Analyses of plants data from the Swiss Biodiversity Monitoring (BDM)

Tobias Roth, Eric Allan, Peter B. Pearman and Valentin Amrhein

2017-06-27

Introduction

Prerequisite

In this vignette we conduct the analyses presented in Roth, Allan, Pearman and Amrhein: Functional ecology and imperfect detection of species. As a requirement to run the presented code the latest version of the `detecionfilter` package should be downloaded from github.

```
devtools::install_github("TobiasRoth/detectionfilter")
```

The presented code depends on the following packages.

```
library(detectionfilter)
library(missForest)
library(RColorBrewer)
library(geometry)
library(mgcv)
library(FD)
```

Data from the Swiss Biodiversity monitoring (BDM)

The `detectionfilter` package contains the data of the plant surveys from the Swiss biodiversity monitoring (`plantsBDM`). The array `y[i,k,j]` contains the observed presence/absence data of the `i=1,2,...,362` $1-km^2$ plots, the `k=1,2,...,1733` observed plant species and the `j=1,2` visits. Furthermore, `plantsBDM` contains the median elevation for each plot (`elevation`), as well the dates (Julian day) for each visit to the plots (`dates`).

We call the species that are occurring at a plot a *community*. In the case of `plantsBDM` we speak of `meta-community` data because they contain observations of 362 communities. Further note, that `y[i,k,j]` contain the information whether or not a species was *observed*. If we refer to *occurrence* we mean the true occurence that is not directly observable during a survey because we are not necessarily detect all species.

```
# Number of plots, species and visits
nplots <- dim(plantsBDM$y)[1]
nspec <- dim(plantsBDM$y)[2]
nvisits <- dim(plantsBDM$y)[3]

# Average and SD number of observed species per plot (i.e. community)
round(mean(apply(apply(plantsBDM$y, c(1,2), max), 1, sum)), 1)
```

```
## [1] 256
```

```
round(sd(apply(apply(plantsBDM$y, c(1,2), max), 1, sum)), 1)
```

```
## [1] 52
```

```r
# Elevational gradient covered by studied plots
range(plantsBDM$elevation)
```

```
## [1]  250 2710
```

```r
mean(plantsBDM$elevation)
```

```
## [1] 1104.144
```

```r
sd(plantsBDM$elevation)
```

```
## [1] 612.2227
```

Trait data

The `detectionfilter` package contains the values for three functional traits for the 1733 species. The data.frame `traitmat` contains the values for (1) specific leaf area (ratio of fresh leaf area to leaf dry mass, SLA), (2) canopy height (CH) and (3) seed mass (SM). Trait values were obtained from the LEDA trait database (Kleyer et al. 2008).

```r
# Give new name for traitmat with NAs
traitmat_NA <- detectionfilter::traitmat

# Correlation between traits
cor(traitmat_NA$sla, traitmat_NA$ch, use = "complete.obs")
```

```
## [1] -0.1793639
```

```r
cor(traitmat_NA$sla, traitmat_NA$sm, use = "complete.obs")
```

```
## [1] -0.06986081
```

```r
cor(traitmat_NA$ch, traitmat_NA$sm, use = "complete.obs")
```

```
## [1] 0.3060805
```

```r
# Median and range of trait values
apply(traitmat_NA, 2, median, na.rm = TRUE)
```

```
##        sla         ch         sm
## 22.0700000  0.3250000  0.8922222
```

```r
apply(traitmat_NA, 2, range, na.rm = TRUE)
```

```
##         sla     ch       sm
## [1,]   2.59  0.004     0.00
## [2,] 150.55 65.000 10611.95
```

```r
# Proportion of species with missing values for functional traits
apply(traitmat_NA, 2, function(x) mean(!is.na(x)))
```

```
##       sla        ch        sm
## 0.6468552 0.7986151 0.7022504
```

In our data set, up to 35% of values were missing for any particular trait. We therefore imputed missing values using random forest estimation implemented in the R package `missForest` (Stekhoven & Buhlmann 2012).

```
# Nonparametric missing value imputation using random forest
traitmat <- missForest(as.matrix(traitmat_NA), maxiter = 10)$ximp
traitmat <- as.data.frame(traitmat)
```

