# STATISTICAL ECOTOXICOLOGY

## TITLE2

by

EDUARD SZÖCS

from ZĂRNESTI / ROMANIA

Submitted Dissertation thesis for the partial fulfillment of the requirements for a
Doctor of Natural Sciences
Fachbereich 7: Natur- und Umweltwissenschaften
Universität Koblenz-Landau

August 24, 2016

# ACKNOWLEDGMENTS

# ABSTRACT

# PUBLICATIONS

This cumulative dissertation includes four scientific publications:

1. Eduard Szöcs and Ralf B. Schäfer (2015). "Ecotoxicology is not normal." In: *Environmental Science and Pollution Research* 22.18, pp. 13990–13999

2. two

3. three

4. Scott A. Chamberlain and Eduard Szöcs (2013). "taxize: taxonomic search and retrieval in R." in: *F1000Research* 2.191

This dissertation is based on publications written by multiple authors. Therefore, the first person plural is used throughout the thesis.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# 1

# INTRODUCTION AND OBJECTIVES

## 1.1 STATISTICAL ECOTOXICOLOGY

## 1.2 CHEMICAL POLLUTION OF FRESHWATER ECOSYSTEMS

## 1.3 DEALING WITH *big data* IN ECO(TOXICO-)LOGY

## 1.4 OBJECTIVES AND STRUCTURE OF THE THESIS

This thesis pursues three objectives:

i to scrutinize new methods in statistical ecotoxicology,

ii explore available monitoring data for ecological risk assessment and

iii provide tools for scientists to deal more effective with the data they generate.

The thesis starts with a comparison of statistical methods for ecotoxicological experiments (Chapter 2). Specific questions addressed were:

- Are statistical methods taking the nature of the data into account more powerful than currently used methods?
- How much statistical power do current experimental designs ecotoxicology have?

Chapters 3 (chemical data) and 4 (biological data) describe software solutions to simplify and accelerate the workflow of:

- validating chemical and taxonomic names
- link them to other dataset
- search properties and identifiers

# 2

# ECOTOXICOLOGY IS NOT NORMAL - A COMPARISON OF STATISTICAL APPROACHES FOR ANALYSIS OF COUNT AND PROPORTION DATA IN ECOTOXICOLOGY

Eduard Szöcs[a] & Ralf B. Schäfer[a]

[a]Institute for Environmental Sciences, University Koblenz-Landau, Landau, Germany

Adapted from the article published in 2015 in *Environmental Science and Pollution Research*, 22(18), 13990-13999.

## 2.1 ABSTRACT

Ecotoxicologists often encounter count and proportion data that are rarely normally distributed. To meet the assumptions of the linear model such data are usually transformed or non-parametric methods are used if the transformed data still violate the assumptions. Generalised Linear Models (GLM) allow to directly model such data, without the need for transformation. Here, we compare the performance of two parametric methods, i.e., (1) the linear model (assuming normality of transformed data), (2) GLMs (assuming a Poisson, negative binomial, or binomially distibuted response), and (3) non-parametric methods.

We simulated typical data mimicking low replicated ecotoxicological experiments of two common data types (counts and proportions from counts). We compared the performance of the different methods in terms of statistical power and Type I error for detecting a general treatment effect and determining the lowest observed effect concentration (LOEC). In addition, we outlined differences on a real world mesocosm data set.

For count data, we found that the quasi-Poisson model yielded the highest power. The negative binomial GLM resulted in increased Type I errors, which could be fixed using the parametric bootstrap. For proportions, binomial GLMs performed better than the linear model, except to determine LOEC at extremely low sample sizes. The compared non-parametric methods had generally lower power.

We recommend that counts in one-factorial experiments should be analysed using quasi-Poisson models and proportions from counts by binomial GLMs. These methods should become standard in ecotoxicology.

## 2.2 INTRODUCTION

Ecotoxicologists perform various kinds of experiments yielding different types of data. Examples are animal counts in mesocosm experiments (non-negative, integer-valued data) or proportions of surviving animals (data bounded between 0 and 1, discrete). These data are typically not normally distributed. Nevertheless, such data are often analysed using methods that assume a normal distribution and variance homogeneity (M. Wang and Riffel, 2011). To meet these assumptions data are usually transformed. For example, ecotoxicological textbooks (Michael C Newman, 2012) and guidelines (EPA, 2002; OECD, 2006) advise that survival data should be transformed using an arcsine square root transformation. For count data from mesocosm experiments a log(Ay + C) transformation is usually applied, where the constants A and C are either chosen arbitrarily or following general recommendations. For example, Brink et al. (2000) suggest to set the term Ay to be 2 for the lowest abundance value (y) greater than zero and C to 1. Other transformations, like the square root or fourth root transformation, are also commonly applied in community ecology (Anderson et al., 2011). Note that there has been little evaluation and advice for practitioners which transformations to use. If the transformed data still do not meet the assumptions of the linear model, non-parametric tests are usually applied (M. Wang and Riffel, 2011).

Generalised linear models (GLM) provide a method to analyse counts or proportions from counts in a statistically sound way (Nelder and Wedderburn, 1972). GLMs can handle various types of data distributions, e.g., Poisson or negative binomial (for count data) or binomial (for proportions); the normal distribution being a special case of GLMs. Despite GLMs being available for more than 40 years, ecotoxicologists do not regularly make use of them. Recent studies concluded that the linear model should not be applied on transformed data and GLMs be used as they have better statistical properties (O'Hara and Kotze 2010; Warton 2005 (counts), Warton and Hui 2011 (proportions from counts)).

Ecotoxicological experiments often involve small sample sizes due to practical constraints. For example, extremely low samples sizes ($n < 5$) are common in many mesocosm studies (Sanderson, 2002; Szöcs et al., 2015). Small sample sizes lead to low power in statistical hypothesis testing, on which many ecotoxiological approaches (e.g. risk assessment for pesticides) rely. Such an endpoint are L/NOEC values (Lowest / No observed effect concentration). Although their use has been heavily criticized in the past (Laskowski, 1995), they are the predominant endpoint in mesocosm experiments (Brock et al., 2015; EFSA PPR, 2013).

We explore how GLMs may enhance, when appropriately used, inference in ecotoxicological studies and compared three types of statistical methods (linear model on transformed data, GLM, non-parametric tests). We first illustrate differences between statistical methods using a data set from a mesocosm study. Then we further elaborate differences in detecting a general treatment effect and determining the LOEC using simulations of two common data types in ecotoxicology: counts and proportions from counts.

## 2.3 METHODS

### 2.3.1 *Models for count data*

#### 2.3.1.1 *Linear model for transformed data*

To meet the assumptions of the standard linear model, count data usually needs to be transformed. We followed the recommendations of Brink et al. (2000) and used a log(Ay + 1) transformation (eqn. 2.1):

$$Y_{new\ i} = \log(AY_i + 1) \tag{2.1}$$

, where $Y_i$ is the measured and $Y_{new\ i}$ the transformed abundance of the $i$th observation. The factor $A$ was chosen in such way that $AY$ equals 2 for the lowest non-zero abundance value ($Y$).

Then we fitted the linear model to the transformed abundances (hereafter LM):

$$\begin{gathered}
Y_{new\ i} \sim N(\mu_i, \sigma^2) \\
E(Y_{new\ i}) = \mu_i \text{ and } var(Y_{new\ i}) = \sigma^2 \\
\mu_i = \beta \times X_i
\end{gathered} \tag{2.2}$$

This model assumes a normal distribution of the transformed abundances. The expected value for each observation $i$ is given by its mean ($\mu_i$) and the variance ($\sigma^2$) is constant between treatments. We allow this mean to vary between treatments ($X_i$ codes the treatments) and $\beta$ are the estimated coefficients related to these changes in transformed abundances between treatments (eqn. 2.2).

#### 2.3.1.2 *Generalised Linear Models*

GLMs extend the linear model to variables that are not normally distributed. Instead of transforming the response variable, the counts could be directly modeled by a Poisson GLM ($GLM_p$):

$$\begin{gathered}
Y_i \sim P(\mu_i) \\
E(Y_i) = var(Y_i) = \mu_i \\
\log(\mu_i) = \beta \times X_i
\end{gathered} \tag{2.3}$$

This model assumes Poisson distributed abundances with mean $\mu_i \geqslant 0$. The expected value for each observation $i$ is given by its mean. Moreover, this model assumes that mean and variance are equal. We are modeling the mean as a function of treatment membership ($X_i$). However, to avoid negative values of the mean this is done on a log scale. Therefore, $\beta$ also describes the differences between treatments on a log scale (eqn. 2.3).

The assumption of equal mean and variance is rarely met with ecological data, which is typically characterized by greater variance than the mean (overdispersion). To overcome this problem a quasi-Poisson model ($GLM_{qp}$) could be used, which models the variance as a linear function of the mean (eqn. 2.4):

$$var(Y_i) = \phi\mu_i \tag{2.4}$$

Here, $\phi$ is used to account for additional variation and is known as overdispersion parameter. The quasi-Poisson model is a post hoc method, meaning that first a Poisson model is estimated (eqn. 2.3) and than the standard errors are scaled by the degree of overdispersion (Hilbe, 2014).

Another possibility to deal with overdispersion is to model abundances by a negative binomial distribution ($GLM_{nb}$, eqn. 2.5):

$$\begin{gathered}
Y_i \sim NB(\mu_i, \kappa) \\
E(Y_i) = \mu_i \text{ and } var(Y_i) = \mu_i + \mu_i^2/\kappa \\
\log(\mu_i) = \beta \times X_i
\end{gathered} \tag{2.5}$$

This models assumes that abundances are negative binomially distributed, with a mean of $\mu_i \geqslant 0$ and a variance $\mu_i + \mu_i^2/\kappa$. Similar to the Poisson model we use a log link between mean and treatments. Note, that the quasi-Poisson model assumes a linear mean-variance relationship (eqn. 2.4), whereas the negative binomial model assumes a quadratic relationship (eqn. 2.5).

The above described models are most commonly used in ecology (Ver Hoef and Boveng, 2007), although other distributions for count data are possible, like the negative binomial model with a linear mean-variance relationship (also known as NB1) or the poisson inverse gaussian model (Hilbe, 2014).

### 2.3.2 *Models for binomial data*

A binomial variable counts how often an event $x$ occurs in a fixed number of independent trials $N$ (e.g. *"5 out of 10 fish survived"*), with an equal probability of occurrence $\pi$ between trials. The number of times an event occurs can also be calculated as proportion $x/N$.

#### 2.3.2.1 *Linear model for transformed data*

To accommodate the assumptions for the standard linear model with such proportions, a special arcsine square root transformation (eqn. 2.6) is suggested (EPA, 2002; Michael C Newman, 2012):

$$Y_{new\ i} = \begin{cases} \arcsin(1) - \arcsin(\sqrt{\frac{1}{4n}}) & \text{, if } Y_i = 1 \\ \arcsin(\sqrt{\frac{1}{4n}}) & \text{, if } Y_i = 0 \\ \arcsin(\sqrt{Y_i}) & \text{, otherwise} \end{cases} \tag{2.6}$$

, where $Y_i$ are the untransformed proportions, $Y_{new\ i}$ are the transformed proportions, and $n$ is the total number of exposed animals per treatment. The transformed proportions are then analysed using the standard linear model (LM, eqn. 2.2). Note, that the coefficients of the linear model are not directly interpretable due to transformation.

#### 2.3.2.2 *Generalised Linear Models*

A more natural way to model such data is the binomial distribution with parameters $N$ and $\pi$ ($GLM_{bin}$):

$$\begin{gathered} Y_i \sim Bin(N, \pi_i) \\ E(Y_i) = \pi_i \times N \text{ and } var(Y_i) = \pi_i(1-\pi_i)/N \\ logit\ (\pi_i) = \beta \times X_i \end{gathered} \tag{2.7}$$

This model assumes that the number of occurrences ($Y_i$) are binomially distributed, where N = number of trials (e.g. exposed animals) and $\pi_i$ is the probability of occurrences (fish survived), which together give the expected number of occurrences. The variance of the binomial distribution is a quadratic function of the mean. We are modeling the probability of occurrence as function of treatment membership ($X_i$) and to ensure that $0 < \pi_i < 1$ we do this on a logit scale (eqn. 2.7). The estimated coefficients ($\beta$) of this model are directly interpretable as changes in log odds between treatments.

Non-independent trials (e.g. fish are grouped in aquaria) may lead to overdispersion (Williams, 1982). Methods to deal with overdispersed binomial data are for example quasi methods (see above) or Generalized Linear Mixed models (GLMM). However, these are not further investigated in this paper (see Warton and Hui (2011) for a comparison).

### 2.3.3 *Statistical Inference*

After model fitting the next step is statistical inference. Ecotoxicologists are generally interested in two hypotheses: (i) is there any treatment related effect? and (ii) which treatments show a treatment effect (to determine the LOEC)?

Following general recommendations (Bolker et al., 2009; Faraway, 2006), we used F-tests (LM and $\mathrm{GLM_{qp}}$) and Likelihood-Ratio (LR) tests ($\mathrm{GLM_{p}}$, $\mathrm{GLM_{nb}}$ and $\mathrm{GLM_{bin}}$) to test the first hypothesis. However, it is well known that the LR test is unreliable with small sample sizes (Wilks, 1938). Therefore, we additionally explored the parametric bootstrap (Faraway, 2006) to assess the significance of the LR. Bootstrapping is computationally very intensive and for this reason we applied it only for the LR test of the negative binomial models (using 500 bootstrap samples, denoted as $\mathrm{GLM_{npb}}$).

To assess the LOEC we used Dunnett contrasts (Dunnett, 1955) with one-sided Wald t tests (normal and quasi-Poisson models) and one-sided Wald Z tests (Poisson, negative binomial and binomial models). Beside these parametric methods we also applied two, in ecotoxicology commonly used, non-parametric methods: The Kruskal-Wallis test (KW) to test for a general treatment effect and a pairwise Wilcoxon test (WT) to determine the LOEC. We adjusted for multiple testing using the method of Holm (1979).

### 2.3.4 *Case study*

Brock et al. (2015) presents a typical example of data from mesocosm studies, which we use to demonstrate differences between methods. The data are mayfly larvae counts on artificial substrate samplers at one sampling date. A total of 18 mesocosms have been sampled from 6 treatments (Control (n = 4), 0.1, 0.3, 1, 3 mg/L (n = 3) and 10 mg/L (n = 2)) (Figure 2.1).

### 2.3.5 *Simulations*

#### 2.3.5.1 *Count data*

To further scrutinise the differences between methods we simulated data sets with known properties. We simulated count data that mimics the data of the case study with five treatments (T1 - T5) and one control group (C). Counts were drawn from a negative binomial distribution with overdispersion at all treatments ($\kappa = 4$, eqn. 2.5). We simulated data sets with different number of

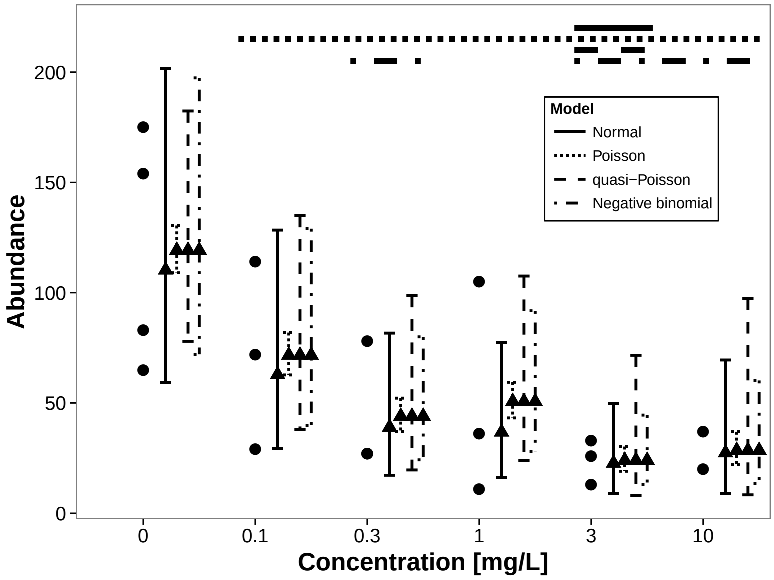


Figure 2.1.: Data from Brock et al. (2015) (dots). Predicted values (triangles) and 95% Wald Z or t confidence intervals from the fitted models (vertical lines) are given beside. Horizontal bars above indicate treatments statistically significant different from the control group (Dunnett contrasts). The data showed considerable overdispersion ($\kappa = 3.91, \phi = 22.41$) and therefore, the Poisson model underestimates the width of confidence intervals.

replicates (N = {3, 6, 9}) and different abundances in control treatments ($\mu_C$ = {2, 4, 8, 16, 32, 64, 128}). For Type I error estimation mean abundance was equal between treatments. For power estimation, mean abundance in treatments T2 - T5 was reduced to half of control and T1 ($\mu_{T_2} = ... = \mu_{T_5} = 0.5\ \mu_C = 0.5\ \mu_{T_1}$), resulting in a theoretical LOEC at T2. We generated 1000 data sets for each combination of N and $\mu_C$ and analysed these using the models outlined in section 2.3.1.

#### 2.3.5.2 *Binomial data*

We simulated data from a commonly used design as described in Weber et al. (1989), with 5 treated (T1 - T5) and one control group (C). Proportions were drawn from a Bin(10, $\pi$) distribution, with varying probability of survival ($\pi$ = {0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95}) and varying number of replicates (N = {3, 6, 9}). For Type I error estimation, $\pi$ was equal between treatments. For power estimation $\pi$ was fixed at 0.95 in C and T1 and varied only in treatments T2 - T5. For each combination we simulated 1000 data sets and analysed these using the models outlined in section 2.3.2.

### 2.3.6 *Data Analysis*

We analysed the case study and the simulated data using the outlined methods. We compared the methods and models in terms of Type I error (detection of an effect when there is none) and power (ability to detect an effect when it is present) at a significance level of $\alpha = 0.05$.

All simulations were done in R (Version 3.1.2) (R Core Team, 2014) on an Amazon EC2 virtual Linux server (64bit, 15GB RAM, 8 cores, 2.8 GHz). Source code to reproduce the simulations and paper is available online at `https://github.com/EDiLD/usetheglm`. Moreover, Supplement A.2 provides worked examples of the data of Brock et al. (2015) and Weber et al. (1989).

## 2.4 RESULTS

### 2.4.1 *Case study*

The data set showed considerably higher variance then expected by the Poisson model ($\phi = 22.41$ (eqn. 2.4), $\kappa = 3.91$ (eqn. 2.5)). Therefore, the Poisson model did not fit to this data and led to underestimated standard errors and confidence intervals, as well as overestimated statistical significance (Figure 2.1). In this case, inferences on the Poisson model are not valid and we do not further discuss its results. The normal (F = 2.57, p = 0.084) and quasi-Poisson model (F = 2.90, p = 0.061), as well as the Kruskal test (p = 0.145) did not show a statistically significant treatment effects. By contrast, the LR test and parametric bootstrap of the negative binomial model indicated a treatment-related effect (LR = 13.99, p = 0.016, bootstrap: p = 0.042).

All methods predicted similar values, except the normal model predicting always lower abundances (Figure 2.1). 95% confidence intervals (CI) were most narrow for the negative binomial model and widest for the quasi-Poisson model - especially at lower estimated abundances. Consequently, the LOECs differed (Normal and quasi-Poisson: 3 mg/L, negative binomial: 0.3 mg/L). The pairwise Wilcoxon test did not detect any treatment different from control.

### 2.4.2 *Simulations*

#### 2.4.2.1 *Count data*

For detecting a general treatment effect, $GLM_{nb}$ and $GLM_{p}$ showed inflated Type I error rates, whereas KW was conservative at low sample sizes. However, using the parametric bootstrap for the negative binomial model ($GLM_{npb}$), as well as LM and $GLM_{qp}$ resulted in appropriate Type I error rates. For detecting a treatment effect,$GLM_{qp}$ had the highest power, followed by $GLM_{npb}$, LM and KW, the latter having least power (Figure 2.2). For our simulation design (reduction in abundance by 50%) a sample size per treatment of n = 9 was needed to achieve a power greater than 80%. At small sample sizes (n = 3, 6) and low abundances ($\mu_C = 2, 4$) many of the negative binomial models ($GLM_{nb}$ and $GLM_{npb}$)

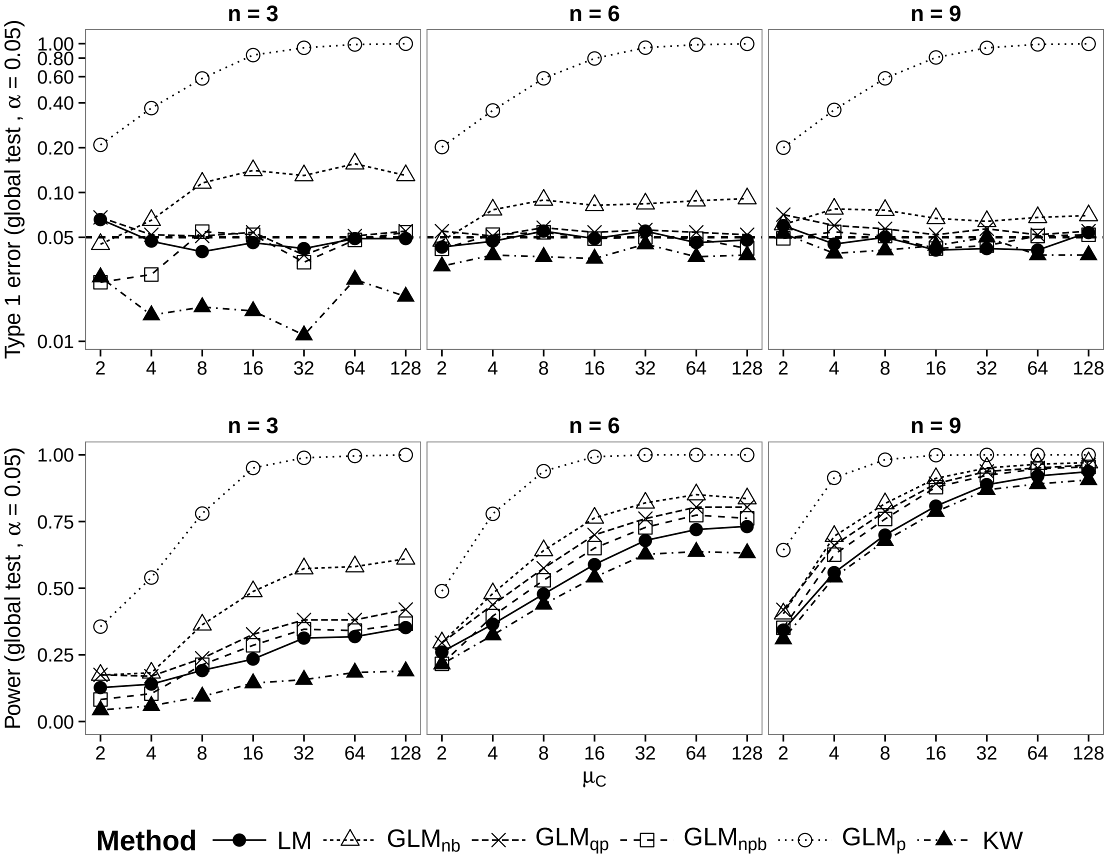


Figure 2.2.: Count data simulations: Type I error (top) and Power (bottom) for the test of a treatment effect. Type I errors are displayed on a logarithmic scale. Power levels for models with inflated Type I errors ($\mathrm{GLM_P}$ and $\mathrm{GLM_{qp}}$) are shown for completeness. For n = {3, 6} and $\mu_C$ = {2, 4} less than 85% of $\mathrm{GLM_{nb}}$ and $\mathrm{GLM_{npb}}$ models did converge. Dashed horizontal line denotes the nominal I error rate at $\alpha = 0.05$.

did not converge to a solution (convergence rate <85% of the simulations, Supplement A.1).

