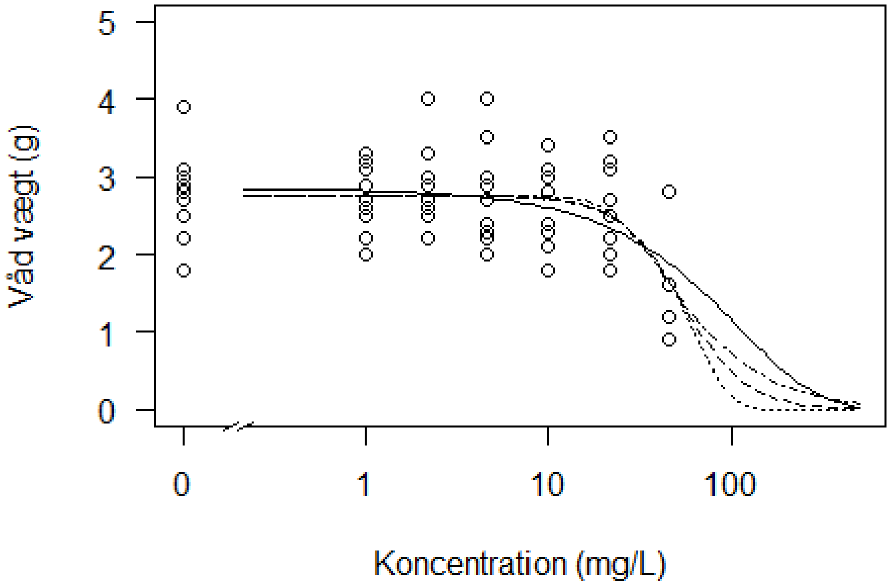
```
bmd(O.mykiss.EXD.2,
    bmr = 0.05,
    backgType = "hybridSD",
    def = "hybridAdd",
    backg = 2)

##       BMD     BMDL
##  12.65354 6.371416
```

Fitter tre andre modeller til data; en tre-parameter log-logistisk model samt to tre-parameter Weibull modeller.

```
O.mykiss.LL.3 <- drm(weight ~ conc,
                     data = O.mykiss,
                     fct = LL.3(),
                     na.action = na.omit)

O.mykiss.W1.3 <- drm(weight ~ conc,
                     data = O.mykiss,
                     fct = W1.3(),
                     na.action = na.omit)

O.mykiss.W2.3 <- drm(weight ~ conc,
                     data = O.mykiss,
                     fct = W2.3(),
                     na.action = na.omit)
```

Plot af alle fire modeller i samme plot

```
plot(O.mykiss.EXD.2, broken = TRUE, type = "all",
     xlim = c(0, 500), ylim = c(0, 5),
     xlab = "Koncentration (mg/L)", ylab = "Våd vægt (g)")

plot(O.mykiss.LL.3, broken = TRUE, type = "all",
     xlim = c(0, 500), ylim = c(0, 5),
     add = TRUE, lty = 2)

plot(O.mykiss.W1.3, broken = TRUE, type = "all",
     xlim = c(0, 500), ylim = c(0, 5),
     add = TRUE, lty = 3)

plot(O.mykiss.W2.3, broken = TRUE, type = "all",
     xlim = c(0, 500), ylim = c(0, 5),
     add = TRUE, lty = 4)
```



Pænere plot som præsenteret i rapporten

```
newdata <- expand.grid(conc = exp(seq(log(0.1), log(100),
length = 100)))
pm <- predict(O.mykiss.EXD.2, newdata = newdata,
interval = "confidence")
newdata$p <- pm[, 1]
newdata$pmin <- pm[, 2]
newdata$pmax <- pm[, 3]

newdata2 <- expand.grid(conc = exp(seq(log(0.1), log(100),
length = 100)))
pm <- predict(O.mykiss.LL.3, newdata = newdata2,
interval = "confidence")
newdata2$p <- pm[, 1]
newdata2$pmin <- pm[, 2]
newdata2$pmax <- pm[, 3]

newdata3 <- expand.grid(conc = exp(seq(log(0.1), log(100),
length = 100)))
pm <- predict(O.mykiss.W1.3, newdata = newdata3,
interval = "confidence")
newdata3$p <- pm[, 1]
newdata3$pmin <- pm[, 2]
newdata3$pmax <- pm[, 3]

newdata4 <- expand.grid(conc = exp(seq(log(0.1), log(100),
length = 100)))
pm <- predict(O.mykiss.W2.3, newdata = newdata4,
interval = "confidence")
newdata4$p <- pm[, 1]
newdata4$pmin <- pm[, 2]
newdata4$pmax <- pm[, 3]
newdata[["Model"]]<-"2-parameter exponential decay"
newdata2[["Model"]]<-"3-parameter log-logistic"
newdata3[["Model"]]<-"3-parameter Weibull 1"
newdata4[["Model"]]<-"3-parameter Weibull 2"
newdata.all<-rbind(newdata,newdata2,newdata3,newdata4)

O.mykiss$conc0 <- O.mykiss$conc
O.mykiss$conc0[O.mykiss$conc0 == 0] <- 0.1
```

Sammenligner model fit med AIC

```
AIC(O.mykiss.EXD.2,
    O.mykiss.LL.3,
    O.mykiss.W1.3,
    O.mykiss.W2.3)
```

```
##                 df      AIC
## O.mykiss.EXD.2  3 106.3066
## O.mykiss.LL.3   4 106.6501
## O.mykiss.W1.3   4 106.5823
## O.mykiss.W2.3   4 106.9406
```

Finder BMD for de sidste tre modeller

```
bmd(O.mykiss.LL.3,
    bmr = 0.05,
    backgType = "hybridSD",
    def = "hybridAdd",
    backg = 2)

##       BMD     BMDL
##  22.91039 9.046736

bmd(O.mykiss.W1.3,
    bmr = 0.05,
    backgType = "hybridSD",
    def = "hybridAdd",
    backg = 2)

##       BMD     BMDL
##  22.71909 8.124225

bmd(O.mykiss.W2.3,
    bmr = 0.05,
    backgType = "hybridSD",
    def = "hybridAdd",
    backg = 2)

##       BMD    BMDL
##  24.25913 14.2518
```

BMD fundet ved model-gennemsnit ved brug af Buckland metoden til at finde BMDL

```
modelList <- list(O.mykiss.EXD.2, O.mykiss.LL.3, O.mykiss.W1.3, O.mykiss.W2.3)

bmdMA(modelList, modelWeights = "AIC",
      bmr = 0.05,
      backgType = "hybridSD",
      def = "hybridAdd",
      backg = 2,
      type = "Buckland")

##    BMD_MA  BMDL_MA
##  19.68024 5.804238
```

BMD fundet ved model-gennemsnit ved brug af bootstrap til at finde BMDL

```
bmdMA(modelList, modelWeights = "AIC",
      bmr = 0.05,
      backgType = "hybridSD",
      def = "hybridAdd",
      backg = 2,
      type = "bootstrap")

##    BMD_MA  BMDL_MA
##  19.68024 8.590238
```

BMD fundet ved model-gennemsnit på hele kurver

```
bmdMA(modelList, modelWeights = "AIC",
      bmr = 0.05,
      backgType = "hybridSD",
      def = "hybridAdd",
      backg = 2,
      type = "curve")

##  BMD_MA  BMDL_MA
##  20.354 7.852576
```

Eksempel 4. Binomiale data i et hierarkisk design: En akut toksicitet test på α-cypermethrin

Data til dette eksempel kan findes på GitHub sammen med denne kode under navnet Alpha_cyp.csv

```
alpha.cyp <- read.csv2(file.choose())
```

De første rækker af datasættet

```
head(alpha.cyp)

##   Dose Total Mobile  Exp
## 1    0     5      5 Exp1
## 2    0     5      4 Exp1
## 3    0     5      5 Exp1
## 4    0     5      5 Exp1
## 5    0     5      5 Exp1
## 6    0     5      5 Exp1
```

Tilføjer en ny kolonne der angiver antallet af immobile

```
alpha.cyp[["Immobile"]] <- with(alpha.cyp, Total - Mobile)
```