To test the imputation, we calculated the mean trait value of the species in the community once with the traitmatrix containing NAs and once with the traitmatrix with imputed values instead of NAs.

```
# Merge observation from two visits
commat_obs <- apply(plantsBDM$y, c(1,2), max)

# Specific leaf area
cor(apply(commat_obs==1, 1, function(x) mean(traitmat$sla[x], na.rm = TRUE)),
    apply(commat_obs==1, 1, function(x) mean(traitmat_NA$sla[x], na.rm = TRUE)))
```

```
## [1] 0.9938858
```

```
# Canopy height
cor(apply(commat_obs==1, 1, function(x) mean(traitmat$ch[x], na.rm = TRUE)),
    apply(commat_obs==1, 1, function(x) mean(traitmat_NA$ch[x], na.rm = TRUE)))
```

```
## [1] 0.99875
```

```
# Seed mass
cor(apply(commat_obs==1, 1, function(x) mean(traitmat$sm[x], na.rm = TRUE)),
    apply(commat_obs==1, 1, function(x) mean(traitmat_NA$sm[x], na.rm = TRUE)))
```

```
## [1] 0.9987595
```

The imputation seem not to strongly alter community composition as community means calculated with the traitmatrix containing NAs were strongly correlated with the community means calculated from traitmatrix with imputed trait values instead of NAs (all $r > 0.99$).

For all further analyses trait values were log scaled (Westoby 1998), then normalized to a mean of 0 and standard deviation of one, allowing comparison among traits (Schielzeth 2010).

```
traitmat$sla <- scale(log(traitmat$sla))[,1]
traitmat$ch <- scale(log(traitmat$ch))[,1]
traitmat$sm <- scale(log(traitmat$sm+0.1))[,1]
traitmat_NA$sla <- scale(log(traitmat_NA$sla))[,1]
traitmat_NA$ch <- scale(log(traitmat_NA$ch))[,1]
traitmat_NA$sm <- scale(log(traitmat_NA$sm+0.1))[,1]
```

Estimating detection-corrected meta-community

To estimate the true occurrence of all species `k` at plot `i` we applied a single season occupancy model to all species separately (MacKenzie et al. 2002). Note that although fieldwork was conducted from 2010 to 2014 each plot was visited only during a single year. That is why a single season occupancy model seemed a sensible choice. Further note that during a single year, the surveyed plots were visited twice. Repeated visits of plots during a single season is a prerequisite to apply single season occupancy models that can account for imperfect detection (MacKenzie 2002).

First we transformed the predictor variables to be small values that are distributed around 0. The reason for this is mainly computational.

```
# Standardize Julian dates and elevation
ele <- plantsBDM$elevation
ele <- (ele - 1000)/100
```

```
dates <- plantsBDM$dates
dates <- (dates - 200) / 7
```

Now, we are ready to apply the occupancy model for each species separately. A convenient way to do so would be to apply a for-loop over all species. However, for-loops in R are usually not very efficient. That is why we aimed to use the `lapply()` function. To do so, we first had to bundle all the calculations that should be applied separately to each species in a single function (`f.speccalc`):

