

# Data analysis

## Data preparation

### Prepare work environment

Install pacman

```
if (!require("pacman")) install.packages("pacman")
```

Loading required package: pacman

Load packages

```
pacman::p_load(car, ggpubr, nlme, lme4, rsq, emmeans, colorspace, lmerTest, pbkrtest, factoextra, multcomp, tidyverse, dplyr, conflicted)
```

Prioritize commands from specific packages

```
conflicts_prefer(dplyr::filter)
```

[conflicted] Will prefer dplyr::filter over any other package.

```
conflicts_prefer(dplyr::select)
```

[conflicted] Will prefer dplyr::select over any other package.

```
conflicts_prefer(dplyr::rename)
```

[conflicted] Will prefer dplyr::rename over any other package.

```
conflicts_prefer(lmerTest::lmer())
```

[conflicted] Will prefer lmerTest::lmer over any other package.

```
conflicts_prefer(stats::sd)
```

[conflicted] Will prefer stats::sd over any other package.

```
conflicts_prefer(base::unname)
```

[conflicted] Will prefer base::unname over any other package.

### Load data

Data sets with all plant and soil data from conditioning phase I, conditioning phase II and the feedback phase. The conditioning I data set includes observations for each individual plant which was grown, including biomass and root exudation rates. The conditioning II data set includes data for all pots with soil which were conditioned by plants, microbial soil inoculum, root exudates or root litter extracted at the end of conditioning phase I. It does

not include the amplicon sequencing data of the soil microbial communities. The feedback data set includes data from plants grown in these soil mixtures, including their biomass.

```
data_conditioning_I <-read.csv("Conditioning_I.csv")
data_conditioning_II <-read.csv("Conditioning_II.csv")
data_feedback <-read.csv("Feedback.csv")
```

### Change data types

Some data is not recognized in the format that we need it in, so it is manually changed. NAs are produced, where there is missing data.

```
data_conditioning_I$block <- as.factor(data_conditioning_I$block)
data_conditioning_II$block <- as.factor(data_conditioning_II$block)
data_conditioning_II$number <- as.character(data_conditioning_II$number)
data_conditioning_II$SOM <- as.numeric(data_conditioning_II$SOM)
```

Warning: NAs introduced by coercion

```
data_feedback$block <- as.factor(data_feedback$block)
data_feedback$rootshootratio <- as.numeric(data_feedback$rootshootratio)
```

Warning: NAs introduced by coercion

### Combine data sets to calculate PSF

In order to calculate PSF we need data from all experimental phases combined in one data frame. The conditioning phase I to determine the conditioning species and the drought treatment, conditioning II to determine the type of soil conditioning and the feedback phase to determine the actual biomass of the plant. Each row in this data frame represents one PSF pair of the 'home' species and the 'away' species, data of both of these plants from the conditioning phases, and the biomass of the home species in the feedback phase in either home or away soil. In that way we are able to calculate the feedback in each row.

### Fuse conditioning I and II

Remove data points of dead plants

To prevent misleading results and outliers, data points of dead plants in the feedback phase are removed. The 'note' column is deleted as it does not carry information anymore.

```
data_feedback <- data_feedback %>%
  filter(note != "dead") %>%
  dplyr::select(!note)
```

Calculate means for Conditioning I

First, we need to calculate means for each species and treatment, because each soil in Conditioning II was mixed from up to 6 plants from Conditioning I. The reason was, that the quantities of root exudates produced by one plant are not sufficient for the amount needed to add to the soil.

```
means_conditioning_I <- do.call(data.frame,aggregate(cbind(shootFW, shootDW,
SDMC, rootFW, rootDW, RDMC, totalFW, totalDW, rootshootratio, exudateC, SER)
~ species * drought * block, FUN = function(x) mean = mean(x), data =
data_conditioning_I))
```

Check resulting data frame

```
glimpse(means_conditioning_I)
```

```
Rows: 24
Columns: 14
$ species      <chr> "Ao", "Dg", "Ra", "Ao", "Dg", "Ra", "Ao", "Dg", "Ra",
"..."
$ drought      <chr> "D", "D", "D", "WW", "WW", "WW", "D", "D", "D", "WW",
"..."
$ block        <fct> 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3,
4...
$ shootFW      <dbl> 1.476150, 1.703950, 1.499167, 1.626500, 1.897167,
2.541...
$ shootDW      <dbl> 0.4034167, 0.4972333, 0.3099500, 0.5011000, 0.5696500,
...
$ SDMC         <dbl> 0.2746333, 0.2943000, 0.2090833, 0.3107667, 0.3027667,
...
$ rootFW       <dbl> 0.8554333, 1.2641833, 1.2220500, 1.4213333, 1.4095000,
...
$ rootDW       <dbl> 0.1847500, 0.2735500, 0.2722000, 0.2875167, 0.3098167,
...
$ RDMC         <dbl> 0.2167667, 0.2164000, 0.2228333, 0.2021833, 0.2193500,
...
$ totalFW      <dbl> 2.331583, 2.968133, 2.721217, 3.047833, 3.306667,
4.731...
$ totalDW      <dbl> 0.5881667, 0.7707833, 0.5821500, 0.7886167, 0.8794667,
...
$ rootshootratio <dbl> 0.4643000, 0.5625500, 0.9133500, 0.5960333, 0.5510167,
...
$ exudateC     <dbl> 230.6533, 199.7992, 232.1075, 300.5408, 131.6950,
178.6...
$ SER          <dbl> 1300.0939, 736.3746, 881.5824, 1044.9974, 432.4111,
306...
```

Fuse conditioning I and II

```
data_conditioning <- data_conditioning_II %>% left_join(means_conditioning_I,
relationship = "many-to-many")
```

Joining with `by = join\_by(block, species, drought)`

Check resulting data frame

```
glimpse(data_conditioning)
```



```

$ rootDW          <dbl> 0.2722000, 0.2735500, 0.2722000, 0.3098167, NA,
0.309...
$ RDMC            <dbl> 0.2228333, 0.2164000, 0.2228333, 0.2193500, NA,
0.219...
$ totalFW         <dbl> 2.721217, 2.968133, 2.721217, 3.306667, NA,
3.306667,...
$ totalDW         <dbl> 0.5821500, 0.7707833, 0.5821500, 0.8794667, NA,
0.879...
$ rootshootratio <dbl> 0.9133500, 0.5625500, 0.9133500, 0.5510167, NA,
0.551...
$ exudateC        <dbl> 232.1075, 199.7992, 232.1075, 131.6950, NA,
131.6950,...
$ SER             <dbl> 881.5824, 736.3746, 881.5824, 432.4111, NA,
432.4111,...

```

Save csv file

```
write.csv(data_conditioning, "data_conditioning.csv")
```

### Create data frame for PSF pairs

It as a full-factorial design in the sense that every species is paired with every other species within each treatment. 'None' is used for the case that the away soil is unconditioned background soil that did not have a plant growing in it.

```

species_home = rep(c("Ao", "Dg", "Ra"), each = 3)
species_away = c("Dg", "Ra", "none", "Ao", "Ra", "none", "Ao", "Dg", "none")

SpeciesPairs <- data.frame(species_home, species_away)
SpeciesPairs$pair <- paste(SpeciesPairs$species_home, sep = "_",
SpeciesPairs$species_away)

```

Check resulting data frame

```

glimpse(SpeciesPairs)

Rows: 9
Columns: 3
$ species_home <chr> "Ao", "Ao", "Ao", "Dg", "Dg", "Dg", "Ra", "Ra", "Ra"
$ species_away <chr> "Dg", "Ra", "none", "Ao", "Ra", "none", "Ao", "Dg",
"none"
$ pair          <chr> "Ao_Dg", "Ao_Ra", "Ao_none", "Dg_Ao", "Dg_Ra",
"Dg_none",...

