

# Workflow using simulated data

Tobias Roth

2017-06-05

## Introduction

TO BE WRITTEN

As a requirement to run the presented code the latest version of the *detecfilter* package should be downloaded using the following code.

```
library(devtools)
install_github("TobiasRoth/detecfilter")
library(detecfilter)
```

The *detecfilter* package provides the function *simcom()* to simulate community data subject to environmental and detection filtering.

## Estimate detection corrected communities

In this chapter we aim to show our workflow to infer the effect of detection filtering on measures of functional diversity. We start with simulating data for a study with 100 sites, each site was visited twice and the regional species pool contains 50 species. It is convenient to use simulated data as this allows to compare the estimates with the truth (i.e. the values used to run the simulations).

```
dat <- simcom(mu.FTfilter.lp = 2, mu.FTfilter.lpsi = -0.5,
             nsite = 100, nspec = 50, nrep = 2)
```

In a traditional analyses we would likely construct a community matrix

Now we aim to estimate the detection corrected community. To do we can use functions provided in the package *unmarked*.

```
library(unmarked)
z <- array(NA, dim = c(dim(dat$y)[1], dim(dat$y)[2]))
a0 <- numeric(dim(dat$y)[2])
a0[!is.na(a0)] <- NA
b1 <- numeric(dim(dat$y)[2])
b1[!is.na(b1)] <- NA
for(k in 1:dim(dat$y)[2]) {
  d <- unmarkedFrameOccu(y = dat$y[,k,], siteCovs = data.frame(gradient = dat$gradient))
  try({
    res <- occu(~ 1 ~ gradient, data = d, se = TRUE)
    a0[k] <- plogis(coef(res)["p(Int)"])
    b1[k] <- coef(res)["psi(gradient)"]
    z[,k] <- bup(ranef(res), stat = "mode")
  })
}
```

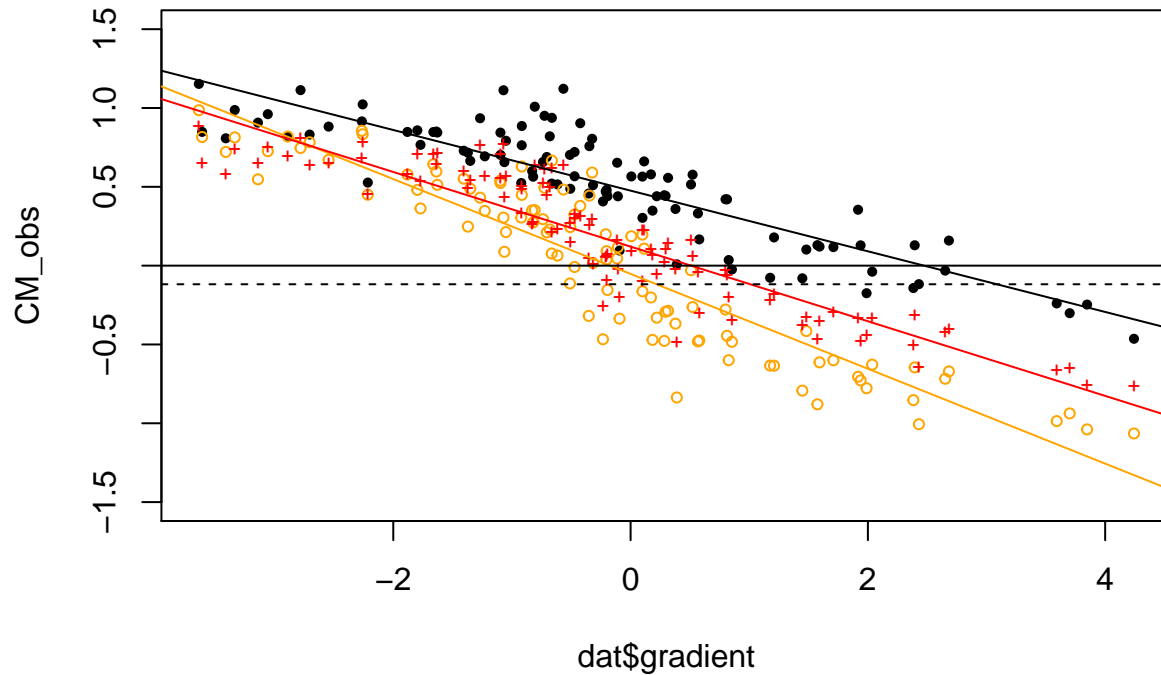
```
# Calculate community mean value of trait expression (CM)
commat_obs <- apply(dat$y, c(1,2), max)
CM_obs <- apply(commat_obs, 1, function(x) mean(dat$traitmat[dat$detected.at.all][x==1]))
```

```

CM_cor <- apply(z>0.5, 1, function(x) mean(dat$traitmat[dat$detected.at.all][x], na.rm = TRUE))
CM_true <- apply(dat$z_true, 1, function(x) mean(dat$traitmat[x==1]))

# Plot CM along gradient
plot(dat$gradient, CM_obs, pch = 16, cex = 0.7, ylim = c(-1.5, 1.5))
points(dat$gradient, CM_true, cex = 0.7, col = "orange")
points(dat$gradient, CM_cor, pch = "+", cex = 0.7, col = "red")
abline(lm(CM_obs ~ dat$gradient))
abline(lm(CM_true ~ dat$gradient), col = "orange")
abline(lm(CM_cor ~ dat$gradient), col = "red")
abline(h=mean(dat$traitmat), lty = 2)
abline(h=0)

```



## References

Kéry, M.; Royle, J. A., 2016. Applied Hierarchical Modeling in Ecology. Academic Press.