Proper experimental design requires randomization/balancing of molecular ecology experiments

Miklós Bálint, Orsolya Márton, Marlene Schatz, Rolf-Alexander Düring, Hans-Peter Grossart

OBITools sequence processing

OBITools v1.2.7, downloaded on July 17, 2016. Sequence processing was performed in a `bash` terminal on a linux computer.

Step 0: create a small dataset to check the script

parallel citation: **Tange 2011**

```
mkdir data
cd data
ln -s /phylodata/mbalint/datasets/160329_Stechlin-preexp/data/*.fastq .
mkdir complete_files
mv *.fastq complete_files/
cd complete_files
ls *.fastq | parallel -j 60 'head -n 400 {} > trial_{.}.fastq'
mv trial*.fastq ../
cd ../
```

Step 1: Quality check

```
mkdir 01_fastqc
ls *.fastq | parallel -j 60 'fastqc -o 01_fastqc'
mv 01_fastqc /phylodata/mbalint/workdir/Stechlin_preexp
cd /phylodata/mbalint/workdir/Stechlin_preexp/01_fastqc
rm *.zip
```

Step 2: Trimming

Check the trimming length with the primers. *Guardiola 2015*: Euka2-F (TYTGTCTGSTTRATTSCG 18bp), Euka2-R (TCACAGACCTGTTATTGC 18bp) Fragment length: most species ~110 bp, some ~150 bp + primers 2 x 20 bp All reads trimmed to 100 bp: this should be shorter then the fragment length, and allows the assembly of the ~150 bp reads (fragment + primer)

```
cd /phylodata/mbalint/workdir/Stechlin_preexp
mkdir 02_trimmed
cd 02_trimmed
ln -s ../data/*.fastq . # link because the parallel input is simpler for me this way
ls *.fastq | parallel -j 60 'fastx_trimmer -t 51 -i {} -o {.}_trimmed.fastq' # -t: remove 51 nucleotide
rm *001.fastq # remove the links. the -j 60 option of the parallel uses 60 processors
```

Step 3: Paired-end assembly

```
cd /phylodata/mbalint/workdir/Stechlin_preexp
mkdir 03_paired-end
cd 03_paired-end
```

Link all files into present folder, because it is less complicated to use the parallel inputting for me

```
ln -s ../02_trimmed/*trimmed.fastq .
```

```
ls *trimmed.fastq | grep -v R2_001_trimmed.fastq | sed 's/R1_001_trimmed.fastq//' | parallel -j 60 'illu
```

1. list the filenames.
2. keep the file name stems by removing the forward-reverse specific file information with ‘sed’.
3. give the filename stems to parallel as `{.}`.

remove the links

```
rm *001_trimmed.fastq
```

Paired quality check

```
mkdir fastqc_paired # create a folder for the fastqc output
ls *_paired.fastq | parallel -j 60 'fastqc -o fastqc_paired {}'
rm fastqc_paired/*.zip # keep only the htmls
```

Remove primers from the read ends Primers are 18 bp Insert the sequence length (seq_length) into the headers

```
ls *.fastq | parallel -j 60 'obiannotate --length --fasta-output {} > {.}_wlength.fasta'
```

Trim the reverse primers from the read ends. Reverse primer bases calculated with the seq_length

```
ls *.fasta | parallel -j 60 'obicut -b 19 -e seq_length-18 --uppercase {} > {.}_noprimer.fasta'
```

Step 4. Remove unaligned

```
cd /phylodata/mbalint/workdir/Stechlin_preexp
mkdir 04_unaligned
cd 04_unaligned
```

the steps are different from the conventional OBITools pipeline from now on: the samples are already demultiplexed, but the sample names are not yet in the fastq header lines.

The sample names should be inserted as ‘sample=xxx;’ before the ‘ali_length=xxx;’ The rename.pl is a small script modified from **Bálint Ecol Evol** Original in /phylodata/mbalint/scripts

```
cp /phylodata/mbalint/scripts/rename_fasta.pl . # copy the rename script
ln -s ../03_paired-end/*_noprimer.fasta .
perl rename_fasta.pl
rm *noprimer.fasta
```

Now combine the files

```
cat *renamed.fasta > combined.fasta
obigrep -p 'mode!="joined"' combined.fasta > combined_ali.fasta
```

Successfully assembled sequences

```
grep -c "^>" combined_ali.fasta
```

Remove sequences shorter, that 50 bp

```
obigrep -l 50 combined_ali.fasta > combined_noshort.fasta
grep -c "^>" combined_noshort.fasta
```

Step 5: chimera checking

```
cd /phylodata/mbalint/workdir/Stechlin_preexp
mkdir 05_chimera
cd 05_chimera
ln -s ../04_unaligned/combined_noshort.fasta .
```

vsearch for chimera checking Documentation here: https://wiki.gacrc.uga.edu/wiki/Vsearch#Documentation

```
vsearch --threads 60 --uchime_denovo combined_noshort.fasta --uchimeout chimera_results.tab --nonchimera
```

Step 6: De-replicate into unique sequences

```
cd /phylodata/mbalint/workdir/Stechlin_preexp
mkdir 06_derep
cd 06_derep
```

Since all were non-chimeric and the VSEARCH chimera checking screws the header, let's use the chimieracheck input file

```
obiuniq -m sample ../05_chimera/combined_noshort.fasta > stechlin_derep.fasta
grep -c "^>" stechlin_derep.fasta
```

Step 7: Denoise the dataset

Get the counts of the 20 rarest sequences

```
cd /phylodata/mbalint/workdir/Stechlin_preexp
mkdir 07_denoise
cd 07_denoise
obistat -c count ../06_derep/stechlin_derep.fasta | sort -nk1 | head -20 > rare_counts.txt
```

Keep only the sequence variants having a count greater or equal to 10:

```
obigrep -p 'count>=10' ../06_derep/stechlin_derep.fasta > stechlin_c10.fasta
grep -c "^>" stechlin_c10.fasta
```

Step 8: Clean the sequences

```
cd /phylodata/mbalint/workdir/Stechlin_preexp
mkdir 08_clean
cd 08_clean
```

keep only head sequences (-H option) if these are sequences with no variants with a count greater than 5% of their own count (-r 0.05 option). This also annotates sequences as head, internal or singleton in a sample.

```
obiclean -s merged_sample -r 0.05 -H ../07_denoise/stechlin_c10.fasta > stechlin_clean.fasta
grep -c "^>" stechlin_clean.fasta
```

Step 9: ecoPCR databases

Download and format the databases. These steps are commented out if the databases exist. Actual database EMBL v128

```
cd /phylodata/mbalint/databases/
mkdir EMBL_128
cd EMBL_128
wget ftp://ftp.ebi.ac.uk/pub/databases/embl/release/std/rel_std*.*
```

Get the actual GenBank taxonomy

```
cd /phylodata/mbalint/databases/
mkdir Taxonomy_160908
cd Taxonomy_160908
wget ftp://ftp.ncbi.nih.gov/pub/taxonomy/taxdump.tar.gz
tar -zxvf taxdump.tar.gz
```

Create the EMBL v128 ecoPCR database. Use ‘–skip-on-error’ to get through some problematic EMBL entries. It may not be necessary to unpack ‘gz’ beforehand, try using ’*.dat.gz’

```
cd /phylodata/mbalint/databases
mkdir ecoPCR_embl_128
cd ecoPCR_embl_128/
obiconvert --skip-on-error --embl -t ../Taxonomy_160908 --ecopcrdb-output=ecopcr_embl_128 ../EMBL_128/*
```

Generate an ecoPCR assignment database with the Euka2 primers

```
cd /phylodata/mbalint/workdir/Stechlin_preexp
mkdir 09_ecopcr
cd 09_ecopcr
ecoPCR -d /phylodata/mbalint/databases/ecoPCR_embl_128/ecopcr_embl_128 -e 3 -l 50 -L 500 TCACAGACCTGTTAT
grep -cv '#' *.ecoPCR
```

Clean the databases.