```

Save csv file

```
write.csv(SpeciesPairs, "ELE_speciespairs.csv", row.names = F)
```

## Extract data of home and away species from the conditioning phase

### Home data conditioning phase

For plant variables that were also measured in the feedback phase the add-on “\_cond” was added, to discriminate between the two. For all variables measured in the home and away species the add-on “\_home” was added. In a final step the data set is fused with SpeciesPairs: Now each pair for the PSF calculations has one row in the data frame

```
conditioning_home <- data_conditioning %>%
  rename(species_home = species) %>%
  right_join(SpeciesPairs, relationship = "many-to-many") %>%
  relocate(pair, .before = block) %>%
  relocate(species_away, .after = species_home) %>%
  rename(shootFW_cond_home = shootFW, shootDW_cond_home = shootDW,
SDMC_cond_home = SDMC, rootFW_cond_home = rootFW,
        rootDW_cond_home = rootDW, RDMC_cond_home = RDMC, totalFW_cond_home
= totalFW, totalDW_cond_home =totalDW,
        rootshootratio_cond_home = rootshootratio, exudateC_home = exudateC,
SER_home = SER, moisture_home = moisture,
        SOM_home = SOM, pH_home = pH, microbialC_home = microbialC,
microbialN_home = microbialN,
        microbialCNratio_home = microbialCNratio, availableN_home =
availableN, availableNO3_home = availableNO3,
        availableNH4_home = availableNH4, availableP_home = availableP,
DOC_home = DOC, TDN_home = TDN, TDC_home = TDC ) %>%
  select(!number & !soil_ID & !species_full)

Joining with `by = join_by(species_home)`
```

### Away data conditioning phase

Similarly, here the add-ons “\_away” and “\_home” were added to discriminate between data measured in different phases and on different plants.

```
conditioning_away <- data_conditioning %>%
  dplyr::rename(species_away = species) %>%
  left_join(SpeciesPairs, relationship = "many-to-many", unmatched = "drop")
%>%
  dplyr::rename(shootFW_cond_away = shootFW, shootDW_cond_away = shootDW,
SDMC_cond_away = SDMC, rootFW_cond_away = rootFW,
        rootDW_cond_away = rootDW, RDMC_cond_away = RDMC, totalFW_cond_away
= totalFW, totalDW_cond_away =totalDW,
        rootshootratio_cond_away = rootshootratio, exudateC_away = exudateC,
SER_away = SER, moisture_away = moisture,
        SOM_away = SOM, pH_away = pH, microbialC_away = microbialC,
microbialN_away = microbialN,
        microbialCNratio_away = microbialCNratio, availableN_away =
availableN, availableNO3_away = availableNO3,
        availableNH4_away = availableNH4, availableP_away = availableP,
DOC_away = DOC, TDN_away = TDN, TDC_away = TDC ) %>%
```

```
relocate(pair, .before = block) %>%
relocate(species_home, .after = drought) %>%
relocate(species_away, .after = species_home) %>%
dplyr::select(!number & !soil_ID & !species_full)
```

Joining with `by = join\_by(species\_away)`

### Extract data from the feedback phase

Filter by species\_home == species\_away. In this way only plants growing in home soil are selected.

```
feedback_home <- data_feedback %>%
  filter(species_home == species_soil) %>%
  select(!species_soil & !soil_ID) %>%
  rename(height_home = height, leaves_home = leaves, shootDW_home = shootDW,
         rootDW_home = rootDW,
         totalDW_home = totalDW, rootshootratio_home = rootshootratio)
```

Filter by species\_home != species\_away. Now only plants growing in away soil are selected

```
feedback_away <- data_feedback %>%
  filter(species_home != species_soil) %>%
  select(!soil_ID) %>%
  rename(species_away = species_soil, height_away = height, leaves_away =
         leaves, shootDW_away = shootDW,
         rootDW_away = rootDW, totalDW_away = totalDW, rootshootratio_away =
         rootshootratio)
```

### Get around merging issue with unconditioned soil

If this is not done, the data frames won't merge properly. We also want to calculate PSF for home soil compared to unconditioned background soil, additional to the comparison home soil vs. away soil. Because we want to do that with all the different treatments, such as drought and soil conditioning type, we have to make sure for each observation in home soil there is a compatible observation in the unconditioned "none" soil with the same combination of treatments.

Mock data frame with all drought and input combinations

```
treatments <- data.frame(drought = rep(c("WW", "D"), each = 4), input =
rep(c("plant", "inoculum", "exudates", "rootlitter"), times = 2))
```

Filter conditioning\_away by away\_species & merge with treatments

```
conditioning_away_none <- conditioning_away %>%
  filter(species_away == "none") %>%
  select(!input & !drought) %>%
  merge(treatments)
```

Replace existing "none" lines in conditioning\_away

```
conditioning_away <- conditioning_away %>%  
  filter(species_away != "none") %>%  
  rbind(conditioning_away_none)
```

Repeat for feedback\_away

```
feedback_away_none <- feedback_away %>%  
  filter(species_away == "none") %>%  
  select(!input & !drought) %>%  
  merge(treatments)
```

```
feedback_away <- feedback_away %>%  
  filter(species_away != "none") %>%  
  rbind(feedback_away_none)
```

### Merge all data frames

In a first step the conditioning\_home is merged with feedback\_home. In a second step the conditioning\_away is added In a third step feedback\_away is added

```
data_PSF <- merge(conditioning_home, feedback_home, all = T) %>%  
  merge(conditioning_away, all = T) %>%  
  merge(feedback_away, all = T)%>%  
  relocate(c(pair, species_home, species_away), .after = block)
```

Check resulting data frame

Now in every row we have data for each species pair and treatment for both the home and away species in the conditioning phase ('\_cond\_home') and away species in the conditioning phase ('\_cond\_away') and for the home species in the feedback phase growing either in home ('\_home') soil or in away soil ('\_away'). Additionally we have the soil data for home soil ('\_home') and away soil ('\_away').

```
glimpse(data_PSF)
```

```
Rows: 288  
Columns: 66  
$ block          <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
...  
$ pair          <chr> "Ao_Dg", "Ao_none", "Ao_Ra", "Ao_Dg",  
"Ao_non...  
$ species_home  <chr> "Ao", "Ao", "Ao", "Ao", "Ao", "Ao", "Ao",  
"Ao...  
$ species_away  <chr> "Dg", "none", "Ra", "Dg", "none", "Ra",  
"Dg",...  
$ drought       <chr> "D", "D", "D", "D", "D", "D", "D", "D", "D",  
...  
$ input         <chr> "exudates", "exudates", "exudates",  
"inoculum...
```

```

$ moisture_home <dbl> 15.3, 15.3, 15.3, 16.2, 16.2, 16.2, 13.8,
13.3...
$ SOM_home <dbl> 2.866345, 2.866345, 2.866345, 2.821911,
2.821...
$ pH_home <dbl> 7.96, 7.96, 7.96, 7.97, 7.97, 7.97, 8.06,
8.0...
$ microbialC_home <dbl> 249.6386, 249.6386, 249.6386, 235.1052,
235.1...
$ microbialN_home <dbl> 29.45645, 29.45645, 29.45645, 21.01144,
21.01...
$ microbialCNratio_home <dbl> 8.474836, 8.474836, 8.474836, 11.189390,
11.1...
$ availableN_home <dbl> 6.760183, 6.760183, 6.760183, 5.759557,
5.759...
$ availableNO3_home <dbl> 6.669846, 6.669846, 6.669846, 5.759557,
5.759...
$ availableNH4_home <dbl> 0.090336521, 0.090336521, 0.090336521,
0.0000...
$ availableP_home <dbl> 0.2894757, 0.2894757, 0.2894757, 0.2597379,
0...
$ TDN_home <dbl> 5.411, 5.411, 5.411, 5.360, 5.360, 5.360,
5.2...
$ DOC_home <dbl> 12.69744, 12.69744, 12.69744, 13.87048,
13.87...
$ TDC_home <dbl> 26.35012, 26.35012, 26.35012, 28.14826,
28.14...
$ shootFW_cond_home <dbl> 1.47615, 1.47615, 1.47615, 1.47615, 1.47615,
...
$ shootDW_cond_home <dbl> 0.4034167, 0.4034167, 0.4034167, 0.4034167,
0...
$ SDMC_cond_home <dbl> 0.2746333, 0.2746333, 0.2746333, 0.2746333,
0...
$ rootFW_cond_home <dbl> 0.8554333, 0.8554333, 0.8554333, 0.8554333,
0...
$ rootDW_cond_home <dbl> 0.1847500, 0.1847500, 0.1847500, 0.1847500,
0...
$ RDMC_cond_home <dbl> 0.2167667, 0.2167667, 0.2167667, 0.2167667,
0...
$ totalFW_cond_home <dbl> 2.331583, 2.331583, 2.331583, 2.331583,
2.331...
$ totalDW_cond_home <dbl> 0.5881667, 0.5881667, 0.5881667, 0.5881667,
0...
$ rootshootratio_cond_home <dbl> 0.4643000, 0.4643000, 0.4643000, 0.4643000,
0...
$ exudateC_home <dbl> 230.6533, 230.6533, 230.6533, 230.6533,
230.6...
$ SER_home <dbl> 1300.0939, 1300.0939, 1300.0939, 1300.0939,
1...
$ leaves_home <int> 12, 12, 12, 14, 14, 14, 8, 8, 8, 11, 11, 11,
...