For LOEC determination $\mathrm{GLM_{nb}}$ and $\mathrm{GLM_p}$ showed an increased Type I error and all other methods were slightly conservative. The inferences on LOEC generally showed less power. LM showed a mean reduction of 20.7% and $\mathrm{GLM_{qp}}$ of 24.3 %. Power to detect the LOEC was highest for $\mathrm{GLM_{qp}}$. LM and *WT* showed less power, with *WT* having no power to detect the LOEC at low sample sizes (Figure 2.3).

### 2.4.2.2 *Binomial data*

$\mathrm{GLM_{bin}}$ showed slightly increased Type I error rates at low sample sizes and small effect sizes. *KW* was more conservative than LM and $\mathrm{GLM_{bin}}$. In addition, $\mathrm{GLM_{bin}}$ exhibited the greatest power for testing the treatment effect. This was especially apparent at low sample sizes (n = 3), with up to 27% higher power

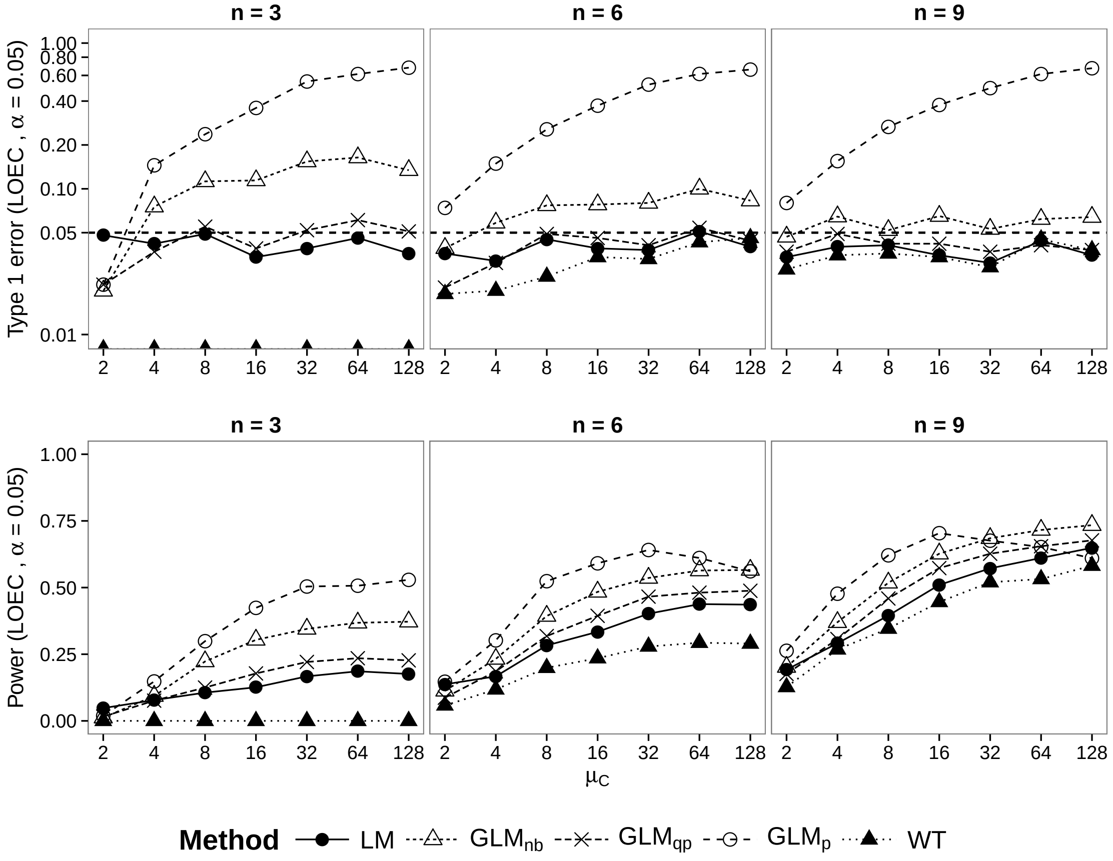


Figure 2.3.: Count data simulations: Type I error (top) and Power (bottom) for determination of LOEC. Type I errors are displayed one a logarithmic scale. Power levels for models with inflated Type I error are shown for completeness. For n = {3, 6} and $\mu_C$ = {2, 4} less than 85% of $GLM_{nb}$ models did converge. Dashed horizontal line denotes the nominal Type I error rate at $\alpha = 0.05$.

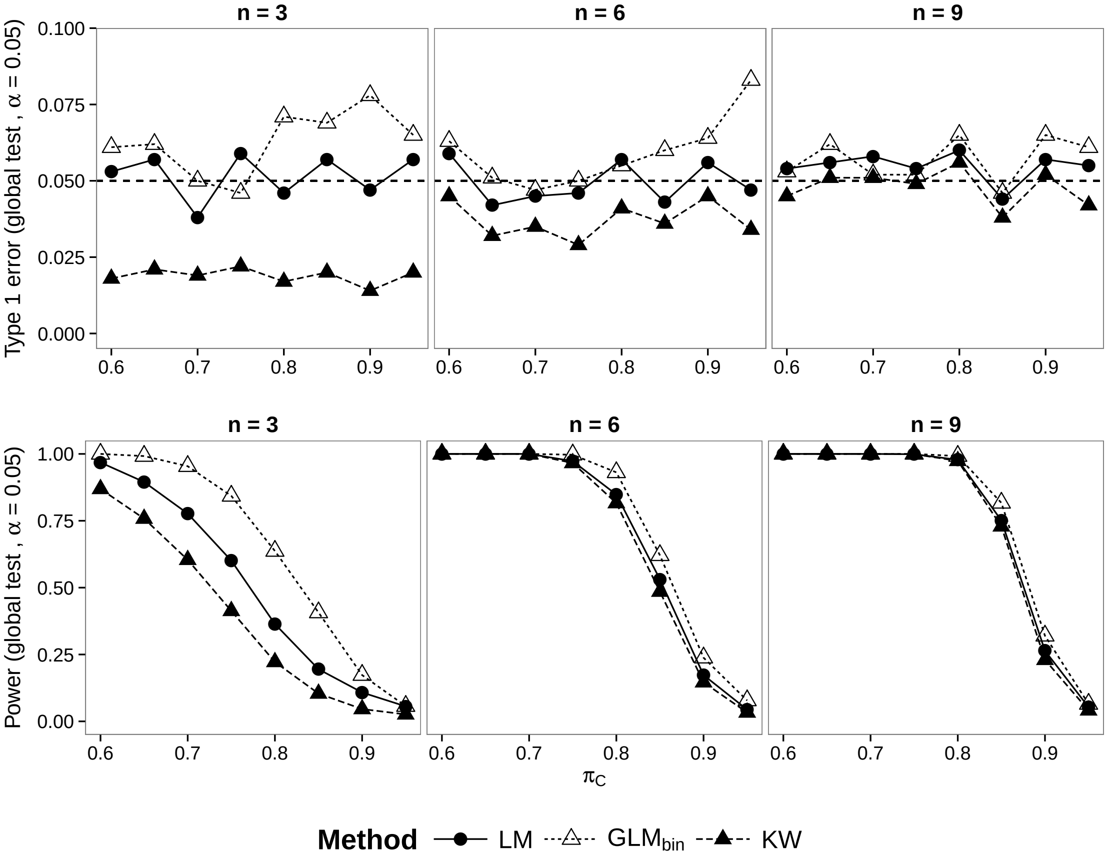


Figure 2.4.: Binomial data simulations: Type I error (top) and power (bottom) for the test of a treatment effect. Dashed horizontal line denotes the nominal Type I error rate at $\alpha = 0.05$.

compared to LM. However, the differences between methods quickly vanished with increasing samples sizes (Figure 2.4).

For inference on LOEC we found that all methods were slightly conservative. WT was generally more conservative and $GLM_{bin}$ especially at low effect sizes ($p_E > 0.7$). Inference on LOEC was not as powerful as inference on the general treatment effect. Contrary to the general treatment effect, LM showed the higher power than $GLM_{bin}$ at small sample sizes (n = 3, 6). WT had no power for n = 3 and showed less power in the other simulation runs (Figure 2.5).

## 2.5 DISCUSSION

### 2.5.1 *Case study*

The outlined case study demonstrates that the choice of the statistical model and procedure can have substantial impact on ecotoxicological inferences and endpoints like the LOEC. Therefore, ecotoxicologists should not base their inferences solely on statistical significance tests, but also on model estimates, their uncertainty and importance (Gelman and Stern, 2006). O'Hara and Kotze (2010)

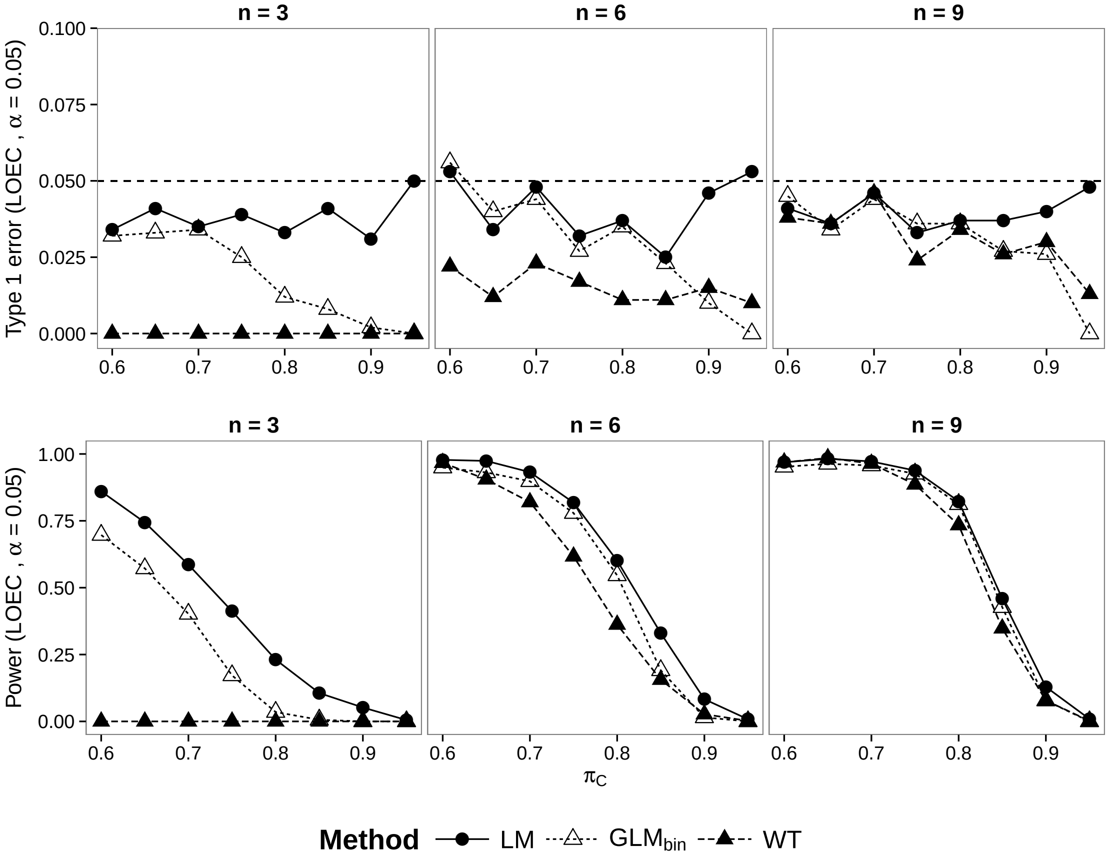


Figure 2.5.: Binomial data simulations: Type I error (top) and power (bottom) for the test for determination of LOEC. Dashed horizontal line denotes the nominal Type I error rate at $\alpha = 0.05$.

showed that the linear model on log transformed data gave unreliable and biased estimates, whereas GLMs performed well with little bias. Bias occurs also when back-transforming fitted means to the original scale, which explains the lower predicted means by LM in Figure 2.1 (Rothery, 1988) and should be corrected for (Michael C. Newman, 1993). When applied to non-transformed data, the linear model would predict identical treatment means as GLMs, because for a categorical predictor the predicted means of the LM and GLM are identical. When applied to non-transformed data, the linear model would result in identical predicted treatment means as GLMs. However, predictions would differ with continuous predictors and GLMs are particularly advantageous in this case.

This is further highlighted by the fact that for the same model (linear model applied to transformed data), Brock et al. (2015) reported a 10-fold lower LOEC (0.3 mg/L) then found in our study (3 mg/L, Figure 2.1). The reasons are manifold: (i) Brock et al. (2015) used a $\log(2\,y+1)$ transformation, whereas we used a $\log(A\,y+1)$ transformation, where A = 2 / 11 = 0.182 (Brink et al., 2000). (ii) We adjusted for multiple testing using Holm's (1979) method. (iii) Brock et al. (2015) used a one-sided Williams test (Williams, 1972), whereas we used one-sided comparisons to the control (Dunnett contrasts). The choice of transformation contributed only little to the differences. If the assumptions of Williams test are met it has strictly greater power than Dunnett contrasts (Jaki and L. A. Hothorn, 2013), which explains the differences in the case study. A generalisation of the Williams test as multiple contrast test (MCT) can be used in a GLM framework (T. Hothorn, Bretz, and Westfall, 2008). Nevertheless, such a Williams-type MCT is not a panacea (L. A. Hothorn, 2014) and our simulated semi-concave dose-response relationship is a situation where it fails and likely underestimates the LOEC (Kuiper, Gerhard, and L. A. Hothorn, 2014).

Overdispersion is common for ecological datasets (Warton, 2005) and the case study illustrates the potential effects of overdispersion that is not accounted for: standard errors will be underestimated and significance overestimated (Figures 2.1). This is also shown by our simulations (Figures 2.2, 2.3) where $\text{GLM}_p$ showed increased Type I error rates because of overdispersed simulated data. However, in factorial designs the mean-variance relationship can be easily checked by plotting mean versus variance of the treatment groups or by inspecting residual versus fitted values plots (see Supplement A.2). Our simulations revealed that the LR test for $\text{GLM}_{nb}$ is invalid because of increased Type I errors. This explains why it had the lowest p-value in the case study.

In the introduction we pointed out that there is little advice how to choose between the plenty of possible transformations - how do GLMs simplify this problem? The distribution modeled can be chosen using knowledge about the data (e.g. bounds, integer or continuous data etc). Knowing what type of data is modeled (see Methods section), the model selection process can be completely guided by the data and diagnostic tools. Therefore, choosing an appropriate model is easier than choosing between possible transformations.

### 2.5.2 *Simulations*

Our simulations showed that GLMs have generally greater power than the linear model applied to transformed data. However, the simulations also suggest that the power at the population level in common mesocosm experiments is low. For common samples sizes ($n \leqslant 4$ ) and a reduction in abundance of 50% we found a low power to detect any treatment-related effect (<50% for methods with appropriate Type I error, Figure 2.2). Statistical power to detect the correct LOEC was even lower (less than 25%), which can be attributed to multiple testing. The low power of all methods to detect significant treatment levels such as the LOEC or NOEC suggests that these endpoints from ecotoxicological studies should be interpreted with caution and underpins their criticism (Laskowski, 1995; Landis and Chapman, 2011).

Mesocosm studies allow also for inferences on the community level. For community analyses *GLM for multivariate data* (Warton, Wright, and Y. Wang, 2012) have been proposed as alternative to Principal Response Curves (PRC) and yielded similar inferences, but better indication of responsive taxa (Szöcs et al., 2015). However, Braak and Šmilauer (2015) argue to use data transformations with community data because of their simplicity and robustness. Although our simulations covered only simple experimental designs at the population level, findings may also extend to more complex situations. Nested or repeated designs with non-normal data could be analysed using Generalised Linear Mixed Models (GLMM) and may have advantages with respect to power (Stroup, 2015).

To counteract the problems with low power at the population level Brock et al. (2015) proposed to take the Minimum Detectable Difference (MDD), a method to assess statistical power *a posteriori*, for inference into account. However, *a priori* power analyses can be performed easily using simulations, even for complex experimental designs (Johnson et al., 2015), and might help to design, interpret and evaluate ecotoxicological studies. Moreover, Brock et al. (2015) proposed that statistical power of mesocosm experiments can be increased by reducing sampling variability through improved sampling techniques and quantification methods, though they also caution against depleting populations through more exhaustive sampling. As we showed, using GLMs can enhance the power at no extra costs.

M. Wang and Riffel (2011) advocated that in the typical case of small sample sizes (n <20) and non-normal data, non-parametric tests perform better than parametric tests assuming normality. In contrast, our results showed that the often applied *KW* and *WT* have less power compared to LM. Moreover, GLMs always performed better than non-parametric tests. Though more powerful non-parametric tests may be available (Konietschke, L. A. Hothorn, and Brunner, 2012), these are focused on hypothesis testing and do not provide estimation of effect sizes. Additionally to testing, GLMs allow the estimation and interpretation of effects that might not be statistically significant, but ecologically relevant. Therefore, we advise using GLMs instead of non-parametric tests for non-normal data.

We found an increased Type-I error for $GLM_{nb}$ at low sample sizes. However, it is well known that the LR statistic is not reliable at small sample sizes (Bolker et al., 2009; Wilks, 1938). Parametric bootstrap ($GLM_{npb}$) is a valuable alternative in such situations and maintains appropriate levels (Figure 2.2). Moreover, at small sample sizes and low abundances a significant amount of negative binomial models did not converge. We used an iterative algorithm to fit these models (Venables and Ripley, 2002) and other methods assessing the likelihood directly may perform better.

$GLM_{qp}$ showed higher statistical power than $GLM_{npb}$ (Figure 2.2, bottom). This could be explained by the simpler mean-variance relationship of $GLM_{qp}$ (eqn. 2.4 and 2.5), because at small samples sizes, low abundances or few treatment groups it is difficult to determine the mean-variance relationship. Our results are similar to Ives (2015), who compared GLMs to LM applied to transformed data for testing regression coefficients. Because of inflated Type I errors for $GLM_{nb}$ and, in the case of multiple explanatory variables in the model, inflated Type I errors of $GLM_{qp}$ he considered the LM on transformed data as most robust and recommended its preferred use. However, we showed that the parametric bootstrap LR test of $GLM_{nb}$ provides appropriate Type I errors and bootstrapping might be an alternative for testing coefficients. Nevertheless, bootstrapping is computationally very intensive and we found no gains in power compared to $GLM_{qp}$ (Figure 2.2). Given the higher power, appropriate Type I errors, stable convergence and reduced bias (O'Hara and Kotze, 2010) we suggest that count data in one factorial experiments should be analysed using the quasi-Poisson model.

Binomial data are often collected in lab trials, where increasing the sample size may be relatively easy to accomplish. We found notable differences in power to detect a treatment effect for all simulated sample sizes. Similarly, Warton and Hui (2011) also found that GLMs have higher power than arcsine transformed linear models. Though we did not simulate overdispersed binomial data, this should be checked and accounted for. In such situations a GLMM may offer an appealing alternative (Warton and Hui, 2011). At low effect sizes $GLM_{bin}$ became conservative with increasing $\pi_C$, although this effect lessened as sample size increased (Figure 2.5). This is because $\pi$ approaches its boundary and is also known as the *Hauck-Donner effect* (Hauck and Donner, 1977). A LR-Test or parametric bootstrap may provide an alternative in such situations (Bolker et al., 2009). This can also explain why LM performed better for deriving LOECs at low sample sizes.

GLMs can be fitted with several statistical software packages and many textbooks are available to introduce ecotoxicologists to these models (e.g. Zuur 2013 or Quinn and Keough 2009). We recommend that ecotoxicologists should change their models instead of their data. GLMs should become a standard method in ecotoxicology and incorporated into respective guidelines.

## 2.6 REFERENCES

Anderson, M. J., T. O. Crist, J. M. Chase, M. Vellend, B. D. Inouye, A. L. Freestone, N. J. Sanders, H. V. Cornell, L. S. Comita, K. F. Davies, S. P. Harrison, N. J. B. Kraft, J. C. Stegen, and N. G. Swenson (2011). "Navigating the multiple meanings of beta diversity: a roadmap for the practicing ecologist." In: *Ecology Letters* 14.1, pp. 19–28.