Filter sequences so that they have a good taxonomic description at the species, genus, and family levels

```
obigrep -d /phylodata/mbalint/databases/ecoPCR_embl_128/ecopcr_embl_128 --require-rank=species --requir
```

remove redundant sequences (obiuniq command below).

```
obiuniq -d /phylodata/mbalint/databases/ecoPCR_embl_128/ecopcr_embl_128 i18S_V9_clean.fasta > i18S_V9_cl
```

ensure that the dereplicated sequences have a taxid at the family level (obigrep command below).

```
obigrep -d /phylodata/mbalint/databases/ecoPCR_embl_128/ecopcr_embl_128 --require-rank=family i18S_V9_cl
```

ensure that sequences each have a unique identification (obiannotate command below)

```
obiannotate --uniq-id i18S_V9_clean_uniq_clean.fasta > db_i18S_v9.fasta
```

Step 10: Taxonomic assignment

```
cd /phylodata/mbalint/workdir/Stechlin_preexp
mkdir 10_assign
cd 10_assign
ecotag -d /phylodata/mbalint/databases/ecoPCR_embl_128/ecopcr_embl_128 -R ../09_ecopcr/db_i18S_v9.fasta
grep -c ">" stechlin_assigned.fasta
```

Step 11: Abundance tables

```
cd /phylodata/mbalint/workdir/Stechlin_preexp
mkdir 11_abundance_tables
cd 11_abundance_tables
```

create assigned abundance matrix

```
obitab -o --output-field-separator="|"  ../10_assign/stechlin_assigned.fasta > stechlin_assigned_190915
```

R analyses

Load libraries

```r
library(vegan)
```

```
## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.4-1
```

```r
library(effects)
library(lme4)
```

```
## Loading required package: Matrix
```

```r
library(mvabund)
```

Read in data

Get the data here: https://figshare.com/s/32dbca0a906c7f06449b

```r
# Read abundances
EmblAssign = read.csv(file="stechlin_assigned_190915.tab",
                      header=T, sep='|', row.names = 1)

# Read experimental setup data
ExpSet = read.csv(file = "sample_infos.csv",
                  header = T, row.names = 1)
RepliExp = ExpSet[1:96,]

# re-order the horizon identities
RepliExp$depth.nominal = factor(RepliExp$depth.nominal)

# re-order the nuclear power plant periods
RepliExp$nuclear = factor(RepliExp$nuclear,
                          levels=c("before",
                                   "during", "after"))
```

How many reads are there?

```r
IndexSample = grep("sample.[EMPS]", names(EmblAssign))
sum(EmblAssign[,IndexSample])
```

```
## [1] 612963
```

Data cleanup

Heads, singletons, internals

This step is specific to the OBITools sequence processing pipeline.

Which columns have the status info for head, internal, singletons?

```
StatusEmbl = EmblAssign[,grepl("obiclean.status", names(EmblAssign))]
```

Keep only sequence variants that were seen at least once as ‘head’ or the ‘singleton’ count is higher than the ‘intermediate’ count.

```
EmblHead = EmblAssign[(EmblAssign$obiclean_headcount) > 0 |
                        EmblAssign$obiclean_singletoncount > EmblAssign$obiclean_internalcount,]
```

Clean up negative controls

Remove the maximum read number of a sequence variant found in a negative control from every sample that contains that variant

Select the extraction controls

```
# Extraction controls
ExtCont = grep("sample.EXT", names(EmblHead))

# PCR controls
PCRCont = grep("sample.PCR", names(EmblHead))

# Multiplexing controls
MPXCont = grep("sample.MPX", names(EmblHead))
```

Write out negative control assignments for more analysis

```
NegCont = cbind(name = EmblHead$scientific_name,
                EmblHead[,c(ExtCont, PCRCont, MPXCont)])

# Aggregate according to taxon
NegRegate = aggregate(. ~ name, NegCont, sum, na.action = na.exclude)
rownames(NegRegate) = NegRegate$name
NegRegate = NegRegate[,2:length(names(NegRegate))]

# Remove taxa that were not seen in negatives
NegRegate = NegRegate[apply(NegRegate,1,sum) > 0,]

# Write for checking negatives
write.csv(file = "negative_control_identities.csv", NegRegate)
```

How many reads are there before cleaning up the negatives?

```
sum(EmblHead[,IndexSample])
```

```
## [1] 575718
```

`sweep` the maximum reads of a sequence variant in any controls from all samples

```
# warnings supressed because of a numeric-non-numeric substitution,
# see below
```

```r
MaxControl = apply(EmblHead[,c(ExtCont,PCRCont,MPXCont)], 1, max)

EmblControlled = EmblHead

# sweep with the MaxControl
EmblControlled[,grep("sample", names(EmblHead))] <-
  sweep(EmblHead[,grep("sample", names(EmblHead))], 1, MaxControl, "-")

# Set negative values to 0. Warnings are because the non-numeric cells
# There are warnings since many fields are non-numeric.
EmblControlled[EmblControlled < 0] <- 0

# Remove sequence variants with no reads left
EmblControlled = EmblControlled[apply(EmblControlled[,grep("sample", names(EmblHead))],1,sum) > 0,]
```

How many reads are there after cleaning up the negatives?

```r
sum(EmblControlled[,IndexSample])
```

```
## [1] 493182
```

Write sequence variants of taxa into table

```r
write.csv(file = "variant_sequences.csv",
          data.frame(name = rownames(EmblControlled),
                     name = EmblControlled$scientific_name,
                     genus = EmblControlled$genus_name,
                     family = EmblControlled$family_name,
                     seq = EmblControlled$sequence))
```

Rare read cleanup with positives controls

Taxa in positive controls

```r
PosCont = grep("^sample.POS", names(EmblControlled))

# Abundances in positives and taxonomic annotations
PosAbundances = EmblControlled[,c(PosCont,409,10,9)]

# Sums of all reads / OTU and add to positive table
SumReads = apply(EmblControlled[,IndexSample], 1, sum)
PosAbundances = cbind(PosAbundances, SumReads)

# keep only OTUs that had reads
PosAbundances = PosAbundances[apply(PosAbundances[,1:2],1,sum) > 0,]
```

Write out positive control samples + scientific name + genus + family name.

```r
write.csv(file="positive_controls.csv", PosAbundances)
```

I manually compared the reads of these OTUs to the taxon list and DNA concentration list of the positives. I looked for a read count rarity threshold of the complete read numbers `942742` that would reflect the original positive taxon lists.

When an OTU is present with less, than `14` reads in a replicate (`0.0015%` of the total reads `sum(ExpSet$reads)` 942742, is considered erroneous and set to `zero`.

```
RareControlled <- EmblControlled
RareControlled[RareControlled < 14] <- 0
```

How many reads after removing rare observations?

```
sum(RareControlled[,IndexSample])
```

```
## [1] 423547
```

Effects of laboratory biases

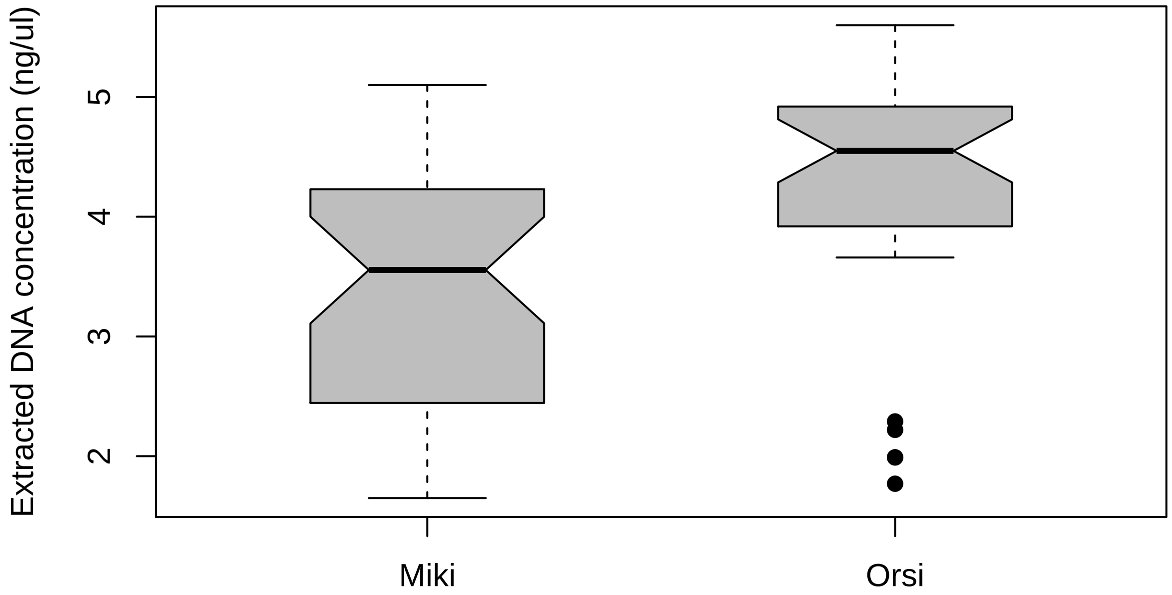
I use all replicates here however weird. The reason - the aim here is to quantify the effects of potentially systemic lab biases, so all biases qualify except standard contamination issues and rare weird sequences.

