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

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

2017-08-22

Contents

Introduction

Prerequisite

This vignette presents analyses from Roth, Allan, Pearman and Amrhein (under revision) "Functional ecology and imperfect detection of species". The latest version of the package `detecionfilter` is required to run the code and should be downloaded from github (make sure you have the package `devtools` installed).

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

The presented code depends on the following packages.

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

Data from the Swiss Biodiversity monitoring (BDM)

The `detectionfilter` package contains a data set of plant surveys from the Swiss biodiversity monitoring (`plantsBDM`) program. The array `y[i,k,j]` contains the detection/non-detection data of `i=1,2,...,362`

$1km^2$ plots, the `k=1,2,...,1733` observed plant species and the `j=1,2` visits. `plantsBDM` also contains the median elevation for each plot (`elevation`), as well the dates (Julian day) of each visit to the plots (`dates`).

We call the assemblages of species that occur at a plot a *community*. In the case of `plantsBDM` we speak of `meta-community` data because the data set contains observations of 362 communities. Further, note that the array `y[i,k,j]` contains information on whether or not a species was *observed*. When we refer to species *occurrence*, we mean the true occurrence state of the species, which is not directly observable during a survey because of imperfect detection of 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
```

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

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

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

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

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

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

Trait data

The `detectionfilter` package contains values for three functional traits of the 1733 species in the data.frame `traitmat`, as 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). The trait values were obtained from the LEDA trait database (Kleyer et al. 2008).

```
# 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
```

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

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

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

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

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

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

```
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
```

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

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

```
# Proportion of records of species with missing values of functional traits
nrec <- apply(apply(plantsBDM$y, c(1,2), max), 2, sum)
apply(traitmat_NA, 2, function(x) sum(as.integer(!is.na(x)) * nrec) / sum(nrec))
```

```
##       sla        ch        sm
## 0.8796910 0.9220203 0.8732817
```

As shown above, originally up to 35% of the detected species had missing values for any particular trait. We therefore imputed missing values using random forest estimation as implemented in the R package `missForest` (Stekhoven & Buhlmann 2012).

```
# Nonparametric missing value imputation using random forest
set.seed(123)
traitmat <- missForest(as.matrix(traitmat_NA))$ximp
traitmat <- as.data.frame(traitmat)
```

To test the imputation, we calculated the mean trait value in each community of detected species, once using the species trait matrix containing NAs and once with the imputed values instead of NAs. For each trait we then examined the correlation between the two sets of community mean values.

```
# 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] 0.9945322
```

```
# 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] 0.9987346
```

```
# 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] 0.9985327
```

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). Thus, the imputation did not strongly alter the trait characteristics of communities.

For all further analyses trait values were log transformed (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 `k` species at all `i` plots, we applied a single season occupancy model to data for each species separately (MacKenzie et al. 2002). Note that although fieldwork was conducted from 2010 to 2014, each plot was visited in only one of those years. Thus, a single season occupancy model seemed a sensible choice. Further, surveyed plots were visited twice during a single year. Repeated visits of plots during a single period, in which unchanging occurrence is assumed, is a prerequisite for applying single season occupancy models to account for imperfect detection (MacKenzie 2002).

We first transformed the predictor variables to be small values to facilitate computation.

```
# Standardize Julian dates and elevation
ele <- plantsBDM$elevation
ele <- ele/1000
dates <- plantsBDM$dates
dates <- (dates - 200) / 7
```

We analysed species with at least four detections because otherwise the algorithm implementing the single season occupancy model failed to converge consistently. We thus remove the 437 species that are 25.2% of all recorded species.

```
# Store full trait table sets with all species
traitmat_all <- traitmat

# Select data from species with at least four observations
selspec <- apply(commat_obs, 2, sum) > 3
commat <- commat_obs[, selspec]
traitmat_NA <- traitmat_NA[selspec, ]
traitmat <- traitmat[selspec, ]
y <- plantsBDM$y[, selspec, ]
```

We now apply the occupancy model to each species. One could do this in a for-loop over the species. However, for-loops in R are generally inefficient. We implement an alternative, using the `lapply()` function. We start by bundling all the calculations to be applied to each species into 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`. The vector `ele[i]` 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 the predictors for detection probability are to the right of the first `~` and the predictors

for occurrence are to the right of the second `~`. Since detection probability is likely to depend on plant phenology, we used survey date (linear and quadratic terms) as predictors for detection probability (Chen et al. 2013). Further, because of the large elevational gradient, we incorporated the linear and quadratic terms of plot elevation as predictors for species occurrence (Chen et al. 2013).

3. 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.

4. A single season occupancy model is a hierarchical model in the form of `f(y[i,j]|z[i])` where `z[i]` is the true species presence at plot `i`. The function `ranef()` estimates posterior distributions of `z` using empirical Bayesian 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 = y[,k,], obsCovs = list(dates = dates),
                         siteCovs = data.frame(ele = ele))

  # Apply single season occupancy model
  res <- occu(~ dates + I(dates^2) ~ ele + I(ele^2), data = d, se = FALSE)

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

We now run the analyses for each species separately. Notably, to reduce computation time we use the `parLapply()` function from the package `parallel`, in order to run the calculations in parallel. It takes around 10 minutes (depending on the computer) to complete the calculations for all species.