```

```

$ height_home <dbl> 22.0, 22.0, 22.0, 20.0, 20.0, 20.0, 19.0,
19...
$ shootDW_home <dbl> 0.1452, 0.1452, 0.1452, 0.1929, 0.1929,
0.192...
$ rootDW_home <dbl> 0.1243, 0.1243, 0.1243, 0.1098, 0.1098,
0.109...
$ totalDW_home <dbl> 0.2695, 0.2695, 0.2695, 0.3027, 0.3027,
0.302...
$ rootshootratio_home <dbl> 0.8560606, 0.8560606, 0.8560606, 0.5692068,
0...
$ moisture_away <dbl> 15.5, 18.0, 19.3, 16.1, 18.0, 15.6, 19.4,
18...
$ SOM_away <dbl> 2.494340, 3.083486, 2.899482, 3.005371,
3.083...
$ pH_away <dbl> 8.04, 8.01, 8.00, 7.98, 8.01, 7.89, 8.01,
8.0...
$ microbialC_away <dbl> 255.3771, 254.7192, 242.1333, 273.5501,
254.7...
$ microbialN_away <dbl> 31.57751, 26.77236, 25.16466, 29.42614,
26.77...
$ microbialCNratio_away <dbl> 8.087310, 9.514260, 9.621957, 9.296161,
9.514...
$ availableN_away <dbl> 4.915364, 6.828679, 8.622422, 6.270168,
6.828...
$ availableNO3_away <dbl> 4.901675, 6.758800, 8.504183, 6.202783,
6.758...
$ availableNH4_away <dbl> 0.01368958, 0.06987897, 0.11823910,
0.0673853...
$ availableP_away <dbl> 0.2168935, 0.3755595, 0.4005551, 0.2619683,
0...
$ TDN_away <dbl> 4.988, 5.267, 5.983, 5.367, 5.267, 4.894,
5.6...
$ DOC_away <dbl> 13.47555, 13.87116, 12.43817, 13.42532,
13.87...
$ TDC_away <dbl> 27.13517, 30.03128, 30.92572, 27.04363,
30.03...
$ shootFW_cond_away <dbl> 1.703950, NA, 1.499167, 1.703950, NA,
1.49916...
$ shootDW_cond_away <dbl> 0.4972333, NA, 0.3099500, 0.4972333, NA,
0.30...
$ SDMC_cond_away <dbl> 0.2943000, NA, 0.2090833, 0.2943000, NA,
0.20...
$ rootFW_cond_away <dbl> 1.2641833, NA, 1.2220500, 1.2641833, NA,
1.22...
$ rootDW_cond_away <dbl> 0.2735500, NA, 0.2722000, 0.2735500, NA,
0.27...
$ RDMC_cond_away <dbl> 0.2164000, NA, 0.2228333, 0.2164000, NA,
0.22...
$ totalFW_cond_away <dbl> 2.968133, NA, 2.721217, 2.968133, NA,
2.72121...

```

```

$ totalDW_cond_away      <dbl> 0.7707833, NA, 0.5821500, 0.7707833, NA,
0.58...
$ rootshootratio_cond_away <dbl> 0.5625500, NA, 0.9133500, 0.5625500, NA,
0.91...
$ exudateC_away          <dbl> 199.7992, NA, 232.1075, 199.7992, NA,
232.107...
$ SER_away               <dbl> 736.3746, NA, 881.5824, 736.3746, NA,
881.582...
$ leaves_away            <int> 14, 16, 13, 17, 16, 6, 13, 16, 5, 16, 16,
13,...
$ height_away            <dbl> 19.0, 18.0, 14.5, 15.0, 18.0, 26.0, 18.0,
18....
$ shootDW_away           <dbl> 0.1603, 0.1331, 0.1488, 0.1706, 0.1331,
0.181...
$ rootDW_away            <dbl> 0.1646, 0.1486, 0.1397, 0.1585, 0.1486,
0.095...
$ totalDW_away           <dbl> 0.3249, 0.2817, 0.2885, 0.3291, 0.2817,
0.276...
$ rootshootratio_away    <dbl> 1.0268247, 1.1164538, 0.9388441, 0.9290739,
1...

```

### Calculate plant-soil feedbacks

This is done for total biomass, but also separately for root and shoot biomass

```

data_PSF$PSFtotal <- log(data_PSF$totalDW_home/data_PSF$totalDW_away)
data_PSF$PSFshoot <- log(data_PSF$shootDW_home/data_PSF$shootDW_away)
data_PSF$PSFroot <- log(data_PSF$rootDW_home/data_PSF$rootDW_away)

```

Add column that specifies feedback type

```

data_PSF$PSF_type <- if_else(data_PSF$species_away == "none",
"unconditioned", "away")

```

Sort variables

```

data_PSF$input <- factor(data_PSF$input, c ("plant", "inoculum",
"exudates","rootlitter"))
data_PSF$drought <- factor(data_PSF$drought, c ("WW", "D", "none"))
data_PSF$species_home <- factor(data_PSF$species_home, c ("Ao", "Dg", "Ra"))
data_PSF$species_away <- factor(data_PSF$species_away, c ("Ao", "Dg", "Ra",
"none"))

```

Check resulting data frame

```

glimpse(data_PSF)
Rows: 288
Columns: 70
$ block      <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
...
$ pair       <chr> "Ao_Dg", "Ao_none", "Ao_Ra", "Ao_Dg",

```

```

"Ao_non...
$ species_home      <fct> Ao, Ao, Ao, Ao, Ao, Ao, Ao, Ao, Ao, Ao, Ao,
A...
$ species_away      <fct> Dg, none, Ra, Dg, none, Ra, Dg, none, Ra,
Dg,...
$ drought           <fct> D, D, D, D, D, D, D, D, D, D, D, D, WW, WW,
W...
$ input             <fct> exudates, exudates, exudates, inoculum,
inocu...
$ moisture_home     <dbl> 15.3, 15.3, 15.3, 16.2, 16.2, 16.2, 13.8,
13....
$ SOM_home          <dbl> 2.866345, 2.866345, 2.866345, 2.821911,
2.821...
$ pH_home           <dbl> 7.96, 7.96, 7.96, 7.97, 7.97, 7.97, 8.06,
8.0...
$ microbialC_home   <dbl> 249.6386, 249.6386, 249.6386, 235.1052,
235.1...
$ microbialN_home   <dbl> 29.45645, 29.45645, 29.45645, 21.01144,
21.01...
$ microbialCNratio_home <dbl> 8.474836, 8.474836, 8.474836, 11.189390,
11.1...
$ availableN_home   <dbl> 6.760183, 6.760183, 6.760183, 5.759557,
5.759...
$ availableNO3_home <dbl> 6.669846, 6.669846, 6.669846, 5.759557,
5.759...
$ availableNH4_home <dbl> 0.090336521, 0.090336521, 0.090336521,
0.0000...
$ availableP_home   <dbl> 0.2894757, 0.2894757, 0.2894757, 0.2597379,
0...
$ TDN_home          <dbl> 5.411, 5.411, 5.411, 5.360, 5.360, 5.360,
5.2...
$ DOC_home          <dbl> 12.69744, 12.69744, 12.69744, 13.87048,
13.87...
$ TDC_home          <dbl> 26.35012, 26.35012, 26.35012, 28.14826,
28.14...
$ shootFW_cond_home <dbl> 1.47615, 1.47615, 1.47615, 1.47615, 1.47615,
...
$ shootDW_cond_home <dbl> 0.4034167, 0.4034167, 0.4034167, 0.4034167,
0...
$ SDMC_cond_home    <dbl> 0.2746333, 0.2746333, 0.2746333, 0.2746333,
0...
$ rootFW_cond_home  <dbl> 0.8554333, 0.8554333, 0.8554333, 0.8554333,
0...
$ rootDW_cond_home  <dbl> 0.1847500, 0.1847500, 0.1847500, 0.1847500,
0...
$ RDMC_cond_home    <dbl> 0.2167667, 0.2167667, 0.2167667, 0.2167667,
0...
$ totalFW_cond_home <dbl> 2.331583, 2.331583, 2.331583, 2.331583,
2.331...
$ totalDW_cond_home <dbl> 0.5881667, 0.5881667, 0.5881667, 0.5881667,

```

```

0...
$ rootshootratio_cond_home <dbl> 0.4643000, 0.4643000, 0.4643000, 0.4643000,
0...
$ exudateC_home <dbl> 230.6533, 230.6533, 230.6533, 230.6533,
230.6...
$ SER_home <dbl> 1300.0939, 1300.0939, 1300.0939, 1300.0939,
1...
$ leaves_home <int> 12, 12, 12, 14, 14, 14, 8, 8, 8, 11, 11, 11,
...
$ height_home <dbl> 22.0, 22.0, 22.0, 20.0, 20.0, 20.0, 19.0,
19...
$ shootDW_home <dbl> 0.1452, 0.1452, 0.1452, 0.1929, 0.1929,
0.192...
$ rootDW_home <dbl> 0.1243, 0.1243, 0.1243, 0.1098, 0.1098,
0.109...
$ totalDW_home <dbl> 0.2695, 0.2695, 0.2695, 0.3027, 0.3027,
0.302...
$ rootshootratio_home <dbl> 0.8560606, 0.8560606, 0.8560606, 0.5692068,
0...
$ moisture_away <dbl> 15.5, 18.0, 19.3, 16.1, 18.0, 15.6, 19.4,
18...
$ SOM_away <dbl> 2.494340, 3.083486, 2.899482, 3.005371,
3.083...
$ pH_away <dbl> 8.04, 8.01, 8.00, 7.98, 8.01, 7.89, 8.01,
8.0...
$ microbialC_away <dbl> 255.3771, 254.7192, 242.1333, 273.5501,
254.7...
$ microbialN_away <dbl> 31.57751, 26.77236, 25.16466, 29.42614,
26.77...
$ microbialCNratio_away <dbl> 8.087310, 9.514260, 9.621957, 9.296161,
9.514...
$ availableN_away <dbl> 4.915364, 6.828679, 8.622422, 6.270168,
6.828...
$ availableNO3_away <dbl> 4.901675, 6.758800, 8.504183, 6.202783,
6.758...
$ availableNH4_away <dbl> 0.01368958, 0.06987897, 0.11823910,
0.0673853...
$ availableP_away <dbl> 0.2168935, 0.3755595, 0.4005551, 0.2619683,
0...
$ TDN_away <dbl> 4.988, 5.267, 5.983, 5.367, 5.267, 4.894,
5.6...
$ DOC_away <dbl> 13.47555, 13.87116, 12.43817, 13.42532,
13.87...
$ TDC_away <dbl> 27.13517, 30.03128, 30.92572, 27.04363,
30.03...
$ shootFW_cond_away <dbl> 1.703950, NA, 1.499167, 1.703950, NA,
1.49916...
$ shootDW_cond_away <dbl> 0.4972333, NA, 0.3099500, 0.4972333, NA,
0.30...
$ SDMC_cond_away <dbl> 0.2943000, NA, 0.2090833, 0.2943000, NA,

