Supplemental Material for the paper:
Analysing chemical-induced changes in macroinvertebrate communities in aquatic mesocosm experiments: A comparison of methods

Eduard Szöcs, Paul J. van den Brink, Laurent Lagadic, Thierry Caquet, Marc Roucaute, Arnaud Auber, Yannick Bayona, Matthias Liess, Peter Ebke, Alessio Ippolito, Cajo J. F. ter Braak, Theo C.M. Brock, Ralf B. Schäfer

October 29, 2014
1 Additional figures (Inter-replicate variability)

Figure 1.1: Kernel density and box plots of multivariate inter-replicate variability (\( \sigma \)) of the studies. Data pooled from treatment and sampling event per study. Study ID refers to Table 1 in the paper.
Figure 1.2: Inter-replicate variability depending on the proportion of zero counts in the samples. Each point represents a sampling event and treatment.

Figure 1.3: Inter-replicate variability of the studies sorted by the total number of taxa in the studies.
Figure 1.4: Inter-replicate variability over study period. Each line represents a treatment.

Figure 1.5: Inter-replicate variability of the studies sorted by the number of replicates for each treatment.
2 Additional figures (method comparison)

Figure 2.1: **Full rank partial RDA vs. GLMmv.** p-values analysing the 11 datasets with full rank partial RDA and Generalized Linear Models for multivariate data. The point of equal performance at p = 0.001 comprises studies 2, 6 and 8. Black line represents the 1:1 line, dashed lines show an alpha-level of 0.05. Study ID refers to table 1.
Figure 2.2: **Aggregated endpoints vs. PRC.** Performance of aggregated endpoints compared to PRCs. p-values of treatment effects per sampling. Black line represents the 1:1 line, dashed lines indicate an alpha-level of 0.05. p-values for ANOVA have been truncated to 0.001 for graphical representation. Study ID refers to table 1.
3 Tutorial: Analysing mesocosm data with R

3.1 Why use R to analyse ecotoxicological data?

There are many reasons to use R for statistical analysis in ecotoxicology:

**It is free.**

**It is platform independent.** Runs on Windows, Mac, Linux.

**It has enormous functionality.** Over 5000 packages extend the functionality - some of them not yet available in other software.

**It is open-source.** Everyone can take a look at the source code to see the exact computation. This is not the case with commercial software. This also enables to spot bugs.

**It facilitates reproducible research.** If the raw data is available and code is distributed everyone can reproduce and check the results.

In the next sections we will show how mesocosm experiments can be analysed using R. Some packages need to be installed before running these examples: `vegan` [Oksanen et al., 2013], `mvabund` [Wang et al., 2014], `multcomp` [Hothorn et al., 2008], `reshape2` [Wickham, 2007] and `ggplot2` [Wickham, 2009] all available on official package repository (The Comprehensive R Archive Network (CRAN), cran.r-project.org).

3.2 Example data

We will analyse the pyrifos data set of [van den Brink and ter Braak, 1999] which is shipped with the `vegan` package. Twelve experimental ditches were used in this experiment: Four ditches served as control and the remaining eight were treated in duplicates once with in the insecticide chlorpyrifos at doses of 0.1, 0.9, 6 and 44 µg / L. Invertebrates were sampled 11 times from week -4 pre-treatment through week 24 post-treatment. A total of 178 taxa were identified, this resulted in a table of 132 rows (11 * 12 samples) and 178 columns (taxa).
Rows correspond to samples and columns are the species (with abbreviated names) - a species x sites matrix as commonly used in community ecology. The rownames code treatment and time, but we will create a separate data.frame with variables providing information on the experimental ditch, sampling time and treatment:

```r
week <- gl(11, 12, labels = c(-4, -1, 0.1, 1, 2, 4, 8, 12, 15, 19, 24))
dose <- factor(rep(c(0.1, 0, 0, 0.9, 0, 44, 6, 0.1, 44, 0.9, 0, 6), 11))
ditch <- gl(12, 1, length=132)
```

Let's have a first look at the abundance of some taxa during the experiment:

The plot (Figure 3.1) shows the abundances during the experiment and for different treatments. Lines represent a non-parametric (LOESS) smooth to show the main patterns in the data. The red vertical line indicates the date of chlorpyrifos application.