```
# 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:sum(selspec), f.speccalc)
stopCluster(cl)

# Save image at this point
save.image("deteccor.RDATA")
```

Finally, we bundle the results into two objects: a vector `P` containing the average detection probability per species, and the detection-corrected meta-community matrix `z[i,k]` containing the (estimated) true occurrence of species `k`at site `i`.

```
P <- as.numeric(sapply(resoccu, function(x) x$Pi))
z <-  sapply(resoccu, function(x) x$z)
ncol(z)
```

```
## [1] 1296
```

Detection filtering

Predictors of 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 apply a linear model with the logit-transformed ($log(\frac{p}{1-p})$) average probability of species detection as the dependent variable and the specific leaf area, the canopy height and the seed mass as predictor variables. Furthermore, widespread species are often locally common, making them easier to detect than sparsely distributed species. We used the estimated number of occupied plots per species as another predictor of the probability of species detection. Finally, we included the average elevation of occupied plots as a predictor to test whether detection probability varies with elevation of occurrence.

```
# Prepare data.frame with all data for linear model
d <- traitmat_NA
d$P <- P
d$nobs <- apply(commat, 2, sum)
d$nobs_sd <- as.vector(scale(d$nobs))

# Add for each species the number of occupied plots (and standardize)
d$nocc <- apply(z, 2, sum)
d$nocc_sd <- as.vector(scale(log(d$nocc)))

# Add for each species the average elevation (and standardize)
d$meanel <- apply(z, 2, function(x) mean(plantsBDM$elevation[x==1]))
d$meanel_sd <- as.vector(scale(log(d$meanel+1)))

# Linear model
mod <- lm(qlogis(P) ~ sla + ch + sm + nocc_sd + meanel_sd, data = d)
round(summary(mod)$coef, 3)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.795      0.082  34.203    0.000
## sla           -0.164      0.080  -2.048    0.041
## ch             0.218      0.092   2.380    0.018
## sm             0.387      0.083   4.654    0.000
## nocc_sd        0.609      0.077   7.893    0.000
## meanel_sd      0.564      0.097   5.821    0.000
```

The intercept corresponds to the species' 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 0.94. Increasing a functional trait value by one standard deviation increases detection probability to 0.95 for canopy height, to 0.96 for seed mass, and decreases the probability to 0.93 for specific leaf area.

The number of days between the first and second visit may indicate the proportion of the vegetation period sampled by the surveys. We asked whether the number of days between first and second visit changes with elevation.

```
difdays <- plantsBDM$dates$Dat2 - plantsBDM$dates$Dat1
round(summary(lm(difdays ~ I(plantsBDM$elevation/100)))$coef, 3)
```

```
##                             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)                  77.833      2.480  31.384    0.000
## I(plantsBDM$elevation/100)   -0.023      0.196  -0.119    0.905
```

On average 77.6 days separated the first and second visits to the plots. This difference was fairly constant along the elevational gradient.

Detection filtering at the community level

Correction of community composition for imperfect species detection (to estimate detection-corrected communities') shows that on average about 94.8% of occurring species are actually observed. The actual proportion depends on community elevation (Fig. 1a). For each species we calculate the mean elevation of its occurrences ('species mean elevation'), and we average these values for the species occurring in each community. Then considering only the subset of species in each community that have unobserved occurrences (the overlooked species), we again average the values of species mean elevation. We plot both these sets of average elevations against community elevation (Fig. 1b). We then calculate the number of occurrences of each species and use these to generate average values of species occurrence for each community. Again we consider only the overlooked species in each community, and generate the average number of occurrences of these species. Both these sets of values are plotted against community elevation (Fig. 1c).

```
# Species richness SR
SR.obs <- apply(commat, 1, sum)
SR.cor <- apply(z, 1, sum)

# Mean elevation
tmp <- apply(commat, 2, function(x) mean(plantsBDM$elevation[x==1]))
meanel.obs <- apply(commat, 1, function(x) mean(tmp[x==1]))
meanel.cor <- apply(z != commat, 1, function(x) mean(tmp[x==1]))

