New robust weighted averaging- and model-based methods for assessing trait-environment relationships.

Cajo J. F. ter Braak

Appendices

A1. Tutorial

Sections A1.1. and A1.2 show how to apply the multilevel model MLM3 and N_2-weighted CWM/SNC regressions, respectively, with graphs and the max test for both and a bootstrap test for MLM3 using R (R Core Team 2018). The R code to run the tutorial can be found at https://doi.org/10.6084/m9.figshare.8152655. The scripts for applying MLM3 (section A1.1) are TutorialMLM3_Revisit.r and TutorialMLM3_Aravo.r. The script for applying N_2-weighted CWM/SNC regressions (section A1.2) is TutorialWA.r. Instead of including source files, you can install and use the R package TraitEnvMLMWA (install from the file TraitEnvMLMWA _0.0.0.9000.tar.gz). Install the latest version of TraitEnvMLMWA from github by using in an R session the statement:

```
remotes::install_github("CajoterBraak/TraitEnvMLMWA")
```

A1.1 Tutorial using multilevel method MLM3

This tutorial shows how to apply the multilevel model MLM3 using R with the library glmmTMB for overdispersed data and the library lme4 if the overdispersion is very minor. Multilevel model MLM2 is not illustrated as it is inferior to MLM3.

Before the release of glmmTMB, practitioners used lme4 for overdispersed data by including to the linear predictor an observation level random effect '(1|obs)', where obs is a factor with $n \times m$ levels. For estimation, such a model requires the (approximate) integration over an $n \times m$ dimensional space, which is challenging; approximations to the integral are likely not very good. The library glmmTMB avoids these problems by allowing response distributions with overdispersion parameters.

Section A1.1 shows analysis of the Revisit data using binomial data with overdispersion as in the main text. Section A1.2 show the analysis of the Aravo data (Appendix A7) using count data, first with overdispersion and then, using lme4, without, as these data show minor overdispersion.

Logit analysis using Revisit data

The data for this example is taken from Miller et al. (2018). This data file `whittakerrevisitdata.csv` is already nearly in the format needed for MLM3

analysis. See the section A1.2 for getting the data in such a format from the format in Figure 1 in the main text (the three items, Y, **t** and **e**). Note also that the trait and environmental variable are already standardized to zero mean and unit variance.

Main analyses

Load the data in to R.

```
dat <- read.csv("data/whittakerrevisitdata.csv")

str(dat)

## 'data.frame':    3900 obs. of  5 variables:
##  $ site   : int  109 113 12 156 157 160 161 163 164 185 ...
##  $ species: Factor w/ 75 levels "Abies concolor",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ trait  : num  1.4 1.4 1.4 1.4 1.4 ...
##  $ env    : num  -0.351 -0.774 1.667 1.016 1.342 ...
##  $ value  : int  0 0 0 25 7 4 4 0 0 0 ...
```

In the data, value is the response variable. For logit analysis in R, it must a matrix consisting of successes and failures.

```
dat$y <- with(dat, cbind(value,100-value))

dat$site <- factor(dat$site)
```

The main analysis is obtained by

```
library(glmmTMB)
formula.MLM3.linear <- y ~ env * trait + (1 + trait| site) + (1 + env|species)
MLM3 <- glmmTMB(formula.MLM3.linear, family = betabinomial,  data=dat)
```

resulting in many warnings. We neglect these warnings (see the subsection Warnings issued by glmmTMB). The summary of the analysis is

```
summary(MLM3)

##  Family: betabinomial  ( logit )
## Formula:
## y ~ env * trait + (1 + env | species) + (1 + trait | site)
## Data: dat
##
##      AIC      BIC   logLik deviance df.resid
##   6193.9   6262.9  -3086.0   6171.9     3889
##
## Random effects:
##
## Conditional model:
##  Groups  Name        Variance Std.Dev. Corr
##  species (Intercept) 2.1786   1.4760
##          env         0.1776   0.4214   0.41
##  site    (Intercept) 0.2255   0.4749
##          trait       0.1134   0.3367   -0.92
## Number of obs: 3900, groups:  species, 75; site, 52
##
## Overdispersion parameter for betabinomial family (): 3.35
##
## Conditional model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.42782    0.21827 -24.868  < 2e-16 ***
## env         -0.68239    0.12503  -5.458 4.82e-08 ***
```

```
## trait        1.00306    0.19788   5.069 4.00e-07 ***
## env:trait    0.25117    0.09583   2.621  0.00876 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimate of the parameter for trait-environment interaction β_{te} is the entry `env:trait 0.25117` and the Wald test in the last column is highly significant (p=0.00876). Estimates for the square-root of the variance components are $\hat{\sigma}_b$ = 0.3367 and $\hat{\sigma}_c$ =0.4214. There is considerable correlation between the (Intercept) and env in the stratum 'species' (0.41) and (Intercept) and trait in the stratum 'site' (-0.92). If one of these correlations is close to 1, it may be difficult to obtain stable estimates of the variance components.

It is wise to check for non-linear main effects. Here a polynomial of order 2 is fitted and summarized.

```r
# extension with polynomial main effects

formula.MLM3.quad <- y ~ poly(env,2)+poly(trait,2)+ env : trait + (1 + trait| site)
+ (1 + env|species)

MLM3.quad <- update(MLM3,formula.MLM3.quad)

summary(MLM3.quad)

##  Family: betabinomial  ( logit )
## Formula:
## y ~ poly(env, 2) + poly(trait, 2) + (1 + env | species) + (1 +
##     trait | site) + env:trait
## Data: dat
##
##      AIC      BIC   logLik deviance df.resid
##   6196.9   6278.4  -3085.4   6170.9     3887
##
## Random effects:
##
## Conditional model:
##  Groups  Name        Variance Std.Dev. Corr
##  species (Intercept) 2.1772   1.4755
##          env         0.1806   0.4249   0.43
##  site    (Intercept) 0.2260   0.4754
##          trait       0.1109   0.3330   -0.92
## Number of obs: 3900, groups:  species, 75; site, 52
##
## Overdispersion parameter for betabinomial family (): 3.35
##
## Conditional model:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -5.43000    0.21858 -24.842  < 2e-16 ***
## poly(env, 2)1   -43.12954    7.88765  -5.468 4.55e-08 ***
## poly(env, 2)2    -1.30393    3.46687  -0.376  0.70683
## poly(trait, 2)1  61.99530   12.35436   5.018 5.22e-07 ***
## poly(trait, 2)2  10.45304   10.95875   0.954  0.34016
## env:trait         0.25310    0.09592   2.639  0.00832 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that the model is an equi-width Gaussian logit response model (ter Braak & Looman 1986; Jamil & ter Braak 2013), because the coefficient for the squared term for env is negative (-1.30393). But it is non-significant. The corresponding coefficient for trait is positive and non-significant. Comparing these models by ANOVA

```
anova(MLM3,MLM3.quad)

## Data: dat
## Models:
## MLM3: y ~ env * trait + (1 + env | species) + (1 + trait | site), zi=~0, disp=~1
## MLM3.quad: y ~ poly(env, 2) + poly(trait, 2) + (1 + env | species) + (1 + , zi=~
0, disp=~1
## MLM3.quad:     trait | site) + env:trait, zi=~0, disp=~1
##           Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## MLM3      11 6193.9 6262.9 -3086.0   6171.9
## MLM3.quad 13 6196.9 6278.4 -3085.4   6170.9 1.0501      2     0.5915
```

shows no evidence for the need for the quadratic terms (p= 0.59). The remainder of the analyses of the Revisit data uses the linear MLM3 model.

A likelihood ratio test (LRT) is often (slightly) superior to the Wald test, but it requires the fit of two models and thus takes more time to compute. The LRT test for the interaction is obtained by the code:

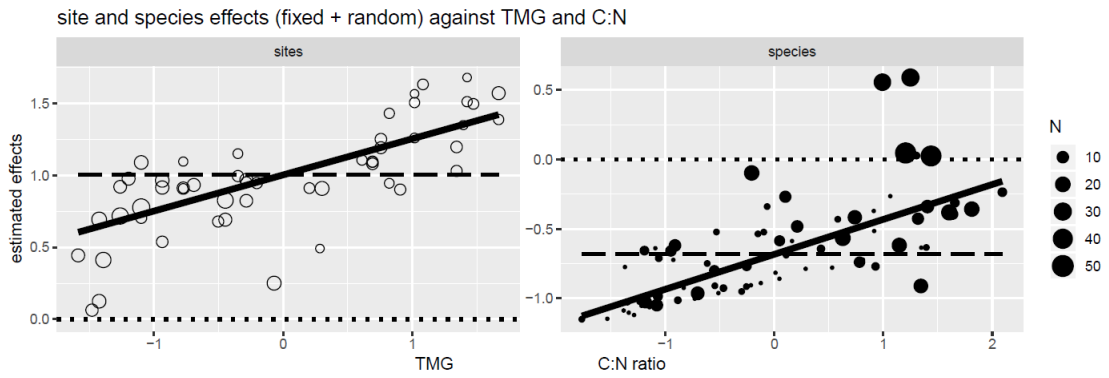
```
# LRT test ----------
# perhaps useful if there are less than 20 species or sites
formulaNULL <- update(formula(MLM3),  ~ . - trait:env )
MLM0 <- glmmTMB(formulaNULL, data=dat, family= family(MLM3))

## Warning in f(par, order = order, ...): value out of range in 'lgamma'

[...]

## Warning in f(par, order = order, ...): value out of range in 'lgamma'

anova(MLM0,MLM3)

## Data: dat
## Models:
## MLM0: y ~ env + trait + (1 + trait | site) + (1 + env | species), zi=~0, disp=~1
## MLM3: y ~ env * trait + (1 + trait | site) + (1 + env | species), zi=~0, disp=~1
##      Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## MLM0 10 6198.8 6261.5 -3089.4   6178.8
## MLM3 11 6193.9 6262.9 -3086.0   6171.9 6.8897      1    0.00867 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The LRT gives p-value 0.00867, which is close the Wald p-value (p=0.00876). In the initial work for the paper, both tests were used, but as I did not find much difference, I switched to the use of the Wald test only.

Graphs of fixed and random effects

Graphs like Figure 3 of the main text can be made using the function `plot.MLM3`.

```
library(ggplot2)
library(dplyr)
source("Rfunctions/plot.MLM3.r")
plot.MLM3(MLM3) +
  ggtitle('site and species effects (fixed + random) against TMG and C:N') +
  xlab("TMG                              C:N ratio") +
  ylab("estimated effects")

## Warning: Removed 75 rows containing missing values (geom_point).
```



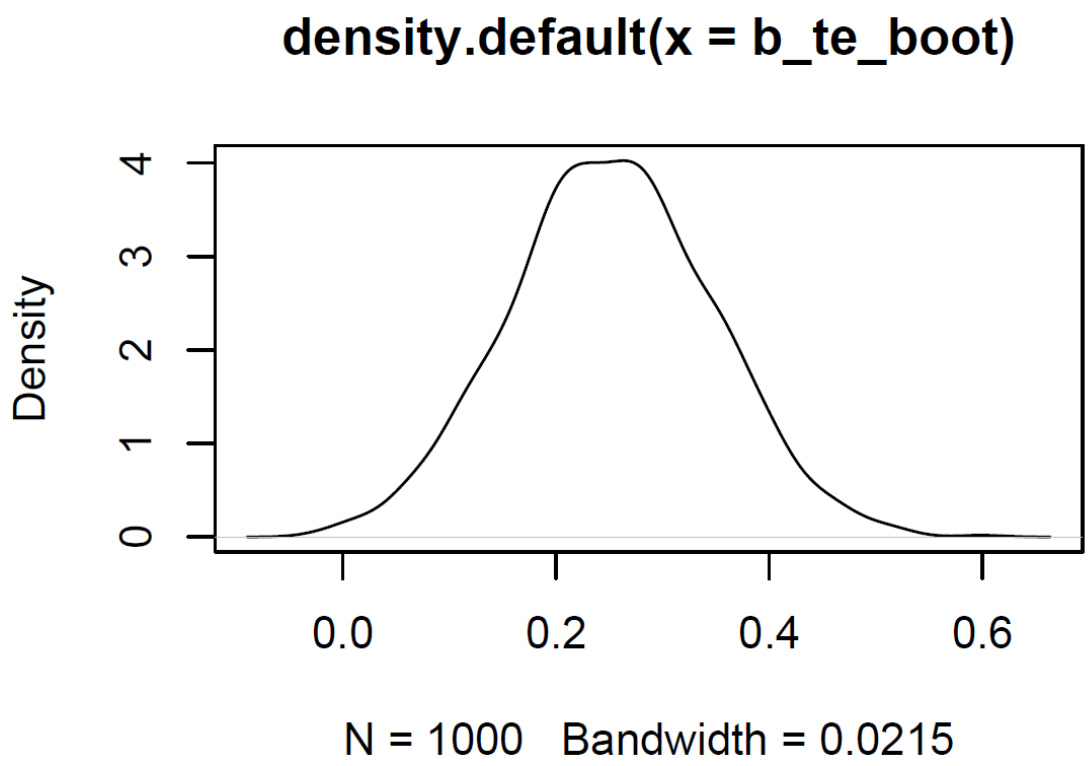
Bootstrap confidence interval for trait-environment parameter

A bootstrap confidence interval can adjust for bias in parametric MLM3. In theory - when glmmTMB with MLM3 works well - the scope for such a bootstrap is limited as MLM3 showed little bias in the simulations, except in logit models with large residual correlation. To deal with such bias, a non- or semi-parametric bootstrap would be needed (such as developed for GLM by Warton et al. 2017), but this is left for future work. The parametric bootstrap simulates without residual correlation, in which case MLM3 did not show bias (Table A8d). By consequence, the parametric bootstrap cannot correct for bias that is due to large residual correlation. However, in practice, the bootstrap is a good check whether glmmTMB with MLM3 worked well. It is a check whether it is warranted to ignore the many warnings glmmTMB issues (see also the subsection Warnings issued by glmmTMB). The code for a parametric bootstrap of the MLM3 model is:

```
getInteractionParameter <- function(MLM3){
  if (class(MLM3)[1] == "glmmTMB"){
        B <- summary(MLM3)$coefficients$'cond'
     } else if (class(MLM3)[1] == "lme4") {
        B <- summary(MLM3)$coefficients
     }
  coef_te <- B[nrow(B),-4]
  return(coef_te)
}
stats_data <- getInteractionParameter(MLM3)
# not implemented in glmmTMB
# bootMer(MLM3, FUN = getInteractionParameter, nsim = 4)
#

dat_boot <- model.frame(MLM3)
nsimul <- 1000
bootstrap_mat<- matrix(0, nrow = nsimul, ncol = length(stats_data))
colnames(bootstrap_mat) <- names(stats_data)
for (i in 1:nsimul){
  dat_boot$y <- as.matrix(simulate(MLM3))
  suppressWarnings(MLM3boot <- glmmTMB(formula(MLM3), family = betabinomial,  data=
dat_boot))
  bootstrap_mat[i, ] <- getInteractionParameter(MLM3boot)
}
summary(bootstrap_mat)

##     Estimate          Std. Error         z value
##  Min.   :-0.03181   Min.   :0.06529   Min.   :-0.2909
```

```
##  1st Qu.: 0.18604   1st Qu.:0.09012   1st Qu.: 1.9296
##  Median : 0.24980   Median :0.09720   Median : 2.5423
##  Mean   : 0.25215   Mean   :0.09762   Mean   : 2.5851
##  3rd Qu.: 0.31582   3rd Qu.:0.10472   3rd Qu.: 3.2402
##  Max.   : 0.59239   Max.   :0.13772   Max.   : 5.6388
```

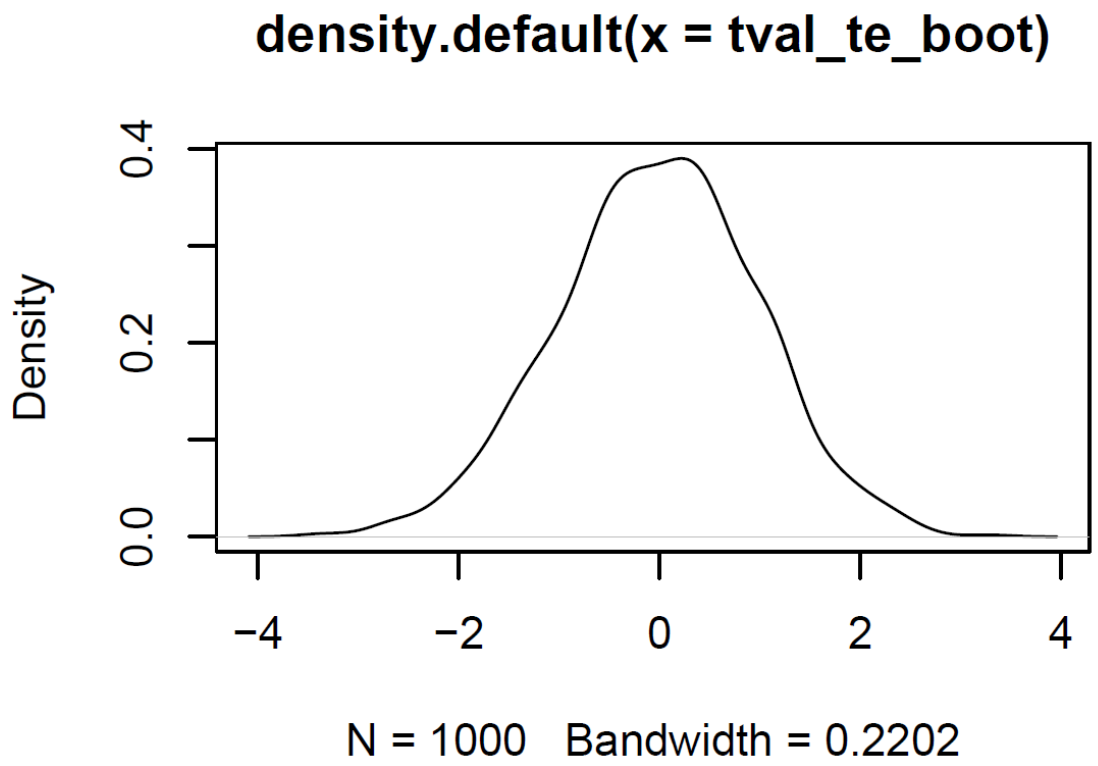
```
# 95% naive/quantile/percentile  bootstap confidence interval (do not use..)
b_te_boot = bootstrap_mat[,1]
plot(density(b_te_boot))
```




```
b_te_qu_ci = quantile(b_te_boot,probs = c(0.025,0.975)) # 95 % quantile / percentri
le bootstrap c.i.
b_te_qu_ci

##       2.5%      97.5%
## 0.05663358 0.46233038

mean(b_te_boot)

## [1] 0.2521484

sd(b_te_boot)

## [1] 0.10179
```

The mean and standard deviation of the bootstrap sample are about the same as the M LM3 estimates (0.25117 and 0.09583, respectively). A bootstrap studentized confide nce interval (Hall 1988, Davison & Hinkley 1997) is constructed as follows:

```
# 95% studentized bootstap confidence interval (use this one)
b_te_est <- stats_data[1]
se_b_te_est <- stats_data[2]
se_b_te_boot <- bootstrap_mat[,2]
# t-value when the true interaction coefficient = b_te_est
tval_te_boot = (b_te_boot - b_te_est)/se_b_te_boot
plot(density(tval_te_boot))
```




```
z_b_te_qu = quantile(tval_te_boot,probs = c(0.025,0.975), na.rm=TRUE)
z_b_te_qu # ideally symmetic -2 to 2, if not, the interval corrects for bias

##      2.5%     97.5%
## -2.081137  2.078249

b_te_Stu_boot_ci = b_te_est - rev(z_b_te_qu)*se_b_te_est
names(b_te_Stu_boot_ci)=rev(names(b_te_Stu_boot_ci))
b_te_Stu_boot_ci

##       2.5%      97.5%
## 0.05202084 0.45060189
```

The bootstrap intervals are almost equal and equal to the parametric one.

The rationale of the `rev(z_b_te_qu)` in the code above comes from the logic of the bootstrap and the hope that the z-value is asymptotically pivotal (Hall 1988,). The logic is that the sample relates to the population in the same way as the bootstrap sample relates to the sample. The reason behind this logic is that the sample forms the truth from which the bootstrap samples. For a small number of bootstrap samples it is better to replace the quantiles `z_b_te_qu` by interpolated ones as in the function `boot.ci` of library, as follows.

```
library(boot)
t0 <- c(b_te_est, se_b_te_est^2)
t <- bootstrap_mat
t[,2] <- t[,2]^2
objboot <- list(t0 = t0, t = t, R = nrow(t))
boot.ci(objboot, type = "stud")

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = objboot, type = "stud")
##
```

```
## Intervals :
## Level     Studentized
## 95%   ( 0.0512,  0.4530 )
## Calculations and Intervals on Original Scale
```

Testing trait-environment interaction by bootstrap

For small number of species or sites (<20, say) MLM3 shows type I error inflation (Appendix A8). A statistical test based on resampling can be useful in this case (by bootstrapping or permutation). This subsection presents the more customary bootstrap test, whereas the next presents the new model-based permutational max test. As there is no proven, reliable non- or semi-parametric bootstrap test, a parametric bootstrap is used. This bootstrap simulates from the model as fitted and is thus based on the same assumptions as MLM3. See Box A2 in Appendix A4. However, the parametric bootstrap test might be able to correct for type I error rate inflation inflation even in the case of structured residual correlation, as the simulations *without* structured residual correlation also show type I error rate inflation (Table A5)[1].

```
# Parametric bootstrap test-------------------------------------------------------
source("Rfunctions/utility_functions.r")
source("Rfunctions/Bootstrap_test.r")
set.seed(123)
nrepet <- 19
res_boot <- Bootstrap_test_prmtrc(MLM3, test_stat = "Wald", nrepet = nrepet, Binomial_total = 100)
```

The output is

```
names(res_boot)

## [1] "p_values"  "nrepet"    "sim.boot"  "MLM0"      "obs"       "test_stat"

round(res_boot$p_values,3)

## p_prmtrc.Wald        p_boot
##         0.009         0.050
## attr(,"nrepet")
## [1] 19
```

The observed test statistic is larger than any of the19 simulations, so that the Monte Carlo *p*-value is 1/(19+1) = 1/20 = 0.05. With 499 permutations, I obtained *p* =0.014.

Testing trait-environment interaction by permutation

The model-based permutation max test can be obtained as follows:

```
# model-based permutation max test ----------------------------------------
source("Rfunctions/setoptions.r")
source("Rfunctions/MLM3_p_max.r")

set.seed(323)
res_perm <- MLM3_p_max(MLM3, test_stat = "Wald", nrepet = nrepet, Binomial_total =
100)
```

with output

[1] If simulations without structured residual correlation would not show type I error rate inflation, the parametric bootstap test is unlikely to correct for type I error inflation, as it samples from such data.

```
names(res_perm)

## [1] "p_values" "nrepet"   "obs"      "sim.row"  "sim.col"

round(res_perm$p_values,3)

##      obs.sites obs.species p.site.permut p.species.permut pmax.permut
## [1,]     0.009       0.009          0.05             0.05        0.05
## attr(,"nrepet")
## [1] 19
```

The observed test statistic is larger than any of the19 simulations in both the site-level and the species-level test, so that both Monte Carlo *p*-values are 0.05. The max test gives thus also $p = 0.05$. With 499 permutations, I obtained $p = 0.012$.

Log-linear analysis using Aravo data

Analysis

This analysis starts from the data format in Figure 1: Y, trait, env. We scale the trait and environmental variable to zero mean and variance one, so as to make the parameter for trait-environment interaction and the variance components comparable among analyses.

```
# read and process data ---------------------------------------------------
library(ade4)
data(aravo)
Y <-aravo$spe
trait <- scale(aravo$traits$SLA)
env <- scale(aravo$env$Snow)
```

The function `make_obj_for_traitenv` checks for empty sites and species without occurrences and creates an object of class TE_obj. This object can then be used to create the expanded data frame for multilevel analysis using `expand4glmm`.

```
source("Rfunctions/utility_functions.r")
obj <- make_obj_for_traitenv(env, Y, trait, cut_off=0)
str(obj)

## List of 3
##  $ L: int [1:75, 1:82] 0 0 3 0 0 0 2 0 0 3 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:75] "AR07" "AR71" "AR26" "AR54" ...
##   .. ..$ : chr [1:82] "Agro.rupe" "Alop.alpi" "Anth.nipp" "Heli.sede" ...
##  $ E: num [1:75, 1] -1.35 -1.35 -1.35 -1.35 -1.35 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:75] "AR07" "AR71" "AR26" "AR54" ...
##   .. ..$ : chr "env"
##  $ T: num [1:82, 1] -1.083 0.13 0.633 -0.65 -0.32 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:82] "Agro.rupe" "Alop.alpi" "Anth.nipp" "Heli.sede" ...
##   .. ..$ : chr "trait"
##  - attr(*, "class")= chr [1:2] "TE_obj" "list"


dat <- expand4glmm(obj)
# note that the formula s below should only contain names available in dat, thus:
names(dat)
```

```
## [1] "y"         "site"      "species"   "obs"       "trait"     "env"
## [7] "trait.env"
```

We start by treating the abundance as negative binomial counts and the quadratic model:

```
library(glmmTMB)

formula.MLM3 <- y ~ poly(env,2) + poly(trait,2) +  env : trait  + (1 + env|species)
+ (1 + trait| site)
MLM3 <- glmmTMB(formula.MLM3, family = "nbinom2",  data=dat)

## Warning in fitTMB(TMBStruc): Model convergence problem; non-positive-
## definite Hessian matrix. See vignette('troubleshooting')

## Warning in fitTMB(TMBStruc): Model convergence problem; false convergence
## (8). See vignette('troubleshooting')

summary(MLM3)

## Warning in sqrt(diag(vcov)): NaNs produced

[...]

## Overdispersion parameter for nbinom2 family (): 1.4e+14
```

The summary shows serious warnings; the estimate of the overdispersion parameter is hugh. In the nbinom2 family, the variance model is $\mu_{ij} + \mu_{ij}^2/\phi$, where ϕ is called the overdispersion parameter. However, the true overdispersion is $1/\phi$, so that a large value for ϕ means lack of overdispersion. Therefore, further analyses use the Poisson distribution. For this, we switch to library lme4 as it is quicker with the below-set options.

```
# switching to lme4 -----------------------------------------------------
library(lme4)
MLM3 <- glmer(formula.MLM3, family = poisson,  data=dat, nAGQ=0, control = glmerCon
trol(calc.derivs=F))
summary(MLM3)

## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 0) [glmerMod]
##  Family: poisson  ( log )
## Formula:
## y ~ poly(env, 2) + poly(trait, 2) + env:trait + (1 + env | species) +
##     (1 + trait | site)
##    Data: dat
## Control: glmerControl(calc.derivs = F)
##
##      AIC      BIC   logLik deviance df.resid
##   6885.5   6966.2  -3430.8   6861.5     6138
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.6329 -0.4272 -0.2099 -0.0212 10.1805
##
## Random effects:
##  Groups  Name        Variance Std.Dev. Corr
##  species (Intercept) 3.10803  1.7630
##          env         2.15746  1.4688   0.80
##  site    (Intercept) 0.06753  0.2599
##          trait       0.06291  0.2508   -0.78
```

```
## Number of obs: 6150, groups:  species, 82; site, 75
##
## Fixed effects:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -2.7424     0.2127 -12.895  < 2e-16 ***
## poly(env, 2)1    -105.2570    14.2424  -7.390 1.46e-13 ***
## poly(env, 2)2     -51.5606     3.8193 -13.500  < 2e-16 ***
## poly(trait, 2)1    24.6091    16.3983   1.501  0.13343
## poly(trait, 2)2   -33.3940    11.2335  -2.973  0.00295 **
## env:trait           1.1782     0.1780   6.618 3.63e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) ply(n,2)1 ply(n,2)2 ply(t,2)1 ply(t,2)2
## poly(nv,2)1  0.779
## poly(nv,2)2  0.122  0.129
## ply(trt,2)1 -0.062 -0.074     0.034
## ply(trt,2)2  0.040 -0.010     0.090     0.077
## env:trait   -0.090 -0.100    -0.122     0.720    -0.087
```

The output shows large negative significant coefficients for the quadratic main effects.

The fitted model is an equi-width Gaussian logit response model (ter Braak & Looman 1986; Jamil & ter Braak 2013). The parameter for trait-environment interaction is 1.1782 and is highly significant (p = 3.63e-11).

A formal test of the quadratic versus the linear model is obtained via

```
# could main effects be linear?
formula.MLM3.linear <- y ~ env * trait  + (1 + env|species) + (1 + trait| site)
MLM3.linear <- update(MLM3,formula.MLM3.linear)
anova(MLM3, MLM3.linear)

## Data: dat
## Models:
## MLM3.linear: y ~ env + trait + (1 + env | species) + (1 + trait | site) +
## MLM3.linear:     env:trait
## MLM3: y ~ poly(env, 2) + poly(trait, 2) + env:trait + (1 + env | species) +
## MLM3:     (1 + trait | site)
##             Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## MLM3.linear 10 7008.6 7075.8 -3494.3   6988.6
## MLM3        12 6885.5 6966.2 -3430.8   6861.5 127.05      2  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The linear model is strongly rejected. It is of interest to note that the interaction parameter and its standard error in the linear model is almost the same as those in the quadratic.

Bootstrap confidence interval for trait-environment parameter

From an lme4 object, a bootstrap can be constructed with the function `bootMer`, as follows. The function `getInteractionParameter` is constructed for use of the function `boot.ci` in library boot for the construction of a studentized confidence interval.

```
# Bootstrap confidence interval ------------------------------------------

getInteractionParameter <- function(MLM3){
```

```
  if (class(MLM3)[1] == "glmmTMB"){
    B <- summary(MLM3)$coefficients$'cond'
  } else if (class(MLM3)[1] == "glmerMod") {
    B <- summary(MLM3)$coefficients
  }
  # estimate with variance for boot.ci
  coef_te <- c(B[nrow(B),1],B[nrow(B),2]^2)
  return(coef_te)
}

aa <- bootMer(MLM3, FUN = getInteractionParameter, nsim = 1000)


b_te_boot <- aa$t[,1]
mean(b_te_boot)

## [1] 1.152826

sd(b_te_boot)

## [1] 0.172938
```

The bootstrap mean and standard deviation are close to the parametric estimates for the interaction coefficient (1.1782) and its standard error (0.1780).

```
library(boot)
boot.ci(aa, index = c(1,2), type = "stud" )
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = aa, type = "stud", index = c(1, 2))
##
## Intervals :
## Level    Studentized
## 95%   ( 0.847,  1.556 )
## Calculations and Intervals on Original Scale
```

The 95% studentized bootstrap interval for the interaction coefficient is (0.847, 1.556), whereas the parametric interval is $1.1782 \pm 2 \times 0.1780 = (0.82, 1.53)$.

Warnings issued by glmmTMB

In fitting the MLM3 model, the library glmmTMB gives numerous warnings. These come in two types:

```
## Warning in f(par, order = order, ...): value out of range in 'lgamma'

## Warning in fitTMB(TMBStruc): Model convergence problem; false convergence
## (8). See vignette('troubleshooting')
```

All results in the paper have been obtained ignoring these warnings. This was warranted for the following reasons. First of all, the parametric bootstrap resulted in estimates corresponding to the estimates of the real data, so that the parametric and bootstrap confidence intervals nearly coincided. Moreover, as an extra assurance, note that the results of the permutation test are not necessarily jeopardized by an unstable fitting process. An unstable fitting process might give a lower power but not an invalid estimate of the significance level (Appendix A4).

References

Davison, A.C. & Hinkley, D.V. (1997) *Bootstrap Methods and Their Applications*. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511802843

Hall, P. (1988) Theoretical comparison of bootstrap confidence intervals (with discussion). *Annals of Statistics,* **16,** 927-981. https://projecteuclid.org/euclid.aos/1176350933

Jamil, T. & ter Braak, C.J.F. (2013) Generalized linear mixed models can detect unimodal species-environment relationships. PeerJ, 1, e95. https://doi.org/10.7717/peerj.95

Miller, J.E.D., Damschen, E.I. & Ives, A.R. (2018) Data from: Functional traits and community composition: a comparison among community-weighted means, weighted correlations, and multilevel models. Dryad Digital Repository. https://doi.org/10.5061/dryad.7gj0s3b

ter Braak, C.J.F. & Looman, C.W.N. (1986) Weighted averaging, logistic regression and the Gaussian response model. *Vegetatio,* **65,** 3-11. https://doi.org/10.1007/BF00032121

R Core Team (2018) *R: A language and environment for statistical computing, version 3.5.1*. R Foundation for Statistical Computing. www.R-project.org, Vienna, Austria.

Warton, D.I., Thibaut, L. & Wang, Y.A. (2017) The PIT-trap—A "model-free" bootstrap procedure for inference about regression models with discrete, multivariate responses. Plos one, 12, e0181790. https://doi.org/10.1371/journal.pone.0181790

A1.2 Tutorial using weighted averaging methods

This tutorial shows how to apply the weighted averaging (WA)-based methods using R with special purpose functions developed for this paper.

