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

June 22, 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.

All the data used for this DR model are in `AggData`.

These are all the names used for the brain sections we want from `AggData`:

```
> BrainSections <- levels(AggData$sctn)[grep("(^BRAIN)|(^WHOLEBRAIN)|(LEFT HEMISPHERE)| (HALFBRAIN)",
+     levels(AggData$sctn))]
```

Now, set up the analysis dataset.

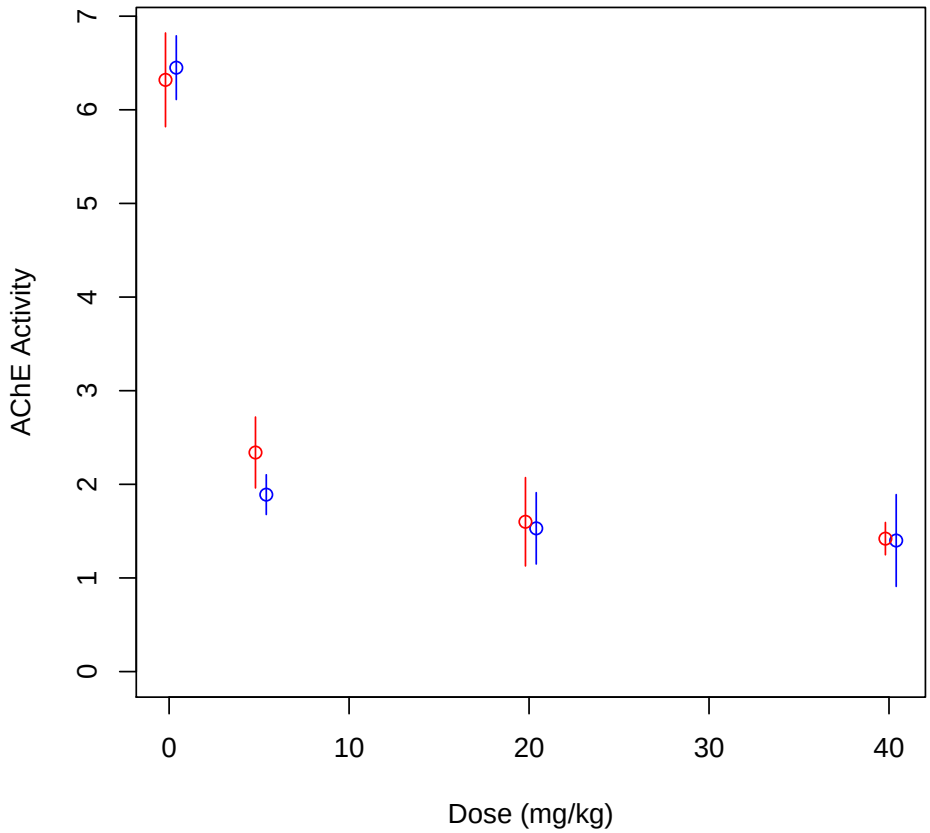
```
> dta <- CleanUp(subset(AggData, chemical %in% "THIODICARB" & species %in%
+     "RAT" & dsmtd %in% "GAVAGE" & sctn %in% BrainSections & !is.na(cheact) &
+     (n == 1 | !is.na(sd)) & (is.na(ta) | ta %in% c("AI")), select = c("cheact",
+     "sd", "n", "dose", "tmonstdy", "tmpstds", "sex", "mrid",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

Summaries of the relevant variables in this dataset:

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

dta$mrid: 45138703
     cheact            sd               n              dose          tmonstdy
 Min.   :1.400   Min.   :0.1390   Min.   : 5.00   Min.   : 0.00   Min.   :1
 1st Qu.:1.817   1st Qu.:0.2567   1st Qu.: 5.00   1st Qu.: 3.75   1st Qu.:1
 Median :6.250   Median :0.3280   Median : 5.00   Median :12.50   Median :1
```

```
 Mean   :4.571   Mean   :0.3932   Mean   : 6.25   Mean   :16.25   Mean   :1
 3rd Qu.:6.433   3rd Qu.:0.3970   3rd Qu.: 6.25   3rd Qu.:25.00   3rd Qu.:1
 Max.   :6.480   Max.   :1.5980   Max.   :10.00   Max.   :40.00   Max.   :1
    tmpstds     sex         mrid    species
 Min.   : 1.0   F:8   45138703:16   RAT:16
 1st Qu.: 1.0   M:8
 Median :12.5
 Mean   :12.5
 3rd Qu.:24.0
 Max.   :24.0

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

         mrid tmonstdy tmpstds
1817 45138703        1       1
1821 45138703        1      24

> with(dta, table(mrid, dose, sex))

, , sex = F

          dose
mrid       0 5 20 40
  45138703 2 2 2  2

, , sex = M

          dose
mrid       0 5 20 40
  45138703 2 2 2  2

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

dta$mrid: 45138703
    tmpstds
dose 1 24
  0  2 2
  5  2 2
  20 2 2
  40 2 2
```