From this plot we may conclude that taxa show very different responses, like

**Negative effects with recovery** *Caenis horaria, Cloeon dipterum and Chaoborus obscursipes*

**Negative persisting effects** *Gammarus pulex*

**Positive effects** Oligochaeta and *Bithynia tentaculata*

**No effects** Agrypnia complex

**Rare taxa** Libellulidae

### 3.3 Principal Response Curves (PRC)

#### 3.3.1 Introduction

Principal Response Curves (PRCs) ([van den Brink and ter Braak, 1998, 1999](#)) is the most widely used method to analyse community-level responses from mesocosm experiments. PRC belongs to the multivariate methods of constrained ordination and is a special form of Redundancy Analysis (RDA), the multivariate extension of linear regression.
# subset species

```r
take <- c('binitent', 'olchaeta', 'caenhora', 'cloedipt', 'chaoobsc', 'gammple', 'libellae', 'agdasphr')

abu <- pyrifos[, names(pyrifos) %in% take]

names(abu) <- c('Oligochaeta', 'Bithynia tentaculata', 'Gammarus pulex', 'Caenis horaria', 'Cloeon dipterum', 'Libellulidae', 'Chaoborus obscuripes', 'Agrypnia/Dasystegia/Phryganea')
```

# data has been ln-transformed - back-transformation to raw abundances

```r
abu_t <- round((exp(abu) - 1)/10)
```

# join with enviromental variables and bring to long format,

```r
require(reshape2)

dfm <- melt(data.frame(dose, week, abu_t), id.vars = c('dose', 'week'))

# week should be numeric for plot

dfm$week <- as.numeric(as.character(dfm$week))
```

```r
# x-axis: week, y-axis: abundances (log-scale), color: doses, splitted by taxon)
require(ggplot2)

ggplot(dfm, aes(x = week, y = value + 1, col = dose)) +
  geom_point() +
  geom_smooth(aes(group = dose), se = FALSE) +
  facet_wrap(~variable, scales = 'free_y') +
  geom_vline(aes_string(xintercept = 0), col = "red") +
  scale_y_log10() +
  theme_bw() +
  ylab('Abundance + 1') +
  xlab('Week')
```

---

**Figure 3.1**: Abundances (on a log-scale) of eight representative taxa showing different responses during a mesocosm study with chlorpyrifos.
As the temporal changes are of minor interest, these are 'partialled out', i.e., the effect of time is fitted to the data and the residuals are used in further analysis (partial RDA). This removes the effect of time from the response. A RDA model is then fitted using treatment and its interaction with time as predictors (Figure 3.2 lower left).

Results of this ordination are usually presented in a diagram, where the first axis of the resulting diagram displays the maximum variation explained by the treatment and treatment × time interaction. If responses are complex also subsequent PRC axes could be explored. This allows a large variety of species responses to be represented by a limited number of diagrams (van den Brink and ter Braak, 1998). RDA preserves the Euclidean distance between samples, therefore abundances should be transformed before analysis to avoid a strong influence of taxa with high abundances.

From such an ordination, one can obtain different information. The three most common in ecotoxicology are:

- Deviations of treated communities from control: Displayed as the mean difference of site scores between treatment and control on first axis over time. This plot shows the sampling date on the x axis and deviations in communities from the control on the y axis (Figure 3.2 upper right).

- Taxa responsible for the observed treatment-related differences: Species scores on the first axis indicate the contribution of taxa to the observed pattern. Taxa with greater absolute score show the principal response curve clearer than those with smaller score (Figure 3.2 middle right).

- Test of significance: Whether the first PRC axis displays a statistically significant amount of variation can be tested using permutations (Figure 3.2 lower right).

Species scores should be interpreted with caution and a low taxon weight does not necessarily translate to a small response (van den Brink and ter Braak, 1999). If a taxon shows a response that is strong, but different from the global pattern, this will result in a low species score. In the example *G. pulex* (Figure 3.1) shows a persistent effect without recovery, but has a relatively low taxon weight. In part this may be explained as well by the relative low abundance of *G. pulex*, because high abundant taxa that show a treatment-related effect may dominate the global response pattern, although this effect is partly down-weighted by the log-transformation.