```

```

0.20...
$ rootFW_cond_away      <dbl> 1.2641833, NA, 1.2220500, 1.2641833, NA,
1.22...
$ rootDW_cond_away      <dbl> 0.2735500, NA, 0.2722000, 0.2735500, NA,
0.27...
$ RDMC_cond_away        <dbl> 0.2164000, NA, 0.2228333, 0.2164000, NA,
0.22...
$ totalFW_cond_away     <dbl> 2.968133, NA, 2.721217, 2.968133, NA,
2.72121...
$ totalDW_cond_away     <dbl> 0.7707833, NA, 0.5821500, 0.7707833, NA,
0.58...
$ rootshootratio_cond_away <dbl> 0.5625500, NA, 0.9133500, 0.5625500, NA,
0.91...
$ exudateC_away         <dbl> 199.7992, NA, 232.1075, 199.7992, NA,
232.107...
$ SER_away              <dbl> 736.3746, NA, 881.5824, 736.3746, NA,
881.582...
$ leaves_away           <int> 14, 16, 13, 17, 16, 6, 13, 16, 5, 16, 16,
13,...
$ height_away           <dbl> 19.0, 18.0, 14.5, 15.0, 18.0, 26.0, 18.0,
18....
$ shootDW_away          <dbl> 0.1603, 0.1331, 0.1488, 0.1706, 0.1331,
0.181...
$ rootDW_away           <dbl> 0.1646, 0.1486, 0.1397, 0.1585, 0.1486,
0.095...
$ totalDW_away          <dbl> 0.3249, 0.2817, 0.2885, 0.3291, 0.2817,
0.276...
$ rootshootratio_away   <dbl> 1.0268247, 1.1164538, 0.9388441, 0.9290739,
1...
$ PSFtotal              <dbl> -0.18694905, -0.04427428, -0.06812670, -
0.083...
$ PSFshoot              <dbl> -0.09893496, 0.08701138, -0.02449102,
0.12285...
$ PSFroot               <dbl> -0.28082029, -0.17856013, -0.11679927, -
0.367...
$ PSF_type              <chr> "away", "unconditioned", "away", "away",
"unc...

```

Save csv file

```
write.csv(data_PSF, "PSF.csv", row.names = F)
```

## Determine functions and visuals

### Statistical functions

Standard error

```
se <- function(x)sd(x) / sqrt(length(x))
```

## Confidence intervals

qr is calculating the quantiles function for a student t distribution, this is better for small sample sizes because it does not depend on the estimated standard deviation

```
conint <- function(x)qt(0.975, df = length(x)-1) * se(x)
```

## Function to specify the number of decimals

```
specify_decimal <- function(x, k) trimws(format(round(x, k), nsmall=k))
```

## Sort variables

### Conditioning phases

```
data_conditioning$input <- factor(data_conditioning$input, c ("plant",  
"inoculum", "exudates","rootlitter", "none"))  
data_conditioning$species <- factor(data_conditioning$species, c ("Ao", "Dg",  
"Ra", "none"))  
data_conditioning$drought <- factor(data_conditioning$drought, c ("WW", "D",  
"none"))
```

### Feedback phase

```
data_feedback$input <- factor(data_feedback$input, c ("plant", "inoculum",  
"exudates","rootlitter", "none"))  
data_feedback$species_home <- factor(data_feedback$species_home, c ("Ao",  
"Dg","Ra"))  
data_feedback$species_soil <- factor(data_feedback$species_soil, c ("Ao",  
"Dg","Ra", "none"))  
data_feedback$drought <- factor(data_feedback$drought, c ("WW", "D", "none"))
```

## Visuals

### Determine colour palettes

```
DroughtPalette <- c("#3C5941", "white", "grey60")  
SpeciesPalette <- c("#009E73", "#F0E442", "#0072B2", "grey60")  
InputPalette <- c(hcl.colors(4, palette = "Fall"), "grey60")
```

### Determine shape palettes

```
Shape_SpeciesDrought <- c(16, 1, 15, 0, 17, 2, 9)
```

### Determine ggplot themes

```
biomasstHEME <- theme(legend.title = element_text(size = 13),  
legend.text = element_text(size=11),  
axis.title = element_text(size = 13),  
axis.text = element_text(face="italic", size=11),  
axis.line = element_line(linewidth = 0),  
legend.position = "bottom")
```

```

soiltheme <- theme(legend.title = element_text(size = 13),
  legend.text = element_text(size=11),
  axis.title = element_text(size = 12),
  axis.text = element_text(size=11),
  axis.line = element_line(linewidth = 0),
  strip.text.x = element_text(size = 10),
  strip.text.y = element_text(size = 10),
  strip.background = element_rect(fill="#FFFFFF", color =
"black", linewidth = 0.5))

PSFtheme <- theme_bw() + theme(legend.title = element_text(size = 13),
  legend.text = element_text(size=11),
  axis.title = element_text(size = 12),
  axis.text = element_text(size=11),
  axis.line = element_line(),
  legend.position = "bottom",
  strip.text.x = element_text(size = 13),
  strip.text.y = element_text(size = 10),
  strip.background = element_rect(fill="white", color =
"black", linewidth = 0.5))

```

Determine facet labels

```

facets_input <- as_labeller(c(`plant` = "plant", `inoculum` =
"inoculum", `exudates` = "root exudates", `rootlitter` = "root litter", `none`
= "unconditioned"))
facets_species <- as_labeller(c(`Ao` = "Anthoxanthum odoratum", `Dg` =
"Dactylis glomerata", `Ra` = "Rumex acetosa"))

```

## Conditioning phase I

This part of the data analysis and visualization contains all plant biomass and root exudation data from the end of conditioning phase I.

Remove dead plants and plants with anomalies

```

data_conditioning_I <- data_conditioning_I %>%
  filter(note == "") %>%
  dplyr::select(!note)

```

## Data visualization

### Plant biomass

Here, we create a figure of the total plant biomass, shoot biomass and root biomass dependent on the drought treatment and species.