1. The function `unmarkedFrameOccu()` of the package `unmarked` is used to bundle the data needed for the single season occupancy model. These are the observations `y[i,j]` that contains 1 if the species was observed in plot `i` during visit `j`, or 0 otherwise. Note that `plantsBDM$y` is three dimensional because it contains the observations for all species. Further, the matrix `dates[i,j]` contains the Julian day when visit `j` was conducted to plot `i` and the vector `ele[i]` that contains the elevation for each plot `i`.
2. The function `occu()` of the package `unmarked` is used to apply the single-season occupancy model to the data. Note that to the right of the first ~ the predictors for the detection probability are added and to the right of the second ~ the predictors for occurrence are added. Since detection probability is likely to depend on phenology, and because phenology changes with elevation, we used the survey date (linear and quadratic terms) as well as their interactions with elevation as predictors for detection probability (Chen et al. 2013). Further, because of the large elevational gradient, we incorporated the linear and quadratic terms of elevation of the plots as predictors for occurrence (Chen et al. 2013).
3. Some species were observed only at very few plots that leads to an error of the `occu()` function because the algorithm did not converge. In order to avoid that the calculation stops at this point we put the `try()` function around these calculations.
4. We estimated the average detectability of a species (`Pi[k]`), , which is independent of the true distribution of the species, by assuming that the species was present on all plots. We averaged the probabilities of detecting the species during at least one of the two surveys across all plots.
5. A single season occupancy model is a hierarchical model in the form of `f(y[i,j]|z[i])` were `z[i]` is the true species presence at plot `i`. The function `ranef()` estimates posterior distributions of the `z` using empirical Bayes methods. Finally, we use the function `bup()` to extract the mode of the posterior probability. Both functions are from the package `unmarked`.

```
# Function that is doing all the calculations per species
f.speccalc <- function(k) {
  # Bundle data
  d <- unmarkedFrameOccu(y = plantsBDM$y[,k,], obsCovs = list(dates = dates),
                         siteCovs = data.frame(ele = ele))
  Pi <- NA
  z <- rep(NA, nplots)

  # Use try() to avoid that calculations stops when something gets wrong for one species
  try({
    # Apply single season occupancy model
    res <- occu(~ (dates + I(dates^2)) * ele ~ ele + I(ele^2), data = d, se = TRUE)

    # Calculate species' average detection probability
    p <- predict(res, type = 'det')$Predicted
    Pi <- mean(1-((1-p[1:nplots])*(1-p[(nplots+1):(2*nplots)])))

    # Mode of posterior probability for species occurrence using empirical Bayes
    z <- bup(unmarked::ranef(res), stat = "mode")
  })

  # Return results
```

```r
  list(Pi = Pi, z = z)
}
```

We now are ready to apply the analyses for each species separately. However, to reduce the time needed to run the analyses we decided to use a function analogous to the `lapply()` function that is able to run the calculations for species in parallel. We thus used the `parLapply` function of the package `parallel`.

```r
# Get the number of cores on the computer used for the calculations
no_cores <- detectCores()

# Run the analyses for each species
cl <- makeCluster(no_cores, type="FORK")
resoccu <- parLapply(cl, 1:nspec, f.speccalc)
stopCluster(cl)

# Später löschen
save.image("deteccor.RDATA")
```

Finally, we need to bundle the results in a vector `P` that contains the average detection probability per species and in the detection-corrected meta-community matrix `z[i,k]`.

```r
P <- as.numeric(sapply(resoccu, function(x) x$Pi))
z <-  sapply(resoccu, function(x) x$z)
(nspec_with_res <- sum(apply(z, 2, sum) > 0, na.rm = TRUE))
```

```
## [1] 1635
```

The `occu()` function successfully calculated average detection probability and true occurrences for 1635 of the totally 1733 species. Thus, 98 (5.7%) species had to be removed from analyses. In average these removed species were observed only at 1.1 plots.

Are traits correlated with species' detection probability?

We define detection filtering as any methodological process that selects the species that are observed from the local community depending on the expression of their functional trait. This implies that the species' detection probability is related to the trait values of the species. To test for this, we applied a linear model with the logit transformed average species' detection probability as the dependent variable and the specific leaf area, the canopy height and the seed mass as predictor variables.

```r
# Prepare data.frame with all data for linear model
d <- traitmat
d$P <- P
d$nobs <- apply(commat_obs, 2, sum)
d <- d[!is.na(d$P),]

# Linear model for all species
round(summary(lm(qlogis(P) ~ sla + ch + sm, data = d))$coef, 3)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.307      0.047  27.577    0.000
## sla           -0.080      0.048  -1.688    0.092
## ch             0.205      0.056   3.654    0.000
## sm            -0.035      0.055  -0.631    0.528
```

The intercept corresponds to the detection probability at the logit scale of a plant species with average trait expression. On the probability scale this corresponds to a detection probability of 0.79. If the functional trait is increased by one standard deviation detection probability increases to 0.82 for canopy height, to 0.81 for seed mass, and to 0.79 for specific leaf area.