If the first PRC axis is significant and shows a treatment × time interaction (i.e., non-parallel lines), the nature of this interaction can be further explored with separate RDAs for every sampling event. This allows community recovery to be determined, which is assumed when per-
Figure 3.2: Principal Response Curves as a special form of partial RDA for the eight species from figure 3.1. Species data should be transformed to avoid overly influence of abundant taxa. Fit a partial RDA to abundance data, partialling out the time effect and using treatment and the treatment x time interaction as explanatory variables. The first axis displays the highest fraction of variation that can be explained by the explanatory variables. The mean difference of site scores between treatment and control on first axis displays the deviation of communities from control. Species scores indicate species responsible for this pattern. Finally, the significance of the first axis can be tested.
mutation tests fail to detect a treatment effect at consecutive time points. Permutation tests can also be performed to test every treatment against the control. When replication is insufficient, alternatively a univariate test can be performed on the sample scores on the first PCA axis (see van den Brink et al. (1996) for a more detailed explanation).

### 3.3.2 Analysis of overall treatment effect

To calculate PRCs we can use the `vegan` package and the `prc()` function therein. It takes the abundance data (it is already log-transformed) as response and the treatment and time as explaining variables:

```r
pyrifos_prc <- prc(response = pyrifos, treatment = dose, time = week)
pyrifos_prc_sum <- summary(pyrifos_prc, scaling = 1)
```

A PRC plot can be produced using `plot()`, see Figure 3.3. Note that only species with scores greater or smaller than 1 are displayed to avoid cluttering.

The plot shows on the x axis the time and on the y-axis the difference from the control treatments. The farther apart from the x-axis the more different are the communities compared to the control group. We clearly see a treatment related effect: After application at week 0 treated communities change treatment-dependent. However, at the end of the experiment the treated and the control communities become similar again, which indicates a recovery.

On the right-hand side of this plot we see the species names and their scores. The more extreme the scores the more this species contributed to the observed pattern. However, you cannot directly infer from these species scores which species are more susceptible. For example *Gammarus pulex* (gammapule) has a relatively low score, although its response pattern (Figure 3.1) shows a strong response, but without recovery. PRC displays global pattern in the community, but the pattern of G. pulex is different from most other species, therefore it has a lower species score.

We can also look at the numerical output for this plot using the `summary` method (only a shortened output is given here.):

```r
summary(pyrifos_prc, scaling = 1)
```

```
## Call:
## prc(response = pyrifos, treatment = dose, time = week)
## Species scores:
##    Simve Ostsp NauLa Strvi binitent caenhora
##  -2.688 -2.312 -4.847 -3.070  1.951 -5.768
##    caenluct cloedipt hytuinae ablaphmo cepogoae chaobsc
```
plot(pyrifos_prc, select = abs(pyrifos_prc_sum$sp) > 1, scaling = 1)

Figure 3.3: PRC of the pyrifos dataset.
This summary returns the numerical species and sites scores. The output of `prc()` gives more detailed information about the RDA model:

```r
pyrifos_prc
```

We see that 21.9% of the variance can be attributed to time (`Conditional`, this is removed in partial RDA), 33.5% can be explained by the treatment regime (`Constrained`) and 44.6% of residual variance (`Unconstrained`), which cannot be explained by time and treatment. The first RDA axis has an eigenvalue of 25.28. If we divide this eigenvalue by the sum of all eigenvalues, we obtain the proportion of explained variance which is displayed on the first axis.

```r
pyrifos_prc$CCA$eig[1]/sum(pyrifos_prc$CCA$eig) * 100
```

Since a PRC is related to RDA, we can also use the `summary()` function for `rda`-objects as a convenient way to access this information:
The output is not displayed because it is humongous. Note that the scores may be different to other programs (e.g. Canoco) because of different scalings used.

The significance of the PRC diagram can be tested via permutations. However, observations from an experimental ditch are not independent, since the same ditch was measured repeatedly during the experiment. We have to take this into account: each ditch represents a time-series. Permuting the whole series of one ditch, keeping the temporal order, take this into account. To setup such a permutation scheme we can use the permute package, which is automatically loaded with vegan:

```r
control = how(plots = Plots(strata = ditch, type = "free"),
              within = Within(type = "none"),
              nperm = 199)
```

This sets the permutation scheme:

- **plots** Permute mesocosms, without any restrictions.
- **within** Within observations from one mesocosm there will be no permutations (keeping the time-series together).
- **nperm** We request 199 permutations.