The first subsection shows the analysis of the Revisit data, with checks on the R-code. The second subsection show the analysis of the Aravo data as in Appendix A7. This subsection concludes with a multi-trait multi-environment test of significance, which is a weighted version of the score test developed in ter Braak (2017). The code shows how to treat categorical traits and environmental variables (factors in R).

N2-weighted lm: Revisit data

The data for this example is taken from Miller et al. (2018). This data file `whittakerrevisitdata.csv` is not in the format needed for WA-based analyses. We first extract the trait, environmental variable and the abundance matrix (**t**, **e** and **Y** in the main text, 'trait', 'env' and 'Y' in the R code) from this data, as follows.

Main analyses

Load the data in to R.

```
dat=read.csv("data/whittakerrevisitdata.csv")

## adapt dataframe dat for WA-based analyses
n_sites <- with(dat, nlevels(factor(site)))
n_species <- with(dat, nlevels(factor(species)))
species <- dat$species[seq(from  = 1, by = n_sites, length.out = n_species)]
sites <- dat$site[1:n_sites]
trait <- dat$trait[seq(from  = 1, by = n_sites, length.out = n_species)]
env <- dat$env[1:n_sites]
Y <- matrix(dat$value, nrow = n_sites,ncol = n_species, dimnames = list(sites=
sites,species=species))
```

The main analysis is obtained with function `CWMSNC_regressions`. The function includes the permutation test and all results needed for plotting; it needs the utility functions so these are sourced first.

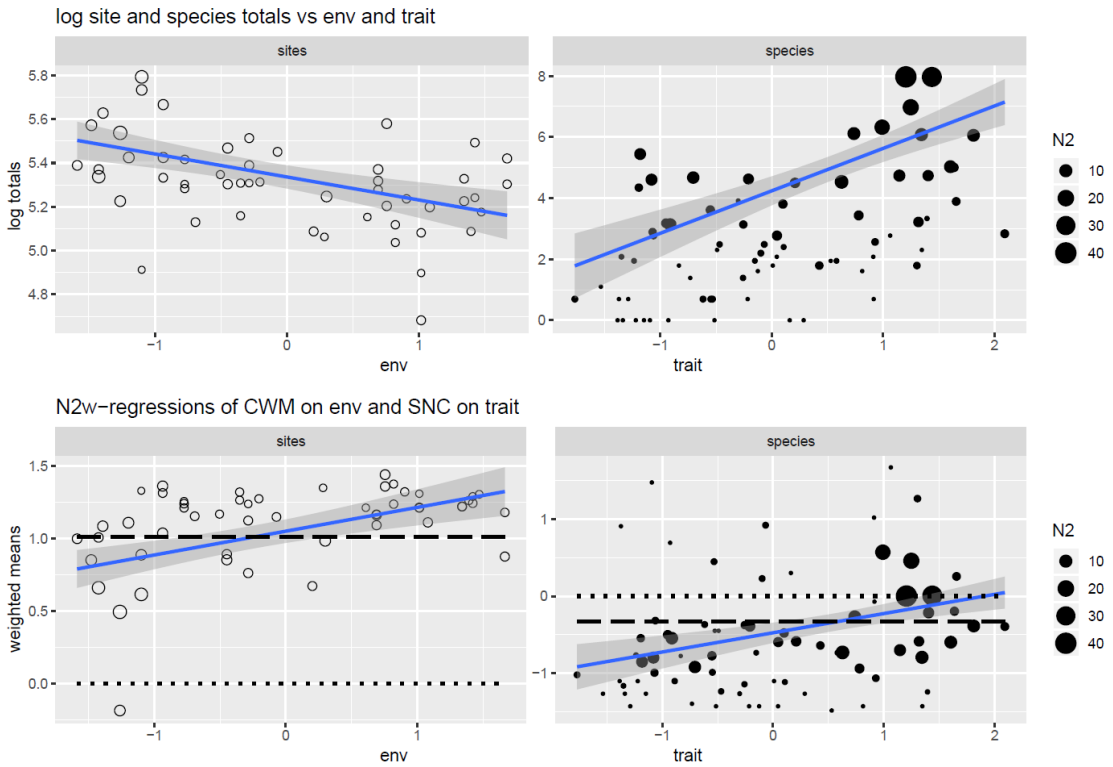
```
#  CWM/SNC regressions ----------------------------------------
source("Rfunctions/utility_functions.r")
source("Rfunctions/WAregressions.r")
set.seed(1231)
result <- CWMSNC_regressions(env, Y, trait, weighing = "N2", nrepet = 499)
summary.CWMSNCr(result)

##                     sites species min/max
## correlations (N2w) 0.170   0.239   0.170
## p-values (N2w)     0.004   0.008   0.008
## attr(,"nrepet")
## [1] 499
```

The summary shows the N_2-weighted fourth-corner correlation (0.170), which is minimum of the site-level and species-level correlations (0.170 and 0.239, respectively), and also the permutation-based p-value, which is the maximum of the site-level and species-level p-values (0.004 and 0.008, respectively). There is thus

statistical evidence of trait-environment association between TMG and C:N ratio; the association is positive.

Plots are created with `plot.CWMSNCr` . The function gives two plots. The first one is about the main effects; it plots the logarithm of the row and column abundance totals against the environmental variable and trait, respectively. The second plot is about the association between the trait and the environmental variable; it plots the CWM and SNC against the environmental variable and the trait, respectively, as in Figure 4 in the main text.

```
plot.CWMSNCr(result)
```



The plots of the totals show that abundance decrease with the environmental variable (TMG) and that abundance increases with the trait (C:N ratio). The plots of CWM and SNC show that there is positive association between TMG and C:N ratio.

Analyses for showing that the short cuts in the R-code work

This subsection shows a check that the fast version of the permutation test, which has many short-cuts, gives the same result as the non-fast version, which recalculates the test statistic without any short-cut. The fast version of `WA_p_max` is used in the main function `CWMSNC_regressions`.

```
# Comparison of two versions of the max test-------------------------------------
-----------------------------
obj <- make_obj_for_traitenv(env,Y, trait, cut_off=0)
set.seed(1231)
```

```r
# slow
system.time(aa<- WA_p_max(obj, nrepet =499, fast = FALSE))

##    user  system elapsed
##    2.33    0.00    2.33

round(aa$p_values,4)

##    p.site.permut p.species.permut      pmax.permut
##            0.004            0.008            0.008
## attr(,"nrepet")
## [1] 499

set.seed(1231)
# default
system.time(bb<- WA_p_max(obj, nrepet =499, fast = TRUE))

##    user  system elapsed
##    0.11    0.00    0.11

round(bb$p_values,4)

##    p.site.prmtrc p.species.prmtrc    p.site.permut p.species.permut
##            1e-04            1e-04            4e-03            8e-03
##      pmax.permut
##            8e-03
## attr(,"nrepet")
## [1] 499

# illustrating their equality
or1<- order(aa$sim.row)
or2<- order(bb$sim.row)
all.equal(or1,or2)

## [1] TRUE

or1<- order(aa$sim.col)
or2<- order(bb$sim.col)
all.equal(or1,or2)

## [1] TRUE
```

The order of the test statistics in the two versions are identical as the test statistic in the fast version is monotonic with the inverse of the parametric *p*-values that is used as test statistic in the non-fast version. The fast version is a factor 10 faster.

The permutation max test of 499 permutations using N2-weighted CWM/SNC regressions takes about 0.11 CPU units in the 'fast' version. A single fit of the MLM3 model took ~65 units (>500 times longer).

N2-weighted lm: Aravo data

This section starts with the analysis of the variables Snow and SLA in the aravo data set. It continues with an analysis of Snow and Spread and with a multi-trait multi-environmental test of association.

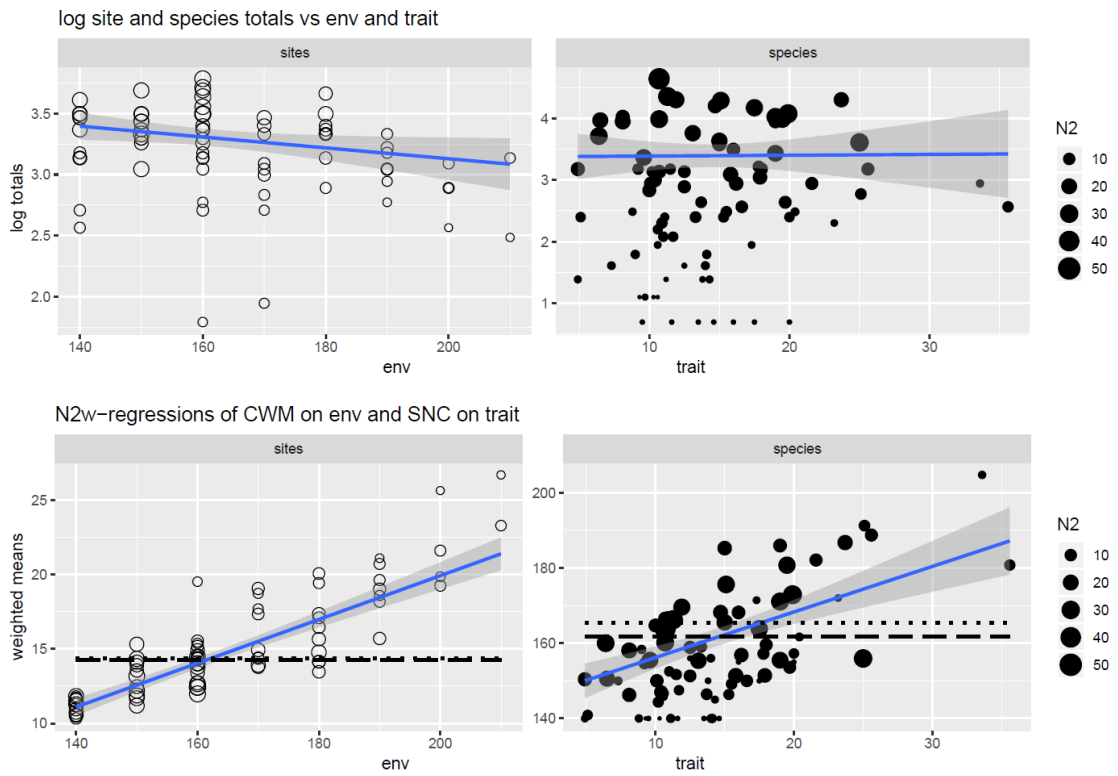
Main analyses

```r
# Aravo data set -------------------------------------------------------
library(ade4)

data("aravo")
```

```
Y <- aravo$spe
SLA <- aravo$traits$SLA
Snow <- aravo$env$Snow

result <- CWMSNC_regressions(Snow, Y, SLA, weighing = "N2", nrepet = 999)

summary.CWMSNCr(result)

##                     sites species min/max
## correlations (N2w) 0.438   0.406   0.406
## p-values (N2w)     0.001   0.001   0.001
## attr(,"nrepet")
## [1] 999
```

The association between Snow and SLA is positive and highly significant. Plots showing main effects and the association

```
plot.CWMSNCr(result)
```



The next analysis investigates the association between Spread and Snow. As the trait Spread is positive and very skew, it is log-transformed before analysis.

```
Spread <- log(aravo$traits$Spread)
result_SLASpread <- CWMSNC_regressions(Snow, Y, Spread, weighing = "N2", nrepet = 9
99)
summary.CWMSNCr(result_SLASpread)

##                     sites species min/max
## correlations (N2w) 0.080   0.072   0.072
## p-values (N2w)     0.002   0.425   0.425
## attr(,"nrepet")
## [1] 999
```

Despite that fact that the site-level permutation test is highly significant, there is no statistical evidence for assocation between Spread and Snow. In this analysis, the trait SLA takes the role of 'unobserved' trait that generates a significant CWM regression.

For illustration, we conclude the tutorial on WA-based methods with a multi-trait multi-environment test. The test statistics is a weighted version of the score test (ter Braak 2017). After looking at histograms of each variable, I decided to log-transform variables that are nonnegative and skew so as to gain some robustness of the outcome. There is one factor in the data, which is transformed into a set of dummy variables. A variable with missing data (ZoogD) has been left out.

```
# Aravo data set: multi-trait multi-env test ------------------------------------
-------------------
summary(aravo$env)

##      Aspect          Slope       Form       PhysD        ZoogD
##  Min.   :1.000   Min.   : 0.00   1:19   Min.   : 0.00   no  :35
##  1st Qu.:5.000   1st Qu.: 0.00   2: 7   1st Qu.:20.00   some:28
##  Median :5.000   Median : 5.00   3:22   Median :30.00   high:12
##  Mean   :6.133   Mean   : 8.76   4:11   Mean   :32.07
##  3rd Qu.:8.000   3rd Qu.:15.00   5:16   3rd Qu.:42.50
##  Max.   :9.000   Max.   :35.00          Max.   :80.00
##       Snow
##  Min.   :140.0
##  1st Qu.:150.0
##  Median :160.0
##  Mean   :165.5
##  3rd Qu.:180.0
##  Max.   :210.0

E <- aravo$env[,-c(5,6)] # without Snow, without ZoogD (ZoogD has NAs)
# Slope is positive and very skew; therefore we logtransform, but there are zeroes
# add minimum non-zero value before taking logs
a <- min(E[E[,"Slope"]>0, "Slope"])
E[,"Slope"] <- log(E[,"Slope"]+ a)
# convert any factor into dummy variables
E <-  model.matrix(as.formula(paste("~ 0", paste(names(E),  collapse= "+"), sep= "+
")), data = E)
summary(E)

##      Aspect          Slope            Form1            Form2
##  Min.   :1.000   Min.   :0.6931   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:5.000   1st Qu.:0.6931   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :5.000   Median :1.9459   Median :0.0000   Median :0.00000
##  Mean   :6.133   Mean   :1.9515   Mean   :0.2533   Mean   :0.09333
##  3rd Qu.:8.000   3rd Qu.:2.8332   3rd Qu.:0.5000   3rd Qu.:0.00000
##  Max.   :9.000   Max.   :3.6109   Max.   :1.0000   Max.   :1.00000
##      Form3            Form4            Form5            PhysD
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   : 0.00
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:20.00
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :30.00
##  Mean   :0.2933   Mean   :0.1467   Mean   :0.2133   Mean   :32.07
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:42.50
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :80.00

T <- aravo$traits[, -6 ] # without SLA
# all traits are skew; therefore we logtransform
T[,c(1,3)] <- log(T[,c(1,3)]+ 1) # traits with zeroes
T[,-c(1,3)] <- log(T[,-c(1,3)]) # traits without zeroes
summary(T)

##      Height           Spread           Angle            Area
##  Min.   :0.0000   Min.   :0.6931   Min.   :0.000   Min.   :0.9163
##  1st Qu.:0.6931   1st Qu.:2.3026   1st Qu.:3.045   1st Qu.:3.8986
##  Median :1.3863   Median :2.9957   Median :3.829   Median :4.9647
##  Mean   :1.5121   Mean   :2.7350   Mean   :3.509   Mean   :4.7353
```

```
##  3rd Qu.:2.1972   3rd Qu.:3.3556   3rd Qu.:4.263   3rd Qu.:5.7177
##  Max.   :3.4340   Max.   :4.6052   Max.   :4.511   Max.   :7.4384
##      Thick             N_mass           Seed
##  Min.   :-2.5257   Min.   :4.528   Min.   :-4.6052
##  1st Qu.:-1.8326   1st Qu.:5.208   1st Qu.:-2.1855
##  Median :-1.6094   Median :5.447   Median :-1.1394
##  Mean   :-1.5147   Mean   :5.472   Mean   :-1.1732
##  3rd Qu.:-1.2040   3rd Qu.:5.752   3rd Qu.:-0.1173
##  Max.   : 0.3365   Max.   :6.139   Max.   : 1.5623

# convert any factor into dummy variables
T <-  model.matrix(as.formula(paste("~ 0", paste(names(T),  collapse= "+"), sep= "+
")), data = T)

obj <- make_obj_for_traitenv(E,Y, T, cut_off=0)
set.seed(1231)
aa<- WA_p_max(obj, nrepet =999, weighing = "N2", score_test = TRUE)
aa$p_values

##    p.site.prmtrc p.species.prmtrc    p.site.permut p.species.permut
##               NA               NA            0.001            0.001
##      pmax.permut
##            0.001
## attr(,"nrepet")
## [1] 999
```

There is evidence in the data for other associations than between SLA and Snow. If `score_test = TRUE` the function uses a score test which takes correlations between traits and between environmental variables (intra-set correlations) into account. If `weighing = "FC"` as well, it gives the test used in double constrained correspondence analysis (ter Braak, Šmilauer & Dray 2018). If `score_test = FALSE` the test ignores such correlations. If `weighing = "FC"` as well, it give the test used in RLQ (Dray *et al.* 2014). Plots of the multi-trait-multi-environment association can be made using RLQ and double constrained correspondence analysis (dc-CA); see Dray et al. 2014 and ter Braak *et al.* 2018. A procedure for selecting functional traits and environmental variables from a greater set of variables is available in Canoco 5.10 (ter Braak & Šmilauer 2018).

References

Dray, S., Choler, P., Dolédec, S., Peres-Neto, P.R., Thuiller, W., Pavoine, S. & ter Braak, C.J.F. (2014) Combining the fourth-corner and the RLQ methods for assessing trait responses to environmental variation. *Ecology,* **95,** 14-21. http://doi.org/10.1890/13-0196.1

Miller, J.E.D., Damschen, E.I. & Ives, A.R. (2018) Data from: Functional traits and community composition: a comparison among community-weighted means, weighted correlations, and multilevel models. Dryad Digital Repository. https://doi.org/10.5061/dryad.7gj0s3b

ter Braak, C.J.F. (2017) Fourth-corner correlation is a score test statistic in a log-linear trait–environment model that is useful in permutation testing. *Environmental and Ecological Statistics,* **24,** 219-242. https://dx.doi.org/10.1007/s10651-017-0368-0

ter Braak, C.J.F. & Šmilauer, P. (2018) *Canoco reference manual and user's guide: software for ordination (version 5.10).* Microcomputer Power, Ithaca, USA. http://www.microcomputerpower.com/

ter Braak, C.J.F., Šmilauer, P. & Dray, S. (2018) Algorithms and biplots for double constrained correspondence analysis. *Environmental and Ecological Statistics,* **25,** 171-197. https://doi.org/10.1007/s10651-017-0395-x

A2. Alternative derivation of the MLM3 model

In this appendix the MLM3 model is derived from a system in which species abundance is governed by four variables, consisting of two independent trait variables **t** and **z** and two independent environmental variables **e** and **x**. The variables **t** and **e** are observed and the variables **x** and **z** are unobserved (also called latent). In the model, the (expected) species abundance is determined by the following model:

$$link(\mu_{ij}) = \alpha_0 + \beta_0 t_j + \gamma_0 e_i + \beta_{te} t_j e_i + \beta_{tx} t_j x_i + \beta_{ze} z_j e_i + d_i + a_j \qquad \text{(A1)}$$

where

- link is the link function, typically the log for count data and the logit for binomial data,
- μ_{ij} is the expected abundance of species *j* in site *i*,
- α_0 is the intercept parameter, accounting for the overall mean abundance,
- β_0 is the slope parameter with respect to the trait **t**, accounting for the (here linear) main effect of **t**,
- γ_0 is the slope parameter with respect to the environmental variable **e**, accounting for the (here linear) main effect of **e**,
- β_{te}, β_{tx} and β_{ze} are trait-environment interaction parameters between the respective trait and environmental variables,
- d_i is a random site main effect, accounting for a site main effect that cannot be represented by the main effect of **e**,
- a_j is a random species main effect, accounting for a species main effect that cannot be represented by the main effect of **t**.

Remark: The effects $\{a_j\}$ and $\{d_i\}$ can represent more complicated main effects of **t** and **e** or by main effects of the unobserved variables **z** and **x**, or by other unobserved variables.

Equation (A1) can be rearranged to

$$link(\mu_{ij}) = \alpha_0 + d_i + a_j + (\beta_0 + \beta_{tx} x_i) t_j + (\gamma_0 + \beta_{ze} z_j) e_i + \beta_{te} t_j e_i \qquad \text{(A2)}$$

and further to the MLM3 model

$$link(\mu_{ij}) = \alpha_0 + d_i + a_j + (\beta_0 + b_i) t_j + (\gamma_0 + c_j) e_i + \beta_{te} t_j e_i \qquad \text{(A3)}$$

with $b_i = \beta_{tx} x_i$ and $c_j = \beta_{ze} z_j$. If **x** and **z** are standard normal variables, then $b_i \sim N(0, \sigma_b^2)$ with $\sigma_b = \beta_{tx}$ and $c_j \sim N(0, \sigma_c^2)$ with $\sigma_c = \beta_{ze}$. The complete MLM3 model also carries the assumptions that the pairs (d_i, b_i) and (a_j, c_j) are each bivariate normal.

Remarks:

1. The linear main effect model in Equations (A.1)-(A.2) can be replaced by nonlinear models without changing the meaning of the linear-by-linear trait-environment interaction $\beta_{te} t_j e_i$. With second-order polynomials (with negative coefficients for the squared terms) the model is an equi-width Gaussian model.

Differential niche widths require species-dependent coefficients for the squared terms. In such models, the ecological meaning of the linear-by-linear trait-environment interaction is not at all clear so that GLMM modelling is less appealing. For an alternative, see Jamil et al. (2014).

2. A slight extension of the model was first used in simulations evaluating MLM(glm) in ter Braak et al. (2017).
3. MLM(glm) lacks the random terms involving (d_i, b_i) and (a_j, c_j) in Equation (A3) or, equivalently the random terms $\{a_j\}$,$\{d_i\}$ and the effects of the unobserved variables **x** and **z**. Brown et al. (2014) and Warton et al.(2015a) attempt to account for residual correlation among the species by a site-based bootstrap procedure. ter Braak et al. (2017) showed that this procedure does not lead to valid statistical inference if $\sigma_c \gg 0$ or equivalently $|\beta_{ze}| \gg 0$, that is, if there is an important unobserved trait that interacts with the observed environmental variable, even if this unobserved trait is independent of the observed trait(s).
4. MLM1 and MLM2 lack the random terms (d_i, b_i) and b_i, respectively in Equation (A3). Both models miss thus the effects of the unobserved environmental variable **x**. This paper shows that MLM2 does not lead to valid statistical inference if $\sigma_b \gg 0$ or equivalently $|\beta_{tx}| \gg 0$, that is, if there is an important unobserved environmental variable that interacts with the observed one(s), even if this unobserved environmental variable is independent of the observed environmental variable(s).
5. MLM3 lacks terms for residual correlation due to competition among species or due to other unobserved traits and environmental variables that interact. Models for such residual correlation are presented by Warton et al. (2015b) and Ovaskainen et al. (2017). In this paper I investigate by simulation whether MLM3 is robust against an extra interaction between unobserved trait en environmental variables. The simulations suggest that it is robust to such extra interaction, except for low numbers of species and sites (*n,* or *m* <10.
6. Note that the remarks 3 and 4 are about robustness to unobserved trait and environmental variables, respectively. MLM3 is robust in both cases. MLM3 appears fairly robust against latent interactions. The MLM3-based max text may make it even more robust against such interactions. The rationale for the site-level test is identical to that of Brown et al. (2014) and Warton et al. (2015a) and the species-level test by considering the trait values as exchangeable units.

As in any GLM(M), the data y_{ij} are assumed to derive from a user-defined response distribution with mean parameter μ_{ij}. Typical response distributions are the binomial distribution for '*k* out of *N*' data (which are presence/absence data for *N* =1) and the Poisson distribution for count data. The response distribution may have additional parameters. The betabinomial and negative binomial distributions have each one additional parameter which measures overdispersion compared to the binomial and the Poisson distributions, respectively.

The 'canonical' link functions for the binomial and Poisson distribution are the logistic (logit) and logarithmic (log) function, respectively. Other link functions appropriate for binomial data are the complementary log-log function and the probit function. Note that in the model-based approach the definition of trait-environment interaction depends on

the chosen link function: with a change in link function the meaning and value of β_{te} change.

References

Brown, A.M., Warton, D.I., Andrew, N.R., Binns, M., Cassis, G. & Gibb, H. (2014) The fourth-corner solution – using predictive models to understand how species traits interact with the environment. *Methods in Ecology and Evolution,* **5,** 344-352. https://doi.org/10.1111/2041-210X.12163

Jamil, T., Kruk, C. & ter Braak, C.J.F. (2014) A Unimodal Species Response Model Relating Traits to Environment with Application to Phytoplankton Communities. *Plos one,* **9,** e97583. https://doi.org/10.1371%2Fjournal.pone.0097583

Ovaskainen, O., Tikhonov, G., Norberg, A., Guillaume Blanchet, F., Duan, L., Dunson, D., Roslin, T. & Abrego, N. (2017) How to make more out of community data? A conceptual framework and its implementation as models and software. *Ecology Letters,* **20,** 561-576. https://doi.org/10.1111/ele.12757

ter Braak, C.J.F., Peres-Neto, P. & Dray, S. (2017) A critical issue in model-based inference for studying trait-based community assembly and a solution. *PeerJ,* **5,** e2885. https://doi.org/10.7717/peerj.2885

Warton, D.I., Shipley, B. & Hastie, T. (2015a) CATS regression – a model-based approach to studying trait-based community assembly. *Methods in Ecology and Evolution,* **6,** 389-398. https://doi.org/10.1111/2041-210X.12280

Warton, D.I., Blanchet, F.G., O'Hara, R.B., Ovaskainen, O., Taskinen, S., Walker, S.C. & Hui, F.K.C. (2015b) So Many Variables: Joint Modeling in Community Ecology. *Trends in Ecology & Evolution,* **30,** 766-779. https://doi.org/10.1016/j.tree.2015.09.007

A3. Weighted Averaging (WA) based methods

A3.1 Introduction

Weighted averaging methods date back to the work of early 20th century great ecologists such as Georgy Gause, Heinz Ellenberg and Robert Whittaker; see ter Braak and Verdonschot (1995) and are commonly used for the reconstruction of palaeo-environments (Birks *et al.* 1990; Walker, Mott & Smol 1991; Juggins & Birks 2012) and best understood in the context of ecological niche models, also called unimodal models (ter Braak & Barendregt 1986; ter Braak & Looman 1986). CWM and SNC are approximate solutions of the maximum likelihood equations of the parameters of such models (ter Braak 1985; ter Braak 1986; ter Braak 1988). But, they are also closely related to the fourth corner (Peres-Neto, Dray & ter Braak 2017) and used in the maxent approach of Shipley (2010), because weighed averages and the fourth corner are sufficient statistics in particular Poisson log-linear models (Warton, Shipley & Hastie 2015; ter Braak 2017) without the need for unimodality.

Section A3.2 gives the definitions of CWM and SNC and the associated regressions. Section A3.3 summarizes the existing evidence that these regressions target interaction (and not main effects), which is then further worked out in detail in section A3.4 by starting from simple Poisson log-linear models with either the trait or the environmental variable as predictor. Finally, section A3.5 proposes two variants of the fourth-corner correlation that measure the trait-environment association in the weighted CWM/SNC regressions.

A3.2 Definitions and CWM/SNC regressions

Recall the notation in the main text: $\{y_{ij}\}$ denotes abundances of m species in n sites (i = 1,...,n ; j = 1,...,m) and the trait and environmental variable are denoted by **t** and **e**, and by t_j and e_i for values of individual species and sites, respectively.

The Community Weighted Mean (CWM) for site i constitutes a trait mean, weighted by abundance, defined as

$$c_i = \sum_{j=1}^{m} y_{ij} t_j / y_{i+} \text{ with } y_{i+} = \sum_{j=1}^{m} y_{ij}. \qquad \text{(A4)}$$

I use a "+" replacing an index (subscript) for the sum over the index, *e.g.* $y_{i+} = \sum_j y_{ij}$. [Note that in Appendix A2, the symbol c_j was used for species-specific random effects; it should be clear from the index letter and the context whether CWMs or species-specific random effects are referred to.] The CWM can, of course, also simply be expressed as linear combination of the species proportions, $p_{ij} = y_{ij}/y_{i+}$,

$$c_i = \sum_{j=1}^{m} t_j \, p_{ij}. \qquad \text{(A5)}$$

The use of proportions links CWM to strictly compositional data, i.e. data in which the totals R = $\{y_{i+}\}$ are uninformative. Community Weighted Means regression (CWMr) is a linear regression of the CWM $\{c_i\}$ on to the environmental variable **e**, in R code by

```
lm_CWMe <- lm(CWM~e, weights = wsites)
```

where the regression is unweighted if the weights `wsites` are all equal or weighted when the weights are unequal. Two sets of weights are considered: weights equal to the site totals $\{y_{i+}\}$ or equal to the effective richness of the site, defined as the Hill N_2-number,

$$N_{2i} = 1/\sum_j (y_{ij}/y_{i+})^2 \tag{A6}$$

which is the inverse of the Simpson index. If all non-zero abundance values are equal, this effective number is equal to their number (i.e. the richness), but for unequal abundances the effective number is lower, and how much lower depends on the unevenness of the sample. The N_2 number can be motivated by the fact that the variance of a mean is the population variance divided by the number of values averaged ($var(\bar{x}) = \sigma^2/N$). With a weighted mean, some values have more influence than the others, resulting in a different effective number, here taken as N_2. This works best if the true within-site trait variance is constant; if it is not, then it is hopefully a step in the right direction.

Comparable to the CWM by interchanging species and sites (and **t** and **e**), the Species Niche Centroid (SNC) constitutes an environmental mean, weighted by abundance, and defined as

$$u_j = \sum_{i=1}^{n} y_{ij} e_i / y_{+j} \text{ with } y_{+j} = \sum_{i=1}^{n} y_{ij}. \tag{A7}$$

The SNC can, of course, also simply be expressed as linear combination of the site proportions $p_{ij}^* = y_{ij}/y_{+j}$

$$u_j = \sum_{i=1}^{n} e_i \, p_{ij}^* \tag{A8}$$

Species Niche Centroid regression (SNCr) is a linear regression of the SNC $\{u_j\}$ on to the trait **t**, in R code by

```
lm_SNCt <- lm(SNC~t, weights = wspecies)
```

where the weights `wspecies` are either all equal (unweighted regression), equal to the species totals K = $\{y_{+j}\}$ or, as proposed in this paper, equal to the effective number of occurrences of the species, defined as the Hill N_2-number,

$$N_{2j} = 1/\sum_i (y_{ij}/y_{+j})^2. \tag{A9}$$

If all non-zero abundance values are equal this effective number is equal to their number (i.e. the number of occurrences), but for unequal abundances the effective number is lower, and how much lower depends on the unevenness of the numbers. This N_2 number can be motivated similarly as for sites. This works best if the true within-species environmental variance is constant; if it is not, then it is hopefully a step in the right direction. In paleo-environmental reconstruction, several attempts have been made to take the within-species environmental variance into account, but without any improved prediction performance (Juggins & Birks 2012). For this reason, this more extended kind of weighting is not considered in this paper.

In the theory of weighted regression, weights should be chosen to be inversely proportional to the variance of the response variable, here either CWM or SNC. Otherwise, the usual statistical inference is not valid, as shown by simulation in Appendix S6 of ter Braak et al. (2018a). But it cannot be guaranteed that either equal weights, the totals or the N_2 weights have this property.

In the main text, the *F*-value of these regressions (weighted and unweighted) are used as test statistic in a Monte Carlo permutation test (or equivalently, the inverse of the *p*-value of the slope parameter). In the case of CWMr the environmental values are

randomly permuted and in the case of SNCr the trait values are randomly permuted. The underlying assumptions here are that the environmental values are exchangeable (or even independent) and that the trait values are exchangeable (or even independent). These assumptions do not hold true when there is spatial autocorrelation, a complex study design or phylogenetic correlation. The details of such permutation tests are described in Appendix A4. Appendix A5 describes the max test, which is the test that combines a site-level test such as CWMr with a species-level test such as SNCr, and its properties.

A3.3 Do fourth corner and CWM/SNC regressions target interactions?

WA- and model-based methods are used both to answer questions about trait-environment association. In model-based methods, the trait-environment association is captured in the single regression coefficient β_{te} associated with the interaction term, the product $t_j e_i$, conditional on having main effects in the model (at least linearly, i.e. $\beta_0 t_j$ and $\gamma_0 e_i$ in Equation (A1)). However, it is not clear at all from the description of the fourth-corner and WA-testing approaches (Dray & Legendre 2008; ter Braak, Cormont & Dray 2012), that WA-based methods target interaction. Moreover, the associated permutation tests are described as if they were main effects! As described in these papers, the site-based permutation test tests whether "species abundance is related to the environmental variable" (the link $\mathbf{Y} \leftrightarrow \mathbf{e}$) and the species-based permutation test tests whether "species abundance is related to the trait" (the link $\mathbf{Y} \leftrightarrow \mathbf{t}$). This section summarizes the existing evidence that WA-based methods target interaction, at least in Poisson log-linear models and approximately so in logit models and models with overdispersion. Fourth corner is considered first and other WA-methods follow. In the next section (section A3.4), I show the close relation between CWM and SNC with parameters in Poisson log-linear models from first principles.

First, Peres-Neto et al. (2017) simulated zero-inflated negative binomial data based on a simple log-linear model with main effects in **t** and **e** only, in particular the model of Equation (A1) with link = log, but without any interaction terms. They found no type I error inflation for the fourth-corner-based max test. Apparently, the fourth-corner correlation is not sensitive to the main effects in such a model.