# Number of occurrences
tmp <- apply(z, 2, sum)
occ.obs <- apply(commat, 1, function(x) mean(tmp[x==1]))
occ.cor <- apply(z != commat, 1, function(x) mean(tmp[x==1]))
```

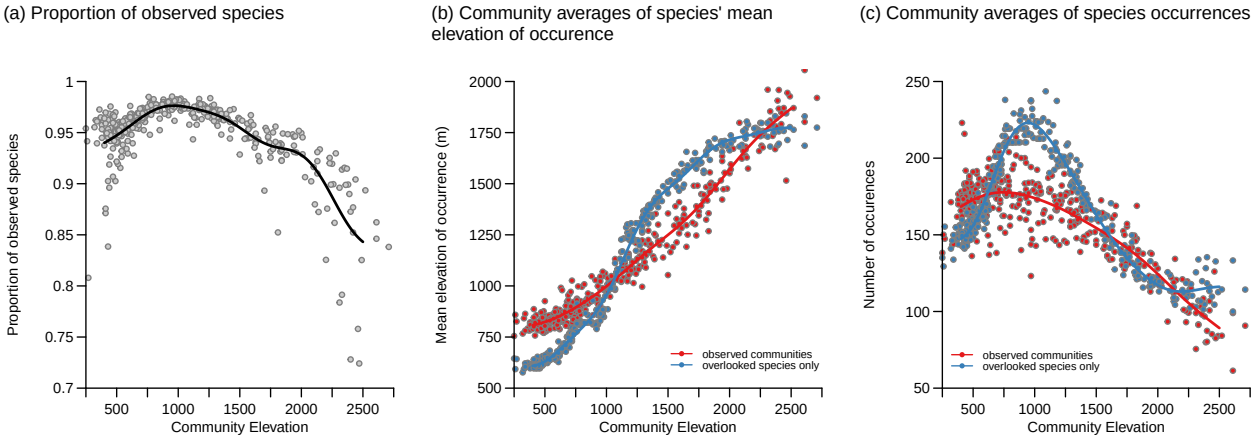

Fig. 1: (a) Change of the observed proportion of occurring species plotted against community elevation. (b) Mean elevation of species occurrence averaged for observed species (red points) and species that were estimated to occur in a community but that were not detected (i.e. overlooked species, blue points), plotted against community elevation. (c) Number of occurrences per species averaged for observed species (red points) and overlooked species (blue points), plotted against community elevation. Smoothed curves are predictions from generalized additive models (GAMs).

Lets look at a few of the species that are responsible for these patterns.

```r
# Species that remain most often undetected at communities below 750m
undet <- apply(z[plantsBDM$elevation < 750, ] != commat[plantsBDM$elevation < 750, ], 2, sum)
row.names(traitmat)[order(undet, decreasing = TRUE)][1:2]
```

```
## [1] "1575" "2094"
```

```r
# Species that remain most often undetected at communities between 750m and 1250m
undet <- apply(z[plantsBDM$elevation > 750 & plantsBDM$elevation < 1250, ] !=
                 commat[plantsBDM$elevation > 750 & plantsBDM$elevation < 1250, ], 2, sum)
row.names(traitmat)[order(undet, decreasing = TRUE)][1]
```

```
## [1] "613"
```

In communities below 750m, the species that most often remain undetected were *Buglossoides arvensis* (species-ID: 1575) a weed of arable land and *Helianthus tuberosus* (species ID 2094) a currently spreading invasive species with late flowering. The species that most often remained undetected in communities between 750m and 1250m was *Descurainia sophia* (species ID 613), a ruderal of fields and dry rock faces.

Community composition and diversity along elevational gradient

In this chapter we will calculate 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`). Differences between measures from observed and detection-corrected communities should be due to detection filtering.

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 for both the observed meta-community (`commat`) and the detection-corrected meta-community (`z`).

We also calculate for each community whether the effect of detection filtering is relevant. We define an effect of detection filtering on community composition as being relevant if it is larger than the change of community composition we observe per 100m along the elevational gradient.

```r
# Specific leaf area (SLA)
f.sla <- function(x) mean(traitmat$sla[as.logical(x)])
CM.sla.obs <- apply(commat, 1, f.sla)
CM.sla.cor <- apply(z, 1, f.sla)
rel <- 100 * abs(lm(CM.sla.cor ~ plantsBDM$elevation)$coef[2])
dif.sla <- abs(CM.sla.obs - CM.sla.cor) > rel

# Canopy height (CH)
f.ch <- function(x) mean(traitmat$ch[as.logical(x)])
CM.ch.obs <- apply(commat, 1, f.ch)
CM.ch.cor <- apply(z, 1, f.ch)
rel <- 100 * abs(lm(CM.ch.cor ~ plantsBDM$elevation)$coef[2])
dif.ch <- abs(CM.ch.obs - CM.ch.cor) > rel

# 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)
```

```
rel <- 100 * abs(lm(CM.sm.cor ~ plantsBDM$elevation)$coef[2])
dif.sm <- abs(CM.sm.obs - CM.sm.cor) > rel
```

Bias due to detection filtering was relevant in 17.1% of communities for SLA, in 2.8% of communities for canopy height and in 31.5% of communities for seed mass. Nonetheless, correlation between community means of observed and detection-corrected communities was rather high (SLA: 0.994, CH: 0.999, SM: 0.996).

We define a function `f.plotFD()` to plot observed community means along the elevational gradient.

