Todo list

EDUARD SZÖCS

QUANTITATIVE ECOTOXICOLOGY

WITH R!

This document was created using LaTeX, knitr and the tufte book class.

First draft, October 2013

Contents

List of Figures

List of Tables

1
Introduction

```
require(devtools)
install_github("qetx", "EDiLD")
```

```
require(qetx)
```

2
The Measurement Process

2.1 Winsorized Mean and Standard Deviation

The following sulfate concentrations (mg/L) were measured during a routine water quality survey of the Savannah River (South Carolina). The data is available in the `qetx` package [1]:

[1] Note that in this case you do not have to assign the data to a name.

```
data(so4)
```

```
so4

##  [1] 1.3 2.3 2.6 3.3 3.5 3.5 3.6 4.0 4.1 4.5 5.2 5.6
## [13] 5.7 6.1 6.2 6.5 6.9 7.1 7.7 7.9 9.9

length(so4)

## [1] 21

mean(so4)

## [1] 5.119

sd(so4)

## [1] 2.137
```

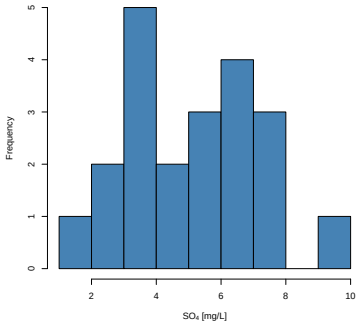

Figure 2.1: A histogramm of the so4 data.

So there are 21 measurements with a mean of 5.12 mg/L and a standard deviation of 2.14 mg/L.

Suppose we have a detection limit of 2.5 mg/L and want to winsorize values below LOD, i.e. replace the two lowest values by 2.6 mg/L and the two highest values by 7.7 mg/L.

Happily there is function in the `qetx` to do this for us: `winsor()`. This function takes a vector of values and a second argument specifying how many values should be winsorized (either by giving a LOD-value or the number of values on each side) [2].

[2] Look at the source of this function - type the function name into the console - to see which computations are done.

```
so4_w <- winsor(so4, lod = 2.5)
so4_w

##  [1] 2.6 2.6 2.6 3.3 3.5 3.5 3.6 4.0 4.1 4.5 5.2 5.6
## [13] 5.7 6.1 6.2 6.9 6.5 7.1 7.7 7.7 7.7
## attr(,"width")
## [1] 2
```

This give the expected results, moreover we see that on each end two observations where modified [3].

[3] stored within the attribute 'width' of the resulting vector. **TODO: verbatim within sidenote.**

```
mean(so4_w)

## [1] 5.081

sd(so4_w)

## [1] 1.792

sd_winsor(so4_w)

## [1] 2.24
```

The Winsorized mean ($\bar{x}_w$) now is 5.08 mg/L, the standard deviation of the modified data set (s) is 1.79 mg/L and the Winsorized standard deviation (s_w) 2.24 mg/L.

2.2 Probability Plotting

3
Bioaccumulation

4
Tests for Detection of Chronic Lethal and Sub-lethal Stress

5
Lethal and Other Quantal Responses to Stress

5.1 Fitting dose-response models

6
Population and Metapopulation Effects

7
Community Effects

7.1 Species Richness

7.2 Analyzing mesocosm data

Introduction

Principle Response Curves (PRC)[1] are commonly used for analyzing ecotoxicological mesocosm experiments. PRC analyses the change of a community due to a treatment over time and is a special form of Redundancy Analysis (RDA) [2].

[1] Van den Brink and Ter Braak, 1999

[2] Legendre and Legendre, 2013

Example data

Here we will analyze the pyrifos data set from the publication [3] which is shipped with the vegan package.

[3] Van den Brink and Ter Braak, 1999

Describe experiment

```
require(vegan)
data(pyrifos)
head(pyrifos[, c(1:8)])
```

```
##        Simve Daplo Cerpu Alogu Aloco Alore Aloaf Copsp
## w.4.c1 3.951     0     0     0     0     0     0 2.773
## w.4.c2 2.303     0     0     0     0     0     0 2.079
## w.4.c3 4.595     0     0     0     0     0     0 3.761
## w.4.c4 2.398     0     0     0     0     0     0 3.296
## w.4.c5 4.025     0     0     0     0     0     0 3.466
## w.4.c6 2.303     0     0     0     0     0     0 2.197
```

So rows correspond to samplings and colums are the species (with abbreviated names), a usual species x sites matrix. The columnames code treatment and time, but we will create a separate data.frame with information about experimental ditch, sampling time and treatment:

```
ditch <- gl(12, 1, length = 132)
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))
pyrifos_env <- data.frame(ditch, week, dose)
```

Overall pattern

With this a hand we can easily calculate and plot [4] the PRC using the prc() function:

[4] Note that only species with scores greater or smaller than 1 are displayed to avoid cluttering of the plot

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

```
plot(pyrifos_prc, select = abs(pyrifos_prc_sum$sp) > 1,
    scaling = 1)
```