Second, ter Braak (2017) showed that the squared fourth-corner correlation (times the abundance total) is the Rao score test statistic for H_0: $\beta_{te} = 0$ in the simple log-linear model with t-e interaction

$$\log(\mu_{ij}) = \delta_i + \alpha_j + \beta_{te}\, t_j e_i, \qquad \text{(A10)}$$

where $\{\delta_i\}$ and $\{\alpha_j\}$ are unknown fixed effects that encompass all possible main effects of species and sites or traits and environmental variables (these are called saturated main effects) and the data are Poisson distributed. In this simple model, it is thus asymptotically as powerful as the Wald test or likelihood ratio test (LRT). Moreover, the Rao test score in the multi-trait, multi environmental variable case was shown to be related to the total inertia of a double constrained correspondence analysis (ter Braak 2017; ter Braak, Šmilauer & Dray 2018). See also the next section.

Third, Peres-Neto et al. (2017) noted the close relation between the fourth-corner correlation and correspondence analysis, a method which is known to analyse deviations from row-column independence (i.e. deviations from the log-linear main

effects model) along several axes. These deviations constitute the row-column interactions. As formulae, these interactions are $y_{ij} - e_{ij}$ with e_{ij} the expected values under independence having the well-known formula $e_{ij} = y_{i+}y_{+j}/y_{++}$. These are the fitted values under the main effects log-linear model (Equation (A10) with $\beta_{te} = 0$) and a Poisson response distribution. In this specific model, the main effects $\{\hat{\delta}_i\}$ and $\{\hat{\alpha}_j\}$ are proportional to the site and species totals. However, for a count distribution with overdispersion, the expected values under the null model are still

$$e_{ij} = \exp(\hat{\delta}_i\hat{\alpha}_j) = \exp(\hat{\delta}_i)\exp(\hat{\alpha}_j), \tag{A11}$$

but the main effects $\{\hat{\delta}_i\}$ and $\{\hat{\alpha}_j\}$ are no longer precisely proportional to the site and species totals. Further, with a logit link, the expected values are $\exp(\hat{\delta}_i\hat{\alpha}_j)/(1 + \exp(\hat{\delta}_i\hat{\alpha}_j))$ and are thus not strictly proportional either to the site totals or species totals.

Within the Poisson (or maxent) context, the fourth-corner correlations can thus be expected to target the same interactions as the log-linear model. Outside this context, the precise target is less clear.

So far, the focus of this section was on the fourth-corner correlation. We now consider other WA-based methods. CWM/SNC regressions are related to fourth-corner (Peres-Neto, Dray & ter Braak 2017; ter Braak, Peres-Neto & Dray 2018) and are thus expected to target the same interactions as the fourth-corner correlation. There is also a simple independent argument why WA-based methods target interaction in the Poisson log-linear model, as follows.

In the formula for CWM, Equation (A4), the division by y_{i+} removes the row (site) main effect (under the Poisson log-linear model), but not necessarily the column main effect. However, in the site-based CWMr permutation test, each test statistic calculated from data with permuted environmental values, carries the same column main effects as the test statistic calculated from the data with unpermuted environmental values. The CWMr test is thus not likely to be sensitive to either main effect. Similarly, the SNCr test is not likely to be sensitive to either main effect under the Poisson log-linear model. I expect more differences between the WA- and model-based methods for logit models and models with overdispersion.

A3.4 CWM and SNC and Poisson log-linear models

The first theory that was developed for weighted averaging methods was developed in the context of predicting environmental variables from species indicator values (ter Braak & Barendregt 1986). Such indicator values can be considered as habitat descriptors, i.e. as a special type of trait (Shipley 2010). It could be shown that the CWM can be consistent and even fully efficient with respect to maximum likelihood in a niche model in which the indicator values are (for consistency) centroids of the response curves of the species with respect to the environmental variable and (for efficiency) the optima of equi-width Gaussian response curves. The use of SNC for the estimation of such optima was discussed by ter Braak and Looman (1986). In this section I show a close relation between CWM and SNC and Poisson log-linear models without requiring a Gaussian response. Instead, closeness to the independence model is required (the independence model is here a multiplicative row-column model as in the

usual chi-square test of independence in a contingency table). Related work is by Goodman (1986) and Ihm and van Groenewoud (1984) in the context of correspondence analysis in comparison with the RC association model and the Gaussian ordination model, respectively.

In the model related to CWM, the abundance data are regressed on to the trait using a Poisson log-linear model with species and site main effects and site-specific responses to the trait, as in CATS (Shipley 2010; Warton, Shipley & Hastie 2015). As shown mathematically in the next subsection, the maximum likelihood estimators of the site-specific response parameters in this model are (almost) proportional to the CWMs when the trait has a weighted mean of zero, where the weights are the species total abundances $\{y_{+j}\}$ (this centering of the trait carries no loss of generality). This result is related to the observations on CWM in Warton et al. (2015). In a similar model related to SNC, with species and site main effects and species-specific responses to the environment, the maximum likelihood estimators for the species-specific responses to the environmental variable are (almost) proportional to the SNCs when the environmental variable has a weighted mean of zero, where the weights are the site total abundances $\{y_{i+}\}$. For both low-structured and highly Gaussian structured ecological data, CWM and SNC thus appear as natural statistics for measuring interaction with the trait and the environmental variable, respectively, in a model that accounts for both species and site main effects.

Derivation of CWM and SNC as approximate parameters of log-linear models

I show and develop the model that is relevant for CWM first. The model for SNC is formulated next in a similar way. For CWM, I start from a log-linear model with site-specific response to the trait; it regresses the abundance data y_{ij} on to the trait with mean model

$$\mu_{ij} = R_i C_j \exp(\beta_i t_j), \tag{A12}$$

where R_i and C_j are site and species (fixed) main effects and β_i is the site-specific response with respect to the trait. This is the core model of CATS regression, except that the C_j are user-defined in CATS as proportions of the species in the regional species pool (instead of as derived from the data set under investigation). Under the assumption that the data are Poisson distributed counts, the minimal sufficient statistics are the row totals $\{y_{i+}\}$, the column totals $\{y_{+j}\}$ and $\{\sum_j y_{ij} t_j\}$. The third set of statistics can be replaced by the CWMs $\{c_i\}$ of Equation (A4). By setting the partial derivatives of the Poisson log-likelihood with respect to the parameters to zero (ter Braak 2017) we obtain the following three equations,

$$y_{i+} = \mu_{i+}, y_{+j} = \mu_{+j} \text{ and } \sum_j (y_{ij} - \mu_{ij}) t_j = 0 \tag{A13}$$

If $\beta_i t_j$ is small, i.e. when the data are close to independence,

$$\mu_{ij} = R_i C_j \exp(\beta_i t_j) \approx R_i C_j (1 + \beta_i t_j) \approx y_{i+} y_{+j} (1 + \beta_i t_j) / y_{++}, \tag{A14}$$

because $\exp(x) \approx 1 + x$ for small x and $\hat{\mu}_{ij} = y_{i+} y_{+j} / y_{++}$ under the null model ($\beta_i = 0$). To ensure that $y_{i+} = \mu_{i+}$ and $y_{+j} = \mu_{+j}$ in the approximation, the trait must be centered so that it has zero weighted mean after centering, where the weights are $\{y_{+j}\}$, and the $\{\beta_i\}$ must also have zero weighted mean, where the weights are $\{y_{i+}\}$, *i.e.* $\sum_j y_{+j} t_j = 0$ and $\sum_i y_{i+} \beta_i = 0$. On inserting approximation (A14) in the third equation in (A13) , we obtain

$$\sum_j y_{ij}t_j - \sum_j y_{i+}y_{+j}(1+\beta_i t_j)t_j/y_{++} = 0. \quad \text{(A15)}$$

By solving this equation for β_i and recalling that the trait has zero weighted mean, we obtain

$$\hat{\beta}_i = \frac{\sum_j y_{ij}t_j}{y_{i+}} / \sum_j \frac{y_{+j}}{y_{++}} t_j^2 \propto c_i, \quad \text{(A16)}$$

the CWM of the centered trait for site i. Because of the centering of the trait, the CWMs $\{c_i\}$ also have zero weighted mean, where the weights are the site totals $\{y_{i+}\}$, as

$$\sum_i y_{i+}c_i = \sum_i y_{i+} \sum_j y_{ij}t_j/y_{i+} = \sum_j t_j \sum_i y_{ij} = \sum_j y_{+j}t_j = 0, \quad \text{(A17)}$$

so that the $\{\hat{\beta}_i\}$ are also centered as required below Equation (A14) and all estimation equations in (A13) are satisfied.

In CWMr, the CWMs $\{c_i\}$ are regressed on to the environmental variable **e** using a simple linear model with intercept. The F-statistic and p-value of such a regression model are invariant to location and scale of the response and/or the predictor variable (*i.e.* do not change after addition of, or multiplication by, or a constant). Therefore it does not matter whether to regress the CWMs of the trait or of the centered trait on to the environmental variable.

For SNC, I start from a log-linear model with species-specific response to the environmental variables. In this model, abundance is related to the environmental variable using a log-linear model with mean

$$\mu_{ij} = R_i C_j \exp(\gamma_j e_i), \quad \text{(A18)}$$

where R_i and C_j are site and species (fixed) main effects and γ_j is the species-specific response with respect to the environmental variable. Similar to the above derivation

$$\hat{\gamma}_j = \frac{\sum_i y_{ij}e_i}{y_{+j}} / \sum_i \frac{y_{i+}}{y_{++}} e_i^2 \propto u_j \quad \text{(A19)}$$

with u_j the SNC of species j and the $\{e_i\}$ have zero weighted mean ($\sum_i y_{i+}e_i = 0$).

In SNCr, the SNCs $\{u_j\}$ are regressed on to the trait **t** in a simple linear model with intercept. The F-statistic and p-value of such a regression model are invariant to location and scale of the response and/or the predictor variable. Therefore it does not matter whether to regress the SNC of the environmental variable or of the centered environmental variable on to the trait.

A3.5 Trait-environment correlations

The correlations that follow directly from the (weighted) CWM/SNC regressions are unsuited for measuring trait-environment association; they can be overly variable and unstable (Peres-Neto, Dray & ter Braak 2017; ter Braak, Peres-Neto & Dray 2018). This section defines more stable measures of correlation that are inspired by the fourth-corner correlation. First, I define a weighted correlation that encompasses the fourth-corner correlation. Then I show how the CWM/SNC regressions are related to these correlations and finally I define new measures of correlation. The values of these new correlations may differ between CWMr and SNCr, *i.e.* between the site- and species-level analyses. The correlations are combined by taking a signed minimum, or, if the two correlations differ in sign, by setting the final correlation to 0. Taking the (signed) minimum is akin to the max test as it takes the weaker relation as the final measure of

evidence for the relation. From the unweighted CWM/SNC regressions, we so obtain the unweighted fourth-corner correlation. From the N_2-weighted CWM/SNC regressions we similarly obtain the N_2-weighted four-corner correlation (or N_2-weighted correlation, for short). CWM/SNC regressions weighted by species and site total lead simply to the original fourth-corner correlation. Appendix 9 contains R-code to compute the N_2-weigthed and unweighted variants of the fourth-corner correlation.

A general definition of weighted correlation

The fourth-corner correlation is a special case of the following definition of a weighted correlation between **t** and **e**. Let $\tilde{t}_j$ and $\tilde{e}_i$ be centered versions of the trait and environmental variable, with weighted-means computed using weights **k** = $\{k_j\}$ and **r** = $\{r_i\}$:

$$\tilde{t}_j = t_j - \sum_j k_j\, t_j/k_+ \text{ and } \tilde{e}_i = e_i - \sum_i r_i e_i/r_+. \tag{A20}$$

Then, the general definition is

$$f_{\mathbf{W},\mathbf{k},\mathbf{r}} = cor_{\mathbf{W},\mathbf{k},\mathbf{r}}(\mathbf{t},\mathbf{e}) = \frac{\sum_{i,j} w_{ij}\tilde{t}_j\tilde{e}_i/w_{++}}{\{\sum_j(k_j\tilde{t}_j^2/k_+ \sum_i(r_i\tilde{e}_i^2/r_+)\}^{1/2}} \tag{A21}$$

With this definition, the fourth-corner correlation is $r_{\mathbf{Y},\mathbf{k},\mathbf{r}}$ with $r_i = y_{i+}$ and $k_j = y_{+j}$, the row and column sums of **Y**. For general weights **r** and **k**, the site-level correlation is defined by setting $w_{ij} = r_i y_{ij}/y_{i+}$ (so that $w_{i+} = r_i$) and the species-level correlation by setting $w_{ij} = k_j y_{ij}/y_{+j}$ (so that $w_{+j} = k_j$). This choice is motivated by the link it gives to the (un)weighted CWM and SNC regressions as we will show in the next subsection. For the unweighted CWM/SNC regressions we use $r_i = 1$ and $k_j = 1$, and for the N_2-weighted regressions we use $r_i = N_{2i}$ and $k_j = N_{2j}$ from Equations (A6) and (A9). With $r_i = y_{i+}$ and $k_j = y_{+j}$, the site- and species-level correlations are equal and identical to the original fourth-corner correlation, but for other weighting schemes, these correlations may differ in value and even in sign. The correlations are combined by taking a signed minimum, or, if the two correlations differ in sign, by setting the final correlation to 0.

We conclude this subsection by noting that the correlation in Equation (A21) can also be expressed as "covariance divided by two standard deviations" as follows:

$$f_{\mathbf{W},\mathbf{k},\mathbf{r}} = cor_{\mathbf{W},\mathbf{k},\mathbf{r}}(\mathbf{t},\mathbf{e}) = \frac{cov_{\mathbf{W}}(\mathbf{t},\mathbf{e})}{sd_{\mathbf{k}}(\mathbf{t})sd_{\mathbf{r}}(\mathbf{e})} \tag{A22}$$

with weighted covariance and standard deviations

$$cov_{\mathbf{W}}(\mathbf{t},\mathbf{e}) = \sum_{i,j} w_{ij}\, \tilde{t}_j\tilde{e}_i/w_{++} \tag{A23}$$

$$sd_{\mathbf{k}}(\mathbf{t}) = \{\sum_j k_j\, \tilde{t}_j^2/k_+\}^{1/2} \text{ and } sd_{\mathbf{r}}(\mathbf{e}) = \{\sum_i r_i\, \tilde{e}_i^2/r_+\}^{1/2} \tag{A24}$$

The new weighted site- and species-level correlations differ only in the definition of the numerator by using different weights **W** (for the site-level $w_{ij} = r_i y_{ij}/y_{i+}$ and the species-level $w_{ij} = k_j y_{ij}/y_{+j}$ with, for the N_2-weighted variant, $r_i = N_{2i}$ and $k_j = N_{2j}$).

Derivation and motivation for the weighted fourth-corner correlations

In order to derive and motivate the new weighted fourth-corner correlations, we first show how the CWM regression with site total weights relates to the fourth-corner correlation and then extend the relation to other weighting schemes. Note that, for $\mathbf{W} = \mathbf{Y}$, the numerator in Equation (A21) can be re-expressed in terms of the CWM **c**

$$\sum_{i,j} y_{ij}\, \tilde{t}_j \tilde{e}_i / y_{++} = \sum_i \{ y_{i+} \sum_j (y_{ij}\, \tilde{t}_j / y_{i+}) \tilde{e}_i / y_{++} \} = \sum_i y_{i+} c_i \tilde{e}_i / y_{++}, \qquad \text{(A25)}$$

which is a covariance between the CWM and the environmental variable, weighted by the site totals $\{y_{i+}\}$. This covariance is used in the CWM regression (*e.g.* in the calculation of the slope of the regression) when weighted by these totals. The correlation associated with this regression divides this covariance by the weighted standard deviations of CWM and the environmental variable with the same weights. However, for a standardized trait, the weighted standard deviation of CWM is a measure of strength of the relation between **Y** and **t** (ter Braak 1987; ter Braak, Peres-Neto & Dray 2018), so it is strange to divide by it (if **Y** and **t** are unrelated there cannot be trait-environment association). Instead, the usual fourth-corner correlation divides the covariance by the weighted standard deviations of **t** and **e** (see Equations (A21) - (A22)).

In the **r**-weighted CWM regression, the essential covariance is $\sum_i r_i c_i \tilde{e}_i / r_+$. Similarly, in the **k**-weighted SNC regression, the essential covariance is $\sum_j k_j u_j \tilde{t}_i / k_+$. The associated site- and species- level correlations are obtained by dividing these two covariances (which may differ in value and sign) by the **k**-weighted standard deviation of the trait and the **r**-weighted standard deviation of the environmental variable. These correlations can be expressed in terms of Equation (A21); for the site-level correlation, by setting $w_{ij} = r_i y_{ij} / y_{i+}$ and, for the species-level correlation, by setting $w_{ij} = k_j y_{ij} / y_{+j}$ as we show now.

These essential covariances can be re-expressed in the form of the numerator of Equation (A21). In the **r**-weighted CWM regression, the essential covariance is

$$\sum_i \frac{r_i c_i \tilde{e}_i}{r_+} = \sum_i r_i \sum_j (y_{ij}\, \tilde{t}_j / y_{i+}) \tilde{e}_i / r_+ = \sum_{i,j} (\frac{r_i y_{ij}}{y_{i+}}) \tilde{t}_j \tilde{e}_i / r_+ = \sum_{i,j} w_{ij}\, \tilde{t}_j \tilde{e}_i / w_{++}. \qquad \text{(A26)}$$

with $w_{ij} = r_i y_{ij} / y_{i+}$ so that $r_i = w_{i+}$ and $r_+ = w_{++}$.

Similarly, in the **k**-weighted SNC regression, the essential covariance is

$$\sum_j k_j u_j \tilde{t}_i / k_+ = \sum_{i,j} (k_j y_{ij} / y_{+j}) \tilde{t}_j \tilde{e}_i / k_+ = \sum_{i,j} w_{ij}\, \tilde{t}_j \tilde{e}_i / w_{++}.$$

for $w_{ij} = k_j y_{ij} / y_{+j}$, so that $k_j = w_{+j}$ and $k_+ = w_{++}$.

With $r_i = y_{i+}$ and $k_j = y_{+j}$, the site- and species-level correlations are equal and identical to the original fourth-corner correlation, but for other weighting schemes, these correlations may differ in value and even in sign. The correlations are combined by taking a signed minimum, or, if the two correlations differ in sign, by setting the final correlation to 0.

Appendix 9 contains R-code to compute the N_2-weigthed and unweighted variants of the fourth-corner correlation.

References

Birks, H.J.B., Line, J.M., Juggins, S., Stevenson, A.C. & ter Braak, C.J.F. (1990) Diatoms and pH reconstruction. *Philosophical Transactions of the Royal Society London, Series B,* **327,** 263-278. https://doi.org/10.1098/rstb.1990.0062

Dray, S. & Legendre, P. (2008) Testing the species traits environment relationships: The fourth-corner problem revisited. *Ecology,* **89,** 3400-3412. https://doi.org/10.1890/08-0349.1

Goodman, L.A. (1986) Some Useful Extensions of the Usual Correspondence Analysis Approach and the Usual Log-Linear Models Approach in the Analysis of Contingency Tables. *International Statistical Review,* **54,** 243-270. https://doi.org/10.2307/1403053

Ihm, P. & van Groenewoud, H. (1984) Correspondence analysis and Gaussian ordination. *Compstat Lectures,* **3,** 5-60. http://www.cfs.nrcan.gc.ca/publications?id=8638

Juggins, S. & Birks, H.J.B. (2012) Quantitative environmental reconstructions from biological data. *Tracking environmental change using lake sediments* (eds H.J.B. Birks, A.F. Lotter, S. Juggins & J.P. Smol), pp. 431-494. Springer, New York. https://doi.org/10.1007/978-94-007-2745-8_14

Peres-Neto, P.R., Dray, S. & ter Braak, C.J.F. (2017) Linking trait variation to the environment: critical issues with community-weighted mean correlation resolved by the fourth-corner approach. *Ecography,* **40,** 806-816.

Shipley, B. (2010) *From plant traits to vegetation structure : chance and selection in the assembly of ecological communities*. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511806971

ter Braak, C.J.F. (1985) Correspondence analysis of incidence and abundance data: properties in terms of a unimodal response model. *Biometrics,* **41,** 859-873. https://doi.org/10.2307/1938672

ter Braak, C.J.F. (1986) Canonical correspondence analysis: a new eigenvector technique for multivariate direct gradient analysis. *Ecology,* **67,** 1167-1179. https://doi.org/10.2307/1938672

ter Braak, C.J.F. (1987) The analysis of vegetation-environment relationships by canonical correspondence analysis. *Vegetatio,* **69,** 69-77. http://dx.doi.org/10.1007/BF00038688

ter Braak, C.J.F. (1988) Partial canonical correspondence analysis. *Classification and related methods of data analysis* (ed. H.H. Bock), pp. 551-558. Elsevier Science Publishers B.V. (North-Holland), Amsterdam. https://edepot.wur.nl/241165

ter Braak, C.J.F. (2017) Fourth-corner correlation is a score test statistic in a log-linear trait–environment model that is useful in permutation testing. *Environmental and Ecological Statistics,* **24,** 219-242. https://dx.doi.org/10.1007/s10651-017-0368-0

ter Braak, C.J.F. & Barendregt, L.G. (1986) Weighted averaging of species indicator values: its efficiency in environmental calibration. *Mathematical Biosciences,* **78,** 57-72. https://doi.org/10.1016/0025-5564(86)90031-3

ter Braak, C.J.F. & Looman, C.W.N. (1986) Weighted averaging, logistic regression and the Gaussian response model. *Vegetatio,* **65,** 3-11. https://doi.org/10.1007/BF00032121

ter Braak, C.J.F. & Verdonschot, P.F.M. (1995) Canonical correspondence analysis and related multivariate methods in aquatic ecology. *Aquatic Sciences,* **57,** 255-289. https://doi.org/10.1007/BF00877430

ter Braak, C.J.F., Cormont, A. & Dray, S. (2012) Improved testing of species traits–environment relationships in the fourth-corner problem. *Ecology,* **93,** 1525-1526. https://doi.org/10.1890/12-0126.1

ter Braak, C.J.F., Peres-Neto, P.R. & Dray, S. (2018a) Simple parametric tests for trait–environment association. *Journal of Vegetation Science,* **29,** 801-811. https://doi.org/10.7717/peerj.2885

ter Braak, C.J.F., Šmilauer, P. & Dray, S. (2018b) Algorithms and biplots for double constrained correspondence analysis. *Environmental and Ecological Statistics,* **25,** 171-197. https://doi.org/10.1007/s10651-017-0395-x

Walker, I.R., Mott, R.J. & Smol, J.P. (1991) Allerød Younger Dryas lake temperatures from midge fossils in Atlantic Canada. *Science,* **253,** 1010-1012. https://www.jstor.org/stable/2878795

Warton, D.I., Shipley, B. & Hastie, T. (2015) CATS regression – a model-based approach to studying trait-based community assembly. *Methods in Ecology and Evolution,* **6,** 389-398. https://doi.org/10.1111/2041-210X.12280

A4. Permutation testing and parametric bootstrap test

If there is no analytical alternative, statistical significance tests can be carried out by Monte Carlo testing, either using permutations or using bootstraps. If available, permutation tests are in my view preferable as they have an exact finite sampling theory (Hemerik & Goeman 2018). The mechanics of the Monte Carlo permutation test (permutation test for short) are described in Box A1. For approximate permutation tests in more complex situations see Anderson and Robinson (2001) and Dekker et al. (2007). For comparison and completeness, Box A2 describes the parametric bootstrap test.

In this paper, permutation tests are carried out by permuting the trait values $t_1, t_{2, ..., } t_m$ in any species-level test or the environmental values $e_1, e_{2, ..., } e_n$ in any site-level test. The reason is that species and sites are the relevant units for assessing trait-environment association.

In the WA-based methods, the site-level permutation test tests for the relation between CWM and **e**. The null hypothesis is that there is no relation between the CWM and **e**. Under the null hypothesis the units in **e** are therefore exchangeable and can be permuted. The inverse of the *p*-value issued from the (weighted) regression of CMW on **e** is used as test statistic. (In this simple case, many other test statistics, such as the slope or correlation from this regression are monotonic with the *p*-value and would have resulted in the same Monte Carlo *p*-value). Similarly, the species-level permutation test in the WA-based methods tests for the relation between SNC and **t**. The null hypothesis is that there is no relation between the SNC and **t**. Under the null hypothesis the units in **t** are therefore exchangeable and can be permuted. The inverse of the *p*-value issued from the (weighted) regression is used as test statistic.

In the model-based methods, the null hypothesis is about a single parameter in a complex model. Moreover, the single parameter is associated with the interaction between **t** and **e**. To the best of my knowledge, there exists no general exact permutation method for this situation. For a discussion of (approximate) methods that have been proposed for complex models see Manly (2006), Dekker et al. (2007) and Pesarin and Salmaso (2010). For testing trait-environment association in the MLM3 model I propose a new approach, again based on both a species-level and a site-level test. The species-level test tests the null hypothesis that the species-specific response ($\{c_j^*$ in Equation (2b) in the main text) are unrelated to the trait **t**, in the presence $\sigma_b > 0$, *i.e.* in the presence of site-specific response to the trait **t**. The site-level test tests the null hypothesis that the site-specific responses ($\{b_i^*\}$ in Equation (2a) in the main text) are unrelated to the environmental variable **e**, in the presence $\sigma_c > 0$, *i.e.* in the presence of species-specific response to the environmental variable **e**. The two tests are combined by taking the maximum of the two resulting *p*-values (max test).

A new model-based max test for trait-environment interaction

We assume exchangeability of the values in **t** and **e**. A random permutation of the values in **t** is denoted by $\mathbf{t}_{\boldsymbol{\pi}}$ with individual values $t_{\pi(j)}$ and a permutation of the values in **e** is denoted by $\mathbf{e}_{\boldsymbol{\pi}}$ with individual values $e_{\pi(i)}$. A max test consists of a species-level test and a site-level test (see Appendix A5).

Recall the derivation of the MLM3 model from the main text, starting from the model

$$L_{ij} = \alpha_0 + a_j + b_i^* t_j + c_j^* e_i + d_i \quad \text{(A27)}$$

to which then the submodels are added

$$b_i^* = \beta_0 + \beta'_{te} e_i + b_i \quad \text{(A28)}$$

$$c_j^* = \gamma_0 + \beta''_{te} t_j + c_j \quad \text{(A29)}$$

In the species-level test we wish to assess whether the species-specific responses $\{c_j^*\}$ in Equation (A29) are related to the trait **t**, in the presence $\sigma_b > 0$ in the MLM3 model, *i.e.* in the presence of site-specific response to the trait **t** (Equation (A28) with $\beta'_{te} = 0$). The null hypothesis of the species-level test is thus $H_0 : \beta''_{te} = 0$. However, a random permutation of **t**, $\mathbf{t}_\pi$ with individual values $t_{\pi(j)}$, does not only nullify the focal interaction term, represented by β''_{te} in Equation (A29), but also the trait main effect β_0 and the variance σ_b of the site-specific responses $\{b_i\}$ to the trait (Equation (A28)), which are part of the complete MLM models (σ_b is already 0 in MLM2). These issues can be solved by extending the data with the variable $\mathbf{t}_{\pi_0}$, where π_0 is a random permutation and not permuting the trait values involving the site-specific response to the trait in the model. The extended data is then analyzed with the (MLM2 or MLM3) model extended with a main effect for $\mathbf{t}_{\pi_0}$. In step 3 of Box A1 the test statistic F_0 for the data is calculated from the model (with, as in the main text, $y_{ij} \sim D(L_{ij})$ and the MLM3 term $(\beta_0 + b_i)t_j$ split into $\beta_0 t_j + b_i t_j$ and $b_i = 0$ in MLM2)

$$L_{ij} = \alpha_0 + d_i + a_j + \beta_0 t_j + \beta_0^* t_{\pi_0(j)} + b_i t_j + (\gamma_0 + c_j) e_i + \beta_{te} t_j e_i \quad \text{(A30)}$$

while in steps 4 and 5 of Box A1, the test statistic is computed from the model in which the t_j-values in the main and interaction terms are permuted:

$$L_{ij} = \alpha_0 + d_i + a_j + \beta_0 t_j + \beta_0^* t_{\pi(j)} + b_i t_j + (\gamma_0 + c_j) e_i + \beta_{te} t_{\pi(j)} e_i. \quad \text{(A31)}$$

In this way, the models for both the data and the permuted data contain the main effects of **t** and $\mathbf{t}_{\boldsymbol{\pi}}$ and also the same term for the site-specific response to the trait and the species-specific response to the environmental variable. The models in (A30) and (A31) are thus identical under H_0: $\beta_{te} = 0$.

The site-level test is analogous to the species-level test, with the pair (species, **t**) interchanged with (site, **e**). Thus, in the site-level test we wish to assess whether the site-specific responses $\{b_i^*\}$ in Equation (A28) are related to the environmental variable **e**, in the presence $\sigma_c > 0$, *i.e.* in the presence of species-specific response to the environmental variable **e** (Equation (A29) with $\beta''_{te} = 0$). The null hypothesis of the species-level test is thus $H_0 : \beta'_{te} = 0$. However, a random permutation of **e**, $\mathbf{e}_\pi$ with individual values $e_{\pi(j)}$, does not only nullify the focal interaction term, represented in Equation (A28) by β'_{te}, but also the environmental main effect γ_0 and the variance σ_c of the species-specific response to the environmental variable (Equation (A29)). These issues can be solved by extending the data with the variable $\mathbf{e}_{\pi_0}$, where π_0 is a random permutation and not permuting the environmental values involving the species-specific response to the environmental variable in the model. The extended data is then analyzed with the (MLM2 or MLM3) model extended with a main effect for $\mathbf{e}_{\pi_0}$. In step 3 of Box A1 the test statistic F_0 for the data is calculated from the model (with the term $(\gamma_0 + c_j)e_i$ split into $\gamma_0 e_i + c_j e_i$)

$$L_{ij} = \alpha_0 + d_i + a_j + \gamma_0 e_i + \gamma_0^* e_{\pi_0(i)} + c_j e_i + (\beta_0 + b_i) t_j + \beta_{te} t_j e_i \quad \text{(A32)}$$

is used, while in steps 4 and 5 of Box A1, the test statistic is computed from the model in which the e_i-values in the main and interaction terms are permuted:

$$L_{ij} = \alpha_0 + d_i + a_j + \gamma_0 e_i + \gamma_0^* e_{\pi(i)} + c_j e_i + (\beta_0 + b_i) t_j + \beta_{te} t_j e_{\pi(i)}. \qquad \text{(A33)}$$

In this way, the models for both the data and the permuted data contain the main effects of $\mathbf{e}$ and $\mathbf{e}_{\boldsymbol{\pi}}$ and also the same terms for the site-specific response to the trait and the species-specific response to the environmental variable. The models in (A32) and (A33) are thus identical under H_0: $\beta_{te} = 0$.

Box A1. Monte Carlo Permutation test
A Monte Carlo Permutation test is a statistical significance test to test a null hypothesis H_0 (such as $\beta_{te} = 0$) against an alternative hypothesis H_1 (such as $\beta_{te} \neq 0$). It carried out using random permutations according to the following steps: 1. Select a test statistic F that is sensitive to the departure of H_0 in the direction of H_1 and select a number K of permutations such a K = 49, 99 or 499 (for reasons shown below). 2. Select a group G of permutations that generates permuted data that are equally likely as the data under H_0 or, more formally, a group that does not change the distribution of F under H_0. 3. Compute the test statistic for the data, yielding the value F_0. 4. Select a random permutation from the group, apply it to the data and calculate the test statistic for the permuted data, yielding F_1. 5. Repeat step 4 K-1 times so as to yield additional values F_2, ..., F_K. 6. Compute the Monte Carlo significance level, *i.e.* compute the number of values F_0, F_1, F_2, ... , F_K that is greater than or equal to F_0 (this number is thus at least 1), and divide by K + 1. Denote the result by p. Remarks: • Under H_0, the data can also be considered as a random permutation (*e.g.* another order of the environmental values would have been observed equally likely). For this reason, the test statistic F_0 is added to the ones obtained after permutation, so that K is chosen as 1 less than a round number, such as 100. More formally, under H_0 the data also result from a permutation from the group (namely, the identity permutation). • Under H_0, the values of the test statistic F_0, F_1, F_2, ... , F_K are exchangeable, so that the Monte Carlo p-value is unbiased, irrespective of the number K. • The complete set of permutations must have group structure, the importance of which is set out in Hemerik and Goeman (2018). Here, the group is the complete set of permutations of the chosen units, either the units of the trait data (i.e. species) or the units of the environmental data (i.e. sites). As there is only a single trait and single environmental variable, either the trait values t_1, t_2, ..., t_m or the environmental values e_1, e_2, ..., e_n are randomly permuted. • The method of calculation of the test statistic in steps 3-5 must be exactly the same as the method used to calculate the test statistic for the data (yielding

F_0). This guarantees the exchangeability of the test statistics F_0, F_1, F_2, ... , F_K.

- The detail of the K+1 in step 6 prevents p-values from being zero; p-values should never be zero (Phipson & Smyth 2010).