Overall, these results remain rather stable if the linear model is applied to the 50% most common species.

```
# Proportion of species occuring on at least 23 plots
round(mean(d$nobs >= 23), 3)
```

```
## [1] 0.503
```

```
# Linear model only for common species
round(summary(lm(qlogis(P) ~ sla + ch + sm, data = d[d$nobs >= 23, ]))$coef, 3)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.345      0.057  41.411    0.000
## sla           -0.031      0.053  -0.582    0.561
## ch             0.249      0.061   4.108    0.000
## sm             0.105      0.060   1.737    0.083
```

Community composition and diversity along elevational gradient

We calculated functional composition (i.e. single trait measures such as community mean) and diversity (i.e. multi trait measures) from observed (`commat`) and detection corrected meta-community (`z`). Since we were not able to estimate the true occurrence for all species, we have to make sure that observed and detection corrected meta-community contains exactly the same species.

```
# Remove species for which occupancy model could not be applied
commat <- commat_obs[, apply(z, 2, function(x) sum(is.na(x))==0)]
z <- z[, apply(z, 2, function(x) sum(is.na(x))==0)]
```

Now, both `commat` and `z` contain 362 rows for each plot and 1635 columns for each species for which true occurrences could be estimated.

Detection filtering and community composition

To estimate community functional composition, we calculated for each community the mean trait value across all species and did this separately for each of the three functional traits and both for the observed meta-community (`commat`) and the detection corrected meta-community (‘z’).

```
# Specific leaf area (SLA)
f.sla <- function(x) mean(traitmat$sla[as.logical(x)])
CM.sla.obs <- apply(commat, 1, f.sla)
CM.sla.cor <- apply(z, 1, f.sla)
dif.sla <- abs(CM.sla.obs - CM.sla.cor) > 0.5*sd(CM.sla.obs)

# Canopy height (CH)
f.ch <- function(x) mean(traitmat$ch[as.logical(x)])
CM.ch.obs <- apply(commat, 1, f.ch)
CM.ch.cor <- apply(z, 1, f.ch)
dif.ch <- abs(CM.ch.obs - CM.ch.cor) > 0.5*sd(CM.ch.obs)

# Seed mass (SM)
f.sm <- function(x) mean(traitmat$sm[as.logical(x)])
CM.sm.obs <- apply(commat, 1, f.sm)
CM.sm.cor <- apply(z, 1, f.sm)
dif.sm <- abs(CM.sm.obs - CM.sm.cor) > 0.5*sd(CM.sm.obs)
```

Bias due to imperfect detection was larger than half a standard deviation of observed community means in 13.5% of communities for SLA, in 22.7% of communities for canopy height and in 36.7% of communities for seed mass. Nonetheless, correlation between community means of observed and detection corrected communities was rather high (SLA: 0.982, CH: 0.984, SM: 0.961).

In the following we define the the function `f.plotFD()` that is plotting the observed community means along the elevational gradient. We consider a difference between observed and detection-corrected mean as relevant if it is larger than half the standard deviation of the community means along the entire gradient. We therefore show communities with larger differences than half a standard deviation in colours.

```
f.plotFD <- function(d, title, tylab, ymin = -0.75,
                     ymax = 0.5, tticks = 0.25, yleg = c(-0.475, -0.59)) {
  ele <- 400:2500
  xax <- seq(250, 2750, 250)
  tcol <- brewer.pal(8, "Set1")
  plot(NA, ylim = c(ymin,ymax), xlim = c(250,2750), axes=F, xlab = "", ylab = "")
  farbe <- rep("grey80", nplots)
  farbe[d$dif & d$obs < d$cor] <- tcol[1]
  farbe[d$dif & d$obs > d$cor] <- tcol[2]
  points(d$elevation, d$obs, pch = 21, cex = 0.5, col = "grey50")
  points(d$elevation, d$obs, pch = 16, cex = 0.5, col = farbe)
  pred <- predict(gam(obs ~ s(elevation), data = d),
                  newdata = data.frame(elevation=ele), type = "response")
  points(ele, pred, ty = "l", lty = 2)
  pred <- predict(gam(cor ~ s(elevation), data = d),
                  newdata = data.frame(elevation=ele), type = "response")
  points(ele, pred, ty = "l")
  axis(side=1, at = xax, labels = rep("", length(xax)), pos=ymin)
  mtext(seq(500,2500,500), 1, at = seq(500,2500,500), cex = 0.7)
  axis(side = 2, at = round(seq(ymin, ymax, tticks), 2), pos = 250, las = 1)
  mtext(text = tylab, side = 2, line = 3, cex = 0.7)
  mtext(text = title, side = 3, at = -420, line = 1.5, cex = 0.8, adj = 0)
  legend(350, yleg[1], c("     underestimated values",
                      "     overestimated values"), col = tcol[1:2],
         bty = "n", cex = 0.8, pch = c(16,16), pt.cex = 1, y.intersp=0.8)
  legend(300, yleg[2], c("prediction from observed communities",
                       "detection corrected prediction"),
         bty = "n", cex = 0.8, lty = c(2,1), y.intersp=0.8)
}
```

Now we apply the function to the date of the three traits separately.