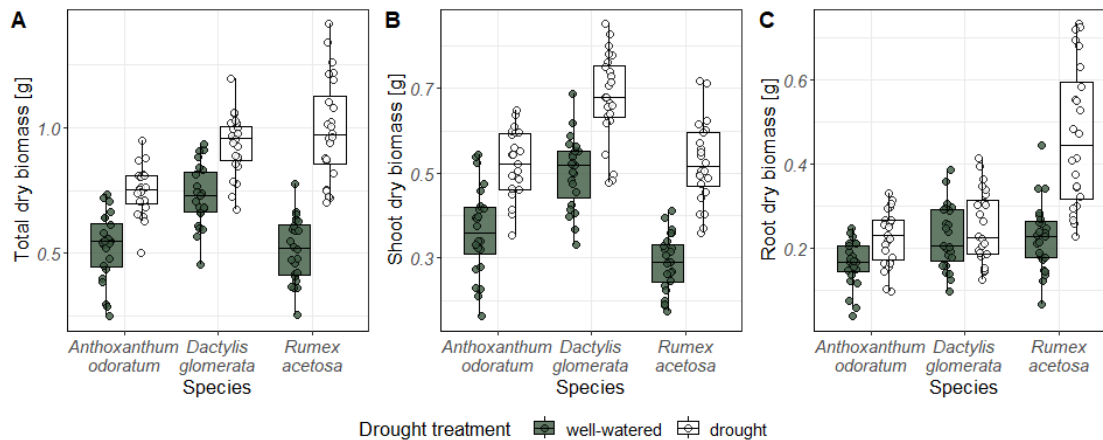
```
P_totalDW <- ggplot(data_conditioning_I, mapping = aes(x = species, y =
totalDW, fill = drought)) +
  geom_boxplot(alpha = 0.8, size = 0, outlier.shape = NA)+
  geom_point(col = "black", shape = 21, alpha = 0.8,
position=position_jitterdodge(0.1), size = 2)+
  geom_boxplot(col = "black", alpha = 0, size = 0.5, outlier.shape = NA,
fatten = 1)+
  scale_fill_manual(values = DroughtPalette, name = "Drought treatment",
labels = c("well-watered", "drought", "none"))+
  labs(x = "Species", y = "Total dry biomass [g]")+
  theme_bw()+
  biomasstheme+
  scale_x_discrete(labels = c("Anthoxanthum\nodoratum",
"Dactylis\nglomerata", "Rumex\nacetosa"))
```

```
P_shootDW <- ggplot(data_conditioning_I, mapping = aes(x = species, y =
shootDW, fill = drought)) +
  geom_boxplot(alpha = 0.8, size = 0, outlier.shape = NA)+
  geom_point(col = "black", shape = 21, alpha = 0.8,
position=position_jitterdodge(0.1), size = 2)+
  geom_boxplot(col = "black", alpha = 0, size = 0.5, outlier.shape = NA,
fatten = 1)+
  scale_fill_manual(values = DroughtPalette, name = "Drought treatment",
labels = c("well-watered", "drought", "none"))+
  labs(x = "Species", y = "Shoot dry biomass [g]")+
  theme_bw()+
  biomasstheme+
  scale_x_discrete(labels = c("Anthoxanthum\nodoratum",
"Dactylis\nglomerata", "Rumex\nacetosa"))
```

```
P_rootDW <- ggplot(data_conditioning_I, mapping = aes(x = species, y =
rootDW, fill = drought)) +
  geom_boxplot(alpha = 0.8, size = 0, outlier.shape = NA)+
  geom_point(col = "black", shape = 21, alpha = 0.8,
position=position_jitterdodge(0.1), size = 2)+
  geom_boxplot(col = "black", alpha = 0, size = 0.5, outlier.shape = NA,
fatten = 1)+
  scale_fill_manual(values = DroughtPalette, name = "Drought treatment",
labels = c("well-watered", "drought", "none"))+
  labs(x = "Species", y = "Root dry biomass [g]")+
  theme_bw()+
  biomasstheme+
  scale_x_discrete(labels = c("Anthoxanthum\nodoratum",
```

```
"Dactylis\nglomerata", "Rumex\nacetosa"))
```

```
P_DW_combined <- ggarrange(P_totalDW, P_shootDW, P_rootDW, nrow = 1,
common.legend = T, legend = "bottom", labels = "AUTO")
P_DW_combined
```



Save figure

```
ggsave(plot = P_DW_combined, "plantDW_combined.tiff", width = 30, height =
12, units = "cm", dpi = 1000)
```

## Root exudation

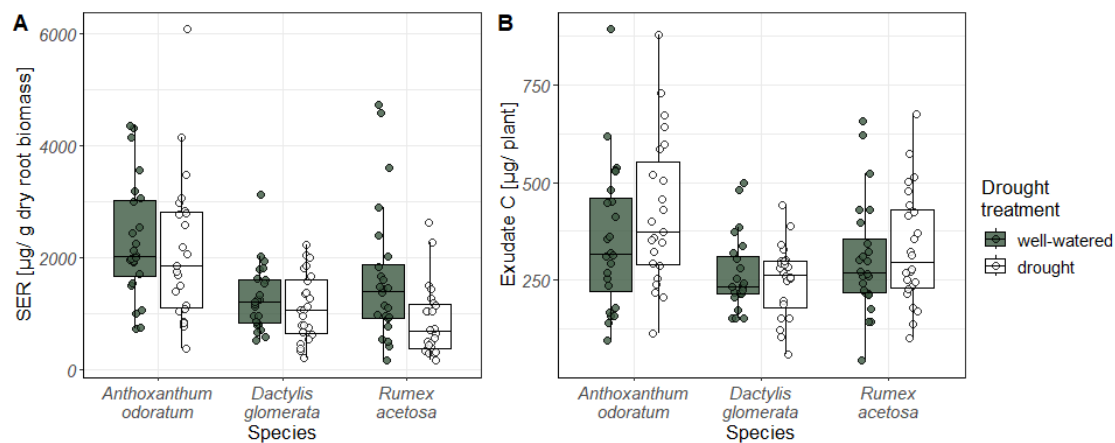
Here, we create a figure of the specific exudation rate (SER) and the total exudate C per plant (exudate C) dependent on the drought treatment and species.

```
P_SER <- ggplot(data_conditioning_I, mapping = aes(x = species, y = SER, fill
= drought)) +
  geom_boxplot(alpha = 0.8, size = 0, outlier.shape = NA)+
  geom_point(col = "black", shape = 21, alpha = 0.8,
position=position_jitterdodge(0.1), size = 2)+
  geom_boxplot(col = "black", alpha = 0, size = 0.5, outlier.shape = NA,
fatten = 1)+
  scale_fill_manual(values = DroughtPalette, name = "Drought\ntreatment",
labels = c("well-watered", "drought", "none"))+
  labs(x = "Species", y = "SER [µg/ g dry root biomass]")+
  theme_bw()+
  biomasstheme+
  scale_x_discrete(labels = c("Anthoxanthum\nodoratum",
"Dactylis\nglomerata", "Rumex\nacetosa"))
```

```
P_exudateC <- ggplot(data_conditioning_I, mapping = aes(x = species, y =
exudateC, fill = drought)) +
  geom_boxplot(alpha = 0.8, size = 0, outlier.shape = NA)+
  geom_point(col = "black", shape = 21, alpha = 0.8,
position=position_jitterdodge(0.1), size = 2)+
  geom_boxplot(col = "black", alpha = 0, size = 0.5, outlier.shape = NA,
```

```
fatten = 1)+
  scale_fill_manual(values = DroughtPalette, name = "Drought\treatment",
labels = c("well-watered", "drought", "none"))+
  labs(x = "Species", y = "Exudate C [ $\mu\text{g}/\text{plant}$ ])+
  theme_bw()+
  biomasstheme+
  scale_x_discrete(labels = c("Anthoxanthum\nodoratum",
"Dactylis\nglomerata", "Rumex\nacetosa"))

P_exudation <- ggarrange(P_SER, P_exudateC, nrow = 1, common.legend = T,
legend = "right", labels = "AUTO")
P_exudation
```



Save figure

```
ggsave(plot = P_exudation, "exudation_combined.tiff", width = 30, height =
12, units = "cm", dpi = 1000)
```

## Statistics

### Plant biomass

#### Total plant biomass

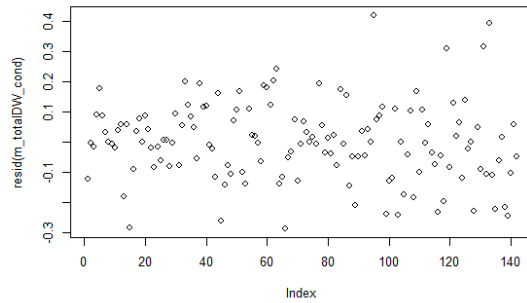
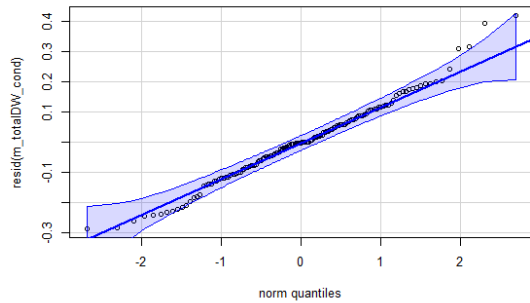
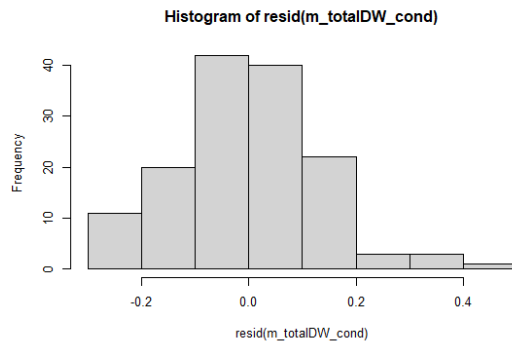
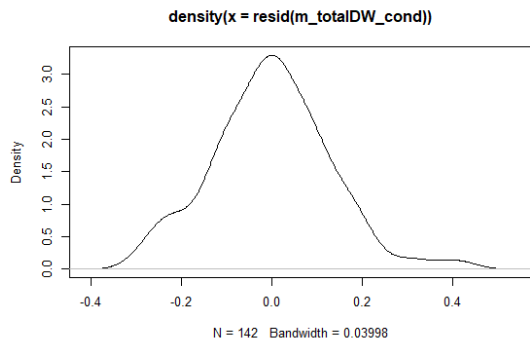
### Create model

```
m_totalDW_cond <- lmer(totalDW ~ species * drought + (1|block), data =
data_conditioning_I)
```