Data are available from a single study, MRID 45138703, and then only for the one hour time point. Clearly, only dose-response modeling for this dataset. Here is a figure:



By 24 hours (the next time point available), the lower two non-control doses are back to control values, and the high dose is getting close. This time point provides no useful information for estimating either `lD` or `lTr`.

```
> dta <- CleanUp(subset(dta, tmpstds < 24))
```

2 Dose-Response Modeling

2.1 strategy

Since there are no low-dose points, assume `lg` = 0, set `lTr` to log(1.5), delta to 0.5 hours, and just estimate sex-specific background, `lD`, and `tz`.

Fitting the model will follow these steps:

1. First, use `GetInitialValues()` to get starting values for the model against these data,
2. Next, fit `tcmfn4()` using the parameterizations determined in the previous step, and a constant variance model, allowing `lD` to vary among mrids.

   Try fitting with other variance models the same model for fixed and random effects, using AIC to identify the best variance model.

   Since there are no random effects, just fit using `gnls()`

2.2 Initial Values

Save the initial values so that we do not need to go through all this to re-run the analysis. Also, set the argument `delta` to 0.5, the earliest non-zero time point.

```
> formals(tcmfn4)$delta <- 0.5
> formals(tcmfn4)$lg <- 0
> formals(tcmfn4)$lTr <- log(1.5)
> initfile <- paste("initvals-brain-DR-1.RData", sep = "")
> if (!file.exists(initfile)) {
+     lA.start <- lm(I(log(cheact)) ~ sex - 1, data = CleanUp(subset(dta,
+         dose %in% 0)))
+     Start <- c(coef(lA.start), log(5), 0, -2, 0)
+     init1 <- GetInitialValues(cbind(resp = cheact, sd = sd, n = n) ~
+         tcmfn4(dose, tmpstds, lA = lA, tz = tz, lD = lD), data = dta,
+         params = list(lA ~ sex - 1, lD ~ sex, tz ~ sex), start = Start,
+         weights = varPower(value = 1))
+     save(init1, file = initfile)
+ } else load(initfile)
> tmp <- t(init1$Redundancy[[1]]$Eigens)
> tmp <- tmp[-grep("^lA", rownames(tmp)), ]
> round(tmp[, 1:5], digits = 2)
```

```
               [,1] [,2] [,3] [,4] [,5]
CondIndex      3.47 2.57 1.87 1.65 1.48
mu             0.46 0.62 0.85 0.97 1.08
lD.(Intercept) 0.20 0.54 0.01 0.15 0.07
lD.sexM        0.24 0.51 0.00 0.20 0.00
tz.(Intercept) 0.73 0.08 0.07 0.07 0.00
tz.sexM        0.80 0.03 0.06 0.03 0.03
```

The above shows the results of a redundancy analysis. It should be easy to estimate the remaining parameters in the model.

```
> Start <- init1$start$beta
> idta <- init1$data
> drmod1 <- gnls(cheact ~ tcmfn4(dose, tmpstds, lA = lA, tz = tz,
+     lD = lD), data = idta, params = list(lA ~ sex - 1, lD ~ sex,
+     tz ~ sex, ), start = Start)
```

Try adding `varPower()` to the variance model:

```
> drmod2 <- update(drmod1, weights = varPower())
```

Compare the two:

```
> anova(drmod1, drmod2)
```

```
       Model df      AIC      BIC    logLik   Test    L.Ratio p-value
drmod1     1  7 26.66324 38.48540 -6.331620
drmod2     2  8 28.58960 42.10064 -6.294803 1 vs 2 0.07363527  0.7861
```

