

# An introduction to the mcprofile package

Daniel Gerhard

September 9, 2013

## Contents

<b>1</b>	<b>Motivation</b>	<b>1</b>
<b>2</b>	<b>Methods</b>	<b>1</b>
2.1	Generalized linear models . . . . .	1
2.2	General linear hypotheses . . . . .	2
2.3	Profile statistics . . . . .	2
2.3.1	Wald statistic . . . . .	2
2.3.2	Signed likelihood root . . . . .	2
2.3.3	Modified likelihood root . . . . .	3
2.4	Confidence intervals . . . . .	3
2.5	Multiple hypotheses testing . . . . .	3
<b>3</b>	<b>The mcprofile() function</b>	<b>3</b>
3.1	plot methods . . . . .	6
3.2	Modifying the profile support grid . . . . .	8
3.3	Simultaneous confidence intervals and multiple testing . . . . .	9
<b>4</b>	<b>Wald-type statistics</b>	<b>10</b>
<b>5</b>	<b>Modified likelihood root</b>	<b>11</b>
<b>6</b>	<b>Simultaneous confidence intervals for predictions in logistic regression models</b>	<b>12</b>

## 1 Motivation

The package mcprofile contains functions for simultaneous inference based on a set of profile statistics, controlling the family-wise error rate (FWER) at a specified level. Focus is set to testing general linear hypotheses in generalized linear models and provide a set of compatible simultaneous confidence intervals. The multiple testing framework of Hothorn et al. 2008 is adopted to signed likelihood root and modified likelihood root statistics to improve the performance especially at small sample sizes.

## 2 Methods

### 2.1 Generalized linear models

A vector of  $i = 1, \dots, n$  observations  $y_i$  is assumed to be a realization of a random variable  $Y_i$ , where each component of  $Y_i$  is assumed to have a distribution in the exponential family. The systematic component of the assumed generalized linear model is

$$g(\mu_i) = \eta_i = \sum_{j=1}^p x_{ij} \beta_j$$

with a link function  $g(\cdot)$ , a  $p$ -dimensional vector of parameters  $\beta_j$ , and a  $(n \times p)$  matrix with design covariates  $x_{ij}$ . Thus, the log-likelihood can be written as  $l(\mu_i; y_i) = \sum_{i=1}^n \log f_i(y_i; \mu_i)$ . Instead of the likelihood function,

the scaled deviance

$$D(y_i; \mu_i) = 2l(y_i; y_i) - 2l(\mu_i; y_i)$$

can be used as a goodness-of-fit criterion. To estimate a coefficient vector  $\hat{\beta}_j$  an iteratively re-weighted least squares algorithm can be applied for finding the minimum of the deviance function.

## 2.2 General linear hypotheses

In many applications the experimental questions are specified through  $k = 1, \dots, q$  linear combinations of the model parameters,  $\vartheta_k = \sum_{j=1}^p a_{kj}\beta_j$ , which are defined by a  $(q \times p)$  coefficient matrix  $a_{kj}$ . When constructing simultaneous confidence intervals or hypotheses tests, the multiple comparison problem of testing all hypotheses at a nominal level of  $\alpha$ , the overall type I error rate, has to be considered. A unifying simultaneous inference framework for these parameter linear combinations in general parametric models is presented in Hothorn et al. 2008. They consider the general linear hypotheses:

$$H_0 : \sum_{j=1}^p a_{kj}\beta_j = m_k$$

where  $m_k$  is a vector, of order  $q$ , of specified constants.

The key factor of this single-step inference is the assumption of a multivariate normal-distribution of the standardized estimator  $\hat{\vartheta}_k$  with a correlation structure, which is directly obtained from the  $(p \times p)$  observed information matrix at the parameter estimates  $j(\beta) = -\frac{\partial^2 l(\mu_i; y_i)}{\partial \beta \partial \beta^T}$ .

## 2.3 Profile statistics

Several test statistics can be used to test the set of general linear hypotheses.

### 2.3.1 Wald statistic

$$w(\vartheta_k) = \left( \hat{\vartheta}_k - \vartheta_k \right) j^{\frac{1}{2}}(\hat{\vartheta}_k),$$

where  $j^{-1}(\hat{\vartheta}) = A j^{-1}(\hat{\beta}) A^T$  is the observed information matrix of the contrast parameters, with  $A$  being the  $(q \times p)$  coefficient matrix with elements  $a_{kj}$ . As  $\hat{\vartheta}_k$  and the observed information are fixed at the maximum likelihood estimates,  $w(\vartheta_k)$  is a linear function with increasing  $\vartheta_k$ , and therefore approximates the deviance by a quadratic function  $w(\vartheta_k)^2$ , which holds exactly for a Gaussian linear model.

### 2.3.2 Signed likelihood root

$$q(\vartheta_k) = \text{sign} \left( \hat{\vartheta}_k - \vartheta_k \right) \sqrt{\frac{D(y_i; \hat{\mu}_i) - D(y_i; \tilde{\mu}_i)}{\phi}}$$

$\hat{\mu}_i$  is the linear predictor at the maximum likelihood estimates  $\hat{\beta}_j$ .  $\tilde{\mu}_i$  denotes the linear predictor obtained for restricted parameter estimates  $\tilde{\beta}_j$ , obtained under the  $k$ th linear constraint  $\sum_{j=1}^p a_{kj}\beta_j = \vartheta_k$ .  $\phi$  is a dispersion parameter, which accounts for extra variation in the data. This parameter is fixed at 1 e.g. for a Binomial or Poisson model, but can also be estimated from the data, like the residual error in a Gaussian linear model.