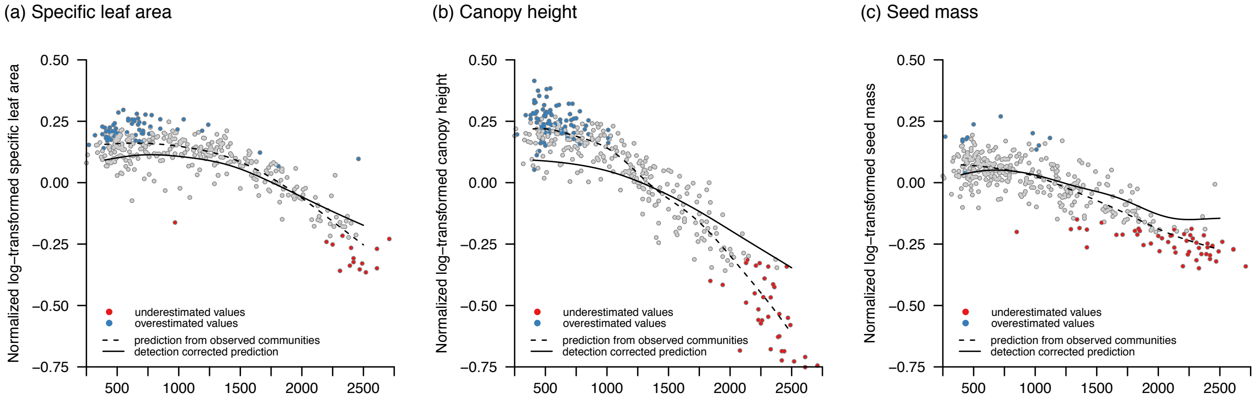

Fig. 1: Change of community means (CMs) of log-transformed and normalized (z-score) trait values along the elevational gradient for the three functional traits (a) specific leaf area, (b) canopy height and (c) seed mass. Points give CMs of the 362 observed communities. Coloured points indicate communities where imperfect detection affected estimates of CMs by more than half a standard deviation of CMs along the gradient (red points: observed CMs are lower than detection-corrected CMs; blue points: observed CMs are larger than detection-corrected CMs). The lines represent the predictions from the generalized additive model (GAM) applied to the observed communities (dotted line) and to the detection-corrected communities (solid line).

Community diversity along elevational gradient

We quantify functional diversity for each community as the multivariate convex hull volume, i.e. functional richness (FRic), and as the mean nearest neighbour distance, using the Euclidean distance between species in multivariate trait space (Swenson & Weiser 2014).

```
# Functional trait space (FRic)
f.FRic <- function(x) convhulln(traitmat[as.logical(x),], "FA")$vol
FRic.obs <- apply(commat, 1, f.FRic)
FRic.cor <- apply(z, 1, f.FRic)
FRic.dif <- abs(FRic.obs - FRic.cor) > 0.5*sd(FRic.obs)

## Calculate distance matrix from trait matrix
dist <- as.matrix(dist(traitmat))

# Mean nearest neighbout distance (mnnd)
f.mnnd <- function(x) {
  sample.dis <- dist[as.logical(x),as.logical(x)]
  mean(apply(sample.dis, 1, function(x) min(x[x>0])))
}
mnnd.obs <- apply(commat, 1, f.mnnd)
mnnd.cor <- apply(z, 1, f.mnnd)
mnnd.dif <- abs(mnnd.obs - mnnd.cor) > 0.5*sd(mnnd.obs)

# Taxonomic richness (species richness SR)
SR.obs <- apply(commat, 1, sum)
SR.cor <- apply(z, 1, sum)
SR.dif <- abs(SR.obs - SR.cor) > 0.5*sd(SR.obs)
```

In average only about 66% of the species in of detection corrected communities were actually observed.