There is no improvement of the fit. Can we simplify the first model by eliminating the separate estimates of `lD` and `tz` by sex? Neither is significant by itself:

```
> summary(drmod1)$tTable
```

```
                    Value  Std.Error     t-value      p-value
lA.sexF         1.8436812 0.02175763  84.7372090 3.799442e-41
lA.sexM         1.8640785 0.02131837  87.4400048 1.312667e-41
lD.(Intercept) -1.1578951 0.10313572 -11.2269063 5.556049e-13
lD.sexM        -0.3556851 0.17995556  -1.9765165 5.625363e-02
tz.(Intercept) -3.0545078 0.53272367  -5.7337564 1.909972e-06
tz.sexM        -0.4523104 0.94853093  -0.4768536 6.365171e-01
```

Now, test whether there is a gender difference in `lD` and `tz`:

```
> L <- diag(2)
> dimnames(L) = list(NULL, c("lD.sexM", "tz.sexM"))
> anova(drmod1, L = L)
```

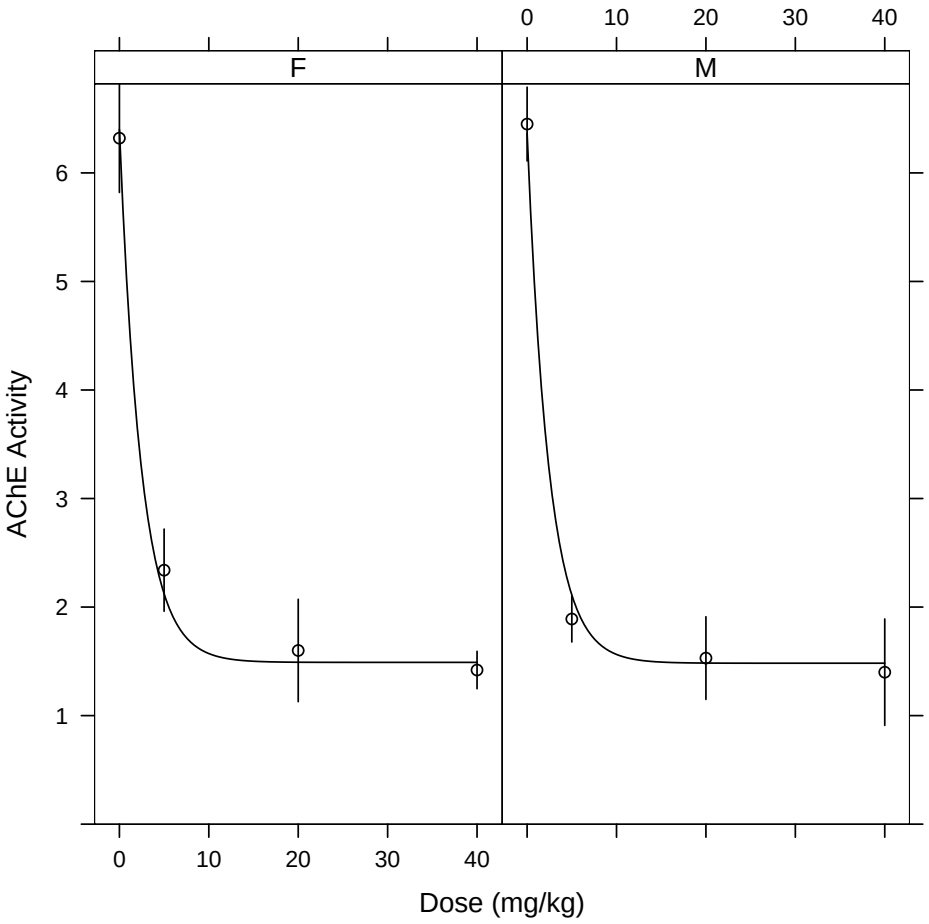
```
Denom. DF: 34
 F-test for linear combination(s)
  lD.sexM tz.sexM
1       1       0
2       0       1
  numDF  F-value p-value
1     2 2.697395  0.0818
```