To obtain  $\tilde{\mu}_i$ , the weighted least squares step in the IRWLS algorithm can be modified by using a weighted regression of  $z$  on the design covariates  $x_1, \dots, x_p$  with weight  $W$ , which allows to apply linear equality constraints on the regression parameters at each iteration. A quadratic programming algorithm can be used to obtain the restricted parameter estimates.

### 2.3.3 Modified likelihood root

$$r(\vartheta_k) = q(\vartheta_k) + \frac{1}{q(\vartheta_k)} \log \left( \frac{w(\vartheta_k) \rho(\vartheta_k, \hat{\vartheta}_k)}{q(\vartheta_k)} \right),$$

with  $\rho(\vartheta_k, \hat{\vartheta}_k) = \sqrt{\frac{|j_{\lambda\lambda}(\hat{\beta}_j)|}{|j_{\lambda\lambda}(\tilde{\beta}_j)|}}$ .  $|j_{\lambda\lambda}(\cdot)|$  denotes the determinant of a subset of the observed information matrix to summarize the information about the nuisance parameters that are not subject to the linear constraint. The observed information is evaluated at the maximum likelihood estimates  $\hat{\beta}_j$  and the restricted estimates  $\tilde{\beta}_j$  under the linear constraint  $\sum_{j=1}^p a_{kj} \beta_j = \vartheta_k$ .

## 2.4 Confidence intervals

A Wald-type confidence interval is defined as

$$I = \{\vartheta_k : -c_{1-\alpha} \leq w(\vartheta_k) \leq c_{1-\alpha}\}$$

with a critical value  $c_{1-\alpha}$ . Analogously,  $w(\vartheta_k)$  can be substituted by the different profile statistics  $q(\vartheta_k)$  and  $r(\vartheta_k)$ . As the confidence limits are found separately for each of the  $q$  linear combinations of parameters, the combined set of confidence limits describes a rectangular confidence set in a  $q$ -dimensional space.

The critical value should be chosen in a way that the FWER is controlled, considering the correlation between the derived parameters. The statistics are assumed to follow a  $q$ -variate normal or  $t$ -distribution. The correlation structure for this distribution is obtained by standardizing the variance-covariance matrix of the contrast parameters,

$$\Phi(\hat{\vartheta}) = \Delta(\hat{\vartheta}_k) j^{-1}(\hat{\vartheta}) \Delta(\hat{\vartheta}_k), \quad \Delta(\hat{\vartheta}_k) = j^{\frac{1}{2}}(\hat{\vartheta}_k) I_q,$$

using the estimates from the data and treat them as if it were the true correlation matrix. In order to assign the same weight or error level to each of the  $q$  hypotheses an equicoordinate quantile  $c_{1-\alpha}$  is calculated from this  $q$ -dimensional distribution.

For an effective search for the confidence limits, a grid of values for each of the  $\theta_k$  around the maximum likelihood estimates is established, interpolating the resulting  $q(\vartheta_k)$  or  $r(\vartheta_k)$  by a cubic spline function. The confidence limits are found by evaluating the inverse of this interpolating function at  $-c_{1-\alpha}$  and  $c_{1-\alpha}$ .

## 2.5 Multiple hypotheses testing

Instead of simultaneous confidence intervals, the analogue testing procedure of the set of null hypotheses  $H_0 : \vartheta_k = m_k$  is available. To maintain the FWER, the global null hypothesis is rejected, if at least one elementary hypothesis is rejected; thus, focus is set on the maximum of test statistics. The distribution of this maximum evaluated at the specific test margins  $m_k$  can be specified for a two-sided testing procedure as

$$P(\max |w(m_k)| \leq t) = g(\Phi(\hat{\vartheta}), t)$$

for any  $t \in \mathbb{R}^q$ . Adjusted p-values controlling the FWER are calculated as

$$p_k = 1 - g(\Phi(\hat{\vartheta}), |t_k|)$$

where  $t_k$  is the observed statistic, either using  $w(\cdot)$ ,  $q(\cdot)$ , or  $r(\cdot)$ .

## 3 The mcprofile() function

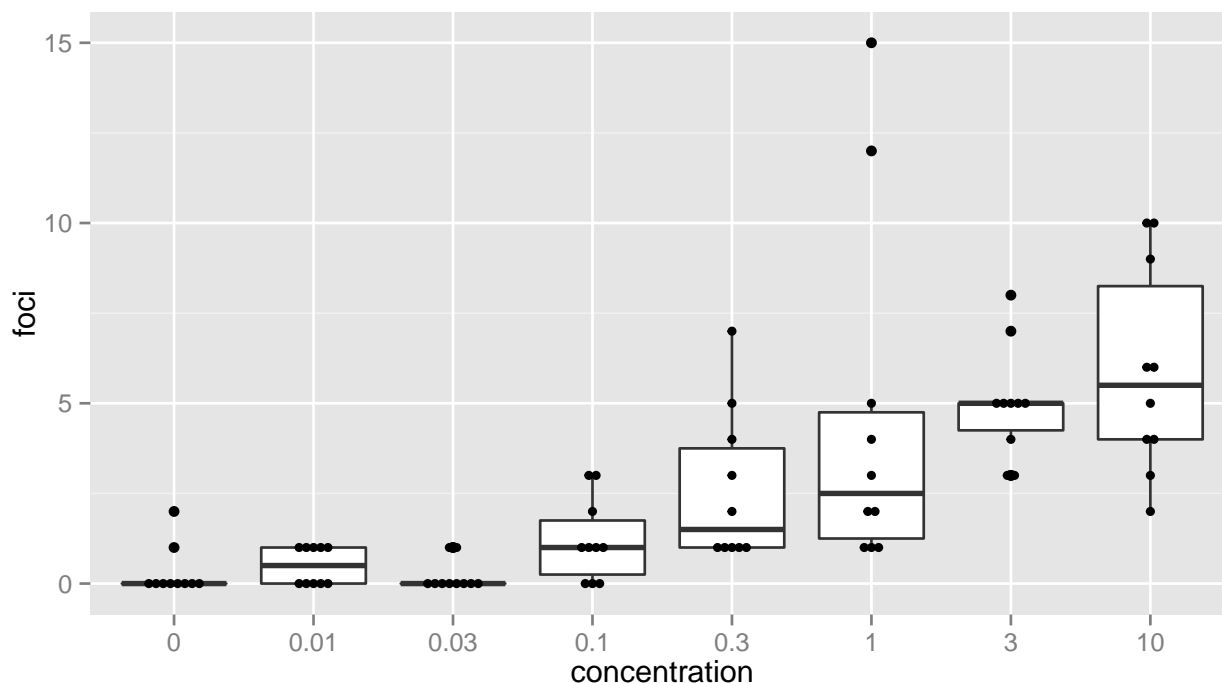
The mcprofile function needs at least two components, an object of class glm and a contrast matrix to define several linear combinations of glm coefficients.