Fitter en to-parameter log-logistisk model for hvert del-eksperiment

```
alpha.m1 <- drm(Immobile/Total ~ Dose, weights = Total,
                data = subset(alpha.cyp, Exp == "Exp1"),
                type = "binomial", fct = LL.2())
alpha.m2 <- drm(Immobile/Total ~ Dose, weights = Total,
                data = subset(alpha.cyp, Exp == "Exp2"),
                type = "binomial", fct = LL.2())
alpha.m3 <- drm(Immobile/Total ~ Dose, weights = Total,
                data = subset(alpha.cyp, Exp == "Exp3"),
                type = "binomial", fct = LL.2())
alpha.m4 <- drm(Immobile/Total ~ Dose, weights = Total,
                data = subset(alpha.cyp, Exp == "Exp4"),
                type = "binomial", fct = LL.2())
alpha.m5 <- drm(Immobile/Total ~ Dose, weights = Total,
                data = subset(alpha.cyp, Exp == "Exp5"),
                type = "binomial", fct = LL.2())
alpha.m6 <- drm(Immobile/Total ~ Dose, weights = Total,
                data = subset(alpha.cyp, Exp == "Exp6"),
                type = "binomial", fct = LL.2())
alpha.m7 <- drm(Immobile/Total ~ Dose, weights = Total,
                data = subset(alpha.cyp, Exp == "Exp7"),
                type = "binomial", fct = LL.2())
alpha.m8 <- drm(Immobile/Total ~ Dose, weights = Total,
                data = subset(alpha.cyp, Exp == "Exp8"),
                type = "binomial", fct = LL.2())
alpha.m9 <- drm(Immobile/Total ~ Dose, weights = Total,
                data = subset(alpha.cyp, Exp == "Exp9"),
                type = "binomial", fct = LL.2())
```

Finder BMD for hver del-eksperiment. BMR = 0.05 og definitionen er “excess risk”

```
alpha.m1.bmd <- bmd(alpha.m1, bmr = 0.05,
                    backgType = "modelBased", def = "excess")

##         BMD      BMDL
##  0.1696149 0.1004246

alpha.m2.bmd <- bmd(alpha.m2, bmr = 0.05,
                    backgType = "modelBased", def = "excess")

##        BMD       BMDL
##  0.386818 -0.5760919

alpha.m3.bmd <- bmd(alpha.m3, bmr = 0.05,
                    backgType = "modelBased", def = "excess")

##          BMD       BMDL
##  0.04421228 0.02135657

alpha.m4.bmd <- bmd(alpha.m4, bmr = 0.05,
                    backgType = "modelBased", def = "excess")
```

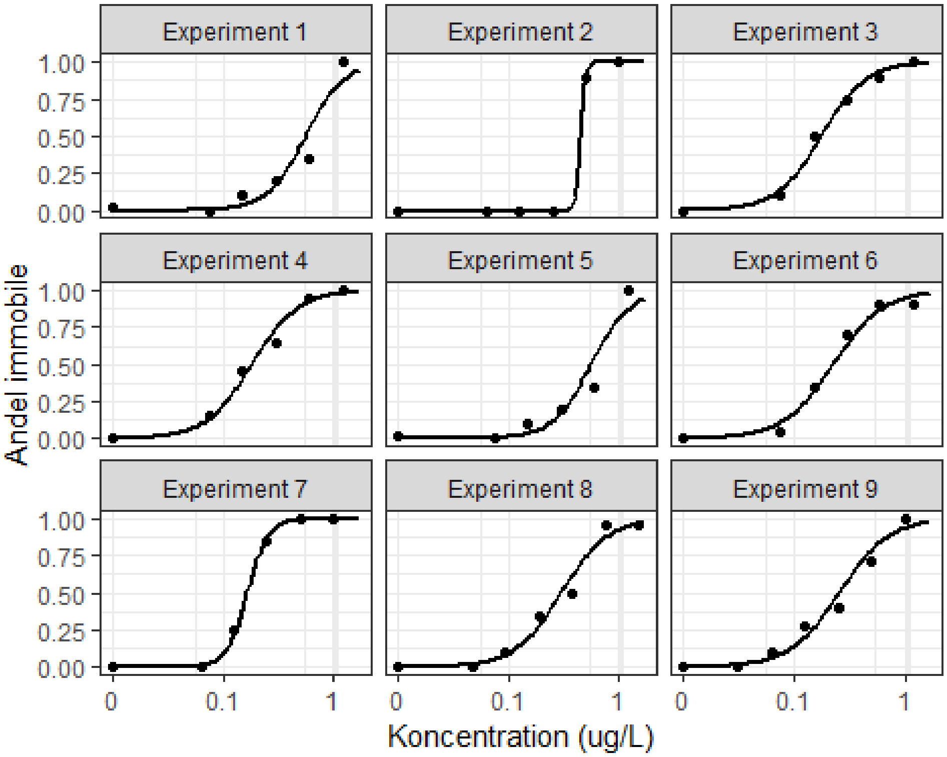
```
##         BMD      BMDL
##  0.04364098 0.0204931

alpha.m5.bmd <- bmd(alpha.m5, bmr = 0.05,
                    backgType = "modelBased", def = "excess")

##        BMD      BMDL
##  0.1696149 0.1004246

alpha.m6.bmd <- bmd(alpha.m6, bmr = 0.05,
                    backgType = "modelBased", def = "excess")

##         BMD       BMDL
##  0.04767556 0.02138498

alpha.m7.bmd <- bmd(alpha.m7, bmr = 0.05,
                    backgType = "modelBased", def = "excess")

##         BMD      BMDL
##  0.08687902 0.0640227

alpha.m8.bmd <- bmd(alpha.m8, bmr = 0.05,
                    backgType = "modelBased", def = "excess")

##         BMD       BMDL
##  0.07183345 0.03979943

alpha.m9.bmd <- bmd(alpha.m9, bmr = 0.05,
                    backgType = "modelBased", def = "excess")

##         BMD       BMDL
##  0.05533272 0.02903962
```

Plotter modellen for hvert af de 9 del-eksperimenter

```
newdata <- expand.grid(conc = exp(seq(log(0.01), log(1.7),
length = 100)))
pm <- predict(alpha.m1, newdata = newdata,
interval = "confidence")
newdata$p <- pm[, 1]
newdata$pmin <- pm[, 2]
newdata$pmax <- pm[, 3]

newdata2 <- expand.grid(conc = exp(seq(log(0.01), log(1.7),
length = 100)))
pm <- predict(alpha.m2, newdata = newdata2,
interval = "confidence")
newdata2$p <- pm[, 1]
newdata2$pmin <- pm[, 2]
newdata2$pmax <- pm[, 3]
```

```
newdata3 <- expand.grid(conc = exp(seq(log(0.01), log(1.7),
length = 100)))
pm <- predict(alpha.m3, newdata = newdata3,
interval = "confidence")
newdata3$p <- pm[, 1]
newdata3$pmin <- pm[, 2]
newdata3$pmax <- pm[, 3]

newdata4 <- expand.grid(conc = exp(seq(log(0.01), log(1.7),
length = 100)))
pm <- predict(alpha.m4, newdata = newdata4,
interval = "confidence")
newdata4$p <- pm[, 1]
newdata4$pmin <- pm[, 2]
newdata4$pmax <- pm[, 3]

newdata5 <- expand.grid(conc = exp(seq(log(0.01), log(1.7),
length = 100)))
pm <- predict(alpha.m5, newdata = newdata5,
interval = "confidence")
newdata5$p <- pm[, 1]
newdata5$pmin <- pm[, 2]
newdata5$pmax <- pm[, 3]

newdata6 <- expand.grid(conc = exp(seq(log(0.01), log(1.7),
length = 100)))
pm <- predict(alpha.m6, newdata = newdata6,
interval = "confidence")
newdata6$p <- pm[, 1]
newdata6$pmin <- pm[, 2]
newdata6$pmax <- pm[, 3]

newdata7 <- expand.grid(conc = exp(seq(log(0.01), log(1.7),
length = 100)))
pm <- predict(alpha.m7, newdata = newdata7,
interval = "confidence")
newdata7$p <- pm[, 1]
newdata7$pmin <- pm[, 2]
newdata7$pmax <- pm[, 3]

newdata8 <- expand.grid(conc = exp(seq(log(0.01), log(1.7),
length = 100)))
pm <- predict(alpha.m8, newdata = newdata8,
interval = "confidence")
newdata8$p <- pm[, 1]
newdata8$pmin <- pm[, 2]
```

```
newdata8$pmax <- pm[, 3]

newdata9 <- expand.grid(conc = exp(seq(log(0.01), log(1.7),
length = 100)))
pm <- predict(alpha.m9, newdata = newdata9,
interval = "confidence")
newdata9$p <- pm[, 1]
newdata9$pmin <- pm[, 2]
newdata9$pmax <- pm[, 3]


newdata[["Exp"]]<-"Exp1"
newdata2[["Exp"]]<-"Exp2"
newdata3[["Exp"]]<-"Exp3"
newdata4[["Exp"]]<-"Exp4"
newdata5[["Exp"]]<-"Exp5"
newdata6[["Exp"]]<-"Exp6"
newdata7[["Exp"]]<-"Exp7"
newdata8[["Exp"]]<-"Exp8"
newdata9[["Exp"]]<-"Exp9"


newdata.all<-rbind(newdata, newdata2, newdata3,
                   newdata4, newdata5, newdata6,
                   newdata7, newdata8, newdata9)

alpha.cyp$conc0 <- alpha.cyp$Dose
alpha.cyp$conc0[alpha.cyp$conc0 == 0] <- 0.01

alpha.cyp.red<-ddply(alpha.cyp, .(Exp, conc0), summarize,
      prop = mean(Immobile/Total))

exp.labs <- c("Experiment 1","Experiment 2","Experiment 3",
                              "Experiment 4","Experiment 5","Experiment 6",
                              "Experiment 7","Experiment 8","Experiment 9")
names(exp.labs) <- c("Exp1","Exp2","Exp3", "Exp4","Exp5","Exp6",
                              "Exp7","Exp8","Exp9")
```