Note that we requested for demonstration purpose only 199 permutations here, so the best achievable p-value is \(1/200 = 0.005\). Usually 1000 or more permutations should be used (giving a minimal p-value of \(1/1000 = 0.001\)). Permutation tests for first PRC axis can be performed using the `anova()` function of vegan:

```r
set.seed(1234)
anova(pyrifos_prc, permutations = control, first = TRUE)
```

```
## Permutation test for rda under reduced model
## Plots: ditch, plot permutation: free
## Permutation: none
## Number of permutations: 199
##
## Model: prc(response = pyrifos, treatment = dose, time = week)
##
## Df Variance   F Pr(>F)
## RDA1         1 25.282 15.096 0.01 **
## Residual    77 128.959
##
## Signif. codes:  0 '***'  0.001 '**'  0.01 '*'  0.05 '.'  0.1 ' ' 1
```

This runs a permutation test for the first eigenvalue of our model. We see that our first axis explains a statistically significant proportion of variation.
One can also test the significance of each axis separately (Legendre et al., 2011) - the results for the first axis are identical, but testing only the first eigenvalue is computationally more efficient since we are generally not interested in all axes.

```r
anova(pyrifos_prc, permutations = control, by = "axis")
```

or the terms in the model:

```r
anova(pyrifos_prc, permutations = control, by = "terms")
```

## Permutation test for rda under reduced model
## Terms added sequentially (first to last)
## Plots: ditch, plot permutation: free
## Permutation: none
## Number of permutations: 199
##
## Model: rda(formula = pyrifos ~ dose * week + Condition(week))
## Df Variance F Pr(>F)
## dose 4 30.695 4.5819 0.045 *
## week 0 0.000
## dose:week 40 65.989 0.9850 0.050 *
## Residual 77 128.959
##
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

This shows that the interaction term in the model is statistically significant. Note that no output for week is generated - this is because with the specified permutation scheme the significance of a week effect cannot be assessed.

### 3.3.3 Effects per week

The PRC indicates that there is a treatment x time interaction and we may want to look for effects at individual time-points to explore its nature.

We follow here van den Brink and ter Braak (1999) and use the ln-transformed nominal dose as continuous explanatory variable (regression-like analysis). However, dose can also be used as a categorical explanatory variable, giving a anova-like analysis. This code creates the log-transformed continuous dose:

```r
dose_c <- log(20 * as.numeric(levels(dose))[dose] + 1)
```

Now we need to compute for every week a RDA and run a permutation test. This could be done using a for-loop:
```r
rdas <- NULL
for (i in levels(week)) {
  take_spec <- pyrifos[week == i, ]
  take_dose <- dose_c[week == i]
  rdas[[i]]$rda <- rda(take_spec ~ take_dose)
  rdas[[i]]$anova <- anova(rdas[[i]]$rda, by = "terms",
                           permutations = 99)
}
rdas
```

This returns a very long list: one list entry per week and each entry itself contains two lists: `rda` (RDA-Model) and `anova` (permutation test). Digging through this manually is too laborious, though, we have to extract only a small part of the information. The results for week one are stored in the fourth position:

```r
evels(week)
## [1] "-4" "-1" "0.1" "1" "2" "4" "8" "12" "15" "19" "24"
rdas[[4]]
## $rda
## Call: rda(formula = take_spec ~ take_dose)
##
## Inertia Proportion Rank
## Total 259.4800 1.0000
## Constrained 77.2700 0.2978 1
## Unconstrained 182.2000 0.7022 10
## Inertia is variance
##
## Eigenvalues for constrained axes:
## RDA1
## 77.27
##
## Eigenvalues for unconstrained axes:
## PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8 PC9 PC10
## 32.98 28.99 25.51 21.03 17.45 15.63 12.72 11.18 10.01 6.70
##
## $anova
## Permutation test for rda under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 99
##
## Model: rda(formula = take_spec ~ take_dose)
## Df Variance  F Pr(>F)
## take_dose 1 77.271 4.2409 0.01 **
## Residual 10 182.205
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Or we can use the name of the week:

```r
levels(week)[18]
```
We can use `sapply()` to apply a function to every list entry and return results in a vector. For example, to extract the p-values for each week we can use:

```r
sapply(rdas, function(x) x$anova[1, 4])
```

We can find effects from the application date until week 19.

### 3.3.4 NOEC

Besides the overall significance of treatment and the effects per week ecotoxicologists are also interested which treatment differ from control. Usually, a no-observed-effect concentration (NOEC) [= the concentration below the lowest significant concentration] is derived from the data. However, the usage of NOEC has been criticised in the past. Testing via permutations fails here, because there are insufficient permutations left (we have only 2 treated and 4 control mesocosm per sampling date).
One solution is to break down the multivariate data into a univariate one and use a univariate test. To break down the community data into one variable one can use Principal Component Analysis (PCA), an unconstrained ordination technique, and take the sites scores on the first axis (which explains) as response variable. Usually a Williams-Test for trend is performed to look whether the treatment effect is statistically significant, but also a Dunnett-Test (comparing every treatment to control) could be applied.

Both (as contrast-versions) can be calculated using the multcomp package. Note that this contrast-based test is different to the original and widely used Williams test.

As in the previous section calculation are done within every week using a for-loop. For every week we need to:

1. Run a PCA.
2. Extract site scores on first axis.
3. Run a Dunnett/Williams-test on the site scores

The following runs these steps and returns the p-values for the comparisons using Dunnett's test:

```r
df <- data.frame(dose = dose, week = week)
require(multcomp)
out <- NULL
for (i in levels(week)) {
  # subset data to week
  take_spec <- pyrifos[week == i, ]
  # PCA-scores
  pca <- rda(take_spec)
  pca_scores <- scores(pca, display = "sites", choices = 1, scaling = 1)
  # linear model + multiple comparisons
  mod_aov <- aov(pca_scores ~ dose, data = df[week == i, ])
  out[[i]] <- summary(glht(mod_aov, linfct = mcp(dose = "Dunnett")))
}
```

This stores the multiple comparisons of PCA scores into a list, to access the results for week 1 we can use:

```r
out[['1']]```

```
## Simultaneous Tests for General Linear Hypotheses
## Multiple Comparisons of Means: Dunnett Contrasts
##
## Fit: aov(formula = pca_scores ~ dose, data = df[week == i, ])
## Linear Hypotheses:
```
| Estimate Std. Error t value Pr(>|t|) |
|----------------------------------|
| 0.1 - 0 == 0 -0.05056 0.41742 -0.121 0.99986 |
| 0.9 - 0 == 0 1.26898 0.41742 3.040 0.06102 ** |
| 6 - 0 == 0 2.48018 0.41742 5.942 0.00202 ** |
| 44 - 0 == 0 2.69986 0.41742 6.468 0.00115 ** |

---

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Adjusted p values reported -- single-step method)

In week 1 the NOEC would be 0.9 µg/L, as 6 µg/L is the lowest statistically significant treatment different to the control group. To perform a Williams contrast, change mcp(dose = "Dunnett") to mcp(dose = "Williams").

Again we could use an extractor function to gather the information we want from the list.

### 3.4 Generalized Linear Models for multivariate data

#### 3.4.1 Introduction

GLMs are a valuable tool for modelling data with an inherently not normally distributed response variable. Other distributions than the normal distribution should be used to model fractions (between 0 and 1), counts (positive integers) or presence/absence data (0 or 1). Generalized Linear Models for multivariate data (hereafter GLM mv) are the extension of GLMs to a multivariate response. As demonstrated recently, these may have higher statistical power compared to multivariate techniques like RDA that commonly have been used by ecologists [Warton et al., 2011].

GLM mv fit separate GLMs to each taxon in the dataset. Therefore, it generalizes the regression model approach of (partial) RDA, as it is also possible to specify a normal response distribution. Two commonly used distributions to model count data are the Poisson and the negative binomial distribution. GLM mv differ from PRCs in that they cannot perform dimension reduction. Model assumptions of individual GLMs can be checked using residual plots.

The individual GlM are then combined using sum-of-Likelihood-Ratios ( \( \sum LR \) ) statistic, which allows testing for a significant community response [Warton, 2011, Warton et al., 2011]. Univariate responses are directly available because a GLM is fitted to each taxon. Contributions of taxa to the community response can be derived from the deviance of the univariate GLMs. Different (nested) models could be fitted and compared. For example, the model for the abundance of taxon k \( (y_k) \):

\[
y_k \sim time + treatment + timetreatment
\]
Figure 3.4: Overview about Generalized Linear Models for multivariate data. Separate GLMs are fitted to each species (upper middle). These provide individual species responses (lower middle). Using the sum-of-Likelihood-Ratios from these models allows testing the community response. The deviance of each taxon indicates how much it contributes to the community response (upper right). Model assumptions can be checked using residual plots (lower right).
incorporates a treatment × time interaction, therefore allows the treatment effect to vary over time. This model can be compared to the model without interaction

\[ y_k \sim \text{time} + \text{treatment} \]

to test for statistical significance of the interaction. Comparing it to a model including only time as predictor,

\[ y_k \sim \text{time} \]

allows testing for any treatment related effect.