### Check assumptions

```
par(mfrow = c(2, 2))

plot(density(resid(m_totalDW_cond)))
hist(resid(m_totalDW_cond))
qqPlot(resid(m_totalDW_cond), id = F)
plot(resid(m_totalDW_cond))
```



Check explained variation

```
rsq.lmm(m_totalDW_cond)
```

```
$model
[1] 0.6690375
```

```
$fixed
[1] 0.6455823
```

```
$random
[1] 0.02345523
```

```
ranova(m_totalDW_cond)
```

ANOVA-like table for random-effects: Single term deletions

```
Model:
totalDW ~ species + drought + (1 | block) + species:drought
```

	npar	logLik	AIC	LRT	Df	Pr(>Chisq)
<none>	8	68.910	-121.82			
(1   block)	7	66.933	-119.87	3.9536	1	0.04677 *

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Type III ANOVA

```
anova_totalDW_cond <- anova(m_totalDW_cond, type = 3, ddf="Kenward-Roger")
anova_totalDW_cond
```

Type III Analysis of Variance Table with Kenward-Roger's method

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)
species	0.8970	0.4485	2	133.03	24.971	6.193e-10 ***
drought	3.1873	3.1873	1	133.03	177.454	< 2.2e-16 ***
species:drought	0.5578	0.2789	2	133.02	15.527	8.699e-07 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Pairwise Tukey test

```
pairs_totalDW_cond <- lsmeans(m_totalDW_cond, ~ drought | species)
tukey_totalDW_cond <- as.data.frame(pairs(pairs_totalDW_cond, adjust =
"tukey"))
tukey_totalDW_cond
```

species = Ao:

contrast	estimate	SE	df	t.ratio	p.value
D - WW	-0.2196900	0.03911645	133.03	-5.616	<.0001

species = Dg:

contrast	estimate	SE	df	t.ratio	p.value
D - WW	-0.2034036	0.03911645	133.03	-5.200	<.0001

species = Ra:

contrast	estimate	SE	df	t.ratio	p.value
D - WW	-0.4762083	0.03868801	133.00	-12.309	<.0001

Degrees-of-freedom method: kenward-roger

## Shoot biomass

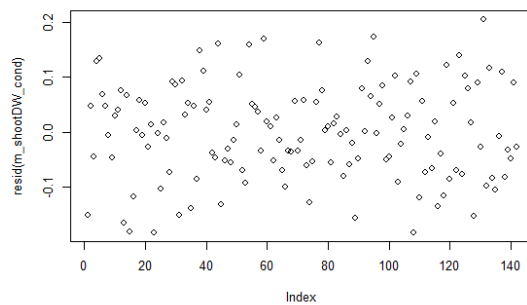
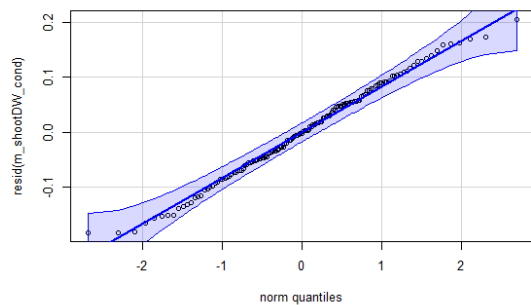
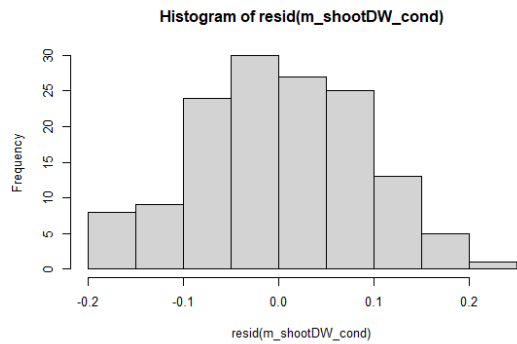
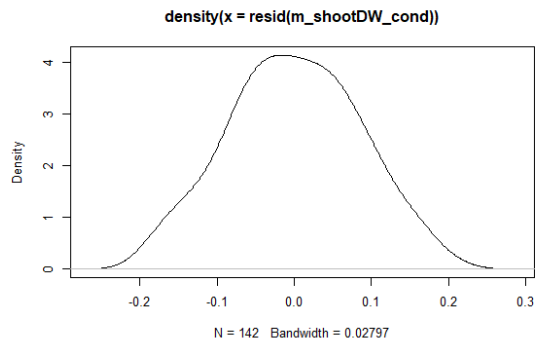
### Create model

```
m_shootDW_cond <- lmer(shootDW ~ species * drought + (1|block), data =
data_conditioning_I)
```

### Check assumptions

```
par(mfrow = c(2, 2))

plot(density(resid(m_shootDW_cond)))
hist(resid(m_shootDW_cond))
qqPlot(resid(m_shootDW_cond), id = F)
plot(resid(m_shootDW_cond))
```



Check explained variation

```
rsq.lmm(m_shootDW_cond)
```

```
$model
```

```
[1] 0.6932591
```

```
$fixed
```

```
[1] 0.6717173
```

```
$random
```

```
[1] 0.02154182
```

```
ranova(m_shootDW_cond)
```

ANOVA-like table for random-effects: Single term deletions

Model:

```
shootDW ~ species + drought + (1 | block) + species:drought
```

	npar	logLik	AIC	LRT	Df	Pr(>Chisq)
<none>	8	126.92	-237.85			
(1   block)	7	125.00	-236.01	3.8415	1	0.05 *

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Type III ANOVA

```
anova_shootDW_cond <- anova(m_shootDW_cond, type = 3, ddf="Kenward-Roger")
anova_shootDW_cond
```

```
Type III Analysis of Variance Table with Kenward-Roger's method
              Sum Sq Mean Sq NumDF  DenDF  F value Pr(>F)
species          0.89683  0.44842     2 133.03  58.5875 < 2e-16 ***
drought          1.28925  1.28925     1 133.03 168.4456 < 2e-16 ***
species:drought  0.03891  0.01946     2 133.02   2.5419 0.08253 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Pairwise Tukey test

```
pairs_shootDW_cond <- lsmeans(m_shootDW_cond, ~ drought | species)
tukey_shootDW_cond <- as.data.frame(pairs(pairs_shootDW_cond, adjust =
"tukey"))
tukey_shootDW_cond
```

```
species = Ao:
contrast estimate          SE      df t.ratio p.value
D - WW   -0.1584294 0.02553468 133.03  -6.204 <.0001
```

```
species = Dg:
contrast estimate          SE      df t.ratio p.value
D - WW   -0.1775077 0.02553468 133.03  -6.952 <.0001
```

```
species = Ra:
contrast estimate          SE      df t.ratio p.value
D - WW   -0.2360208 0.02525500 133.00  -9.346 <.0001
```

Degrees-of-freedom method: kenward-roger

## Root biomass

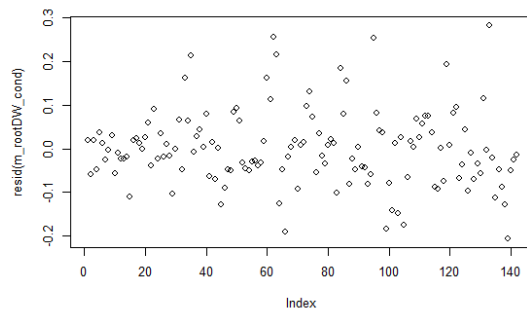
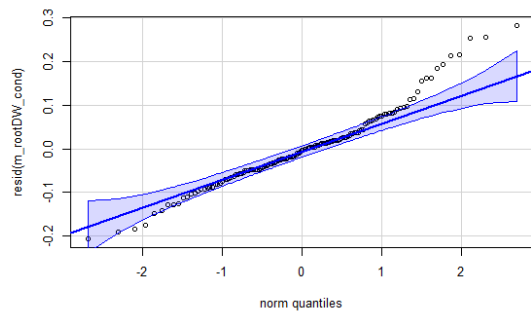
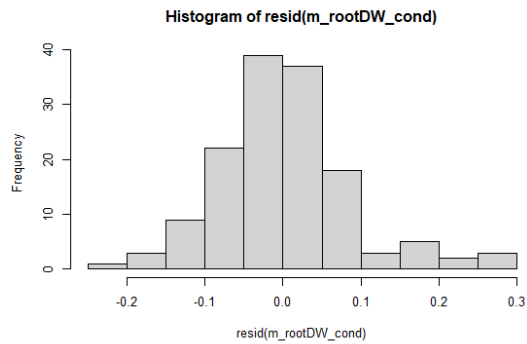
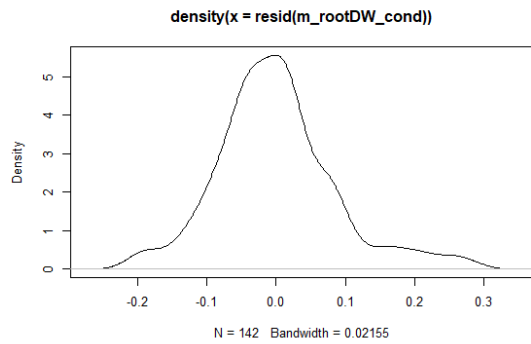
### Create model