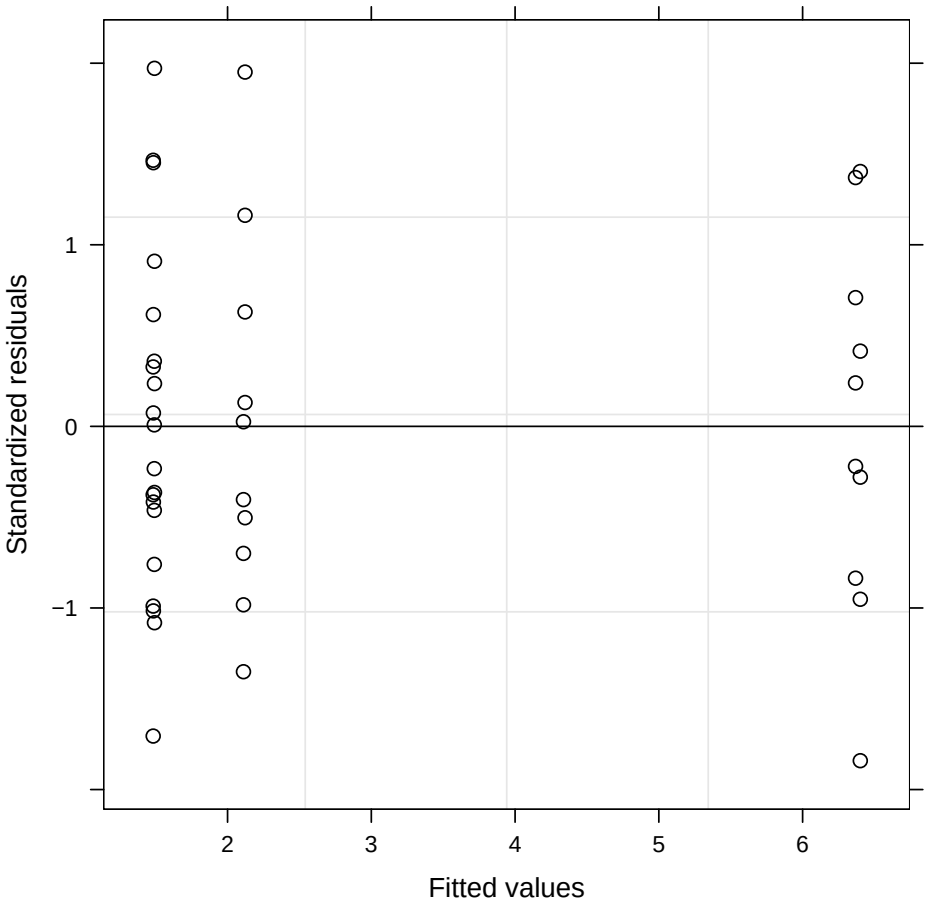
These factors do not differ from 0 jointly, so build a new model with a single estimate for `lD` and `tz`:

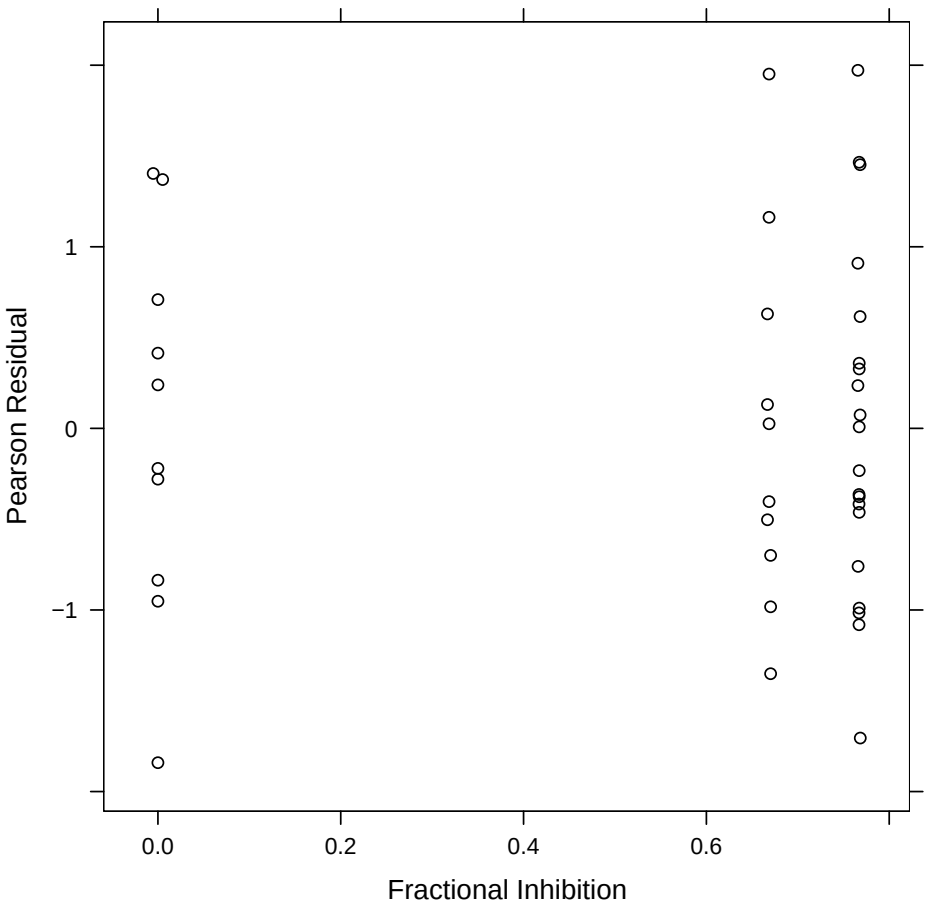
```
> Start <- coef(drmod1)
> Start <- Start[-grep("lD.sexM", names(Start))]
> Start <- Start[-grep("tz.sexM", names(Start))]
> drmod3 <- update(drmod1, params = list(lA ~ sex - 1, lD ~ 1,
+     tz ~ 1), start = Start)
```

