Dose-Time Response Modeling of Rat Brain AChE Activity: Primicarb Gavage Dosing

June 24, 2005

1 Preamble

Here is some code to set up the analysis: loading required libraries and datasets, and defining some functions.

First, `CarbamateData` loads the full dataset for this risk assessment, and causes the library `DRUtils` to be loaded.

```
> library(CarbamateData)

Loading required package: DRUtils
Loading required package: nlme
Loading required package: tcltk
```

Set up lattice to use B&W instead of color:

```
> library(lattice)
> ltheme <- canonical.theme(color = FALSE)
> ltheme$strip.background$col <- "transparent"
> lattice.options(default.theme = ltheme)
```

Use package `Hmisc` for some formatting support.

```
> library(Hmisc)

Hmisc library by Frank E Harrell Jr

Type library(help='Hmisc'), ?Overview, or ?Hmisc.Overview')
to see overall documentation.

NOTE:Hmisc no longer redefines [.factor to drop unused levels when
subsetting.  To get the old behavior of Hmisc type dropUnusedLevels().

Attaching package 'Hmisc':


        The following object(s) are masked from package:stats :

         ecdf
```

All the rat gavage data for this analysis are in `AggData`. The following code prints out documentation for the datasets in use:

```
> printDataDoc(AggData)
```

```
----------------------------------------
Data set: AggData

Dataset creation date: Wed Jun 22 12:07:24 2005
Script name: /home/setzer/tasks/CarbamateCRA/EmpiricalDoseResponsesFinal/Data/getdata.R
Script last modified: 2005-06-22 12:05:47
sysname: Linux
release: 2.4.24-openmosix2
version: #1 Wed Jul 28 14:33:52 CEST 2004
nodename: node24.cluster2.org
machine: i686
login: unknown
user: setzer
----------------------------------------
```

The following function turns out to be quite useful on subsetted dataframes. It just eliminates unused levels of all factors in the data frame:

```
> CleanUp <- function(x) {
+     for (nm in names(x)) {
+         if (is.factor(x[, nm]))
+             x[, nm] <- factor(x[, nm])
+     }
+     x
+ }
```

To get starting values, we often have to extract values from a previously fit model. The following function simplifies that. The argument `what` is a regular expression:

```
> getParms <- function(what, Par) {
+     Par[grep(what, names(Par))]
+ }
```

This script is for modeling the dose-time response for rat brain. via gavage dosing. It includes acute and subchronic studies.

All the data used for this DR model are from mrids 113638, 44233103, and 44485301, in dataset `AggData`.

Now, set up the analysis dataset.

```
> dta <- CleanUp(subset(AggData, chemical %in% "PIRIMICARB" & species %in%
+     "RAT" & dsmtd %in% "GAVAGE" & sctn %in% "WHOLEBRAIN" & !is.na(cheact) &
+     (n == 1 | !is.na(sd)), select = c("cheact", "sd", "n", "dose",
+     "tmonstdy", "tmpstds", "sex", "mrid", "species")))
```

Summary of the relevant variables in this dataset:

```
> by(dta, dta$mrid, summary)

dta$mrid: 113638
     cheact            sd              n          dose           tmonstdy
 Min.   :25.00   Min.   :1.000   Min.   :5   Min.   : 0.000   Min.   :28
 1st Qu.:26.50   1st Qu.:3.000   1st Qu.:5   1st Qu.: 0.000   1st Qu.:28
 Median :30.00   Median :3.000   Median :5   Median : 1.500   Median :28
 Mean   :30.12   Mean   :3.125   Mean   :5   Mean   : 3.875   Mean   :28
 3rd Qu.:33.00   3rd Qu.:3.250   3rd Qu.:5   3rd Qu.: 5.375   3rd Qu.:28
 Max.   :38.00   Max.   :5.000   Max.   :5   Max.   :12.500   Max.   :28
    tmpstds  sex         mrid   species
 Min.   :1   F:4   113638  :8   RAT:8
```

```
 1st Qu.:1   M:4   44233103:0
 Median :1         44485301:0
 Mean   :1
 3rd Qu.:1
 Max.   :1
------------------------------------------------------------
dta$mrid: 44233103
     cheact            sd              n          dose       tmonstdy
 Min.   :32.10   Min.   :2.800   Min.   :5   Min.   :25   Min.   :84
 1st Qu.:33.12   1st Qu.:2.925   1st Qu.:5   1st Qu.:25   1st Qu.:84
 Median :34.15   Median :3.050   Median :5   Median :25   Median :84
 Mean   :34.15   Mean   :3.050   Mean   :5   Mean   :25   Mean   :84
 3rd Qu.:35.17   3rd Qu.:3.175   3rd Qu.:5   3rd Qu.:25   3rd Qu.:84
 Max.   :36.20   Max.   :3.300   Max.   :5   Max.   :25   Max.   :84
    tmpstds    sex         mrid   species
 Min.   :672   F:1   113638  :0   RAT:2
 1st Qu.:672   M:1   44233103:2
 Median :672         44485301:0
 Mean   :672
 3rd Qu.:672
 Max.   :672
------------------------------------------------------------
dta$mrid: 44485301
     cheact            sd               n              dose          tmonstdy
 Min.   : 9.68   Min.   :0.1500   Min.   :3.000   Min.   :  0.0   Min.   :1
 1st Qu.:11.81   1st Qu.:0.5000   1st Qu.:4.750   1st Qu.:  7.5   1st Qu.:1
 Median :12.19   Median :0.8350   Median :5.000   Median : 25.0   Median :1
 Mean   :12.08   Mean   :0.9413   Mean   :4.625   Mean   : 40.0   Mean   :1
 3rd Qu.:12.71   3rd Qu.:1.0225   3rd Qu.:5.000   3rd Qu.: 57.5   3rd Qu.:1
 Max.   :13.84   Max.   :2.8400   Max.   :5.000   Max.   :110.0   Max.   :1
    tmpstds      sex         mrid    species
 Min.   :  3.0   F:8   113638  : 0   RAT:16
 1st Qu.:  3.0   M:8   44233103: 0
 Median :169.5         44485301:16
 Mean   :169.5
 3rd Qu.:336.0
 Max.   :336.0

> unique(subset(dta, select = c("mrid", "tmonstdy", "tmpstds")))

         mrid tmonstdy tmpstds
1235   113638       28       1
1271 44233103       84     672
1281 44485301        1       3
1285 44485301        1     336

> by(dta, dta$mrid, function(x) with(x, table(dose, tmonstdy)))

dta$mrid: 113638
      tmonstdy
dose   28
  0    4
  3    2
  12.5 2
------------------------------------------------------------
```

```
dta$mrid: 44233103
    tmonstdy
dose 84
  25 2
------------------------------------------------------------
dta$mrid: 44485301
     tmonstdy
dose  1
  0   4
  10  4
  40  4
  110 4
```