As an application example the cta dataset is used, where Balb/c 3T3 cells are treated with different concentrations of a carcinogen to test for toxicity. Cells treated with a carcinogen do not stop proliferation, hence the number of foci (cell accumulations) are used as a measure of toxicity. The number of foci is counted for 10 independent replicates per concentration level.

```
data(cta)
str(cta)

## 'data.frame': 80 obs. of 2 variables:
## $ conc: num 0 0 0 0 0 0 0 0 0 0 ...
## $ foci: int 0 1 0 0 0 0 0 0 0 2 ...

cta$concf <- factor(cta$conc, levels=unique(cta$conc))
ggplot(cta, aes(y=foci, x=concf)) +
  geom_boxplot() +
  geom_dotplot(binaxis = "y", stackdir = "center", binwidth = 0.2) +
  xlab("concentration")
```



In a first analysis, the distance between dose levels is not taken into account, comparing the expected values for each dose levels with the one of the control. A generalized linear model is fitted to the data, assuming that the number of foci is following a Poisson distribution.

$$y_i \sim \text{Poisson}(\mu_i), \quad \log(\mu_i) = \eta_i, \quad \eta_i = \beta_0 + \sum_{j=1}^7 x_{ij}\beta_j$$

The dummy covariates  $x_{ij}$  are chosen in a way, that the  $\beta_j$  represent differences of each dose levels to the intercept  $\beta_0$ .

```
(m1 <- glm(foci ~ conf, data = cta, family = poisson(link = "log"))
##
## Call: glm(formula = foci ~ conf, family = poisson(link = "log"), data = cta)
##
## Coefficients:
## (Intercept)      conf0.01      conf0.03      conf0.1      conf0.3
##      -1.204         0.511        -0.405         1.386         2.159
##      conf1      conf3      conf10
##       2.730       2.813       2.979
##
## Degrees of Freedom: 79 Total (i.e. Null); 72 Residual
## Null Deviance:      272
## Residual Deviance: 106 AIC: 283
```

```
# confidence intervals for beta_j
confint(m1, parm = 2:8)

##           2.5 % 97.5 %
## concf0.01 -0.8942  2.095
## concf0.03 -2.4326  1.392
## concf0.1  0.2396  2.865
## concf0.3  1.1159  3.593
## concf1    1.7258  4.146
## concf3    1.8133  4.227
## concf10   1.9860  4.389
```

The same confidence intervals can be reproduced with `mcprofile`, but using linear contrasts instead of the `parms` argument. With the contrast matrix a linear combination of model coefficients is defined in each row; hence in each row the coefficient of 1 refers to the position of the  $\beta_j$  in the coefficient vector of the glm object. Names are passed down from the rownames of the contrast matrix.

Based on the profiles of the likelihood root, confidence intervals are calculated with the argument `adjust="none"` to control the per comparison error rate with a separate type-I-error for each comparison.

```
cm1 <- cbind(0, diag(7))
rownames(cm1) <- paste(levels(cta$concf)[-1], levels(cta$concf)[1], sep = " - ")
print(cm1)

##           [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## 0.01 - 0      0      1      0      0      0      0      0
## 0.03 - 0      0      0      1      0      0      0      0
## 0.1 - 0       0      0      0      1      0      0      0
## 0.3 - 0       0      0      0      0      1      0      0
## 1 - 0         0      0      0      0      0      1      0
## 3 - 0         0      0      0      0      0      0      1
## 10 - 0        0      0      0      0      0      0      1

p1 <- mcprofile(m1, cm1)
confint(p1, adjust = "none")

##
##      mcprofile - Confidence Intervals
##
## level:    0.95
## adjustment: none
##
##           Estimate lower upper
## 0.01 - 0      0.511 -0.894  2.09
## 0.03 - 0     -0.405 -2.432  1.39
## 0.1 - 0       1.386  0.240  2.87
## 0.3 - 0       2.159  1.116  3.59
## 1 - 0         2.730  1.726  4.15
## 3 - 0         2.813  1.813  4.23
## 10 - 0        2.979  1.986  4.39
```

Instead of directly estimating the difference to the control, the coefficient vector can be reparameterized to represent the (log-) expected values for each concentration level by omitting the intercept in the model.

```
(m2 <- glm(foci ~ concf - 1, data = cta, family = poisson(link = "log"))

##
## Call:  glm(formula = foci ~ concf - 1, family = poisson(link = "log"),
##      data = cta)
##
## Coefficients:
##      concf0 concf0.01 concf0.03 concf0.1 concf0.3 concf1
##      -1.204   -0.693   -1.609    0.182    0.956    1.526
##      concf3 concf10
##      1.609    1.775
```

```
##
## Degrees of Freedom: 80 Total (i.e. Null); 72 Residual
## Null Deviance: 404
## Residual Deviance: 106 AIC: 283

# m1 and m2 are the same models with different parameterization
all.equal(deviance(m1), deviance(m2))

## [1] TRUE
```