Generate the needed matrices: OTU abundances and corresponding predictors.

```
# Predictors of lab effects
ExpLab = RepliExp[13:96,]

# OTU abundance matrix
OTULab = RareControlled[,IndexSample]
OTULab = data.frame(t(OTULab[,13:96]))

# remove OTUs with zero reads
OTULab = OTULab[,apply(OTULab,2,sum) > 0]

# there is a sample with no reads left, it is removes
OTULab = OTULab[apply(OTULab,1,sum) > 0,]

# Adjust the explaining vars
ExpFilter = rownames(ExpLab) %in% rownames(OTULab)

ExpLab = ExpLab[ExpFilter,]
```

The possible systemic bias to be tested is the person who performed the DNA extraction. The extractions were initially planned to be performed by the first author of the paper, but finally the second half of the extractions were performed by the second author due to an unexpected urgency. Visually it seems that the extractions done by the second author generally yielded more DNA.

```
boxplot(ExpLab$conc ~ ExpLab$person,
     ylab="Extracted DNA concentration (ng/ul)",
     main = "", pch = 19, col="grey", boxwex=0.5, notch = T)
```



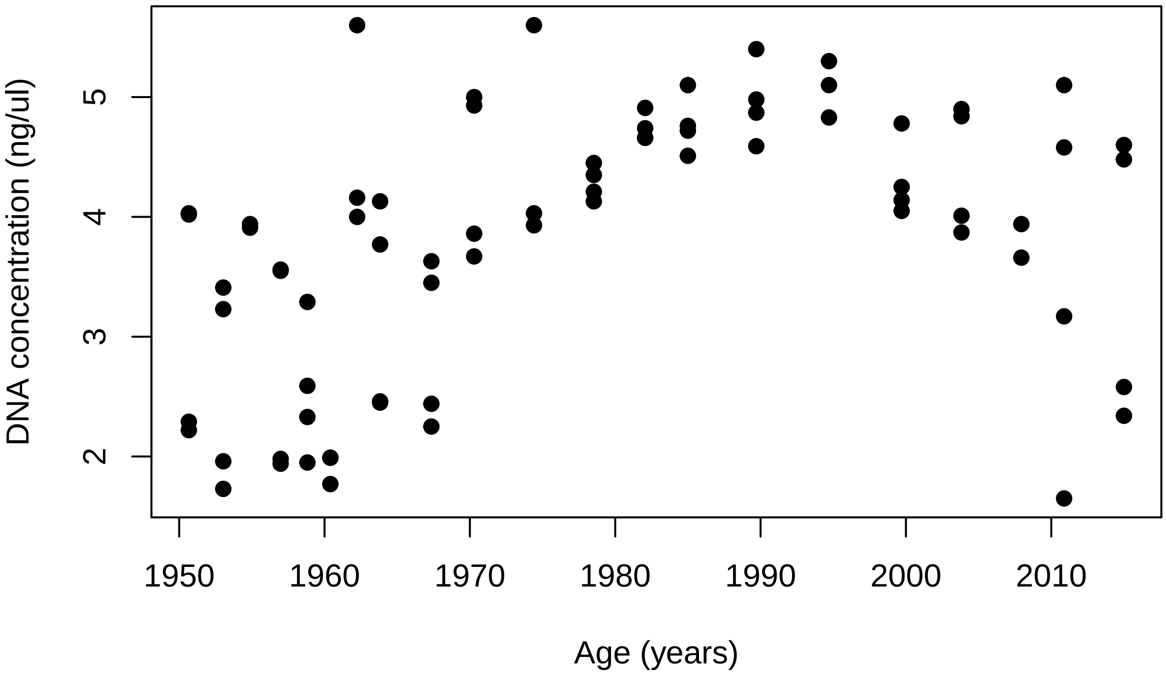
Linear mixed effect models are used for the singe response models with the sediment profile identity as the random effect.

The variables of interest and their predictors

1. DNA concentration

- Expected laboratory effect:
  - sample weight
  - extraction kit
- Unexpected laboratory effect: +lab person
- Biological effect of interest:
  - the age of the sediment

```r
plot(ExpLab$conc ~ ExpLab$age, pch=19,
     xlab = "Age (years)", ylab = "DNA concentration (ng/ul)")
```



2. PCR success as indicated by the number of sequence reads per sample. PCR product concentrations were not normalized before multiplexing the samples for sequencing, thus the differences in read numbers may be used to evaluate PCR performance

- Expected laboratory effect:
  - DNA concentration
  - extraction kit
- Unexpected laboratory effect:
  - lab person
- biological effect of interest:
  - sediment age

3. Community structure: diversity indicators (Hill's 1st, 2nd and 3rd numbers) and community composition

- Expected laboratory effect:
  - read numbers
  - extraction kit
- Unexpected laboratory effect:
  - lab person
- biological effect of interest:
  - the effects of the construction/operation period of a nuclear power plant (1960 - 1990). We know that lake communities were strongly changed after building the plant from previous morphology-based works.

DNA concentration

Fit the full model

```
conc.weight.kit.person.age =
  lmer(conc ~ weight + kit +
       person + age +
         (1|depth.nominal),
     data = ExpLab)
```

```
conc.weight.kit.age =
  lmer(conc ~ weight + kit +
       age +
         (1|depth.nominal),
     data = ExpLab)
```

Most of the variation in the DNA concentration is explained by the isolation kit. The lab person explains about as much variation as the age of the sediment.

```
anova(conc.weight.kit.person.age)
```

```
## Analysis of Variance Table
##        Df  Sum Sq Mean Sq F value
## weight  1  0.0562  0.0562  0.3278
## kit     1 12.9499 12.9499 75.4851
## person  1  1.7096  1.7096  9.9650
## age     1  1.8260  1.8260 10.6438
```

The DNA concentration model is marginally statistically significantly improved by accounting for the lab person.