Samler BMD, BMDL og SE fra alle del-eksperimenter i et datasæt

```
BMDList <- list(alpha.m1.bmd, alpha.m2.bmd, alpha.m3.bmd,
                alpha.m4.bmd, alpha.m5.bmd, alpha.m6.bmd,
                alpha.m7.bmd, alpha.m8.bmd, alpha.m9.bmd)

BMDVec<-sapply(BMDList, FUN = function(x) {x$Results[1]})
SEVec<-sapply(BMDList, FUN = function(x) {x$SE})
BMDLVec<-sapply(BMDList, FUN = function(x) {x$Results[2]})

step2Data<-data.frame(BMD = BMDVec,
                      BMDL = BMDLVec,
                      SE = SEVec,
                      Exp = seq(1,9))
```

Step2Data

```
step2Data

##          BMD        BMDL         SE Exp
## 1 0.16961488  0.10042458 0.04206472   1
## 2 0.38681802 -0.57609187 0.58540765   2
## 3 0.04421228  0.02135657 0.01389528   3
## 4 0.04364098  0.02049310 0.01407291   4
## 5 0.16961488  0.10042458 0.04206472   5
## 6 0.04767556  0.02138498 0.01598354   6
## 7 0.08687902  0.06402270 0.01389566   7
## 8 0.07183345  0.03979943 0.01947530   8
## 9 0.05533272  0.02903962 0.01598507   9
```

Fitter en meta-analytisk model til step2Data

```
meta.m1 <- rma(BMD,
              SE^2,
              data = step2Data,
              level = 0.9)
```

Finder kombineret BMD og BMDL (som nedre grænse af 90% konfidensintervallet)

```
summary(meta.m1)

## 
## Random-Effects Model (k = 9; tau^2 estimator: REML)
## 
##   logLik  deviance       AIC       BIC      AICc  
##  11.2866  -22.5733  -18.5733  -18.4144  -16.1733  
## 
## tau^2 (estimated amount of total heterogeneity): 0.0007 (SE = 0.0006)
## tau (square root of estimated tau^2 value):      0.0260
## I^2 (total heterogeneity / total variability):   65.62%
## H^2 (total variability / sampling variability):  2.91
## 
## Test for Heterogeneity:
## Q(df = 8) = 21.1721, p-val = 0.0067
## 
## Model Results:
## 
## estimate      se    zval    pval   ci.lb   ci.ub     
##   0.0705  0.0116  6.0573  <.0001  0.0514  0.0897  *** 
## 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Eksempel 5. Til-event data: Immobilisering af daphia som reaktion på alpha-cypermethrin

Datasættet Daphnia.csv kan findes på GitHub sammen med denne kode

```
Daphnia <- read.csv2(file.choose())
```

Der afprøves tre event-time modeller på data: En log-logistisk og to Weibull modeller. Modellerne fittes med en kurve for hver koncentration fratrukket koncentration 0 hvor der ikke er registreret nogle events.

```
m1 <- drm(Dead ~ Start + End, curveid = Conc, data = subset(Daphnia, Conc!=0),
          fct = LL.2(), type = "event")
m2 <- drm(Dead ~ Start + End, curveid = Conc, data = subset(Daphnia, Conc!=0),
          fct = W2.2(), type = "event")
m3 <- drm(Dead ~ Start + End, curveid = Conc, data = subset(Daphnia, Conc!=0),
          fct = W1.2(), type = "event")
```

Modellerne sammenlignes med AIC

```
AIC(m1,m2,m3)

##    df      AIC
## m1 35 474.4621
## m2 35 462.3731
## m3 35 488.5865
```

Model m2 har klart det bedste fit til data og er derfor den der bruges herfra.

Bestemmelde af t50 for alle kurver (koncentrationer) og den fælles varians-kovarians matrix.

```
names(m2$coefficients) <- c("b1","b2","b3","b4","b5",
                            "e1","e2","e3","e4","e5")

M <- mjust(list(m2,m2,m2,m2,m2),
           list("exp(log(-log(0.5))/b1+log(e1))",
                "exp(log(-log(0.5))/b2+log(e2))",
                "exp(log(-log(0.5))/b3+log(e3))",
                "exp(log(-log(0.5))/b4+log(e4))",
                "exp(log(-log(0.5))/b5+log(e5))"))

An <- M$covar
```

Konstruerer et nyt datasæt med koncentrationer og de fundne t50.

```
step2data <- data.frame(ED50 = M$coef[,1],
                        Conc = c(0.07,0.26,0.9,3.1,11))
```

Fitter en vægtet koncentration-respons model til de estimerede t50

```
step2model <- drm(ED50 ~ Conc, data = step2data, fct = W2.4(), varcov = An)
```

BMD hørende til BMR = 0.05 fundet ved brug af en relative definition og ved brug af semi-parameterisk bootstrap til BMDL

```
set.seed(12345678)
bmdBoot(step2model, 0.05, backgType = "modelBased",
        def = "relative", bootType = "semiparametric")

##       BMD      BMDL
##  0.148658 0.1285995
```

En simpel analyse der kun bruger data fra slut tidspunktet

```r
data.simple <- data.frame(Conc = c(0,0.07,0.26,0.9,3.1,11),
                          total = rep(24,6),
                          immobile = c(0,3,10,24,24,24))

m.simple <- drm(immobile/total ~ Conc, data = data.simple,
                type = "binomial", weights = total, fct = W2.2())

set.seed(12345678)
bmdBoot(m.simple, 0.05, backgType = "modelBased",
        def = "excess", bootType = "nonparametric")

##         BMD      BMDL
##  0.05038527 0.0246898
```

Eksempel 6. SSD

Koden her er kun kun angivet for det ene af de fire saponiner, quillaja. Alle datasæt er tilgængelige på GitHub samme sted som denne kode.

Først sættes et tomt datasæt op til senere brug. Heri fyldes resultaterne for de enkelte arter efterhånden.

```r
ResMatrix1 <- matrix(NA, nrow=12, ncol=7)
rownames(ResMatrix1) <- c("Vibrio fischeri",
                          "Pseudokirchneriella subcapitata",
                          "Lemna minor",
                          "Daphnia magna",
                          "Chironomus riparius",
                          "Chaoborus crystallinus",
                          "Gammarus pulex",
                          "Tubifex tubifex",
                          "Lumbriculus variegatus",
                          "Lymnaea stagnalis",
                          "Lymnaea stagnalis embryo",
                          "Danio rerio embryo")

colnames(ResMatrix1) <- c("ED50", "ED50 lower", "ED50 upper",
                          "BMD10", "BMDL10",
                          "BMD5","BMDL5")


ResMatrix1 <- as.data.frame(ResMatrix1)
```

Indlæser data for Vibrio fisheri, Vibero.csv

```r
data.vibero <- read.csv2(file.choose())
```

Fitter dosis-respons model. En to-parameter log-logistisk model hvor nedre grænse er 0 og øvre er 100.

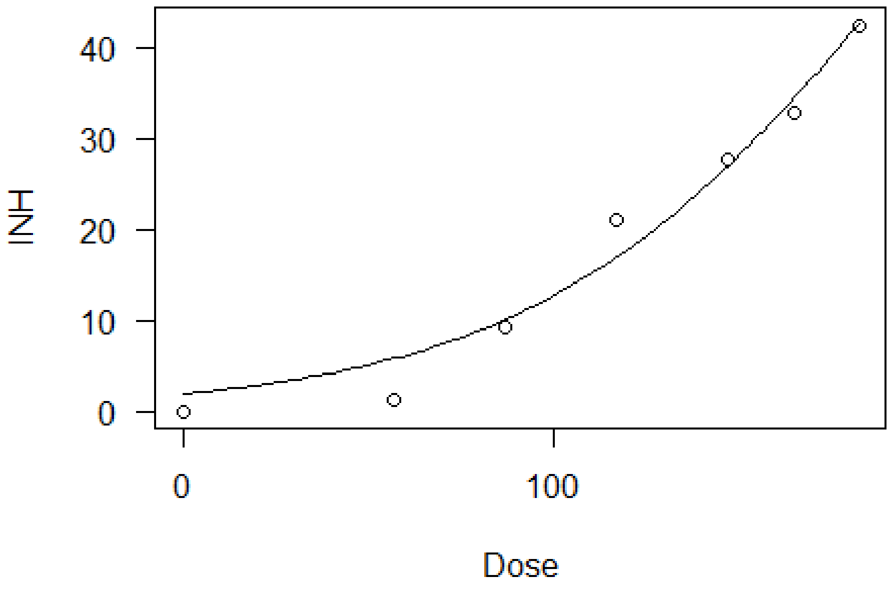
```
model.vibero <- drm(INH ~ Dose, data = subset(data.vibero, Curve=="QS"),
              fct = LL.3(fixed = c(NA,100,NA)))
```

Finder ED50