Box A2. Parametric bootstrap test

The parametric bootstrap test is a statistical significance test to test a null hypothesis H_0 (such as $\beta_{te} = 0$) against an alternative hypothesis H_1 (such as $\beta_{te} \neq 0$). It carried out by simulating data from the null model according to the following steps:

1. Select a test statistic F that is sensitive to the departure of H_0 in the direction of H_1 and select a number K of permutations such a K = 49, 99 or 499 (for reasons shown below).
2. Fit the null model M_0 to the data, *i.e.* the model of the null hypothesis, here with $\beta_{te} = 0$.
3. Compute the test statistic for the data, yielding the value F_0.
4. Simulate a data set from the null model M_0, fit it to the simulated data and calculate the test statistic, yielding F_1.
5. Repeat step 4 K-1 times so as to yield additional values F_2, ..., F_K.
6. Compute the Monte Carlo significance level, *i.e.* compute the number of values F_0, F_1, F_2, ... , F_K that is greater than or equal to F_0 (this number is thus at least 1), and divide by K + 1. Denote the result by p.

Remarks:

- Under H_0, the data can also be considered as a bootstrap sample. For this reason, the test statistic F_0 is added to the ones obtained after permutation, so that K is chosen as 1 less than a round number, such as 100.
- With the assumption that the data under the null hypothesis are a random sample from model M_0, the values of the test statistic F_0, F_1, F_2, ... , F_K are exchangeable under H_0, so that the Monte Carlo p-value is unbiased, irrespective of the number K. If the assumption does not (approximately) hold true, the parametric bootstrap test lacks theoretical underpinning. For example, if the response distribution using in the fit of M_0 is binomial or Poisson, whereas the data have overdispersion, the parametric bootstrap test is likely to have inflated type I error rate.
- The method of calculation of the test statistic in steps 3-5 must be exactly the same as the method used to calculate the test statistic for the data (yielding F_0). This guarantees the exchangeability of the test statistics F_0, F_1, F_2, ... , F_K.
- The detail of the K+1 in step 6 prevents p-values from being zero; p-values should never be zero (Phipson & Smyth 2010).

References

Anderson, M.J. & Robinson, J. (2001) Permutation tests for linear models. *Australian & New Zealand Journal of Statistics,* **43,** 75-88. https://doi.org/10.1111/1467-842X.00156

Dekker, D., Krackhardt, D. & Snijders, T.A.B. (2007) Sensitivity of MRQAP Tests to Collinearity and Autocorrelation Conditions. *Psychometrika,* **72,** 563-581. https://doi.org/10.1007/s11336-007-9016-1

Hemerik, J. & Goeman, J. (2018) Exact testing with random permutations. *Test,* **27,** 811-825. https://doi.org/10.1007/s11749-017-0571-1

Manly, B.F.J. (2006) *Randomization and Monte Carlo methods in biology*. Chapman and Hall, London. ISBN 9781584885412

Pesarin, F. & Salmaso, L. (2010) *Permutation Tests for Complex Data: Theory, Applications and Software*. Wiley. ISBN: 978-0-470-51641-6

Phipson, B. & Smyth, G.K. (2010) Permutation P-values should never be zero: calculating exact P-values when permutations are randomly drawn. *Statistical Applications in Genetics and Molecular Biology,* **9,** Article39. https://doi.org/0.2202/1544-6115.1585

A5. The max test combining site- and species-level tests

In this paper, the *p*-values of the site- and species-level tests are combined by taking the maximum. Knowing about the issues in multiple testing, you might wonder whether this is a valid procedure and how it relates to multiple testing. First, note that the issue arises if one looks at the minimum of the *p*-values of the individual tests and compares that with the nominal level α (e.g. $\alpha = 0.05$). The Bonferoni adjustment then says that, with two tests, α should be divided by two, so as control the Type I error rate. The word "control" means that Type I error rate is guaranteed to be smaller than or equal to α.

In the max test, the maximum is taken and no division by two (or other adjustment) is needed. The rationale follows from sequential testing theory (Goeman & Solari 2010; ter Braak, Cormont & Dray 2012) as explained in Box A3.

In the case of trait-environment association, the Type I error rate is smaller than the nominal level if the null hypotheses of both the site-level and the species-level tests are true. The Type I error rate is equal to the nominal level, if only one of the null hypotheses is true.

A5.1 WA-based methods

In WA-based methods, the site-level null hypothesis is that CWM is unrelated to **e** and the species-level null hypothesis is that SNC is related to **t**. If both null hypotheses are rejected, there is thus statistical evidence that CWM and SNC are related to **e** and **t**, respectively. These relations are a pre-requisite for trait-environment association; they imply that species abundance is related to both **e** and **t**, but not necessarily that **e** and **t** have an interaction effect on abundance. If σ_b and/or σ_c (or, equivalently, β_{xt} and/or β_{ze}) are zero, the max text controls the Type I error rate, but if both σ_b and σ_c are nonzero, the max text may show some Type I error inflation. In this section, I present the theory why this happens. In Appendices A6-A8 I investigate by simulation how serious the inflation is for the three WA-based methods.

The max test was developed for testing trait-environment association with the rationale that there exists trait-environment association if both **t** and **e** are related to the species abundances **Y** (ter Braak, Cormont & Dray 2012). The null hypothesis was that either **t** or **e**, or both **t** and **e** are not linked to **Y**, leading naturally to the max test (Box A3). In simulations using a one dimensional Gaussian model the max test controlled the Type I error rate and had also power to detect trait-environment association when there was one. Later, it was shown by simulation (Peres-Neto, Dray & ter Braak 2017) and by theory (ter Braak 2017) that the max test based on the squared fourth-corner correlation targets interaction in log-linear models and is not sensitive to main effects. Recently it was shown that this permutation-based max test is equivalent with the max test combining the permutation-based *p*-values of weighted CWMr and SNCr using the site and species totals as weights, respectively, and permuting the environmental values in the weighted CWMr and the trait values in the weighted SNCr (ter Braak, Peres-Neto & Dray 2018).

Box A3. Max test (ter Braak, Cormont & Dray 2012)

Setup and assumptions

Two possibly correlated tests are applied to the same data, addressing different null hypotheses H_{01} and H_{02}. The joint null hypothesis is the composite: (H_{01} and H_{02}) or (H_{01} or H_{02}) are true and the alternative hypothesis is that both H_{01} and H_{02} are false. So both null hypotheses must be rejected to reject the joint null hypothesis.

In sequential testing, H_{01} is tested first at level α and, only if it is rejected (for p-value $p_1 < \alpha$), H_{02} is tested at level α second, resulting in p-value p_2. It is assumed that the individual tests are unbiased, *i.e.* have level α when applied individually). The joint null hypothesis is rejected when both $p_1 < \alpha$ and $p_2 < \alpha$, i.e. that $\max(p_1, p_2)<\alpha$ (max test).

Theorem

The overall type I error rate of the max test is at most α.

Proof

When the first test is judged significant, we either commit a type I error (if H_{01} is true with rate α) or we take the right decision when H_{01} is false, and we proceed with the second test. When the second test is judged significant, we either commit a type I error (if H_{02} is true with rate α) or take the right decision (if H_{02} is false). If H_{01} and H_{02} are both true, the overall type I error rate will be smaller than α (both H_{01} and H_{02} are falsely rejected; if the tests are independent, the actual rate is α^2); if either H_{01} or H_{02} is true, the overall type I error rate is α (either H_{01} or H_{02} is falsely rejected). Note that one can start with testing H_{02} first and then H_{01}, without any change in the outcome[2]. The same outcome is obtained by always carrying out both tests and judging significance if $\max(p_1, p_2)<\alpha$. We conclude that the max test has almost an type I error rate of α.

Remarks

- H_{01} and H_{02} can also address the same null hypothesis with different test procedures. Then, both H_{01} and H_{02} are either true or false. If true, the overall type I error rate will be equal to or smaller than α (equal if the tests are perfectly correlated, and otherwise smaller, *e.g.* α^2, if the tests are independent). This max test thus controls the type I error rate also, but may suffer in terms of power.
- The individual tests can be Monte Carlo tests (Goeman & Solari 2010).

The main issue to be understood is why (weighted or unweighted) CWMr is not sufficient to test for trait-environment association. The issue is that the CWM of any trait, *e.g.* **z** with CWM $\sum_{j=1}^{m} y_{ij} z_j / y_{i+}$, depends, at least to some small extent, on the environmental variable **e**, if the abundances **Y** $=\{y_{ij}\}$ are, at least to some small extent, related to any interaction term involving the environmental variable **e**. The dependence

[2] ter Braak et al. (2012) add an extra requirement here on the test statistic, but that requirement is redundant.

is bigger the stronger the **Y- e** relation[3]. Even a random trait interacting with **e** can therefore yield a significant CWMr.

Another way of seeing the first issue is that the calculation of a CWM can be considered as a half-iteration in the iterative algorithm for correspondence analysis, alias reciprocal averaging (ter Braak & Prentice 1988). The issue can thus be expected to be bigger the higher the correlation between **e** and the first CA-ordination. The issue exists in both parametric CWMr and the permutation version in which the environmental values $\{e_i\}$ are permuted (as in the max test for trait-environment association). The issue can be countered by permuting the trait values $\{t_j\}$ as proposed by Zelený and coworkers (Zelený & Schaffers 2012; Zelený 2016). However, the resulting test, called the ZS-modified test in ter Braak et al. (2018), is very conservative in the situation where it is most wanted, namely when both σ_b and σ_c are nonzero (ter Braak, Peres-Neto & Dray 2018).

The analogous issue exists in SNCr. The issue is that the SNC of any environmental variable, *e.g.* **x** with SNC $\sum_{i=1}^{n} y_{ij}x_i/y_{+\mathrm{j}}$, depends, at least to some small extent, on the trait **t**, if the abundances **Y** $=\{y_{ij}\}$ are, at least to some small extent, related to any interaction term involving the trait **t**, and the dependence is bigger the stronger the **Y-t** relation. A log-linear main effect of **t** is not an issue, but a logit-linear main effect may be an issue, as in the CWM-case. Even a random environmental variable can therefore yield a significant SNCr. The issue exists in both parametric SNCr and the permutation version in which the trait values $\{t_j\}$ are permuted (as in the max test for trait-environment association).

If the CWMr shows significance, there is thus evidence for the link **Y**$\leftrightarrow$ **e** and, if the SNCr shows significance, there is thus evidence for the link **Y**$\leftrightarrow$ **t**. If both show significance, it is thus possible that there is **t-e** association.

The WA-based max test controls the type I error rate in the MLM3 model if σ_b and/or σ_c (or, equivalently, β_{xt} and/or β_{ze}) are zero, for the following reason. Under the null hypothesis ($\beta_{te} = 0$), abundance does not depend on any interaction term involving the environmental variable **e** if $\sigma_c = \beta_{ze} = 0$. There is thus no unobserved trait interacting with **e** that gives spurious significance in CWMr. CWMr thus does not carry any spurious significance. A significant CWMr is thus true evidence against the null hypothesis $\beta_{te} = 0$. Informally said, in this case, "CWMr is only significant when **e** truly interacts with **t** (and not with another trait, as there isn't any)". There might be an unobserved environmental variable interacting with the trait ($\sigma_b > 0$ or $\beta_{tx} \neq 0$), that gives spurious significance in SNCr, but as the max test takes the maximum of the *p*-values, this does not have an effect on the Type I error control of the max test.

Reversely, if $\sigma_b = \beta_{tx} = 0$, abundance does not depend on any interaction term involving the trait **t**, except when $\beta_{te} \neq 0$ in which case **t** interacts with **e**. There is thus no unobserved environmental variable interacting with **t** that gives spurious significance in SNCr. SNCr thus does not carry any spurious significance. A significant SNCr is thus true evidence against the null hypothesis $\beta_{te} = 0$. Informally said, "SNCr is only significant when **t** truly interacts with **e** (and not with another environmental variable, as there isn't any)". There might be an unobserved trait interacting with the

[3] A log-linear main effect of **e** is not an issue because it factors out in the sample total y_{i+} and therefore cancels out via the division by y_{i+}; but, a logit-linear main effect cancels out only partially, particularly when some of the probabilities are large (say >0.3), so for logit-models main effects may be an issue (see section A3.3).

environmental variable ($\sigma_c > 0$ or $\beta_{ze} \neq 0$), that gives spurious significance in CWMr, but as the max test takes the maximum of the *p*-values, this does not have an effect on the Type I error control of the max test.

However, if both σ_b and σ_c are nonzero, both CWMr and SNCr carry spurious significance (**Y**↔ **e** and **Y**↔ **t**, but **e**↮t), and the max text may show some Type I error inflation. This scenario does not exist in the one-dimensional (Gaussian) model. The scenario where the max test may fail, was thus not detected until two-dimensional (Gaussian) models and simulations models such as in Appendix A8 were included in the empirical evaluation of the max test (ter Braak, Peres-Neto & Dray 2017; ter Braak, Peres-Neto & Dray 2018). By simulation I investigate how serious the Type I error rate inflation is and whether the inflation and power differ among the different WA-based methods.

A5.2 Model-based methods

In MLM-based methods, the site-level test tests the null hypothesis that the site-specific responses ($\{b_i^*\}$ in Equation (A28)) are unrelated to the environmental variable **e**, in the presence $\sigma_c > 0$, *i.e.* in the presence of species-specific response to the environmental variable **e**. The species-level test tests the null hypothesis that the species-specific response ($\{c_j^*\}$ in Equation (A29)) are unrelated to the trait **t**, in the presence $\sigma_b > 0$, *i.e.* in the presence of site-specific response to the trait **t**. The max test combines both tests by taking the maximum of the *p*-values. Both tests address the trait-environment interaction coefficient β_{te}. One may therefore wonder whether these two null hypotheses are identical (which would perhaps result in an overly conservative test, see first Remark in Box A3). However, in MLM3 they are not; the site-based test addresses submodel (A28) with H_1: β'_{te}=0 and the species-based test addresses submodel (A29) with H_2: β''_{te}=0.

MLM2 lacks submodel (A28); MLM3 is thus more strict in judging any trait-environment interaction than MLM2. The site-based component of the max test using a MLM2-based analysis may be able to correct for the missing site term when applied to data conforming to the MLM3 model.

Model-based max test versus bootstrap test: simulation results

A small simulation study was carried out to investigate the Type I error rate and power of the model-based max test (Boxes A1 and A3) and the more usual parametric bootstrap test (Box A2). Both tests use the inverse of the Wald test-based *p*-value as test statistic.

Six species and 100 sites with a log-linear model are simulated using the simulation model of Appendix A8. The parameters were set to $\sigma_b = \sigma_c$ = 0.5 with structured noise with coefficient $\beta_{z^*x^*} = 1$ (see Appendix A8). In this scenario with $\beta_{te} = 0$, the parametric MLM3 Wald test has an inflated Type I error rate. Using 250 simulated data sets, the rate is estimated as 0.127 (at the 0.05 level)[4]. The same data sets were also analyzed with model-based site and species permutation (as in section A5.2) and the

[4] Replacement of the Wald test by the likelihood ratio test reduced the Type I error rate inflation only slightly.

bootstrap (49 randomizations per test). Site and species permutation resulted in rejection rates of 0.108 and 0.031, showing that model-based site permutation only is not sufficient to control the Type I error rate. The model-based max test and the bootstrap test both had a rejection rate of 0.031 (compared to 0.042 for the max test using N_2-weighted CWM/SNC lm). The power was investigated by setting $\beta_{te} = 1$. The rejection rates were 0.61 and 0.70 for the model-based max test and the bootstrap test, respectively, compared to 0.55 for N_2-weighted CWM/SNC lm. The first two p-values are significantly different (p <0.01), the second and third are not (p =0.21), when tested with the exact binomial test.

This small simulation study suggests that (1) the model-based permutation test is not necessarily overly conservative; (2) the bootstrap test is robust against structured residual variation that is not included in the MLM3 model; (3) the bootstrap test is more powerful than the model-based permutation test (15% more rejections) and N_2-weighted CWM/SNC lm (28% more rejections), at the cost of more assumptions and (compared to N_2-weighted CWM/SNC lm) more computing time.

A5.3 A geometric view on the testing problem

The interaction between the single quantitative trait **t** and the single environmental variable **e** defines a one-dimensional subspace in the full (n-1)(m-1)-dimensional space of site×species interactions. One might argue that the error in the (n-1)(m-1)-1 dimensional space of such interactions (without the **t**×**e** interaction) is a sufficient basis for testing for trait-environment association. However, any structured and unstructured interactions are part of this space, but their total size is not very useful to judge the size of the **t**×**e** interaction.

The max test is motivated by considering two subspaces, (1) the (n-1)-dimensional space of the **t**×sites interactions (Equation (A12)) and (2) the (m-1)-dimensional space of the species×**e** interactions (Equation (A18)), the intersection of which is the one-dimensional **t**×**e** interaction. The variance components σ_b^2 and σ_c^2 of MLM3 measure the size of the interactions in these two subspaces. In the max test, the size of the **t**×**e** interaction is compared to the size of interactions in each of these two subspaces. The size of the interactions in the remaining (n-1)(m-1)-(n-1)-(m-1)+1-dimensional space is considered of less importance.

The statistical machinery underlying the test of **t**-**e** interaction in MLM3 evaluates the sizes of interactions in each of the spaces mentioned. The simulations in Figure A2 suggest that the standard error of $\hat{\beta}_{te}$ depends much more strongly on σ_b^2 and σ_c^2 than on the other structured and unstructured interactions.

References

Goeman, J.J. & Solari, A. (2010) The sequential rejection principle of familywise error control. *Annals of Statistics,* **38,** 3782-3810. https://doi.org/10.1214/10-AOS829

Peres-Neto, P.R., Dray, S. & ter Braak, C.J.F. (2017) Linking trait variation to the environment: critical issues with community-weighted mean correlation

resolved by the fourth-corner approach. *Ecography,* **40,** 806-816. https://doi.org/10.1111/ecog.02302

ter Braak, C.J.F. (2017) Fourth-corner correlation is a score test statistic in a log-linear trait–environment model that is useful in permutation testing. *Environmental and Ecological Statistics,* **24,** 219-242. https://dx.doi.org/10.1007/s10651-017-0368-0

ter Braak, C.J.F., Cormont, A. & Dray, S. (2012) Improved testing of species traits–environment relationships in the fourth-corner problem. *Ecology,* **93,** 1525-1526. https://doi.org/10.1890/12-0126.1

ter Braak, C.J.F., Peres-Neto, P. & Dray, S. (2017) A critical issue in model-based inference for studying trait-based community assembly and a solution. *PeerJ,* **5,** e2885. https://doi.org/10.7717/peerj.2885

ter Braak, C.J.F., Peres-Neto, P.R. & Dray, S. (2018) Simple parametric tests for trait–environment association. *Journal of Vegetation Science,* **29,** 801-811. https://doi.org//10.1111/jvs.12666

ter Braak, C.J.F. & Prentice, I.C. (1988) A theory of gradient analysis. *Advances in Ecological Research,* **18,** 271-317. https://doi.org/10.1016/S0065-2504(08)60183-X

Zelený, D. (2016) Bias in community-weighted mean analysis of plant functional traits and species indicator values. *bioRxiv*, 1-37. https://doi.org/10.1101/046946

Zelený, D. & Schaffers, A.P. (2012) Too good to be true: pitfalls of using mean Ellenberg indicator values in vegetation analyses. *Journal of Vegetation Science,* **23,** 419-431. http://dx.doi.org/10.1111/j.1654-1103.2011.01366.x

A6. Further analysis of simulation results for the Revisit data

A6.1 Further analysis

For the type I error rate and power simulations, I calculated the efficiency of the max test based on N_2-weighted regressions with respect to the Wald test based on MLM3, defined as the rejection rate at the 5% nominal level of the first method divided by that of the second method. I also calculated the agreement between the test results, defined as the percentage of simulated data sets for which both test outcomes results agreed at the 5% level. Concerning disagreement, I also calculated the percentage of data sets in which MLM3 showed significance and the WA method did not (M+N−) and the reverse case (M−N+). I also calculated the correlation between the logarithm of the p-values of the MLM3 Wald test and the max test based on N_2-weighted CWM/SNC regressions.

In simulations that focused on bias and variance of estimators, I also calculated the correlation between $\hat{\beta}_{te}$ and the variants of the fourth-corner correlations, defined in section 3.5 of Appendix A3.

A6.2 Results

Table A1 shows the additional results from the simulations based on the Revisit data. The relative power of N_2-weighted regressions with respect to MLM3 and the agreement are at least 0.82 and 78%, respectively. MLM2 and MLM3 showed little bias, except in the MLM2-based standard error. All correlations are >0.6 with one exception (0.22 in a case where all p-values are small).

Table A1. Measures of correspondence between WA- and MLM-based results in the simulations of Figure 4 based on the Revisit data in the main text (Agree and disagree are percentages of test outcomes at the 5% level, M= MLM3, N = N_2-w = N_2-weighted CWM/SNC lm, unw = unweighted, + = p<-0.05, - = p>0.05, FC = fourth-corner correlation, se= standard error, sd = standard deviation).

true		Relative power	Agree %	Disagree %		Bias in $\hat{\beta}_{te}$		% Bias se($\hat{\beta}_{te}$): w.r.t. sd of $\hat{\beta}_{te}$ in simulations		Correlations of p-value and variants of fourth-corner correlation with p-value and $\hat{\beta}_{te}$ from MLM3			
β_{te}	σ_b	$\frac{N_2\ w\ lm}{MLM3}$	$\frac{N_2\ w\ lm}{MLM3}$	M+ N-	M- N+	MLM2	MLM3	MLM2	MLM3	p-val N_2-w	FC N_2-w	FC	FC unw
0	0.34	0.92	96	2	2	-0.01	0.00	-15	3	0.65	0.80	0.69	0.76
0.1	0.34	0.93	87	7	6	-0.02	-0.01	-18	-2	0.81	0.80	0.72	0.78
0.2	0.34	0.82	78	15	6	-0.03	-0.02	-18	-1	0.76	0.78	0.72	0.76
0.3	0.34	0.91	87	10	2	-0.03	-0.02	-22	-6	0.72	0.77	0.72	0.77
0.4	0.34	0.94	94	6	0	-0.02	-0.01	-17	3	0.59	0.69	0.66	0.71
0.6	0.34	1.00	100	0	0	-0.03	-0.01	-23	-4	0.22	0.72	0.71	0.64
0	1.00	0.94	95	3	2	-0.01	0.00	-57	-7	0.75	0.90	0.86	0.86

A7. Analysis and simulations based on the Aravo data

A7.1 Introduction

Data analysis and simulations in the main text are confined to a single data set, the Revisit data, for which a betabinomial model with logit link is used. In this appendix the same analyses are carried out based on another data set, the aravo data set in the R package ade4 (Dray & Dufour 2007), for which Poisson and negative binomial models with log link are used. As this data shows negligible overdispersion, the simulations were based on the Poisson version of the MLM3 model. These simulations are also a test of how the N_2-weighted variant performs in a Poisson log-linear model, where the fourth-corner correlation is expected to perform well (ter Braak 2017).

A7.2 Material and methods

Data set

The data is the aravo data set in the R package ade4 (Dray & Dufour 2007). It has 82 plant species and 75 sites and derives from a study by Choler (2005) who investigated the shift in Alpine plant traits along a snow-melt gradient. Here we considered only one trait and one environmental variable, namely SLA (Specific Leaf Area) and Snow (mean snowmelt date in Julian day averaged over 1997-1999). The abundance data is recorded on a six-level scale based on the percentage of cover of each vascular plant. In ade4, the abundances are integer values 0-5.

Data analysis and simulations

The WA- and model-based methods were applied to the aravo data, resulting in estimates for the trait-environment interaction and parametric and permutation-based *p*-values for testing this interaction.

ter Braak et al. (2017) analysed the aravo data using a negative binomial distribution and did not detect any clear pattern that would suggest an ill-specified mean-variance relation. Initial analyses based on this distribution (called 'nbinom2' in glmmTMB) showed tiny overdispersion compared to the Poisson distribution. Therefore the subsequent analyses and simulations were based on the Poisson distribution. Because the glmmTMB library cannot fit models without overdispersion, we switched to the R library lme4 (Bates *et al.* 2015) to fit MLM2 and MLM3. The code for MLM3 was:

```
library(lme4)
fpol2 <- y~ poly(trait,2) + poly(env,2) + trait:env +
                              (1+trait|site)+ (1+env|species))
MLM3 <- glmer(fpol2, data=dat, family= poisson,
                   nAGQ=0, control = glmerControl(calc.derivs=F))
summary(MLM3)
```

The quadratic terms in this model are highly significant and were therefore kept in the subsequent analyses. The Poisson and negative binomial models have the log link by

default. The trait and environmental variables were standardized to zero mean and unit variance so as to get standardized estimates of β_{te}, σ_b and σ_c.

The resulting MLM3 model was used to simulate new data sets with the R code `simulate(MLM3)`, from which a parametric bootstrap confidence interval for β_{te} was constructed. Subsequently, the `MLM3` object was modified by changing the entry for $\hat{\beta}_{te}$ so as to simulate models without and with varying strengths of trait-environment interaction. Additional simulations were carried out to investigate the robustness of the methods to large site-specific trait effects, both in terms of type I error rate (for $\beta_{te} = 0$) and bias (for $\beta_{te} > 0$). For this, the entry for $\hat{\sigma}_b$ was increased to 1. Sometimes, species had no occurrence in the simulation data set and such species were removed before fitting the models.

A7.3 Results

Data set

With a squared main effects included, MLM3 gave a better fit than MLM2 ($\chi_2^2 = 40$, $p << 0.0001$). The coefficients of squared SLA and squared Snow variables were both negative and significant ($p < 0.01$). The trait- and environmental related standard deviations are $\hat{\sigma}_b = 0.25$ and $\hat{\sigma}_c = 1.47$, compared to $\hat{\sigma}_b = 0.34$ and $\hat{\sigma}_c = 0.42$ in the Revisit data.

The MLM3 estimate of β_{te} is 1.18 with a standard error of 0.18. Two thousand parametric bootstraps gave an almost normal distribution of $\hat{\beta}_{te}$ with mean 1.19 and standard deviation 0.18. The usual normality-based confidence interval of 1.18±2*0.18 = (0.82, 1.53) thus almost agrees with the bootstrap confidence interval. The usual fourth-corner correlation is 0.48, the N_2-weighted and unweighted correlations are 0.41 and 0.49, respectively.

The parametric and permutation-based *p*-values for testing the trait-environment interaction were <0.001 for all methods.

Simulations

Fig. A1 shows rejection rates (Type I error rate for $\beta_{te} = 0$ and power for $\beta_{te} > 0$) of each method, when reality is the same as for the MLM3 model with parameters from the fit to the Aravo data using a Poisson response distribution. For $\sigma_b = 1$, MLM2 and fourth corner show strong type I error rate inflation (rejection rates of 0.21 and 0.15, respectively). Of the methods that have good type I error control, MLM3 has the highest power, closely followed by lm CWM/SNC. The power of N_2-weighted regression is at least 70% of the power of MLM3 in these simulations. As in the Revisit data (Table A1), the correlation of $\hat{\beta}_{te}$ with the N_2-weighted fourth-corner correlation is higher than that with the usual fourth-corner correlation (Table A2).

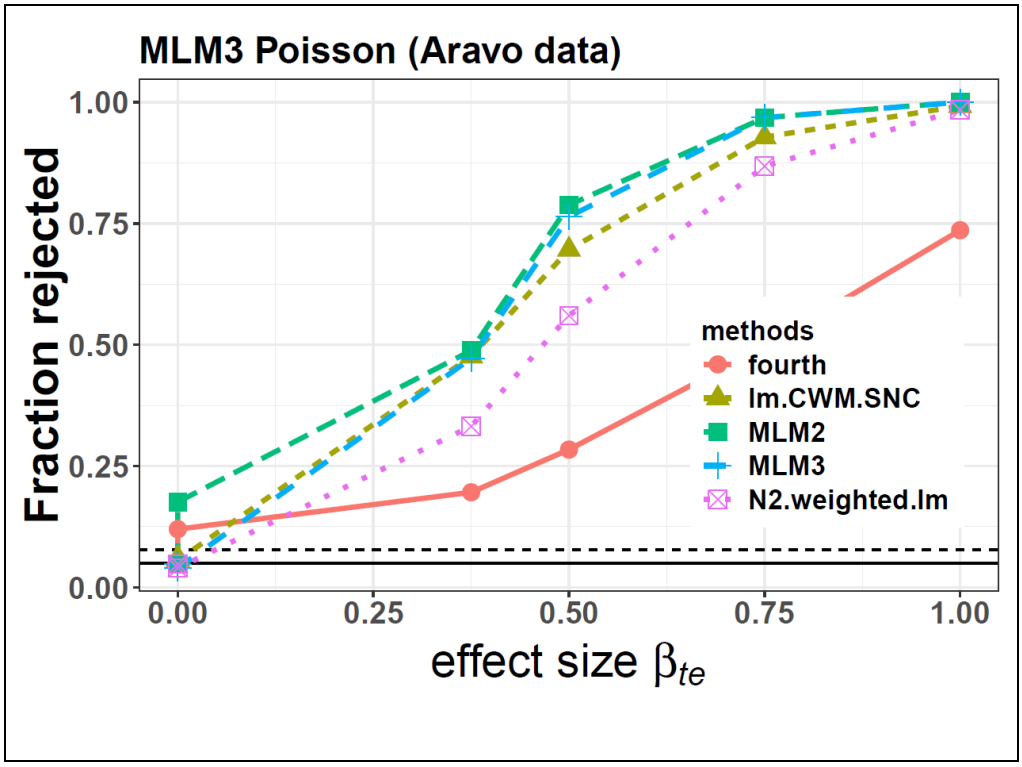


Fig. A1. Rejection rates of methods for data from the MLM3 model with parameters fit to the Aravo data using a Poisson response distribution except that β_{te} was set post-hoc to the value on the abscissa, based on 250 simulated data sets for each value of β_{te} (Wald test for MLM2 and MLM3 and permutation-based max test for the WA-based methods with F-value test statistic and 499 site and species permutations). For $\beta_{te} = 0$, additional simulations with $\sigma_b = 1$ were carried out and the lines start at this scenario so as to emphasize potential Type I error rate inflation. The horizontal solid line is at the nominal significance threshold; rates above the dashed line (at 0.078) are significantly greater than 0.05.

Table A2. Measures of correspondence between WA- and MLM-based results in the simulations based on the Aravo data in Figure A1 (Agree and disagree are fractions of test outcomes at the 5% level, M= MLM3, N = N_2-w = N_2-weighted CWM/SNC lm, unw = unweighted, + = p<-0.05, - = p>0.05, FC = fourth-corner correlation, w.r.t. = with respect to).

true		Relative power	Agree %	Disagree %		Bias $\hat{\beta}_{te}$		% Bias se($\hat{\beta}_{te}$): w.r.t. sd of $\hat{\beta}_{te}$ in simulations		Correlations of p-value and variants of fourth-corner correlation with p-value and $\hat{\beta}_{te}$ from MLM3			
β_{te}	σ_b	$\frac{N_2\ w\ lm}{MLM3}$	$\frac{N_2\ w\ lm}{MLM3}$	M+ N-	M- N+	MLM2	MLM3	MLM2	MLM3	p-val N_2-w	FC N_2-w	FC	FC unw
0	0.25	1.00	96	2	2	**-0.04**	-0.04	1	3	0.66	0.71	0.42	0.84
0.375	0.25	0.70	74	20	6	-0.01	-0.01	-4	1	0.77	0.71	0.42	0.72
0.5	0.25	0.73	73	24	3	-0.01	-0.01	-3	-1	0.75	0.63	0.38	0.68
0.75	0.25	0.90	88	11	1	0.00	0.00	-3	-1	0.64	0.71	0.42	0.74
1	0.25	0.98	98	2	0	0.01	0.01	1	4	0.31	0.59	0.23	0.65
0	1	1.00	96	2	2	**0.04**	-0.01	-28	0	0.55	0.77	0.35	0.75

References

Bates, D., Maechler, M., Bolker, B. & Walker, S.A.W. (2015) Fitting linear mixed-effects models using lme4. *Journal of Statistical Software,* **67,** 1-48. https://doi.org/10.18637/jss.v067.i01

Choler, P. (2005) Consistent Shifts in Alpine Plant Traits along a Mesotopographical Gradient. *Arctic, Antarctic, and Alpine Research,* **37,** 444-453. https://doi.org/10.18637/jss.v067.i01

Dray, S. & Dufour, A.B. (2007) The ade4 package: implementing the duality diagram for ecologists. *Journal of Statistical Software,* **22,** 1-20. https://doi.org/10.18637/jss.v022.i04

ter Braak, C.J.F. (2017) Fourth-corner correlation is a score test statistic in a log-linear trait–environment model that is useful in permutation testing. *Environmental and Ecological Statistics,* **24,** 219-242. https://dx.doi.org/10.1007/s10651-017-0368-0

ter Braak, C.J.F., Peres-Neto, P. & Dray, S. (2017) A critical issue in model-based inference for studying trait-based community assembly and a solution. *PeerJ,* **5,** e2885. https://doi.org/10.7717/peerj.2885

A8. Simulations of count and binomial data

A8.1 Introduction

The simulations in the main text and in Appendix A7 have the advantage that realistic parameter values are being used, as these as obtained from the data, but have the disadvantages that (1) the simulation model is identical to the analysis model (at least for MLM3) and (2) the number of species and sites cannot be varied freely. This appendix extends MLM3 with structured noise and gives simulation results using this extended model. The particular aims of these simulations are to investigate (1) whether MLM3 is sensitive to structured noise, (2) how it performs with small number of species and/or sites and (3) whether WA-based methods are able to control the type I error when both σ_b and σ_c are large (*i.e.* in a situation of concern for the max text as explained in section A5.1) .