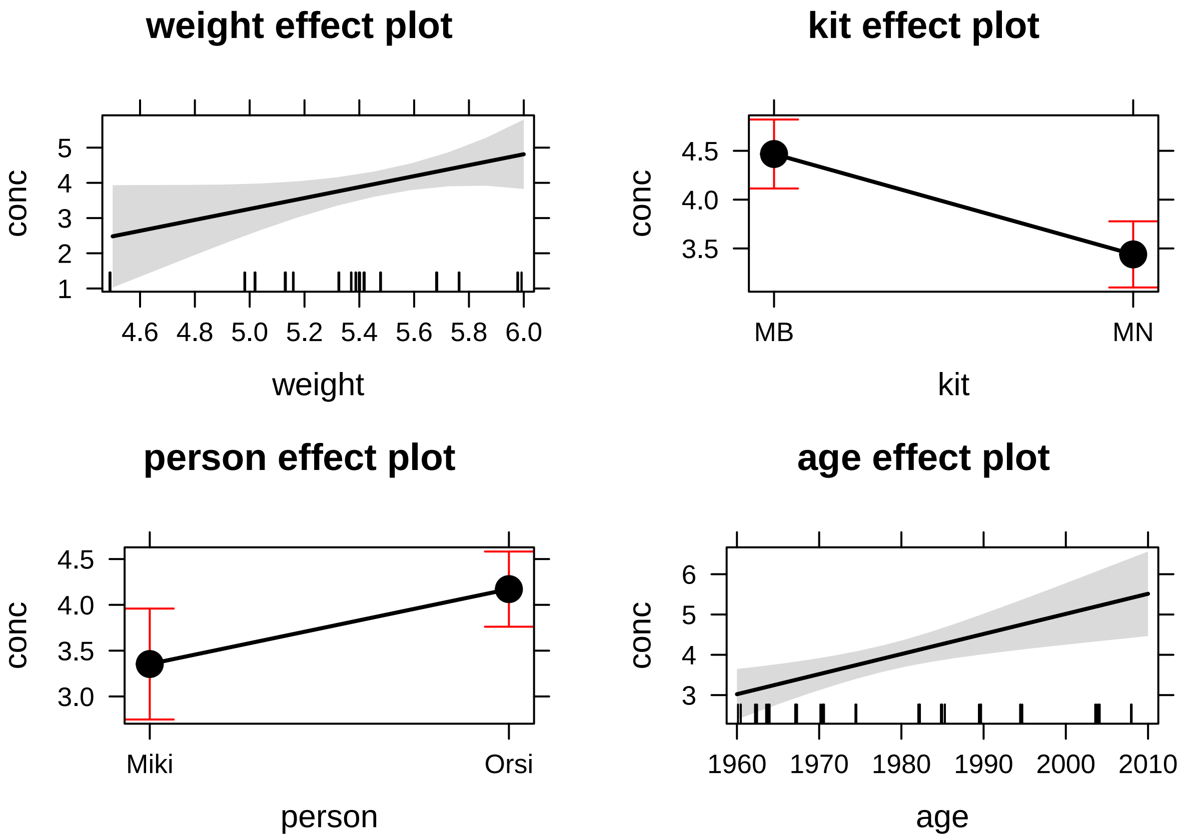
```
anova(conc.weight.kit.person.age,
      conc.weight.kit.age)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: ExpLab
## Models:
## conc.weight.kit.age: conc ~ weight + kit + age + (1 | depth.nominal)
## conc.weight.kit.person.age: conc ~ weight + kit + person + age + (1 | depth.nominal)
##                            Df    AIC    BIC  logLik deviance  Chisq Chi Df
## conc.weight.kit.age         6 105.66 117.82 -46.832   93.664
## conc.weight.kit.person.age  7 102.43 116.61 -44.215   88.430 5.2339      1
##                            Pr(>Chisq)
## conc.weight.kit.age
## conc.weight.kit.person.age    0.02215 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Visualize the variable effects. For some reason the `effects` package does not plot into the knitted document, so I use a saved image.

```
plot(allEffects(conc.weight.kit.person.age,
                multiline=TRUE, confidence.level = 0.95))
```



PCR success

```
read.conc.kit.pers.age =
  lmer(reads ~ conc + kit + person + age + (1|depth.nominal),
     data = ExpLab)

read.conc.kit.age =
  lmer(reads ~ conc + kit + age + (1|depth.nominal),
       data = ExpLab)
```

The lab person explained far the most variation in PCR success, with the samples extracted by the second authors yielding consistently less reads (all PCR reactions were performed by the second author).

```
anova(read.conc.kit.pers.age)
```

```
## Analysis of Variance Table
##        Df   Sum Sq  Mean Sq F value
## conc    1    58222    58222  0.0064
## kit     1  2719604  2719604  0.2969
## person  1 81413118 81413118  8.8891
## age     1   198964   198964  0.0217
```

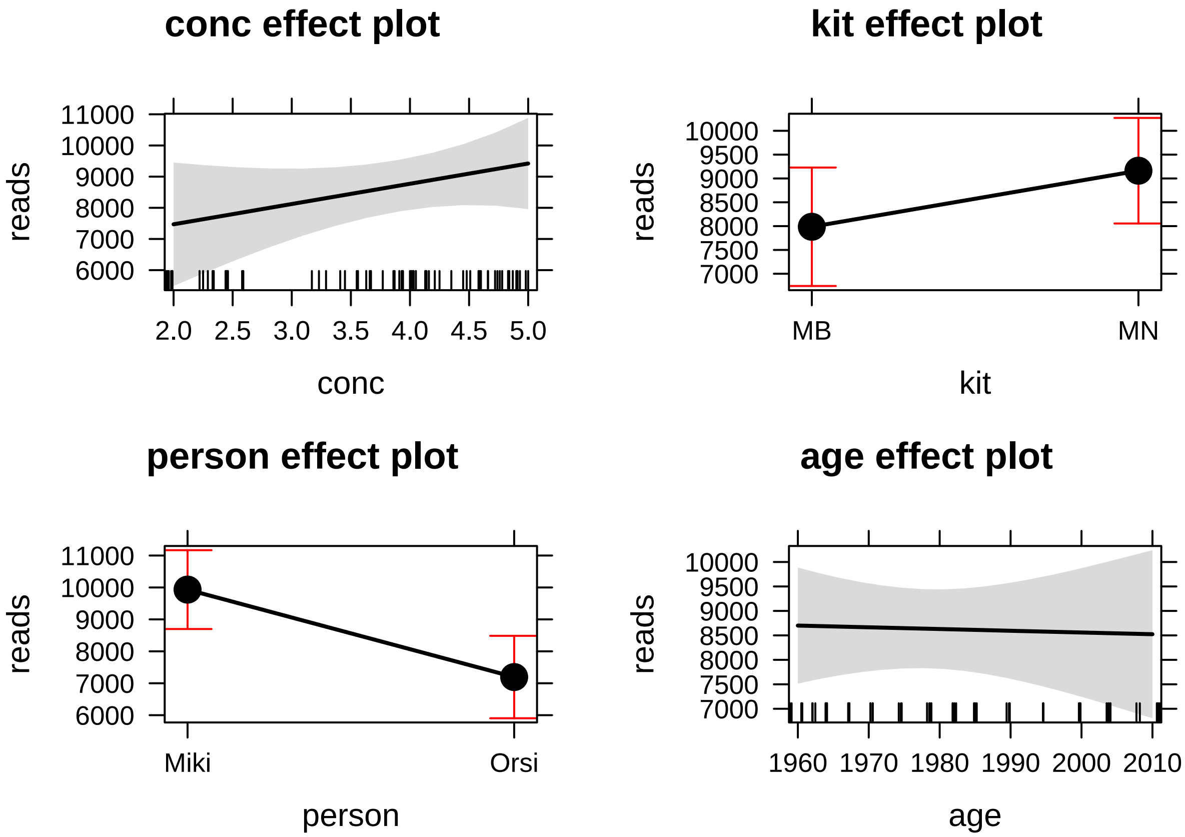
The model of PCR success was statistically significantly improved by considering the lab person.

```
anova(read.conc.kit.pers.age,
      read.conc.kit.age)
```

```
## refitting model(s) with ML (instead of REML)
## Data: ExpLab
## Models:
## read.conc.kit.age: reads ~ conc + kit + age + (1 | depth.nominal)
## read.conc.kit.pers.age: reads ~ conc + kit + person + age + (1 | depth.nominal)
##                        Df    AIC    BIC  logLik deviance Chisq Chi Df
## read.conc.kit.age       6 1454.8 1468.7 -721.37   1442.8
## read.conc.kit.pers.age  7 1448.8 1465.2 -717.42   1434.8 7.917      1
##                        Pr(>Chisq)
## read.conc.kit.age
## read.conc.kit.pers.age   0.004897 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The **DNA concentration** had no consistent effects on the PCR success within a 95% confidence interval (although PCRs produced more reads with higher template concentrations). The **age** of the sediment had no effect on PCR success.

```r
plot(allEffects(read.conc.kit.pers.age,
                multiline=TRUE, confidence.level = 0.95))
```



Community structure

Diversity indicators

Hill's first three numbers of diversity corrspond to richness (Hill's 1), the exponent of Shannon diversity (Hill's 2) and the inverse of the Simpson diversity (Hill's 3).

```
HillLab = renyi(OTULab, scale = c(0,1,2), hill = T)
names(HillLab) = c("hill1", "hill2", "hill3")
```

Hill's 1 (richness)

```
hill1.read.kit.pers.nucl =
  lmer(HillLab$hill1 ~ reads + person + kit + nuclear +
         (1|depth.nominal),
     data = ExpLab)
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```
hill1.read.kit.nucl =
  lmer(HillLab$hill1 ~ reads + kit + nuclear +
         (1|depth.nominal),
     data = ExpLab)
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

Most variation in richness was explained by the PCR success, followed by the operation period of the nuclear power plant. The extraction kits also consistently predicted variation in richness, with more OTUs recorded in the replicates extracted with the Macherey-Nagel kit. The variation explained by the lab person was minor. Communities in the lake were less species-rich following the construction of the plant.