```
ED(model.vibero, c(50), interval = "delta")

##
## Estimated effective doses
##
##          Estimate Std. Error   Lower   Upper
## e:1:50   942.77     126.14  618.53 1267.01
```

Tjekker model fit

```
plot(model.vibero)
```



Bestemmer BMD hørende til BMR = 0.1 ved brug af hybrid metoden hvor 2 SD bruges som cutoff

```
a1 <- bmdBoot(model.vibero, bmr = 0.1,
              backgType = "hybridSD", def = "hybridAdd", backg = 2)

##       BMD     BMDL
##  12.63858 3.881976
```

```
ResMatrix1[1,1:3] <- ED(model.vibero, c(50), interval = "delta")[c(1,3,4)]

##
## Estimated effective doses
##
##        Estimate Std. Error   Lower   Upper
## e:1:50   942.77     126.14  618.53 1267.01

ResMatrix1[1,4:5] <- a1$Results
```

Indlæser data for Pseudokirchneriella subcapitata, Pseudokirchneriella.csv

```
data.pseudokirchneriella <- read.csv2(file.choose())
```

Fitter dosis-respons model. Der bruges en tre-parameter log-logistisk model

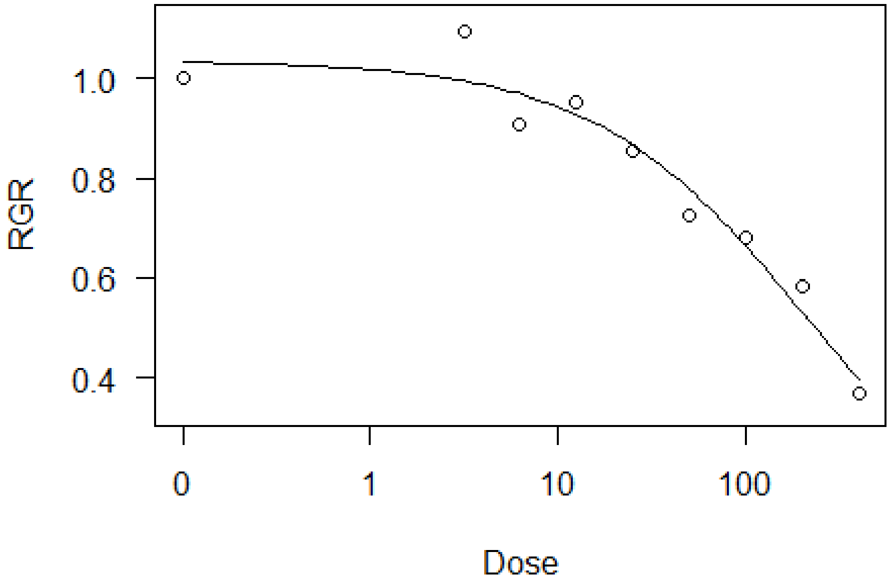
```
model.pseudokirchneriella <- drm(RGR ~ Dose,
                                 data = subset(data.pseudokirchneriella,
                                               Curve=="QS"),
                                 fct = LL.3())
```

Finder ED50

```
ED(model.pseudokirchneriella, c(50), interval = "delta")

##
## Estimated effective doses
##
##        Estimate Std. Error   Lower   Upper
## e:1:50  216.742     30.472 153.852 279.633
```

Tjekker model fit

```
plot(model.pseudokirchneriella)
```



Bestemmer BMD hørende til BMR = 0.1 ved brug af hybrid metoden hvor 2 SD bruges som cutoff

```
a2 <- bmdBoot(model.pseudokirchneriella, bmr = 0.1,
              backgType = "hybridSD", def="hybridAdd", backg = 2)

##       BMD     BMDL
##  5.035746 3.452908

ResMatrix1[2,1:3] <- ED(model.pseudokirchneriella, c(50),
                        interval = "delta")[c(1,3,4)]

##
## Estimated effective doses
##
##         Estimate Std. Error   Lower   Upper
## e:1:50  216.742     30.472 153.852 279.633

ResMatrix1[2,4:5] <- a2$Results
```

Indlæser data for Lemna minor, lemna.csv

```
data.lemna <- read.csv2(file.choose())
```

Fitter dosis-respons model. Der bruges en tre-parameter log-logistisk model

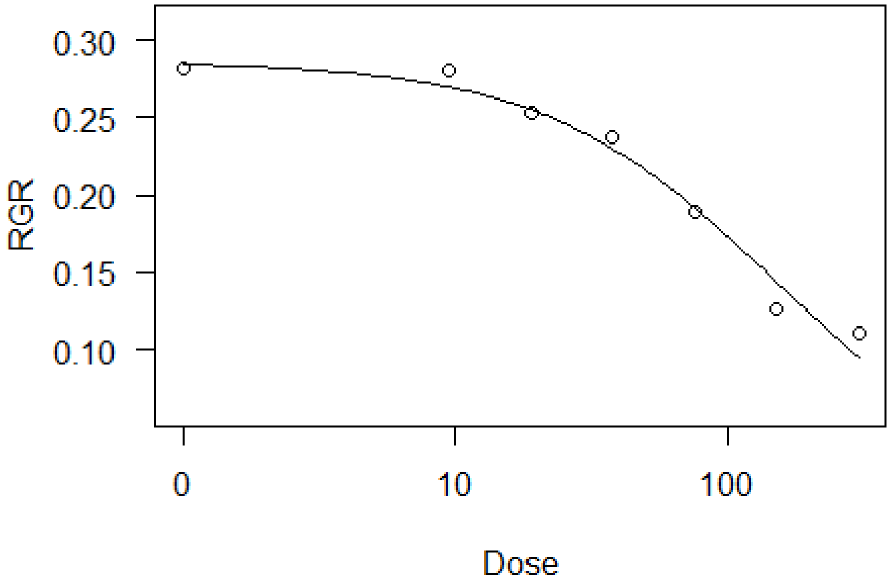
```
model.lemna <- drm(RGR ~ Dose,
                   data = subset(data.lemna, Curve=="QS"),
                   fct=LL.3())
```

Bestemmer ED50

```
ED(model.lemna, c(50), interval = "delta")

##
## Estimated effective doses
##
##        Estimate Std. Error   Lower   Upper
## e:1:50  152.936     16.544 119.613 186.258
```

Tjekker model fit

```
plot(model.lemna)
```



Bestemmer BMD hørende til BMR = 0.1 ved brug af hybrid metoden hvor 2 SD bruges som cutoff

```
a3 <- bmdBoot(model.lemna, bmr = 0.1,
              backgType = "hybridSD", def="hybridAdd", backg = 2)

##       BMD     BMDL
##  13.52413 7.180535
```

```
ResMatrix1[3,1:3] <- ED(model.lemna, c(50), interval = "delta")[c(1,3,4)]

##
## Estimated effective doses
##
##         Estimate Std. Error   Lower   Upper
## e:1:50  152.936     16.544 119.613 186.258

ResMatrix1[3,4:5] <- a3$Results
```

Indlæser data for Daphnia magma, DaphniaMagma.csv

```
data.daphnia <- read.csv2(file.choose())
```

Fitter dosis-respons model. To-parameter log-logistisk model

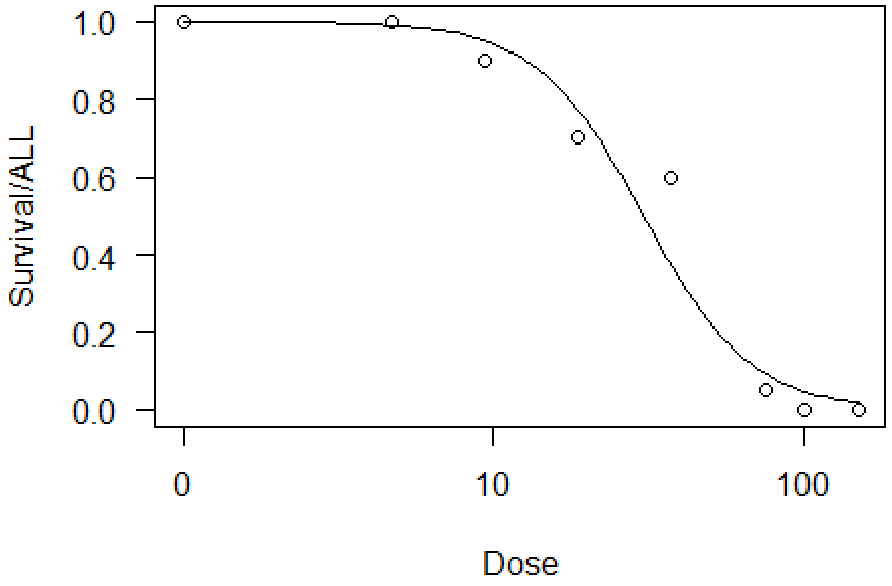
```
model.daphnia <- drm(Survival/ALL ~ Dose,
              data = subset(data.daphnia, Curve=="QS"),
              type="binomial", weights=ALL, fct=LL.2())
```

Finder ED50