Bolker, B.M, M.E Brooks, C.J Clark, S.W Geange, J.R Poulsen, M.H.H Stevens, and J.S.S White (2009). "Generalized linear mixed models: a practical guide for ecology and evolution." In: *Trends in Ecology & Evolution* 24.3, pp. 127–135.

Braak, Cajo JF ter and Petr Šmilauer (2015). "Topics in constrained and unconstrained ordination." In: *Plant Ecology* 216.5, pp. 683–696.

Brink, P. J. van den, J. Hattink, T. C. M. Brock, F. Bransen, and E. van Donk (2000). "Impact of the fungicide carbendazim in freshwater microcosms. II. Zooplankton, primary producers and final conclusions." In: *Aquatic Toxicology* 48.2-3, pp. 251–264.

Brock, T. C. M., M. Hammers-Wirtz, U. Hommen, T. G. Preuss, H-T. Ratte, I. Roessink, T. Strauss, and P. J. Van den Brink (2015). "The minimum detectable difference (MDD) and the interpretation of treatment-related effects of pesticides in experimental ecosystems." In: *Environmental Science and Pollution Research* 22.2, pp. 1160–1174.

Dunnett, Charles W. (1955). "A Multiple Comparison Procedure for Comparing Several Treatments with a Control." In: *Journal of the American Statistical Association* 50.272, pp. 1096–1121.

EFSA PPR (2013). "Guidance on tiered risk assessment for plant protection products for aquatic organisms in edge-of-field surface waters." In: *EFSA Journal* 11.7, p. 3290.

EPA (2002). *Methods for Measuring the Acute Toxicity of Effluents and Receiving Waters to Freshwater and Marine Organisms*. U.S. Environmental Protection Agency.

Faraway, Julian James (2006). *Extending the linear model with R: Generalized linear, mixed effects and nonparametric regression models*. Boca Raton: Chapman & Hall.

Gelman, A. and H. Stern (2006). "The difference between "significant" and "not significant" is not itself statistically significant." In: *The American Statistician* 60.4, pp. 328–331.

Hauck, Walter W. and Allan Donner (1977). "Wald's Test as Applied to Hypotheses in Logit Analysis." In: *Journal of the American Statistical Association* 72.360, p. 851.

Hilbe, Joseph M. (2014). *Modeling Count Data*. New York, NY: Cambridge University Press.

Holm, Sture (1979). "A simple sequentially rejective multiple test procedure." In: *Scandinavian journal of statistics* 6.2, pp. 65–70.

Hothorn, Ludwig A. (2014). "Statistical evaluation of toxicological bioassays – a review." In: *Toxicol. Res.* 3.6, pp. 418–432.

Hothorn, T., F. Bretz, and P. Westfall (2008). "Simultaneous inference in general parametric models." In: *Biometrical Journal* 50.3, pp. 346–363.

Ives, Anthony R (2015). "For testing the significance of regression coefficients, go ahead and log-transform count data." In: *Methods in Ecology and Evolution* 6.7, pp. 828–835.

Jaki, Thomas and Ludwig A. Hothorn (2013). "Statistical evaluation of toxicological assays: Dunnett or Williams test—take both." In: *Archives of Toxicology* 87.11, pp. 1901–1910.

Johnson, Paul C. D., Sarah J. E. Barry, Heather M. Ferguson, and Pie Müller (2015). "Power analysis for generalized linear mixed models in ecology and evolution." In: *Methods in Ecology and Evolution* 6.2, pp. 133–142.

Konietschke, Frank, Ludwig A. Hothorn, and Edgar Brunner (2012). "Rank-based multiple test procedures and simultaneous confidence intervals." In: *Electronic Journal of Statistics* 6, pp. 738–759.

Kuiper, Rebecca M., Daniel Gerhard, and Ludwig A. Hothorn (2014). "Identification of the Minimum Effective Dose for Normally Distributed Endpoints Using a Model Selection Approach." In: *Statistics in Biopharmaceutical Research* 6.1, pp. 55–66.

Landis, Wayne G and Peter M Chapman (2011). "Well past time to stop using NOELs and LOELs." In: *Integrated Environmental Assessment and Management* 7.4, pp. vi–viii.

Laskowski, R. (1995). "Some good reasons to ban the use of NOEC, LOEC and related concepts in ecotoxicology." In: *Oikos* 73.1, pp. 140–144.

Nelder, J. A. and R. W. M. Wedderburn (1972). "Generalized Linear Models." In: *Journal of the Royal Statistical Society. Series A (General)* 135.3, pp. 370–384.

Newman, Michael C. (1993). "Regression analysis of log-transformed data: Statistical bias and its correction." In: *Environmental Toxicology and Chemistry* 12.6, pp. 1129–1133.

Newman, Michael C (2012). *Quantitative ecotoxicology*. Boca Raton, FL: Taylor & Francis.

OECD (2006). *Current Approaches in the Statistical Analysis of Ecotoxicity Data: A Guidance to Application*. Series on Testing and Assessment 54. Paris: OECD.

O'Hara, Robert B. and D. Johan Kotze (2010). "Do not log-transform count data." In: *Methods in Ecology and Evolution* 1.2, pp. 118–122.

Quinn, Gerry P. and Michael J. Keough (2009). *Experimental design and data analysis for biologists*. Cambridge: Cambridge Univ. Press.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. URL: http://www.R-project.org/.

Rothery, P. (1988). "A cautionary note on data transformation: bias in back-transformed means." In: *Bird Study* 35.3, pp. 219–221.

Sanderson, Hans (2002). "Pesticide studies." In: *Environmental Science and Pollution Research* 9.6, pp. 429–435.

Stroup, Walter W (2015). "Rethinking the analysis of non-normal data in plant and soil science." In: *Agronomy Journal* 107.2, pp. 811–827.

Szöcs, Eduard, 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, and Ralf B. Schäfer (2015).

"Analysing chemical-induced changes in macroinvertebrate communities in aquatic mesocosm experiments: a comparison of methods." In: *Ecotoxicology* 24.4, pp. 760–769.

Venables, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with S*. Fourth edition. New York: Springer.

Ver Hoef, Jay M. and Peter L. Boveng (2007). "Quasi-Poisson vs. negative binomial regression: how should we model overdispersed count data?" In: *Ecology* 88.11, pp. 2766–2772.

Wang, M. and M. Riffel (2011). "Making the right conclusions based on wrong results and small sample sizes: interpretation of statistical tests in ecotoxicology." In: *Ecotoxicology and Environmental Safety* 74.4, pp. 684–92.

Warton, David I. (2005). "Many zeros does not mean zero inflation: comparing the goodness-of-fit of parametric models to multivariate abundance data." In: *Environmetrics* 16.3, pp. 275–289.

Warton, David I. and Francis K. C. Hui (2011). "The arcsine is asinine: the analysis of proportions in ecology." In: *Ecology* 92.1, pp. 3–10.

Warton, David I., Stephen T. Wright, and Yi Wang (2012). "Distance-based multivariate analyses confound location and dispersion effects." In: *Methods in Ecology and Evolution* 3.1, pp. 89–101.

Weber, C. I., W. H. Peltier, T. J. Norbert-King, W. B. Horning, F.A. Kessler, J. R. Menkedick, T. W. Neiheisel, P. A. Lewis, D. J. Klemm, Q.H. Pickering, E. L. Robinson, J. M. Lazorchak, L.J. Wymer, and R. W. Freyberg (1989). "Short-term methods for estimating the chronic toxicity of effluents and receiving waters to fresh- water organisms." In: EPA/600/4–89/001.

Wilks, Samuel S. (1938). "The large-sample distribution of the likelihood ratio for testing composite hypotheses." In: *The Annals of Mathematical Statistics* 9.1, pp. 60–62.

Williams, D. A. (1972). "The comparison of several dose levels with a zero dose control." In: *Biometrics*, pp. 519–531.

Williams, D. A. (1982). "Extra-Binomial Variation in Logistic Linear Models." In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 31.2, pp. 144–148.

Zuur, Alain F (2013). *A beginner's guide to GLM and GLMM with R: a frequentist and Bayesian perspctive for ecologists*. Newburgh: Highland Statistics.

# 3

# WEBCHEM: AN R PACKAGE TO RETRIEVE CHEMICAL INFORMATION FROM THE WEB

Eduard Szöcs[a] & Ralf B. Schäfer[a]

[a]Institute for Environmental Sciences, University Koblenz-Landau, Landau, Germany

In Review in *Journal of Statistical Software*, 2016.

## 3.1 abstract

4

# TAXIZE: TAXONOMIC SEARCH AND RETRIEVAL IN R

Scott A. Chamberlain[a] & Eduard Szöcs[b]

[a]Biology Department, Simon Fraser University, Burnaby, BC, Canada,
[b]Institute for Environmental Sciences, University Koblenz-Landau, Landau, Germany

Adapted from the article published in 2013 in *F1000Research*, 2.191.

This chapter reflects the software state in 2013. In the meantime there have been many changes to taxize, so that not all parts presented here and thre respective supplementary materials work anymore. For a more recent description please visit the project homepage https://github.com/ropensci/taxize.

## 4.1 ABSTRACT

All species are hierarchically related to one another, and we use taxonomic names to label the nodes in this hierarchy. Taxonomic data is becoming increasingly available on the web, but scientists need a way to access it in a programmatic fashion that's simple and reproducible. We have developed taxize, an open-source software package for the R language (freely available from `http://cran.r-project.org/web/packages/taxize`). taxize provides simple, programmatic access to taxonomic data for 13 data sources around the web. We discuss the need for a taxonomic toolbelt in R, and outline a suite of use cases for which taxize is ideally suited (including a full workflow as an appendix). The taxize package facilitates open and reproducible science by allowing taxonomic data collection to be done in the open-source R platform.

## 4.2 INTRODUCTION

Evolution by natural selection has led to a hierarchical relationship among all living organisms. Thus, species are categorized using a taxonomic hierarchy, starting with the binomial species name (e.g, *Homo sapiens*), moving up to genus (*Homo*), then family (*Hominidae*), and on up to Domain (*Eukarya*). Although taxonomic classifications are human constructs created to understand the real phylogeny of life (Benton, 2000), they are nonetheless essential to organize the vast diversity of organisms. Biologists, whether studying organisms at the cell, organismal, or community level, can put their study objects into taxonomic context, allowing them to infer close and distant relatives, find relevant literature, and more.

The use of taxonomic names is, unfortunately, not straightforward. Taxonomic names often vary due to name revisions at the generic or specific levels, lumping or splitting lower taxa (genera, species) among higher taxa (families), and name spelling changes. For example, a study found that a compilation of 308,000 plant observations from 51 digitized herbarium records had 22,100 unique taxon names, of which only 13,000 were accepted names (Weiser et al., 2007; Boyle et al., 2013). In addition, there is no one authoritative source of taxonomic names for all taxa - although, there are taxon specific sources that are used by many scientists. Different sources (e.g., uBio [Universal Biological Indexer and Organizer], Tropicos, ITIS [Integrated Taxonomic Information Service]) may use different accepted names for the same taxon. For example, while ITIS has *Helianthus x glaucus* as an accepted name, The Plant List (`http://www.theplantlist.org`) gives that name as unresolved. But *Helianthus glaucus* is an accepted name in The Plant List, while ITIS does not list this name.

One attempt to help inconsistencies in taxonomy is the use of numeric codes. For example, ITIS assigns a Taxonomic Serial Number (TSN) to each taxon, while uBio assigns each taxon a NameBank identifier (namebankID), and Tropicos assigns their own identifier to each taxon. Codes are helpful within a database as they can easily refer to, for example, *Helianthus annuus* with a code like 123456 instead of its whole name. However, each database uses their own code; in this case for *Helianthus annuus*, ITIS uses 36616, uBio uses 2658020, and Tropicos uses 40022652. As there are no universal codes for taxa across databases, this can lead to additional confusion. Last, name comparisons across databases have to be done with the actual names, not the codes.

Taxonomic data is getting easier to obtain through the web (e.g., http://eol.org/). However, there are a number of good reasons to obtain taxonomic information programatically rather than through a web interface. First, if you have more than a few names to look up on a website, it can take quite a long time to enter each name, get data, and repeat for each species. Programatically getting taxonomic names solves the problem by looping over a list of names. In addition, doing taxonomic searching, etc. becomes reproducible. With increasing reports of irreproducibility in science (Stodden, 2010; Zimmer, 2012), it is extremely important to make science workflows repeatable.

The R language is widely used by biologists, and now has over 5,000 packages on the Comprehensive R Archive Network (CRAN) to extend R. R is great for manipulating, visualizing and fitting statistical models to data. Gentleman et al. Gentleman et al. (2004) give a detailed discussion of advantages of R in computational biology. Getting data from the web will be increasingly common as more and more data gets moved to the cloud. Therefore, there is a need to get data from the web directly into R. Increasingly, data is available from the web via application programming interfaces (API). These allow computers to talk to one another using code that is not human readable, but is machine readable. Web APIs often define a number of methods that allow users to search for a species name, or retrieve the synonyms for a species name, for example. A further advantage of APIs is that they are language agnostic, meaning that data can be consumed in almost any computing context, allowing users to interact with the web API without having to know the details of the code. Moreover data can be accessed from every computer, whereas for example an Excel file can only be opened in a few programs.

The goal of taxize is to make many use cases that involve retrieving and resolving taxonomic names easy and reproducible. In taxize, we have written a suite of R functions that interact with many taxonomic data sources via their web APIs (Table 4.1). The interface to each function is usually a simple list of species names, just as a user would enter when interacting with a website. Therefore, we hope that moving from a web to an R interface for taxonomic names will be relatively seamless (if one is already nominally familiar with R).

Here, we justify the need for programmatic taxonomic resolution tools like taxize, discuss our data sources, and run through a suite of use cases to demonstrate the variety of ways that users can use taxize.

Table 4.1.: Some key functions in taxize, what they do, and their data sources

| Function name | What it does | Source |
|---|---|---|
| apg_lookup() | Changes names to match the APGIII list | Angiosperm Phylogeny Group http://www.mobot.org/MOBOT/research/APweb/ |
| classification() | Upstream classification | Various |

Table 4.1 – *Cont.*

| Function name | What it does | Source |
|---|---|---|
| col_children() | Direct children | Catalogue of Life http://www.catalogueoflife.org/ |
| col_downstream() | Downstream taxa to specified rank | Catalogue of Life http://www.catalogueoflife.org/ |
| eol_hierarchy() | Upstream classification | Encyclopedia of Life http://eol.org/ |
| eol_search() | Search EOL taxon information | Encyclopedia of Life http://eol.org/ |
| get_seqs() | Get NCBI sequences | National Center for Biotechnology Information (Federhen, 2012) |
| get_tsn() | Get ITIS TSN | Integrated Taxonomic Information System http://www.itis.gov/ |
| get_uid() | Get NCBI UID | National Center for Biotechnology Information (Federhen, 2012) |
| gisd_isinvasive() | Invasiveness status | Global Invasive Species Database http://www.issg.org/database/welcome/ |
| gni_parse() | Parse scientific names into components | Global Names Index http://gni.globalnames.org/ |
| gni_search() | Search EOL's global names index | Global Names Index http://gni.globalnames.org/ |
| gnr_resolve() | Resolve names using EOL's global names index | Global Names Resolver http://resolver.globalnames.org/ |
| itis_downstream() | Downstream taxa to specified rank | Integrated Taxonomic Information System http://www.itis.gov/ |
| iucn_status() | IUCN status | IUCN Red List http://www.iucnredlist.org |
| phylomatic_tree() | Get a plant Phylogeny | Phylomatic (Webb and Donoghue, 2005) |
| plantminer() | Search Plantminer | Plantminer (Carvalho, Cianciaruso, and Batalha, 2010) |
| searchby-commonname() | Search ITIS by common name | Integrated Taxonomic Information System http://www.itis.gov/ |
| searchby-scientificname() | Search ITIS by scientific name | Integrated Taxonomic Information System http://www.itis.gov/ |
| tax_name() | Get taxonomic name for specific rank | Various |

Table 4.1 – *Cont.*

| Function name | What it does | Source |
|---|---|---|
| tax_rank() | Get rank of a taxonomic name | Various |
| tnrs() | Resolve names using iPlant | iPlant Taxonomic Name Resolution Service http://tnrs.iplantcollaborative.org/ |
| tp_acceptednames() | Check for accepted names using Tropicos | Tropicos http://www.tropicos.org/ |
| tpl_search() | Search the Plant List | The Plant List http://www.theplantlist.org |
| ubio_namebank() | Search uBio | uBio http://www.ubio.org/index.php?pagename=sample_tools |

## 4.3 WHY DO WE NEED TAXIZE?

There is a large suite of applications developed around the problem of searching for, resolving, and getting higher taxonomy for species names. For example, Linnaeus (http://linnaeus.sourceforge.net/) provides the ability to search for taxonomic names in documents and normalize those names found. In addition, there are many web interfaces to search for and normalize names such as Encyclopedia of Life's Global Names Resolver (http://resolver.globalnames.org/), uBio tools (www.ubio.org/index.php?pagename=sample_tools), and iPlant's Taxonomic Name Resolution Service (http://tnrs.iplantcollaborative.org/).

All of these data repositories provide ways to search for taxonomic names and resolve them in some cases. However, scientists ideally need a tool that is free and can be used programmatically, thereby facilitating reproducible research. The goal of taxize is to facilitate the creation of reproducible and easy to use workflows for searching for taxonomic names, resolving them, getting higher taxonomic names, and other tasks related to research dealing with species.

## 4.4 DATA SOURCES AND PACKAGE DETAILS

taxize uses many data sources (Table 4.1), and more can be easily added. There are two common tasks provided by the data sources: name search and name resolution. Other functionality in taxize includes retrieving a classification tree for a species, or retrieving child taxa of a focal taxon. O ne of the data sources (Phylomatic) returns phylogenies, while another (NCBI) returns genetic sequence data. However, there are other R packages that are focused solely on sequence data, such as rsnps (Chamberlain and Ushey, 2013), rentrez (Winter, 2013), BoSSA (Lefeuvre, 2010), and ape (Paradis, Claude, and Strimmer, 2004), so taxize does not venture deeply into these other domains.

Some of the data sources taxize interacts with require authentication. That is, in addition to the search terms the user provides (e.g., *Homo sapiens*), the data provider requires an alphanumeric identification key. This is necessary in some cases so that API providers can 1) better prevent databases crashing from too many requests, 2) collect analytics on requests to their API to provide better performance, etc., and 3) provide user level modification of rules for interacting with the API. The services that require an API key in taxize are: Encyclopedia of Life (EOL) (http://eol.org/), the Universal Biological Indexer and Organizer (uBio) (http://www.ubio.org/index.php?pagename=sample_tools), Tropicos (http://www.tropicos.org/), and Plantminer (Carvalho, Cianciaruso, and Batalha, 2010). One can easily obtain API keys by visiting the website of each service (see Table 4.1 for links to each site). There are two typical ways of using API keys. First, you can pass in your API key in a function call (e.g., *ubio_namebank(srchName='Ursus americanus', key='your_alphanumeric_key')*). Second, you can store your key in the .Rprofile file, which is a common place to store settings. We recommend the second option as it simplifies function calls as taxize detects the stored keys.

taxize would not have been possible without the work of others. taxize uses httr (Wickham, 2012a) and RCurl (Lang, 2013a) for performing calls to web APIs, XML (Lang, 2013c) for parsing XML, RJSONIO (Lang, 2013b) for parsing JSON, and stringr (Wickham, 2012b) and plyr (Wickham, 2011) for manipulating data.

New data sources can be added: for example, we plan to add the following sources: Wikispecies (https://species.wikimedia.org) and The Tree of Life (http://tolweb.org/tree/). A connection to www.freshwaterecology.info (a database with autecological characteristics, ecological preferences and biological traits as well as distribution patterns of more than 12,000 European freshwater organisms belonging to fish, macro-invertebrates, macrophytes, diatoms and phytoplankton) will be finished when their new API is released. In addition, the authors welcome further suggestions of data sources to be added.

## 4.5 USE CASES

### 4.5.1 *First, install taxize*

First, one must install and load taxize into the R session.

```r
install.packages("taxize")
library(taxize)
```

Advanced users can also download and install the latest development copy from GitHub https://github.com/ropensci/taxize, also permanently available at http://dx.doi.org/10.5281/zenodo.7097.

### 4.5.2 *Resolve taxonomic names*

This is a common task in biology. We often have a list of species names and we want to know a) if we have the most up to date names, b) if our names are spelled correctly, and c) the scientific name for a common name. One way to resolve names is via the Global Names Resolver (GNR) service provided by the Encyclopedia of Life (http://resolver.globalnames.org/). Here, on can search for two misspelled names:

```
temp <- gnr_resolve(names = c("Helianthos annus",
                              "Homo saapiens"))
temp[ , -c(1,4)]

#                  matched_name       data_source_title
# 1        Helianthus annuus L.       Catalogue of Life
# 2            Helianthus annus GBIF Taxonomic Backbone
# 3            Helianthus annus                     EOL
# 4         Helianthus annus L.                     EOL
# 5            Helianthus annus           uBio NameBank
# 6 Homo sapiens Linnaeus, 1758       Catalogue of Life
```

The correct spellings are *Helianthus annuus* and *Homo sapiens*. Another approach uses the Taxonomic Name Resolution Service via the Taxosaurus API (http://taxosaurus.org/) developed by iPLant and the Phylotastic organization. In this example is a list of species names, some of which are misspelled, and then call the API with the *tnrs* function.

```
mynames <- c("Helianthus annuus", "Pinus contort",
             "Poa anua", "Abis magnifica", "Rosa california",
             "Festuca arundinace", "Sorbus occidentalos",
             "Madia sateva")
tnrs(query = mynames)[ , -c(5:7)]

#          submittedName        acceptedName    sourceId score
# 9    Helianthus annuus   Helianthus annuus iPlant_TNRS     1
# 10   Helianthus annuus   Helianthus annuus        NCBI     1
# 4        Pinus contort      Pinus contorta iPlant_TNRS  0.98
# 5             Poa anua           Poa annua iPlant_TNRS  0.96
# 3       Abis magnifica     Abies magnifica iPlant_TNRS  0.96
# 7      Rosa california    Rosa californica iPlant_TNRS  0.99
# 8      Rosa california          California        NCBI     1
# 2   Festuca arundinace Festuca arundinacea iPlant_TNRS  0.99
# 1  Sorbus occidentalos Sorbus occidentalis iPlant_TNRS  0.99
# 6         Madia sateva        Madia sativa iPlant_TNRS  0.97
```

It turns out there are a few corrections: e.g., *Madia sateva* should be *Madia sativa*, and *Rosa california* should be *Rosa californica*. Note that this search worked because fuzzy matching was employed to retrieve names that were close, but not exact matches. Fuzzy matching is only available for plants in the TNRS service, so we advise using EOL's Global Names Resolver if you need to resolve animal names.

taxize takes the approach that the user should be able to make decisions about what resource to trust, rather than making the decision on behalf of the user. Both the EOL GNR and the TNRS services provide data from a variety of data sources. The user may trust a specific data source, and thus may want to use the names from that data source. In the future, we may provide the ability for taxize to suggest the best match from a variety of sources.