```
m_rootDW_cond <- lmer(rootDW ~ species * drought + (1|block), data =
data_conditioning_I)
```

### Check assumptions

```
par(mfrow = c(2, 2))

plot(density(resid(m_rootDW_cond)))
hist(resid(m_rootDW_cond))
qqPlot(resid(m_rootDW_cond), id = F)
plot(resid(m_rootDW_cond))
```



Check explained variation

```
rsq.lmm(m_rootDW_cond)
```

```
$model
[1] 0.5895903
```

```
$fixed
[1] 0.5130531
```

```
$random
[1] 0.07653718
```

```
ranova(m_rootDW_cond)
```

ANOVA-like table for random-effects: Single term deletions

```
Model:
rootDW ~ species + drought + (1 | block) + species:drought
```

	npar	logLik	AIC	LRT	Df	Pr(>Chisq)
<none>	8	122.88	-229.75			
(1   block)	7	115.92	-217.84	13.913	1	0.0001914 ***

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Type III ANOVA

```
anova_rootDW_cond <- anova(m_rootDW_cond, type = 3, ddf="Kenward-Roger")
anova_rootDW_cond
```

Type III Analysis of Variance Table with Kenward-Roger's method

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)	
species	0.58057	0.29028	2	133.01	36.379	2.499e-13	***
drought	0.42177	0.42177	1	133.01	52.857	2.755e-11	***
species:drought	0.31455	0.15727	2	133.01	19.710	3.185e-08	***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Pairwise Tukey test

```
pairs_rootDW_cond <- lsmeans(m_rootDW_cond, ~ drought | species)
tukey_rootDW_cond <- as.data.frame(pairs(pairs_rootDW_cond, adjust =
"tukey"))
tukey_rootDW_cond
```

species = Ao:

contrast	estimate	SE	df	t.ratio	p.value
D - WW	-0.06107840	0.02607203	133.02	-2.343	0.0206

species = Dg:

contrast	estimate	SE	df	t.ratio	p.value
D - WW	-0.02587089	0.02607203	133.02	-0.992	0.3229

species = Ra:

contrast	estimate	SE	df	t.ratio	p.value
D - WW	-0.24018750	0.02578679	133.00	-9.314	<.0001

Degrees-of-freedom method: kenward-roger

## Root exudation

### SER

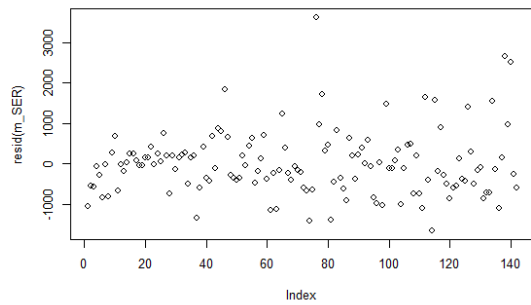
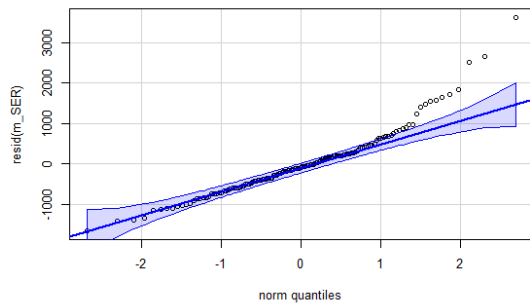
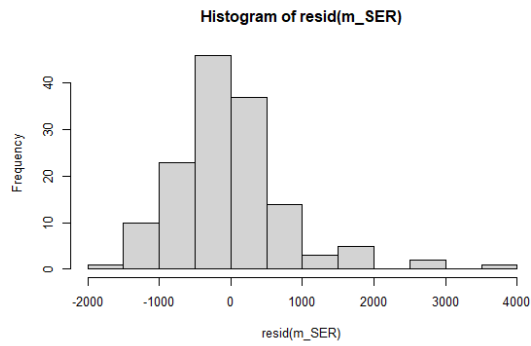
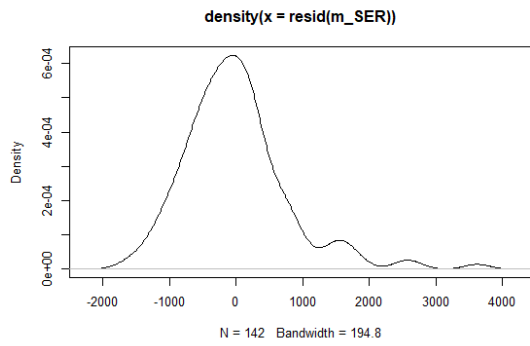
#### Create model

```
m_SER <- lmer(SER ~ species * drought + (1|block), data =
data_conditioning_I)
```

#### Check assumptions

```
par(mfrow = c(2, 2))

plot(density(resid(m_SER)))
hist(resid(m_SER))
qqPlot(resid(m_SER), id = F)
plot(resid(m_SER))
```



Check explained variation

```
rsq.lmm(m_SER)
```

```
$model
[1] 0.4509433
```

```
$fixed
[1] 0.2409695
```

```
$random
[1] 0.2099739
```

```
ranova(m_SER)
```

ANOVA-like table for random-effects: Single term deletions

Model:

```
SER ~ species + drought + (1 | block) + species:drought
```

	npar	logLik	AIC	LRT	Df	Pr(>Chisq)
<none>	8	-1118.7	2253.4			
(1   block)	7	-1134.4	2282.8	31.441	1	2.056e-08 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Type III ANOVA

```
anova_SER <- anova(m_SER, type = 3, ddf="Kenward-Roger")
anova_SER
```

```
Type III Analysis of Variance Table with Kenward-Roger's method
              Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
species      31043177 15521589     2 133.01 23.2232 2.236e-09 ***
drought       4620955  4620955     1 133.01  6.9138 0.009561 **
species:drought 3089337  1544669     2 133.00  2.3111 0.103120
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Pairwise Tukey test

```
pairs_SER <- lsmeans(m_SER, ~ drought | species)
tukey_SER <- as.data.frame(pairs(pairs_SER, adjust = "tukey"))
tukey_SER
```

```
species = Ao:
contrast estimate      SE    df t.ratio p.value
D - WW    139.9179 238.6118 133.01  0.586 0.5586
```

```
species = Dg:
contrast estimate      SE    df t.ratio p.value
D - WW    166.4112 238.6118 133.01  0.697 0.4868
```

```
species = Ra:
contrast estimate      SE    df t.ratio p.value
D - WW    776.4879 236.0025 133.00  3.290 0.0013
```

Degrees-of-freedom method: kenward-roger

## Exudate C

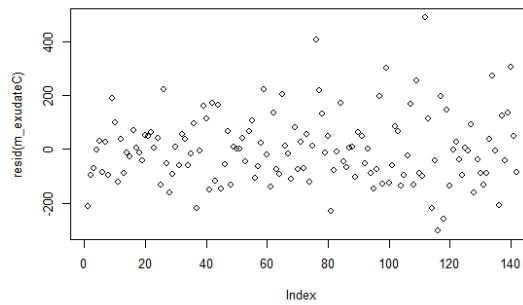
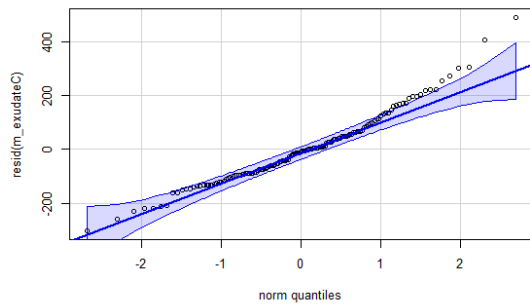
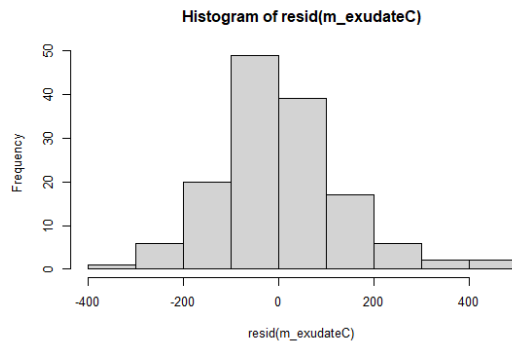
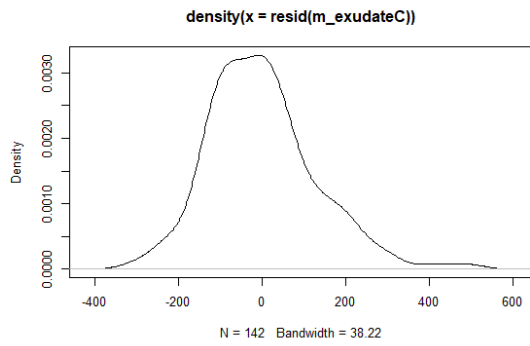
### Create model

```
m_exudateC <- lmer(exudateC ~ species * drought + (1|block), data =
data_conditioning_I)
```