```
ED(model.daphnia, c(50), interval = "delta")

##
## Estimated effective doses
##
##         Estimate Std. Error   Lower   Upper
## e:1:50  30.5399     3.4891 23.7013 37.3785
```

Tjekker model fit

```
plot(model.daphnia)
```



Bestemmer BMD hørende til BMR = 0.1 ved brug af "additional risk" definitionen

```
a4 <- bmdBoot(model.daphnia, bmr = 0.1,
              backgType = "modelBased", def="additional",
              bootType = "parametric")

##       BMD     BMDL
##  12.74404 6.415126

ResMatrix1[4,1:3] <- ED(model.daphnia, c(50), interval = "delta")[c(1,3,4)]

## 
## Estimated effective doses
## 
##        Estimate Std. Error   Lower   Upper
## e:1:50  30.5399     3.4891 23.7013 37.3785

ResMatrix1[4,4:5] <- a4$Results
```

Indlæser data for Chironomus, Chironomous.csv

```
data.chironomous <- read.csv2(file.choose())
```

Fitter dosis-respons model. To-parameter log-logistisk model

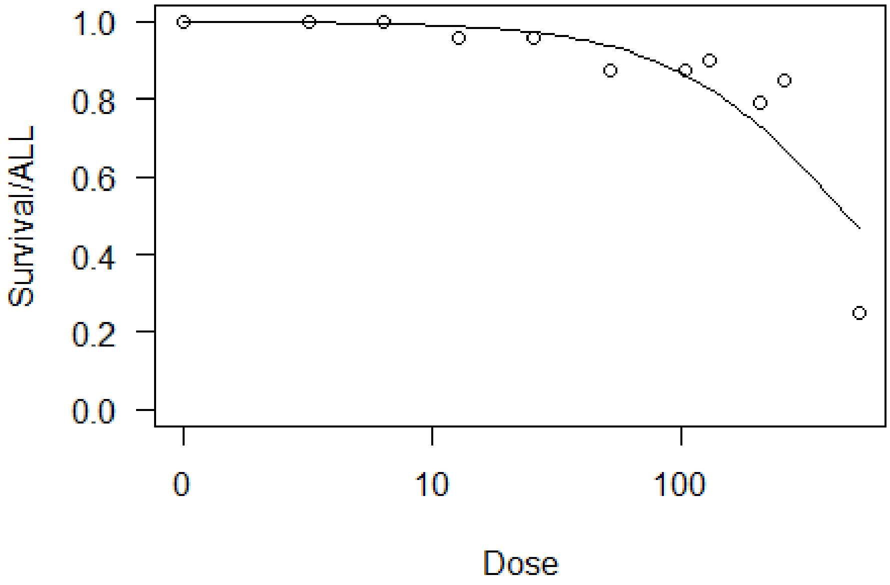
```
model.chironomous <- drm(Survival/ALL ~ Dose,
                         data = subset(data.chironomous, Curve=="QS"),
                         type="binomial", weights=ALL, fct=LL.2())
```

Finder ED50

```
ED(model.chironomous, c(50), interval = "delta")

##
## Estimated effective doses
##
##        Estimate Std. Error  Lower  Upper
## e:1:50   465.57     118.35 233.61 697.54
```

Tjekker model fit

```
plot(model.chironomous)
```



Bestemmer BMD hørende til BMR = 0.1 ved brug af “additional risk” definitionen

```
a5 <- bmdBoot(model.chironomous, bmr = 0.1,
              backgType = "modelBased", def="additional",
              bootType = "parametric")

##       BMD    BMDL
##  77.72825 20.9865
```

```
ResMatrix1[5,1:3]<-ED(model.chironomous, c(50), interval = "delta")[c(1,3,4)]

##
## Estimated effective doses
##
##         Estimate Std. Error  Lower  Upper
## e:1:50   465.57     118.35 233.61 697.54

ResMatrix1[5,4:5]<-a5$Results
```

Indlæser data for Chaoborus crystallinus, Chaoborus.csv

```
data.chaoborus <- read.csv2(file.choose())
```

Fitter dosis-respons model. To-parameter log-logistisk model

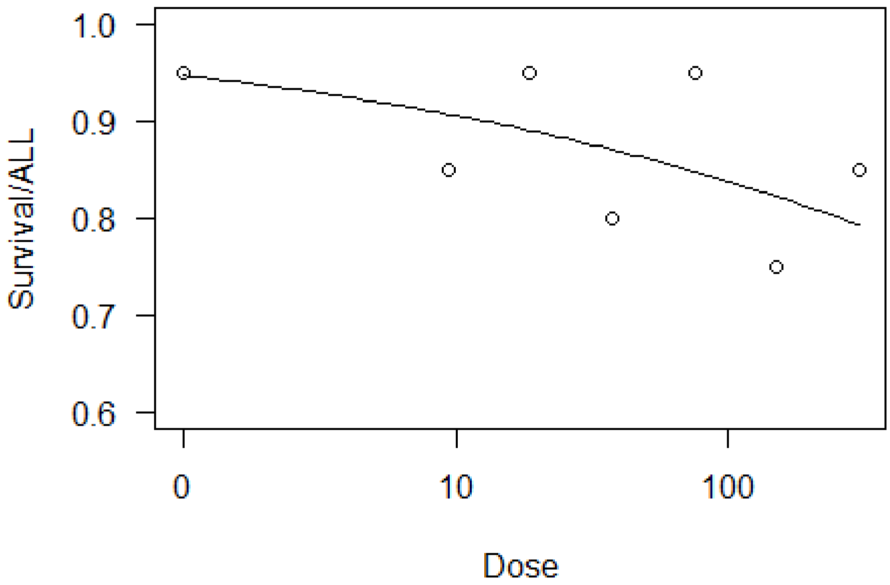
```
model.chaoborus <- drm(Survival/ALL ~ Dose,
                       data = subset(data.chaoborus, Curve == "QS"),
                       type = "binomial", weights = ALL, fct = LL.2())
```

Finder ED50

```
ED(model.chaoborus, c(50), interval = "delta")

##
## Estimated effective doses
##
##         Estimate Std. Error   Lower   Upper
## e:1:50    44025     110779 -173098  261148
```

Tjekker model fit

```
plot(model.chaoborus)
```



Bestemmer BMD hørende til BMR = 0.1 ved brug af “additional risk” definitionen

```
a6 <- bmdBoot(model.chaoborus, bmr = 0.1,
              backgType = "modelBased", def = "additional",
              bootType = "parametric")

##      BMD         BMDL
##  12.7851 9.444163e-07

ResMatrix1[6,1:3]<-ED(model.chaoborus, c(50), interval = "delta")[c(1,3,4)]

##
## Estimated effective doses
##
##         Estimate Std. Error   Lower   Upper
## e:1:50    44025     110779 -173098  261148



ResMatrix1[6,4:5]<-a6$Results
```

Indlæser data for Gammarus pulex, Gammarus.csv

```
data.gammarus <- read.csv2(file.choose())
```

Fitter dosis-respons model. To-parameter log-logistisk model. Her er startværdier nødvendige for at opnå konvergens af modellen

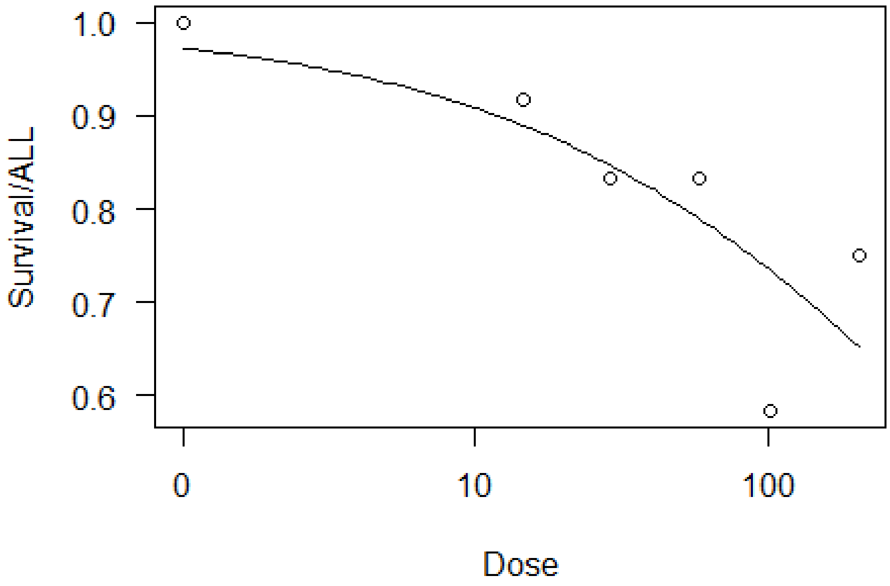
```
model.gammarus <- drm(Survival/ALL ~ Dose,
                      data = subset(data.gammarus, Curve == "QS"),
                      type = "binomial", weights = ALL,
                      fct = LL.2(), start = c(0.05,33))
```

Finder ED50