In study 44485301, there is a 336 hour time point (after dosing). Drop it>

```
> dta <- CleanUp(subset(dta, tmpstds < 330))
```

In study 44233103, only the 25 mg/kg dose group was examined for brain activity, with no concurrent control measurements, so it is useless for dose-response modeling. Drop it:

```
> dta <- CleanUp(subset(dta, !mrid %in% c("44233103")))
```

Use `PhonyDF()` to set up a pseudo-individual dataset.

```
> idta <- with(dta, PhonyDF(dose, n, cheact, sd, "dose", "cheact",
+     dta[, c("mrid", "sex", "tmonstdy", "tmpstds")]))
```

Create some new factors, for splitting up the background parameters and allowing groups to have different variances. Brain AChE was assayed only once in each study.

```
> idta$mridXsex <- with(idta, interaction(mrid, tmonstdy, sex,
+     drop = TRUE, sep = ":"))
```

We have no useful time course data, so `lTr` will be fixed at log(1.5 hours), and `delta` at 0.5 hours, as we have done in all the other chemical.

2 Dose-Response Modeling

2.1 strategy

Use the model with simple exponential recovery (`tcmfn4()`). It looks as if the time to peak effect for all these chemicals is likely to be less than a half-hour (the earliest time point in this dataset), so the exponential recovery model is probably indistinguishable from the one with the more complex time course. Since there is no time course data, this is dose-response modeling, complicated by the fact that the responses are taken at different times after dosing (1 hour in two studies, 3 hours in the other).

One of these studies looked at the same animals at multiple time points. It would make sense to think of animal and time point as random effects, crossed with each other, and nested within MRID. That model may be fittable with nlme(), but it is complicated. Let's first model just the repeated measures, treating mrid and time in mrid as nested fixed effects. Test whether the differences among times within mrid are different. If they are, then we need to first look to see if there is a trend, and, if not, try to fit the model with nested and crossed effects.

Furthermore, in the RBC compartment, there was no evidence of AChE inhibition due to primicarb exposure. Before trying to fit the more complicated dose-response model used for the other chemicals, fit a simple, unconstrained exponential model to use as a trend test:

```
> drmod <- deriv(~exp(lA - m * x), c("lA", "m"), c("x", "lA", "m"))
> lA.start <- lm(I(log(cheact)) ~ mridXsex - 1, data = CleanUp(subset(idta,
+     dose %in% 0)))
> Start <- c(coef(lA.start), rep(0, nlevels(idta$mridXsex)))
> drmod0 <- gnls(cheact ~ drmod(dose, lA = lA, m = m), data = idta,
+     params = list(lA ~ mridXsex - 1, m ~ mridXsex - 1), start = Start,
+     weights = varIdent(form = ~1 | mridXsex))
```