```
anova(hill1.read.kit.pers.nucl)
```

```
## Analysis of Variance Table
##         Df  Sum Sq Mean Sq  F value
## reads    1 14834.7 14834.7 208.8352
## person   1    37.9    37.9   0.5341
## kit      1   672.6   672.6   9.4691
## nuclear  2  1758.3   879.1  12.3759
```

Considering the lab person did not

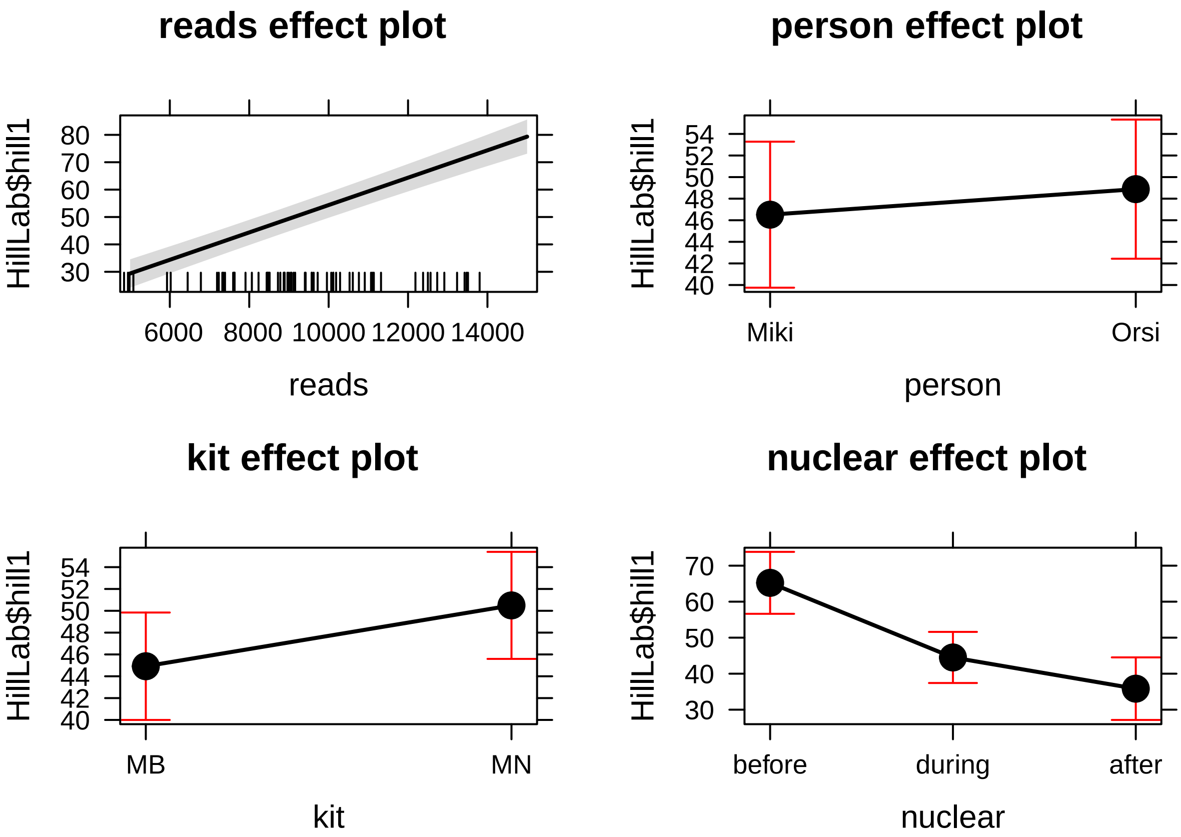
```
anova(hill1.read.kit.pers.nucl,
      hill1.read.kit.nucl)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: ExpLab
## Models:
## hill1.read.kit.nucl: HillLab$hill1 ~ reads + kit + nuclear + (1 | depth.nominal)
## hill1.read.kit.pers.nucl: HillLab$hill1 ~ reads + person + kit + nuclear + (1 | depth.nominal)
##                          Df    AIC    BIC  logLik deviance  Chisq Chi Df
## hill1.read.kit.nucl       7 635.15 652.08 -310.58   621.15
## hill1.read.kit.pers.nucl  8 636.85 656.20 -310.43   620.85 0.2999      1
##                          Pr(>Chisq)
## hill1.read.kit.nucl
## hill1.read.kit.pers.nucl     0.5839
```

The richness model was not improved by including the lab person as predictor.

```
plot(allEffects(hill1.read.kit.pers.nucl,
                multiline=TRUE, confidence.level = 0.95))
```



Hill's 2 (exp(Shannon diversity))

```
hill2.read.kit.pers.nucl =
  lmer(HillLab$hill2 ~ reads + kit + person + nuclear +
         (1|depth.nominal),
     data = ExpLab)
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```
hill2.read.kit.nucl =
  lmer(HillLab$hill2 ~ reads + kit + nuclear +
         (1|depth.nominal),
     data = ExpLab)
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

The identity of the lab person explains almost no variation in Hill's 2.

```
anova(hill2.read.kit.pers.nucl)
```

```
## Analysis of Variance Table
##         Df  Sum Sq Mean Sq F value
## reads    1 1156.28 1156.28 51.0518
## kit      1  197.95  197.95  8.7400
## person   1    0.89    0.89  0.0392
## nuclear  2  867.02  433.51 19.1402
```

The model of Hill's 2 was not improved by considering the lab person's identity.

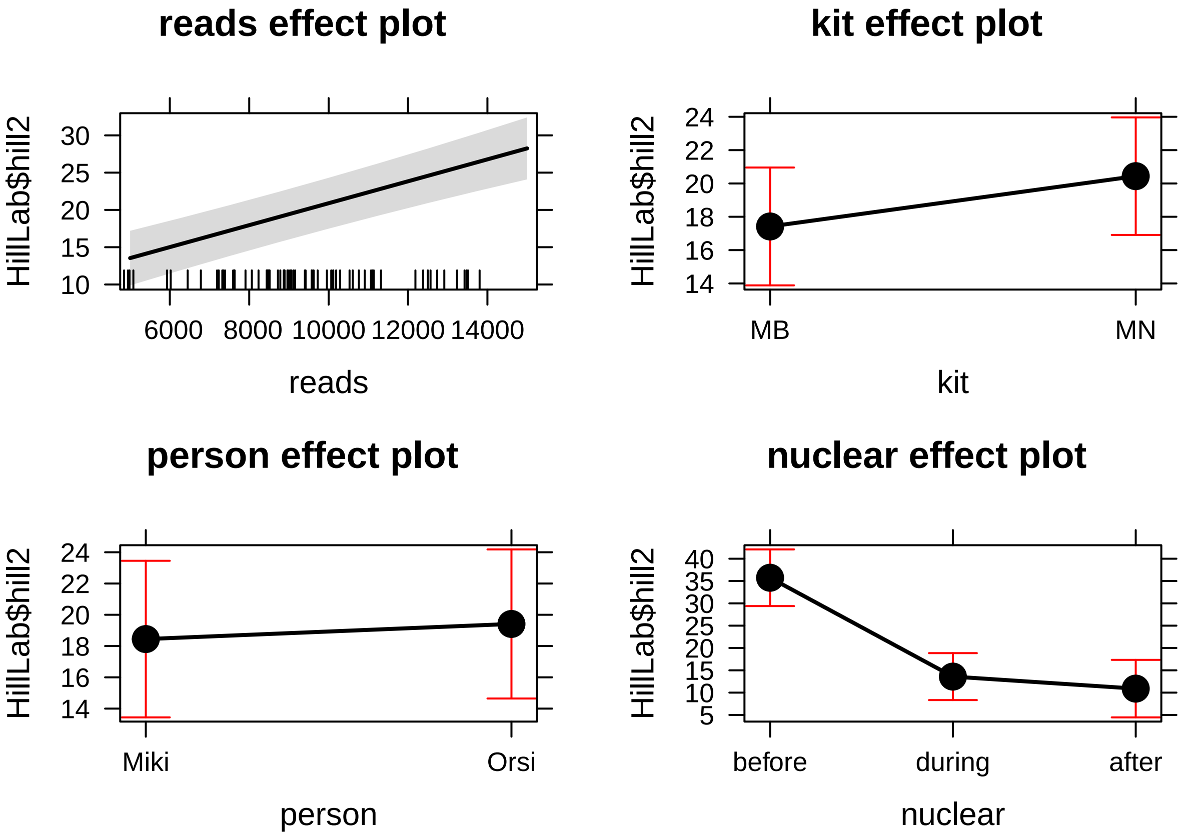
```
anova(hill2.read.kit.pers.nucl,
      hill2.read.kit.nucl)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: ExpLab
## Models:
## hill2.read.kit.nucl: HillLab$hill2 ~ reads + kit + nuclear + (1 | depth.nominal)
## hill2.read.kit.pers.nucl: HillLab$hill2 ~ reads + kit + person + nuclear + (1 | depth.nominal)
##                          Df    AIC    BIC  logLik deviance  Chisq Chi Df
## hill2.read.kit.nucl       7 551.50 568.44 -268.75   537.50
## hill2.read.kit.pers.nucl  8 553.41 572.76 -268.70   537.41 0.0944      1
##                          Pr(>Chisq)
## hill2.read.kit.nucl
## hill2.read.kit.pers.nucl     0.7587
```