A8.2 Material and methods

The simulation model

The simulation model adds structured and unstructured noise to the MLM3 model of Equation (A1) as in ter Braak et al. (2017). The structured noise is obtained from two other unobserved variables, the trait variable $\mathbf{z}^*$ and environmental variables $\mathbf{x}^*$, that interact with coefficient $\beta_{z^*x^*}$. Both are drawn as independent standard normals. The unstructured noise is obtained from a standard normal variable ϵ of length *nm* scaled to standard deviation σ_ϵ. Furthermore, the linear trait and environmental effects are replaced by general noisy functions, in the Equation (A34) represented by the symbols R_i and C_j. The details are as follows.

In the simulation model the expected abundance μ_{ij} of species *j* with trait value t_j in site *i* with environmental value e_i is

$$link(\mu_{ij}) = \alpha_0 + R_i + C_j + \beta_{te} t_j e_i + \beta_{tx} t_j x_i + \beta_{ze} z_j e_i + \beta_{z^*x^*} z_j^* x_i^* + \epsilon_{ij} \qquad \text{(A34)}$$

with

- α_0 is the intercept parameter, accounting for the overall mean abundance,
- R_i is the row main effect plus noise that is a function of *i* and/or e_i only,
- C_j is the column main effect plus noise that is a function of *j* and/or t_j only,
- $\beta_{te}, \beta_{ze}, \beta_{tx}$ and $\beta_{z^*x^*}$ are parameters that govern the importance of the associated interaction terms,
- $z_j \sim N(0,1)$ and $z_j^* \sim N(0,1)$, two multivariate normal random variables with $\text{cor}(z_j, z_j^*) = \rho_1$, but independent of t_j, that could represent two unobserved trait variables that are independent of the observed variable t_j.

- $x_i \sim N(0,1)$ and $x_i^* \sim N(0,1)$, two multivariate normal random variables with cor(x_i, x_i^*) = ρ_2, but independent of e_i, that could represent two unobserved environmental variables that are independent of the observed variable e_i.
- $\epsilon_{ij} \sim N(0, \sigma_\epsilon^2)$, an independent normal random variable (with variance σ_ϵ^2) that could represent unobserved full rank variation leading to overdispersion in the abundance beyond the overdispersion via the conditional distribution of y_{ij} given μ_{ij}.

Recall that in terms of a mixed model (Equation (3) of the main text and Equation (A3)) $\beta_{tx} = \sigma_b$ and $\beta_{ze} = \sigma_c$. Unless noted explicitly otherwise, we used in our simulations

- α_0 = log(30) with log link for count data and α_0= logit(0.3) with logit link for binomial data,
- $R_i = 0.05e_i - 0.1e_i^2 + \varepsilon_{ri}$ with ε_{ri} an independent normal random variable: $\varepsilon_{ri} \sim N(0, 0.01)$, giving a common Gaussian response component with optimum at 0.25 and tolerance 2.24,
- $C_j = 0.05t_j - 0.1t_j^2 + \varepsilon_{cj}$ with ε_{cj} an independent normal random variable: $\varepsilon_{cj} \sim N(0, 0.01)$, giving a common Gaussian response component with optimum at 0.25 and tolerance 2.24,
- $\sigma_\epsilon = 0.2$, giving unstructured interaction among species and sites that is unrelated to the trait-environment association between $\{t_j\}$ and $\{e_i\}$.
- $\rho_1 = \rho_2 = 0$, making the structured noise ($\mathbf{x}^*$ and $\mathbf{z}^*$) independent of the random effects ($\mathbf{x}$ and $\mathbf{z}$).

The site and species random effects d_i and a_j in Equation (A3) are represented by ε_{ri} and ε_{cj} in the simulation model; these are uncorrelated with x_i and z_j, whereas they can be correlated in the MLM3 model as fitted to data. They are small compared to those in the Revisit and aravo data.

For count data, the abundance y_{ij} of species *j* in site *i* is drawn from a negative-binomial distribution with mean parameter μ_{ij}, overdispersion parameter 1, giving variance function $\mu_{ij} + \mu_{ij}^2$. For binomial data, the abundance y_{ij} is drawn from a binomial distribution with total 100 (as in the Revisit data) and probability μ_{ij}.

For the purpose of trait-environment association the key parameters are $\beta_{te}, \beta_{ze}, \beta_{tx}$ and $\beta_{z^*x^*}$, and the most important distinction is whether $\beta_{te} = 0$ or $\beta_{te} \neq 0$. If $\beta_{te} = 0$, then the trait and the environment may influence abundance through the row and column effects R_i and C_j, but there is no trait-environment association. However, if $\beta_{te} \neq 0$ then there is trait-environment association. With the simulations, we investigated whether the statistical methods under investigation can measure the trait-environment association and whether the associated significance tests can detect it, in particular whether these tests are valid and useful to test the null hypothesis H_0: $\beta_{te} = 0$ versus the alternative hypothesis H_1: $\beta_{te} \neq 0$. For validity, the tests should control the type I error at rate α, taken as $\alpha = 0.05$, that is, they should lead to at most 100 α% rejections of the null hypothesis if in fact H_0: $\beta_{te} = 0$ holds (closer to 100 α% is better). For usefulness, they should have sufficient power, that is, they should lead to a large fraction of rejections under the alternative hypothesis (larger is better).

Simulation scenarios

In two series of simulations the number of species (m) and sites (n) were varied: 10, 50 and 100 species or sites in a log-linear model series (series I) and 15, 50 and 100 in a logit model series (series II). In these series all nuisance interaction coefficients (all except β_{te}) were set to 0.5, unless noted otherwise. These settings made the MLM3 model symmetric in species and sites, so that only six of the nine possible combinations of n and m are presented. To investigate whether the methods are robust against correlation between the random effect and the structured noise, these series were repeated with a correlation of 0.5 between $\mathbf{x}$ and $\mathbf{x}^*$ and between $\mathbf{z}$ and $\mathbf{z}^*$ ($\rho_1 = \rho_2 = 0.5$). For the study of the MLM estimators of β_{te} and their variance, these series were also repeated with all nuisance interaction coefficients set to 1.

These series revealed some type I error rate inflation in the Wald test of MLM3 for n=10 or m=10. In a third series (series III), this is further investigated with n, m = 5, 10, 20 for log-linear models and with $\beta_{z^*x^*} = 0$ to see whether the inflation is still present without structured noise.

In the fourth series (series IV), the WA-based methods are compared in more detail. In this series, the nuisance interaction coefficients were varied: β_{tx} , β_{ze} and $\beta_{z^*x^*}$ were either 0 or 0.5. This series was to falsify the claim that the WA-based max test always controls the type I error rate when either β_{tx} or β_{ze} is zero or both are zero, and to study the type I error rate inflation when both β_{tx} and β_{ze} are nonzero.

In a fifth series (series V), the model-based methods are studied in more detail in terms of bias in either the estimate of β_{te} and its standard error. In this series, with n=50 and m=30 and $\beta_{te} = 0$, the parameters β_{tx} , β_{ze} and $\beta_{z^*x^*}$ are either 0 or 0.5, giving eight scenarios. A similar series with $\beta_{te} = 0.3$, which led to similar conclusions as series V, is not shown.

In each scenarios the number of simulated data sets was 250 and the number of permutations in WA-based methods was 499. All methods were applied to the same data sets and the same permutations. The formula used in fitting MLM2 was

```
fpol2MLM2 <- y~ poly(trait,2) + poly(env,2) + trait:env +
                              (1|site)+ (1+env|species))
```

and in fitting MLM3

```
fpol2MLM3 <- y~ poly(trait,2) + poly(env,2) + trait:env +
                              (1+trait|site)+ (1+env|species))
```

The results were analyzed as in section AA6.1.

A8.3 Results

Type I error rates and power

In Series I-III (with nonzero nuisance interaction coefficients: Tables A3-A5) MLM2 shows considerable type I error rate inflation (rejection rates between 0.10 and 0.53). MLM3 and N_2- weighted lm perform much better with moderate inflation (up to 0.12). The inflation occurs with small n or m for MLM3 and with large n and m for N_2-weighted lm. This inflation could not be made smaller by changing from the Ward test to the LRT test; it might be removed by resampling using the bootstrap test or the model-based max test (section A5.2). The type I error inflation in N_2-weighted lm is generally somewhat lower than in fourth corner and lm CWM/SNC (Tables A3-A6).

The results for simulations with and without correlation between the random effects and the structure noise are rather similar (Tables A3a,b and Tables A4a,b), except that, with correlation, MLM2 has even higher type I error rate inflation than without correlation, and the WA-methods appear to have somewhat higher type I error inflation, for large n and m, with than without correlation.

In Series I-II the relative power of N_2-weighted lm with respect to MLM3 ranges from 0.70 to 1.05. For small n or m, N_2-weighted lm is also somewhat less powerful than fourth corner and lm CWM/SNC (relative power range: 0.62 to 0.96); for larger n or m, N_2-weighted lm is only less powerful when fourth corner and lm CWM/SNC do not control the type I error rate.

The agreement between test results of MLM3 and N_2-weighted lm ranges between 75 and 98%; the disagreement percentages reflect the differences between the two methods in type I error rates and power for null models. The correlation between the log p-values of the methods ranges between 0.80 and 0.97, except when the power of methods is above 0.9 (so that most of the p-values are small or very small; the number of permutations limits the p-values of the WA-methods to be above 0.002, whereas MLM3 has no limit).

In Series IV, all three WA-based methods fully controlled the type I error rate when either $\beta_{tx} = 0$ or $\beta_{ze} = 0$ (Table A6a). When both $\beta_{tx} \neq 0$ and $\beta_{ze} \neq 0$ the methods showed some inflation, up to a rejection rate of 0.10. Out of the 10, the numbers of scenarios with inflation are 6, 5 and 2 for fourth corner, lm CWM/SNC and N_2-weighted lm (Table A6a). Increasing the nuisance interactions from 0.5 to 1 led to lower (or no) Type I error inflation (as can be seen by comparing the first three versus the last three rows of the second halve of Table A6a). The somewhat better type I error rate control of N_2-weighted lm led to slightly less power (Table A6b).

Table A3. Series I: rejection rates of, and agreement between, four methods to detect trait-environment interaction for simulated negative binomial count data with $\beta_{tx} = \beta_{ze} = \beta_{z^*x^*} = 0.5$. (Agree and disagree are percentages of test outcomes at the 5% level, M= MLM3, N = N_2-w = N_2-weighted CWM/SNC lm, unw = unweighted, + = p<-0.05, - = p>0.05, FC = fourth-corner correlation, Cor = correlation between p-values of N_2-weighted CWM/SNC lm and MLM3). Inflated type I error rates (values above 0.078) are shown in bold, and the corresponding (relative) power in gray; for null models, all relative rejection fractions are grayed (250 simulations for each scenario; MLM uses the log-link with negative binomial response distribution).

Table A3a. Structured noise ($\mathbf{x}^*$ and $\mathbf{z}^*$) independent of the random effects ($\mathbf{x}$ and $\mathbf{z}$).

sites	species	Fraction rejected					Relative power	Agree %	Disagree %		Cor
n	*m*	Fourth corner	lm CWM/ SNC	MLM2	MLM3	N_2-weighted lm	$\frac{N_2\ w\ lm}{MLM3}$	$\frac{N_2\ w\ lm}{MLM3}$	M+ N-	M- N+	*p*-val N_2-w, MLM3
Null models: $\beta_{te} = 0$											
10	10	0.044	0.036	**0.160**	**0.084**	0.032	0.38	95	5	0	0.86
10	50	0.068	0.072	**0.124**	**0.096**	0.060	0.63	95	4	1	0.85
10	100	0.036	0.036	**0.116**	**0.092**	0.032	0.35	93	6	0	0.87
50	50	**0.100**	**0.108**	**0.180**	0.052	0.064	1.23	96	1	2	0.88
50	100	0.076	**0.088**	**0.112**	0.052	0.072	1.38	97	0	2	0.89
100	100	**0.132**	**0.120**	**0.188**	0.056	0.056	1.00	95	2	2	0.89
Non-null models: $\beta_{te} = 0.25$											
10	10	0.120	0.108	0.326	0.204	0.104	0.51	90	10	0	0.91
10	50	0.200	0.196	0.376	0.352	0.176	0.50	82	18	0	0.86
10	100	0.180	0.188	0.360	0.328	0.156	0.48	83	17	0	0.88
50	50	0.640	0.716	0.820	0.652	0.624	0.96	91	6	3	0.85
50	100	0.760	0.844	0.888	0.808	0.744	0.92	90	8	2	0.80
100	100	0.916	0.952	0.972	0.924	0.916	0.99	96	2	2	0.72

Table A3b. Structured noise ($\mathbf{x}^*$ and $\mathbf{z}^*$) correlated with the random effects ($\mathbf{x}$ and $\mathbf{z}$) with a correlation of 0.5 between $\mathbf{x}$ and $\mathbf{x}^*$ and between $\mathbf{z}$ and $\mathbf{z}^*$ ($\rho_1 = \rho_2 = 0.5$).

sites	species	Fraction rejected					Relative power	Agree %	Disagree %		Cor
n	*m*	Fourth corner	lm CWM/ SNC	MLM2	MLM3	N_2-weighted lm	$\frac{N_2\ w\ lm}{MLM3}$	$\frac{N_2\ w\ lm}{MLM3}$	M+ N-	M- N+	*p*-val N_2-w, MLM3
Null models: $\beta_{te} = 0$											
10	10	0.056	0.060	**0.231**	**0.104**	0.056	0.54	94	5	0	0.84
10	50	0.052	0.048	0.431	**0.088**	0.044	0.50	95	5	0	0.85
10	100	**0.084**	**0.088**	**0.528**	**0.120**	0.052	0.43	92	8	1	0.89
50	50	**0.100**	**0.100**	**0.200**	0.048	0.064	1.33	95	2	3	0.80
50	100	**0.084**	**0.072**	**0.232**	0.044	0.056	1.27	96	1	2	0.81
100	100	**0.148**	**0.144**	**0.196**	0.060	**0.104**	1.73	94	1	5	0.79
Non-null models: $\beta_{te} = 0.25$											
10	10	0.132	0.140	0.414	0.224	0.116	0.52	87	12	1	0.84
10	50	0.264	0.284	0.714	0.348	0.208	0.60	86	14	0	0.87
10	100	0.248	0.284	0.820	0.412	0.160	0.39	75	25	0	0.87
50	50	0.732	0.748	0.836	0.668	0.568	0.85	87	12	2	0.83
50	100	0.812	0.852	0.928	0.788	0.644	0.82	82	16	2	0.80
100	100	0.940	0.980	0.984	0.948	0.872	0.92	92	8	0	0.70

Table A4. Series II: rejection rates of, and agreement between, four methods to detect trait-environment interaction for simulated binomial count data with binomial total of 100 and $\beta_{tx} = \beta_{ze} = \beta_{z^*x^*} = 0.5$. (Agree and disagree are percentages of test outcomes at the 5% level, M= MLM3, N = N_2-w = N_2-weighted CWM/SNC lm, unw = unweighted, + = p<-0.05, - = p>0.05, FC = fourth-corner correlation, Cor = correlation between p-values of N_2-weighted CWM/SNC lm and MLM3). Inflated type I error rates (values above 0.078) are shown in bold, and the corresponding (relative) power in gray; for null models, all relative rejection fractions are grayed (250 simulations for each scenario; MLM uses the logit-link with beta-binomial response distribution).

Table A4a. Structured noise ($\mathbf{x}^*$ and $\mathbf{z}^*$) independent of the random effects ($\mathbf{x}$ and $\mathbf{z}$).

sites	species	Fraction rejected					Relative power	Agree %	Disagree %		Cor
n	*m*	Fourth corner	lm CWM/ SNC	MLM2	MLM3	N_2-weighted lm	$\frac{N_2\ w\ lm}{MLM3}$	$\frac{N_2\ w\ lm}{MLM3}$	M+ N-	M- N+	*p*-val N_2-w, MLM3
Null models: $\beta_{te} = 0$											
15	15	0.060	0.048	**0.170**	0.064	0.052	0.81	98	2	0	0.96
50	15	0.068	0.064	**0.120**	0.068	0.064	0.94	97	2	1	0.97
100	15	0.064	0.060	**0.109**	**0.088**	0.056	0.64	95	4	1	0.96
50	50	**0.112**	**0.092**	**0.200**	0.040	**0.092**	2.30	95	0	5	0.97
100	50	**0.088**	**0.080**	**0.112**	0.044	0.068	1.55	98	0	2	0.96
100	100	**0.108**	**0.100**	**0.164**	0.056	**0.096**	1.71	96	0	4	0.97
Non-null models: $\beta_{te} = 0.25$											
15	15	0.288	0.276	0.512	0.272	0.264	0.97	94	4	3	0.90
50	15	0.332	0.320	0.536	0.376	0.300	0.80	90	9	1	0.87
100	15	0.400	0.380	0.541	0.504	0.360	0.71	86	14	0	0.89
50	50	0.780	0.780	0.862	0.676	0.756	1.12	90	1	9	0.83
100	50	0.836	0.828	0.884	0.776	0.816	1.05	94	1	5	0.81
100	100	0.964	0.964	0.972	0.920	0.944	1.03	97	0	3	0.62

Table A4b. Structured noise ($\mathbf{x}^*$ and $\mathbf{z}^*$) correlated with the random effects ($\mathbf{x}$ and $\mathbf{z}$) with a correlation of 0.5 between $\mathbf{x}$ and $\mathbf{x}^*$ and between $\mathbf{z}$ and $\mathbf{z}^*$ ($\rho_1 = \rho_2 = 0.5$).

sites	species	Fraction rejected					Relative power	Agree %	Disagree %		Cor
n	*m*	Fourth corner	lm CWM/ SNC	MLM2	MLM3	N_2-weighted lm	$\frac{N_2\ w\ lm}{MLM3}$	$\frac{N_2\ w\ lm}{MLM3}$	M+ N-	M- N+	*p*-val N_2-w, MLM3
Null models: $\beta_{te} = 0$											
15	15	0.068	0.076	**0.214**	0.064	0.072	1.12	98	1	2	0.91
50	15	0.076	0.084	**0.116**	**0.088**	0.072	0.82	98	2	0	0.95
100	15	0.032	0.036	**0.094**	0.064	0.032	0.50	96	4	0	0.96
50	50	**0.104**	**0.092**	**0.194**	0.044	**0.092**	2.09	95	0	5	0.96
100	50	**0.084**	**0.088**	**0.128**	0.056	**0.088**	1.57	96	0	4	0.97
100	100	**0.128**	**0.128**	**0.180**	0.064	**0.124**	1.94	94	0	6	0.96
Non-null models: $\beta_{te} = 0.25$											
15	15	0.252	0.252	0.410	0.264	0.252	0.95	96	3	2	0.92
50	15	0.372	0.360	0.502	0.428	0.336	0.79	91	9	0	0.93
100	15	0.320	0.336	0.484	0.420	0.296	0.70	87	13	0	0.92
50	50	0.752	0.740	0.859	0.636	0.716	1.13	90	1	9	0.85
100	50	0.868	0.844	0.916	0.792	0.808	1.02	94	2	4	0.81
100	100	0.972	0.968	0.980	0.920	0.956	1.04	96	0	4	0.64

Table A5. Series III: rejection rates of, and agreement between, four methods to detect trait-environment interaction for simulated negative binomial count data with $\beta_{tx} = \beta_{ze} = 0.5$ and $\beta_{z^*x^*} = 0$. (LRT = likelihood ratio test; - failed). Inflated type I error rates (values above 0.078) are shown in bold, and the corresponding (relative) power in gray; for null models, all relative rejection fractions are grayed (250 simulations for each scenario; MLM uses the log-link with negative binomial response distribution).

sites	species	Fraction rejected					
n	*m*	Fourth corner	lm CWM/ SNC	MLM2	MLM3	MLM3 LRT	N_2-weighted lm
Null models: $\beta_{te} = 0$							
5	5	0.012	0.012	**0.199**	**0.159**	-	0.012
5	10	0.028	0.032	**0.202**	**0.153**	-	0.024
5	20	0.044	0.048	**0.299**	**0.164**	-	0.048
10	10	0.028	0.060	**0.229**	**0.116**	-	0.060
10	20	0.036	0.040	**0.177**	**0.080**	**0.112**	0.040
20	20	0.048	0.072	**0.185**	**0.010**	**0.099**	0.072

Table A6. Series IV: rejection rates of three WA-based methods for $\beta_{te} = 0$ (a) and $\beta_{te} = 0.375$ (b) to detect trait-environment interaction for simulated binomial count data with $n = m$=50 and binomial total of 100. Inflated type I error rates (values above 0.078) are shown in bold, and the corresponding (relative) power in gray; for null models, all relative rejection fractions are grayed (250 simulations for each scenario).

Table A6a Null models: $\beta_{te} = 0$

(a)			Fraction rejected			Relative power w.r.t N_2-weighted lm	
β_{tx} σ_b	β_{ze} σ_c	$\beta_{z^*x^*}$	Fourth corner	lm CWM/ SNC	N_2-weighted lm	$\frac{FC}{N_2\ w\ lm}$	$\frac{lm\ C/S}{N_2\ w\ lm}$
Either $\beta_{tx} = 0$ or $\beta_{ze} = 0$							
0	0	0	0.042	0.042	0.042	1.00	1.00
0	0	0.5	0.004	0.004	0.006	0.67	0.67
0	0.5	0	0.058	0.062	0.058	1.00	1.07
0	0.5	0.5	0.044	0.046	0.048	0.92	0.96
0	1	0	0.054	0.050	0.052	1.04	0.96
0	1	0.5	0.060	0.054	0.060	1.00	0.90
0.5	0	0	0.038	0.040	0.036	1.06	1.11
0.5	0	0.5	0.054	0.060	0.056	0.96	1.07
1	0	0	0.042	0.050	0.054	0.78	0.93
1	0	0.5	0.040	0.044	0.052	0.77	0.85
Both $\beta_{tx} \neq 0$ and $\beta_{ze} \neq 0$							
0.5	0.5	0	**0.092**	**0.094**	**0.090**	1.02	1.04
0.5	0.5	0.5	**0.098**	**0.100**	**0.084**	1.17	1.19
0.5	0.5	1	**0.080**	**0.074**	0.066	1.21	1.12
0.5	1	0	0.058	0.060	0.048	1.21	1.25
0.5	1	0.5	**0.080**	0.072	0.066	1.21	1.09
1	0.5	0	0.070	0.070	0.056	1.25	1.25
1	0.5	0.5	0.058	0.060	0.058	1.00	1.03
1	1	0	**0.090**	**0.084**	0.070	1.29	1.20
1	1	0.5	**0.088**	**0.086**	0.076	1.16	1.13
1	1	1	0.074	0.072	0.064	1.16	1.13

Table A6b Non-null models: $\beta_{te} = 0.375$

(b)			Fraction rejected			Relative power	
β_{tx}	β_{ze}	$\beta_{z^*x^*}$	Fourth corner	lm CWM/ SNC	N_2-weighted lm	$\frac{FC}{N_2\ w\ lm}$	$\frac{lm\ C/S}{N_2\ w\ lm}$
Either $\beta_{tx} = 0$ or $\beta_{ze} = 0$							
0	0	0	1.000	1.000	1.000	1.00	1.00
0	0	0.5	1.000	1.000	1.000	1.00	1.00
0	0.5	0	1.000	1.000	1.000	1.00	1.00
0	0.5	0.5	0.992	0.996	1.000	0.99	1.00
0	1	0	0.676	0.664	0.624	1.08	1.06
0	1	0.5	0.716	0.704	0.656	1.09	1.07
0.5	0	0	0.996	0.996	0.996	1.00	1.00
0.5	0	0.5	0.996	0.996	0.992	1.00	1.00
1	0	0	0.676	0.660	0.652	1.04	1.01
1	0	0.5	0.708	0.696	0.680	1.04	1.02
Both $\beta_{tx} \neq 0$ and $\beta_{ze} \neq 0$							
0.5	0.5	0	0.984	0.980	0.976	1.01	1.00
0.5	0.5	0.5	0.976	0.972	0.968	1.01	1.00
0.5	0.5	1	0.968	0.968	0.964	1.00	1.00
0.5	1	0	0.660	0.644	0.628	1.05	1.03
0.5	1	0.5	0.660	0.656	0.648	1.02	1.01
1	0.5	0	0.636	0.608	0.608	1.05	1.00
1	0.5	0.5	0.588	0.568	0.568	1.04	1.00
1	1	0	0.520	0.480	0.448	1.16	1.07
1	1	0.5	0.488	0.472	0.456	1.07	1.04
1	1	1	0.524	0.492	0.480	1.09	1.03

Estimation bias and standard error

In Series V (with n=50 and m=30), MLM2 estimates β_{te} without visible bias in log-linear null models, but underestimates the standard error of $\hat{\beta}_{te}$ when there are sizeable random effects; no bias is visible in MLM3 (Fig. A2). In MLM3, the standard error of $\hat{\beta}_{te}$ increases sharply with the size of the variance components σ_b^2 and σ_c^2 but only slightly with the size the structured noise ($\beta_{z^*x^*}$); MLM2 fails to pick up the variance in σ_b^2, as it does not include this random component (Fig. A2).

Series I and II (Tables A7 and A8) show almost no bias in $\hat{\beta}_{te}$ in both MLM2 and MLM3, except in logit-models in non-null models with large nuisance interactions (Tables A8c). This bias in logistic non-null models disappears in MLM3, but not in MLM2, when the parameter governing residual correlation ($\beta_{z^*x^*}$) is set to zero (Table A8d).

Series I and II (Tables A7 and A8) show large downward bias in the estimator of the standard error in MLM2 and, for small n or m, some downward bias in MLM3.

The correlations between the MLM3 estimator of the interaction and the N_2-weighted fourth-corner correlation ranges from 0.88 to 0.98 in the Series I and II and from 0.70 to 0.97 in Series I and II with the nuisance interactions set to 1. The correlations with the original fourth-corner correlations are generally somewhat lower.

Fig. A2. Series V: $\hat{\beta}_{te}$ (est. of β_{te}) and its standard error (s.e.) given by MLM2 and MLM3 in simulated null models ($\beta_{te} = 0$) with varying amounts of sizes of random effects (n = 50, m = 30; log-linear model with negative binomial response distribution).

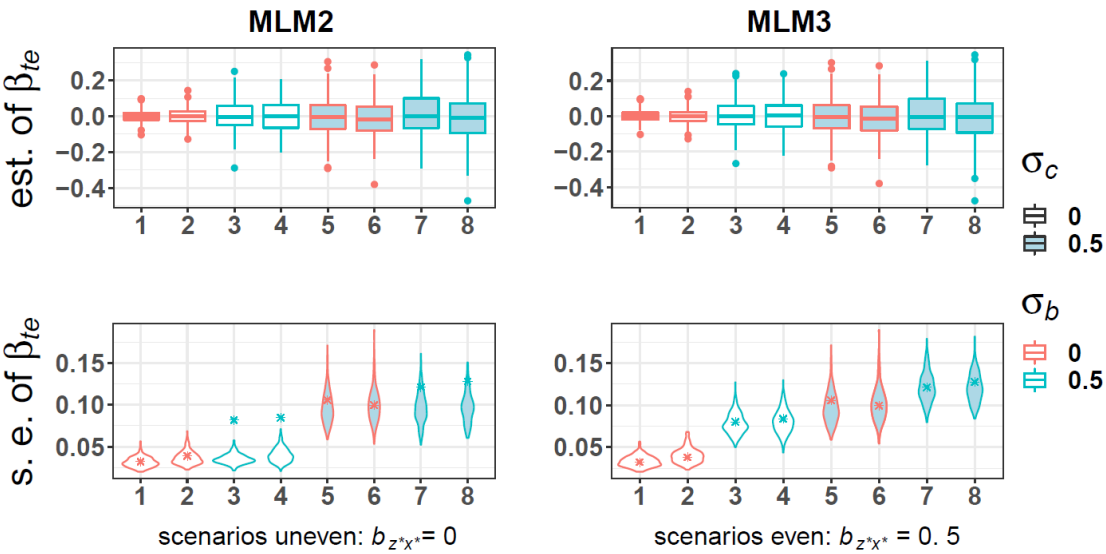


Table A7. Series I: Bias in $\hat{\beta}_{te}$ and its standard error given by MLM2 and MLM3, and the correlation between the MLM3 estimate of β_{te} and three variants of the fourth-corner correlation in simulated negative binomial count data with $\beta_{tx} = \beta_{ze} = \beta_{z^*x^*} = 0.5$. (Cor = correlation; FC = fourth-corner correlation, N_2-w = N_2-weighted, unw = unweighted; 250 simulations for each scenario; MLM uses the log-link with negative binomial response distribution).

(A7a) Structured noise ($\mathbf{x}^*$ and $\mathbf{z}^*$) independent of the random effects ($\mathbf{x}$ and $\mathbf{z}$).

(A7b). Structured noise ($\mathbf{x}^*$ and $\mathbf{z}^*$) correlated with the random effects ($\mathbf{x}$ and $\mathbf{z}$) with a correlation of 0.5 between $\mathbf{x}$ and $\mathbf{x}^*$ and between $\mathbf{z}$ and $\mathbf{z}^*$ ($\rho_1 = \rho_2 = 0.5$).

(A7c). Series I with large nuisance interactions ($\beta_{tx} = \beta_{ze} = \beta_{z^*x^*} = 1$).

Table A7a. Structured noise ($\mathbf{x}^*$ and $\mathbf{z}^*$) independent of the random effects ($\mathbf{x}$ and $\mathbf{z}$).

sites	species	Bias in $\hat{\beta}_{te}$		Bias in se($\hat{\beta}_{te}$): % of sd of $\hat{\beta}_{te}$ in simulations		Cor of variants of FC correlation with $\hat{\beta}_{te}$ from MLM3		
n	*m*	MLM2	MLM3	MLM2	MLM3	FC N_2-w	FC	FC unw
Null models: $\beta_{te} = 0$								
10	10	0.03	0.04	-26	-10	0.90	0.87	0.91
50	10	0.02	0.02	-11	0	0.92	0.86	0.93
100	10	-0.01	0.00	-9	-4	0.93	0.87	0.94
50	50	0.00	0.00	-31	-5	0.94	0.88	0.95
100	50	0.00	0.00	-19	0	0.94	0.87	0.97
100	100	0.00	0.00	-28	1	0.95	0.90	0.97
Non-null models: $\beta_{te} = 0.25$								
10	10	-0.01	-0.01	-36	-21	0.88	0.82	0.88
50	10	0.02	0.02	-21	-11	0.90	0.85	0.92
100	10	0.00	0.00	-22	-17	0.88	0.85	0.91
50	50	0.00	0.00	-29	-3	0.91	0.78	0.93
100	50	-0.01	-0.01	-23	-7	0.89	0.82	0.93
100	100	0.00	0.00	-35	-10	0.92	0.79	0.93

Table A7b. Structured noise ($\mathbf{x}^*$ and $\mathbf{z}^*$) correlated with the random effects ($\mathbf{x}$ and $\mathbf{z}$) with a correlation of 0.5 between $\mathbf{x}$ and $\mathbf{x}^*$ and between $\mathbf{z}$ and $\mathbf{z}^*$ ($\rho_1 = \rho_2 = 0.5$).

sites	species	Bias in $\hat{\beta}_{te}$		Bias in se($\hat{\beta}_{te}$): % of sd of $\hat{\beta}_{te}$ in simulations		Cor of variants of FC correlation with $\hat{\beta}_{te}$ from MLM3		
n	*m*	MLM2	MLM3	MLM2	MLM3	FC N_2-w	FC	FC unw
Null models: $\beta_{te} = 0$								
10	10	-0.01	-0.01	-38	-16	0.90	0.87	-0.01
50	10	0.01	0.01	-61	-14	0.94	0.90	0.01
100	10	0.00	-0.01	-69	-11	0.93	0.89	0.00
50	50	0.00	-0.01	-23	8	0.91	0.80	0.00
100	50	0.00	0.00	-38	6	0.92	0.83	0.00
100	100	0.00	0.00	-29	-1	0.93	0.84	0.00
Non-null models: $\beta_{te} = 0.25$								
10	10	0.02	0.02	-40	-15	0.86	0.84	0.88
50	10	0.00	0.00	-61	-15	0.86	0.85	0.91
100	10	0.00	0.01	-67	-5	0.85	0.86	0.91
50	50	0.01	0.01	-26	4	0.89	0.72	0.91
100	50	0.01	0.01	-42	-1	0.88	0.74	0.91
100	100	0.00	0.00	-29	-2	0.91	0.75	0.93

Table A7c. Series I with large nuisance interactions ($\beta_{tx} = \beta_{ze} = \beta_{z^*x^*} = 1$).

sites	species	Bias in $\hat{\beta}_{te}$		Bias in se($\hat{\beta}_{te}$): % of sd of $\hat{\beta}_{te}$ in simulations		Cor of variants of FC correlation with $\hat{\beta}_{te}$ from MLM3		
n	*m*	MLM2	MLM3	MLM2	MLM3	FC N_2-w	FC	FC unw
Null models: $\beta_{te} = 0$								
10	10	0.02	0.01	-36	-11	0.86	0.76	0.89
50	10	0.00	-0.01	-22	-14	0.88	0.76	0.94
100	10	0.00	0.00	-18	-14	0.89	0.76	0.94
50	50	-0.01	-0.01	-26	3	0.88	0.67	0.91
100	50	0.00	0.00	-16	2	0.88	0.62	0.91
100	100	-0.01	-0.01	-26	4	0.87	0.63	0.91
Non-null models: $\beta_{te} = 0.5$								
10	10	0.06	0.03	-35	-11	0.75	0.58	0.77
50	10	-0.01	-0.01	-15	-4	0.75	0.54	0.79
100	10	-0.07	-0.06	-17	-13	0.73	0.57	0.78
50	50	-0.02	-0.02	-32	-5	0.78	0.52	0.82
100	50	-0.02	0.00	-11	8	0.67	0.38	0.75
100	100	-0.03	-0.01	-29	2	0.70	0.38	0.72

Table A8. Series II: Bias in $\hat{\beta}_{te}$ and its standard error given by MLM2 and MLM3, and the correlation between the MLM3 estimate of β_{te} and three variants of the fourth-corner correlation in simulated binomial count data with binomial total of 100 and $\beta_{tx} = \beta_{ze} = \beta_{z^*x^*} = 0.5$. (Cor = correlation; FC = fourth-corner correlation, N_2-w = N_2-weighted, unw = unweighted; 250 simulations for each scenario; MLM uses the logit-link with beta-binomial response distribution).