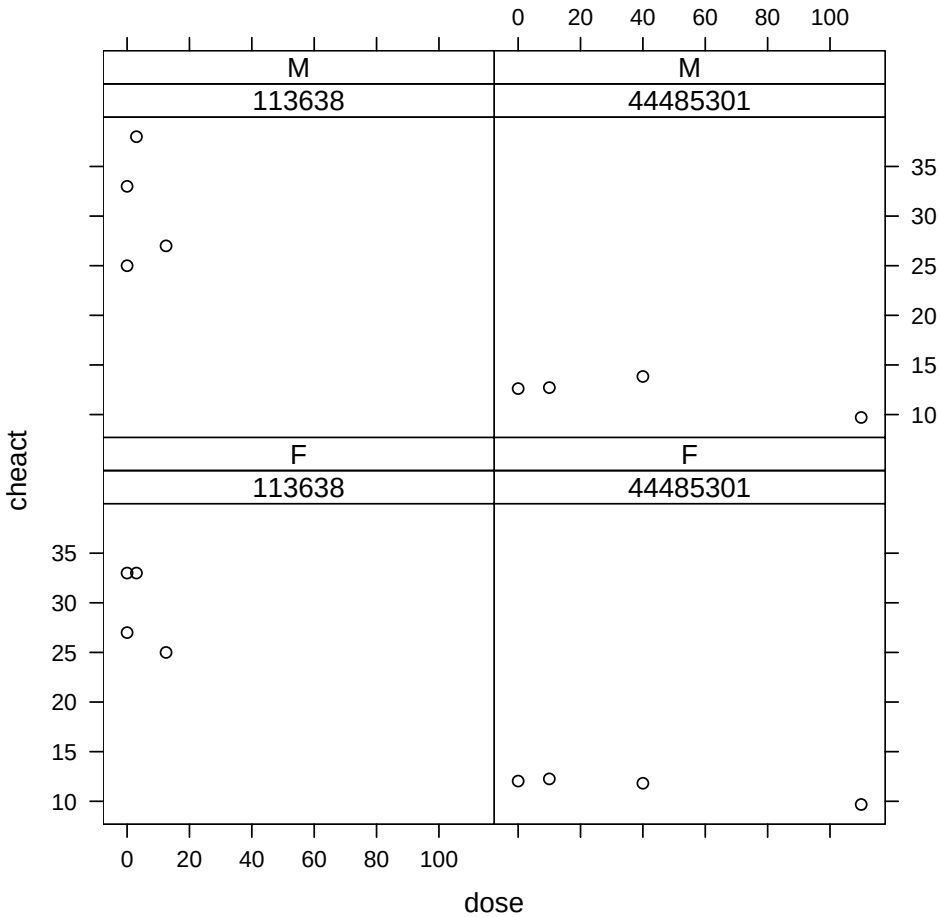
Do the estimates of `m` differ from each other?

```
> Parms <- getParms("^m", coef(drmod0))
> NP <- length(Parms)
> L <- diag(NP) - matrix(1, nrow = NP, ncol = NP)/NP
> L <- L[-NP, ]
> colnames(L) <- names(Parms)
> anova(drmod0, L = L)

Denom. DF: 72
 F-test for linear combination(s)
  m.mridXsex113638:28:M m.mridXsex113638:28:F m.mridXsex44485301:1:M
1                  0.75                 -0.25                  -0.25
2                 -0.25                  0.75                  -0.25
3                 -0.25                 -0.25                   0.75
  m.mridXsex44485301:1:F
1                  -0.25
2                  -0.25
3                  -0.25
  numDF  F-value p-value
1     3 1.083367  0.3617
```

No. Is their mean different from 0?

```
> L <- matrix(1, nrow = 1, ncol = NP)/4
> colnames(L) <- names(Parms)
> anova(drmod0, L = L)

Denom. DF: 72
 F-test for linear combination(s)
 m.mridXsex113638:28:M  m.mridXsex113638:28:F m.mridXsex44485301:1:M
                  0.25                   0.25                   0.25
m.mridXsex44485301:1:F
                  0.25
  numDF  F-value p-value
1     1 5.425815  0.0227
```

Yes. Here is a plot of the data:



The doses in 113638 are much lower than in 44485301, and the data in the latter study are much less variable, apparently. Thus, the studies are consistent with each other, and the second study has high enough doses that the effect is barely detectable. Go ahead and fit the main dose response function, setting `tz` to -10, `lg` to 0:

```
> Start0 <- c(getParms("^lA", coef(drmod0)), log(-log(0.9)/mean(getParms("^m",
+     coef(drmod0)))), 0)
> Start <- Start0
> formals(tcmfn4)$tz <- -10
> formals(tcmfn4)$lg <- 0
> formals(tcmfn4)$lTr <- log(1.5)
> formals(tcmfn4)$delta <- 0.5
> icnt <- 1
> Maxcnt <- 20
> repeat {
+     drmod1 <- try(gnls(cheact ~ tcmfn4(dose, tmpstds, lA = lA,
+         lD = lD), data = idta, params = list(lA ~ mridXsex -
+         1, lD ~ sex, ), weights = varIdent(form = ~1 | mridXsex),
+         start = Start), silent = TRUE)
+     if (!inherits(drmod1, "try-error") || icnt > Maxcnt) {
+         if (icnt <= Maxcnt)
+             writeLines(paste("Successful in ", icnt, if (icnt >
+                 1)
+                 "tries"
```

```
+             else "try"))
+         else writeLines("Maxcnt exceeded")
+         break
+     }
+     icnt <- icnt + 1
+     Start <- Start0 * (1 + rnorm(length(Start0), sd = 0.2))
+ }
```

```
Successful in  1 try
```