Another common use case is when there are many synonyms for a species. In this example, there are six synonyms of the currently accepted name for a species.

```r
library(plyr)
mynames <- c("Helianthus annuus ssp. jaegeri",
             "Helianthus annuus ssp. lenticularis",
             "Helianthus annuus ssp. texanus",
             "Helianthus annuus var. lenticularis",
             "Helianthus annuus var. macrocarpus",
             "Helianthus annuus var. texanus")
tsn <- get_tsn(mynames)
ldply(tsn, itis_acceptname)

#   submittedTsn      acceptedName acceptedTsn
# 1       525928 Helianthus annuus       36616
# 2       525929 Helianthus annuus       36616
# 3       525930 Helianthus annuus       36616
# 4       536095 Helianthus annuus       36616
# 5       536096 Helianthus annuus       36616
# 6       536097 Helianthus annuus       36616
```

### 4.5.3 *Retrieve higher taxonomic names*

Another task biologists often face is getting higher taxonomic names for a taxa list. Having the higher taxonomy allows you to put into context the relationships of your species list. For example, you may find out that species A and species B are in Family C, which may lead to some interesting insight, as opposed to not knowing that Species A and B are closely related. This also makes it easy to aggregate/standardize data to a specific taxonomic level (e.g., family level) or to match data to other databases with different taxonomic resolution (e.g., trait databases).

A number of data sources in taxize provide the capability to retrieve higher taxonomic names, but we will highlight two of the more useful ones: Integrated Taxonomic Information System (ITIS) (http://www.itis.gov/) and National Center for Biotechnology Information (NCBI) (Federhen, 2012). First, search for two species, *Abies procera* and *Pinus contorta* within ITIS.

```
specieslist <- c("Abies procera", "Pinus contorta")
classification(specieslist, db = "itis")

# $‘Abies procera‘
#          rankName       taxonName    tsn
# 1         Kingdom         Plantae 202422
# 2      Subkingdom  Viridaeplantae 846492
# 3    Infrakingdom    Streptophyta 846494
# 4        Division    Tracheophyta 846496
# 5     Subdivision Spermatophytina 846504
# 6   Infradivision    Gymnospermae 846506
# 7           Class       Pinopsida 500009
# 8           Order         Pinales 500028
# 9          Family        Pinaceae  18030
# 10          Genus           Abies  18031
# 11        Species   Abies procera 181835
#
# $‘Pinus contorta‘
#          rankName       taxonName    tsn
# 1         Kingdom         Plantae 202422
# 2      Subkingdom  Viridaeplantae 846492
# 3    Infrakingdom    Streptophyta 846494
# 4        Division    Tracheophyta 846496
# 5     Subdivision Spermatophytina 846504
# 6   Infradivision    Gymnospermae 846506
# 7           Class       Pinopsida 500009
# 8           Order         Pinales 500028
# 9          Family        Pinaceae  18030
# 10          Genus           Pinus  18035
# 11        Species  Pinus contorta 183327
```

It turns out both species are in the family Pinaceae. You can also get this type of information from the NCBI by excuting the following code in R: *classification(specieslist, db = 'ncbi')*.

Instead of a full classification, you may only want a single name, say a family name for your species of interest. The function *tax_name* is built just for this purpose. As with the *classification*-function you can specify the data source with the *db* argument, either ITIS or NCBI.

```
tax_name(query = "Helianthus annuus", get = "family",
         db = "itis")

#       family
# 1 Asteraceae

tax_name(query = "Helianthus annuus", get = "family",
         db = "ncbi")

#       family
# 1 Asteraceae
```

If a data source does not provide information on the queried species, the result could be taken from another source and the results from the different sources could be pooled.

### 4.5.4 *Interactive name selection*

As mentioned previously most databases use a numeric code to reference a species. A general workflow in taxize is: Retrieve Code for the queried species and then use this code to query more data/information. Below are a few examples. When you run these examples in R, you are presented with a command prompt asking for the row that contains the name you would like back; that output is not printed below for brevity. In this example, the search term has many matches. The function returns a data.frame of the matches, and asks for the user to input which row number to accept.

```
get_tsn(searchterm = "Heliastes", searchtype = "sciname")

#            combinedname    tsn
# 1     Heliastes bicolor 615238
# 2   Heliastes chrysurus 615250
# 3     Heliastes cinctus 615573
# 4  Heliastes dimidiatus 615257
# 5  Heliastes hypsilepis 615273
# 6 Heliastes immaculatus 615639
# 7 Heliastes opercularis 615300
# 8      Heliastes ovalis 615301
#  1
# NA
# attr(,"class")
# [1] "tsn"
```

In another example, you can pass in a long character vector of taxonomic names:

```
splist <- c("annona cherimola", 'annona muricata',
            "quercus robur", "shorea robusta",
            "pandanus patina", "oryza sativa",
            "durio zibethinus")
get_tsn(searchterm = splist, searchtype = "sciname")

# [1] "506198" "18098"  "19405"  "506787" "507376" "41976"
# [7] "506099"
# attr(,"class")
# [1] "tsn"
```

In another example, note that no match at all returns an NA:

```
get_uid(sciname = c("Chironomus riparius", "aaa vva"))

# [1] "315576" NA
# attr(,"class")
# [1] "uid"
```

### 4.5.5 *Retrieve a phylogeny*

Ecologists are increasingly taking a phylogenetic approach to ecology, applying phylogenies to topics such as the study of community structure (Webb, Ackerly, et al., 2002), ecological networks (Rafferty and Ives, 2013), functional trait ecology (Poff et al., 2006). Yet, Many biologists are not adequately trained in reconstructing phylogenies. Fortunately, there are some sources for getting a phylogeny without having to know how to build one; one of these is for angiosperms, called Phylomatic (Webb and Donoghue, 2005). We have created a workflow in taxize that accepts a species list, and taxize works behind the scenes to get higher taxonomic names, which are required by Phylomatic to get a phylogeny. Here is a short example, producing the tree in figure 4.1.

```
taxa <- c("Poa annua", "Abies procera", "Helianthus annuus")
tree <- phylomatic_tree(taxa = taxa)
tree$tip.label <- capwords(tree$tip.label)
plot(tree, cex = 1)
```

Behind the scenes the function *phylomatic_tree* retrieves a Taxonomic Serial Number (TSN) from ITIS for each species name, then a string is created for each species like this *poaceae/oryza/oryzasativa* (with format "family/genus/genus_epithet"). These strings are submitted to the Phylomatic API, and if no errors occur, a phylogeny in newick format is returned. The *phylomatic_tree()* function also cleans up the newick string and converts it to an ape *phylo* object, which can be used for plotting and phylogenetic analyses. Be aware that Phylomatic has certain limitations - refer to the paper describing Phylomatic (Webb and Donoghue, 2005) and the website `http://phylodiversity.net/phylomatic/`.

### 4.5.6 *What taxa are children of the taxon of interest?*

If someone is not a taxonomic specialist on a particular taxon they probably do not know what children taxa are within a family, or within a genus. This task becomes especially unwieldy when there are a large number of taxa downstream. You can of course go to a website like Wikispecies (`http://species.wikimedia.org`) or Encyclopedia of Life (`http://eol.org/`) to get downstream names. However, taxize provides an easy way to programatically search for downstream taxa, both for the Catalogue of Life (CoL) (`http://www.catalogueoflife.org/`) and the Integrated Taxonomic Information System (`http://www.itis.gov/`). Here is a short example using the CoL in which we want to find all the species within the genus *Apis* (honey bees).

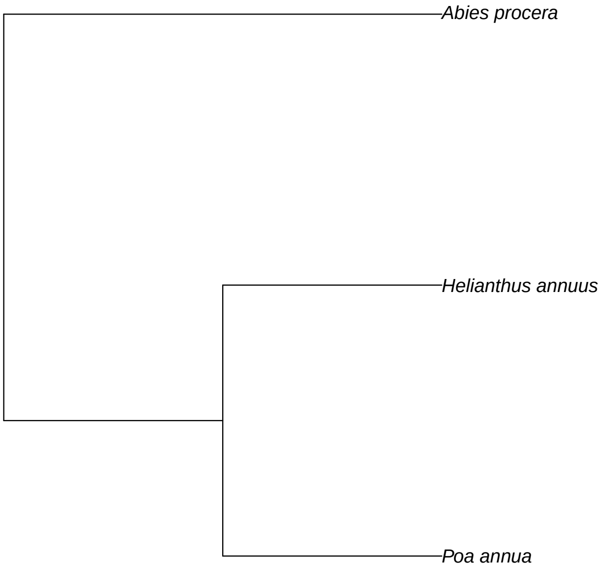


Figure 4.1.: A phylogeny for three species. This phylogeny was produced using the *phylomatic_tree* function, which queries the Phylomatic database, and prunes a previously created phylogeny of plants.

```
col_downstream(name = "Apis", downto = "Species")[[1]]

#   childtaxa_id     childtaxa_name childtaxa_rank
# 1      6971712 Apis andreniformis        Species
# 2      6971713        Apis cerana        Species
# 3      6971714       Apis dorsata        Species
# 4      6971715        Apis florea        Species
# 5      6971716 Apis koschevnikovi        Species
# 6      6845885     Apis mellifera        Species
# 7      6971717   Apis nigrocincta        Species
```

The result from the above call to *col_downstream()* is a data.frame that gives a number of columns of different information.

### 4.5.7 *IUCN Status*

There are a number of things a user can do once they have the correct taxonomic names. One thing a user can do is ask about the conservation status of a species (IUCN Red List of Threatened Species (`http://www.iucnredlist.org`)). We have provided a set of functions, *iucn_summary* and *iucn_status*, to search for species names, and extract the status information, respectively. Here, you can search for the panther and lynx.

```
ia <- iucn_summary(c("Panthera uncia", "Lynx lynx"))
iucn_status(ia)

# Panthera uncia      Lynx lynx
#           "EN"           "LC"
```

It turns out that the panther has a status of endangered (EN) and the lynx has a status of least concern (LC).

### 4.5.8 *Search for available genes in GenBank*

Another use case available in taxize deals with genetic sequences. taxize has three functions to interact with GenBank to search for available genes (*get_genes_avail*), download genes by GenBank ID (*get_genes*), and download genes via taxonomic name search, including retrieving a congeneric if the searched taxon does not exist in the database (*get_seqs*). In this example, one can search for gene sequences for *Umbra limi*.

```
out <- get_genes_avail(taxon_name = "Umbra limi",
                       seqrange = "1:2000",
                       getrelated = FALSE)
```

Then one can ask if 'RAG1' exists in any of the gene names.

```
out[grep("RAG1", out$genesavail, ignore.case = TRUE), -3]

#          spused length access_num       ids
# 413 Umbra limi    732   JX190826 394772608
# 427 Umbra limi    959   AY459526  45479841
# 434 Umbra limi   1631   AY380548  38858304
```

It turns out that there are 430 different unique records found. However, this doesn't mean that there are 430 different genes found as the API does not provide metadata to classify genes. You can use regular expressions (e.g., *grep*) to search for the gene of interest.

### 4.5.9 *Matching species tables with different taxonomic resolution*

Biologists often need to match different sets of data tied to species. For example, trait-based approaches are a promising tool in ecology (Statzner and Bêche, 2010). One problem is that abundance data must be matched with trait databases such as the NCBI Taxonomy database (Usseglio-Polatera et al., 2000). These two data tables may contain species information on different taxonomic levels and data might have to be aggregated to a joint taxonomic level, so that the data can be merged. taxize can help in this data-cleaning step, providing a reproducible workflow.

A user can use the mentioned *classification*-function to retrieve the taxonomic hierarchy and then search the hierarchies up- and downwards for matches. Here is an example to match a species (A) with names of on different taxonomic levels (B1 & B2).

```
A <- "gammarus roeseli"
B1 <- "gammarus"
B2 <- "gammaridae"
A_clas <- classification(A, db = ’ncbi’)
B1_clas <- classification(B1, db = ’ncbi’)
B2_clas <- classification(B2, db = ’ncbi’)
A_clas[[1]]$Rank[tolower(A_clas[[1]]$ScientificName) %in% B1]

# [1] "genus"

A_clas[[1]]$Rank[tolower(A_clas[[1]]$ScientificName) %in% B2]

# [1] "family"
```

If one finds a direct match (here *Gammarus roeseli*), they will be lucky. However, Gammaridae can also be matched with *Gammarus roeseli*, but on a lower taxonomic level. A more comprehensive and realistic example (matching a trait table with an abundance table) is given in the supplemental materials.

### 4.5.10 *Aggregating data to a specific taxonomic rank*

In biology, one can ask questions at varying taxonomic levels. This use case is easily handled in taxize. A function called *tax_agg()* will aggregate community data to a specific taxonomic level. In this example, one can take the data for three species and aggregate them to family level. Again one can specify whether they want to use data from ITIS or NCBI. The rows in the *data.frame* are different communities.

```
data(dune, package = ’vegan’)
df <- dune[ , c(1,3:4)]
colnames(df) <- c("Bellis perennis", "Juncus bufonius",
                  "Juncus articulatus")
head(df)

#    Bellis perennis Juncus bufonius Juncus articulatus
# 2                3               0                  0
# 13               0               3                  0
# 4                2               0                  0
# 16               0               0                  3
# 6                0               0                  0
# 1                0               0                  0
```

footnotesize

```
agg <- tax_agg(df, rank = 'family', db = 'ncbi')
agg

#
#  Aggregated community data
#
# Level of Aggregation: FAMILY
# No. taxa before aggregation: 3
# No. taxa after aggregation: 2
# No. taxa not found: 0
```

```
head(agg$x)

#    Asteraceae Juncaceae
# 2           3         0
# 13          0         3
# 4           2         0
# 16          0         3
# 6           0         0
# 1           0         0
```

The two *Juncus* species are aggregated to the family Juncaceae and their abundances are summed. There was only a single species in the family Asteraceae, so the data for *Bellis perennis* are carried over.

## 4.6 conclusions

Taxonomic information is increasingly sought by biologists as we take phylogenetic and taxonomic approaches to science. Taxonomic data are becoming more widely available on the web, yet scientists require programmatic access to this data for developing reproducible workflows. taxize was created to bridge this gap - to bring taxonomic data on the web into R, where the data can be easily manipulated, visualized, and analyzed in a reproducible workflow.

We have outlined a suite of use cases in taxize that will likely fit real use cases for many biologists. Of course we have not thought of all possible use cases, so we hope that the biology community can give us feedback on what use cases they want to see available in taxize. One thing we could change in the future is to make functions that fit use cases, and then allow users to select the data source as a parameter in the function. This could possibly make the user interface easier to understand.

taxize is currently under development and will be for some time given the large number of data sources knitted together in the package, and the fact that APIs for each data source can change, requiring changes in taxize code. Contributions to taxize are strongly encouraged, and can be easily done using GitHub here: https://github.com/ropensci/taxize. We hope taxize will be taken up by the community and developed collaboratively, making it progressively better through time as new use cases arise, bug reports are squashed, and contributions are merged.

## 4.7 REFERENCES

Benton, Michael J. (2000). "Stems, nodes, crown clades, and rank-free lists: is Linnaeus dead?" In: *Biological Reviews* 75.4, pp. 633–648.

Boyle, Brad, Nicole Hopkins, Zhenyuan Lu, Juan Antonio Raygoza Garay, Dmitry Mozzherin, Tony Rees, Naim Matasci, Martha L Narro, William H Piel, Sheldon J Mckay, and et al. (2013). "The taxonomic name resolution service: an online tool for automated standardization of plant names." In: *BMC Bioinformatics* 14.1, p. 1.

Carvalho, Gustavo Henrique, Marcus Vinicius Cianciaruso, and Marco Antônio Batalha (2010). "Plantminer: a web tool for checking and gathering plant species taxonomic information." In: *Environmental Modelling & Software* 25.6, pp. 815–816.

Chamberlain, Scott and Kevin Ushey (2013). *rsnps: Interface to SNP data on the web.* R package version 0.0.4. URL: https://github.com/ropensci/rsnps.

Federhen, Scott (2012). "The NCBI Taxonomy database." In: *Nucleic Acids Research* 40.D1, pp. D136–D143.

Gentleman, Robert C, Vincent J Carey, Douglas M Bates, Ben Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis, Laurent Gautier, Yongchao Ge, Jeff Gentry, et al. (2004). "Bioconductor: open software development for computational biology and bioinformatics." In: *Genome biology* 5.10, p. 1.

Lang, Duncan Temple (2013a). *RCurl: General network (HTTP/FTP/...) client interface for R*. R package version 1.95-4.1. URL: http://CRAN.R-project.org/package=RCurl.

Lang, Duncan Temple (2013b). *RJSONIO: Serialize R objects to JSON, JavaScript Object Notation*. R package version 1.0-3. URL: http://CRAN.R-project.org/package=RJSONIO.

Lang, Duncan Temple (2013c). *XML: Tools for parsing and generating XML within R and S-Plus.* R package version 3.95-0.2. URL: http://CRAN.R-project.org/package=XML.

Lefeuvre, Pierre (2010). *BoSSA: a Bunch of Structure and Sequence Analysis*. R package version 1.2. URL: http://CRAN.R-project.org/package=BoSSA.

Paradis, E., J. Claude, and K. Strimmer (2004). "APE: analyses of phylogenetics and evolution in R language." In: *Bioinformatics* 20, pp. 289–290.

Poff, N LeRoy, Julian D Olden, Nicole KM Vieira, Debra S Finn, Mark P Simmons, and Boris C Kondratieff (2006). "Functional trait niches of North American lotic insects: traits-based ecological applications in light of phylogenetic relationships." In: *Journal of the North American Benthological Society* 25.4, pp. 730–755.

Rafferty, Nicole E and Anthony R Ives (2013). "Phylogenetic trait-based analyses of ecological networks." In: *Ecology* 94.10, pp. 2321–2333.

Statzner, B. and L.A Bêche (2010). "Can biological invertebrate traits resolve effects of multiple stressors on running water ecosystems?" In: *Freshwater Biology* 55, pp. 80–119.

Stodden, Victoria C (2010). "Reproducible research: Addressing the need for data and code sharing in computational science." In: *Computing in Science & Engineering* 12.5, pp. 8–12.

Usseglio-Polatera, Philippe, Michel Bournaud, Philippe Richoux, and Henri Tachet (2000). "Biological and ecological traits of benthic freshwater macroinvertebrates: relationships and definition of groups with similar traits." In: *Freshwater Biology* 43.2, pp. 175–205.

Webb, Campbell O, David D Ackerly, Mark A McPeek, and Michael J Donoghue (2002). "Phylogenies and community ecology." In: *Annual Review of Ecology and Systematics*, pp. 475–505.

Webb, Campbell O and Michael J Donoghue (2005). "Phylomatic: tree assembly for applied phylogenetics." In: *Molecular Ecology Notes* 5.1, pp. 181–183.

Weiser, Michael D., Brian J. Enquist, Brad Boyle, Timothy J. Killeen, Peter M. Jørgensen, Gustavo Fonseca, Michael D. Jennings, Andrew J. Kerkhoff, Thomas E. Lacher Jr, Abel Monteagudo, and et al. (2007). "Latitudinal patterns of range size and species richness of New World woody plants." In: *Global Ecology and Biogeography* 16.5, pp. 679–688.

Wickham, Hadley (2011). "The Split-Apply-Combine Strategy for Data Analysis." In: *Journal of Statistical Software* 40.1, pp. 1–29.

Wickham, Hadley (2012a). *httr: Tools for working with URLs and HTTP*. R package version 0.2. URL: http://CRAN.R-project.org/package=httr.

Wickham, Hadley (2012b). *stringr: Make it easier to work with strings*. R package version 0.6.2. URL: http://CRAN.R-project.org/package=stringr.

Winter, David (2013). *rentrez: Entrez in R*. R package version 0.2.1. URL: https://github.com/ropensci/rentrez.

Zimmer, Carl (2012). "A Sharp Rise in Retractions Prompts Calls for Reform." In: *New York Times*. URL: http://www.nytimes.com/2012/04/17/science/rise-in-scientific-journal-retractions-prompts-calls-for-reform.html?_r=0.