```
ED(model.gammarus, c(50), interval = "delta")

##
## Estimated effective doses
##
##         Estimate Std. Error     Lower    Upper
## e:1:50    639.29     942.95 -1208.85  2487.42
```

Tjekker model fit

```
plot(model.gammarus)
```



Bestemmer BMD hørende til BMR = 0.1 ved brug af “additional risk” definitionen

```
a7 <- bmdBoot(model.gammarus, bmr = 0.1,
              backgType = "modelBased", def="additional",
              bootType = "parametric")

##       BMD      BMDL
##  11.95775 0.7164563

ResMatrix1[7,1:3]<-ED(model.gammarus,c(50), interval = "delta")[c(1,3,4)]

##
## Estimated effective doses
##
##         Estimate Std. Error    Lower    Upper
## e:1:50   639.29     942.95 -1208.85  2487.42

ResMatrix1[7,4:5]<-a7$Results
```

Indlæser data for Tubifex tubifex, Tubifex.csv

```
data.tubifex <- read.csv2(file.choose())
```

Fitter dosis-respons model. To-parameter log-logistisk model

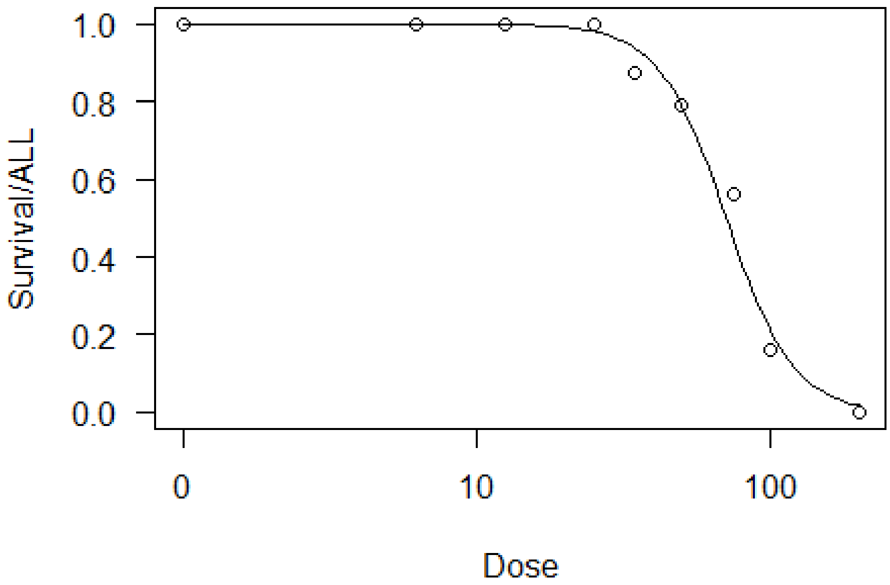
```
model.tubifex <- drm(Survival/ALL ~ Dose,
                     data = subset(data.tubifex, Curve == "QS"),
                     type = "binomial", weights = ALL, fct = LL.2())
```

Finder ED50

```
ED(model.tubifex, c(50), interval = "delta")

##
## Estimated effective doses
##
##         Estimate Std. Error   Lower   Upper
## e:1:50  70.8059     5.5555 59.9174 81.6945
```

Tjekker model fit

```
plot(model.tubifex)
```



Bestemmer BMD hørende til BMR = 0.1 ved brug af “additional risk” definitionen

```
a8 <- bmdBoot(model.tubifex, bmr = 0.1, backgType = "modelBased",
              def="additional", bootType = "parametric")

##       BMD     BMDL
##  39.80378 16.70855

ResMatrix1[8,1:3] <- ED(model.tubifex, c(50), interval = "delta")[c(1,3,4)]

##
## Estimated effective doses
##
##         Estimate Std. Error   Lower   Upper
## e:1:50  70.8059     5.5555 59.9174 81.6945

ResMatrix1[8,4:5] <- a8$Results
```

Indlæser data for Lumbriculus variegatus, Lumbriculus.csv

```
data.lumbriculus <- read.csv2(file.choose())
```

Fitter dosis-respons model. To-parameter log-logistisk model

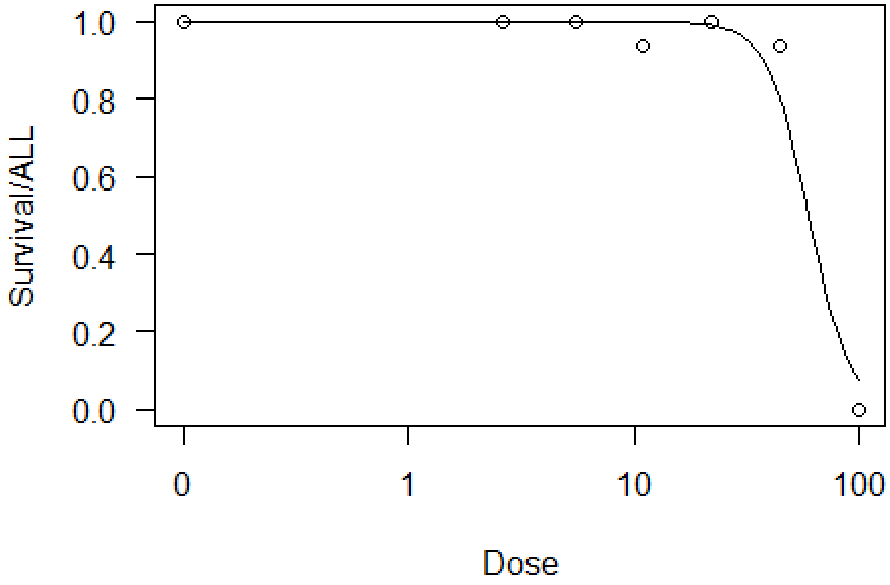
```r
model.lumbriculus <- drm(Survival/ALL ~ Dose,
                         data = subset(data.lumbriculus, Curve == "QS"),
                         type = "binomial", weights = ALL, fct = LL.2())
```

Finder ED50

```r
ED(model.lumbriculus, c(50), interval = "delta")

##
## Estimated effective doses
##
##          Estimate Std. Error   Lower   Upper
## e:1:50   59.5628     5.9725 47.8570 71.2687
```

Tjekker model fit

```r
plot(model.lumbriculus)
```



Bestemmer BMD hørende til BMR = 0.1 ved brug af "additional risk" definitionen

```r
a9 <- bmdBoot(model.lumbriculus, bmr = 0.1, backgType = "modelBased",
              def = "additional", bootType = "parametric")

##        BMD     BMDL
##   37.68787 8.601496
```

```
ResMatrix1[9,1:3] <- ED(model.lumbriculus, c(50), interval = "delta")[c(1,3,4)]

##
## Estimated effective doses
##
##        Estimate Std. Error   Lower   Upper
## e:1:50  59.5628     5.9725 47.8570 71.2687

ResMatrix1[9,4:5] <- a9$Results
```

Indlæser data for Lymnaea stagnalis, Lymnaea.csv