(A8a) Structured noise ($\mathbf{x}^*$ and $\mathbf{z}^*$) independent of the random effects ($\mathbf{x}$ and $\mathbf{z}$).

(A8b). Structured noise ($\mathbf{x}^*$ and $\mathbf{z}^*$) correlated with the random effects ($\mathbf{x}$ and $\mathbf{z}$) with a correlation of 0.5 between $\mathbf{x}$ and $\mathbf{x}^*$ and between $\mathbf{z}$ and $\mathbf{z}^*$ ($\rho_1 = \rho_2 = 0.5$).

(A8c). Series II with large nuisance interactions ($\beta_{tx} = \beta_{ze} = \beta_{z^*x^*} = 1$).

(A8d). Series II with large nuisance interactions ($\beta_{tx} = \beta_{ze} = 1$), but no residual correlation ($\beta_{z^*x^*} = 0$).

Table A8a. Structured noise ($\mathbf{x}^*$ and $\mathbf{z}^*$) independent of the random effects ($\mathbf{x}$ and $\mathbf{z}$).

sites	species	Bias in $\hat{\beta}_{te}$		Bias in se($\hat{\beta}_{te}$): % of sd of $\hat{\beta}_{te}$ in simulations		Cor of variants of FC correlation with $\hat{\beta}_{te}$ from MLM3		
n	*m*	MLM2	MLM3	MLM2	MLM3	FC N_2-w	FC	FC unw
Null models: $\beta_{te} = 0$								
15	15	0.00	0.00	-27	2	0.97	0.97	0.97
50	15	-0.01	-0.01	-22	-10	0.97	0.97	0.97
100	15	0.00	0.00	-4	3	0.98	0.98	0.98
50	50	0.00	0.00	-28	2	0.98	0.99	0.98
100	50	0.00	0.00	-28	-11	0.99	0.99	0.99
100	100	0.00	0.00	-31	-2	0.99	0.99	0.99
Non-null models: $\beta_{te} = 0.25$								
15	15	-0.05	-0.04	-30	-2	0.93	0.94	0.94
50	15	-0.03	-0.02	-17	-5	0.93	0.93	0.93
100	15	-0.03	-0.02	-18	-12	0.94	0.94	0.94
50	50	-0.02	-0.01	-28	1	0.94	0.94	0.94
100	50	-0.03	-0.01	-16	2	0.94	0.94	0.95
100	100	-0.03	-0.02	-30	0	0.94	0.95	0.95

Table A8b. Structured noise ($\mathbf{x}^*$ and $\mathbf{z}^*$) correlated with the random effects ($\mathbf{x}$ and $\mathbf{z}$) with a correlation of 0.5 between $\mathbf{x}$ and $\mathbf{x}^*$ and between $\mathbf{z}$ and $\mathbf{z}^*$ ($\rho_1 = \rho_2 = 0.5$).

sites	species	Bias in $\hat{\beta}_{te}$		Bias in se($\hat{\beta}_{te}$): % of sd of $\hat{\beta}_{te}$ in simulations		Cor of variants of FC correlation with $\hat{\beta}_{te}$ from MLM3		
n	*m*	MLM2	MLM3	MLM2	MLM3	FC N_2-w	FC	FC unw
Null models: $\beta_{te} = 0$								
15	15	0.00	-0.01	-35	-7	0.96	0.96	0.96
50	15	0.00	0.00	-19	-7	0.97	0.98	0.98
100	15	0.00	-0.01	-21	-14	0.97	0.98	0.98
50	50	0.00	-0.01	-32	-3	0.98	0.99	0.98
100	50	0.00	-0.01	-23	-3	0.98	0.99	0.99
100	100	0.00	-0.01	-34	-5	0.99	0.99	0.99
Non-null models: $\beta_{te} = 0.25$								
15	15	-0.04	-0.03	-37	-12	0.93	0.93	0.93
50	15	-0.04	-0.03	-16	-4	0.94	0.95	0.95
100	15	-0.04	-0.03	-17	-11	0.94	0.94	0.94
50	50	-0.03	-0.02	-30	-2	0.95	0.95	0.95
100	50	-0.04	-0.03	-21	-3	0.95	0.95	0.95
100	100	-0.04	-0.04	-27	4	0.94	0.95	0.95

Table A8c. Series II with large nuisance interactions ($\beta_{tx} = \beta_{ze} = \beta_{z^*x^*} = 1$).

sites	species	Bias in $\hat{\beta}_{te}$		Bias in se($\hat{\beta}_{te}$): % of sd of $\hat{\beta}_{te}$ in simulations		Cor of variants of FC correlation with $\hat{\beta}_{te}$ from MLM3		
n	*m*	MLM2	MLM3	MLM2	MLM3	FC N_2-w	FC	FC unw
Null models: $\beta_{te} = 0$								
15	15	-0.04	-0.04	-31	-3	0.94	0.95	0.94
50	15	0.00	0.01	-22	-10	0.96	0.97	0.96
100	15	0.00	0.00	-19	-11	0.97	0.97	0.97
50	50	0.01	0.02	-32	-3	0.97	0.98	0.98
100	50	0.00	0.00	-17	2	0.97	0.98	0.98
100	100	0.00	0.00	-28	3	0.97	0.98	0.98
Non-null models: $\beta_{te} = 0.5$								
15	15	-0.16	-0.11	-36	-7	0.91	0.91	0.91
50	15	-0.16	-0.10	-17	-3	0.91	0.91	0.91
100	15	-0.16	-0.10	-13	-7	0.92	0.93	0.93
50	50	-0.17	-0.12	-29	1	0.92	0.93	0.93
100	50	-0.17	-0.12	-24	-5	0.92	0.93	0.93
100	100	-0.17	-0.11	-36	-8	0.93	0.94	0.94

Table A8d. Series II with large nuisance interactions ($\beta_{tx} = \beta_{ze} = 1$), but without residual correlation ($\beta_{z^*x^*} = 0$).

sites	species	Bias in $\hat{\beta}_{te}$		Bias in se($\hat{\beta}_{te}$): % of sd of $\hat{\beta}_{te}$ in simulations		Cor of variants of FC correlation with $\hat{\beta}_{te}$ from MLM3		
n	*m*	MLM2	MLM3	MLM2	MLM3	FC N_2-w	FC	FC unw
Non-null models: $\beta_{te} = 0.5$								
15	15	-0.10	-0.01	-40	-8	0.90	0.92	0.91
50	15	-0.10	0.01	-23	-13	0.93	0.93	0.93
100	15	-0.12	-0.02	-18	-11	0.92	0.92	0.92
50	50	-0.10	0.02	-31	-4	0.91	0.93	0.92
100	50	-0.11	-0.01	-20	-3	0.92	0.93	0.93
100	100	-0.11	0.00	-30	2	0.91	0.92	0.91

References

ter Braak, C.J.F., Peres-Neto, P. & Dray, S. (2017) A critical issue in model-based inference for studying trait-based community assembly and a solution. *PeerJ,* **5,** e2885. https://doi.org/10.7717/peerj.2885

A9. R functions used in the tutorial

This appendix presents the R functions used in the Tutorial (Appendix A1).

A9.1 Utility functions in file utility_functions.r

The utility functions to make trait-environment data suitable for GLMM, which are used to fit the MLM models to data, are:

make_obj_for_traitenv - forms an object of class `TE_obj` (Trait-Environment object), with testing for empty sites and species

expand4glmm - function to change a `TE_obj` into a dataframe for use in glm and glmm (MLM); suitable for multiple traits and environmental variables

dat4MLM2TE_obj - the reverse of expand4glmm (single trait and single env variable only)

There are some more functions for internal usage. The full code is as follows.

Code

```
# Appendix to ter Braak 2019
#New robust weighted averaging- and model-based methods for assessing trait-environ
ment relationships.

make_obj_for_traitenv <- function(E1, L, T1, cut_off = 0){
  # makes object of class TE_obj from matrices or dataframes from L,E1 and T1 (L=Y)
  # no factors allowed, as all is converted to matrices
  # the object is a list of three matrices: L, E, T
  # @param cut_off species occurring strictly less then cut_off are deleted

  L <- as.matrix(L)
  E1 <- as.matrix(E1)
  T1 <- as.matrix(T1)
  # check_L()
  rows<-seq_len(nrow(L))
  cols<-seq_len(ncol(L))
  rni <-which(rowSums(L)==0)
  repeat {
    if (length(rni)) {L <- L[-rni,,drop = FALSE]; rows <-rows[-rni]}
    ksi <- which(colSums(L)==0)
    if (length(ksi)) {L <- L[,-ksi, drop = FALSE]; cols <- cols[-ksi]}
    rni <-which(rowSums(L)==0)
    if ( length(rni)==0 & length(ksi)==0){break}
  }
  E1 <-as.matrix(E1)[rows,,drop = FALSE]
  T1 <-as.matrix(T1)[cols,,drop = FALSE]
  # end check_L()
  if(cut_off >0 ){
    abOcc <- apply(L>0,2,mean)
    L <- L[,abOcc>cut_off, drop =FALSE]
    T1 <- as.matrix(T1)[abOcc>cut_off,, drop = FALSE]
  }
  #print(str(L))
  if(is.null(rownames(L))) rownames(L)<- paste("site", 1:nrow(L), sep="")
  if(is.null(colnames(L))) colnames(L)<- paste("spec",1:ncol(L),sep="")
```

```
  rownames(E1)= rownames(L); if(is.null(colnames(E1))){ if (ncol(E1)>1) colnames(E1
) <-  paste("E", 1:ncol(E1), sep = "") else colnames(E1) <- "env"}
  rownames(T1)= colnames(L); if(is.null(colnames(T1))){ if (ncol(T1)>1) colnames(T1
) <-  paste("T", 1:ncol(T1), sep = "") else colnames(T1) <- "trait"}
  obj = list(L=L, E=E1,T = T1)
  class(obj) = c("TE_obj", class(obj))
  return(obj)
}

expand4glmm <- function(obj, K= 0){
  # K  = 0 for Poisson/Negative binomial counts and Binomial total for (beta-)binom
ial
  # adapted from Jamil et al 2013
  if ('trait0' %in% names(obj)) inPermut_r_c = TRUE else inPermut_r_c = FALSE
  with(obj, {
    sitespec <- expand.grid(rownames(L),colnames(L))
    site <-sitespec[,1]; species<-sitespec[,2]
    y <- as.vector(as.matrix(L))
    Evec <-  E[site,, drop = FALSE];  rownames(Evec)= NULL
    Tvec <-  T[species,, drop = FALSE]; rownames(Tvec)= 1:nrow(Tvec)
    if (inPermut_r_c){
      Evec0 <-  env0[site,, drop = FALSE];  rownames(Evec0)= NULL
      Tvec0 <-  trait0[species,, drop = FALSE]; rownames(Tvec0)= 1:nrow(Tvec0)
    } else {Evec0 =NULL; Tvec0 = NULL}
    if (is.null(colnames(E)))  {
      colnames(Evec) = paste("env", 1:ncol(Evec), sep = "")
    }
    if (is.null(colnames(T))) {
      colnames(Tvec) = paste("trait", 1:ncol(Tvec), sep = "")
    }
    ET <- Rten2_with_names(Evec,Tvec)
    if (inPermut_r_c){
      XYZ <- data.frame(y,site,species, obs = 1:length(y),Tvec, Evec, ET, Tvec0,  E
vec0)
      names(XYZ)[c(5:6, 8:9)] <- c("trait", "env","trait0", "env0")
    } else {
      XYZ <- data.frame(y,site,species, obs = 1:length(y),Tvec, Evec, ET)
      names(XYZ)[5:6] <- c("trait", "env")
    }
    if (K[1]>0) XYZ$y = cbind(y, K-y)
    return(XYZ) # XYZ notation of Jamil et al 2013
  })
}


dat4MLM2TE_obj<- function(dat, cut_off = 0){
  # create abundance matrix TE_obj from a dataframe dat that is in standard order
  #     first all data from site 1, then site 2,
  #        with the all species listed in each site (so include 0 abundances) and
  #        species having the same order in each site
  # @param dat dataframe with names y, sites, species
  # create abundance matrix L, environmental variable vector E and trait vector T
  # BEWARE: currently works for a single trait and environmental variable only!!!!
  dat$site=factor(dat$site)
  dat$species=factor(dat$species)
  n_sites <-nlevels(factor(dat$site))
  n_species <-nlevels(dat$species)
  species <- dat$species[seq(from  = 1, by = n_sites, length.out = n_species)]
  sites <- dat$site[1:n_sites]
  # create the usual trait T, environment E and abundance matrix L;
  # these would normally the original data.. but are here derived from the existing
data.frame dat
  T <- matrix(dat$trait[seq(from  = 1, by = n_sites, length.out = n_species)], nrow
= n_species, dimnames = list(species, "trait"))
  E <- matrix(dat$env[1:n_sites], nrow = n_sites, dimnames = list(sites, "env"))
```

```
  if (length(dat$y)>nrow(dat)){ # e.g. with binomial response
    L <- matrix(dat$y[,1], nrow = n_sites,ncol = n_species, dimnames = list(sites=
sites,species=species))
  }
  else { # e.g. with count data
    L <- matrix(dat$y, nrow = n_sites,ncol = n_species, dimnames = list(sites= site
s,species=species))
  }
  # check the data for empty species and sites and bring them in standard format of
class TE_obj
  obj <- make_obj_for_traitenv(E1 = E, L = L,T1 = T, cut_off = cut_off)
}


Rten2_with_names <- function(X1,X2) {
  one.1 <- matrix(1,1,ncol(X1))
  one.2 <- matrix(1,1,ncol(X2))
  res  <- kronecker(X1,one.2)*kronecker(one.1,X2) #  column of X2 runs fastest
  colnames(res) <- kronecker(colnames(X1),colnames(X2),function(a,b){ paste(b, a, s
ep = ":")} )
  rownames(res) <- rownames(X1)
  return(res)
}
```

A9.2 MLM3 plots in file plot.MLM3.r

The function `plot.MLM3` plots the fixid and random effect of an MLM3 object using ggplot2. The current R code has restrictions on the names of variables. The names in the data.frame (model.frame) of MLM3 should be in any order y, species, sites, trait, env. The data should be in column-wise order (first all the data of the first species, then the second....), as is the case when the data for an MLM3 fit is generated by the utility function `make_obj_for_traitenv` (see Appendix 1.1 Aravo data for an example). The function uses the libraries ggplot2 and dplyr.

```
plot.MLM3 <- function(MLM3, verbose = FALSE, title = paste( 'site and species effec
ts (fixed + random) against env and trait' )){
  # plot of a fitted MLM3 object
  # the names in the data.frame (model.frame) of MLM3 should be in any order
  # y, species, sites, trait, env
  # data should be in column-wise order (first all the data of the first species, t
hen the second....)
  library(ggplot2)
  library(dplyr)
  # get data in Y, trait, env form (Figure 1)-------------------------------------
--------------------------
  dat <-model.frame(MLM3)
  n_sites <- with(dat, nlevels(factor(site)))
  n_species <- with(dat, nlevels(factor(species)))
  species <- dat$species[seq(from  = 1, by = n_sites, length.out = n_species)]
  sites <- dat$site[1:n_sites]
  trait <- dat$trait[seq(from  = 1, by = n_sites, length.out = n_species)]
  env <- dat$env[1:n_sites]
  if (length(dat$y) > nrow(dat)){
    # binomial data
     Y <- matrix(dat$y[,1], nrow = n_sites,ncol = n_species, dimnames = list(sites=
sites,species=species))
  } else {
    # count data
     Y <- matrix(dat$y, nrow = n_sites,ncol = n_species, dimnames = list(sites= sit
es,species=species))
```

```
  }

  # extract random effect and b_te non-null ------------------------------------
---

  if (verbose) print(summary(MLM3))
  B <- summary(MLM3)$coefficients$'cond'

  b_te <- B[nrow(B),1]; gamma0 <- B["env",1]; beta0 <- B["trait",1]
  xx <- ranef(MLM3)$'cond'
  b_i <- xx$site$trait; c_j <-xx$species$env

  b_i_star <- beta0 + b_te*env + b_i      # Eq (2a)
  c_j_star <- gamma0 + b_te*trait + c_j   # Eq (2b)

  ETdat <- data.frame(env, b_i_star, N = rowSums(Y>0), level = "sites")
  names(ETdat)[2]<- "b_star"
  ETdat$beta0 = beta0
  ETdat$b_tslope = beta0 + b_te*env

  TEdat <- data.frame(trait, c_j_star= c_j_star, N = colSums(Y>0), level = "species
")
  names(TEdat)[2]<-"c_star"
  TEdat$gamma0 = gamma0
  TEdat$b_eslope = gamma0 + b_te*trait

  ETTEdat <- full_join(ETdat, TEdat, by= c("N","level"))
  # facet plot ------------------------------------------------------------
  pp <- ggplot(ETTEdat) +
    geom_point(aes(x= env, y = b_star, size = N), shape = 1) +
    geom_point(aes(x= trait, y = c_star, size = N), shape = 16) +
    geom_line(aes(x = env, y = b_tslope), size = 2) +
    geom_line(aes(x = trait, y = b_eslope), size = 2) +
    geom_hline(yintercept = 0, linetype = 'dotted',size = 1) +
    geom_line(aes(x = env, y = beta0), linetype = 'longdash', size = 1) +
    geom_line(aes(x = trait, y = gamma0), linetype = 'longdash', size = 1) +
    facet_wrap(~level, scales = "free") +
    ggtitle(title) +
    xlab("environmental variable                                    trait") +
    ylab("estimated effects")
  return(pp)
}
```

A9.3 Parametric bootstrap test in file Bootstrap_test_prmtrc.r

The function `Bootstrap_test_prmtrc` performs a bootstrap significance test of a fitted MLM3 object. The test statistic can be "Wald" or "LRT"; the latter is slower. The parameter `Binomial_total` is to decide whether the response should be matrix (as in logit models) or vector (as in log-linear models). The parameter `Binomial_total` should be set to the Binomial total N in the former case. The parameter `nAGQ` is used in lme4 to decide on the accuracy of numerical approximation to integrals; the value 0 gives the quickest fit.

Code

```
# Appendix to ter Braak 2019
#New robust weighted averaging- and model-based methods for assessing trait-environ
ment relationships.
Bootstrap_test_prmtrc <- function(MLM3, test_stat = "Wald", nrepet = 19, Binomial_t
otal = 0, nAGQ = 0, ...) {
  # Bootstrap test based on test_stat (Wald or LRT or both) according to Box A2 in
Appendix A3.
  # MLM3 is a fitted glmmTMB or lme4 objects
  # Binomial_total  = 0 for Poisson/Negative binomial counts and Binomial total for
(beta-)binomial
  # BEWARE the interaction coefficient is likely the last one, in the code nrow(B),
.... adapt if not

  # Step 1: arguments test_stat (Wald or LRT) and nrepet
  # Step 2: fit the null model
  formulaNULL <- update(formula(MLM3),  ~ . - trait:env )
  dat <-model.frame(MLM3)
  if (class(MLM3)[1] =="glmmTMB") {
  #  suppressWarnings(MLM0 <- glmmTMB(formulaNULL, data=dat, family= family(MLM3)))
    suppressWarnings(MLM0 <- update(MLM3, formulaNULL))
  } else {
    suppressWarnings(MLM0 <- update(MLM3, formulaNULL,  nAGQ=nAGQ, control = glmerC
ontrol(calc.derivs=F)))
  }
  # Step 3: get F0: the test statistic from the data
  if (test_stat %in% c("Wald", "both")) {
    if (class(MLM3)[1] == "glmmTMB"){
      B <- summary(MLM3)$coefficients$'cond'
    } else { # lme4
      B <- summary(MLM3)$coefficients
    }
    # BEWARE the interaction coefficient is likely the last one.... adapt if not
    obs1 <- 1/B[nrow(B),4] # 1/p-value
    names(obs1) <- "Wald"
  }
  if (test_stat %in% c("LRT", "both")) {
    obs2 <-  logLik(MLM3) - logLik(MLM0)
    names(obs2) <- "LRT"
  }
  if (test_stat=="Wald") obs <- obs1 else if (test_stat=="LRT") obs <- obs2 else ob
s<- c(obs1,obs2)
  if (test_stat == "both") id_LRT <-2 else id_LRT <- 1
  obj <- dat4MLM2TE_obj(dat) # TE_obj
  n_sites <- nrow(obj$L); n_species <- ncol(obj$L)

  sim.boot <- matrix(0, nrow = nrepet, ncol = length(obs))
  colnames(sim.boot) <- paste("boot",names(obs), sep ="")
  for (i in 1:nrepet){
    # step 4 and 5: simulated from the null model and calculate the test statistic
    Ysim <- simulate(MLM0)
    L <- matrix(Ysim[[1]], nrow = n_sites,ncol = n_species, dimnames = list(sites=
rownames(obj$L),species=colnames(obj$L)))
    obj.sim <- make_obj_for_traitenv(obj$E, L, obj$T)
    dat.sim <- expand4glmm(obj.sim, K = Binomial_total)
    if (class(MLM3)[1] == "glmmTMB"){
      suppressWarnings(MLM3.sim <- try(glmmTMB(formula(MLM3), data=dat.sim, family=
family(MLM3))))
      if (!class(MLM3.sim)[1]=="try-error"){
        B <- summary(MLM3.sim)$coefficients$'cond'}
      else {B <- matrix(NA, nrow =2, ncol = 4); MLM3.sim = 0}
    } else if (class(MLM3)[1] == "glmerMod") {
      suppressWarnings(MLM3.sim <- try(glmer(formula(MLM3), data=dat.sim, family= f
```

```
amily(MLM3),  nAGQ=nAGQ, control = glmerControl(calc.derivs=F))))
      if (!class(MLM3.sim)[1]=="try-error"){
        B <- summary(MLM3.sim)$coefficients
      } else {B <- matrix(NA, nrow =2, ncol = 4); MLM3.sim = 0}
    }
    if (test_stat %in% c("Wald","both")){
      sim.boot[i,1] <- 1/B[nrow(B),4]
    }
    if (test_stat %in% c("LRT","both")) { # LRT: refit null model
      if (class(MLM3.sim)[1] =="glmmTMB") {
        suppressWarnings(MLM0.sim <- try(glmmTMB(formulaNULL, data=dat.sim, family=
family(MLM3))))
      } else {
        suppressWarnings(MLM0.sim <- try(update(MLM3.sim, formulaNULL,  nAGQ=nAGQ,
control = glmerControl(calc.derivs=F))))
      }
      if (class(MLM0)[1]=="try-error") {
        sim.boot[i,id_LRT] <- NA
      } else sim.boot[i,id_LRT] <- logLik(MLM3.sim) - logLik(MLM0.sim)
    }
  }
  # step 6:     Compute the Monte Carlo significance level,
  #           i.e. compute the number of values F 0, F 1, F 2, ... , FK
  #           that is greater than or equal to F0 (this number is thus at least 1),
  #           and divide by K + 1 (with K = nrepet).
  if (length(obs)==1){
    isna.r <-  sum(is.na(sim.boot))
    p_boot <- (sum(abs(sim.boot) >= abs(obs), na.rm = TRUE) + 1)  / (nrepet- isna.r
+ 1)
  } else {
    obs.mat <- matrix(rep(abs(obs),each=nrepet), nrow= nrepet, ncol=length(obs))
    isna.r <-  colSums(is.na(sim.boot))
    p_boot <- (colSums(abs(sim.boot) >= obs.mat, na.rm=TRUE) + 1)/ (nrepet - isna.r
+ 1)
  }
  if (test_stat %in% c("Wald","both")) p_val1 <- 1/obs[1]
  if (test_stat %in% c("LRT","both"))  p_val2 <- pchisq(2*obs[id_LRT], df =1, lower
.tail = FALSE)
  if (test_stat=="Wald") p_val <- p_val1 else if (test_stat=="LRT") p_val <- p_val2
else p_val<- c(p_val1,p_val2)
  if (test_stat == "both") {
    result <- cbind(p_prmtrc= p_val, p_boot = p_boot)

  } else result <- c(p_prmtrc= p_val, p_boot = p_boot)
  attr(result, "nrepet")<- nrepet
  return(list(p_values=result, nrepet = nrepet, sim.boot=sim.boot, MLM0 = MLM0, obs
= obs, test_stat = test_stat))
}
```

A9.4 MLM3-based permutational max test in the file MLM3_p_max.r

The function `MLM3_p_max` performs a permutation-based max test of an fitted MLM3. The test statistic can be "Wald" or "LRT"; the latter is slower. The parameter `Binomial_total` is to decide whether the response should be matrix (as in logit models) or vector (as in log-linear models). The parameter `Binomial_total` should be set to the Binomial total N in the former case. The parameter `nAGQ` is used in lme4 to decide on the accuracy of numerical approximation to integrals; the value 0 gives the quickest fit.

The function `MLM3_p_max` depends on a number of functions of which the most important ones are:

MLM_Wald_bte – function that calculates the parametric *p*-value or its inverse for an MLM model.

PermutationTest_r_c_max - max test by randomizing sites and species; the function differs from pmax_PermutationTest (ter Braak et al. 2017 PeerJ) (from which is adapted) by allowing different test statistics for the site-level and species-level test; the test statistic(s) must be specified by the function argument `FUN_test_statistics`, which is set by the options. Options are set by the argument `options = setoptions4pmax_test()`.