```
f.plotFD <- function(d, title, tylab, ymin = -0.8, leg.cex = 0.8, lxtext = 3,
                     ymax = 0.6, tticks = 0.2, yleg = c(-0.475, -0.59),
                     xleg = c(350, 300), confint = FALSE) {
  ele <- 400:2500
  xax <- seq(250, 2750, 250)
  if(!confint) tcol <- brewer.pal(8, "Set1")
  if(confint) {
    tcol <- brewer.pal(4, "Paired")
    tcol <- paste0(tcol, c("50", "99", "50", 99))
  }
  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]
  if(!confint) {
    points(d$elevation, d$obs, pch = 21, cex = 0.5, col = "grey50")
    points(d$elevation, d$obs, pch = 16, cex = 0.5, col = farbe)
  }
  if(confint) {
    nsim <- 1000
    tmp <- array(0, dim = c(length(ele), nsim))
    for(s in 1:nsim) {
      tmp[, s] <- predict(gam(obs ~ s(elevation),
                              data = d[sample(1:nrow(d), nrow(d), replace = TRUE),]),
                          newdata = data.frame(elevation=ele), type = "response")
    }
    polygon(c(ele, rev(ele)),
            c(apply(tmp, 1, quantile, probs = 0.025),
              rev(apply(tmp, 1, quantile, probs = 0.975))), col = tcol[1], border = tcol[1])
    tmp <- array(0, dim = c(length(ele), nsim))
    for(s in 1:nsim) {
      tmp[, s] <- predict(gam(cor ~ s(elevation),
                              data = d[sample(1:nrow(d), nrow(d), replace = TRUE),]),
                          newdata = data.frame(elevation=ele), type = "response")
    }
    polygon(c(ele, rev(ele)),
            c(apply(tmp, 1, quantile, probs = 0.025),
              rev(apply(tmp, 1, quantile, probs = 0.975))), col = tcol[3], border = tcol[3])
  }
  pred <- predict(gam(obs ~ s(elevation), data = d),
                  newdata = data.frame(elevation=ele), type = "response")
  points(ele, pred, ty = "l", lty = 2, col = ifelse(confint, tcol[2], "black"))
  pred <- predict(gam(cor ~ s(elevation), data = d),
                  newdata = data.frame(elevation=ele), type = "response")
  points(ele, pred, ty = "l", col = ifelse(confint, tcol[4], "black"))
  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 = seq(ymin, ymax, tticks), pos = 250, las = 1,
       labels = rep("", length(seq(ymin, ymax, tticks))))
  mtext(round(seq(ymin, ymax, tticks), 2), 2, at = seq(ymin, ymax, tticks),
        cex = 0.7, las = 1)
  mtext(text = "Community Elevation", side = 1, line = 1, cex = 0.7)
  mtext(text = tylab, side = 2, line = lxtext, cex = 0.7)
  mtext(text = title, side = 3, at = -420, line = 1.5, cex = 0.8, adj = 0)
  if(!confint) {