The definition of a contrast matrix in `mcprofile` allows also to compute the same confidence intervals for the difference of concentration levels to the control, similar to the confidence intervals of the first modeling approach. The functionality of the package `multcomp` with the `contrMat` function can be used to use several pre-defined structures for the contrast matrix.

```
(cm2 <- contrMat(table(cta$concf), type = "Dunnett"))
```

```
##
## Multiple Comparisons of Means: Dunnett Contrasts
##
##      0 0.01 0.03 0.1 0.3 1 3 10
## 0.01 - 0 -1      1    0  0  0 0 0
## 0.03 - 0 -1      0    1  0  0 0 0
## 0.1 - 0 -1      0    0  1  0 0 0
## 0.3 - 0 -1      0    0  0  1 0 0
## 1 - 0 -1      0    0  0  0 1 0
## 3 - 0 -1      0    0  0  0 0 1
## 10 - 0 -1      0    0  0  0 0 0 1
```

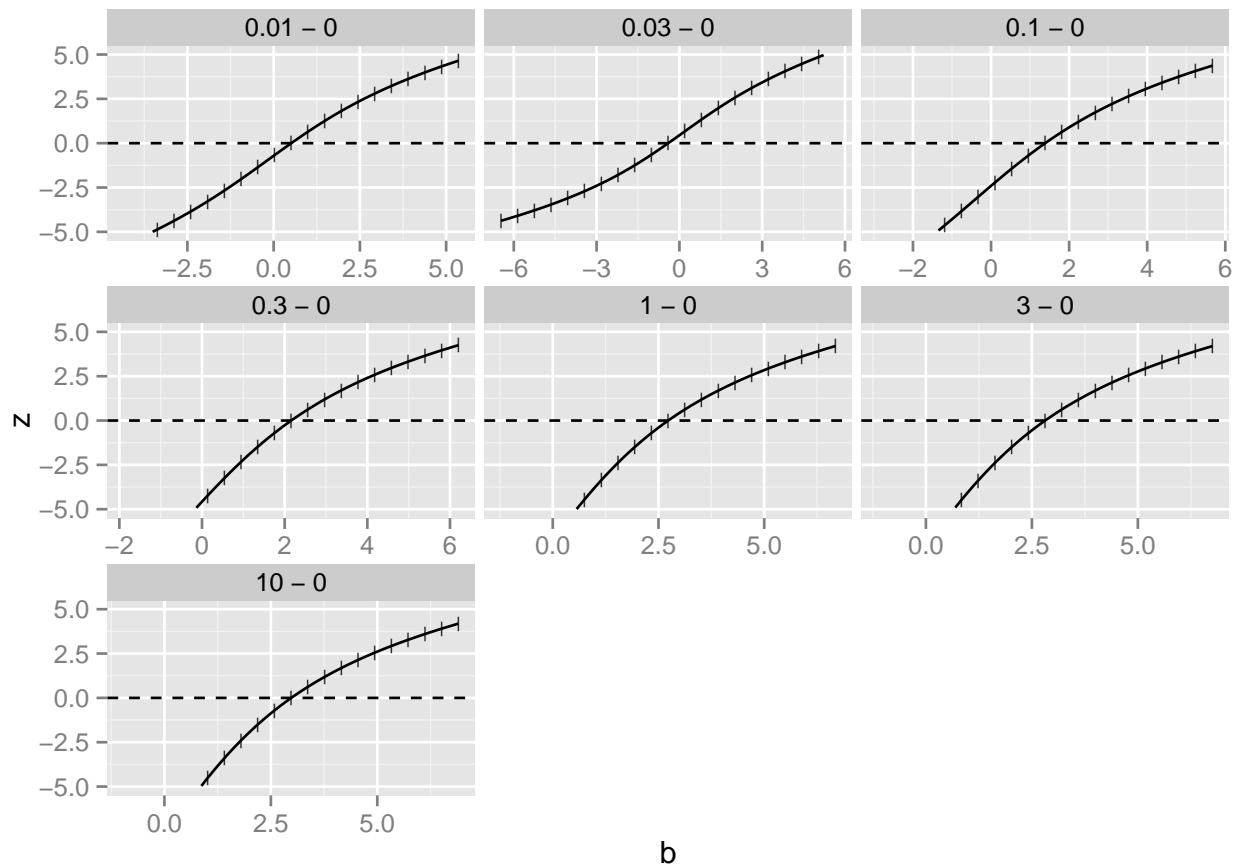
```
p2 <- mcprofile(m2, cm2)
(ci <- confint(p2, adjust = "none"))
```

```
##
## mcprofile - Confidence Intervals
##
## level: 0.95
## adjustment: none
##
##      Estimate lower upper
## 0.01 - 0      0.511 -0.894 2.09
## 0.03 - 0     -0.405 -2.432 1.39
## 0.1 - 0       1.386  0.240 2.87
## 0.3 - 0       2.159  1.116 3.59
## 1 - 0         2.730  1.726 4.15
## 3 - 0         2.813  1.813 4.23
## 10 - 0        2.979  1.986 4.39
```