Hill's 2 was strongly influenced by the PCR success. The replicates extracted with the MN-kit consistently had higher Hill's 2 values. Hill's 2 strongly decreased after the construction of the power plant.

```
plot(allEffects(hill2.read.kit.pers.nucl,
                multiline=TRUE, confidence.level = 0.95))
```



Hill's 3 (inverse Simpson)

Fit the full model.

```
hill3.read.kit.pers.nucl =
  lmer(HillLab$hill3 ~ reads + kit + person + nuclear +
         (1|depth.nominal),
     data = ExpLab)
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```
hill3.read.kit.nucl =
  lmer(HillLab$hill3 ~ reads + kit + nuclear +
         (1|depth.nominal),
     data = ExpLab)
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

The lab person explained little variation in Hill’s 3.

```
anova(hill3.read.kit.pers.nucl)
```

```
## Analysis of Variance Table
##         Df Sum Sq Mean Sq F value
## reads    1 172.34 172.339 16.2251
## kit      1  67.78  67.781  6.3813
## person   1   1.87   1.867  0.1758
## nuclear  2 435.49 217.745 20.4998
```

The lab person identity did not improve the Hill’s 2 model.

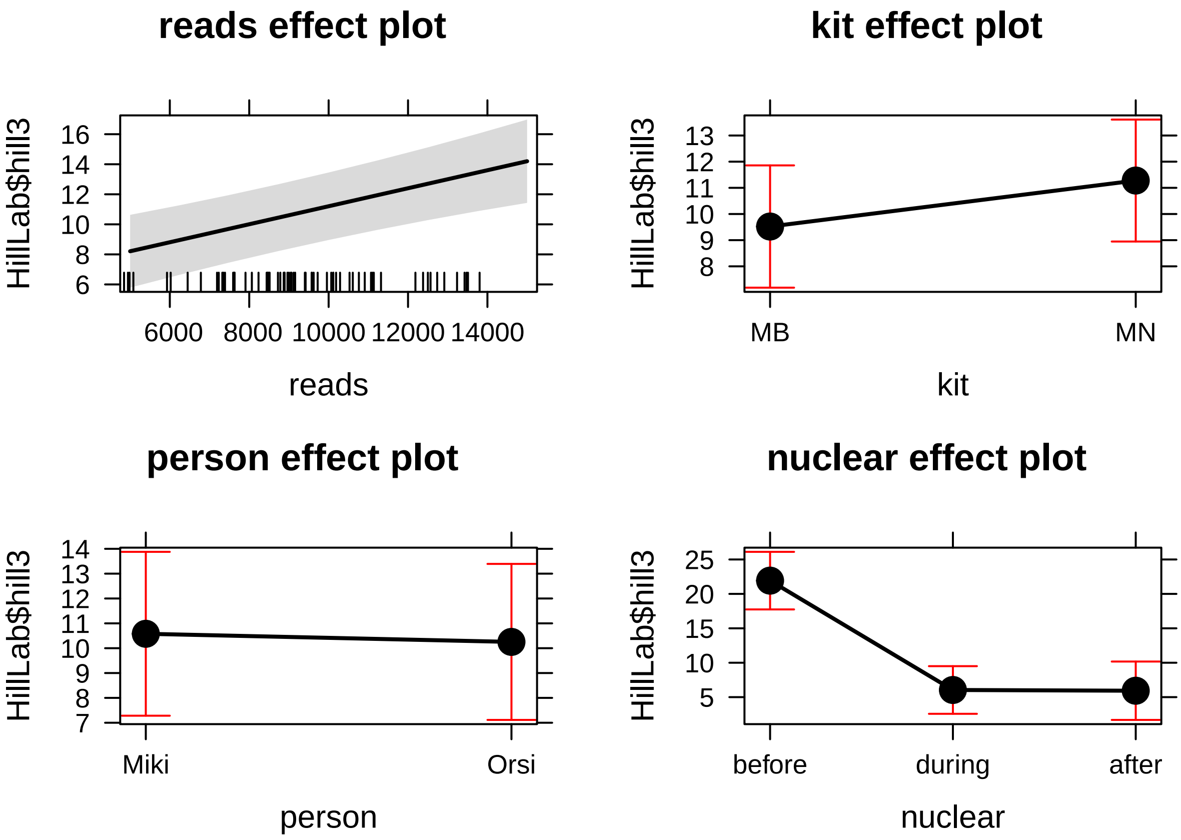
```
anova(hill3.read.kit.pers.nucl,
      hill3.read.kit.nucl)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: ExpLab
## Models:
## hill3.read.kit.nucl: HillLab$hill3 ~ reads + kit + nuclear + (1 | depth.nominal)
## hill3.read.kit.pers.nucl: HillLab$hill3 ~ reads + kit + person + nuclear + (1 | depth.nominal)
##                          Df    AIC    BIC  logLik deviance  Chisq Chi Df
## hill3.read.kit.nucl       7 486.92 503.86 -236.46   472.92
## hill3.read.kit.pers.nucl  8 488.90 508.25 -236.45   472.90 0.0225      1
##                          Pr(>Chisq)
## hill3.read.kit.nucl
## hill3.read.kit.pers.nucl     0.8807
```

The effects of the nuclear power plant explained most variation in Hill’s 3, resulting in lower diversity after. These were followed by the PCR success and the effects.

```
plot(allEffects(hill3.read.kit.pers.nucl,
                multiline=TRUE, confidence.level = 0.95))
```



Community composition

Colors for ordination

```
# colors
colfunc <- colorRampPalette(c("green", "brown"))
MyColors = colfunc(83)
```

Models of community

I need a general, community-level statistic. `mvabund` cannot deal with random effects: I ommit the horizon identity completely and treat the replicates as independent although they are not

I will compare this with `PERMANOVA` in the `adonis` with `strata` specified as horizons to consider nestedness.

Fit the community model

```
# input data
OTU.mva = mvabund(OTULab)

# full model fit
OTU.read.kit.pers.nucl =
  manyglm(OTU.mva ~ reads + kit + person + nuclear,
          data = ExpLab, family = "negative.binomial")
```

Variation partitioning and predictor stat. significance. Save the ANOVA results.