The difference between the sexes in `lD` is not significant:

```
> summary(drmod1)$tTable["lD.sexM", ]
```

```
    Value Std.Error   t-value   p-value
0.2532302 0.6454529 0.3923295 0.6959425
```

so refit without the sex effect:

```
> Par <- coef(drmod1)
> Start0 <- c(getParms("^lA", Par), Par["lD.(Intercept)"])
> Start <- Start0
> icnt <- 1
> Maxcnt <- 20
> repeat {
+     drmod2 <- try(gnls(cheact ~ tcmfn4(dose, tmpstds, lA = lA,
+         lD = lD), data = idta, params = list(lA ~ mridXsex -
+         1, lD ~ 1, ), weights = varIdent(form = ~1 | mridXsex),
+         start = Start), silent = TRUE)
+     if (!inherits(drmod2, "try-error") || icnt > Maxcnt) {
+         if (icnt <= Maxcnt)
+             writeLines(paste("Successful in ", icnt, if (icnt >
+                 1)
+                 "tries"
+             else "try"))
+         else writeLines("Maxcnt exceeded")
+         break
+     }
+     icnt <- icnt + 1
+     Start <- Start0 * (1 + rnorm(length(Start0), sd = 0.2))
+ }
```

```
Successful in  1 try
```

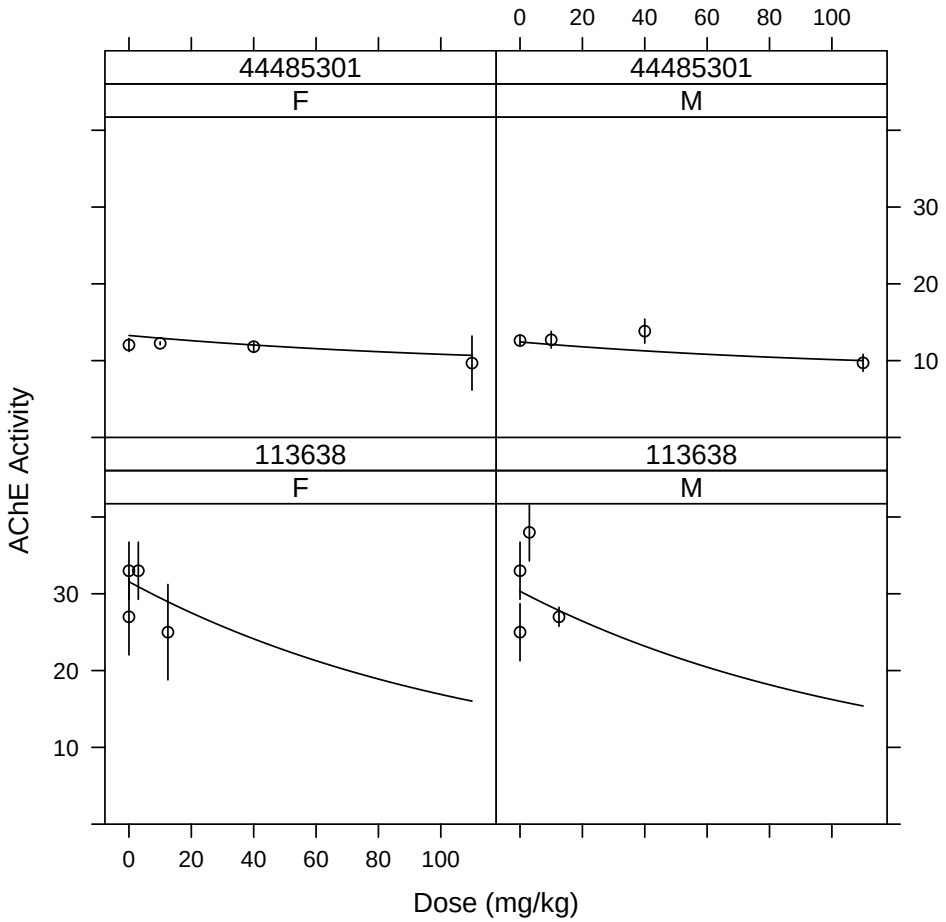
A summary of this fit:

```
> tTab <- summary(drmod2)$tTable
> Ints <- intervals(drmod2, type = "coef")$coef
> Ints90 <- intervals(drmod2, type = "coef", level = 0.9)$coef
> summary(drmod2)
```