If there is a statistically significant interaction between treatment and time, this can be scrutinised by separate analyses per date. \( \text{GLM}_{mv} \) assess the significance analogous to PRCs via permutations, therefore restricted permutations should also be used.

### 3.4.2 Analysis of overall treatment effect

The data shipped with the vegan package have already been log-transformed. However, to use GLMs for multivariate data it is not necessary to transform abundances and therefore we first back-transform the abundances (this step is usually not needed, since raw counts are measured).

```r
pyrifos_t <- round((exp(pyrifos) - 1)/10)
```

To calculate \( \text{GLM}_{mv} \) we use the \texttt{mvabund} package. To investigate the overall treatment effect we first build a model with dose, week and their interaction as explaining variables:

```r
require(mvabund)
pyrifos_mv <- mvabund(pyrifos_t)
env <- data.frame(dose, week)
mod_full <- manyglm(pyrifos_mv ~ dose + week + dose:week, data = env)
```

This sets up the GLMs for every species using a negative binomial error distribution. Other possible distributions are: poisson (for count data), gaussian (linear regression) and binomial (for presence-absence data).

The \texttt{plot()} function shows a residual vs. fitted values plot and can used to check the model assumptions (Figure 3.5 there should be no obvious pattern in the residuals).

\( \text{GLM}_{mv} \) tests the significance by permutation, as with PRC we need to take the temporal autocorrelation into account. However, we need to construct a \textit{permutation matrix}, holding the permutations generated by the permutation design. This matrix is passed directly to \texttt{anova()} instead of passing the permutation design.
Figure 3.5: Residual vs. Fitted plot for the GLM model

```
manyglm(pyrifos_mv ~ dose + week + dose:week ...)
```
Significance can be tested using the `anova` function of mvabund:

```r
aov_mglm <- anova(mod_full, bootID = permutations, p.uni = "unadjusted",
                  test = "LR", resamp = "perm.resid")
```

Here we specify that

- `bootID = permutations` Use permutations from the specified permutation design.
- `p.uni = 'unadjusted'` Return univariate test statistics (unadjusted for multiple testing).
- `test = 'LR'` Use the sum-of-likelihood ratio as test statistics.
- `resamp = 'perm.resid'` permute residuals

As output we obtain the result of the multivariate test, as well as the univariate tests:

```r
## Analysis of Deviance Table
## Model: manyglm(formula = pyrifos_mv ~ dose + week + dose:week,
## data = pyrifos_env)
## Multivariate test:
## Res.Df Df.diff Dev Pr(>Dev)
## (Intercept) 131
## dose 127 4 1430 0.13
## week 117 10 4129 0.01 **
## dose:week 77 40 4111 0.01 **
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Univariate Tests:
## Simve Daplo Cerpu
## Dev Pr(>Dev) Dev Pr(>Dev) Dev
## (Intercept) 2.7 0.56 34.36 0.07 16.83
## dose 36.45 0.42 52.2 0.07 60.82
## week 104.74 0.09 27.52 0.21 5.43
## ....
## Arguments:
## Test statistics calculated assuming uncorrelated response
## (for faster computation).
## P-value calculated using 99 resampling iterations via residual
## permutation (without replacement) resampling (to account for
## correlation in testing.)
```
We see that there is a statistically significant interaction between treatment and time for the community, which indicates that the effect of treatment varies over time (*Multivariate Tests*). We also obtain univariate tests (*Univariate Tests*), the deviance indicates how much this species contributes to the community response.