```
OTU.anova = anova(OTU.read.kit.pers.nucl,
                  nBoot = 100, test = "LR")
save(OTU.anova, file="lab-methods_OTU_anova.RData")
```

Load the saved ANOVA results

```
load("lab-methods_OTU_anova.RData")
```

Variations explained and statistical significances. Most variation is explained by the biological effect, followed by the person (although both are only marginally significant). The kit doesn't seem to influence the community composition.

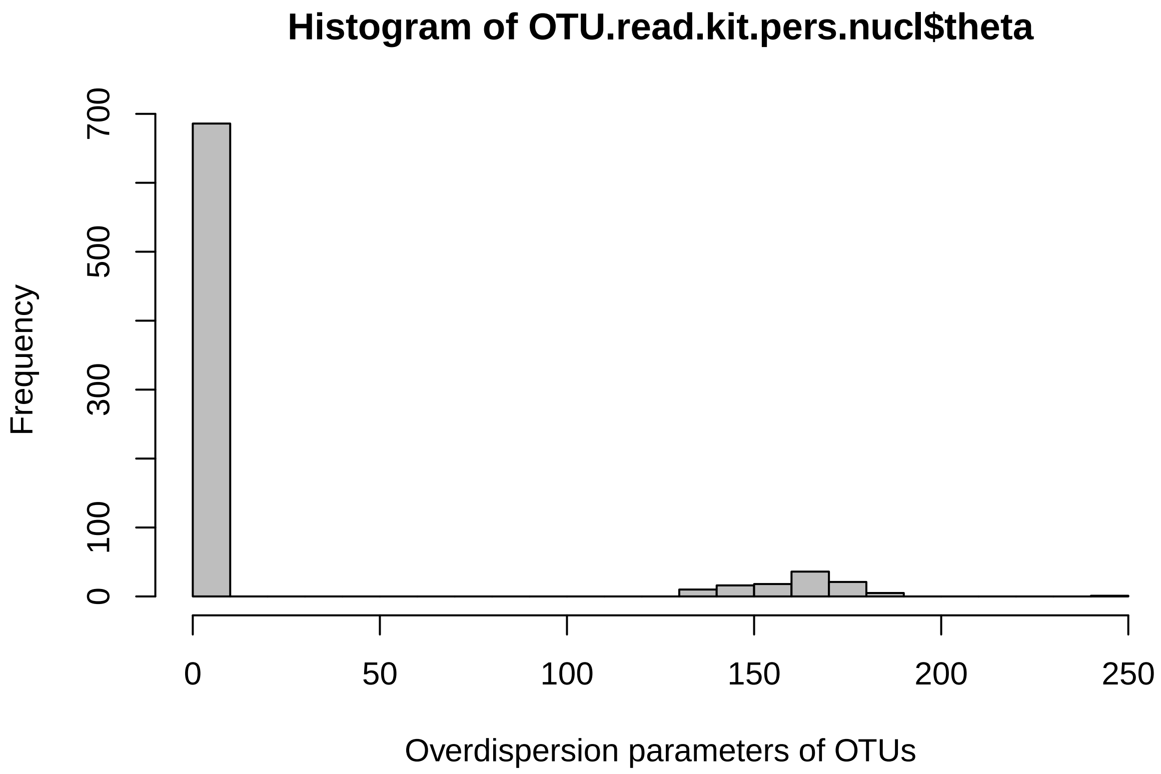
```
OTU.anova
```

```
## Analysis of Deviance Table
##
## Model: manyglm(formula = OTU.mva ~ reads + kit + person + nuclear, family = "negative.binomial",
## Model:     data = ExpLab)
##
## Multivariate test:
##             Res.Df Df.diff  Dev Pr(>Dev)
## (Intercept)     82
## reads           81       1 1163    0.020 *
## kit             80       1  906    0.733
## person          79       1 2018    0.089 .
## nuclear         77       2 5488    0.069 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Arguments:
##  Test statistics calculated assuming uncorrelated response (for faster computation)
##  P-value calculated using 100 resampling iterations via PIT-trap resampling (to account for correlat
```

Ordination with latent variables

Are there OTUs with weird overdispersion parameters? These are calculated by `mvabund`.

```
hist(OTU.read.kit.pers.nucl$theta,
     col="grey", xlab="Overdispersion parameters of OTUs", nclass=20)
```



Set the dispersion prior (the last of hypparams) according to the range of **theta** from the `mvabund` run.

```
set.prior = list(type = c("normal","normal","normal","uniform"),
                 hypparams = c(100, 20, 100, 30))
```

Fit the `boral` model only with the OTUs that are not exceedingly overdispersed. Run on the `malloy` since it takes a long time.

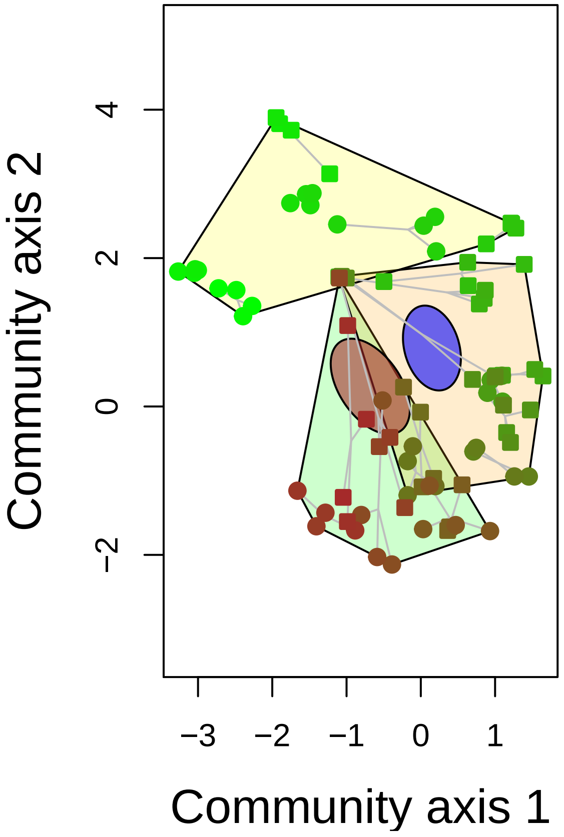
```
# LV ordination done on all OTUs with relatively low overdispersion
save.image(file="Lab_Boral_Data.RData")


LabLVMOrd = boral(OTULab[,OTU.read.kit.pers.nucl$theta < 31],
                  X = ExpLab$reads,
                  family = "negative.binomial",
                  prior.control = set.prior,
                  num.lv = 2, n.burnin = 10000,
                  n.iteration = 40000, n.thin = 30)

save(LabLVMOrd, file="Lab_LV_model_40000-iter.RData")
```

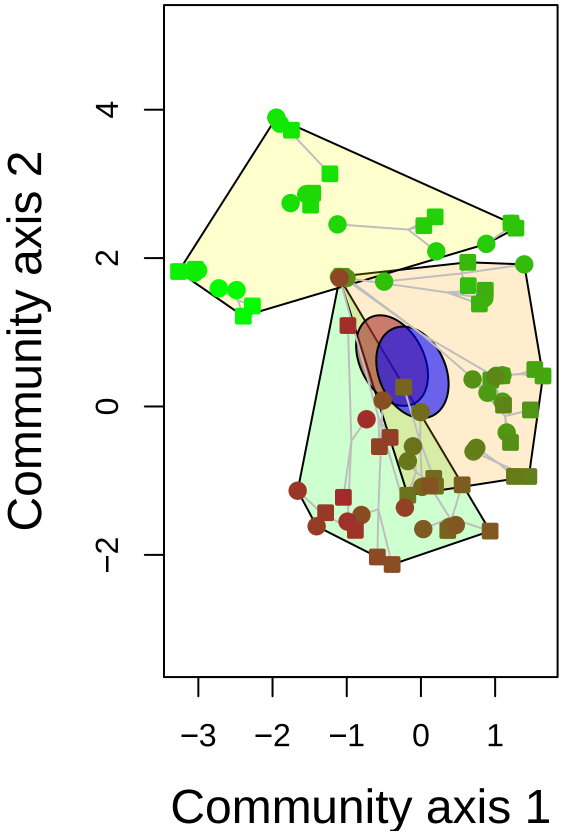
Load the `boral` results