  legend(xleg[1], yleg[1], c("     underestimated values",
                             "     overestimated values"), col = tcol[1:2],
         bty = "n", cex = leg.cex, pch = c(16,16), pt.cex = 1, y.intersp=0.8)
  }
  legend(xleg[2], yleg[2], c("prediction from observed communities",
                             "detection-corrected prediction"),
         bty = "n", cex = leg.cex, lty = c(2,1), y.intersp=0.8,
         col = ifelse(confint, tcol[c(2,4)], c("black", "black")))
}
```

Now we apply the function to the community means of each trait separately.

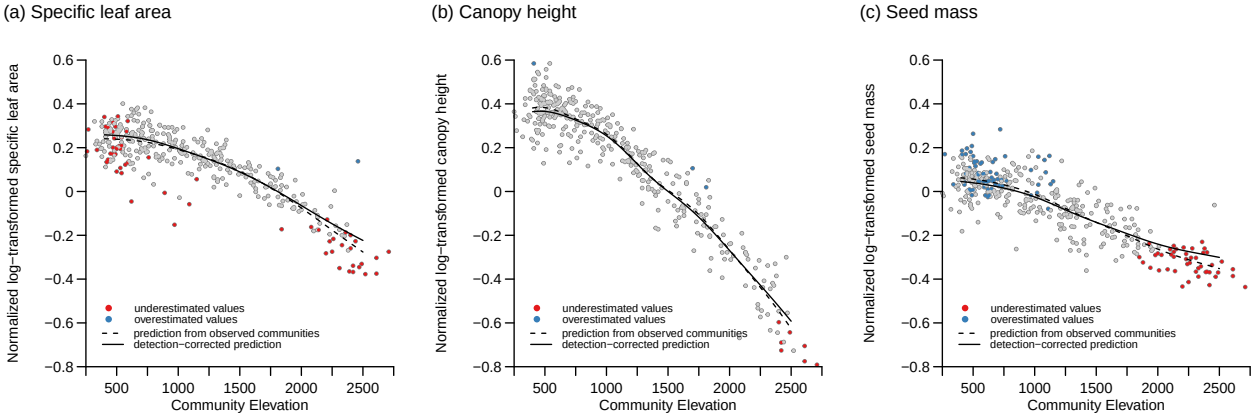

Fig. 2.1: Change of community means (CMs) of log-transformed and normalized (z-score) trait values along the elevational gradient for the three functional traits: (a) specific leaf area, (b) canopy height and (c) seed mass. Points give CMs of the 362 observed communities. Coloured points indicate communities where imperfect detection affected estimates of CMs more than the change of community composition we observe per 100m along the elevational 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).

Adding confidence intervals to Fig. 2.1 resulted in overloaded figures. We thus re-draw the figures with removing the CMs for the 362 communities and adding the bootstraped 95% confidence intervals for the predictions from the generalized additive model.

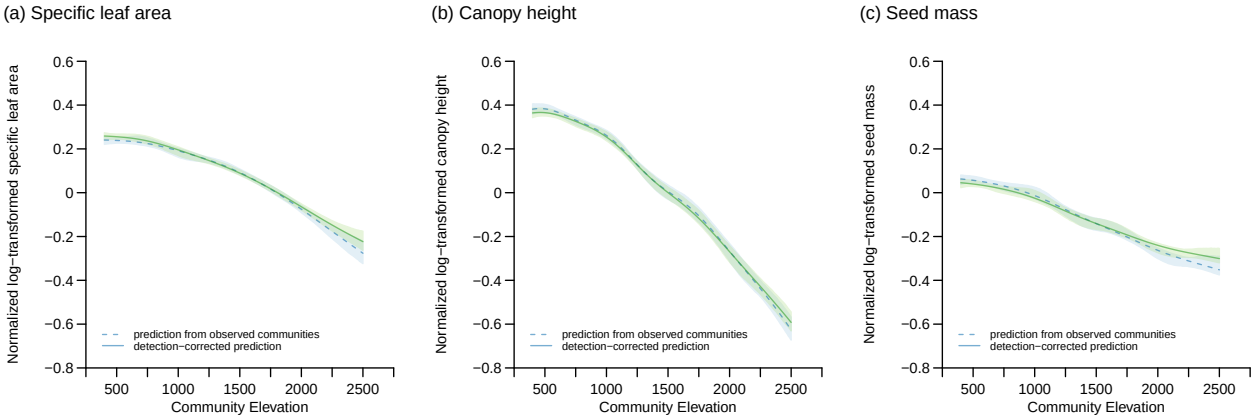

Fig. 2.2: Change of community means (CMs) of log-transformed and normalized (z-score) trait values along the elevational gradient for the three functional traits: (a) specific leaf area, (b) canopy height and (c) seed mass. The lines represent the predictions from the generalized additive model (GAM) applied to the observed communities (dotted line) and to the detection-corrected communities (solid line); the shaded area give the region of the bootstraped 95% confidence intervals of predictions from the GAM.

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")$vol
FRic.obs <- apply(commat, 1, f.FRic)
FRic.cor <- apply(z, 1, f.FRic)
rel <- 100 * abs(lm(FRic.obs ~ plantsBDM$elevation)$coef[2])
FRic.dif <- abs(FRic.obs - FRic.cor) > rel

## 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)
rel <- 100 * abs(lm(mnnd.obs ~ plantsBDM$elevation)$coef[2])
mnnd.dif <- abs(mnnd.obs - mnnd.cor) > rel

# Species richness SR
SR.obs <- apply(commat, 1, sum)
SR.cor <- apply(z, 1, sum)
rel <- 100 * abs(lm(SR.obs ~ plantsBDM$elevation)$coef[2])
SR.dif <- abs(SR.obs - SR.cor) > rel
```