### 3.1 plot methods

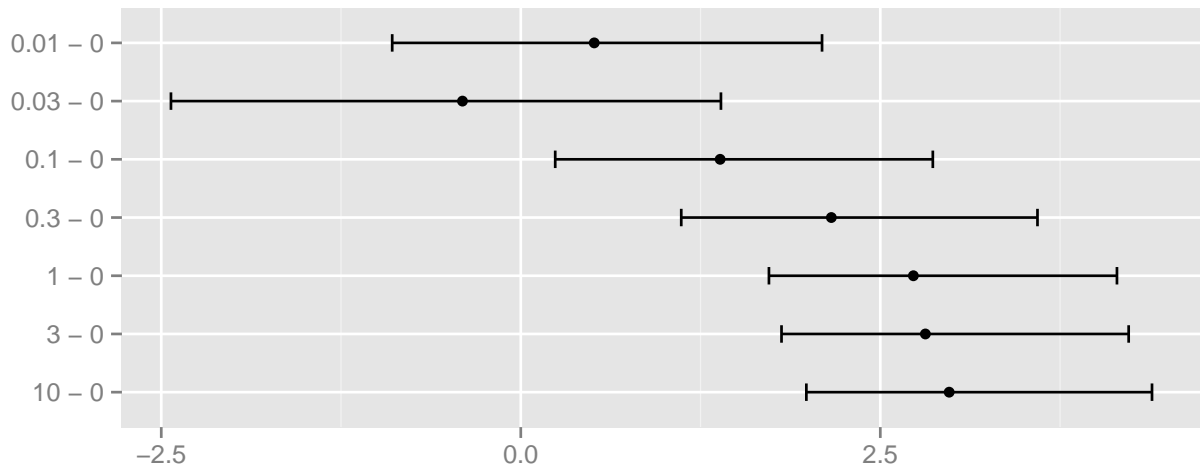
The plot functionality in `mcprofile` is based on the package `ggplot2`. The calculated profiles can simply be plotted by calling `plot`. The `ggplot2` function `ylim` is used to restrict the y-axis scale to a range  $[-5, 5]$

```
plot(p2) + ylim(c(-5, 5))
```



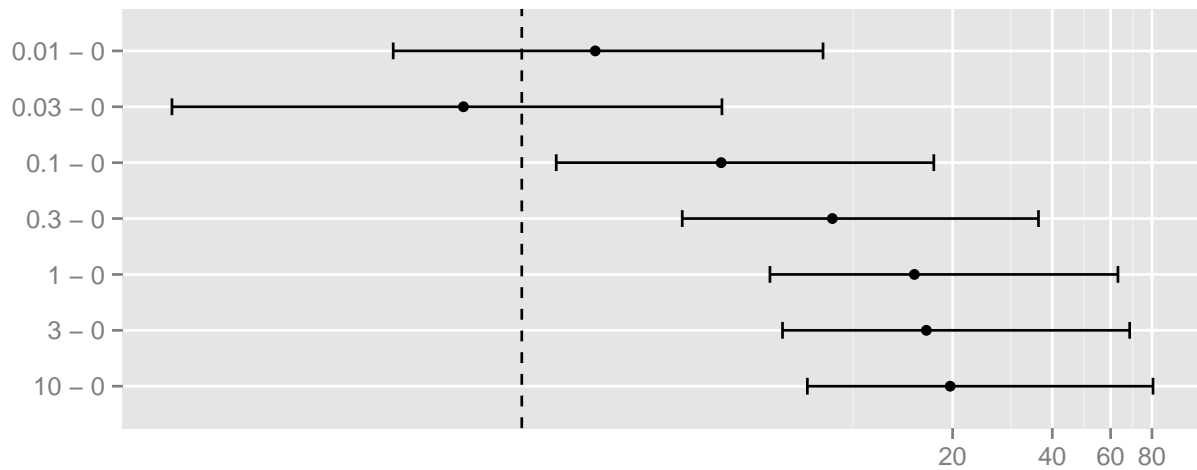
A visualization of the confidence intervals can be obtained by

```
plot(ci)
```



The confidence intervals are shown on the scale of the linear predictor. The `mcprofile` package contains an `exp` method to transform the scale. Further `ggplot2` functions can be used to change the plot layout.

```
plot(exp(ci)) +  
  coord_trans(x="log") +  
  geom_vline(xintercept=1, lty=2)
```



### 3.2 Modifying the profile support grid

As the default, the grid with support points, for which contrast coefficient values the profile statistics are calculated, is defined automatically. With the argument `grid`, a matrix with support points in each column defines the sequence of points in the parameter space of the contrast parameters defined in each row of the contrast matrix.