```
load("Lab_LV_model_40000-iter.RData")
```

Plot the ordination. The replicates are colored according to the sample. Note the weird replicates those all pull toward the negative - positive controls on the complete ordination (not yet shown). The ellipses show the 95% CI group centroids of replicates from before 1960 (green), between 1960-1990 (orange), after 1990 (blue). The countours mar the sediment age.

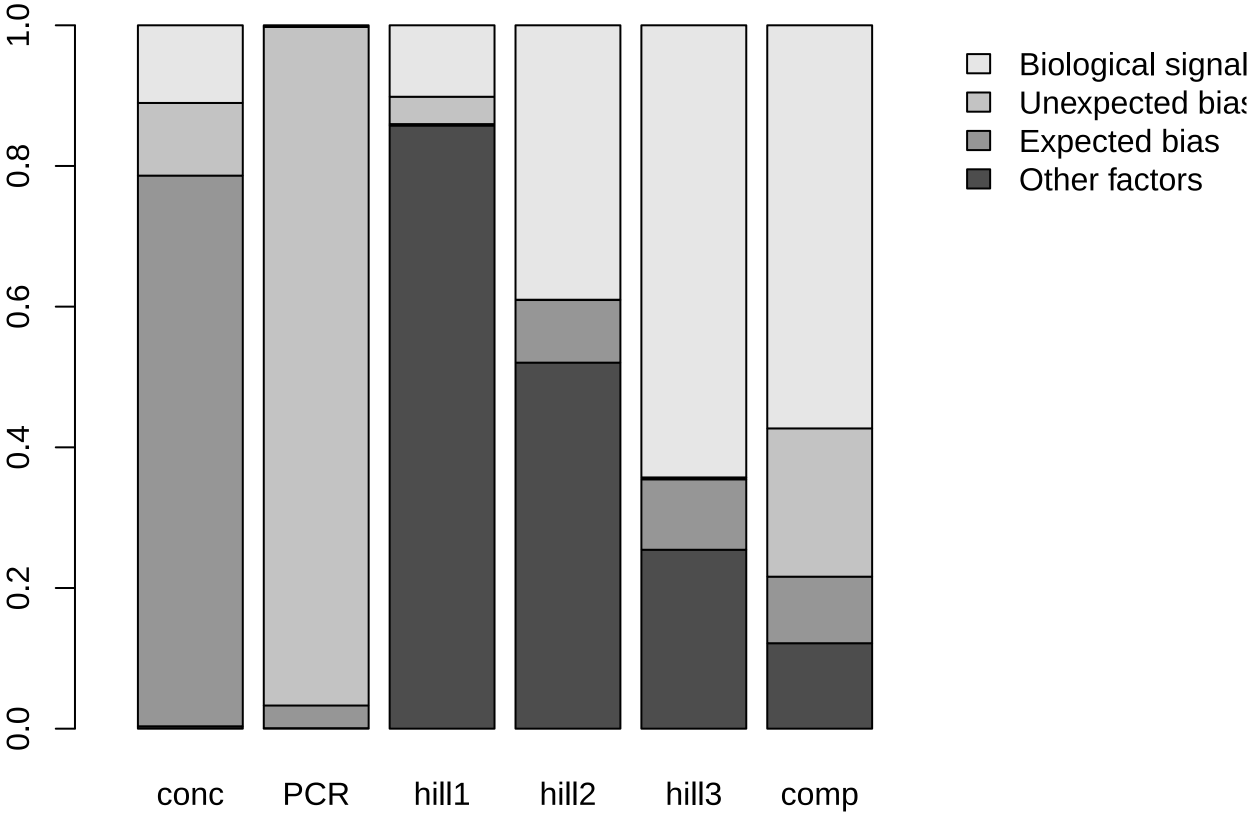
```
par(mar=c(4,5,1,1), mfrow = c(1,2))
ordicomm = ordiplot(LabLVMOrd$lv.median[order(ExpLab$depth),],
                    choices = c(1,2),
                    type = "none", cex =0.5,
                    display = "sites", xlab = "Community axis 1",
                    ylab = "Community axis 2", cex.lab = 1.5)
ordihull(ordicomm, ExpLab$nuclear[order(ExpLab$depth)],cex=.5,
            draw="polygon", col=c("green"),
            alpha=50, show.groups=(c("before")))
ordihull(ordicomm, ExpLab$nuclear[order(ExpLab$depth)],cex=.5,
            draw="polygon", col=c("orange"),
            alpha=50, show.groups=(c("during")))
ordihull(ordicomm, ExpLab$nuclear[order(ExpLab$depth)],cex=.5,
            draw="polygon", col=c("yellow"),
            alpha=50, show.groups=(c("after")))
ordiellipse(ordicomm, ExpLab$person[order(ExpLab$depth)],cex=.5,
            draw="polygon", col=c("brown"),
            alpha=150,kind="se",conf=0.95,
            show.groups=(c("Miki")))
ordiellipse(ordicomm, ExpLab$person[order(ExpLab$depth)],cex=.5,
            draw="polygon", col=c("blue"),
            alpha=150,kind="se",conf=0.95,
            show.groups=(c("Orsi")))
ordispider(ordicomm, ExpLab$depth.nominal, col="grey")
points(ordicomm, "sites", lwd=2,
       col=MyColors, bg = MyColors, pch = 20 + as.numeric(ExpLab$person[order(ExpLab$depth)]))

# Ordination plot with the kit highlighted
ordicomm = ordiplot(LabLVMOrd$lv.median[order(ExpLab$depth),],
                    choices = c(1,2),
                    type = "none", cex =0.5,
                    display = "sites", xlab = "Community axis 1",
                    ylab = "Community axis 2", cex.lab = 1.5)
ordihull(ordicomm, ExpLab$nuclear[order(ExpLab$depth)],cex=.5,
            draw="polygon", col=c("green"),
            alpha=50, show.groups=(c("before")))
ordihull(ordicomm, ExpLab$nuclear[order(ExpLab$depth)],cex=.5,
            draw="polygon", col=c("orange"),
            alpha=50, show.groups=(c("during")))
ordihull(ordicomm, ExpLab$nuclear[order(ExpLab$depth)],cex=.5,
            draw="polygon", col=c("yellow"),
            alpha=50, show.groups=(c("after")))
ordiellipse(ordicomm, ExpLab$kit[order(ExpLab$depth)],cex=.5,
            draw="polygon", col=c("brown"),
            alpha=150,kind="se",conf=0.95,
            show.groups=(c("MB")))
ordiellipse(ordicomm, ExpLab$kit[order(ExpLab$depth)],cex=.5,
            draw="polygon", col=c("blue"),
            alpha=150,kind="se",conf=0.95,
            show.groups=(c("MN")))
ordispider(ordicomm, ExpLab$depth.nominal, col="grey")
points(ordicomm, "sites", lwd=2,
       col=MyColors, bg =MyColors, pch = 20+as.numeric(ExpLab$kit[order(ExpLab$depth)]))
```





Variance partitioning graph

```
MyVariances = cbind(conc = anova(conc.weight.kit.person.age)[1:4,2],
                    PCR = anova(read.conc.kit.pers.age)[1:4,2],
                    hill1 = anova(hill1.read.kit.pers.nucl)[1:4,2],
                    hill2 = anova(hill2.read.kit.pers.nucl)[1:4,2],
                    hill3 = anova(hill3.read.kit.pers.nucl)[1:4,2],
                    comp = OTU.anova$table[2:5,3])
rownames(MyVariances) = c("weight/conc/reads", "kit", "person", "age/nucl")

MyVariancePerc = apply(MyVariances, 2, function(x){x/sum(x)})

par(mar = c(2,2,2,1))
barplot(MyVariancePerc,
        legend.text=c("Other factors", "Expected bias",
                      "Unexpected bias","Biological signal"),
        xlim=c(0, ncol(MyVariancePerc) + 4),
        args.legend=list(
          x=ncol(MyVariancePerc) + 5,
          y=max(colSums(MyVariancePerc)),
          bty = "n"))
```


1.0
0.8
0.6
0.4
0.2
0.0
conc
PCR
hill1
hill2
hill3
comp
Biological signal
Unexpected bias
Expected bias
Other factors