Bias due to imperfect detection was relevant in 7.2% 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 high (FRic: 0.997, mnnd: 0.987).

While occupancy models adds unobserved species that were estimated to be occurring to the detection-corrected communities, they assume that false-positives (i.e. observations of species that do not occur) do not occur. Consequently taxonomic richness of observed communities is an underestimation of the true community species richness along the entire elevational gradient.

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

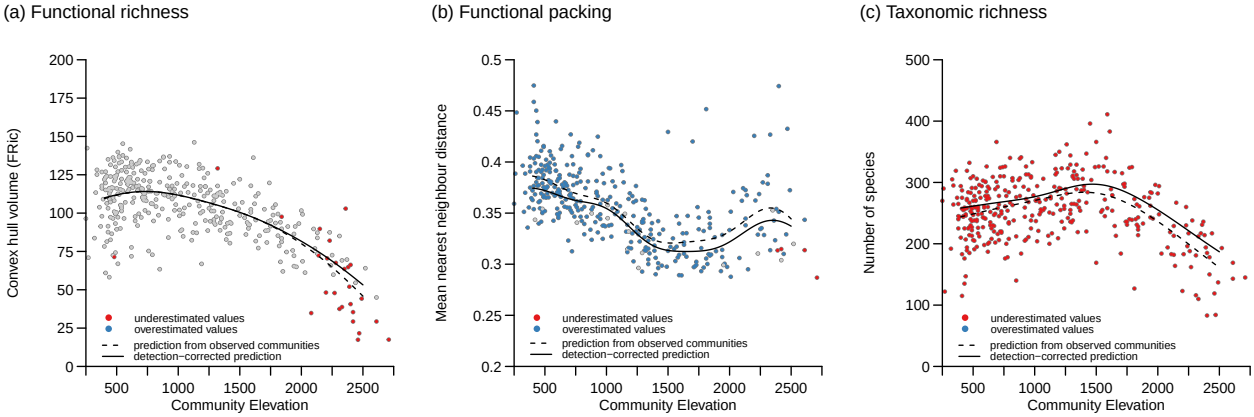

Fig. 3.1: Changes in (a) functional richness (convex hull volume of the three functional dimensions' specific leaf area, canopy height and seed mass) (b) functional packing (mean nearest neighbour distance) and (c) taxonomic diversity (number of species) along the elevational gradient. Points give the estimates of the 362 observed communities. Coloured points indicate communities where imperfect detection affected estimates more than the change of community diversity we observe per 100m along the elevational 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).

Again we make a separate figure with adding the bootstraped 95% confidences intervals and removing the estimates of the 362 communities.

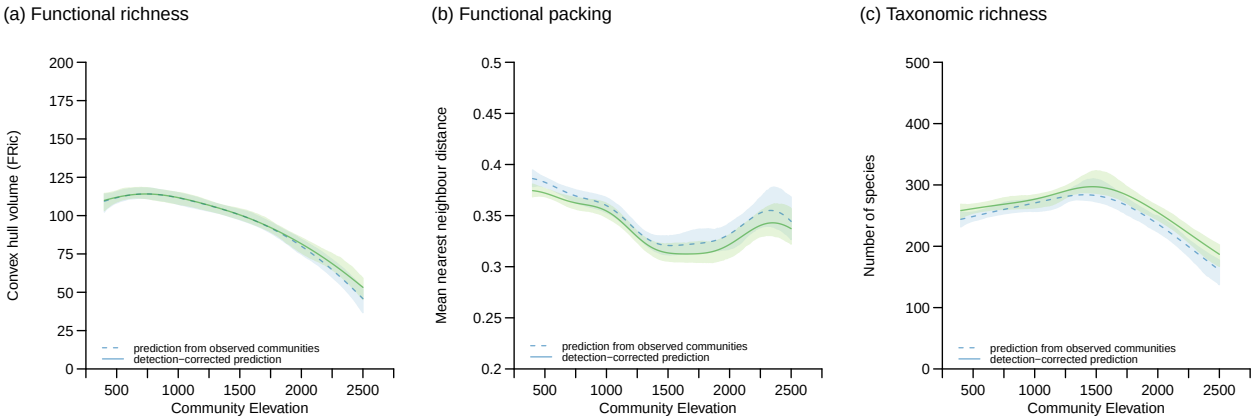

Fig. 3.2: Changes in (a) functional richness (convex hull volume of the three functional dimensions' specific leaf area, canopy height and seed mass) (b) functional packing (mean nearest neighbour distance) and (c) taxonomic diversity (number of species) along the elevational gradient. The lines represent the predictions from the generalized additive model (GAM) applied to the observed communities (dotted lines) and to the detection-corrected communities (solid line); the shaded area give the region of the bootstraped 95% confidence intervals of predictions from the GAM.

```r
nsim <- 10

# Function to get a random sample of
tsam <- function(x) {
  z[x, sample(which(z[x,] == 1), SR.cor[x] - SR.obs[x], replace = FALSE)] <- 0
  z[x,]
}

# Run simulation and calculate mean over all simulations
simFRic <- simmnnd <- array(0, dim = c(nrow(commat), nsim))
for(s in 1:nsim) {
  new.z <- t(sapply(1:nrow(commat), tsam))
  simFRic[, s] <- apply(new.z, 1, f.FRic)
  simmnnd[, s] <- apply(new.z, 1, f.mnnd)
}

FRic.cor <- apply(simFRic, 1, mean)
mnnd.cor <- apply(simmnnd, 1, mean)
```