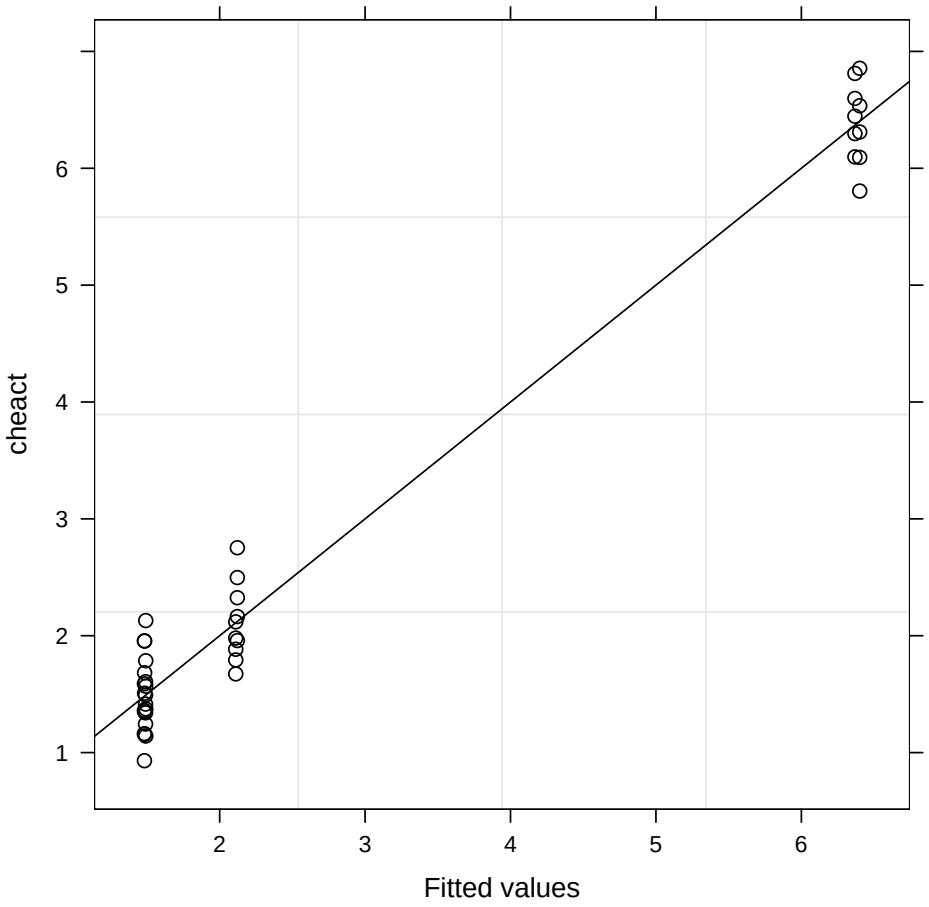
Diagnostic graphics for the model contained in `drmod3`:

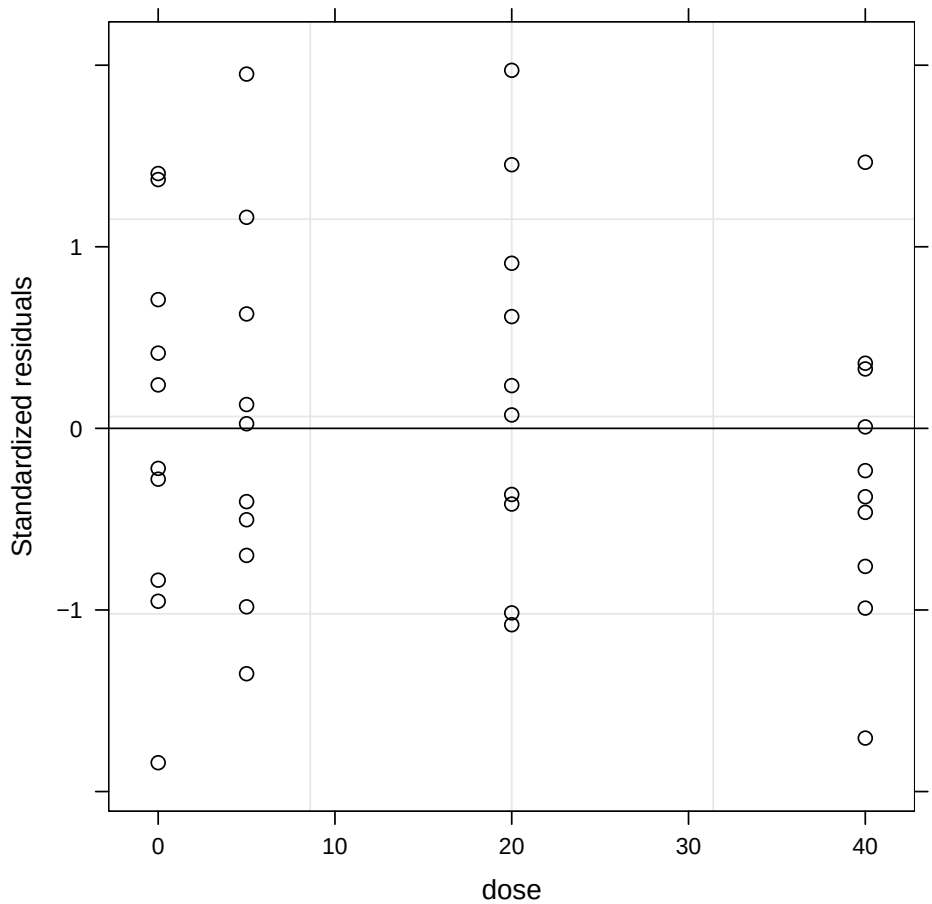
Data and dose-response:


F
M
AChE Activity
Dose (mg/kg)
0
10
20
30
40
1
2
3
4
5
6


Standardized residuals
1
0
−1
2
3
4
5
6
Fitted values


Pearson Residual
1
0
−1
0.0
0.2
0.4
0.6
Fractional Inhibition


cheact
Fitted values
1
2
3
4
5
6
2
3
4
5
6



Details of this final model fit:

```
> tTab <- summary(drmod3)$tTable
> Ints <- intervals(drmod3, which = "coef")$coef
> Ints90 <- intervals(drmod3, which = "coef", level = 0.9)$coef
> summary(drmod3)
```

```
Generalized nonlinear least squares fit
  Model: cheact ~ tcmfn4(dose, tmpstds, lA = lA, tz = tz, lD = lD)
  Data: idta
       AIC      BIC    logLik
  29.05891 37.50331 -9.529457

Coefficients:
            Value Std.Error   t-value p-value
lA.sexF  1.856485 0.0215815  86.02201       0
lA.sexM  1.851324 0.0216822  85.38472       0
lD      -1.323580 0.0875085 -15.12516       0
tz      -3.255025 0.4686612  -6.94537       0

 Correlation:
        lA.sxF lA.sxM lD
lA.sexM  0.098
```

```
lD      -0.130 -0.130
tz      -0.231 -0.230 -0.313

Standardized residuals:
       Min         Q1        Med         Q3        Max
-1.8409569 -0.7144962 -0.1059735  0.6193924  1.9719059

Residual standard error: 0.3236729
Degrees of freedom: 40 total; 36 residual
```

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.

species RAT

mrid `[1] "45138703"`

`lD` **(standard error)** -1.32357980159691 (0.0875085039627899)

BMD (95% CI) 0.266180720211340 (0.222894820994649, 0.317872687647276)

BMDL 0.229622135823441

Save everything:

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

Save the results for incorporating into a database:

```
> thiodicarb.oral.brain <- list(mrid = levels(dta$mrid), species = levels(dta$species),
+     BMDs = list(combined = list(lD = tTab["lD", 1], lD.se = tTab["lD",
+         2], BMD = exp(Ints["lD", 2]), BMD.CI = exp(Ints["lD",
+         c(1, 3)]), BMDL = exp(Ints90["lD", 1]))))
> save(thiodicarb.oral.brain, file = file.path("..", "..", "01Summaries",
+     "thiodicarb.oral.brain.RData"))
```