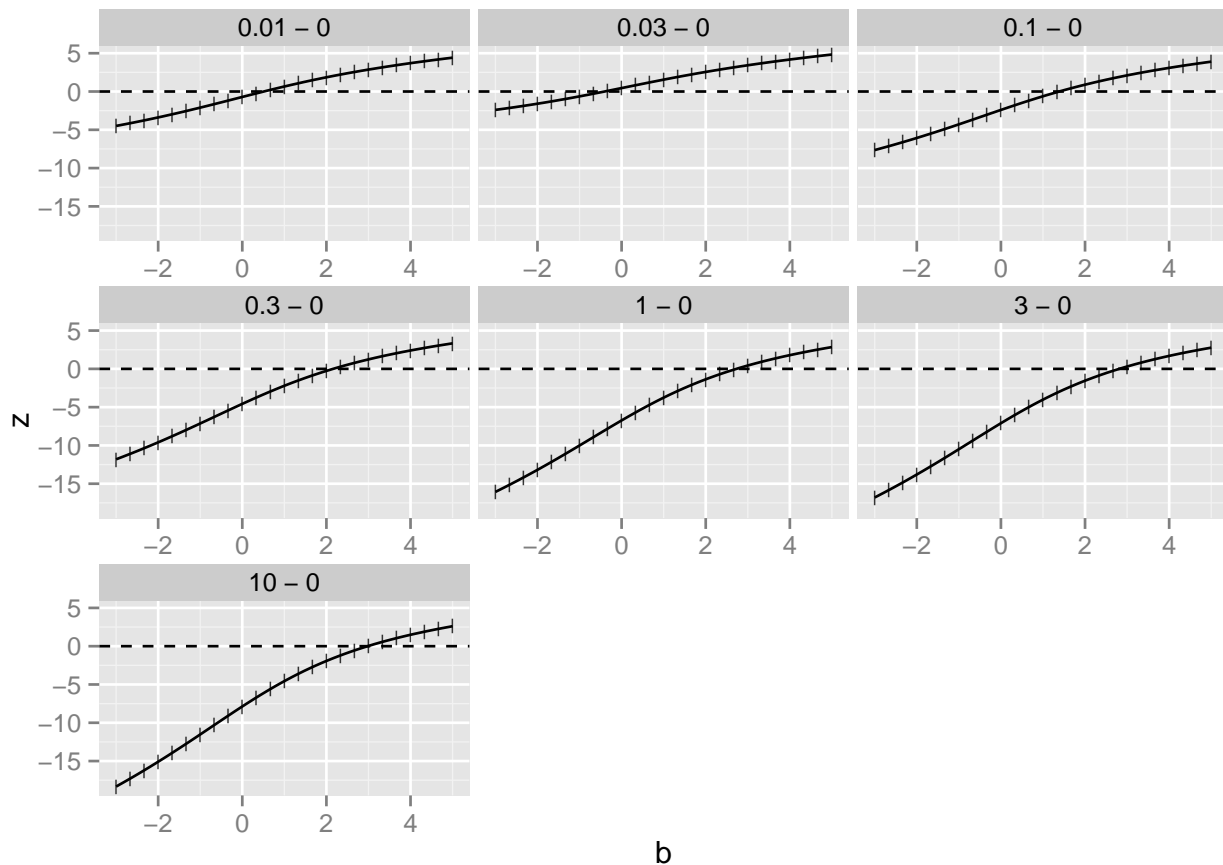
For example, the profiles can be calculated for a range of 25 support values between  $[-3, 5]$ .

```
cgrid <- sapply(1:nrow(cm2), function(i) seq(-3, 5, length=25))
head(cgrid)

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
## [1,] -3.000 -3.000 -3.000 -3.000 -3.000 -3.000 -3.000
## [2,] -2.667 -2.667 -2.667 -2.667 -2.667 -2.667 -2.667
## [3,] -2.333 -2.333 -2.333 -2.333 -2.333 -2.333 -2.333
## [4,] -2.000 -2.000 -2.000 -2.000 -2.000 -2.000 -2.000
## [5,] -1.667 -1.667 -1.667 -1.667 -1.667 -1.667 -1.667
## [6,] -1.333 -1.333 -1.333 -1.333 -1.333 -1.333 -1.333

p3 <- mcprofile(m2, cm2, grid=cgrid)
plot(p3)
```





### 3.3 Simultaneous confidence intervals and multiple testing

The default, when using the `confint` function, is to adjust for multiplicity by a single-step method, calculating a quantile of a multivariate normal or *t*-distribution. As an alternative the multiplicity adjustment by the Bonferroni method is available.

```
(ci2 <- confint(p2, adjust = "single-step"))
```

```
##
##   mcprofile - Confidence Intervals
##
## level:    0.95
## adjustment: single-step
##
##      Estimate   lower upper
## 0.01 - 0      0.511 -1.2491  2.55
## 0.03 - 0     -0.405 -3.0521  1.87
## 0.1 - 0       1.386 -0.0174  3.31
## 0.3 - 0       2.159  0.8960  4.04
## 1 - 0         2.730  1.5193  4.59
## 3 - 0         2.813  1.6082  4.67
## 10 - 0        2.979  1.7834  4.83
```

Compatible tests are available with the `summary` method. Here, further multiple testing adjustment is available through the function `p.adjust`. A single test margin can be specified to define the Null-hypotheses together with a test direction. A test if the difference of the average log-number of foci is larger than 1 for any concentration level compared to the control can be calculated by

```
summary(p2, margin = 1, alternative = "greater")
```