```
data.lymnaea <- read.csv2(file.choose())
```

Fitter dosis-respons model. To-parameter log-logistisk model

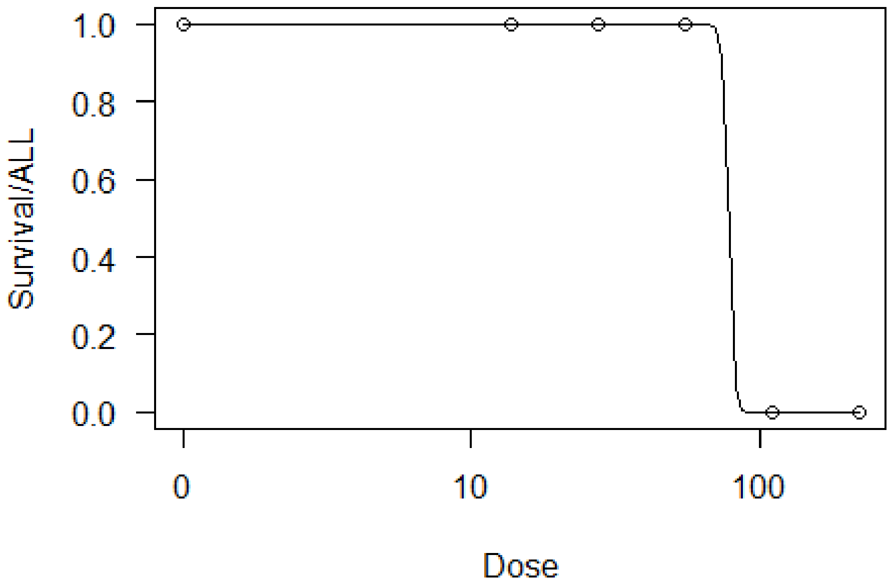
```
model.lymnaea <- drm(Survival/ALL ~ Dose,
                     data = subset(data.lymnaea, Curve == "QS"),
                     type = "binomial", weights = ALL, fct = LL.2())
```

Finder ED50

```
ED(model.lymnaea, c(50), interval = "delta")

##
## Estimated effective doses
##
##         Estimate Std. Error     Lower     Upper
## e:1:50    78.183    662.489 -1220.272  1376.637
```

Tjekker model fit

```
plot(model.lymnaea)
```



Bestemmer BMD hørende til BMR = 0.1 ved brug af “additional risk” definitionen

```
a10 <- bmdBoot(model.lymnaea, bmr = 0.1, backgType = "modelBased",
               def = "additional", bootType = "parametric")

##        BMD     BMDL
##   74.45165 28.76417

ResMatrix1[10,1:3] <- ED(model.lymnaea, c(50), interval = "delta")[c(1,3,4)]

##
## Estimated effective doses
##
##          Estimate Std. Error     Lower     Upper
## e:1:50    78.183    662.489 -1220.272  1376.637

ResMatrix1[10,4:5] <- a10$Results
```

Indlæser data for Lymnaea stagnalis embryo, EmbryoLymnaea.csv

```
data.embryoLymnaea <- read.csv2(file.choose())
```

Fitter dosis-respons model. To-parameter log-logistisk model

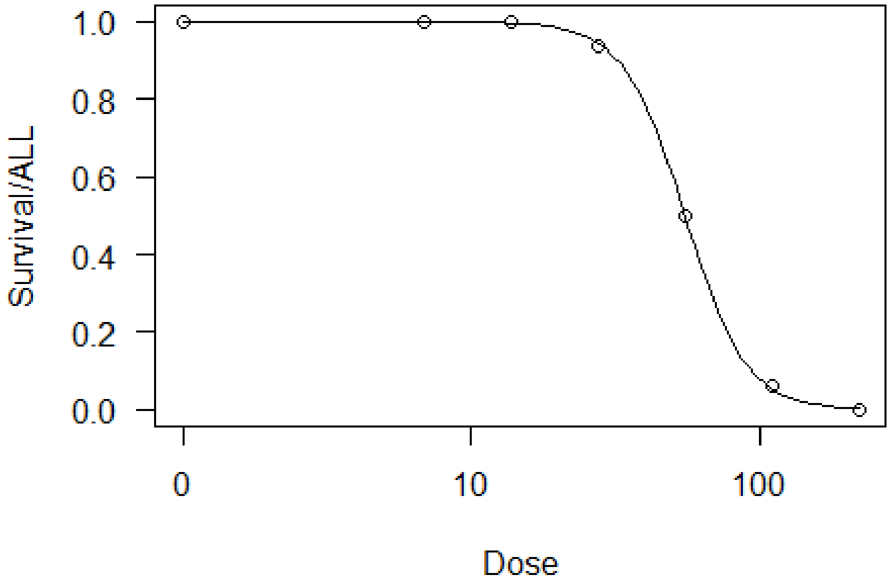
```
model.embryoLymnaea <- drm(Survival/ALL ~ Dose,
                           data = subset(data.embryoLymnaea, Curve == "QS"),
                           type = "binomial", weights = ALL, fct = LL.2())
```

Finder ED50

```
ED(model.embryoLymnaea, c(50), interval = "delta")

##
## Estimated effective doses
##
##         Estimate Std. Error   Lower   Upper
## e:1:50  55.0075     5.5873 44.0565 65.9584
```

Tjekker model fit

```
plot(model.embryoLymnaea)
```



Bestemmer BMD hørende til BMR = 0.1 ved brug af “additional risk” definitionen

```
a11 <- bmdBoot(model.embryoLymnaea, bmr = 0.1, backgType = "modelBased",
               def = "additional", bootType = "parametric")

##       BMD    BMDL
##   32.15759 19.8894
```

```
ResMatrix1[11,1:3] <- ED(model.embryoLymnaea, c(50), interval = "delta")[c(1,3,4)]

##
## Estimated effective doses
##
##        Estimate Std. Error   Lower   Upper
## e:1:50  55.0075     5.5873 44.0565 65.9584

ResMatrix1[11,4:5] <- a11$Results
```

Indlæser data for Danio rerio embryo, Danio.csv

```
data.danio <- read.csv2(file.choose())
```

Fitter dosis-respons model. To-parameter log-logistisk model

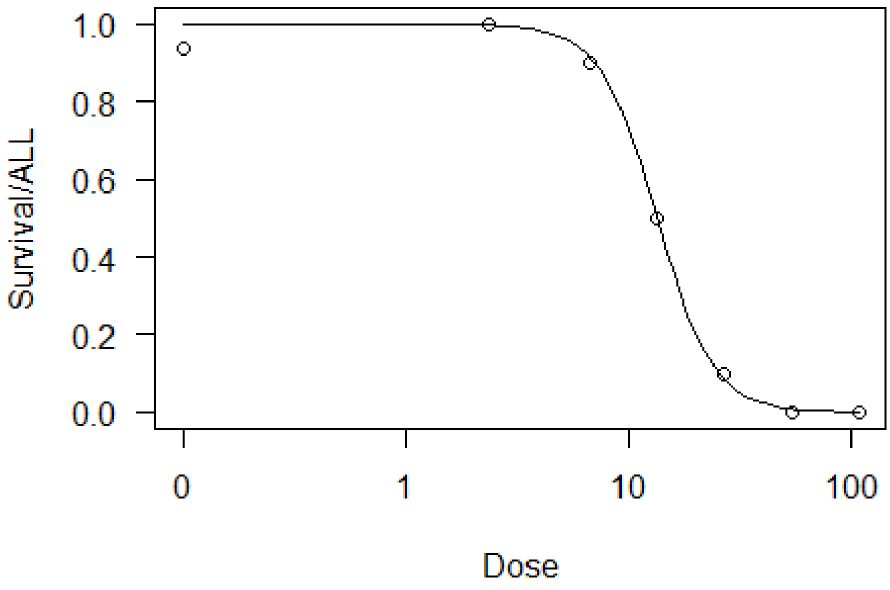
```
model.danio <- drm(Survival/ALL ~ Dose,
                   data = subset(data.danio, Curve == "QS"),
                   type = "binomial", weights = ALL, fct = LL.2())
```

Finder ED50

```
ED(model.danio, c(50), interval = "delta")

##
## Estimated effective doses
##
##        Estimate Std. Error   Lower   Upper
## e:1:50  13.4263     1.9280  9.6474 17.2051
```

Tjekker model fit

```
plot(model.danio)
```



Bestemmer BMD hørende til BMR = 0.1 ved brug af "additional risk" definitionen

```
a12 <- bmdBoot(model.danio, bmr = 0.1, backgType = "modelBased",
               def = "additional", bootType = "parametric")

##       BMD     BMDL
##  7.035372 3.522241

ResMatrix1[12,1:3] <- ED(model.danio, c(50), interval = "delta")[c(1,3,4)]

##
## Estimated effective doses
##
##        Estimate Std. Error   Lower   Upper
## e:1:50  13.4263     1.9280  9.6474 17.2051