Bias due to imperfect detection was larger than half a standard deviation of estimates across communities in

85.1% of communities for functional richness, and in 95.6% of communities for functional packing. Nonetheless, correlation between estimates of observed and detection corrected communities was rather high (FRic: 0.929, mnnd: 0.929).

We again use the `f.plotFD()` function to make the plots for functional richness, functional packing and species richness.

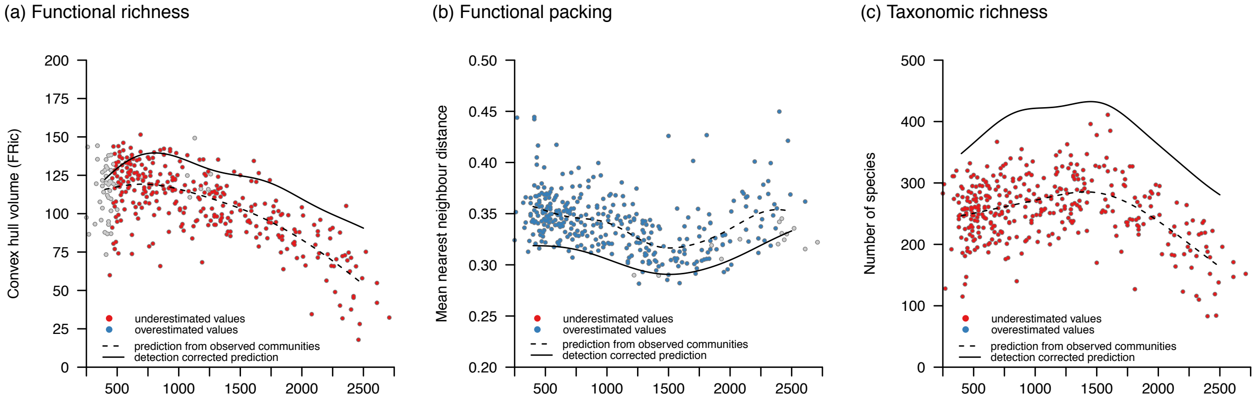

Fig. 2: Changes in (a) functional richness (convex hull volume of the three functional dimensions' specific leaf area, canopy height and seed mass) (b) functional packing (mean nearest neighbour distance) and (c) taxonomic diversity (number of species) along the elevational gradient. Points give the estimates of the 362 observed communities. Coloured points indicate communities where imperfect detection affected estimates by more than half a standard deviation of estimates along the gradient (red points: observed estimates are below the detection-corrected estimates; blue points: observed estimates are above the detection-corrected estimates). The lines represent the predictions from the generalized additive model (GAM) applied to the observed communities (dotted lines) and to the detection-corrected communities (solid line).

Referenzen

Chen, G.; Kéry, M.; Plattner, M.; Ma, K.; Gardner, B., 2013. Imperfect detection is the rule rather than the exception in plant distribution studies. - Journal of Ecology 101: 183–191.

Kleyer, M.; Bekker, R. M.; Knevel, I. C.; Bakker, J. P.; Thompson, K.; Sonnenschein, M., . . . Peco, B., 2008. The LEDA traitbase: a database of life-history traits of the Northwest European flora. - Journal of Ecology 96: 1266-1274.

MacKenzie, D. I.; Nichols, J. D.; Lachman, G. B.; Droege, S.; Royle, J. A.; Langtimm, C. A., 2002. Estimating site occupancy rates when detection probabilities are less than one. - Ecology 83: 2248-2255.

Schielzeth, H., 2010. Simple means to improve the interpretability of regression coefficients. - Methods in Ecology and Evolution 1: 103-113.

Stekhoven, D. J.; Buhlmann, P., 2012. MissForest-non-parametric missing value imputation for mixed-type data. - Bioinformatics 28: 112-118.

Swenson, N. G.; Weiser, M. D., 2014. On the packing and filling of functional space in eastern North American tree assemblages. - Ecography 37: 1056-1062.

Westoby, M., 1998. A leaf-height-seed (LHS) plant ecology strategy scheme. - Plant and Soil 199: 213-227.