Moreover we can compare different models:

```r
mod_reduced <- manyglm(pyrifos_mv ~ week, data = env)
aov_mglm2 <- anova(mod_reduced, mod_full, bootID = permutations, 
                   p.uni = "unadjusted", test = "LR", resamp = "perm.resid")
```

## Using bootID matrix from input.
## Time elapsed: 0 hr 3 min 36 sec

```r
aov_mglm2
```

```r
## Analysis of Deviance Table
##
## mod2: pyrifos_mv ~ week
## mod: pyrifos_mv ~ dose + week + dose:week
##
## Multivariate test:
## Res.Df Df.diff Dev Pr(>Dev)
## mod2 121
## mod 77 44 5998 0.01 **
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Univariate Tests:
## Simve Daplo Cerpu
## Dev Pr(Dev) Dev Pr(Dev) Dev Pr(Dev)
## mod2
## mod 123.55 0.10 49.95 0.12 31.21 0.35
```

Here we fitted a reduced model including only time as predictor and compared this to the model including treatment and it's interaction with time. The output indicates that the two models differ and therefore we can conclude that the treatment effect is substantial (including the interaction with time).

We can take a look at the species responsible for the difference (the deviance of single species GLMs):
This sort and display the 10 species with highest deviances. Comparing to figure 3.3 this gives similar results to PRC.

### 3.4.3 Effects per week

Now that we know that the treatment effect is varying with time, we can investigate this further and look at effects at individual time points. The concept is like in section 3.3.3: Loop through time and for every time point fit and test the model.

Here is a function that does this for us:

```r
mv_per_time <- function(response, treatment, time, nperm = NULL) {
  df <- data.frame(treatment = treatment, time = time,
                   stringsAsFactors = FALSE)
  out <- NULL
  for (i in levels(time)) {
    rsp <- mvabund(response[df$time == i, ])
    out[[i]]$mod <- manyglm(rsp ~ treatment, data = df[df$time == i, ])
    out[[i]]$anova <- anova(out[[i]]$mod, nBoot = nperm, p.uni = "unadjusted",
                           test = "LR", resamp = "perm.resid",
                           show.time = "none")
  }
  return(out)
}
```

And here we run it:

```r
per_week <- mv_per_time(pyrifos_mv, env$dose, env$week, nperm = 99)
```

The output is enormous (for every week a model with multivariate and univariate responses) and is skipped here.

P-values can be extracted for every week:

```r
sapply(per_week, function(y) y$anova$table[2, 4])
```

<table>
<thead>
<tr>
<th></th>
<th>-4</th>
<th>-1</th>
<th>0.1</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.7576</td>
<td>0.7778</td>
<td>0.1515</td>
<td>0.0101</td>
<td>0.0101</td>
<td>0.0101</td>
<td>0.0404</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>15</td>
<td>19</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.1313</td>
<td>0.2929</td>
<td>0.2525</td>
<td>0.4949</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Moreover, other information could be extracted: E.g., which taxa showed when the strongest response? Note that this is a different model than in section 3.3.3: there we used a log-transformed continuous explanatory variable, here we use treatment as factor (dummy variable), which explains the differences.
3.4.4 NOEC

Since GLM$_{mv}$ use also permutations to compute a p-value the same problem as with RDA arises: there are insufficient permutations left. Therefore we cannot use GLM$_{mv}$ to derive a NOEC and could use the PCA method (section 3.3.4).
References


SessionInfo

sessionInfo()

## R version 3.1.1 (2014-07-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8 LC_NUMERIC=C
## [3] LC_TIME=en_GB.UTF-8 LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=en_GB.UTF-8 LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=en_GB.UTF-8 LC_NAME=C
## [9] LC_ADDRESS=C LC_TELEPHONE=C
##
## attached base packages:
## [1] splines stats graphics grDevices utils datasets methods
## [8] base
##
## other attached packages:
## [1] mvabund_3.9.1 multcomp_1.3-7 TH.data_1.0-3 survival_2.37-7
## [5] mvtnorm_1.0-0 ggplot2_1.0.0 reshape2_1.4 vegan_2.1-41
## [9] lattice_0.20-29 permute_0.8-3 knitr_1.7
##
## loaded via a namespace (and not attached):
## [1] colorspace_1.2-4 digest_0.6.4 evaluate_0.5.5 formatR_1.0
## [5] grid_3.1.1 gtable_0.1.2 highr_0.4 labeling_0.3
## [9] MASS_7.3-35 munsell_0.4.2 plyr_1.8.1 proto_0.3-10
## [13] Rcpp_0.11.3 sandwich_2.3-2 scales_0.2.4 statmod_1.4.20
## [17] stringr_0.6.2 tools_3.1.1 tweedie_2.2.1 zoo_1.7-11