Code

```
# Appendix to ter Braak 2019
#New robust weighted averaging- and model-based methods for assessing trait-environ
ment relationships.
MLM3_p_max <- function(MLM3, nrepet = 19, Binomial_total = 0, test_stat = "Wald", n
AGQ =0) {
  #n_sites <- nrow(obj$L); n_species <- ncol(obj$L)

  options1 <- setoptions4pmax_test(verbose = TRUE,
            FUN_test_statistics =  list(init= init_MLM_pmax, test= MLM_Wald_bte),
                                   with.MLM= TRUE, nrepet = nrepet,
                                   perm.mat.spe = 0 , perm.mat.sit = 0, print = 10,
                                   test_statistics_are_1divp = c(TRUE), nmax = 6, f
ilek = 1)

  if (test_stat == "Wald") {with.LRT<-FALSE; LRT <- FALSE} else {with.LRT<-TRUE; LR
T <- TRUE; }

  if (class(MLM3)[1] =="glmmTMB") library = "glmmTMB" else library = "lme4"
  options2 <- setoptions4MLM( formula = formula(MLM3),
                              family = family(MLM3), K = Binomial_total,
                              model.name= c("MLM3"), estimation = FALSE,
                              library = library, output = "short", nAGQ =nAGQ,
                              with.LRT= with.LRT, LRT.only = LRT, verbose = FALSE)
  options3 <- c(options1,options2[-length(options2)])
  obj <- dat4MLM2TE_obj(model.frame(MLM3)) # TE_obj
  # aa <- MLM_Wald_bte(obj, options3)
  # aa
  result <- PermutationTest_r_c_max(obj, options = options3)
  return(result)
}

MLM_Wald_bte <- function(obj, options,  ...){
  #  function for use in PermutationTest_r_c_max
  # @param obj object of class TE_obj (made by function make_obj_for_traitenv(E, L,
T,cutoff))
  # @formula model formula using names created by expand4glmm(obj, K = K)[use verbo
se = TRUE to see these names]
  # @param K is binomial total; if K[1]==0 data are not binomial or presence / abse
nce
  #                      K can be a scalar or vector of n*m (number of sites * numb
er of species)
  # @param family glm family; use character form
  # @param library use "glmmTMB" or else (e.g. "lme4")
  # @param verbose If TRUE, prints the names for use in formula.
  # @param estimation if TRUE, the interaction coefficient is given, else testing i
s carried out via the Wald test
  # @value test statistic: 1/pWald_b_te
  result <- with(options,
```

```
 {
   if (estimation){
     col_b_te = 1 # column position of the interaction coefficient to select it
     invert_p_value = FALSE
   } else col_b_te = 4   # column position of the p-value to select it
   dat <- expand4glmm(obj, K = K)
   #names(dat)[c(5:6, 8:9)] <- c("trait", "env","trait0", "env0")
   #if (verbose) print(str(dat))
   if (library == "glmmTMB"){
     suppressWarnings( MLM <- try(glmmTMB(formula, data=dat, family= family)))
     if (!class(MLM)[1]=="try-error"){
       B <- summary(MLM)$coefficients$'cond'}
     else {B <- matrix(NA, nrow =2, ncol = 4); MLM = 0}
   } else if  (library == "lme4") {
     suppressWarnings(MLM <- try(glmer(formula, data=dat, family= family,  nAGQ=nAG
Q, control = glmerControl(calc.derivs=F))))
     if (!class(MLM)[1]=="try-error"){
       B <- summary(MLM)$coefficients
     } else {B <- matrix(NA, nrow =2, ncol = 4); MLM = 0}
   } else {print(paste("library", library ,"not implemented"))}
   # BEWARE the interaction coefficient is likely the last one.... adapt if not
   p_val_Wald <- B[nrow(B),col_b_te]
   if (verbose){
     print(summary(MLM))
     if (estimation) {names(p_val_Wald) <- "b_te"} else names(p_val_Wald) <- "p_val
_Wald"
     print(p_val_Wald)
   }
   if (is.character(family) ) testnam <- paste(library, family, "MLMWald", sep = ".
") else testnam <- paste(library, "MLMWald", sep = ".")
   if (invert_p_value & !estimation) test_stat <- 1/p_val_Wald else test_stat <- p_
val_Wald
   #test_stat <- matrix(test_stat, nrow = 2, ncol = 1, dimnames =list(c("rows","col
s"), testnam) )
   if (with.LRT){
     formulaNULL <- update(formula(MLM),  ~ . - trait:env )
     if (class(MLM)[1] =="glmmTMB") {
       suppressWarnings(MLM0 <- try(update(MLM, formula = formulaNULL)))
     } else {
       suppressWarnings(MLM0 <- try(update(MLM, formulaNULL,  nAGQ=nAGQ, control =
glmerControl(calc.derivs=F))))
     }
     if (class(MLM0)[1]=="try-error") {
       if (class(MLM)[1] =="glmmTMB") suppressWarnings(MLM0 <- try(glmmTMB(formulaN
ULL, data=dat, family= family(MLM)))) else
         suppressWarnings(MLM0 <- try(glmer(formula, data=dat, family= family,  nAG
Q=nAGQ, control = glmerControl(calc.derivs=F))))
       if (class(MLM0)[1]=="try-error") {
         p_val_LRT <- NA
       } else {
         p_val_LRT <- pchisq(2*( logLik(MLM) - logLik(MLM0)), df =1, lower.tail = F
ALSE)
       }
     } else p_val_LRT <- pchisq(2*( logLik(MLM) - logLik(MLM0)), df =1, lower.tail
= FALSE)
     if (invert_p_value) test_stat2 <- 1/p_val_LRT else test_stat2 <- p_val_LRT
     names(test_stat2) <- "pval LRT"
     if (is.character(family) ) testnam2 <- paste(library, family, "MLM.LRT", sep =
".") else testnam2 <- paste(library, "MLM.LRT", sep = ".")

     #test_stat2 <- matrix(test_stat2, nrow = 2, ncol = 1, dimnames =list(c("rows",
"cols"), testnam2) )
     # print(test_stat2)
     if (LRT.only) {
       test_stat <- test_stat2
       } else {
```

```
      test_stat <- c(test_stat,test_stat2)
      names(test_stat) <- c("p_val_Wald", "p_val_LRT")
    }
  }
  #print(test_stat)
  if (estimation) {
    test_stat <- c(test_stat, B[nrow(B),2]) # standard error
    names(test_stat) <- c("b_te","se_b_te")
  } #else rownames(test_stat)<- c("teststat.site","teststat.species")
  if (verbose) print(test_stat)
  if (output =="short"){
    result <- test_stat
  } else {
    result <- list(test_stat = test_stat, model = MLM )
  }
  return(result)
                 })
  return(result)
}


init_MLM_pmax <- function(obj, options){list(obj=obj, options=options)}


PermutationTest_r_c_max <- function(obj, options = c(setoptions4pmax_test(),setopti
ons4MLM()),  ...) {
  # Specialized max test for WA-based and model-based MLM based methods
  #function to determine randomization p-values based on test statistics for rows a
nd columns
  # based on row and column based permutations of (possibly model-based) test-stati
stics,
  # such as the anova  F-value or chisq-value or equivalently, 1/p-values
  # The function examimes exceedance and takes the absolute value of the test-stati
stics, that explains why 1/p-value can be used.
  #
  # @param obj object of class TE_obj (from make_obj_for_traitenv)
  # @param FUN_test_statistics  function that returns one or more test statistics (
value or vector)
  #                        it should return 1/pval
  #
  #
  # obj from make_obj_for_traitenv(E, L, T, cut_off): matrices L,E and T (one envir
onmental variable, 1-many traits)
  # e.g.FUN_test_statistic =
  # ... options for FUN_test_statistic
  # BEWARE: the size calculation is adapted from Miller et al 2018 MEE but
  # is only trustworthy if the null model holds true!!!!
  # For a non-null model the size is biased upwards (ter Braak, 2019 in prep.)

  # for  permutation of model-based methods trait0 and env0 are not permuted, where
as E and T are.
  # the model formula can thus contain the terms env0 and trait0



  if (is.matrix(options$perm.mat.spe)) options$perm.mat.spe <- options$perm.mat.spe
[sample(nrow(options$perm.mat.spe)),]
  if (is.matrix(options$perm.mat.site)) options$perm.mat.site <- options$perm.mat.s
ite[sample(nrow(options$perm.mat.site)),]

  result <- options$FUN_test_statistics$init(obj,options)
  obj <- result$obj; options <- result$options
  rm(result)
  options_sites <- options_species <- options
  options_sites$verbose  <- options_species$verbose <- FALSE
```

```
  options_species$level <- "species"
  options_sites$level <- "sites"
  if (options$with.MLM){
    # add random T and E to data: trait0 and env0 and update formulas accordingly
    i_spe0 <- sample(1:nrow(obj$T), 1)
    i_sit0 <- sample(1:nrow(obj$E), 1)
    per.col <- syssample(i = i_spe0, options$perm.mat.spe, n =ncol(obj$L))
    obj$trait0 <- matrix(obj$T[per.col,,drop =FALSE],nrow= nrow(obj$T), ncol =ncol(
obj$T))  # used in species level
    per.row <- syssample(i = i_sit0, options$perm.mat.site, n =nrow(obj$L))
    obj$env0 <- matrix(obj$E[per.row,,drop =FALSE],nrow= nrow(obj$E), ncol =ncol(ob
j$E)) # used in site level

    #  e.g:  y ~  trait*env  + (1 + trait| site) + (1 + env|species) becomes
    #       y ~  trait*env  + trait0 + (1 + trait| site) + (1 + env|species) #for s
pecies
    # #     y ~  trait*env  + env0 + (1 + trait| site) + (1 + env|species) # for si
tes

    options_species$formula <-  update(options_species$formula, ~ . + trait0 )
    options_sites$formula  <-  update(options_sites$formula, ~ . + env0 )
  }
  obs.species <- options_species$FUN_test_statistics$test(obj, options = options_sp
ecies)
  obs.sites <- options_sites$FUN_test_statistics$test(obj, options = options_sites)
  obs <- rbind(obs.sites,obs.species)
  if (options$verbose) print(obs)
  if (length(options$test_statistics_are_1divp)==1) {
    options$test_statistics_are_1divp <- rep(options$test_statistics_are_1divp, nco
l(obs))
  } else if (length(options$test_statistics_are_1divp)!=ncol(obs)){
    print(paste("Warning: length of options$test_statistics_are_1divp  (", length(o
ptions$test_statistics_are_1divp) ,") is not equal to number of teststatistics: ",n
col(obs) ))
    options$test_statistics_are_1divp <- rep(options$test_statistics_are_1divp, nco
l(obs))
  }
  if (options$with.MLM){
    # change trait0 and env0 to T and E in data and update formulas so that only en
v:trait is permuted
    obj$trait0 <- obj$T
    obj$env0 <- obj$E
    #  e.g:  y ~  trait*env  + trait0 + (1 + trait| site) + (1 + env|species) becom
es
    #       y ~  trait*env  + trait0 + (1 + trait0| site) + (1 + env|species) #for
species
    # #     y ~  trait*env  + env0 + (1 + trait| site) + (1 + env0|species) # for s
ites

    options_species$formula <- update(options_species$formula,~ . -(1+trait|site)+(
1+trait0|site))
    options_sites$formula <- update(options_sites$formula,~ . -(1+env|species)+(1+e
nv0|species))

  }
  nrepet <- options$nrepet
  if(is.matrix(options$perm.mat.spe)|| is.matrix(options$perm.mat.sit)){
    nrepet <- min(c(nrow(options$perm.mat.spe)-1,nrow(options$perm.mat.sit)-1, nrep
et) )#  minus identity ispe_0 isit_0 permutation (which gives collinearity)
  } else { nrepet <- options$nrepet}
  sim.row <- matrix(0, nrow = nrepet, ncol = ncol(obs))
  sim.col <- matrix(0, nrow = nrepet, ncol = ncol(obs))


  for(i in 1:nrepet){
```

```
    per.row <- syssample(i = i, exclude = i_sit0, perm.mat = options$perm.mat.site,
n =nrow(obj$E))
    per.col <- syssample(i = i, exclude = i_spe0, perm.mat = options$perm.mat.speci
es, n =nrow(obj$T))
    #permute_rows_columns <- function(obj, per.row,per.col){
    obj.row <- obj.col <-  obj  # obj.rc <- obj
    obj.row$E <-obj.row$E[per.row,,drop =FALSE]
    obj.col$T <-obj.col$T[per.col,,drop =FALSE]
    suppressWarnings(sim.row[i, ] <- options_sites$FUN_test_statistics$test(obj.row
,options = options_sites))
    suppressWarnings(sim.col[i, ] <- options_species$FUN_test_statistics$test(obj.c
ol,options = options_species))
    if (!i%%options$print){
      iteration <- i
      print(iteration)
      if (options$filek >=0 ) save(options, sim.row, sim.col, obs, iteration, file
=paste("perm_r_c",options$filek,".rdata", sep = ""))
    }
  }
  ialpha <- 1/options$alpha
  if (ncol(obs)==1){
    isna.r <-  sum(is.na(sim.row))
    isna.c <-  sum(is.na(sim.col))
    pval.row <- (sum(abs(sim.row) >= abs(obs[1,]), na.rm = TRUE) + 1)  / (nrepet- i
sna.r  + 1)
    pval.col <- (sum(abs(sim.col) >= abs(obs[2,]), na.rm = TRUE) + 1)  / (nrepet- i
sna.c  + 1)
    if (options$falseSize){
      size.row <- (sum(sim.row >= ialpha, na.rm=TRUE))/ (nrepet - isna.r)
      size.col <- (sum(sim.col >= ialpha, na.rm=TRUE))/ (nrepet - isna.c)
      size.max <- (sum(max(sim.row,sim.col) >= ialpha, na.rm=TRUE))/ (nrepet - isna
.c)
    }
  } else {
    obs.mat.row <- matrix(rep(abs(obs[1,]),each=nrepet), nrow= nrepet, ncol=ncol(ob
s))
    isna.r <-  colSums(is.na(sim.row))
    pval.row <- (colSums(abs(sim.row) >= obs.mat.row, na.rm=TRUE) + 1)/ (nrepet - i
sna.r + 1)
    isna.c <-  colSums(is.na(sim.col))
    obs.mat.col <- matrix(rep(abs(obs[2,]),each=nrepet), nrow= nrepet, ncol=ncol(ob
s))
    pval.col <- (colSums(abs(sim.col) >= obs.mat.col, na.rm=TRUE) + 1)/ (nrepet - i
sna.c + 1)
    if(options$falseSize){
      size.row <- (colSums(sim.row >= ialpha, na.rm=TRUE))/ (nrepet - isna.r)
      size.col <- (colSums(sim.col >= ialpha, na.rm=TRUE))/ (nrepet - isna.c)
      colnames(sim.row) = colnames(obs)
      colnames(sim.col) = colnames(obs)
    }
  }
  obs[,options$test_statistics_are_1divp] <- 1/obs[,options$test_statistics_are_1di
vp]
  #result <- t(rbind(test_stat = obs, p.site.permut = pval.row, p.species.permut =
pval.col, pmax.permut = pmax(pval.row, pval.col)))
  result <- c(c(obj$p_values[-3]), p.site.permut = pval.row, p.species.permut = pva
l.col, pmax.permut = pmax(pval.row, pval.col))

  attr(result, "nrepet")<- nrepet
  if (options$falseSize) result <- cbind(result, t(rbind(size.row = size.row, size.
col = size.col)))
  return(list(p_values=result, nrepet = nrepet, obs = obs,sim.row=sim.row, sim.col
= sim.col))
}

syssample <- function(i = 0, exclude = 0, perm.mat = 0, n = 10){
```

```
  # function to generate either random samples (if !is.matrix(perm.mat)) or
  # a systematic sample: i-th or for i>= exclude: i+1 th row of permat (i.e. withou
t the excluded sample)
  # exclude = 0 for no exclude, or permutation number to exclude
  if (is.matrix(perm.mat)){
    if (!exclude){ ii <- i} else{
      if(i<exclude) ii <- i else {ii <- i + 1}
    }
    sam <- perm.mat[ii,]
  } else {
    sam <-sample(x=n)
  }
  return(sam)
}

f_allPerms <- function(n){
  # all permutation matrix without identity and reverse order
  library(permute)
  result <- allPerms(n)[1:(numPerms(n)/2 - 1)  ,]
  result[sample(nrow(result)),]
}
```

Code for setting options (file setoptions.r)

These are functions to set options in the MLM3 fit and max test in the functions `MLM_Wald_bte` and `PermutationTest_r_c_max`.

```
setoptions4pmax_test <- function(verbose = TRUE, FUN_test_statistics = CWMSNC, with
.MLM= FALSE
 , nrepet = 49, level = 'sites', perm.mat.spe = 0 , perm.mat.sit = 0, print = 10,
 falseSize = FALSE, alpha = 0.05, test_statistics_are_1divp = TRUE, nmax = 6, filek
= -1){
  # set options for use in PermutationTest_r_c_max
  # perm.mat.spe =0 : random sample; else number of species,
  # if perm.mat.spe or sit >= nmax : random sampling only
  # or if the number of systematic samples
  if (perm.mat.spe > 0 & perm.mat.spe < nmax) perm.mat.spe <- f_allPerms(perm.mat.s
pe)
  if (perm.mat.sit > 0 & perm.mat.sit < nmax) perm.mat.sit <- f_allPerms(perm.mat.s
it)
  if (filek ==0 ) filek <- sample(1000,1)
  list(verbose = verbose, FUN_test_statistics = FUN_test_statistics, with.MLM= with
.MLM,
       nrepet= nrepet, level = level,perm.mat.spe=perm.mat.spe, perm.mat.sit= perm.
mat.sit,        print = print,
       falseSize = falseSize, alpha = alpha, test_statistics_are_1divp =
                      test_statistics_are_1divp, filek = filek )
}


setoptions4MLM<- function(   formula = y~ poly(trait,2) + poly(env,2) + trait:env +
(1+trait|site)+ (1+env|species),
                             family = "nbinom2", K = 0,
                             model.names= c("MLM3"),
                             library = "glmmTMB", nAGQ=0,
                             estimation = FALSE, invert_p_value = TRUE, output = "s
hort",
                             with.LRT= FALSE, LRT.only = FALSE, verbose = FALSE){
  # LRT: use the LRT test  instead of the Wald test
  # if LRT == FALSE and with.LRT == TRUE then LRT is an extra test statistic; name
```

```
it in model.names!!!
  if (LRT.only) with.LRT <- TRUE
  list(formula = formula,  family =  family, model.names = model.names, library=lib
rary, nAGQ=nAGQ,
       estimation = estimation, invert_p_value = invert_p_value, K= K, output = out
put,
       with.LRT= with.LRT, LRT.only =LRT.only,  verbose = verbose)
}
```

A9.5 WA-based analyses in file WAregressions.r

The functions for the weighted averaging(WA)-based methods are in the file WAregressions.r. There are three functions for users:

- `CWMSNC_regressions`: WA regressions (CWM/SNC regressions) with permutation testing.
- `summary.CWMSNCr` : summary of `CWMSNC_regressions`
- `plot.CWMSNCr` : Plots of `CWMSNC_regressions` (main effects and trait-env association)

There are three weighing options in `CWMSNC_regressions`,

1. `weighing = "N2"` for N_2-weighted CWM/SNC regression (N_2-weighted lm)
2. `weighing = "unw"` for unweighted, giving (permutation-based) lm CWM/SNC
3. `weighing = "FC"` for the fourth-corner correlation (fourth).

There is a cutoff option in `CWMSNC_regressions` to set a minimal number of occurrences of a species and in a site (default 0, which only removes only empty sites and species without any presence; see the utility function `make_obj_for_traitenv`;see section A9.1).

The remaining functions are for internal use in the above three main functions. The tutorial also uses `WA_p_max` for checking of the fast verson of the permutational max test used in `CWMSNC_regressions`. For details on the WA-based regression, see the function `CWMSNC_regressions0`, which is used once in the fast version but repeatedly in the non-fast version of the permutational max test. The R-code for the permutational max test (function `PermutationTest_r_c_max`) has been given in section A9.4.

Code

```
# Appendix to ter Braak 2019
#New robust weighted averaging- and model-based methods for assessing trait-environ
ment relationships.

# functions for the statistical analysis of trait-environment association
# 1 CWMSNC_regressions: Weighted averaging (WA) regressions (CWM/SNC regressions) w
ith permutation testing,
#                       with and without N2- or RK (Fourth-corner correlations) wei
ghting
# 2 summary.CWMSNCr   : summary of CWMSNC_regressions
```

```
# 3 plot.CWMSNCr      : Plots of CWMSNC_regressions (main effects and trait-env ass
ociation)
#
# the remaining functions are for internal use in the above three main functions



CWMSNC_regressions<-function(E, L=0, T=0, weighing = "N2", cutoff = 0, nrepet = 499
){
  # combination of CWM- and SNC-based regressions, possibly weighted.
  # @param E  object from class TE_obj from make_obj_for_traitenv(E, L, T,cutoff) o
r n-vector or nx1 matrix with the environmental values
  # @param L  n x m matrix of abundance values or nonspecified if E = object from f
unction make_obj_for_traitenv(E,L,T,cutoff)
  # @param T  m-vector or m x 1 matrix with trait values or nonspecified if E = obj
ect from function make_obj_for_traitenv(E,L,T,cutoff)
  # @ weighing = 0 or "unw": no weights
  #          = 1 or "RK" or "FC"  weights are the row and columns totals of L, as i
n the original fourth-corner
  #          = 2 or "N2" weights are the Hill N2-effective numbers
  #               (effective number of occurrences of a species and N2-diversity of
a site)
  # @cutoff  if E is not a TE_obj, cutoff on minimal number of occurrences of a spe
cies

  if(class(E)[1]!="TE_obj"){
    obj <- make_obj_for_traitenv(E,  L,  T,cut_off = cutoff)
  } else obj <- E
  result <- CWMSNC_regressions0(obj, weighing = weighing)
  # correlations
  result$wcorCWMSNC <- with(result, c(wcor(CWM,E),wcor(SNC,T)))
  names( result$wcorCWMSNC) <-  paste(result$weighing, c("_CWM_E","_SNC_T"),"_cor",
sep="")
  result$wFC <- with(result,FC_cor_generalized(obj, wE = wsites, wT = wspecies, R =
R, K = K))
  names(result$wFC) <- paste(result$weighing, c("_sites","_species", "_fourth_corne
r"),"_cor", sep="")
  # p-values of association
  result$p_values <- WA_p_max1(obj, weighing= weighing, nrepet =nrepet)$p_values
  attr(result$p_values, which = "nrepet") <- nrepet
  class(result) <- "CWMSNCr"
  return(result)
}

summary.CWMSNCr <- function(result, digits = 3){
  # result should be an object resulting from CWMSNC_regressions
  res <- rbind(result$wFC, result$p_values[-c(1,2)])
  rownames(res) <- paste(c("correlations", "p-values"), " (", result$weighing, ")"
, sep = "")
  colnames(res) <- c("sites","species","min/max")
  attr(res, "nrepet")<- attr(result$p_values, which = "nrepet")
  print(res, digits = digits)
}


plot.CWMSNCr <- function(result,title = paste(result$weighing, '-regressions of CWM
on env and SNC on trait',sep= "") ){
  # plot of a CWMSNCr object

  #title = paste(result$weighing, '-regressions of CWM on env and SNC on trait',sep
= "")

  library(ggplot2)

  ETdat <- with(result, data.frame( E,  CWM, R, wsites, meanT, wmeanCWM, level = "s
```

```
ites"))
  names(ETdat) <- c("E", "CWM", "R", "N2", "meanT","wmeanCWM" ,"level")

  TEdat <- with(result, data.frame( T,  SNC, K, wspecies, meanE, wmeanSNC, level =
"species"))
  names(TEdat) <- c("T", "SNC", "K", "N2", "meanE","wmeanSNC","level")


  ETTEdat <- suppressMessages(dplyr::full_join(ETdat,TEdat))

  # facet plot ------------------------------------------------------------
  suppressMessages(p.main <-  ggplot(data= ETTEdat) +
                      geom_point(aes(x= E, y = log(R), size = N2),shape = 1) +  stat
_smooth(aes(x= E, y = log(R), weight=N2), method = "gam") +
                      geom_point(aes(x= T, y = log(K), size = N2), shape = 16) + sta
t_smooth(aes(x= T, y = log(K), weight=N2), method = "gam") +
                      xlab(" env
trait")+
                      ylab(" log totals") +
                      facet_wrap(~level, scales = "free") +
                      ggtitle("log site and species totals vs env and trait"))
  suppressMessages(p.assoc <-  ggplot(data= ETTEdat) +
                      geom_point(aes(x= E, y = CWM, size = N2),shape = 1) +  stat_sm
ooth(aes(x= E, y = CWM, weight=N2), method = "gam") +
                      geom_point(aes(x= T, y = SNC, size = N2), shape = 16) + stat_s
mooth(aes(x= T, y = SNC, weight=N2), method = "gam") +
                      xlab(" env
trait")+
                      ylab("weighted means") +
                      geom_line(aes(x = E, y = meanT), linetype = 'dotted', size = 1
)+
                      geom_line(aes(x = T, y = meanE), linetype = 'dotted', size = 1
) +
                      geom_line(aes(x = E, y = wmeanCWM), linetype = 'longdash', siz
e = 1)+
                      geom_line(aes(x = T, y = wmeanSNC), linetype = 'longdash', siz
e = 1) +
                      facet_wrap(~level, scales = "free") +
                      ggtitle(title))
  #  print(p.main)
  #  print(p.assoc)
  return(list(p.main,p.assoc))
}

# the remaining functions are for internal use in the above three main functions

FC_cor_generalized <- function(E, L=0, T=0, wE=rep(1,nrow(L)) , wT= rep(1,ncol(L)),
wNam = "w", R = rowSums(L), K = colSums(L)){
  # generalized fourth-corner correlation: wE and wT are weights for sites and spec
ies respectively.
  # if wE == R and wT == K it gives the fourth-corner correlation
  # value: three correlations, one for sites and one for species and the signed min
imum (0 if different in sign)

  # local function
  f_cor_min <- function(x){
    # takes the signed minimum correlation if the correlations have the same sign (
close to min)
    # 0  if the correlations differ in sign
    if (sum(is.na(x))) return(NA)
    if (prod(x) < 0 ) { # the two elements in x diffent sign
      g <- 0
    } else {
      g <-  sign(x[1]) * min(abs(x)) # min
    }
```

```
    g
  }
  # end local function
  if(class(E)[1]!="TE_obj"){
    obj <- make_obj_for_traitenv(E,L, T,cutoff)
  } else obj <- E
  wE <- wE/sum(wE)
  wT <- wT/sum(wT)
  result <- with(obj, {
    Estd_w <- standardize_w(E,wE)
    Tstd_w <-  standardize_w(T,wT)
    Lw_si <- diag(wE/R)%*% L
    Lw_sp <-  L %*% diag(wT/K)
    #Fourthcorner  <- t(Estd_w)  %*% (L%*% Tstd_w)
    r_wE <- (t(Estd_w) %*% (Lw_si %*% Tstd_w))/sum(wE) # sum(wE)== sum(Lw_si)
    r_wT <- sum((t(Lw_sp)%*% Estd_w) *  Tstd_w)/sum(wT) # sum(wT)== sum(Lw_sp)
    result <- c(r_wE, r_wT)
    result <- c(result, f_cor_min(result))
    names(result) <-paste(wNam, c("wFC_r_sites", "wFC_r_species", "wFC_r_min"), sep
= "_")
    return(result)
  })
  return(result)
}



# Hill number of order 2: N2
fN2 <- function(x){x <- x/sum(x); 1/sum(x*x)}


CWMSNC_regressions0<-function(E,L=0,T=0, weighing = 0, cutoff = 0){
# combination of CWM- and SNC-based regressions, possibly weighted.
# @param E  object from class TE_obj from make_obj_for_traitenv(E,L,T,cutoff) or n-
vector or nx1 matrix with the environmental values
# @param L  n x m matrix of abundance values or nonspecified if E = object from fun
ction make_obj_for_traitenv(E,L,T,cutoff)
# @param T  m-vector or m x 1 matrix with trait values or nonspecified if E = objec
t from function make_obj_for_traitenv(E,L,T,cutoff)
# @ weighing = 0 or "unw": no weights
#          = 1 or "RK" or "FC" weights are the row and columns totals of L, as in t
he original fourth-corner
#          = 2 or "N2" weights are the Hill N2-effective numbers
#               (effective number of occurrences of a species and N2-diversity of a
site)
# @cutoff  if E is not a TE_obj, cutoff on minimal number of occurrences of a speci
es

  if(class(E)[1]!="TE_obj"){
    obj <- make_obj_for_traitenv(E1 = E, L = L,T1 = T,cutoff)
  } else obj <- E
  result <- with(obj,{
      R <- rowSums(L) # the site totals
      K <- colSums(L) # the species totals
      if (weighing %in% c(0,"unw") ){
        wsites <- rep(1, nrow(L))
        wspecies <-rep(1,ncol(L))
        weighing = "unw"
      } else if (weighing %in% c(1,"RK","FC")){
        wsites <- R
        wspecies <-K
        weighing = "RKw"
      } else if (weighing %in% c(2,"N2")){
        N2_si <- apply(L, 1, fN2)
        N2_sp <- apply(L, 2, fN2)
```

```
        wsites <- N2_si
        wspecies <- N2_sp
        weighing = "N2w"
      }
      CWM <- (L%*%T)/R #Community weighted mean wrt to trait T (mean trait value pe
r site)
      SNC <- (t(L)%*%E)/K  # Species niche centroid wrt to environmental variable E
(mean environmental value per species)
      meanE <- mean_w(E)
      meanT <- mean_w(T)
      wmeanCWM <- mean_w(CWM, w = wsites)
      wmeanSNC <- mean_w(SNC, w = wspecies)
      # site-level regression of CWM on E
      lm_CWMe <- lm(CWM~E, weights = wsites)
      if (ncol(CWM) ==1) {
        anova_site <- anova(lm_CWMe)
        F_sites <- anova_site$`F value`[1]# get F-value and p-value of site-level t
est
        Prob.site<- anova_site$"Pr(>F)"[1]
      } else {F_sites <-NA; Prob.site <- NA}
      # species-level regression of SNC on T
      lm_SNCt <- lm(SNC~T, weights =   wspecies)
      if (ncol(SNC)==1){
        anova_species <- anova(lm_SNCt)
        F_species <- anova_species$`F value`[1]# get F-value and P-value of species
-level test
        Prob.species <- anova_species$"Pr(>F)"[1]
      } else {F_species <-NA; Prob.species <- NA}
      result<-list(p_values = c(p.site.prmtrc=Prob.site,p.species.prmtrc=Prob.speci
es,p.max.prmtrc=max(Prob.site,Prob.species)), F_values =  c(F_sites = F_sites, F_sp
ecies= F_species),
                   lm_CWMe = lm_CWMe, lm_SNCt = lm_SNCt, CWM = CWM, SNC = SNC,
                   E = E, T = T, L = L, meanE = meanE, meanT = meanT, wmeanCWM = wm
eanCWM, wmeanSNC = wmeanSNC,
                   R=R, K= K, wsites = wsites, wspecies=wspecies,weighing=weighing,
cutoff=cutoff)
    return(result)
    }
  )
  return(result)
}

CWMr<-function(E,L=0,T=0, weighing = "unw", cutoff = 0){
  # CWMr CWM-based regressions, possibly weighted.
  # @param E  object from class TE_obj from make_obj_for_traitenv(E,L,T,cutoff) or
n-vector or nx1 matrix with the environmental values
  # @param L  n x m matrix of abundance values or nonspecified if E = object from f
unction make_obj_for_traitenv(E,L,T,cutoff)
  # @param T  m-vector or m x 1 matrix with trait values or nonspecified if E = obj
ect from function make_obj_for_traitenv(E,L,T,cutoff)
  # @ weighing = 0 or "unw": no weights
  #          = 1 or "RK" or "FC" or "regular" weights are the row and columns total
s of L
  #          = 2 or "N2" weights are the Hill N2-effective numbers
  #               (effective number of occurrences of a species and N2-diversity of
a site)
  # @cutoff  if E is not a TE_obj, cutoff on minimal number of occurrences of a spe
cies

  if(class(E)[1]!="TE_obj"){
    obj <- make_obj_for_traitenv(E,L,T,cutoff)
  } else obj <- E
  result <- with(obj,{
    R <- rowSums(L) # the site totals
    if (weighing %in% c(0,"unw") ){
      wsites <- NULL
```

```
    } else if (weighing %in% c(1,"RK", "FC")){
      wsites <- R
    } else if (weighing %in% c(2,"N2")){
      N2_si <- apply(L, 1, fN2)
      wsites <- N2_si
    }
    CWM <- (L%*%T)/R #Community weighted mean wrt to trait T (mean trait value per
site)
    # site-level regression of CWM on E
    lm_CWMe <- lm(CWM~E, weights = wsites)
    anova_site <- anova(lm_CWMe)
    F_sites <- anova_site$`F value`[1]# get F-value and p-value of site-level test
    Prob.site<- anova_site$"Pr(>F)"[1]
    result<-list(p_values =Prob.site, F_value = F_sites,
                 lm_CWMe = lm_CWMe,  CWM = CWM,  E = E, T = T, L = L, wsites = wsit
es,weighing=weighing,cutoff=cutoff)

    return(result)
  }
  )
  return(result)
}


SNCr<-function(E,L=0,T=0, weighing = 0, cutoff = 0){
  # CWMr CWM-based regressions, possibly weighted.
  # @param E  object from class TE_obj from make_obj_for_traitenv(E,L,T,cutoff) or
n-vector or nx1 matrix with the environmental values
  # @param L  n x m matrix of abundance values or nonspecified if E = object from f
unction make_obj_for_traitenv(E,L,T,cutoff)
  # @param T  m-vector or m x 1 matrix with trait values or nonspecified if E = obj
ect from function make_obj_for_traitenv(E,L,T,cutoff)
  # @ weighing = 0: no weights
  #          = 1 or "RK" or "FC" or  weights are the row and columns totals of L
  #          = 2 or "N2" weights are the Hill N2-effective numbers
  #               (effective number of occurrences of a species and N2-diversity of
a site)
  # @cutoff  if E is not a TE_obj, cutoff on minimal number of occurrences of a spe
cies

  if(class(E)[1]!="TE_obj"){
    obj <- make_obj_for_traitenv(E,L,T,cutoff)
  } else obj <- E
  result <- with(obj,{
    #R <- rowSums(L) # the site totals
    K <- colSums(L) # the species totals
    if (weighing %in% c(0,"unw") ){
      wsites <- NULL
      wspecies <-NULL
    } else if (weighing %in% c(1,"RK", "FC")){
      #wsites <- R
      wspecies <-K
    } else if (weighing %in% c(2,"N2")){
      N2_sp <- apply(L, 2, fN2)
      wspecies <-N2_sp
    }
    SNC <- (t(L)%*%E)/K  # Species niche centroid wrt to environmental variable E (
mean environmental value per species)
    #species-level regression of SNC on T
    lm_SNCt <- lm(SNC~T, weights =   wspecies)
    anova_species <- anova(lm_SNCt)
    F_species <- anova_species$`F value`[1]# get F-value and P-value of species-lev
el test
    Prob.species<- anova_species$"Pr(>F)"[1]
    result<-list(p_values =Prob.species, F_value = F_species,
```

```
                  lm_SNCt = lm_SNCt,  SNC = SNC,  E = E, T = T, L = L, wspecies = ws
pecies,weighing=weighing,cutoff=cutoff)

    return(result)
  }
  )
  return(result)
}

CWMSNCr <- function(obj, options){
  # level in c("species","sites","both")
  if(class(obj)[1]!="TE_obj") {print("obj must be of class TE_obj in CWMSNCr"); ret
urn(obj)}
  WA <- with(options,{
  if (level == "sites") WA <- c(site_p = CWMr(obj, weighing = weighing)$p_values) e
lse
    if (level == "species") {
      WA <- c(species_p =SNCr(obj, weighing =weighing )$p_values)
    } else if (level == "both") {
      WA <- c(CWMr(obj, weighing =weighing)$p_values,SNCr(obj, weighing =weighing )
$p_values)
      names(WA)<- c("site_p", "species_p")
    }
  return(WA)
  })
  if (options$invert_p_value) WA <- 1/WA
  return(WA)
}

WA_p_max <- function(obj, nrepet = 19,  weighing = "N2", score_test = TRUE, fast =
TRUE) {
  if (!fast){
    # following the description with test stat = 1/p_value of regressions performed
in full
    result <- WA_p_max0(obj, nrepet = nrepet,  weighing = weighing)
  } else {
    # short cut, which is even more general (allows multi-trait multi-envi)
    if(ncol(obj$E)==1 && ncol(obj$T == 1)) score_test = FALSE
    result <- WA_p_max1(obj, nrepet = nrepet,  weighing = weighing, wsvd = score_te
st)
  }
  return(result)
}

WA_p_max0 <- function(obj, nrepet = 19,  weighing = "N2") {
  #n_sites <- nrow(obj$L); n_species <- ncol(obj$L)

  options1 <- setoptions4pmax_test(verbose = FALSE,
                       FUN_test_statistics = list(init=function(obj,options){list(o
bj =obj, options = options)}, test=CWMSNCr),
                       with.MLM= FALSE, nrepet = nrepet,
                       perm.mat.spe = 0 , perm.mat.sit = 0, print = 1000,
                       test_statistics_are_1divp = c(TRUE), nmax = 6, filek = -1)
  options2 <- setoptions4WA(weighing = weighing , invert_p_value = TRUE, fast = FAL
SE)
  options3 <- c(options1,options2[-length(options2)])
  result <- PermutationTest_r_c_max(obj, options = options3)
  return(result)
}



# local functions
mean_w <- function(X,w = rep(1/nrow(X),nrow(X))){t(w/sum(w))%*% X}
center_w <- function(X,w = rep(1/nrow(X),nrow(X))){ X - rep(1,length(w))%*%t(w)%*%
```

```
X }
standardize_w <- function(X,w = rep(1/nrow(X),nrow(X)), wsvd = FALSE){
  # NB requires w to be have sum 1
  ones <- rep(1,length(w))
  Xc <- X - ones %*% t(w)%*% X
  Xstd <- Xc / ones%*%sqrt(t(ones)%*%(Xc*Xc*w))
  if (wsvd) Xstd <- wSVD(Xstd, w)
return(Xstd)
}

wcor <- function(X, Y=X, w = rep(1,nrow(X))){
  # weighted correlation between matrix X and Y
  w <- w/sum(w)
  Xstd <- standardize_w(X, w)
  Ystd <- standardize_w(Y, w)
  t(Xstd) %*% diag(w) %*% Ystd
  }

wSVD <- function(Y,w=rep(1/nrow(Y),nrow(Y))){
  sw <- sqrt(w)
  Ystar <- Y*sw
  svdY <- svd(Ystar)
  Ustar <- svdY$u
  id <- which(svdY$d > 1.e-6)
  return(Ustar[,id, drop = FALSE]/sw)
} # returns w-orthogonalized Y

init_WA4_pmax <- function(obj, options){
 res <- CWMSNC_regressions0(obj, weighing = options$weighing)
 Wn <- res$wsites/sum(res$wsites)
 Ws <- res$wspecies/sum(res$wspecies)
 names(res)
 E <-standardize_w(res$E, Wn, wsvd = options$wsvd)
 T <- standardize_w(res$T, Ws, wsvd = options$wsvd)
 CWM <- res$L%*%T/res$R #CWM wrt to standardized T (trait), and with wSVD orthogona
lized
 SNC <- t(res$L)%*%E/res$K # SNC wrt to standardized E (environment) and with wSVD
orthogonalized
 result <- list(E=E,T=T, CWM=CWM, SNC = SNC, p_values = res$p_values, Wn = Wn, Ws=W
s)
 return(list(obj = result, options = options))
}

#obj_init <- init_WA4pmax(obj,options)

test_stat_WA <- function(obj, options){
  test_stat <- with(obj, {
    if (options$level =="sites"){
      Estd <- standardize_w(E, Wn, wsvd = options$wsvd)
      test_stat <- t(Estd)%*%diag(Wn)%*% CWM
      #test_stat <- sum(test_stat*test_stat)
      return(t(rep(1,ncol(Estd))) %*% (test_stat*test_stat)%*% rep(1,ncol(CWM)))

    } else {
      Tstd <- standardize_w(T, Ws, wsvd = options$wsvd)
      test_stat <- t(Tstd)%*% diag(Ws)%*% SNC
      return(t(rep(1,ncol(Tstd))) %*% (test_stat*test_stat)%*% rep(1,ncol(SNC)))

    }
     #return(sum(test_stat*test_stat))
    })
 return(test_stat)
}

#test_stat_WA(obj_init,options)
```

```
WA_p_max1 <- function(obj, nrepet = 19,  weighing = "N2", wsvd = TRUE) {
  #n_sites <- nrow(obj$L); n_species <- ncol(obj$L)

  options1 <- setoptions4pmax_test(verbose = FALSE,
                                   FUN_test_statistics = list(init= init_WA4_pmax,
test= test_stat_WA),
                                   with.MLM= FALSE, nrepet = nrepet,
                                   perm.mat.spe = 0 , perm.mat.sit = 0, print = 500
0,
                                   test_statistics_are_1divp = c(FALSE), nmax = 6,
filek = -1)
  options2 <- setoptions4WA(weighing = weighing, invert_p_value = FALSE, wsvd = wsv
d, fast = TRUE)
  options3 <- c(options1,options2[-length(options2)])
  result <- PermutationTest_r_c_max(obj, options = options3)
  return(result)
}
setoptions4WA<- function(weighing = 0, level = "both", invert_p_value = TRUE, fast
= TRUE, wsvd=TRUE,  verbose = FALSE){
  list(weighing = weighing, level = level, invert_p_value = invert_p_value, fast =
fast, wsvd= wsvd, verbose = verbose)
}
```

A10. R functions for simulation

Simulations in the paper are either based on modifying a fitted MLM3 model or using the simulation model described in section A8.2. The code is in the folder Simulations with subfolders Rdata and Rfunctions, containing the fitted MLM3 models for the Revisit and Aravo data, and the data generating function `generate_data` implementing data derived simulation and the simulation model of Appendix 8 (section A8.2).