# 5

# DISCUSSION

## 5.1 STATISTICAL ECOTOXICOLOGY

## 5.2 LEVERAGING MONITORING DATA FOR ECOLOGICAL RISK ASSESSMENT

## 5.3 OPEN DATA AND REPRODUCIBILITY IN ECOTOXICOLOGY

## 5.4 CONCLUSIONS AND OUTLOOK

# A

# SUPPLEMENTAL MATERIAL FOR: ECOTOXICOLOGY IS NOT NORMAL - A COMPARISON OF STATISTICAL APPROACHES FOR ANALYSIS OF COUNT AND PROPORTION DATA IN ECOTOXICOLOGY

## A.1 SUPPLEMENTARY TABLES

Table A.1.: Count data simulations - Proportion of models converged. N = sample sizes, $\mu_C$ = mean abundance in control, LM = Linear model after transformation, $GLM_{nb}$ = negative binomial model, $GLM_{qp}$ = quasi-Poisson model, $GLM_p$ = Poisson model

| N | $\mu_C$ | LM | $GLM_{nb}$ | $GLM_{qp}$ | $GLM_p$ |
|---|---|---|---|---|---|
| 3.00 | 2.00 | 1.00 | 0.33 | 1.00 | 1.00 |
| 3.00 | 4.00 | 1.00 | 0.53 | 1.00 | 1.00 |
| 3.00 | 8.00 | 1.00 | 0.79 | 1.00 | 1.00 |
| 3.00 | 16.00 | 1.00 | 0.94 | 1.00 | 1.00 |
| 3.00 | 32.00 | 1.00 | 0.99 | 1.00 | 1.00 |
| 3.00 | 64.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 3.00 | 128.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 2.00 | 1.00 | 0.63 | 1.00 | 1.00 |
| 6.00 | 4.00 | 1.00 | 0.85 | 1.00 | 1.00 |
| 6.00 | 8.00 | 1.00 | 0.98 | 1.00 | 1.00 |
| 6.00 | 16.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 32.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 64.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 128.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 9.00 | 2.00 | 1.00 | 0.76 | 1.00 | 1.00 |
| 9.00 | 4.00 | 1.00 | 0.95 | 1.00 | 1.00 |
| 9.00 | 8.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 9.00 | 16.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 9.00 | 32.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 9.00 | 64.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 9.00 | 128.00 | 1.00 | 1.00 | 1.00 | 1.00 |

Table A.2.: Count data simulations - Power to detect a treatment effect. N = sample sizes, $\mu_C$ = mean abundance in control, LM = Linear model after transformation, $\text{GLM}_{nb}$ = negative binomial model, $\text{GLM}_{qp}$ = quasi-Poisson model, $\text{GLM}_{qp}$ = Poisson model, np = pairwise Wilcoxon test.

| N | $\mu_C$ | LM | $\text{GLM}_{nb}$ | $\text{GLM}_{qp}$ | $\text{GLM}_{p}$ | np | NA |
|---|---|---|---|---|---|---|---|
| 3.00 | 2.00 | 0.13 | 0.17 | 0.17 | 0.08 | 0.36 | 0.04 |
| 3.00 | 4.00 | 0.14 | 0.18 | 0.17 | 0.10 | 0.54 | 0.06 |
| 3.00 | 8.00 | 0.19 | 0.36 | 0.24 | 0.21 | 0.78 | 0.09 |
| 3.00 | 16.00 | 0.23 | 0.49 | 0.33 | 0.29 | 0.95 | 0.14 |
| 3.00 | 32.00 | 0.31 | 0.57 | 0.38 | 0.35 | 0.99 | 0.16 |
| 3.00 | 64.00 | 0.32 | 0.58 | 0.38 | 0.34 | 1.00 | 0.18 |
| 3.00 | 128.00 | 0.35 | 0.61 | 0.42 | 0.37 | 1.00 | 0.19 |
| 6.00 | 2.00 | 0.26 | 0.30 | 0.29 | 0.22 | 0.49 | 0.21 |
| 6.00 | 4.00 | 0.36 | 0.48 | 0.44 | 0.40 | 0.78 | 0.32 |
| 6.00 | 8.00 | 0.48 | 0.64 | 0.57 | 0.53 | 0.94 | 0.44 |
| 6.00 | 16.00 | 0.59 | 0.76 | 0.70 | 0.65 | 0.99 | 0.54 |
| 6.00 | 32.00 | 0.68 | 0.82 | 0.76 | 0.73 | 1.00 | 0.63 |
| 6.00 | 64.00 | 0.72 | 0.85 | 0.80 | 0.77 | 1.00 | 0.64 |
| 6.00 | 128.00 | 0.73 | 0.84 | 0.80 | 0.76 | 1.00 | 0.63 |
| 9.00 | 2.00 | 0.34 | 0.40 | 0.42 | 0.35 | 0.64 | 0.31 |
| 9.00 | 4.00 | 0.56 | 0.69 | 0.66 | 0.63 | 0.91 | 0.54 |
| 9.00 | 8.00 | 0.70 | 0.82 | 0.79 | 0.76 | 0.98 | 0.68 |
| 9.00 | 16.00 | 0.81 | 0.91 | 0.89 | 0.88 | 1.00 | 0.79 |
| 9.00 | 32.00 | 0.89 | 0.95 | 0.94 | 0.92 | 1.00 | 0.87 |
| 9.00 | 64.00 | 0.92 | 0.96 | 0.95 | 0.95 | 1.00 | 0.89 |
| 9.00 | 128.00 | 0.94 | 0.97 | 0.96 | 0.95 | 1.00 | 0.91 |

Table A.3.: Count data simulations - Power to detect LOEC. N = sample sizes, $\mu_C$ = mean abundance in control, LM = Linear model after transformation, $GLM_{nb}$ = negative binomial model, $GLM_{qp}$ = quasi-Poisson model, $GLM_p$ = Poisson model, np = pairwise Wilcoxon test.

| N | $\mu_C$ | LM | $GLM_{nb}$ | $GLM_{qp}$ | $GLM_p$ | np |
|---|---|---|---|---|---|---|
| 3.00 | 2.00 | 0.05 | 0.01 | 0.02 | 0.02 | 0.00 |
| 3.00 | 4.00 | 0.08 | 0.09 | 0.08 | 0.15 | 0.00 |
| 3.00 | 8.00 | 0.11 | 0.22 | 0.12 | 0.30 | 0.00 |
| 3.00 | 16.00 | 0.13 | 0.30 | 0.18 | 0.42 | 0.00 |
| 3.00 | 32.00 | 0.17 | 0.35 | 0.22 | 0.50 | 0.00 |
| 3.00 | 64.00 | 0.19 | 0.37 | 0.23 | 0.51 | 0.00 |
| 3.00 | 128.00 | 0.18 | 0.37 | 0.23 | 0.53 | 0.00 |
| 6.00 | 2.00 | 0.14 | 0.11 | 0.09 | 0.15 | 0.06 |
| 6.00 | 4.00 | 0.17 | 0.23 | 0.19 | 0.30 | 0.12 |
| 6.00 | 8.00 | 0.28 | 0.39 | 0.32 | 0.52 | 0.20 |
| 6.00 | 16.00 | 0.33 | 0.48 | 0.39 | 0.59 | 0.23 |
| 6.00 | 32.00 | 0.40 | 0.54 | 0.47 | 0.64 | 0.28 |
| 6.00 | 64.00 | 0.44 | 0.56 | 0.48 | 0.61 | 0.29 |
| 6.00 | 128.00 | 0.44 | 0.57 | 0.49 | 0.56 | 0.29 |
| 9.00 | 2.00 | 0.19 | 0.20 | 0.18 | 0.26 | 0.13 |
| 9.00 | 4.00 | 0.29 | 0.37 | 0.31 | 0.48 | 0.27 |
| 9.00 | 8.00 | 0.40 | 0.52 | 0.46 | 0.62 | 0.35 |
| 9.00 | 16.00 | 0.51 | 0.63 | 0.57 | 0.70 | 0.45 |
| 9.00 | 32.00 | 0.57 | 0.69 | 0.63 | 0.68 | 0.52 |
| 9.00 | 64.00 | 0.61 | 0.72 | 0.66 | 0.65 | 0.53 |
| 9.00 | 128.00 | 0.65 | 0.73 | 0.68 | 0.61 | 0.58 |

Table A.4.: Count data simulations - Type 1 error to detect a global treatment effect. N = sample sizes, $\mu_C$ = mean abundance in control, LM = Linear model after transformation, $GLM_{nb}$ = negative binomial model, $GLM_{qp}$ = quasi-Poisson model, $GLM_{pb}$ = negative binomial model with parametric boostrap, $GLM_p$ = Poisson model, np = Kruskal-Wallis test.

| N | $\mu_C$ | LM | $GLM_{nb}$ | $GLM_{qp}$ | $GLM_{pb}$ | $GLM_p$ | np |
|---|---|---|---|---|---|---|---|
| 3.00 | 2.00 | 0.07 | 0.04 | 0.02 | 0.07 | 0.21 | 0.03 |
| 3.00 | 4.00 | 0.05 | 0.07 | 0.03 | 0.05 | 0.37 | 0.01 |
| 3.00 | 8.00 | 0.04 | 0.12 | 0.05 | 0.05 | 0.58 | 0.02 |
| 3.00 | 16.00 | 0.05 | 0.14 | 0.05 | 0.05 | 0.84 | 0.02 |
| 3.00 | 32.00 | 0.04 | 0.13 | 0.03 | 0.04 | 0.94 | 0.01 |
| 3.00 | 64.00 | 0.05 | 0.16 | 0.05 | 0.05 | 0.99 | 0.03 |
| 3.00 | 128.00 | 0.05 | 0.13 | 0.05 | 0.06 | 1.00 | 0.02 |
| 6.00 | 2.00 | 0.04 | 0.05 | 0.04 | 0.06 | 0.20 | 0.03 |
| 6.00 | 4.00 | 0.05 | 0.08 | 0.05 | 0.05 | 0.36 | 0.04 |
| 6.00 | 8.00 | 0.06 | 0.09 | 0.05 | 0.06 | 0.58 | 0.04 |
| 6.00 | 16.00 | 0.05 | 0.08 | 0.05 | 0.05 | 0.80 | 0.04 |
| 6.00 | 32.00 | 0.06 | 0.08 | 0.05 | 0.06 | 0.94 | 0.04 |
| 6.00 | 64.00 | 0.05 | 0.09 | 0.05 | 0.05 | 0.98 | 0.04 |
| 6.00 | 128.00 | 0.05 | 0.09 | 0.04 | 0.05 | 1.00 | 0.04 |
| 9.00 | 2.00 | 0.06 | 0.06 | 0.05 | 0.07 | 0.20 | 0.05 |
| 9.00 | 4.00 | 0.04 | 0.08 | 0.05 | 0.06 | 0.36 | 0.04 |
| 9.00 | 8.00 | 0.05 | 0.08 | 0.05 | 0.06 | 0.58 | 0.04 |
| 9.00 | 16.00 | 0.04 | 0.07 | 0.04 | 0.05 | 0.81 | 0.04 |
| 9.00 | 32.00 | 0.04 | 0.06 | 0.04 | 0.06 | 0.94 | 0.05 |
| 9.00 | 64.00 | 0.04 | 0.07 | 0.05 | 0.05 | 0.99 | 0.04 |
| 9.00 | 128.00 | 0.05 | 0.07 | 0.05 | 0.06 | 1.00 | 0.04 |

Table A.5.: Count data simulations - Type 1 error to detect LOEC. N = sample sizes, $\mu_C$ = mean abundance in control, LM = Linear model after transformation, $GLM_{nb}$ = negative binomial model, $GLM_{qp}$ = quasi-Poisson model, $GLM_p$ = Poisson model, np = pairwise Wilcoxon.

| N | $\mu_C$ | LM | $GLM_{nb}$ | $GLM_{qp}$ | $GLM_p$ | np |
|---|---|---|---|---|---|---|
| 3.00 | 2.00 | 0.05 | 0.02 | 0.02 | 0.02 | 0.00 |
| 3.00 | 4.00 | 0.04 | 0.08 | 0.04 | 0.14 | 0.00 |
| 3.00 | 8.00 | 0.05 | 0.11 | 0.06 | 0.24 | 0.00 |
| 3.00 | 16.00 | 0.03 | 0.11 | 0.04 | 0.36 | 0.00 |
| 3.00 | 32.00 | 0.04 | 0.15 | 0.05 | 0.55 | 0.00 |
| 3.00 | 64.00 | 0.05 | 0.16 | 0.06 | 0.61 | 0.00 |
| 3.00 | 128.00 | 0.04 | 0.13 | 0.05 | 0.68 | 0.00 |
| 6.00 | 2.00 | 0.04 | 0.04 | 0.02 | 0.07 | 0.02 |
| 6.00 | 4.00 | 0.03 | 0.06 | 0.03 | 0.15 | 0.02 |
| 6.00 | 8.00 | 0.04 | 0.08 | 0.05 | 0.26 | 0.03 |
| 6.00 | 16.00 | 0.04 | 0.08 | 0.05 | 0.37 | 0.03 |
| 6.00 | 32.00 | 0.04 | 0.08 | 0.04 | 0.52 | 0.03 |
| 6.00 | 64.00 | 0.05 | 0.10 | 0.05 | 0.61 | 0.04 |
| 6.00 | 128.00 | 0.04 | 0.08 | 0.04 | 0.66 | 0.05 |
| 9.00 | 2.00 | 0.03 | 0.05 | 0.04 | 0.08 | 0.03 |
| 9.00 | 4.00 | 0.04 | 0.06 | 0.05 | 0.15 | 0.04 |
| 9.00 | 8.00 | 0.04 | 0.05 | 0.04 | 0.27 | 0.04 |
| 9.00 | 16.00 | 0.04 | 0.07 | 0.04 | 0.38 | 0.03 |
| 9.00 | 32.00 | 0.03 | 0.05 | 0.04 | 0.49 | 0.03 |
| 9.00 | 64.00 | 0.04 | 0.06 | 0.04 | 0.61 | 0.04 |
| 9.00 | 128.00 | 0.04 | 0.06 | 0.04 | 0.67 | 0.04 |

Table A.6.: Binomial data simulations - Power to detect a global treatment effect. N = sample sizes, $p_E$ = probability in effect treatments, LM = Linear model after transformation, GLM = binomial model, np = Kruskal-Wallis test.

| N | $p_E$ | LM | GLM | np |
|---|---|---|---|---|
| 3.00 | 0.60 | 0.97 | 1.00 | 0.87 |
| 3.00 | 0.65 | 0.90 | 0.99 | 0.76 |
| 3.00 | 0.70 | 0.78 | 0.95 | 0.60 |
| 3.00 | 0.75 | 0.60 | 0.84 | 0.41 |
| 3.00 | 0.80 | 0.36 | 0.64 | 0.22 |
| 3.00 | 0.85 | 0.20 | 0.41 | 0.10 |
| 3.00 | 0.90 | 0.11 | 0.17 | 0.05 |
| 3.00 | 0.95 | 0.06 | 0.06 | 0.03 |
| 6.00 | 0.60 | 1.00 | 1.00 | 1.00 |
| 6.00 | 0.65 | 1.00 | 1.00 | 1.00 |
| 6.00 | 0.70 | 1.00 | 1.00 | 1.00 |
| 6.00 | 0.75 | 0.97 | 1.00 | 0.97 |
| 6.00 | 0.80 | 0.85 | 0.93 | 0.82 |
| 6.00 | 0.85 | 0.53 | 0.62 | 0.48 |
| 6.00 | 0.90 | 0.17 | 0.24 | 0.15 |
| 6.00 | 0.95 | 0.04 | 0.08 | 0.03 |
| 9.00 | 0.60 | 1.00 | 1.00 | 1.00 |
| 9.00 | 0.65 | 1.00 | 1.00 | 1.00 |
| 9.00 | 0.70 | 1.00 | 1.00 | 1.00 |
| 9.00 | 0.75 | 1.00 | 1.00 | 1.00 |
| 9.00 | 0.80 | 0.98 | 0.99 | 0.97 |
| 9.00 | 0.85 | 0.75 | 0.82 | 0.73 |
| 9.00 | 0.90 | 0.26 | 0.32 | 0.23 |
| 9.00 | 0.95 | 0.05 | 0.07 | 0.04 |

Table A.7.: Count data simulations - Power to detect LOEC. N = sample sizes, $p_E$ = probability in effect treatments, LM = Linear model after transformation, GLM = binomial model, np = pairwise Wilcoxon.

| N | $p_E$ | LM | GLM | np |
|---|---|---|---|---|
| 3.00 | 0.60 | 0.86 | 0.70 | 0.00 |
| 3.00 | 0.65 | 0.74 | 0.57 | 0.00 |
| 3.00 | 0.70 | 0.59 | 0.40 | 0.00 |
| 3.00 | 0.75 | 0.41 | 0.17 | 0.00 |
| 3.00 | 0.80 | 0.23 | 0.04 | 0.00 |
| 3.00 | 0.85 | 0.11 | 0.01 | 0.00 |
| 3.00 | 0.90 | 0.05 | 0.00 | 0.00 |
| 3.00 | 0.95 | 0.01 | 0.00 | 0.00 |
| 6.00 | 0.60 | 0.98 | 0.95 | 0.97 |
| 6.00 | 0.65 | 0.97 | 0.93 | 0.91 |
| 6.00 | 0.70 | 0.93 | 0.90 | 0.82 |
| 6.00 | 0.75 | 0.82 | 0.78 | 0.62 |
| 6.00 | 0.80 | 0.60 | 0.55 | 0.36 |
| 6.00 | 0.85 | 0.33 | 0.19 | 0.16 |
| 6.00 | 0.90 | 0.08 | 0.01 | 0.03 |
| 6.00 | 0.95 | 0.01 | 0.00 | 0.00 |
| 9.00 | 0.60 | 0.97 | 0.95 | 0.97 |
| 9.00 | 0.65 | 0.98 | 0.96 | 0.98 |
| 9.00 | 0.70 | 0.97 | 0.96 | 0.96 |
| 9.00 | 0.75 | 0.94 | 0.93 | 0.89 |
| 9.00 | 0.80 | 0.82 | 0.81 | 0.73 |
| 9.00 | 0.85 | 0.46 | 0.43 | 0.35 |
| 9.00 | 0.90 | 0.13 | 0.08 | 0.08 |
| 9.00 | 0.95 | 0.01 | 0.00 | 0.00 |

Table A.8.: Binomial data simulations - Type 1 error to detect a global treatment effect. N = sample sizes, p = probability, LM = Linear model after transformation, GLM = binomial model, np = Kruskal-Wallis test.

| N | p | LM | GLM | np |
|---|---|---|---|---|
| 3.00 | 0.60 | 0.05 | 0.06 | 0.02 |
| 3.00 | 0.65 | 0.06 | 0.06 | 0.02 |
| 3.00 | 0.70 | 0.04 | 0.05 | 0.02 |
| 3.00 | 0.75 | 0.06 | 0.05 | 0.02 |
| 3.00 | 0.80 | 0.05 | 0.07 | 0.02 |
| 3.00 | 0.85 | 0.06 | 0.07 | 0.02 |
| 3.00 | 0.90 | 0.05 | 0.08 | 0.01 |
| 3.00 | 0.95 | 0.06 | 0.07 | 0.02 |
| 6.00 | 0.60 | 0.06 | 0.06 | 0.04 |
| 6.00 | 0.65 | 0.04 | 0.05 | 0.03 |
| 6.00 | 0.70 | 0.04 | 0.05 | 0.04 |
| 6.00 | 0.75 | 0.05 | 0.05 | 0.03 |
| 6.00 | 0.80 | 0.06 | 0.06 | 0.04 |
| 6.00 | 0.85 | 0.04 | 0.06 | 0.04 |
| 6.00 | 0.90 | 0.06 | 0.06 | 0.04 |
| 6.00 | 0.95 | 0.05 | 0.08 | 0.03 |
| 9.00 | 0.60 | 0.05 | 0.05 | 0.04 |
| 9.00 | 0.65 | 0.06 | 0.06 | 0.05 |
| 9.00 | 0.70 | 0.06 | 0.05 | 0.05 |
| 9.00 | 0.75 | 0.05 | 0.05 | 0.05 |
| 9.00 | 0.80 | 0.06 | 0.07 | 0.06 |
| 9.00 | 0.85 | 0.04 | 0.05 | 0.04 |
| 9.00 | 0.90 | 0.06 | 0.07 | 0.05 |
| 9.00 | 0.95 | 0.06 | 0.06 | 0.04 |

Table A.9.: Binomial data simulations - Type 1 error to detect LOEC. N = sample sizes, p = probability, LM = Linear model after transformation, GLM = binomial model, np = pairwise Wilcoxon.

| N | $p_E$ | LM | GLM | np |
|---|---|---|---|---|
| 3.00 | 0.60 | 0.03 | 0.03 | 0.00 |
| 3.00 | 0.65 | 0.04 | 0.03 | 0.00 |
| 3.00 | 0.70 | 0.04 | 0.03 | 0.00 |
| 3.00 | 0.75 | 0.04 | 0.03 | 0.00 |
| 3.00 | 0.80 | 0.03 | 0.01 | 0.00 |
| 3.00 | 0.85 | 0.04 | 0.01 | 0.00 |
| 3.00 | 0.90 | 0.03 | 0.00 | 0.00 |
| 3.00 | 0.95 | 0.05 | 0.00 | 0.00 |
| 6.00 | 0.60 | 0.05 | 0.06 | 0.02 |
| 6.00 | 0.65 | 0.03 | 0.04 | 0.01 |
| 6.00 | 0.70 | 0.05 | 0.04 | 0.02 |
| 6.00 | 0.75 | 0.03 | 0.03 | 0.02 |
| 6.00 | 0.80 | 0.04 | 0.04 | 0.01 |
| 6.00 | 0.85 | 0.03 | 0.02 | 0.01 |
| 6.00 | 0.90 | 0.05 | 0.01 | 0.01 |
| 6.00 | 0.95 | 0.05 | 0.00 | 0.01 |
| 9.00 | 0.60 | 0.04 | 0.04 | 0.04 |
| 9.00 | 0.65 | 0.04 | 0.03 | 0.04 |
| 9.00 | 0.70 | 0.05 | 0.04 | 0.05 |
| 9.00 | 0.75 | 0.03 | 0.04 | 0.02 |
| 9.00 | 0.80 | 0.04 | 0.04 | 0.03 |
| 9.00 | 0.85 | 0.04 | 0.03 | 0.03 |
| 9.00 | 0.90 | 0.04 | 0.03 | 0.03 |
| 9.00 | 0.95 | 0.05 | 0.00 | 0.01 |

## A.2 WORKED R EXAMPLES

### A.2.1 *Count data example*

#### A.2.1.1 *Introduction*

In this example we will analyse data from (Brock et al., 2015). The data are count of mayfly larvae in Macroinvertebrate Artificial Substrate Samplers in 18 mesocosms at one sampling day. There are 5 treatments and one control group.