Adding random species to a community will result in functional richness (FRic) to increase and mean nearest neighbour distance (mnnd) to decrease (i.e. species in communities become higher packed). We aimed to infer whether the differences in the functional packing we observed between detection-corrected and observed communities (see Fig. 3.1b and Fig 3.2b) could be explained simply because detection-corrected communities contain more species than observed communities. We thus randomly removed species in detection-corrected communities in a way that the number of species per community was the same for the subsampled detection-corrected communities and the observed communities. We repeated that 100 times and for all simulations compared the resulting measures of functional richness. The resulting figure (Fig. 3.3) suggests that the differences we observe in functional diversity measures between detection-corrected and observed communities is mainly due to the fact that detection corrected communities contain more species than observed communities and not necessarily due to differences in functional traits of missed species.

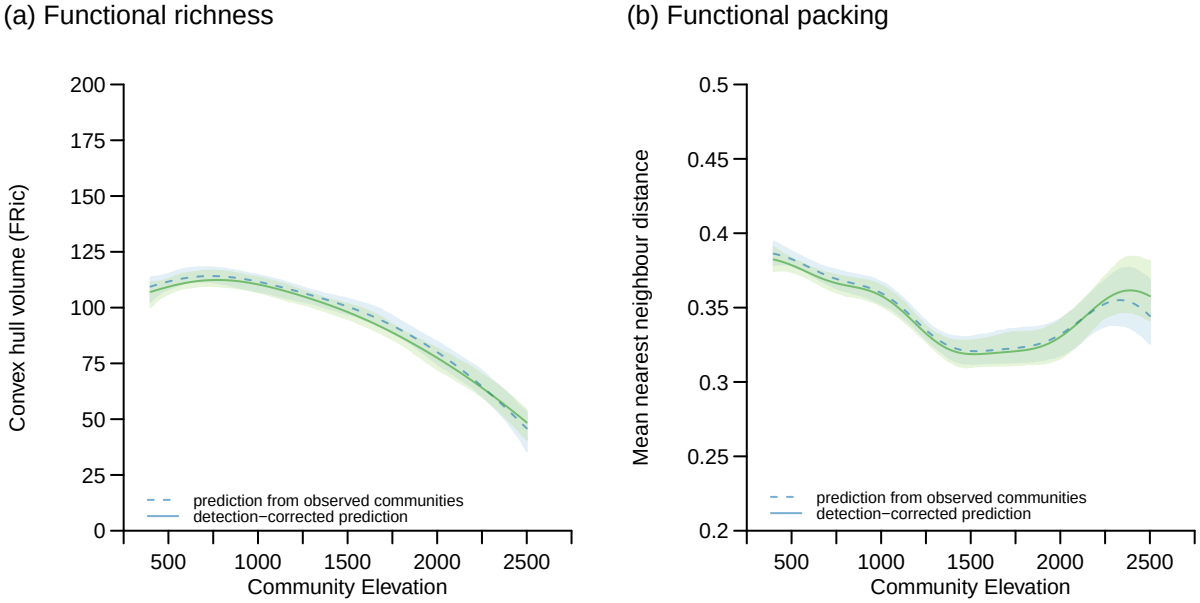

Fig. 3.3: Changes in (a) functional richness (convex hull volume of the three functional dimensions' specific leaf area, canopy height and seed mass) and (b) functional packing (mean nearest neighbour distance) along the elevational gradient. Points give the estimates of the 362 observed communities. 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); the shaded area give the region of the bootstraped 95% confidence intervals of predictions from the GAM. Detection-corrected estimates were obtained from randomly subsampled detection-corrected communities that contained the same number of species than the observed communities.

Effect of remooving rare species

For all analyses we removed species that were observed at less than 4 sites. This is because it was not able to apply the hierarchical models to some of these species. To infer whether this may have impacted our results we calculated all measures of functional composition and diversity with and without missing species and inferred how strong the measures were correlated.

```
# Specific leaf area (SLA)
f <- function(x) mean(traitmat$sla[as.logical(x)])
used <- apply(commat, 1, f)
f <- function(x) mean(traitmat_all$sla[as.logical(x)])
allobs <- apply(commat_obs, 1, f)
cor(used, allobs)
```

```
## [1] 0.9995347
```

```
# Canopy height (CH)
f <- function(x) mean(traitmat$ch[as.logical(x)])
used <- apply(commat, 1, f)
f <- function(x) mean(traitmat_all$ch[as.logical(x)])
allobs <- apply(commat_obs, 1, f)
cor(used, allobs)
```

```
## [1] 0.9997365
```

```
# Seed mass (SM)
f <- function(x) mean(traitmat$sm[as.logical(x)])
used <- apply(commat, 1, f)
f <- function(x) mean(traitmat_all$sm[as.logical(x)])
allobs <- apply(commat_obs, 1, f)
cor(used, allobs)
```

```
## [1] 0.9993308
```

```
# Functional trait space (FRic)
f <- function(x) convhulln(traitmat[as.logical(x),], "FA")$vol
used <- apply(commat, 1, f)
f <- function(x) convhulln(traitmat_all[as.logical(x),], "FA")$vol
allobs <- apply(commat_obs, 1, f)
cor(used, allobs)
```

```
## [1] 0.9959309
```

```
## Calculate distance matrix from complete trait matrix
dist_all <- as.matrix(dist(traitmat_all))

# Mean nearest neighbout distance (mnnd)
f <- function(x) {
  sample.dis <- dist[as.logical(x),as.logical(x)]
  mean(apply(sample.dis, 1, function(x) min(x[x>0])))
}
```

```
used <- apply(commat, 1, f)
f <- function(x) {
  sample.dis <- dist_all[as.logical(x),as.logical(x)]
  mean(apply(sample.dis, 1, function(x) min(x[x>0])))
}
allobs <- apply(commat_obs, 1, f)
cor(used, allobs)
```

```
## [1] 0.9979234
```

```
# Species richness SR
used <- apply(commat, 1, sum)
allobs <- apply(commat_obs, 1, sum)
cor(used, allobs)
```

```
## [1] 0.9984418
```