A10.1 Simulation scenarios

The simulated data using fitted MLM3 models were generated as shown in the files

- Revisit_based_SimulationsFig5.r,
- Revisit_based_SimulationsFig6.r and
- Aravo_based_SimulationsFigA1.r.

The simulation Series I-V are in files with these names. As examples, I present here the code of Revisit_based_SimulationsFig5 and simulation Series V. The code of the other files is similar.

Code Revisit_based_SimulationsFig5.r

```
#Appendix to ter Braak 2019
#New robust weighted averaging- and model-based methods for assessing trait-environ
ment relationships.
```

```r
rm(list=ls(all=TRUE))  # remove all existing items from the workspace
#source("Rfunctions/WAregressionsMLM_rand_max_test.r")
#source("Rfunctions/PowerSimulation_functions3.r") #NEW version
# source("Rfunctions/generate_trait_env_community_data.r")
# source("Rfunctions/setoptions.r")
source("Rfunctions/utility_functions.r")

library(glmmTMB)

load("Rdata/MLM3.Rdata")
summary(MLM3)

##  Family: betabinomial  ( logit )
## Formula:
## y ~ env * trait + (1 + trait | site) + (1 + env | species)
## Data: dat
##
##      AIC      BIC   logLik deviance df.resid
##   6193.9   6262.9  -3086.0   6171.9     3889
##
## Random effects:
##
## Conditional model:
##  Groups  Name        Variance Std.Dev. Corr
##  species (Intercept) 2.1786   1.4760
##          env         0.1776   0.4214   0.41
##  site    (Intercept) 0.2255   0.4749
##          trait       0.1134   0.3367   -0.92
## Number of obs: 3900, groups:  species, 75; site, 52
##
## Overdispersion parameter for betabinomial family (): 3.35
##
## Conditional model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.42782    0.21827 -24.868  < 2e-16 ***
## env         -0.68239    0.12503  -5.458 4.82e-08 ***
## trait        1.00306    0.19788   5.069 4.00e-07 ***
## env:trait    0.25117    0.09583   2.621  0.00876 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# set interaction coef and variance parameters,
# sigma_b = sd of trait slopes, sigma_c = sd of env slopes
id_sigma_b = which(MLM3$fit$parfull == MLM3$fit$par[10]) # 264
id_sigma_c = which(MLM3$fit$parfull == MLM3$fit$par[7]) # 261
b_te = MLM3$sdr$par.fixed[4]
sigma_b = exp(MLM3$fit$parfull[id_sigma_b])
sigma_c = exp(MLM3$fit$parfull[id_sigma_c])
sigma_b # trait.site = effect other env

##     theta
## 0.3367028

sigma_c # env.species = effect other trait

##     theta
## 0.4213998

b_te_values = c(0, 0.1, 0.2, 0.3, 0.4, 0.6)
sigma_b_values <-  c(sigma_b, 1)
sigma_c_values <-  c(sigma_c)
scenarios <- expand.grid(b_te_values,sigma_b_values,sigma_c_values)
names(scenarios) = c("b_te","sigma_b", "sigma_c")
id  <-  scenarios[,"sigma_b"]  == sigma_b | scenarios[,"b_te"] == 0
```

```
scenarios <- scenarios[ id ,]
nrow(scenarios)

## [1] 7

scenarios

##   b_te   sigma_b   sigma_c
## 1  0.0 0.3367028 0.4213998
## 2  0.1 0.3367028 0.4213998
## 3  0.2 0.3367028 0.4213998
## 4  0.3 0.3367028 0.4213998
## 5  0.4 0.3367028 0.4213998
## 6  0.6 0.3367028 0.4213998
## 7  0.0 1.0000000 0.4213998

n_simul <- 2 # 250
#  ---------------------------------------------------------------------
klist <- 1:nrow(scenarios)
Scenario.list <-  vector("list", length = nrow(scenarios))


for (k in klist){
  b_te <-scenarios[k, "b_te"]
  sigma_b <- scenarios[k, "sigma_b"]
  sigma_c <- scenarios[k, "sigma_c"]

  MLM3$sdr$par.fixed[4] <- b_te # not needed, but to be sure
  MLM3$fit$par[4]<- b_te
  MLM3$fit$parfull[4] <- b_te
  MLM3$fit$parfull[id_sigma_b] <- log(sigma_b)
  MLM3$fit$parfull[id_sigma_c] <- log(sigma_c)

  options <- list()
  options$model <- "MLM"; options$MLM = MLM3
  if (options$model == "MLM"){
    # Simulation from an existing MLM model for which simulate(mod) exists(lme4 or
glmm.TMB model)
    mod <- options$MLM
    dat<- model.frame(mod)
    obj <- dat4MLM2TE_obj(dat) # TE_obj
    n_sites <- nrow(obj$L); n_species <- ncol(obj$L)
  } else {
    n_sites <- scenarios$n[k]; n_species <- scenarios$m[k]
    }
  # one trait one environmental variables  data----------------------------------
  for (i in 1:n_simul){
    if (options$model == "MLM"){
      Ysim <- simulate(mod)
      L <- matrix(Ysim[[1]], nrow = n_sites,ncol = n_species, dimnames = list(sites
= rownames(obj$L),species=colnames(obj$L)))
      obj.sim <- make_obj_for_traitenv(L,obj$E,obj$T)
    } else {
      if (options$model %in% c("RC","loglin")) model.coefs = options_RC else model.
coefs = options_Gau
      obj.sim <- generate_data(n_species,n_communities = n_sites, model= options$mo
del, model.coefs = model.coefs)
    }
    Scenario.list[[k]][[i]] = obj.sim
  }
}

save.image(paste("SimsForFig5",".Rdata ", sep = ""))
```

Code Series V

```
#Appendix to ter Braak 2019
#New robust weighted averaging- and model-based methods for assessing trait-environ
ment relationships.

rm(list=ls(all=TRUE))  # remove all existing items from the workspace
source("Rfunctions/generate_trait_env_community_data.r")
source("Rfunctions/utility_functions.r")


b_te_values <-  c(0) # b_te
b_tx_values <-  c(0,0.5) # sigma_b
b_ze_values <- c(0,0.5) # sigma_c
b_zx_star_values <-  c(0,0.5) # b_z*x*
cor_z_zstar <- 0 # rho_1
cor_x_xstar <- 0# rho_2
n_species_values <- 30
n_sites_values <- 50

scenarios <- expand.grid(b_te_values,n_sites_values, n_species_values, b_zx_star_va
lues, b_tx_values, b_ze_values)
names(scenarios) = c("b_te","n","m", "b_zx*","b_tx", "b_ze")

nrow(scenarios)

## [1] 8

scenarios

##   b_te  n  m b_zx* b_tx b_ze
## 1    0 50 30   0.0  0.0  0.0
## 2    0 50 30   0.5  0.0  0.0
## 3    0 50 30   0.0  0.5  0.0
## 4    0 50 30   0.5  0.5  0.0
## 5    0 50 30   0.0  0.0  0.5
## 6    0 50 30   0.5  0.0  0.5
## 7    0 50 30   0.0  0.5  0.5
## 8    0 50 30   0.5  0.5  0.5

n_simul <- 2 # 250

Scenario.list <-  vector("list", length = 2)

m0 <- 30
klist <- 1:nrow(scenarios)

Scenario.list <-  vector("list", length = nrow(scenarios))


for (k in klist){
  b_ze <- scenarios[k, "b_ze"]
  b_te <- scenarios[k, "b_te"]
  b_zx_star <- scenarios[k, "b_zx*"]
  b_tx <- scenarios[k, "b_tx"]
  n_species <- scenarios[k, "m"]
  n_sites <- scenarios[k, "n"]
  interacts <- set_interactions(b_te = b_te, b_ze =b_ze, b_tx = b_tx, b_zx_star = b
_zx_star, cor_x_xstar = cor_x_xstar, cor_z_zstar = cor_z_zstar)
  options_RC <- setoptions4RC( intercept =  m0,  interactions = interacts,
                      main_E = set_main(linear = 0.05, quadratic= -0.1, noise = 0.1
),# main effect models for E and x
                      main_T = set_main(linear = 0.05, quadratic= -0.1, noise = 0.1
```

```
),  # main effect models for T and z
                      sigma_epsilon_ij = 0.2,
                      link= "log", size= 0,
                      distribution = "negbin") #
  options <- options_RC
  options$model <- "RC"; options$MLM = NA
  if (options$model == "MLM"){
    # Simulation from an existing MLM model for which simulate(mod) exists(lme4 or
glmm.TMB model)
    mod <- options$MLM
    dat<- model.frame(mod)
    obj <- dat4MLM2TE_obj(dat) # TE_obj
    n_sites <- nrow(obj$L); n_species <- ncol(obj$L)
  } else {
    n_sites <- scenarios$n[k]; n_species <- scenarios$m[k]
    }
  # one trait one environmental variables  data----------------------------------
  for (i in 1:n_simul){
    if (options$model == "MLM"){
      Ysim <- simulate(mod)
      L <- matrix(Ysim[[1]], nrow = n_sites,ncol = n_species, dimnames = list(sites
= rownames(obj$L),species=colnames(obj$L)))
      obj.sim <- make_obj_for_traitenv(L,obj$E,obj$T)
    } else {
      if (options$model %in% c("RC","loglin")) model.coefs = options_RC else model.
coefs = options_Gau
      obj.sim <- generate_data(n_species,n_communities = n_sites, model= options$mo
del, model.coefs = model.coefs)
    }
    Scenario.list[[k]][[i]] = obj.sim
  }
}

save.image(paste("SeriesV",".Rdata ", sep = ""))
```

A10.2 generate_data

The code for the simulation model of Appendix A8.2 is as follows. The code is based on that of ter Braak et al. (2018).

References

ter Braak, C.J.F., Peres-Neto, P.R. & Dray, S. (2018) Simple parametric tests for trait–environment association. *Journal of Vegetation Science,* **29,** 801-811.

Code

```
#Appendix to ter Braak 2019
#New robust weighted averaging- and model-based methods for assessing trait-environ
ment relationships.

generate_data <- function(n_species,n_communities, model= "RC", model.coefs){
```

```
  # model = RC, Gau, Gau2d
  # for RC/loglin:
  #  model.coefs= list(coefs=c(b_te, b_ze, b_tx, b_zx_star),sigma_epsilon_ij  = sig
ma_epsilon_ij, mu0=mudat0,
  #                     coefsE=  coefsE,coefsT=  coefsT)
  #          coefsE = c(d1,d2,d3):   d1*E + d2*E^2 + d3*rnorm(n, 0,1)
  #          coefsT = c(d1,d2,d3):   d1*T + d2*T^2 + d3*rnorm(n, 0,1),
  #          coefsx = c(d1,d2,d3):   d1*x + d2*x^2 + d3*rnorm(n, 0,1),
  #          coefsz = c(d1,d2,d3):   d1*z + d2*z^2 + d3*rnorm(n, 0,1)
  #          added noise if d3 is nonzero
  # for Gau and Gau2d
  # model.coefs =
  # list(tolerance = tolerance,
  #       mu_max = mu0, rho_T = rho_T,  rho_E = rho_E, distribution = distri_Y,
  #       link.mu = "log", sigma = sigma, nu = nu)
  # data: one trait one environmental variables observed -------------------------
---------
  T <- sort(as.matrix(model.coefs$r_dist(n_species)))
  E <- sort(as.matrix(model.coefs$r_dist(n_communities)))
  if (model %in% c("RC", "loglin")){
    Y <- generate_RC_community(E= E, T= T, model.coefs)
    E_true = E; T_true = T
  } else if (model=="Gau2d"){
    # null model if rho_E and rho_T are length 1
    z1 <- rnorm(n_species); x1 <- rnorm(n_communities); err <-rnorm(n_communities)
    z2 <- rnorm(n_species); x2 <- rnorm(n_communities); trr <-rnorm(n_species)
    rho_E1=model.coefs$rho_E[1]
    rho_T1=model.coefs$rho_T[1]
    if (length(model.coefs$rho_E)==1) rho_E2=0 else rho_E2=model.coefs$rho_E[2]
    if (length(model.coefs$rho_T)==1) {rho_T2=model.coefs$rho_T[1]; rho_T1=0} else
rho_T2=model.coefs$rho_T[2]
    c1 <- (1-rho_E1^2-rho_E2^2); d1 <-(1-rho_T1^2-rho_T2^2)
    if (c1 <0 | d1<0) stop("sum square correlations in Gau2d exceed 1 in test_model
s()")
    E = rho_E1*x1 + rho_E2*x2+sqrt(1-rho_E1^2-rho_E2^2)*err
    T = rho_T1*z1 + rho_T2*z2+sqrt(1-rho_T1^2-rho_T2^2)*trr
    Y <- generate_community_Gaussian_response_2d(E = x1, T = z1,x=x2,z=z2, model.co
efs)
    E_true = x1; T_true = z1
  } else if (model=="Gau"){
    z <- rnorm(n_species); x <- rnorm(n_communities)
    rho_E=model.coefs$rho_E[1]
    rho_T=model.coefs$rho_T[1]
    z_star <- rho_T * T + sqrt(1 - rho_T^2)*z # rho_T = 0 == trait random; Eqn A.2
    x_star <- rho_E * E + sqrt(1 - rho_E^2)*x # rho_E = 1 == env NOT random  Eqn A.
2
    # thus: as rho_T = 0 and rho_E = 1
    # z_star = z # the trait governing the Gaussian model, but, alas, unobserved (l
atent)
    # x_star = E # the observed trait that is also the one governing the Gaussian m
odel
    Y <- generate_community_Gaussian_response(E = x_star, T = z_star, model.coefs)
    E_true = x_star; T_true = z_star
  }
  E<- as.matrix(E); colnames(E)= "E1"
  T <- as.matrix(T); colnames(T)= "T1"
  obj = list(L=Y, E = E, T = T) #, E_true = E_true, T_true = T_true, model.coefs= m
odel.coefs )
  class(obj) = c("TE_obj", class(obj))
  return(obj)
}

# function to generate data: RC-model/ loglinear model with y~NegBin eqn A.3-------
-------------
generate_RC_community <- function(E,T, model.coefs, fct = 0.8, max.try = 5){
  #
```

```
  # one environmental variable E, one trait T
  # main effects:
  #    row and column effects functions
  #       f_row and f_col: a polynomial in E, T,x and z respectively
  #           d1*E + d2*E^2 + d3*rnorm(n, 0,1), with coefsE = c(d1,d2,d3)
  #           d1*T + d2*T^2 + d3*rnorm(n, 0,1), with coefsT = c(d1,d2,d3)
  #           d1*x + d2*x^2 + d3*rnorm(n, 0,1), with coefsx = c(d1,d2,d3)
  #           d1*z + d2*z^2 + d3*rnorm(n, 0,1), with coefsz = c(d1,d2,d3)
  #           added noise if d3 is nonzero
  # interaction effects
  #        b1*T*E + b2*z*E + b3*T*x + b4*xx*zz eps with coefs = c(b1,b2,b3,b4)
  #        with z ~ N(0,1), x~N(0,1), zz ~ N(0,1), xx~N(0,1) and eps~sigma_epsilon_
ij*N(0,1)
  #        Note:  trait-environment association only if b1 unequal to 0
  #        Note:  confounding random effects if coefs[2:3] are none-zero
  #
  # dist ="negbin", "ZINBI" , "pois", "binomial"
  # link =  link function "log", "logit"
  # if center. = TRUE then the interaction is double centered
  # the repeat loop is to ensure that the final number of  species and sites is wit
hin fct (0.8) of the number asked for in  argument

  # The equi-tolerance versions of the Gaussian models
  # of Peres-Neto, Dray & ter Braak (2016, Ecography http://dx.doi.org/10.1111/ecog
.02302) and
  # ter Braak, Peres-Neres & Dray (2016, PeerJ) can be simulated
  # by setting coefsE-coefsz and coef as follows:
  # The environmentally structured, random trait case corresponds to
  # coefsE=c(0,-0.5/tolerance^2, 0);  coefsx=c(0,0,0)
  # coefsT=c(0,0, 0);  coefsz=c(0,-0.5/tolerance^2,0)
  # The trait structured, random environment case corresponds to
  # coefsE=c(0,0, 0);  coefsx=c(0,-0.5/tolerance^2,0)
  # coefsT=c(0,-0.5/tolerance^2, 0);  coefsz=c(0,0,0)
  # The both random case corresponds to
  # coefsE=c(0,0, 0);  coefsx=c(0,-0.5/tolerance^2,0)
  # coefsT=c(0,0, 0);  coefsz=c(0,-0.5/tolerance^2,0)

  #  local functions: marginal effects
  f_row <- function(E, coefsE){
    effects <- coefsE[1]*E + coefsE[2]*E*E + coefsE[3]*rnorm(length(E))
    effects
  }
  f_col <- function(T, coefsT){f_row(T,coefsE=coefsT)}
  # end local functions
  coefs =  model.coefs$coefs;
  coefsE=  model.coefs$coefsE
  coefsT=  model.coefs$coefsT
  coefsx=  model.coefs$coefsx
  coefsz=  model.coefs$coefsz
  coefsx_star=  model.coefs$coefsx_star
  coefsz_star=  model.coefs$coefsz_star
  cor_x_xstar <- coefs[5] #coefs["cor_x_xstar"]
  cor_z_zstar <- coefs[6] #coefs["cor_z_zstar"]
  sigma_epsilon_ij  = model.coefs$sigma_epsilon_ij; mu0=model.coefs$mu0
  distribution = model.coefs$distribution
  E <- c(E);
  T <- c(T)
  n_communities <- length(E)
  n_species <- length(T)
  i.try <- 1
  if (length(coefs)<6) coefs <- c(coefs,rep(0, 6-length(coefs)))
  repeat {
    i.try <- i.try + 1
    x <-  rnorm(length(E)); z <-  rnorm(length(T))
    xx <-  rnorm(length(E)); zz <-  rnorm(length(T))
    zz <- cor_z_zstar * z + sqrt(1 - cor_z_zstar^2)*zz # correlation of cor_z_star
```

```
between z and zstar
    xx <- cor_x_xstar * x + sqrt(1 - cor_x_xstar^2)*xx
    mu.interaction <-
      coefs[1] * outer(E,T,FUN = "*") +
      coefs[2] * outer(E,z,FUN = "*") +
      coefs[3] * outer(x,T,FUN = "*") +
      coefs[4] * outer(xx,zz,FUN = "*") +
      sigma_epsilon_ij * matrix(rnorm(n_communities*n_species), nrow =n_communities
)
    mu.marginal <- outer(rep(0,n_communities), f_col(T,coefsT), FUN="+") +
      outer(f_row(E,coefsE), rep(0,n_species), FUN="+") +
      outer(rep(0,n_communities), f_col(z,coefsz), FUN="+") +
      outer(f_row(x,coefsx), rep(0,n_species), FUN="+") +
      outer(rep(0,n_communities), f_col(z,coefsz_star), FUN="+") +
      outer(f_row(x,coefsx_star), rep(0,n_species), FUN="+")
    mu.lp <- mu.marginal + mu.interaction
    # for numerical stability we needed to limit mu.lp
    mu.lp <- ifelse(mu.lp > 10, matrix(10, nrow=nrow(mu.lp),ncol=ncol(mu.lp)), mu.l
p)
    mu.lp <- ifelse(mu.lp < -10,matrix(-10, nrow=nrow(mu.lp),ncol=ncol(mu.lp)), mu.
lp)
    if (mu0 > 0.9 && (distribution =="binomial"|| model.coefs$link.mu =="logit"))
mu0 <- 0.5 #
    mu <- switch(model.coefs$link.mu,
                 log =  mu0 * exp(mu.lp) ,
                 logit = 1/(1 + exp(-( mu.lp + log(mu0/(1-m0)))) )
    )
    L <- switch( distribution,
                 poisson = ZIpoisson(mu,nu=nu),
                 binomial = matrix(rbinom(prod(dim(mu)),size = model.coefs$size, mu
),nrow = nrow(mu),ncol = ncol(mu)),
                 negbin = ZInbinom(mu = mu, sigma= model.coefs$sigma, nu = 0),
                 ZINBI =  ZInbinom(mu = mu, sigma= model.coefs$sigma, nu = model.co
efs$nu)
    )
    nspecies_c <- sum(colSums(L)!=0) # _c for check
    ncommunities_c <- sum(rowSums(L)!=0)
    if ((nspecies_c >= fct * n_species) && (ncommunities_c >=fct *n_communities ||
i.try> max.try)){break}
  }
  rownames(L)<- paste("site", seq_len(nrow(L)), sep="")
  colnames(L)<- paste("spec",seq_len(ncol(L)),sep="")
  return(L) # L = Y in the paper
}

set_interactions <- function(b_te = 0, b_ze =0.2, b_tx = 0.2, b_zx_star =0.2, cor_x
_xstar = 0,cor_z_zstar=0){
  return(c(b_te=b_te, b_ze= b_ze, b_tx= b_tx, b_zx_star= b_zx_star, cor_x_xstar = c
or_x_xstar,cor_z_zstar = cor_z_zstar))
}
set_main <- function(linear= 0, quadratic= 0, noise= 0){
  return(c(linear = linear, quadratic = quadratic, noise = noise))
}
#model.coefs.RC=
setoptions4RC <- function( intercept = 30,  interactions = set_interactions(),
                           main_E = set_main(linear = 0.05, quadratic= -0.1, noise
= 0.1),# main effect models for E and x
                           main_T = set_main(linear = 0.05, quadratic= -0.1, noise
= 0.1),  # main effect models for T and z
                           main_x = set_main(0,0,0), # main effect models for E and
x
                           main_z = set_main(0,0,0), # main effect models for T and
z
                           main_xstar = set_main(0,0,0),
                           main_zstar = set_main(0,0,0),
                           sigma_epsilon_ij = 0.2,
```

```
                            link= "log",
                            size = 100,
                            distribution = "negbin",
                            dispersion = 1,
                            p_zero_inf = 0,
                            r_dist = rnorm #
){
  # distribution ="negbin", "ZINBI" , "pois", "binomial"
  #link =  link function "log", "logit"
  # r_dist = distribution of T and E  (only in RC/loglin models)
  return(
    list(coefs=interactions,
         coefsE=  main_E,
         coefsT=  main_T,
         coefsx=  main_x,
         coefsz=  main_x,
         coefsx_star=  main_xstar,
         coefsz_star=  main_zstar,
         sigma_epsilon_ij  = sigma_epsilon_ij, mu0=intercept,
         distribution = distribution, size = size,
         link.mu = link, sigma = dispersion, nu = p_zero_inf, r_dist = r_dist)
  )
}
setoptions4Gau <- function(tolerance = 2,  mu_max = 10, rho_T = c(1,0),  rho_E = c(
1,0),
                            link = "log",  distribution = "negbin", dispersion = 1,
p_zero_inf = 0, r_dist = rdist_norm ){
  return(list(tolerance = tolerance, mu_max = mu_max, rho_T = rho_T,  rho_E = rho_E
, distribution = distribution, link.mu = link, sigma = dispersion, nu = p_zero_inf,
r_dist = r_dist))
}

# function to generate one- and multi-trait data ---------------------------------
-----
generate_Tdata <-function(n_species,n_traits, correlation = 0, b){
  # n_species = number of species; n_traits = number of traits;
  # correlation = AR(1) autocorrelation between subsequent variables
  # b = coefficient of the linear combination (c in the paper)
  # generate n_traits correlated normally distibuted trait variables in Tdat
  # a linear combination of these  Tdat%*%b  in T
  # rescale b and T so that T ~ N(0,1),
  # value: list with Tdat, T and coefficients (rescaled b)
  ii <- 1:n_traits
  Sigma <-  correlation^abs(outer(ii,ii,  '-'))
  sdT <-  sqrt(b %*% Sigma %*%b)
  coefs <- b/c(sdT)
  Tdat<-MASS::mvrnorm(n_species,mu=rep(0,n_traits),Sigma)
  T <- Tdat %*%coefs
  rownames(T)<- rownames(Tdat)<- paste("spec", seq_len(nrow(Tdat)), sep="")
  return(list(T = T, Tdat= Tdat, coefficients = coefs))
}

generate_Edata <-function(n,p, correlation = 0, b, mixfactor = 0){
  # generate p correlated normally distibuted E variables in Edat
  # a linear combination of these in E
  # rescale b and E so that E ~N(0,1),
  # value: list with Edat, E and coefficients (rescaled b)
  ii <- 1:p
  Sigma <-  correlation^abs(outer(ii,ii,  '-'))
  sdE <-  sqrt(b %*% Sigma %*%b)
  coefs <- b/c(sdE)
  Edat<-MASS::mvrnorm(n,mu=rep(0,p),Sigma) + mixfactor * matrix(2*((runif(n*p) > 0.
5)-0.5),nrow = n,ncol =p)
  E <- Edat %*%coefs
  rownames(E)<- rownames(Edat)<- paste("site", seq_len(nrow(Edat)), sep="")
  return(list(E = E, Edat= Edat, coefficients = coefs))
```

```
}

# function to generate (zero inflated ) negative binomial data ------------
ZInbinom <- function(mu,sigma = 1, nu= 0){
  L <- rnbinom(length(mu), size = 1/sigma, mu = mu)
  # variance = mu + mu^2/size = mu + sigma * mu^2
  L <- ifelse(runif(length(mu)) < nu,  0,L)
  matrix(L,nrow = nrow(mu),ncol = ncol(mu))
}

# function to generate (zero inflated ) poisson data ------------
ZIpoisson <- function(mu, nu= 0){
  L <- rpois(length(mu), lambda = mu)
  L <- ifelse(runif(length(mu)) < nu,  0,L)
  matrix(L,nrow = nrow(mu),ncol = ncol(mu))
}


# function to generate data according to the Gaussian response model Eqn A1.1 and y
~NegBin-----------------------------------------------
generate_community_Gaussian_response <- function(E,T, model.coefs,  n_species = len
gth(T),n_communities = length(E), fct = 0.8, max.try = 5, equi.tol=FALSE){
  # using outer instead of a for loop
  # tolerance sets the maximum width of the random sigma_j in the paper (set to 2 i
n the paper)
  # with checks on empty rows and columns
  # the repeat loop is to ensure that the final number of  species and sites is wit
hin fct (0.8) of the number asked for in  argument
    i.try <- 0
    distribution = model.coefs$distribution
    tolerance =    model.coefs$tolerance
    mu_max = model.coefs$mu_max
   # sigma = 1, nu= 0
  repeat {
    i.try <- i.try + 1
    # one trait, one environmental variable
    E <- c(E)
    T <- c(T)
    if (equi.tol){
      h<-  rep(0.7, n_species)
      sigma<- rep(tolerance,n_species)
    } else{
      h<-  runif(n_species,min=0.3,max=1)
      sigma<- runif(n_species)*tolerance
    }
    mu <- outer(rep(mu_max,n_communities),h) * exp(outer(c(E),c(T),FUN = "-")^2 / o
uter(rep(1,n_communities),-2*sigma^2))+1.e-10
    L <- switch( distribution,
                 poisson = ZIpoisson(mu,nu=nu),
                 binomial = matrix(rbinom(prod(dim(mu)),1, mu),nrow = nrow(mu),ncol
= ncol(mu)),
                 negbin = ZInbinom(mu = mu, sigma= model.coefs$sigma, nu = 0),
                 ZINBI =  ZInbinom(mu = mu, sigma= model.coefs$sigma, nu = model.co
efs$nu)
    )
    n_species_c<-sum(colSums(L)!=0) # _c for check
    n_communities_c<-sum(rowSums(L)!=0)
    if ((n_species_c >= fct * n_species) && (n_communities_c >=fct *n_communities |
| i.try > max.try)){break}
  }
  rownames(L)<- paste("site", seq_len(nrow(L)), sep="")
  colnames(L)<- paste("spec",seq_len(ncol(L)),sep="")
  return(L)  # L = Y in the paper
}
```

```
generate_community_Gaussian_response_2d <- function(E,T, x,z, model.coefs,  n_speci
es = length(T),n_communities = length(E), fct = 0.8, max.try = 5, equi.tol=FALSE){
  # using outer instead of a for loop
  # tolerance sets the maximum width of the random sigma_j in the paper (set to 2 i
n the paper)
  # with checks on empty rows and columns
  # the repeat loop is to ensure that the final number of  species and sites is wit
hin fct (0.8) of the number asked for in  argument
  i.try <- 0
  distribution = model.coefs$distribution
  tolerance =    model.coefs$tolerance[1]
  if (length(model.coefs$tolerance)==1) tolerance2 = model.coefs$tolerance[1] else
tolerance2 =    model.coefs$tolerance[2]
  mu_max = model.coefs$mu_max
  # sigma = 1, nu= 0
  repeat {
    i.try <- i.try + 1
    # one trait, one environmental variable
    E <- c(E)
    T <- c(T)
    # x <-  rnorm(length(E)); z <-  rnorm(length(T))
    # xx <-  rnorm(length(E)); zz <-  rnorm(length(T))
    if (equi.tol){
      h<-  rep(0.7, n_species)
      sigma<- rep(tolerance,n_species)
      sigma2<- rep(tolerance2,n_species)

    } else{
      h<-  runif(n_species,min=0.3,max=1)
      sigma<- runif(n_species)*tolerance
      sigma2<- runif(n_species)*tolerance2
    }
    mu <- outer(rep(mu_max,n_communities),h) *
      exp(outer(c(E),c(T),FUN = "-")^2 / outer(rep(1,n_communities),-2*sigma^2) +
            outer(c(x),c(z),FUN = "-")^2 / outer(rep(1,n_communities),-2*sigma2^2)
      )+1.e-10
    L <- switch( distribution,
                 poisson = ZIpoisson(mu,nu=nu),
                 binomial = matrix(rbinom(prod(dim(mu)),1, mu),nrow = nrow(mu),ncol
= ncol(mu)),
                 negbin = ZInbinom(mu = mu, sigma= model.coefs$sigma, nu = 0),
                 ZINBI =  ZInbinom(mu = mu, sigma= model.coefs$sigma, nu = model.co
efs$nu)
    )
    n_species_c<-sum(colSums(L)!=0) # _c for check
    n_communities_c<-sum(rowSums(L)!=0)
    if ((n_species_c >= fct * n_species) && (n_communities_c >=fct *n_communities |
| i.try > max.try)){break}
  }
  rownames(L)<- paste("site", seq_len(nrow(L)), sep="")
  colnames(L)<- paste("spec",seq_len(ncol(L)),sep="")
  return(L)  # L = Y in the paper
}
```