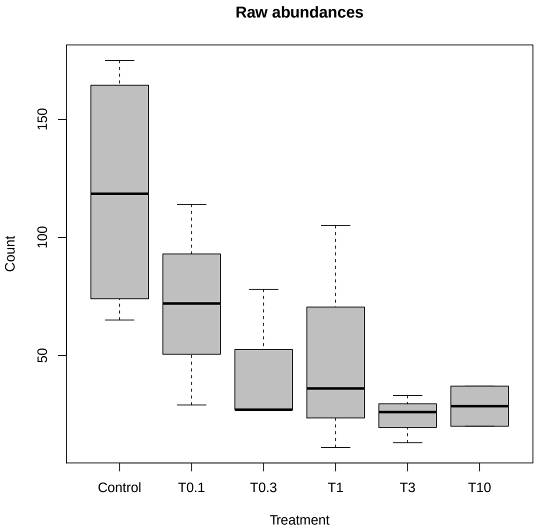
First, we load the data, bring it to the long format and remove NA values.

```
df <- read.table(header = TRUE,
                 text = ’Control  T0.1 T0.3  T1  T3  T10
                         175 29   27   36   26   20
                         65  114 78   11   13   37
                         154 72   27   105 33   NA
                         83  NA   NA   NA   NA   NA’)
require(reshape2)
dfm <- melt(df, value.name = ’abu’, variable.name = ’treatment’)
dfm <- dfm[!is.na(dfm[’abu’]), ]
head(dfm)

##   treatment abu
## 1   Control 175
## 2   Control  65
## 3   Control 154
## 4   Control  83
## 5      T0.1  29
## 6      T0.1 114
```

This results in a table with two columns - one indicating the treatment and one with the measured abundances.

Let’s have a first look at the data:

```
boxplot(abu ~ treatment, data = dfm, xlab = ’Treatment’,
        ylab = ’Count’, col = ’grey75’, main = ’Raw abundances’)
```



We clearly see a treatment related response. Moreover, we may note that variances are increasing with increasing abundances.

### A.2.1.2 *Assuming a normal distribution of transformed abundances*

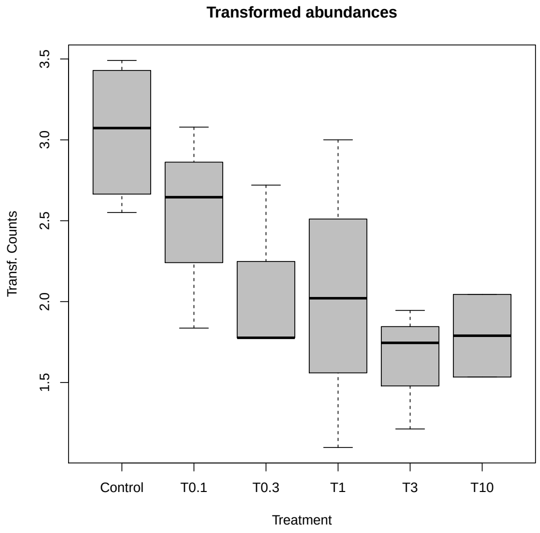
### A.2.1.3 *Data transformation*

Next we transform the data using a ln(Ax + 1) transformation. A is chosen so that the term Ax equals two for the lowest non-zero abundance. We add these transformed abundances as extra column to our table.

```
A <- 2 / min(dfm$abu[dfm$abu != 0])
A

## [1] 0.1818182

dfm$abu_t <- log(A * dfm$abu + 1)
head(dfm)

##   treatment abu    abu_t
## 1   Control 175 3.490983
## 2   Control  65 2.550865
## 3   Control 154 3.367296
## 4   Control  83 2.778254
## 5      T0.1  29 1.836211
## 6      T0.1 114 3.078568
```

It looks like the transformation does a good job in equalizing the variances:

```
boxplot(abu_t ~ treatment, data = dfm,
        xlab = 'Treatment', ylab = 'Transf. Counts',
        col = 'grey75', main = 'Transformed abundances')
```



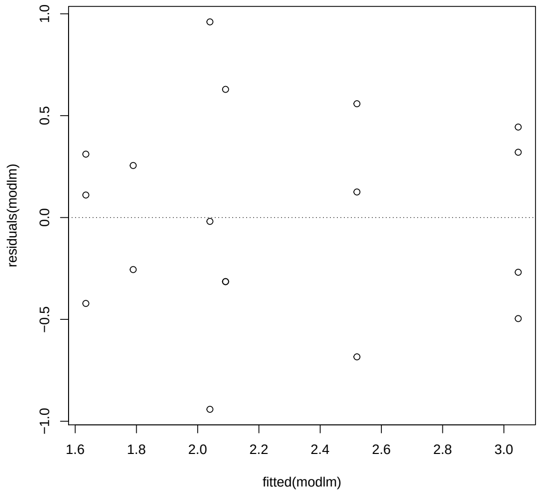
### A.2.1.4 *Model fitting*

The model from eqn. 2 can be easily fitted using the `lm()` function:

```
modlm <- lm(abu_t ~ treatment, data = dfm)
```

The residuals vs. fitted values diagnostic plot show no problematic pattern, though it might be difficult to decide with such a small sample size

```
plot(residuals(modlm) ~ fitted(modlm))
abline(h = 0, lty = 'dotted')
```



The `summary()` gives the estimated coefficients with standard errors and Wald t tests:

```
summary(modlm)


##
```

```
## Call:
## lm(formula = abu_t ~ treatment, data = dfm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.94133 -0.31454  0.04576  0.31813  0.96033
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.0468     0.2970  10.260 2.71e-07 ***
## treatmentT0.1  -0.5267     0.4536  -1.161  0.26814
## treatmentT0.3  -0.9558     0.4536  -2.107  0.05682 .
## treatmentT1    -1.0069     0.4536  -2.220  0.04646 *
## treatmentT3    -1.4121     0.4536  -3.113  0.00897 **
## treatmentT10   -1.2575     0.5144  -2.445  0.03089 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5939 on 12 degrees of freedom
## Multiple R-squared:  0.5167,Adjusted R-squared:  0.3154
## F-statistic: 2.566 on 5 and 12 DF,  p-value: 0.08406
```

### A.2.1.5 *Inference on general treatment effect*

Or, if you want to have the ANOVA table with an F-test:

```
summary.aov(modlm)

##             Df Sum Sq Mean Sq F value Pr(>F)
## treatment    5  4.526  0.9052   2.566 0.0841 .
## Residuals   12  4.233  0.3528
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From this output we might infer that we cannot detect any treatment effect (F = 2.566, p = 0.084).

### A.2.1.6 *Inference on LOEC*

Let's move on to the LOEC determination. This can be easily done using the multcomp package (T. Hothorn, F. Bretz, and P. Westfall, 2008):

Here we perform a one-sided (`alternative = 'less'`) using Dunnett contrasts of treatment (`mcp(treatment='Dunnett')`). Moreover, we adjust for multiple testing using Holm's method (`test = adjusted('holm')`):

```
require(multcomp)
summary(glht(modlm, linfct = mcp(treatment = 'Dunnett'),
alternative = 'less'),
        test = adjusted('holm'))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Dunnett Contrasts
##
##
## Fit: lm(formula = abu_t ~ treatment, data = dfm)
##
## Linear Hypotheses:
##                      Estimate Std. Error t value Pr(<t)
## T0.1 - Control >= 0  -0.5267     0.4536  -1.161 0.1341
## T0.3 - Control >= 0  -0.9558     0.4536  -2.107 0.0697 .
## T1 - Control >= 0    -1.0069     0.4536  -2.220 0.0697 .
## T3 - Control >= 0    -1.4121     0.4536  -3.113 0.0224 *
## T10 - Control >= 0   -1.2575     0.5144  -2.445 0.0618 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## (Adjusted p values reported -- holm method)
```

Here only treatment 3 mg/L shows a statistically significant difference from control and is the determined LOEC. The column `'Estimate'` gives the estimated difference in means between treatments and control and `'Std. Error'` the standard errors of these estimates.

To determine the LOEC we could also use a Williams type contrast (Frank Bretz, Torsten Hothorn, and P. H. Westfall, 2010).

Here I use a step-up Williams contrast. First we need to define a contrast matrix (see also `?contrMat()`):

```
# observations per treatment
n <- tapply(dfm$abu_t, dfm$treatment, length)
k <- length(n)
CM <- c()
for (i in 1:(k - 1)) {
  help <- c(-1, n[2:(i + 1)] / sum(n[2:(i + 1)]), rep(0 , k - i - 1))
  CM <- rbind(CM, help)
}
rownames(CM) <- paste("C", 1:nrow(CM))
CM

##             T0.1
## C 1 -1 1.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## C 2 -1 0.5000000 0.5000000 0.0000000 0.0000000 0.0000000
## C 3 -1 0.3333333 0.3333333 0.3333333 0.0000000 0.0000000
## C 4 -1 0.2500000 0.2500000 0.2500000 0.2500000 0.0000000
## C 5 -1 0.2142857 0.2142857 0.2142857 0.2142857 0.1428571
```

Then we supply this contrast matrix to `glht()`:

```
summary(glht(modlm, linfct = mcp(treatment = CM),
             alternative = ’less’),
        test = adjusted(’holm’))

##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: User-defined Contrasts
##
##
## Fit: lm(formula = abu_t ~ treatment, data = dfm)
##
## Linear Hypotheses:
##          Estimate Std. Error t value Pr(<t)
## C 1 >= 0  -0.5267     0.4536  -1.161 0.1341
## C 2 >= 0  -0.7413     0.3834  -1.934 0.0771 .
## C 3 >= 0  -0.8298     0.3569  -2.325 0.0576 .
## C 4 >= 0  -0.9754     0.3429  -2.845 0.0295 ∗
## C 5 >= 0  -1.0157     0.3367  -3.016 0.0268 ∗
## ---
## Signif. codes:  0 ’∗∗∗’ 0.001 ’∗∗’ 0.01 ’∗’ 0.05 ’.’ 0.1 ’ ’ 1
## (Adjusted p values reported -- holm method)
```

This indicates a LOEC at 3 mg/L.

If we do not adjust for multiple testing (`test = adjusted(’none’)`), we end up with the same NOEC (0.1 mg/L) as Brock et al. (2015):

```
summary(glht(modlm, linfct = mcp(treatment = CM),
             alternative = ’less’),
        test = adjusted(’none’))

##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: User-defined Contrasts
##
##
## Fit: lm(formula = abu_t ~ treatment, data = dfm)
##
## Linear Hypotheses:
##          Estimate Std. Error t value  Pr(<t)
## C 1 >= 0  -0.5267     0.4536  -1.161 0.13407
## C 2 >= 0  -0.7413     0.3834  -1.934 0.03855 ∗
## C 3 >= 0  -0.8298     0.3569  -2.325 0.01921 ∗
## C 4 >= 0  -0.9754     0.3429  -2.845 0.00739 ∗∗
## C 5 >= 0  -1.0157     0.3367  -3.016 0.00537 ∗∗
## ---
## Signif. codes:  0 ’∗∗∗’ 0.001 ’∗∗’ 0.01 ’∗’ 0.05 ’.’ 0.1 ’ ’ 1
## (Adjusted p values reported -- none method)
```

Note, this multiple contrast test is different from the original Williams test (Williams, 1972) used by (Brock et al., 2015). See Frank Bretz (1999) for a comparison.

### A.2.1.7 *Assuming a Poisson distribution of abundances*

### A.2.1.8 *Model fitting*

We are dealing with count data, so a Poisson GLM might be a good choice. GLMs can be fitted using the `glm()` function and here we fit the model from eqn. 3:

```
modpois <- glm(abu ~ treatment, data = dfm,
family = poisson(link = ’log’))
```

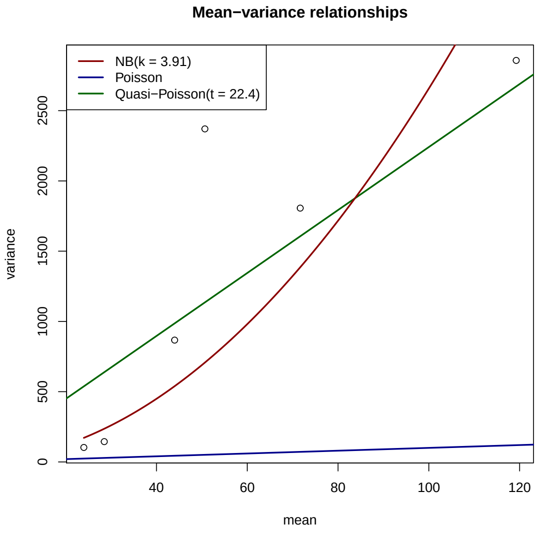
Here `family = poisson(link = ’log’)` specifies that we want to fit a poisson model using a log link between response and predictors.

The `summary` gives the estimated coefficients, standard errors and Wald Z tests:

```
(sum_pois <- summary(modpois))

##
## Call:
## glm(formula = abu ~ treatment, family = poisson(link = "log"),
##     data = dfm)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -6.7625  -2.7621  -0.8219   2.7172   6.6602
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    4.78122    0.04579 104.423  < 2e-16 ***
## treatmentT0.1 -0.50920    0.08214  -6.199 5.69e-10 ***
## treatmentT0.3 -0.99703    0.09835 -10.138  < 2e-16 ***
## treatmentT1   -0.85595    0.09314  -9.190  < 2e-16 ***
## treatmentT3   -1.60317    0.12643 -12.680  < 2e-16 ***
## treatmentT10  -1.43132    0.14014 -10.213  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 604.79  on 17  degrees of freedom
## Residual deviance: 273.77  on 12  degrees of freedom
## AIC: 387.63
##
## Number of Fisher Scoring iterations: 5
```

But is a poisson distribution appropriate here? A property of the poisson distribution is that its variance is equal to the mean. A simple diagnostic would be to plot group variances vs. group means:

```
require(plyr)
# mean and variance per treatment
musd <- ddply(dfm, .(treatment), summarise,
              mu = mean(abu),
              var = var(abu))
musd

##   treatment        mu      var
## 1   Control 119.25000 2857.583
## 2      T0.1  71.66667 1806.333
## 3      T0.3  44.00000  867.000
## 4        T1  50.66667 2370.333
## 5        T3  24.00000  103.000
## 6       T10  28.50000  144.500

# plot mean vs var
plot(var ~ mu, data = musd,
     xlab = 'mean', ylab = 'variance',
     main = 'Mean-variance relationships')
# poisson
abline(a = 0, b = 1, col = 'darkblue', lwd = 2)
# quasi-Poisson
abline(a = 0, b = 22.41, col = 'darkgreen', lwd = 2)
# negative binomial
curve(x + (x^2 / 3.91), from = 24, to = 119.25, add = TRUE,
      col = 'darkred', lwd = 2)
legend('topleft',
       legend = c('NB(k = 3.91)','Poisson','Quasi-Poisson(t=22.4)'),
       col = c('darkred', 'darkblue', 'darkgreen'),
       lty = c(1,1, 1),
       lwd = c(2,2, 2))
```



I also added the assumed mean-variance relationships of the Poisson, quasi-Poisson and negative binomial models (see below). We clearly see that the variance increases much more than would be expected under the poisson distribution (the data is overdispersed). Moreover, we could check overdispersion from the `summary`: If the ratio of residual deviance to degrees of freedom is >1 the data is overdispersed.

```
sum_pois$deviance / sum_pois$df.residual

## [1] 22.81412
```

### A.2.1.9 *Apply quasi-Poisson to deal with overdispersion*

The plot above suggests that the variance may increasing stronger than the mean and a quasi-Poisson or negative binomial model might be more appropriate for this data.

### A.2.1.10 *Model fitting*

Fitting a quasi-Poisson model (eqn. 4) is straight forward:

```
modqpois <- glm(abu ~ treatment, data = dfm, family = 'quasipoisson')
```

The summary gives the estimated coefficients:

```
summary(modqpois)

##
## Call:
## glm(formula = abu ~ treatment, family = "quasipoisson",
##     data = dfm)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -6.7625  -2.7621  -0.8219   2.7172   6.6602
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     4.7812     0.2168  22.058 4.43e-11 ***
## treatmentT0.1  -0.5092     0.3889  -1.309   0.2149
## treatmentT0.3  -0.9970     0.4656  -2.142   0.0534 .
## treatmentT1    -0.8560     0.4409  -1.941   0.0761 .
## treatmentT3    -1.6032     0.5985  -2.679   0.0201 *
## treatmentT10   -1.4313     0.6634  -2.157   0.0519 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for quasipoisson family taken to be 22.411)
##
##     Null deviance: 604.79  on 17  degrees of freedom
## Residual deviance: 273.77  on 12  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

, with the dispersion parameter $\Theta = 22.41055$. Note, that the coefficients estimates are the same as from the Poisson model, only the standard errors are scaled/wider.

### A.2.1.11 *Inference on general treatment effect*

An F-test can be performed using `drop1()`:

```
drop1(modqpois, test = ’F’)

## Single term deletions
##
## Model:
## abu ~ treatment
##           Df Deviance F value  Pr(>F)
## <none>          273.77
## treatment  5   604.79  2.9019 0.06059 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

Here we would reject that there is treatment effect (at alpha = 0.05).

### A.2.1.12 *Inference on LOEC*

The LOEC can be determined with `multcomp`:

```
summary(glht(modqpois, linfct = mcp(treatment = ’Dunnett’),
             alternative = ’less’),
        test = adjusted(’holm’))

##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Dunnett Contrasts
##
##
## Fit: glm(formula = abu ~ treatment, family = "quasipoisson",
##          data = dfm)
##
## Linear Hypotheses:
##                     Estimate Std. Error z value Pr(<z)
## T0.1 - Control >= 0  -0.5092     0.3889  -1.309 0.0952 .
## T0.3 - Control >= 0  -0.9970     0.4656  -2.142 0.0619 .
## T1 - Control >= 0    -0.8560     0.4409  -1.941 0.0619 .
## T3 - Control >= 0    -1.6032     0.5985  -2.679 0.0185 *
## T10 - Control >= 0   -1.4313     0.6634  -2.157 0.0619 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## (Adjusted p values reported -- holm method)
```

, which determines 3 mg/L as LOEC.

#### A.2.1.13 *Assuming a negative binomial distribution of abundances*

#### A.2.1.14 *Model fitting*

To fit a negative binomial GLM (eqn. 5) we could use `glm.nb()` from the MASS package (Venables and Ripley, 2002):

```
require(MASS)
modnb <- glm.nb(abu ~ treatment, data = dfm)
```

The estimated coefficients:

```
summary(modnb)

##
## Call:
## glm.nb(formula = abu ~ treatment, data = dfm,
##     init.theta = 3.905898474,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2554  -0.8488  -0.3020   0.5954   1.5899
```

```
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     4.7812     0.2571  18.596  < 2e-16 ***
## treatmentT0.1  -0.5092     0.3951  -1.289  0.19746
## treatmentT0.3  -0.9970     0.3988  -2.500  0.01241 *
## treatmentT1    -0.8560     0.3975  -2.153  0.03130 *
## treatmentT3    -1.6032     0.4066  -3.943 8.05e-05 ***
## treatmentT10   -1.4313     0.4601  -3.111  0.00186 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(3.9059)
##    family taken to be 1)
##
##     Null deviance: 39.057  on 17  degrees of freedom
## Residual deviance: 18.611  on 12  degrees of freedom
## AIC: 181.24
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  3.91
##           Std. Err.:  1.37
##
##  2 x log-likelihood:  -167.238
```

, with $\kappa = 3.91$.

### A.2.1.15 *Inference on general treatment effect (LR-test)*

For an LR-Test we need to first fit a reduced model:

```
modnb.null <- glm.nb(abu ~ 1, data = dfm)
```

, so that the dispersion parameter $\kappa$ is re-estimated for the reduced model. Then we can compare these two models with a LR-Test:

```
anova(modnb, modnb.null, test = 'Chisq')

## Likelihood ratio tests of Negative Binomial Models
##
## Response: abu
##       Model    theta Resid. df    2 x log-lik.   Test df LR stat.
## 1         1 1.861577        17       -181.2281
## 2 treatment 3.905898        12       -167.2383 1 vs 2  5 13.98985
##    Pr(Chi)
## 1
## 2 0.015674
```

, which suggests a treatment related effect on abundances.

#### A.2.1.16 *Inference on general treatment effect (parametric bootstrap)*

To test the LR statistic using paramtric bootstrap, we use two custom functions:

The first function myPBrefdist generates a boostrap sample and return the LR statistic for this sample:

```r
#' PB of LR statistic
#' @param m1 Full model
#' @param m0 reduced model
#' @param  data data used in the models
#' @return LR of boostrap
# generate reference distribution
myPBrefdist <- function(m1, m0, data){
  # simulate from null
  x0 <- simulate(m0)
  # refit with new data
  newdata0 <- data
  newdata0[ , as.character(formula(m0)[[2]])] <- x0
  m1r <-  try(update(m1, .~., data = newdata0), silent = TRUE)
  m0r <- try(update(m0, .~., data = newdata0), silent = TRUE)
  # check convergence (otherwise return NA for LR)
  if(inherits(m0r, "try-error") | inherits(m1r, "try-error")){
    LR <- 'convergence error'
  } else {
    if(!is.null(m0r[['th.warn']]) | !is.null(m1r[['th.warn']])){
      LR <- 'convergence error'
    } else {
      LR <- -2 * (logLik(m0r) - logLik(m1r))
    }
  }
  return(LR)
}
```

The second one (myPBmodcomp) repeats myPBrefdist many time and returns a p-value:

```r
#' generate LR distribution and return p value
#' @param m1 Full model
#' @param m0 reduced model
#' @param data data used in m1 and m0
#' @param npb number of bootstrap samples
#' @return p-value of boostrapped LR values
myPBmodcomp <- function(m1, m0, data, npb){
  ## calculate reference distribution
  LR <- replicate(npb, myPBrefdist(m1 = m1, m0 = m0, data = data),
                  simplify = TRUE)
  LR <- as.numeric(LR)
  nconv_LR <- sum(!is.na(LR))
  ## original stats
  LRo <- c(-2 * (logLik(m0) - logLik(m1)))
  ## p-value from parametric bootstrap
  p.pb <- mean(c(LR, LRo) >= LRo, na.rm = TRUE)
  return(list(nconv_LR = nconv_LR, p.pb = p.pb))
}
```

Sounds complicated, but we can easily apply this to the negativ binomial model using:

```
set.seed(1234)
myPBmodcomp(modnb, modnb.null, data = dfm, npb = 500)

## $nconv_LR
## [1] 499
##
## $p.pb
## [1] 0.042
```

Here, we specify to generate 500 bootstrap samples (`npb = 500`). Of these 500 samples, 499 converged (`nconv_LR`) (one did not and throws some errors) and gives a p-value of 0.042.

### A.2.1.17 *Inference on LOEC*

This is similar to the other parametric models:

```
summary(glht(modnb, linfct = mcp(treatment = 'Dunnett'),
        alternative = 'less'),
        test = adjusted('holm'))

##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Dunnett Contrasts
##
##
## Fit: glm.nb(formula = abu ~ treatment, data = dfm,
##     init.theta = 3.905898474,
##     link = log)
##
## Linear Hypotheses:
##                      Estimate Std. Error z value   Pr(<z)
## T0.1 - Control >= 0   -0.5092     0.3951  -1.289 0.098731 .
## T0.3 - Control >= 0   -0.9970     0.3988  -2.500 0.018615 *
## T1 - Control >= 0     -0.8560     0.3975  -2.153 0.031300 *
## T3 - Control >= 0     -1.6032     0.4066  -3.943 0.000201 ***
## T10 - Control >= 0    -1.4313     0.4601  -3.111 0.003727 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- holm method)
```

which suggests a LOEC at the 0.3 mg/l treatment.

#### A.2.1.18 *Non-parametric methods*

#### A.2.1.19 *Kruskal-Wallis Test*

We can use the Kruskal-Wallies test to check if there is a difference between treatments:

```
kruskal.test(abu ~ treatment, data = dfm)

##
##  Kruskal-Wallis rank sum test
##
## data:  abu by treatment
## Kruskal-Wallis chi-squared = 8.219, df = 5, p-value = 0.1446
```

#### A.2.1.20 *Pairwise Wilcoxon test*

To determine the LOEC we could use a Pairwise Wilcoxon test. The built-in `pairwise.wilcox.test()` compares by default all levels (Tukey-contrasts). We are only interested in a subset of these comparisons (Dunnett-contrast). Therefore, we use a custom function, which is a wrapper around `wilcox.exact` from the exactRankTests package:

```
#’ pairwise wilcox.test with dunnett contrasrs
#’ @param y numeric; vector of data values
#’ @param g factor; grouping vector
#’ @param dunnett logical; if TRUE dunnett contrast, otherwise
Tukey-contrasts
#’ @param padj character; method for p-adjustment, see ?p.adjust.
#’ @param ... other arguments passed to exactRankTests::wilcox.exact
pairwise_wilcox <- function(y, g, dunnett = TRUE, padj=’holm’,...){
  if(!require(exactRankTests)){
    stop(’Install exactRankTests package’)
  }
  tc <- t(combn(nlevels(g), 2))
  # take only dunnett comparisons
  if(dunnett){
    tc <- tc[tc[ , 1] == 1, ]
  }
  pval <- numeric(nrow(tc))
  # use wilcox.exact (for tied data)
  for(i in seq_len(nrow(tc))){
    pval[i] <- wilcox.exact(y[as.numeric(g) == tc[i, 2]],
                            y[as.numeric(g) == tc[i, 1]],exact=TRUE,
                            ...)$p.value
  }

  # adjust p-values
  pval <- p.adjust(pval, padj)
  names(pval) = paste(levels(g)[tc[,1]], levels(g)[tc[,2]],
           sep = ’ vs. ’)
  return(pval)
}
```

Here, we use one-sided Dunnett contrasts and adjust p-values using Holm’s method:

```
pairwise_wilcox(y = dfm$abu, g = dfm$treatment,
                dunnett = TRUE, p.adj = ’holm’, alternative = ’less’)

## Control vs. T0.1 Control vs. T0.3  Control vs. T1  Control vs. T3
##        0.2285714        0.2285714       0.2285714       0.1428571
##  Control vs. T10
##        0.2285714
```

This indicates no treatment effect at no level of concentration.

### A.2.2 *Binomial data example*

#### A.2.2.1 *Introduction*

Here we will show how to analyse binomial data (*x out of n*). Data is provided in Newman (2012) (example 5.1, page 223) and EPA (2002). Ten fathead minnow (*Pimephales promelas*) larvals were exposed to sodium pentachlorophenol (NaPCP) and proportions of the total number alive at the end of the exposure reported.

First we load the data:

```
df <- read.table(header = TRUE, text = ’conc A B C D
0 1 1 0.9 0.9
32 0.8 0.8 1 0.8
64 0.9 1 1 1
128 0.9 0.9 0.8 1
256 0.7 0.9 1 0.5
512 0.4 0.3 0.4 0.2’)
df

##   conc   A   B   C   D
## 1    0 1.0 1.0 0.9 0.9
## 2   32 0.8 0.8 1.0 0.8
## 3   64 0.9 1.0 1.0 1.0
## 4  128 0.9 0.9 0.8 1.0
## 5  256 0.7 0.9 1.0 0.5
## 6  512 0.4 0.3 0.4 0.2
```

The we do some house-keeping, reformat the data and convert concentration to a factor:

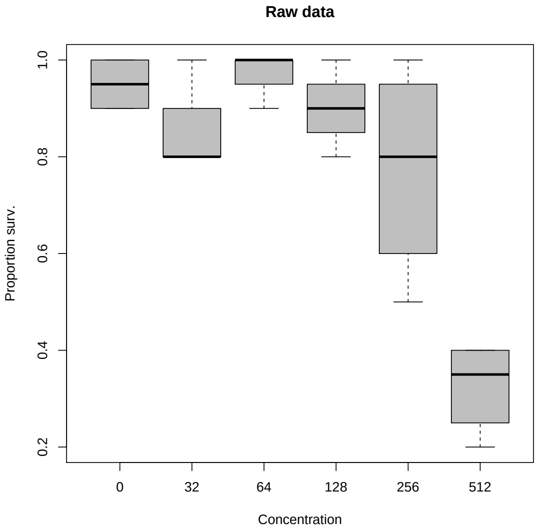
```
require(reshape2)
# wide to long
dfm <- melt(df, id.vars = 'conc', value.name = 'y',
            variable.name = 'tank')
# conc as factor
dfm$conc <- factor(dfm$conc)
```

So after data cleaning the data looks like

```
head(dfm)

##   conc tank   y
## 1    0    A 1.0
## 2   32    A 0.8
## 3   64    A 0.9
## 4  128    A 0.9
## 5  256    A 0.7
## 6  512    A 0.4
```

Let's have a first look at the data:

```
boxplot(y ~ conc, data = dfm,
        xlab = 'Concentration', ylab = 'Proportion surv.',
        main = 'Raw data', col = 'grey75')
```



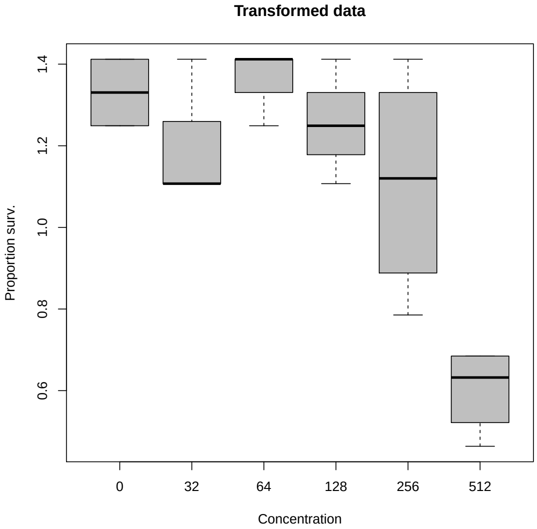
This plot indicates a strong effect at the highest concentration.

#### A.2.2.2 *Assuming a normal distribution of transformed proportions*

First, we arcsine transform (eqn. 6) the proportions:

```
dfm$y_asin <- ifelse(dfm$y == 1,
                     asin(1) - asin(sqrt(1/40)),
                     ifelse(dfm$y == 0,
                            asin(sqrt(1/40)),
                            asin(sqrt(dfm$y))
                            )
                     )
```

```
boxplot(y_asin ~ conc, data = dfm,
        xlab = 'Concentration', ylab = 'Proportion surv.',
        main = 'Transformed data', col = 'grey75')
```



Then, like in the count data example we fit the model using `lm()`:

```
modlm <- lm(y_asin ~ conc, data = dfm)
```

The summary gives the estimated coefficients:

```
summary(modlm)

##
## Call:
## lm(formula = y_asin ~ conc, data = dfm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32401 -0.08149 -0.00527  0.08150  0.30261
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.33053    0.07693  17.295 1.16e-12 ***
## conc32      -0.14717    0.10880  -1.353   0.1929
## conc64       0.04074    0.10880   0.374   0.7124
## conc128     -0.07622    0.10880  -0.701   0.4925
## conc256     -0.22113    0.10880  -2.032   0.0571 .
## conc512     -0.72735    0.10880  -6.685 2.86e-06 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.1539 on 18 degrees of freedom
## Multiple R-squared:  0.7871,Adjusted R-squared:  0.7279
## F-statistic: 13.31 on 5 and 18 DF,  p-value: 1.612e-05
```

The F-test suggests a treatment related effect:

```
drop1(modlm, test = ’F’)

## Single term deletions
##
## Model:
## y_asin ~ conc
##        Df Sum of Sq     RSS     AIC F value    Pr(>F)
## <none>              0.42613 -84.746
## conc    5    1.5753 2.00142 -57.621  13.308 1.612e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

And the LOEC is at the highest concentration:

```
summary(glht(modlm, linfct = mcp(conc = ’Dunnett’),
        alternative = ’less’),
        test = adjusted(’holm’))

##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Dunnett Contrasts
##
##
## Fit: lm(formula = y_asin ~ conc, data = dfm)
##
## Linear Hypotheses:
##                Estimate Std. Error t value   Pr(<t)
## 32 - 0 >= 0   -0.14717    0.10880  -1.353    0.289
## 64 - 0 >= 0    0.04074    0.10880   0.374    0.644
## 128 - 0 >= 0 -0.07622    0.10880  -0.701    0.493
## 256 - 0 >= 0 -0.22113    0.10880  -2.032    0.114
## 512 - 0 >= 0 -0.72735    0.10880  -6.685 7.14e-06 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## (Adjusted p values reported -- holm method)
```

#### A.2.2.3 *Assuming a binomial distribution*

The binomial model with a logit link (eqn. 7) between predictors and response can be fitted using the `glm()` function:

```
modglm <- glm(y ~ conc , data = dfm, family = binomial(link=’logit’),
              weights = rep(10, nrow(dfm)))
```

Here the weights arguments, indicates how many fish where exposed in each treatment (N=10, eqn .7).

The summary gives the estimated coefficients:

```
summary(modglm)

##
## Call:
## glm(formula = y ~ conc, family = binomial(link = "logit"),
##     data = dfm,
##     weights = rep(10, nrow(dfm)))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8980  -0.5723   0.0000   0.7869   2.2578
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.9444     0.7255   4.059 4.94e-05 ***
## conc32       -1.2098     0.8499  -1.423   0.1546
## conc64        0.7191     1.2458   0.577   0.5638
## conc128      -0.7472     0.8967  -0.833   0.4047
## conc256      -1.7077     0.8183  -2.087   0.0369 *
## conc512      -3.6753     0.8002  -4.593 4.37e-06 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 88.672  on 23  degrees of freedom
## Residual deviance: 23.889  on 18  degrees of freedom
## AIC: 72.862
##
## Number of Fisher Scoring iterations: 5
```

To perform a LR-test we can used the `drop1()` function:

```
drop1(modglm, test = ’Chisq’)

## Single term deletions
##
## Model:
## y ~ conc
##        Df Deviance     AIC    LRT   Pr(>Chi)
## <none>      23.889  72.862
## conc    5   88.672 127.645 64.783 1.243e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Also with the binomial model the LOEC is at the highest concentration:

```
summary(glht(modglm, linfct = mcp(conc = 'Dunnett'),
        alternative = 'less'),
        test = adjusted('holm'))

##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Dunnett Contrasts
##
##
## Fit: glm(formula = y ~ conc, family = binomial(link = "logit"),
##     data = dfm,
##     weights = rep(10, nrow(dfm)))
##
## Linear Hypotheses:
##               Estimate Std. Error z value   Pr(<z)
## 32 - 0 >= 0    -1.2098     0.8499  -1.423   0.2319
## 64 - 0 >= 0     0.7191     1.2458   0.577   0.7181
## 128 - 0 >= 0   -0.7472     0.8967  -0.833   0.4047
## 256 - 0 >= 0   -1.7077     0.8183  -2.087   0.0738 .
## 512 - 0 >= 0   -3.6753     0.8002  -4.593 1.09e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- holm method)
```

### A.2.3 *References*

Bretz, Frank (1999). "Powerful modifications of Williams' test on trend." PhD thesis. Universität Hannover.

Bretz, Frank, Torsten Hothorn, and Peter H. Westfall (2010). *Multiple comparisons using R*. London: Chapman /& Hall.

Brock, T. C. M., M. Hammers-Wirtz, U. Hommen, T. G. Preuss, H-T. Ratte, I. Roessink, T. Strauss, and P. J. Van den Brink (2015). "The minimum detectable difference (MDD) and the interpretation of treatment-related effects of pesticides in experimental ecosystems." In: *Environmental Science and Pollution Research* 22.2, pp. 1160–1174.

EPA (2002). *Methods for Measuring the Acute Toxicity of Effluents and Receiving Waters to Freshwater and Marine Organisms*. U.S. Environmental Protection Agency.

Hothorn, T., F. Bretz, and P. Westfall (2008). "Simultaneous inference in general parametric models." In: *Biometrical Journal* 50.3, pp. 346–363.

Newman, Michael C (2012). *Quantitative ecotoxicology*. Boca Raton, FL: Taylor & Francis.

Venables, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with S*. Fourth edition. New York: Springer.

Williams, D. A. (1972). "The comparison of several dose levels with a zero dose control." In: *Biometrics*, pp. 519–531.

# B

# SUPPLEMENTAL MATERIAL FOR: TAXIZE: TAXONOMIC SEARCH AND RETRIEVAL

## B.1 A COMPLETE REPRODUCIBLE WORKFLOW, FROM A SPECIES LIST TO A PHYLOGENY, AND DISTRIBUTION MAP.

If you aren't familiar with a complete workflow in R, it may be difficult to visualize the process. In R, everything is programmatic, so the whole workflow can be in one place, and be repeated whenever necessary. The following is a workflow for taxize, going from a species list to a phylogeny.

First, install taxize

```r
install.packages("taxize")
```

Then load it into R

```r
library(taxize)
```

Most of us will start out with a species list, something like the one below. Note that each of the names is spelled incorrectly.

```r
splist <- c("Helanthus annuus", "Pinos contorta",
         "Collomia grandiflorra", "Rosa california",
         "Mimulus bicolour", "Nicotiana glauca", "Maddia sativa")
```

There are many ways to resolve taxonomic names in taxize. Of course, the ideal name resolver will do the work behind the scenes for you so that you don't have to do things like fuzzy matching. There are a few services in taxize like this we can choose from: the Global Names Resolver service from EOL (see function *gnr_resolve*) and the Taxonomic Name Resolution Service from iPlant (see function *tnrs*). In this case let's use the function *tnrs*.

```r
# The tnrs function accepts a vector of 1 or more
splist_tnrs <- tnrs(query = splist, getpost = "POST",
         source_ = "iPlant_TNRS")
# Remove some fields
(splist_tnrs <- splist_tnrs[, !names(splist_tnrs) %in%
         c("matchedName", "annotations",
    "uri")])
```

```
#           submittedName         acceptedName    sourceId score
# 5      Helanthus annuus    Helianthus annuus iPlant_TNRS  0.98
# 1        Pinos contorta       Pinus contorta iPlant_TNRS  0.96
# 7 Collomia grandiflorra Collomia grandiflora iPlant_TNRS  0.99
# 6       Rosa california     Rosa californica iPlant_TNRS  0.99
# 4      Mimulus bicolour      Mimulus bicolor iPlant_TNRS  0.98
# 3      Nicotiana glauca     Nicotiana glauca iPlant_TNRS     1
# 2          Maddia sativa         Madia sativa iPlant_TNRS  0.97

# Note the scores. They suggest that there were no perfect matches,
# but they were all very close, ranging from 0.77 to 0.99
# (1 is the highest).
# Let’s assume the names in the ’acceptedName’ column
# are correct (and they should
# be).
# So here’s our updated species list
(splist <- as.character(splist_tnrs$acceptedName))

# [1] "Helianthus annuus"  "Pinus contorta"  "Collomia grandiflora"
# [4] "Rosa californica"  "Mimulus bicolor"  "Nicotiana glauca"
# [7] "Madia sativa"
```

Another thing we may want to do is collect common names for our taxa.

```
tsns <- get_tsn(searchterm = splist, searchtype = "sciname",
        verbose = FALSE)
comnames <- lapply(tsns, getcommonnamesfromtsn)
# Unfortunately, common names are not standardized like species
# names, so there are multiple common names for each taxon
sapply(comnames, length)

# [1] 3 3 3 3 3 3 3

# So let’s just take the first common name for each species
comnames_vec <- do.call(c, lapply(comnames,
     function(x) as.character(x[1, "comname"])))
# And we can make a data.frame of our scientific and common names
(allnames <- data.frame(spname = splist, comname = comnames_vec))

#                  spname                       comname
# 1    Helianthus annuus              common sunflower
# 2       Pinus contorta                lodgepole pine
# 3 Collomia grandiflora        largeflowered collomia
# 4     Rosa californica           California wildrose
# 5      Mimulus bicolor yellow and white monkeyflower
# 6     Nicotiana glauca                  tree tobacco
# 7         Madia sativa                 coast tarweed
```

Another common task is getting the taxonomic tree upstream from your study taxa. We often know what family or order our taxa are in, but it we often don’t know the tribes, subclasses, and superfamilies. taxize provides many avenues to getting classifications. Two of them are accessible via a single function (*classification*): the Integrated Taxonomic Information System (ITIS) and National Center for Biotechnology Information (NCBI); and via the Catalogue of Life (see function *col_classification*):

```
# As we already have Taxonomic Serial Numbers from ITIS, let’s just
# get classifications from ITIS.Note that we could use uBio instead.
class_list <- classification(tsns)
sapply(class_list, nrow)

# [1] 12 11 12 12 12 12 12

# And we can attach these names to our allnames data.frame
library(plyr)
gethiernames <- function(x) {
    temp <- x[, c("rankName", "taxonName")]
    values <- data.frame(t(temp[, 2]))
    names(values) <- temp[, 1]
    return(values)
}
class_df <- ldply(class_list, gethiernames)
allnames_df <- merge(allnames, class_df, by.x = "spname",
         by.y = "Species")
# Now that we have allnames_df, we can start to see some
# relationships among species simply by their shared taxonomic names
allnames_df[1:2, ]

#                 spname                comname Kingdom     Subkingdom
# 1 Collomia grandiflora largeflowered collomia Plantae Viridaeplantae
# 2    Helianthus annuus       common sunflower Plantae Viridaeplantae
#   Infrakingdom     Division     Subdivision Infradivision
# 1 Streptophyta Tracheophyta Spermatophytina  Angiospermae
# 2 Streptophyta Tracheophyta Spermatophytina  Angiospermae
#   Class         Superorder     Order        Family      Genus
# 1  Magnoliopsida Asteranae  Ericales Polemoniaceae   Collomia
# 2  Magnoliopsida Asteranae Asterales    Asteraceae Helianthus


# Ah, so Abies and Bartlettia are in different infradivisions, but
# share taxonomic names above that point.
```

However, taxonomy can only get you so far. Shared ancestry can be reconstructed from molecular data, and phylogenies created. Phylomatic is a web service with an API that we can use to get a phylogeny.

```
# Fetch phylogeny from phylomatic
phylogeny <- phylomatic_tree(taxa = as.character(allnames$spname),
    taxnames = TRUE,
    get = "POST", informat = "newick", method = "phylomatic",
    storedtree = "R20120829",
    taxaformat = "slashpath", outformat = "newick", clean = "true",
    parallel = TRUE)
# Format teeth-labels
phylogeny$tip.label <- capwords(phylogeny$tip.label,
     onlyfirst = TRUE)
# plot phylogeny
plot(phylogeny)
```

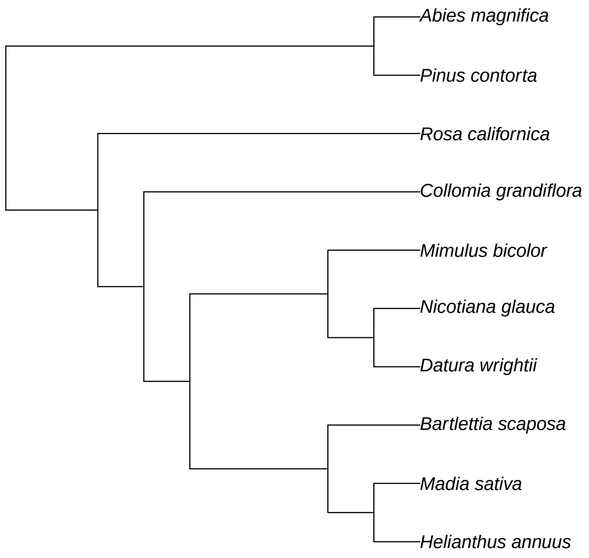


Figure B.1.: A phylogeny created using taxize.

Using the species list, with the corrected names, we can now search for occurrence data. The Global Biodiversity Information Facility (GBIF) has the largest collection of records data, and has a API that we can interact with programmatically from R. First, we need to install rgbif.