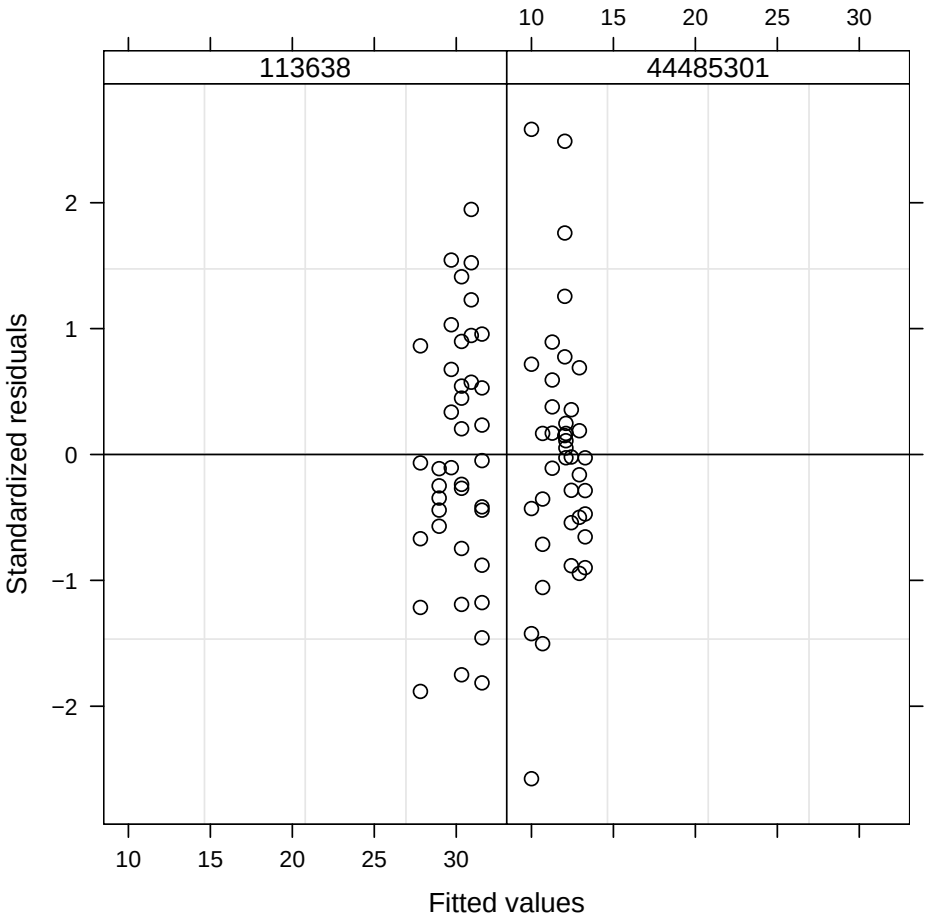
```
Generalized nonlinear least squares fit
  Model: cheact ~ tcmfn4(dose, tmpstds, lA = lA, lD = lD)
  Data: idta
       AIC      BIC    logLik
  399.5642 421.0024 -190.7821
```

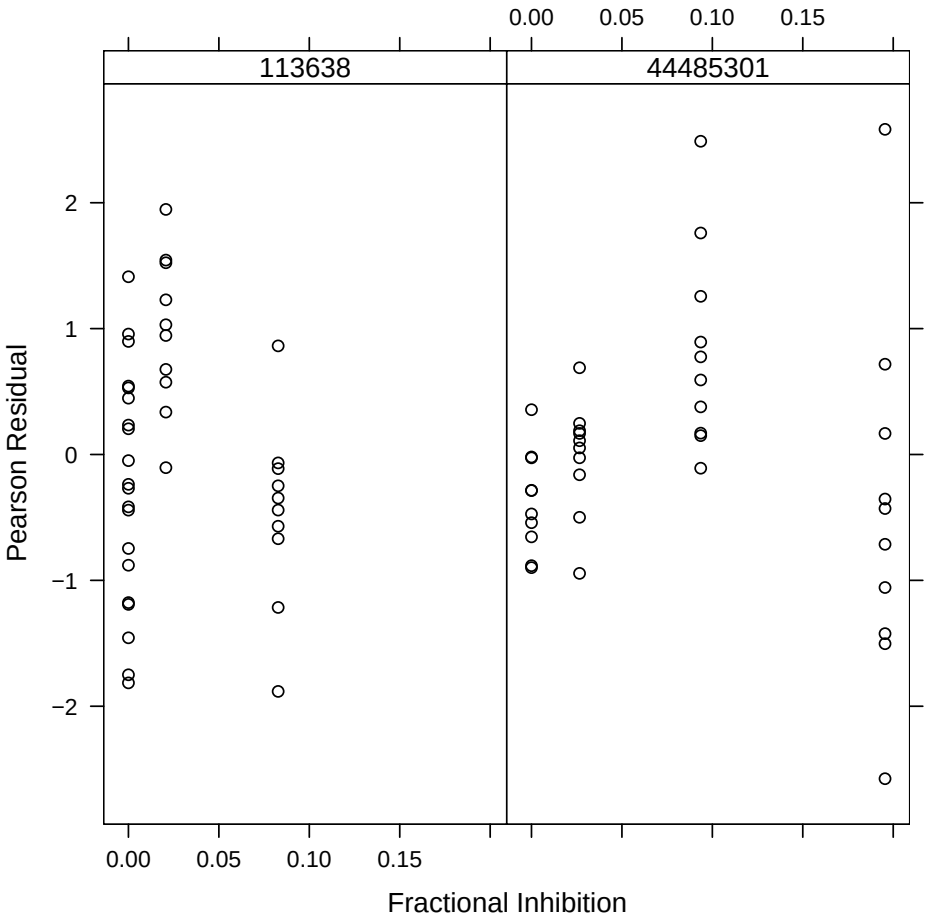
```
Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | mridXsex
 Parameter estimates:
 113638:28:M  113638:28:F 44485301:1:M 44485301:1:F
   1.0000000    0.8322848    0.2465107    0.2511046

Coefficients:
                           Value Std.Error  t-value p-value
lA.mridXsex113638:28:M  3.452225 0.0421111 81.97895       0
lA.mridXsex113638:28:F  3.411995 0.0366947 92.98324       0
lA.mridXsex44485301:1:M 2.586004 0.0311883 82.91596       0
lA.mridXsex44485301:1:F 2.520337 0.0330552 76.24639       0
lD                      2.481407 0.3230579  7.68100       0

 Correlation:
                        lA.X113638:28:M lA.X113638:28:F lA.X44485301:1:M
lA.mridXsex113638:28:F   0.039
lA.mridXsex44485301:1:M  0.106           0.122
lA.mridXsex44485301:1:F  0.100           0.115           0.309
lD                      -0.185          -0.213          -0.572
                        lA.X44485301:1:F
lA.mridXsex113638:28:F
lA.mridXsex44485301:1:M
lA.mridXsex44485301:1:F
lD                      -0.540

Standardized residuals:
        Min          Q1         Med          Q3         Max
-2.57563031 -0.54874769 -0.03745638  0.57872360  2.58329180

Residual standard error: 5.694843
Degrees of freedom: 80 total; 75 residual
```