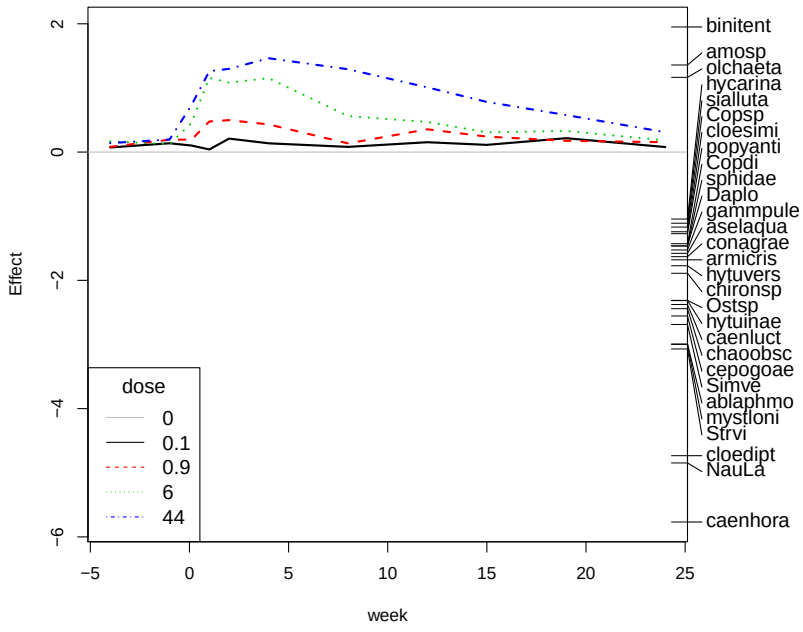

Figure 7.1: Principal response curves (PRC) with species weights for the pyrifos data set indicating effects of the insecticide on the invertebrate community.

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.

We see a clearly treatment-related effect: After application at week 0 the treated communities rapidly change treatment dependent. However to the end of the experiment the treated and the control get similar again, which indicates a 'recovery'.

On the righthand side we see the species names and their scores. The more extreme the scores the more this species contributed to the plotted differences. However, you cannot directly infer from these species scores which species are more susceptible. For example *Gammarus pulex* (gammapule) has a relatively low scores, although it's response pattern (Figure 7.2) show a strong response with no recovery. PRC displays global pattern in the community, but the pattern of *G. pulex* is different from most other species, therefore it gets a lower species scores.

We can also look at the numerical output[5] for this plot using the summary method:

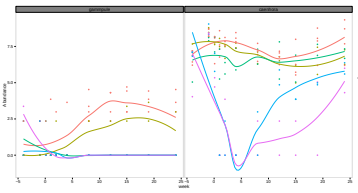

Figure 7.2: Different responses of *Gammarus pulex* and *Caenis horaria* during the experiment to chlorpyrifos treatments.

[5] Only a shortened output is given here.

```
pyrifos_prc_sum
```

```
##
## 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 chaoobsc
##   -2.376   -4.734   -2.316   -2.993   -2.555   -2.442
## mystloni
##   -2.998
##
## Coefficients for dose + week:dose interaction
## which are contrasts to dose 0
## rows are dose, columns are week
##          -4     -1    0.1       1      2      4
## 0.1 0.07218 0.1375 0.1020 0.04068 0.2101 0.1364
## 0.9 0.08106 0.1935 0.1936 0.47699 0.4977 0.4306
## 6   0.16616 0.1232 0.4539 1.15638 1.0835 1.1511
## 44  0.13979 0.1958 0.7308 1.26088 1.2978 1.4627
```

The output of prc() gives us more detailed information about the RDA model:

```
pyrifos_prc
```

```
## Call: prc(response = pyrifos, treatment = dose,
## time = week)
##
##               Inertia Proportion Rank
## Total         288.992      1.000
## Conditional    63.349      0.219   10
## Constrained    96.684      0.335   44
## Unconstrained 128.959      0.446   77
## Inertia is variance
##
## Eigenvalues for constrained axes:
##  RDA1  RDA2  RDA3  RDA4  RDA5  RDA6  RDA7  RDA8
## 25.28  8.30  6.04  4.77  4.15  3.86  3.59  3.33
##
## Eigenvalues for unconstrained axes:
##   PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8
## 17.16  9.19  7.58  6.06  5.73  4.84  4.52  4.10
## (Showed only 8 of all 77 unconstrained eigenvalues)
```

We see that 21.9 % of the variance can be attributed to time (Conditional), 33.5 % can be explaind 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 25.3. If we divide this eigenvalue by the sum of all eigenvalues, we get the proportion of explained variance which is displayed on the first axis[6]:

[6] rda() (and therefore als prc()) returns a huge object with all kind of information stored in it. See ?cca.object for the internal structure. Here I directly access the eigenvalues from this object

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

```
##  RDA1
## 26.15
```

The significance of the PRC diagram can be tested via permuations. However observations from a 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. We will permute the whole series of one ditch, keeping the temporal order.