```
# Install rgbif from github.com
install.packages("devtools")
library(devtools)
install_github("rgbif", "ropensci")
```

Now we can search for occurrences for our species list and make a map.

```
library(rgbif)
library(ggplot2)
```

```
# get occurences
occurr_list <- occurrencelist_many(as.character(allnames$spname),
    coordinatestatus = TRUE,
    maxresults = 100, removeZeros = TRUE,
    fixnames = "changealltorig")
# Make a map
p <- gbifmap_list(occurr_list) +
    guides(col = guide_legend(title = "", nrow = 3,
    byrow = TRUE)) + theme(legend.position = "bottom",
    legend.key = element_blank()) +
    coord_equal()
p
```

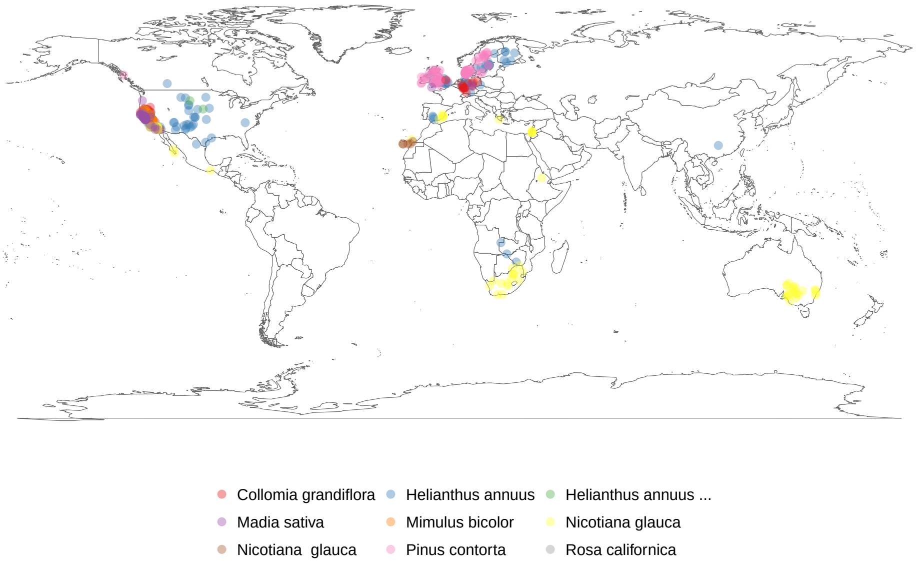


Figure B.2.: A map created using taxize.

## B.2 MATCHING SPECIES TABLES WITH DIFFERENT TAXONOMIC RESOLUTION

Trait-based approaches are a promising tool in ecology. Unlike taxonomy-based methods, traits may not be constrained to biogeographic boundaries (Baird et al., 2011) and have potential to disentangle the effects of multiple stressors (Statzner and Bêche, 2010).

To analyse trait-composition abundance data must be matched with trait databases like (Usseglio-Polatera et al., 2000). However these two datatables may contain species information on different taxonomic levels and perhaps data must be aggregated to a joint taxonomic level.

taxize can help in this data-cleaning step, providing a reproducible workflow. Here we illustrate this on a small fictitious example.

Suppose we have fuzzy coded trait table with 2 traits with 3 respectively 2 modalities:

```
(traits <- read.table(header = TRUE, sep = ’;’, stringsAsFactors=FALSE,
                      text = ’taxon;T1M1;T1M2;T1M3;T2M1;T2M2
Gammarus sp.;0;0;3;1;3
Potamopyrgus antipodarum;1;0;3;1;3
Coenagrion sp.;3;0;1;3;1
Enallagma cyathigerum;0;3;1;0;3
Erythromma sp.;0;0;3;3;1
Baetis sp.;0;0;0;0;0
’))

                     taxon T1M1 T1M2 T1M3 T2M1 T2M2
1             Gammarus sp.    0    0    3    1    3
2 Potamopyrgus antipodarum    1    0    3    1    3
3           Coenagrion sp.    3    0    1    3    1
4    Enallagma cyathigerum    0    3    1    0    3
5           Erythromma sp.    0    0    3    3    1
6               Baetis sp.    0    0    0    0    0
```

And want to match this to a table with abundances:

```
(abundances <- read.table(header = TRUE, sep = ’;’, stringsAsFactors=FALSE,
                          text = ’taxon;abundance;sample
Gammarus roeseli;5;1
Gammarus roeseli;6;2
Gammarus tigrinus;7;1
Gammarus tigrinus;8;2
Coenagrionidae;10;1
Coenagrionidae;6;2
Potamopyrgus antipodarum;10;1
xxxxx;10;2
’))

                     taxon abundance sample
1         Gammarus roeseli         5      1
2         Gammarus roeseli         6      2
3        Gammarus tigrinus         7      1
4        Gammarus tigrinus         8      2
5           Coenagrionidae        10      1
6           Coenagrionidae         6      2
7 Potamopyrgus antipodarum        10      1
8                    xxxxx        10      2
```

First we do some basic data-cleaning and create a lookup-table, that will link taxa in trait table with the abundance table.

```
# first we remove ’ sp.’ from out trait table:
traits$taxon_cleaned <- tolower(gsub(" sp.", "", traits$taxon))
# since abundance tables can be very long with repeating taxa, we look only
# at unique taxon names This will be a lookup-table linking taxon names
# between both tables
lookup <- data.frame(taxon = tolower(unique(abundances$taxon)),
       stringsAsFactors = FALSE)
```

The we query the taxonomic hierarchy for both tables, this will be the backbone of this procedure:

```
library(taxize)
traits_classi <- classification(get_uid(traits$taxon_cleaned))
lookup_classi <- classification(get_uid(lookup$taxon))
```

First we look if we can find any direct matches between taxon names:

```
# first search for direct matches
direct <- match(lookup$taxon, traits$taxon_cleaned)
# and add the matched name to our lookup table
lookup$traits <- tolower(traits$taxon[direct])
lookup$match <- ifelse(!is.na(direct), "direct", NA)
lookup

                     taxon                   traits  match
1         gammarus roeseli                     <NA>   <NA>
2        gammarus tigrinus                     <NA>   <NA>
3           coenagrionidae                     <NA>   <NA>
4 potamopyrgus antipodarum potamopyrgus antipodarum direct
5                    xxxxx                     <NA>   <NA>
```

We found a direct match - *potamopyrgus antipodarum* - so nothing to do here.

Next we look for species which are on a higher taxonomic resolution than our trait table. For these species we will take directly the trait-data since no better information is available.

```
# look for cases where taxonomic resolution in abundance data is higher
# than in trait data: here we take the trait-values for the lower
# resolution.
for (i in which(is.na(lookup$traits))) {
    if (is.data.frame(lookup_classi[[i]])) {
        matches <- tolower(lookup_classi[[i]]$ScientificName) %in%
        traits$taxon_cleaned
        if (any(matches)) {
            lookup$traits[i] <- tolower
            (lookup_classi[[i]]$ScientificName[matches])
            lookup$match[i] <- lookup_classi[[i]]$Rank[matches]
        }
    }
}
lookup

                     taxon                  traits  match
1          gammarus roeseli                gammarus  genus
2         gammarus tigrinus                gammarus  genus
3            coenagrionidae                    <NA>   <NA>
4 potamopyrgus antipodarum potamopyrgus antipodarum direct
5                    xxxxx                    <NA>   <NA>
```

So our abundance data has two *Gammarus* species, however trait data is only on genus level.

The next step is to search for species were we have to aggregate trait-data, since our abundance data is on a lower taxonomic level. We are walking the taxonomic latter for the species in our trait-data upwards and search for matches with out abundance data. Since we'll have many taxa in the trait-data belonging to one taxon, we'll take the median modality scores as an approximation. Of course also other methods may be used here, e.g. weighting by genetic distance.

```
# look for cases taxonomic resolution in abundance data is lower than in
# trait data, here we need to aggregate the trait-values (eg. median value
# for modality)
for (i in which(is.na(lookup$traits))) {
    # find matches
    agg <- sapply(traits_classi, function(x) any(
    tolower(x$ScientificName) %in%
        lookup$taxon[i]))
    if (sum(agg) > 1) {
        # add taxon as aggregate to trait-table
        traits <- rbind(traits, c(paste0(lookup$taxon[i], "-aggregated"),
        apply(traits[agg,
            2:6], 2, median), paste0(lookup$taxon[i], "-aggregated")))
        # fill lookup table
        lookup$traits[i] <- paste0(lookup$taxon[i], "-aggregated")
        lookup$match[i] <- "aggregated"
    }
}
lookup

#                      taxon                    traits      match
# 1         gammarus roeseli                  gammarus      genus
# 2        gammarus tigrinus                  gammarus      genus
# 3           coenagrionidae coenagrionidae-aggregated aggregated
# 4 potamopyrgus antipodarum  potamopyrgus antipodarum     direct
# 5                    xxxxx                      <NA>       <NA>
```

Finally we have only one taxon left - clearly an error. We remove this from our dataset:

```
abundances <- abundances[!abundances$taxon == lookup$taxon[is.na(
lookup$traits)],
    ]
```

Now we can create *species x sites* and *traits x species* matrices, which could be plugged into methods to analyse trait responses [28].

```
# species (as matched with trait table) by site matrix
abundances$traits_taxa <- lookup$traits[match(tolower(abundances$taxon),
        lookup$taxon)]
library(reshape2)
# reshape data to long format and name rows by samples
L <- dcast(abundances, sample ~ traits_taxa, fun.aggregate = sum,
        value.var = "abundance")
rownames(L) <- L$sample
L$sample <- NULL
L

#   coenagrionidae-aggregated gammarus potamopyrgus antipodarum
# 1                        10       12                       10
# 2                         6       14                        0

# traits by species matrix
Q <- traits[, 2:7][match(names(L), traits$taxon_cleaned), ]
rownames(Q) <- Q$taxon_cleaned
Q$taxon_cleaned <- NULL
Q

#                           T1M1 T1M2 T1M3 T2M1 T2M2
# coenagrionidae-aggregated    0    0    1    3    1
# gammarus                     0    0    3    1    3
# potamopyrgus antipodarum     1    0    3    1    3

# check
all(rownames(Q) == colnames(L))

# [1] TRUE
```

This is just an example how taxonomic APIs (via taxize) could be used to search for matches up- and downwards the taxonomic ladder. We are looking forward to integrate other databases into taxize, which will facilitate trait-based analyses in R.

## References

Baird, Donald J, Christopher J O Baker, Robert B Brua, Mehrdad Hajibabaei, Kearon McNicol, Timothy J Pascoe, and Dick de Zwart (2011). “Toward a knowledge infrastructure for traits-based ecological risk assessment.” In: *Integrated Environmental Assessment and Management* 7.2, pp. 209–215.

Statzner, B. and L.A Bêche (2010). "Can biological invertebrate traits resolve effects of multiple stressors on running water ecosystems?" In: *Freshwater Biology* 55, pp. 80–119.

Usseglio-Polatera, Philippe, Michel Bournaud, Philippe Richoux, and Henri Tachet (2000). "Biological and ecological traits of benthic freshwater macroinvertebrates: relationships and definition of groups with similar traits." In: *Freshwater Biology* 43.2, pp. 175–205.

# AUTHOR'S CONTRIBUTIONS

## ARTICLE I

TITLE: Ecotoxicology is not normal - A comparison of statistical approaches for analysis of count and proportion data in ecotoxicology

AUTHORS: Eduard Szöcs and Ralf B. Schäfer

STATUS: Published in 2015 in Environmental Science and Pollution Research, Volume 22, Issue 18, pp 13990-13999

CONTRIBUTIONS: Szöcs (85%) Designed research, Simulated & Analysed data, Discussed results, Wrote manuscript
Schäfer (15%) Designed research, Discussed results, Edited manuscript

## ARTICLE II

TITLE:

AUTHORS:

STATUS:

CONTRIBUTIONS:

## ARTICLE III

TITLE:

AUTHORS: Eduard Szöcs and Ralf B. Schäfer

STATUS:

CONTRIBUTIONS: Szöcs (90%) Schäfer (10%)

## ARTICLE IV

TITLE: taxize: taxonomic search and retrieval in R

AUTHORS: Scott A. Chamberlain and Eduard Szöcs

STATUS: Published in 2013 in F1000Research, Volume 2, Issue 191

CONTRIBUTIONS: Chamberlain (50%) Designed research, Programmed Software, Wrote manuscript
Szöcs (50%) Designed research, Programmed Software, Wrote manuscript

# DECLARATION

I, the author of this work, certify that this work contains no material which has been accepted or submitted for the award of any other degree at any university or other tertiary institution.

The work has been interdependently prepared. All aids and sources have been clearly specified and the contribution of other authors have been documented and reference lists given.

Neustadt a.d. Weinstraße,
August 24, 2016

Eduard Szöcs

# CURRICULUM VITAE

# Eduard Szöcs

## Personal

| | |
|---|---|
| Date of birth | 16.06.1987 |
| Nationality | german |
| Marital Status | single |
| Languages | German (native), English (very good), Romanian (good) |

## Education

| | |
|---|---|
| 04.2014–present | **Ph.D. Environmental Sciences**, *University of Koblenz-Landau*, Landau.<br>Quantification of large scale effects of pesticides on freshwater ecosystems. |
| 04.2012–03.2014 | **M. Sc. Ecotoxicology**, *University of Koblenz-Landau*, Landau.<br>Thesis: Analysing mesocosm experiments: Principal Response Curves vs. Multivariate Generalized Linear Models. |
| 11.2011 | **B. Sc. Umweltwissenschaften**, *University of Koblenz-Landau*, Landau.<br>Thesis: Effects of salinity and pesticides on community structure of macroinvertebrates in Australian streams. |
| 09.2007–11.2011 | **Dipl. Umweltwissenschaften**, *University of Koblenz-Landau*, Landau. |

## Work Experience and Teaching

| | |
|---|---|
| 02.2016 – present | **Research Assistant**, *University of Koblenz-Landau*, Landau.<br>Field Study in Romania, Data analyses, maintenance of databases and servers, PhD research. |
| 11.2015 – present | **Freelance Scientist & Consultant**.<br>Data sourcing, cleaning and analysis with specialization in Environmental & Ecological data. Courses in ecological statistics with the software *"R"*. |
| 04.2015 – 01.2016 | **Research Assistant**, *University of Koblenz-Landau*, Landau.<br>UBA Project: "PSM in Kleingewässern" (FKZ 3714674040/1). Building, maintaining and analysing a nation-wide german pesticide monitoring database. |
| 05.2014 – 04.2015 | **Research Assistant**, *University of Koblenz-Landau*, Landau.<br>Data analyses, maintenance of databases and servers, PhD research. |

*Marktplatz 6 – 67433 Neustadt an der Weinstraße*
☎ *+49 176 621 927 00* • ✉ *eduardszoecs@posteo.de*
*https://edild.github.io*

12.2014 – 02.2015 **Teaching Assistant**, *University of Koblenz-Landau*, Landau.
Multivariate Statistics Course.

12.2013 – 02.2014 **Teaching Assistant**, *University of Koblenz-Landau*, Landau.
Multivariate Statistics Course.

12.2013 **Research Assistant**, *University of Koblenz-Landau*, Landau.
Development of a PostgreSQL-database of german physico-chemical data.

12.2012 – 02.2013 **Teaching Assistant**, *University of Koblenz-Landau*, Landau.
Multivariate Statistics Course.

12.2012 **Research Assistant**, *Department System Ecotoxicology, UFZ – Helmholtz Centre for Environmental Research*, Leipzig.
Development of rspear R-package.

05.2012 – 07.2012 **Internship**, *Department System Ecotoxicology, UFZ – Helmholtz Centre for Environmental Research*, Leipzig.
Field Study on the effects of pesticides on macroinvertebrates.

12.2011 – 02.2012 **Teaching Assistant**, *University of Koblenz-Landau*, Landau.
Multivariate Statistics Course.

07.2010 **Teaching Assistant**, *University of Koblenz-Landau*, Landau.
Aquatic Field Course.

06.2006 – 07.2007 **Internship**, *Landschaftspflegeverband Südpfalz e. V.*, Landau.
Freiwilliges Ökologisches Jahr.

## Programming Skills

Expert R

Intermediate PostgreSQL, PostGIS, GrassGIS, LaTeX, git, regex, html, xml, xpath, cloud computing

Beginner Python, bash, NetLogo, C++, CDK, openbabel

(Beginner = *"I know the basics and can get the job done"*; Intermediate = *"I can effectively apply these tools"*; Expert = *"I can develop and expand these tools."*)

## Software

I have developed or contributed to the following open source software for the R computing environment:

- The **taxize** package (together with Scott Chamberlain) allows taxonomic search, retrieval and handling in R.
- The **webchem** package to retrieve chemical information from the web.
- Contributions to the **vegan** package.
- The **rspear** package calculates $\text{SPEAR}_{\text{pesticides}}$ in R (deprecated).
- A web application to calculate statistical power for population endpoints in mesocosm experiments ( currently offline / deprecated ).
- Various other R packages and functions related to eco(toxico-)logy.

All software is freely available from my github account (`https://github.com/EDiLD`), homepage or CRAN.

## Publications and Conference contributions

### Articles

[1] Laurent Lagadic, Ralf B. Schäfer, Marc Roucaute, **Eduard Szöcs**, Sébastien Chouin, Jérôme de Maupeou, Claire Duchet, Evelyne Franquet, Benoit Le Hunsec, Céline Bertrand, and et al. “No association between the use of Bti for mosquito control and the dynamics of non-target aquatic invertebrates in French coastal and continental wetlands”. In: *Science of The Total Environment* 553 (2016), 486–494.

[2] **Eduard Szöcs** and Ralf B Schäfer. “Statistical hypothesis testing—To transform or not to transform?” In: *Integrated Environmental Assessment and Management* 12.2 (2016), 398–400.

[3] **Eduard Szöcs** and Ralf B. Schäfer. “webchem: an R Package to retrieve Chemical Information from the Web”. In: *Journal of Statistical Software* (under Review, 2016).

[4] John G. Mbaka, **Eduard Szöcs**, and Ralf B. Schäfer. “Meta-analysis on the responses of traits of different taxonomic groups to global and local stressors”. In: *Acta Oecologica* 69 (2015), 65–70.

[5] **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, and Ralf B. Schäfer. “Analysing chemical-induced changes in macroinvertebrate communities in aquatic mesocosm experiments: a comparison of methods”. In: *Ecotoxicology* 24.4 (2015), 760–769.

[6] **Eduard Szöcs** and Ralf B. Schäfer. “Ecotoxicology is not normal: A comparison of statistical approaches for analysis of count and proportion data in ecotoxicology”. In: *Environmental Science and Pollution Research* 22.18 (2015), 13990–13999.

[7] **Eduard Szöcs**, Eckhard Coring, Jürgen Bäthe, and Ralf B. Schäfer. “Effects of anthropogenic salinization on biological traits and community composition of stream macroinvertebrates”. In: *Science of The Total Environment* 468–469 (2014), 943–949.

[8] Scott A. Chamberlain and **Eduard Szöcs**. “taxize: taxonomic search and retrieval in R [v2; ref status: indexed, http://f1000r.es/24v]”. In: *F1000Research* 2.191 (2013). DOI: `10.12688/f1000research.2-191.v2`.

[9] Ralf B. Schäfer, Mirco Bundschuh, Duncan A. Rouch, **Eduard Szöcs**, Peter C. von der Ohe, Vincent Pettigrove, Ralf Schulz, Dayanthi Nugegoda, and Ben J. Kefford. “Effects of pesticide toxicity, salinity and other environmental variables on selected ecosystem functions in streams and the relevance for ecosystem services”. In: *Science of the Total Environment* 415.1 (2012), 69–78.

[10] **Eduard Szöcs**, Ben J. Kefford, and Ralf B. Schäfer. “Is there an interaction of the effects of salinity and pesticides on the community structure of macroinvertebrates?” In: *Science of the Total Environment* 437.1 (2012), 121–126.

### Poster

[1] **Eduard Szöcs** and Ralf B. Schäfer. "Ecotoxicology is not normal". SETAC Europe; Barcelona, 2015.

[2] **Eduard Szöcs** and Scott A. Chamberlain. "taxize: taxonomic search and retrieval in R". International Statistical Ecology Conference; Montpellier, 2014.

[3] **Eduard Szöcs**, Ben J. Kefford, V. Pettigrove, and Ralf B. Schäfer. "Einfluss von Pestiziden und Salinität auf Makroinvertebratengemeinschaften". SETAC GLB; Landau, 2011.

As a service to the scientific community I performed a total of 4 reviews for the journals *Proceedings of the Royal Society B, PhytoKeys, Zookeys* and *Environmental Toxicology and Chemistry*.

## Workshops held

07.2015 **Data analysis in freshwater ecology using R**, *9th Symposium for European Freshwater Sciences*, Geneva.
Workshop held together with Dr. Ralf B. Schäfer and Avit Kumar Bhowmik.
Workshop homepage: `https://github.com/EDiLD/sefs9_Rworkshop`

11.2015 **Data Visualization with ggplot2**, *Workshop held at Young Academics Conference 2015 - Land-Water-Interactions*, Klingenmünster.
Workshop homepage: `https://github.com/EDiLD/r_landau_2015`

Neustadt a.d. Weinstraße, July 20, 2016