We found that measures of functional composition and diversity calculated from the observations of all 1733 species were very strongly correlated with measures calculated from the 1296 species with at least four observations (all $r > 0.995$). We therefore confident that removing species with less than four observations did not strongly bias our results.

To further infer the effect of removing rare species, we compared functional composition and diversity calculated from all observations and only from species with at least four observations. Similar to how we estimated detection effects on communities, we calculated the proportion of communities removing rare species hat a relevant effect.

```
# Specific leaf area (SLA)
f.sla <- function(x) mean(traitmat$sla[as.logical(x)])
CM.sla.obs <- apply(commat, 1, f.sla)
f.sla <- function(x) mean(traitmat_all$sla[as.logical(x)])
CM.sla.cor <- apply(commat_obs, 1, f.sla)
rel <- 100 * abs(lm(CM.sla.cor ~ plantsBDM$elevation)$coef[2])
dif.sla <- abs(CM.sla.obs - CM.sla.cor) > rel

# Canopy height (CH)
f.ch <- function(x) mean(traitmat$ch[as.logical(x)])
CM.ch.obs <- apply(commat, 1, f.ch)
f.ch <- function(x) mean(traitmat_all$ch[as.logical(x)])
CM.ch.cor <- apply(commat_obs, 1, f.ch)
rel <- 100 * abs(lm(CM.ch.cor ~ plantsBDM$elevation)$coef[2])
dif.ch <- abs(CM.ch.obs - CM.ch.cor) > rel

# Seed mass (SM)
f.sm <- function(x) mean(traitmat$sm[as.logical(x)])
CM.sm.obs <- apply(commat, 1, f.sm)
f.sm <- function(x) mean(traitmat_all$sm[as.logical(x)])
CM.sm.cor <- apply(commat_obs, 1, f.sm)
rel <- 100 * abs(lm(CM.sm.cor ~ plantsBDM$elevation)$coef[2])
dif.sm <- abs(CM.sm.obs - CM.sm.cor) > rel
```

Bias due to removing rare species was relevant in 0.6% of communities for SLA, in 0.3% of communities for canopy height and in 1.4% of communities for seed mass.

```
# Functional trait space (FRic)
f.FRic <- function(x) convhulln(traitmat[as.logical(x),], "FA")$vol
FRic.obs <- apply(commat, 1, f.FRic)
```

```
f.FRic <- function(x) convhulln(traitmat_all[as.logical(x),], "FA")$vol
FRic.cor <- apply(commat_obs, 1, f.FRic)
rel <- 100 * abs(lm(FRic.obs ~ plantsBDM$elevation)$coef[2])
FRic.dif <- abs(FRic.obs - FRic.cor) > rel

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

# Mean nearest neighbout distance (mnnd)
dist <- as.matrix(dist(traitmat))
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)
dist <- as.matrix(dist(traitmat_all))
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.cor <- apply(commat_obs, 1, f.mnnd)
rel <- 100 * abs(lm(mnnd.obs ~ plantsBDM$elevation)$coef[2])
mnnd.dif <- abs(mnnd.obs - mnnd.cor) > rel

# Species richness SR
SR.obs <- apply(commat, 1, sum)
SR.cor <- apply(z, 1, sum)
rel <- 100 * abs(lm(SR.obs ~ plantsBDM$elevation)$coef[2])
SR.dif <- abs(SR.obs - SR.cor) > rel
```

Bias due to removing rare species was relevant in 3.3% of communities for functional richness, and in 13% of communities for functional packing. For all measures bias due to removing rare species was less strong than bias due to detection filtering. Again functional packing is the measure that reacts most sensitive to undersampling.

References

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.