ResMatrix1[12,4:5] <- a12$Results
```

For at fitte SSD

Tildeler værdier større end 10000 værdien 10000

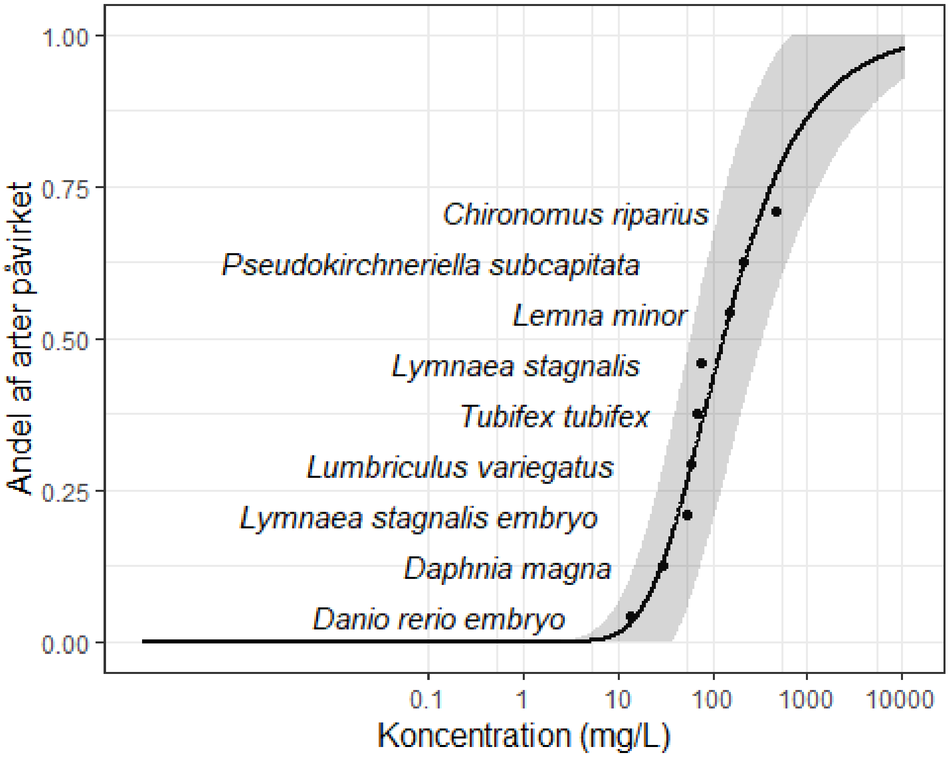
```
ResMatrix1$names <- rownames(ResMatrix1)
data.ssd <- ResMatrix1
data.ssd$ED50[data.ssd$ED50>10000]<-10000
```

Fitter SSD model i form af en generaliseret log-logistisk model (en fem-parameter Burr type III, med to parametre holdt fast) til ED50

```
ssd.m1 <- drm(~ ED50, data = data.ssd, type="ssd",
              fct = LL.5(fixed = c(NA, 0, 1, NA, NA)))
```

Plot af den fittede SSD kurve

```
newdata1 <- expand.grid(Dose = exp(seq(log(0.0001), log(11000), length = 1000)))
pm <- predict(ssd.m1, newdata = newdata1, interval = "confidence")
newdata1$p <- pm[, 1]
newdata1$pmin <- pm[, 2]
newdata1$pmax <- pm[, 3]

data.ssd1 <- arrange(data.ssd,ED50)
data.ssd1$proportion <- (1:length(data.ssd1$ED50))/length(data.ssd1$ED50)-
1/length(data.ssd1$ED50)*0.5
data.ssd1 <- subset(data.ssd1,ED50<600)

QuillajaSED50 <- ggplot(data.ssd1, aes(x = ED50, y = proportion)) +
  geom_point() +
  geom_line(data = newdata1, aes(x = Dose, y=p), size=1) +
  geom_text(aes(label=names),hjust=1.25, fontface="italic", size=4)+
  theme_bw()+
  geom_ribbon(data=newdata1, aes(x=Dose, y=p, ymin=pmin, ymax=pmax),alpha=0.2)+
  coord_trans(x = "log") +
  scale_x_continuous(breaks=c(0.0001, 0.1, 1,10,100, 1000,10000),
                     labels=c(0, 0.1, 1, 10,100, 1000,10000))+
  theme(axis.title = element_text(size=12))+
  ylab("Andel af arter påvirket") + xlab("Koncentration (mg/L)")+
  ylim(0,1)

## Warning: Ignoring unknown aesthetics: y

QuillajaSED50
```



Bestemmelse af hazard koncentrationer

```
ED(ssd.m1, c(5, 10, 50), interval = "inv")

##
## Estimated effective doses
##
##         Estimate    Lower    Upper
## e:1:5    17.2033   8.8627  52.6841
## e:1:10   25.2324  13.1735  67.5205
## e:1:50  131.1156  59.1366 337.7503
```

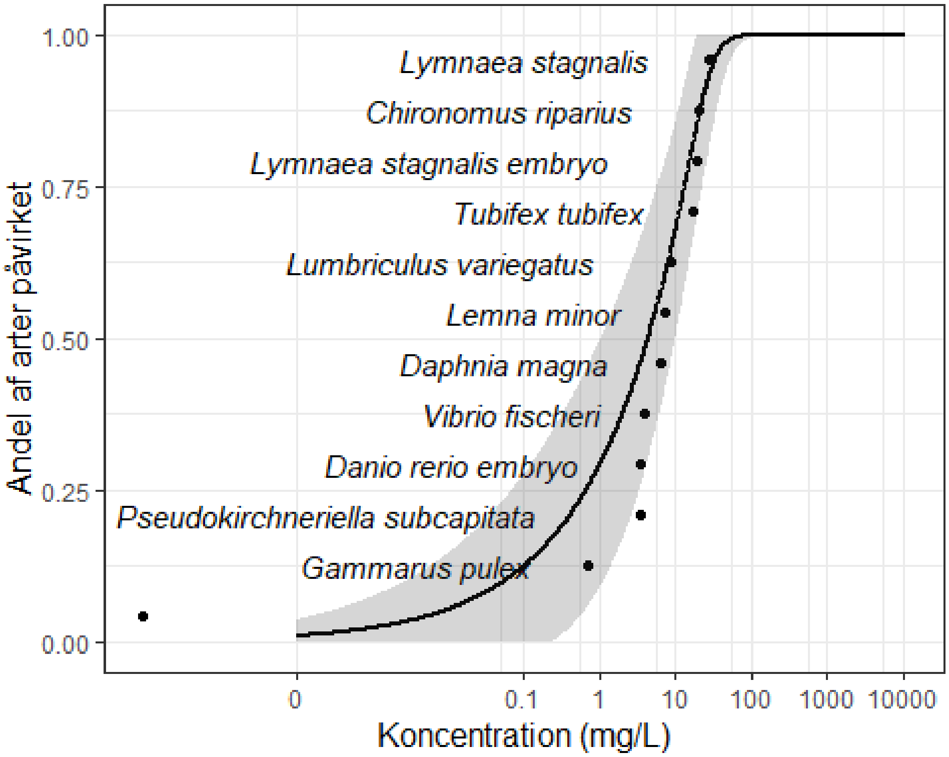
Fitter SSD model i form af en generaliseret log-logistisk model (en fem-parameter Burr type III, med to parametre holdt fast) til BMD hørende til BMR = 0.1.

```
data.ssd <-subset(ResMatrix1, BMD10<10000 & BMDL10>0)
ssd.m2 <- drm(~ BMDL10, data = data.ssd, type="ssd",
              fct = LL.5(fixed = c(NA, 0, 1, NA, NA)))
```

Plotter den fittede model

```
newdata1 <- expand.grid(Dose = exp(seq(log(0.0001), log(11000), length = 1000)))
pm <- predict(ssd.m2, newdata = newdata1, interval = "confidence")
newdata1$p <- pm[, 1]
newdata1$pmin <- pm[, 2]
newdata1$pmax <- pm[, 3]
```

```
#data.ssd$names <- rownames(data.ssd)
data.ssd1 <- arrange(data.ssd,BMDL10)
data.ssd1$proportion <- (1:length(data.ssd1$BMDL10))/length(data.ssd1$BMDL10)-
1/length(data.ssd1$BMDL10)*0.5

QuillajaSBMDL10 <- ggplot(data.ssd1, aes(x = BMDL10, y = proportion)) +
  geom_point() +
  geom_line(data = newdata1, aes(x = Dose, y=p), size=1) +
  geom_text(aes(label=names),hjust=1.25, fontface="italic", size=4)+
  theme_bw()+
  geom_ribbon(data=newdata1, aes(x=Dose, y=p, ymin=pmin, ymax=pmax),alpha=0.2)+
  coord_trans(x = "log") +
  scale_x_continuous(breaks=c(0.0001, 0.1, 1,10,100, 1000, 10000),
                     labels=c(0, 0.1, 1, 10,100, 1000, 10000))+
  theme(axis.title = element_text(size=12))+
  ylab("Andel af arter påvirket") + xlab("Koncentration (mg/L)")+
  ylim(0,1)

## Warning: Ignoring unknown aesthetics: y

QuillajaSBMDL10
```



Bestemmer hazard koncentrationer

```
ED(ssd.m2, c(5, 10, 50), interval = "inv")
```

```
##
## Estimated effective doses
##
##           Estimate      Lower      Upper
## e:1:5  0.00806635 0.00029835 0.62190319
## e:1:10 0.05346562 0.00273642 1.12200223
## e:1:50 4.31875964 1.03193757 9.92209929
```