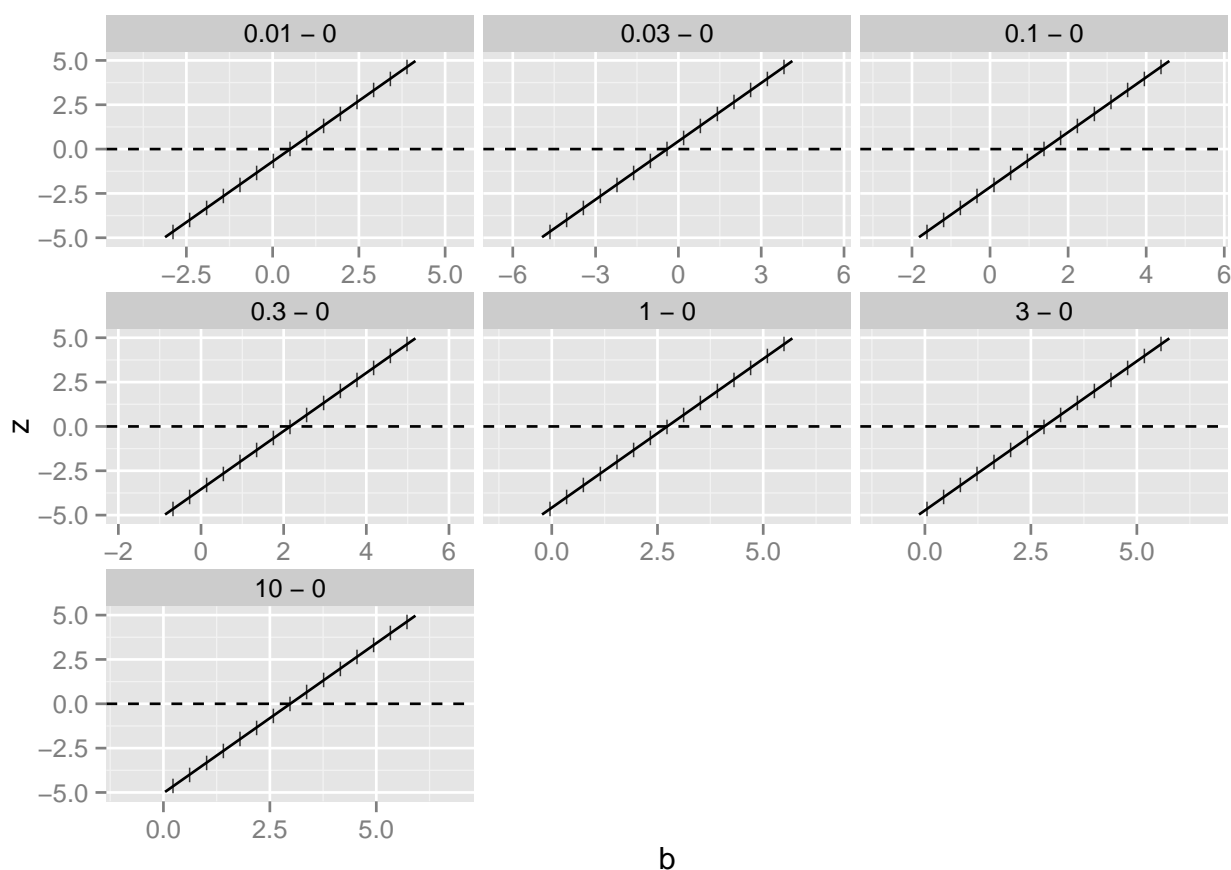
```
##
##   mcprofile - Multiple Testing
```

```
##
## Adjustment:  single-step
## Margin:      1
## Alternative:  greater
##
##           Estimate Statistic Pr(>z)
## 0.01 - 0 <= 1    0.51    -0.65  0.924
## 0.03 - 0 <= 1   -0.41   -1.55  0.993
## 0.1 - 0 <= 1     1.39     0.62  0.507
## 0.3 - 0 <= 1     2.16     2.21  0.043 *
## 1 - 0 <= 1       2.73     3.76 <0.001 ***
## 3 - 0 <= 1       2.81     4.02 <0.001 ***
## 10 - 0 <= 1      2.98     4.56 <0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## 4 Wald-type statistics

Instead of the likelihood root, the Wald-type statistic presented in Section 2.3.1 can be used to construct a profile. The function `wald` directly transforms the profile object.

```
w2 <- wald(p2)
plot(w2) + ylim(c(-5, 5))
```



The simultaneous confidence intervals are exactly the same as the ones obtained with the function `glht` in package `multcomp`.

```
confint(glht(m2, cm2))
##
## Simultaneous Confidence Intervals
```

```
##
## Multiple Comparisons of Means: Dunnett Contrasts
##
##
## Fit: glm(formula = foci ~ concf - 1, family = poisson(link = "log"),
##       data = cta)
##
## Quantile = 2.438
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##           Estimate lwr      upr
## 0.01 - 0 == 0   0.511  -1.270  2.291
## 0.03 - 0 == 0  -0.405  -2.631  1.820
## 0.1 - 0 == 0   1.386  -0.187  2.960
## 0.3 - 0 == 0   2.159   0.673  3.646
## 1 - 0 == 0     2.730   1.277  4.183
## 3 - 0 == 0     2.813   1.364  4.263
## 10 - 0 == 0    2.979   1.536  4.422

confint(w2)

##
##      mcprofile - Confidence Intervals
##
## level:    0.95
## adjustment: single-step
##
##           Estimate  lower upper
## 0.01 - 0     0.511 -1.269  2.29
## 0.03 - 0    -0.405 -2.631  1.82
## 0.1 - 0      1.386 -0.187  2.96
## 0.3 - 0      2.159  0.673  3.65
## 1 - 0        2.730  1.277  4.18
## 3 - 0        2.813  1.364  4.26
## 10 - 0       2.979  1.536  4.42
```

## 5 Modified likelihood root

At small sample sizes, the likelihood root statistic can be exchanged with the modified likelihood root, presented in Section 2.3.3, to improve the performance of the confidence intervals.

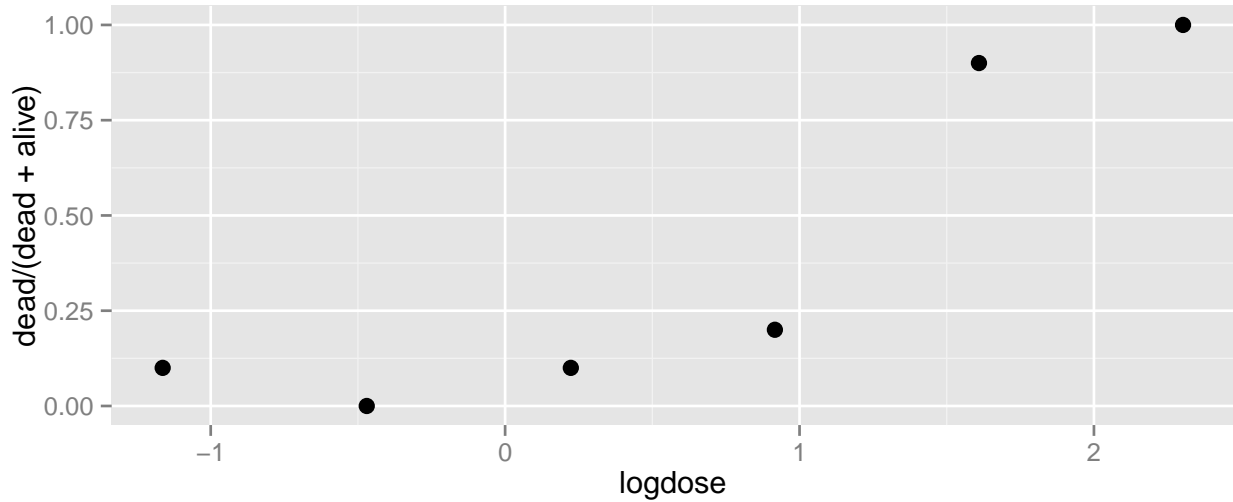
```
h2 <- hoa(p2)
confint(h2)