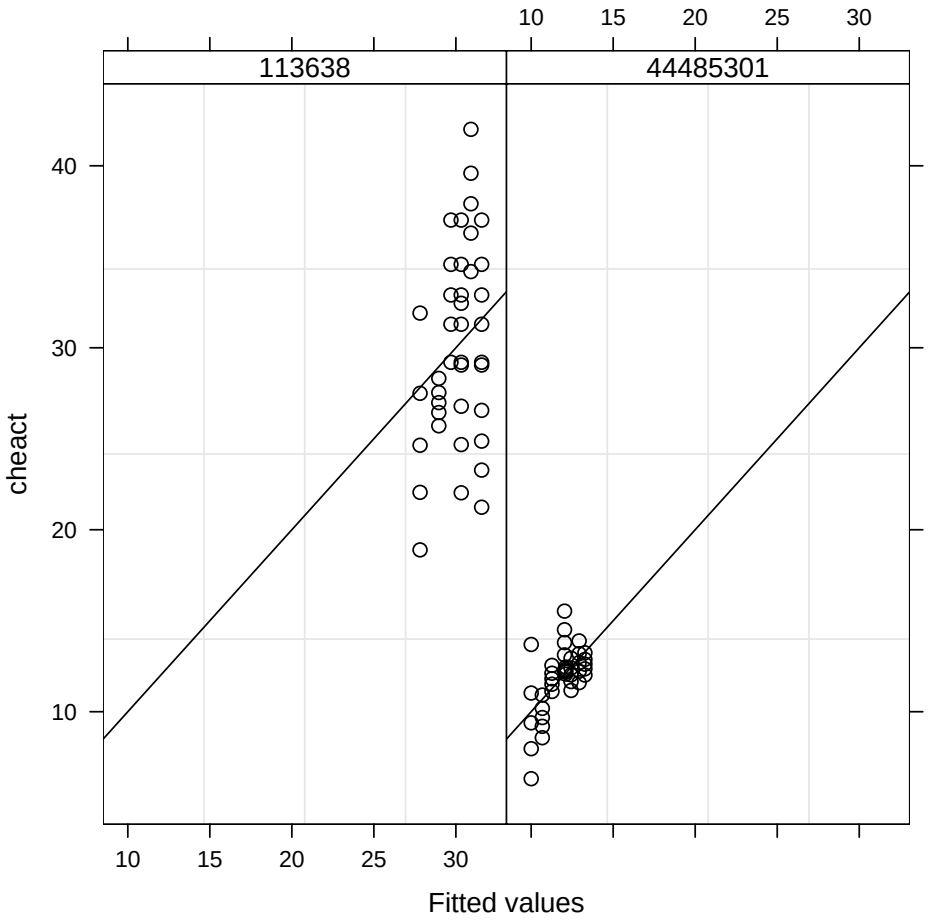
Diagnostic plots for this model:

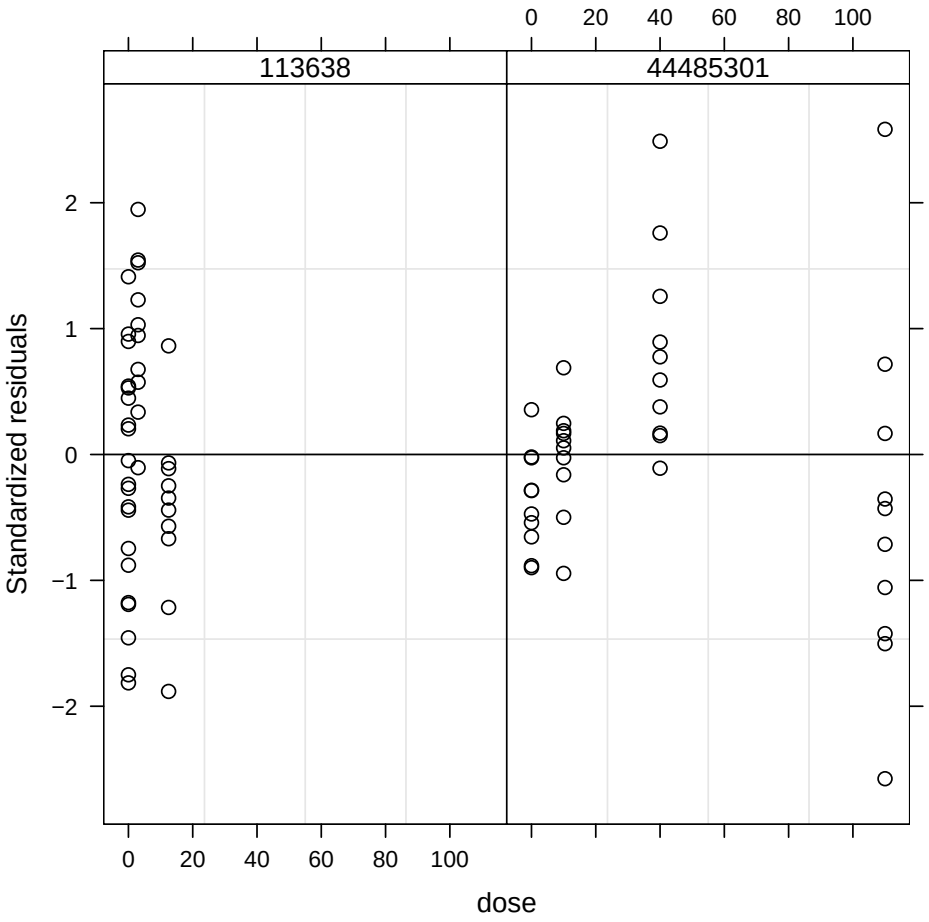


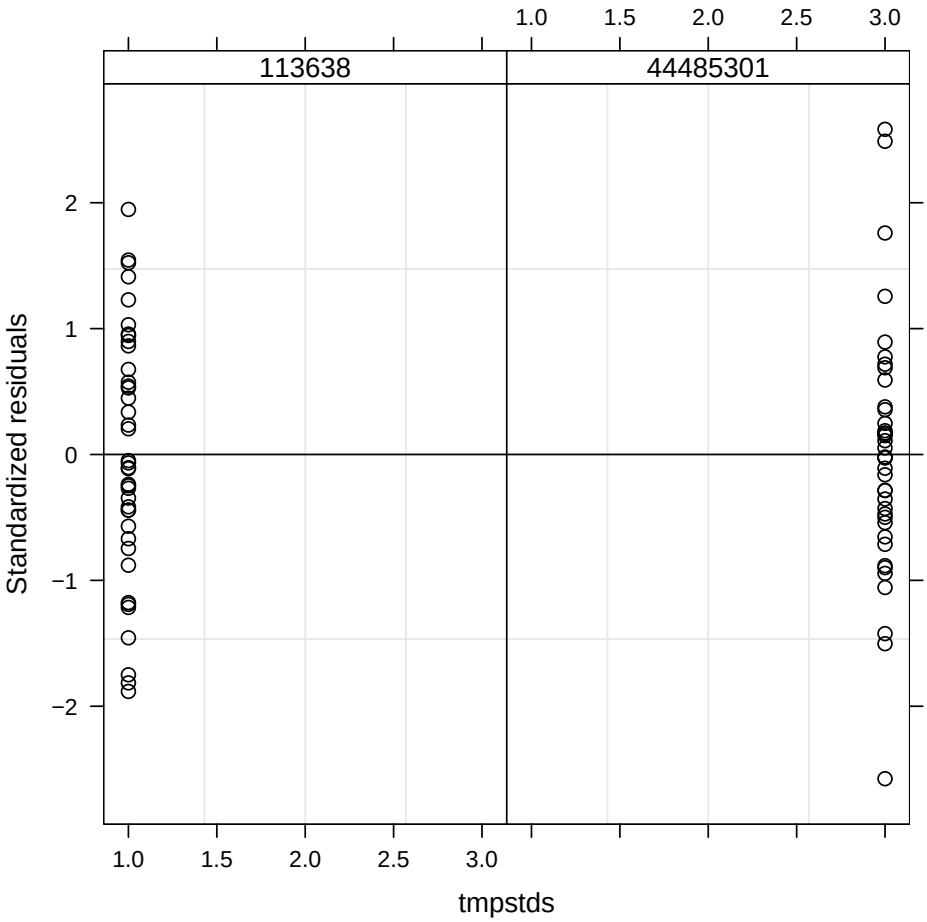
Data and modelfit


113638
44485301
Standardized residuals
Fitted values
10
15
20
25
30
2
1
0
−1
−2


113638
44485301
Pearson Residual
Fractional Inhibition
0.00
0.05
0.10
0.15
2
1
0
−1
−2