For example, if we have 3 ditches observed for 4 weeks:

```
##    ditch week
## 1      1    1
## 2      1    2
## 3      1    3
## 4      1    4
```

```
## 5      2    1
## 6      2    2
## 7      2    3
## 8      2    4
## 9      3    1
## 10     3    2
## 11     3    3
## 12     3    4
```

One possible permutation would be

```
##    ditch week
## 9      3    1
## 10     3    2
## 11     3    3
## 12     3    4
## 1      1    1
## 2      1    2
## 3      1    3
## 4      1    4
## 5      2    1
## 6      2    2
## 7      2    3
## 8      2    4
```

To setup such a permutation scheme we use the permute package, which automatically loaded with vegan:

```
control = how(plots = Plots(strata = ditch, type = "free"),
    within = Within(type = "none"), nperm = 99)
```

This sets up our permutation scheme:

plots We will permute ditches, without any restrictions.

within But within one ditch there will be no permutations

nperm We want 99 permutations

With this setup we can create a permutation matrix. Each row therein is one permuation, the values are the rownumbers of the original data set.

```
set.seed(1234)
permutations <- shuffleSet(nrow(pyrifos), control = control)
```

This can be passed to permutest, testing the first eigenvalue of our model. [7]

```
mod_perm <- permutest(pyrifos_prc,
                      permutations = permutations,
                      first = TRUE)
mod_perm
```

[7] At the moment the permute-package isn't fully hooked up into vegan. vegan is in active development and hopefully in the future we will be able to directly pass our permutation scheme.

```
##
## Permutation test for rda
##
## Call: prc(response = pyrifos, treatment = dose,
## time = week)
## Permutation test for first constrained eigenvalue
## Pseudo-F:  15.1 (with 1, 77 Degrees of Freedom)
## Significance:  0.01
## Based on 99 permutations under reduced model.
```

We see that our first axis shows us a statistically significant amount of variation. The minimum p-value that we could get is 0.01 (=1/no. of permutations). .

Hier noch mehr erklÃd'ren

Effects per week

After looking at the overall treatment effect, we may want to look at effects at individual time-points. We follow here [8] and use the ln-transformed nominal dose as continuous explanatory variable [9].

[8] Van den Brink and Ter Braak, 1999

[9] Beforehand we have to convert dose from a factor to a numeric vector via as.numeric(levels(x))[x]

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

No we could program a for-loop and compute for every week a RDA and a permutation test. However there is a convenience function in the `qetx` package:

```
rdas <- rda_per_time(pyrifos, dose_c, week)
```

```
sapply(rdas, function(x) x$anova[1, 5])

##   -4   -1  0.1    1    2    4    8   12   15   19   24
## 0.45 0.95 0.03 0.01 0.01 0.01 0.01 0.01 0.02 0.04 0.12
```

Other methods

Other methods to analyse mesocosm experiments include:

multivariate GLMs [10] In R: mvabund-package.

trait-based indicators [11] Currently no package, but look at rspear-package.

community endpoints [12] Can use vegan for all computations.

[10] Warton et al., 2011; and Wang et al., 2012

[11] Liess and Beketov, 2011

[12] Sanchez-Bayo and Goka, 2012

7.3 Species Sensitivity Distributions

```
require(fitdistrplus)
# or
require(drc)
```

R Session Info

```
sessionInfo()

## R version 3.0.2 (2013-09-25)
## Platform: x86_64-pc-linux-gnu (64-bit)
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8
##  [2] LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8
##  [4] LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8
##  [6] LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8
##  [8] LC_NAME=C
##  [9] LC_ADDRESS=C
## [10] LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8
## [12] LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets
## [6] methods   base
##
## other attached packages:
## [1] vegan_2.1-35    lattice_0.20-23 permute_0.7-4
## [4] qetx_0.0.1      knitr_1.5
##
## loaded via a namespace (and not attached):
## [1] codetools_0.2-8 digest_0.6.3    evaluate_0.5.1
## [4] formatR_0.9     grid_3.0.2      highr_0.2.1
## [7] stringr_0.6.2   tools_3.0.2
```

Bibliography

Legendre, P. and Legendre, L. (2013). *Numerical ecology*. Elsevier, Amsterdam; Boston.

Liess, M. and Beketov, M. (2011). Traits and stress: keys to identify community effects of low levels of toxicants in test systems. *Ecotoxicology*, 20(6):1328–1340.

Sanchez-Bayo, F. and Goka, K. (2012). Evaluation of suitable endpoints for assessing the impacts of toxicants at the community level. *Ecotoxicology*, 21(3):667–80. pdf RS.

Van den Brink, P. and Ter Braak, C. (1999). Principal response curves: Analysis of time-dependent multivariate responses of biological community to stress. *Environmental Toxicology and Chemistry*, 18(2):138–148.

Wang, Y., Naumann, U., Wright, S. T., and Warton, D. I. (2012). mvabund- an r package for model-based analysis of multivariate abundance data. *Methods in Ecology and Evolution*, 3(3):471âĂŞ474.

Warton, D. I., Wright, S. T., and Wang, Y. (2011). Distance-based multivariate analyses confound location and dispersion effects. *Methods in Ecology and Evolution*, 3(1):89–101.