##
##      mcprofile - Confidence Intervals
##
## level:    0.95
## adjustment: single-step
##
##           Estimate  lower upper
## 0.01 - 0     0.511 -1.2533  2.50
## 0.03 - 0    -0.405 -2.9578  1.86
## 0.1 - 0      1.386 -0.0427  3.23
## 0.3 - 0      2.159  0.8658  3.94
## 1 - 0        2.730  1.4878  4.49
## 3 - 0        2.813  1.5766  4.57
## 10 - 0       2.979  1.7516  4.73
```

## 6 Simultaneous confidence intervals for predictions in logistic regression models

As a more advanced example, point-wise simultaneous confidence bands are computed based on a logistic regression model. In a dose-response experiment the lethal effect of an insecticide is tested. The data example is artificially generated, based on an excerpt of a real experiment, assuming very small sample sizes.

```
data(toxinLD)
toxinLD$logdose <- log(toxinLD$dose)
ggplot(toxinLD, aes(x=logdose, y=dead/(dead+alive))) +
  geom_point(size=3)
```



The dose-response curve is modeled by a logistic regression model with

$$y_i \sim \text{Binomial}(n_i, \mu_i), \quad \eta_i = \log\left(\frac{\mu_i}{1 - \mu_i}\right), \quad \eta_i = \beta_1 + \log(x_i)\beta_2.$$

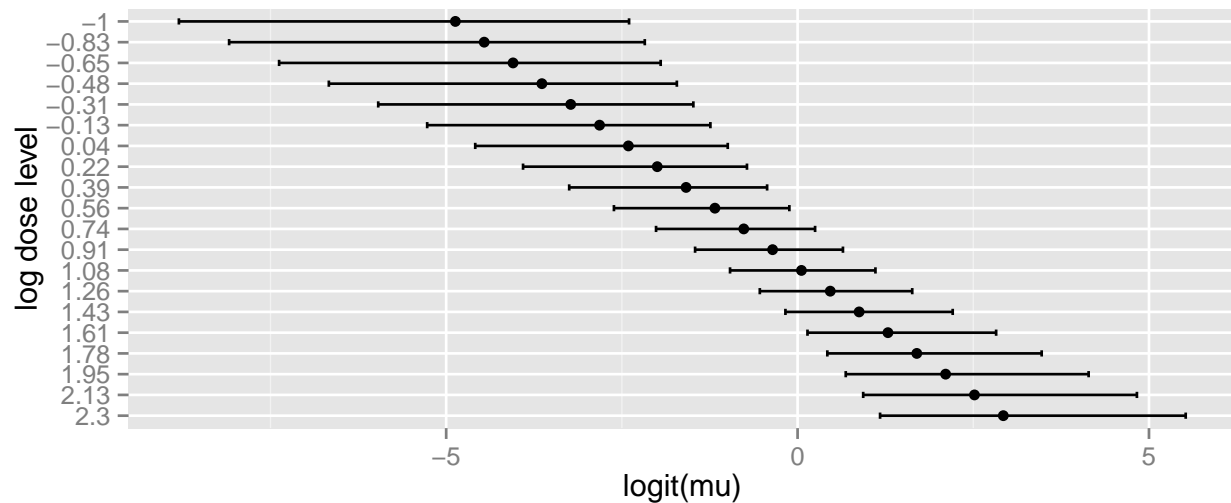
```
m1r <- glm(cbind(dead, alive) ~ logdose, data = toxinLD, family = binomial(link = "logit"))
```

The transformed parameters  $\vartheta_k = \sum_{j=1}^p a_{kj}\beta_j$  are representing predictions for prespecified dose levels  $a_{k2}$  and intercept design coefficients  $a_{k1}$ . Predictions should be made for 20 logarithmic dose values between  $[-1, 2.3]$ .

```
pdose <- seq(-1, 2.3, length=20)
cmlr <- model.matrix(~ pdose)
rownames(cmlr) <- round(pdose, 2)
head(cmlr)
```

```
##      (Intercept)  pdose
## -1             1 -1.0000
## -0.83           1 -0.8263
## -0.65           1 -0.6526
## -0.48           1 -0.4789
## -0.31           1 -0.3053
## -0.13           1 -0.1316
```

```
plr <- mcprofile(m1r, cmlr)
cplr <- confint(plr)
plot(cplr) +
  xlab("logit(mu)") +
  ylab("log dose level")
```



The confidence bands can be presented in a graphic as a shaded area around the predicted curve.

```
pdat <- data.frame(logdose=pdose)
pdat$estimate <- mlr$family$linkinv(cilr$estimate$Estimate)
pdat$lower <- mlr$family$linkinv(cilr$confint$lower)
pdat$upper <- mlr$family$linkinv(cilr$confint$upper)
ggplot(toxinLD, aes(x=logdose, y=dead/(dead+alive))) +
  geom_ribbon(data=pdat, aes(y=estimate, ymin=lower, ymax=upper), size=0.95, fill="lightblue") +
  geom_line(data=pdat, aes(y=estimate), size=0.95) +
  geom_point(size=3) +
  ylab("Mortality rate")
```