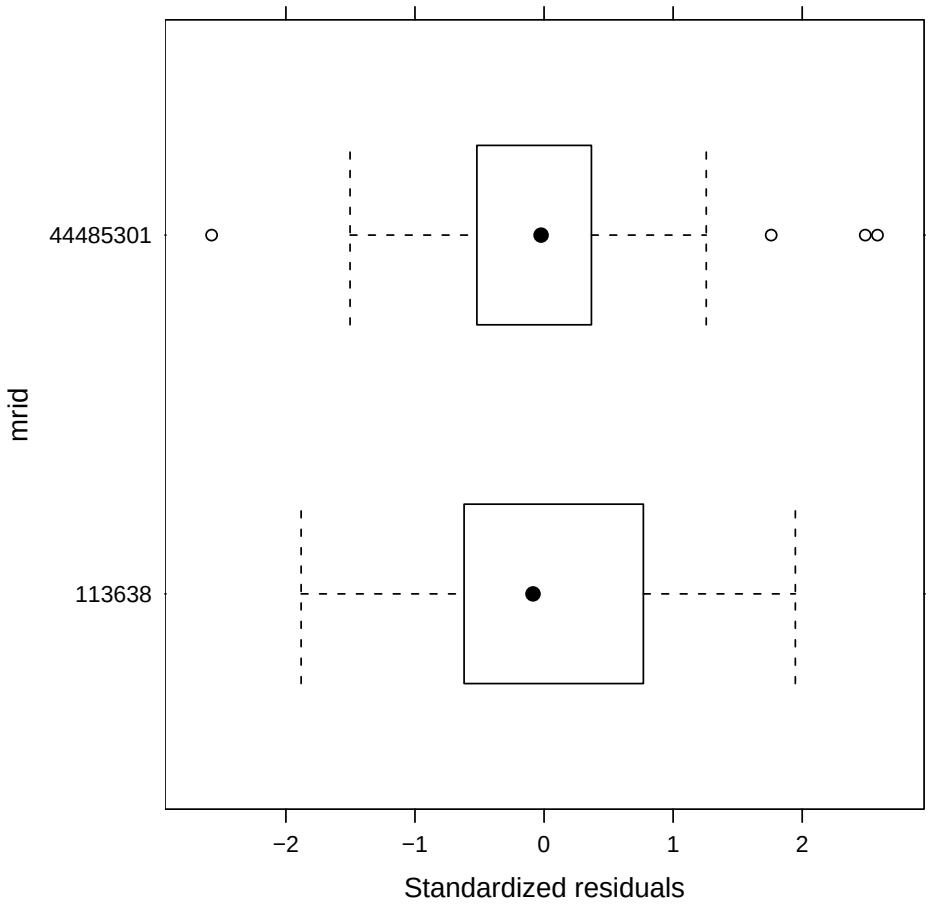

113638
44485301
cheact
Fitted values
10
15
20
25
30
40


113638
44485301
Standardized residuals
dose
0
20
40
60
80
100
2
1
0
−1
−2



Some more graphs:

Pearson residuals split by mrid and sex, and QQ plots for the Pearson residuals and random effects:



There is no point in plotting the distribution of the error residuals here, since the original data were all aggregated. The "individual" data used in the modeling functions were constructed (using a lognormal distribution) to give the observed mean and standard deviation.

3 Summary

The critical estimates from this analysis are listed below. They are printed with greater than usual precision, in case they are to be used in further computation. For reporting, round to two or three significant digits. In these data, the BMD differs between sexes, and only a single value for recovery half life, pooled over doses and sexes, was estimated. BMD has units mg/kg, and times are in hours.

species RAT

mrid `[1] "113638"  "44485301"`

`lD` **(se)** 2.48140667309007 (0.323057893541743)

BMD (95% CI) 11.9580736929729 (6.28296716561366, 22.7592350361429)

BMDL, the one-sided lower 95% CL 6.98229640231989

Save the results:

```
> save.image(file = "RatBrainDR.RData")
```

Save the results for incorporating into a database:

```
> primicarb.oral.brain <- list(mrid = levels(dta$mrid), species = "RAT",
+     BMDs = list(combined = list(lD = tTab["lD", "Value"], lD.se = tTab["lD",
+         "Std.Error"], BMD = exp(Ints["lD", "est."]), BMD.CI = exp(Ints["lD",
+         c("lower", "upper")]), BMDL = exp(Ints90["lD", "lower"]))))
> save(primicarb.oral.brain, file = file.path("..", "..", "01Summaries",
+     "primicarb.oral.brain.RData"))
```