### Check assumptions

```
par(mfrow = c(2, 2))

plot(density(resid(m_exudateC)))
hist(resid(m_exudateC))
qqPlot(resid(m_exudateC), id = F)
plot(resid(m_exudateC))
```



Check explained variation

```
rsq.lmm(m_exudateC)
```

```
$model
```

```
[1] 0.3196296
```

```
$fixed
```

```
[1] 0.1399797
```

```
$random
```

```
[1] 0.1796499
```

```
ranova(m_exudateC)
```

ANOVA-like table for random-effects: Single term deletions

Model:

```
exudateC ~ species + drought + (1 | block) + species:drought
```

	npar	logLik	AIC	LRT	Df	Pr(>Chisq)
<none>	8	-871.31	1758.6			
(1   block)	7	-881.76	1777.5	20.898	1	4.844e-06 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Type III ANOVA

```
anova_exudateC <- anova(m_exudateC, type = 3, ddf="Kenward-Roger")
anova_exudateC
```

```
Type III Analysis of Variance Table with Kenward-Roger's method
```

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)
species	420026	210013	2	133.01	11.853	1.832e-05 ***
drought	26401	26401	1	133.01	1.490	0.2244
species:drought	47274	23637	2	133.01	1.334	0.2669

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Pairwise Tukey test

```
pairs_exudateC <- lsmeans(m_exudateC, ~ drought | species)
tukey_exudateC <- as.data.frame(pairs(pairs_exudateC, adjust = "tukey"))
tukey_exudateC
```

```
species = Ao:
contrast estimate      SE      df t.ratio p.value
D - WW    -72.21745 38.85057 133.01  -1.859  0.0653
```

```
species = Dg:
contrast estimate      SE      df t.ratio p.value
D - WW     17.52037 38.85057 133.01   0.451  0.6527
```

```
species = Ra:
contrast estimate      SE      df t.ratio p.value
D - WW    -27.14931 38.42565 133.00  -0.707  0.4811
```

Degrees-of-freedom method: kenward-roger

## Conditioning phase II

This part of the data analysis and visualization contains all soil physical-chemical and biological data from the end of conditioning phase II, excluding microbial community data.

Filter out unconditioned background soil

```
data_conditioning_inputs <- filter(data_conditioning, input != "none")
```

## Data visualization

*Summarize all soil data in a table*

Make summary

```
summary_conditioning <- do.call(data.frame,aggregate(. ~ species * input *
drought , FUN = function(x) c(mean = mean(x),median = median(x), sd = sd(x),
se = se(x), conint = conint(x)), data =
```

```
dplyr::select(data_conditioning_inputs, !number & !soil_ID & !block &
!species_full)))
```

Create table from summary

```
table_soil <- summary_conditioning %>% mutate(pH =
factor(paste(specify_decimal(pH.mean, 2), specify_decimal(pH.se, 2), sep = "
± ")), moisture = factor(paste(specify_decimal(moisture.mean, 2),
specify_decimal(moisture.se, 2), sep = " ± ")), SOM =
factor(paste(specify_decimal(SOM.mean, 2), specify_decimal(SOM.se, 2), sep =
" ± ")), DOC = factor(paste(specify_decimal(DOC.mean, 2),
specify_decimal(DOC.se, 2), sep = " ± ")), TDC =
factor(paste(specify_decimal(TDC.mean, 2), specify_decimal(TDC.se, 2), sep =
" ± ")), availableP = factor(paste(specify_decimal(availableP.mean, 3),
specify_decimal(availableP.se, 3), sep = " ± ")), availableN =
factor(paste(specify_decimal(availableN.mean, 2),
specify_decimal(availableN.se, 2), sep = " ± ")), availableNO3 =
factor(paste(specify_decimal(availableNO3.mean, 2),
specify_decimal(availableNO3.se, 2), sep = " ± ")), availableNH4 =
factor(paste(specify_decimal(availableNH4.mean, 2),
specify_decimal(availableNH4.se, 2), sep = " ± ")), TDN =
factor(paste(specify_decimal(TDN.mean, 2), specify_decimal(TDN.se, 2), sep =
" ± ")), microbialN = factor(paste(specify_decimal(microbialN.mean, 2),
specify_decimal(microbialN.se, 2), sep = " ± ")), microbialC =
factor(paste(specify_decimal(microbialC.mean, 2),
specify_decimal(microbialC.se, 2), sep = " ± "))) %>%
  dplyr::select(species, drought, input, moisture, pH, SOM, DOC, TDC,
availableP, availableN, availableNO3, availableNH4, TDN, microbialN,
microbialC)
```

```
glimpse(table_soil)
```

```
Rows: 24
Columns: 15
$ species      <fct> Ao, Dg, Ra, Ao, Dg, Ra, Ao, Dg, Ra, Ao, Dg, Ra, Ao, Dg,
R...
$ drought      <fct> WW, WW, WW, WW, WW, WW, WW, WW, WW, WW, WW, WW, D, D, D,
...
$ input        <fct> plant, plant, plant, inoculum, inoculum, inoculum,
exudat...
$ moisture     <fct> 18.82 ± 1.09, 18.62 ± 0.44, 18.53 ± 1.79, 19.58 ± 0.76,
1...
$ pH           <fct> 8.05 ± 0.03, 8.04 ± 0.03, 8.06 ± 0.03, 8.11 ± 0.03, 8.08
...
$ SOM          <fct> 2.83 ± 0.04, 2.88 ± 0.07, 2.79 ± 0.05, 2.66 ± 0.08, 2.74
...
$ DOC          <fct> 15.06 ± 0.96, 15.23 ± 0.28, 14.76 ± 1.06, 14.61 ± 0.66,
1...
$ TDC          <fct> 33.22 ± 3.32, 34.09 ± 2.23, 33.40 ± 3.74, 36.36 ± 3.09,
3...
```

```

$ availableP <fct> 0.280 ± 0.038, 0.270 ± 0.045, 0.223 ± 0.012, 0.301 ±
0.03...
$ availableN <fct> 4.27 ± 0.56, 5.29 ± 0.18, 4.20 ± 0.46, 5.95 ± 1.19, 4.74
...
$ availableNO3 <fct> 4.08 ± 0.58, 4.98 ± 0.20, 4.08 ± 0.43, 5.74 ± 1.14, 4.61
...
$ availableNH4 <fct> 0.19 ± 0.11, 0.31 ± 0.05, 0.12 ± 0.06, 0.21 ± 0.05, 0.13
...
$ TDN <fct> 4.07 ± 0.18, 4.64 ± 0.25, 4.13 ± 0.04, 5.01 ± 0.55, 4.35
...
$ microbialN <fct> 34.64 ± 1.83, 33.27 ± 1.04, 34.51 ± 2.04, 29.40 ± 1.42,
3...
$ microbialC <fct> 271.44 ± 16.77, 280.40 ± 9.49, 297.82 ± 9.55, 241.94 ±
19...

```

Save table as csv file

```

write.csv(table_soil, "table_soil.csv", row.names = FALSE,
fileEncoding="Windows-1252")

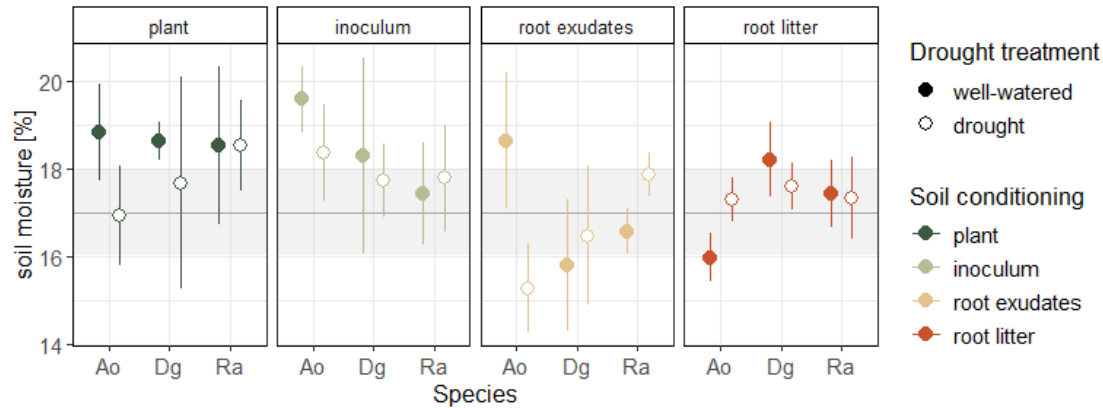
```

*Make figures for all physical-chemical soil parameters*

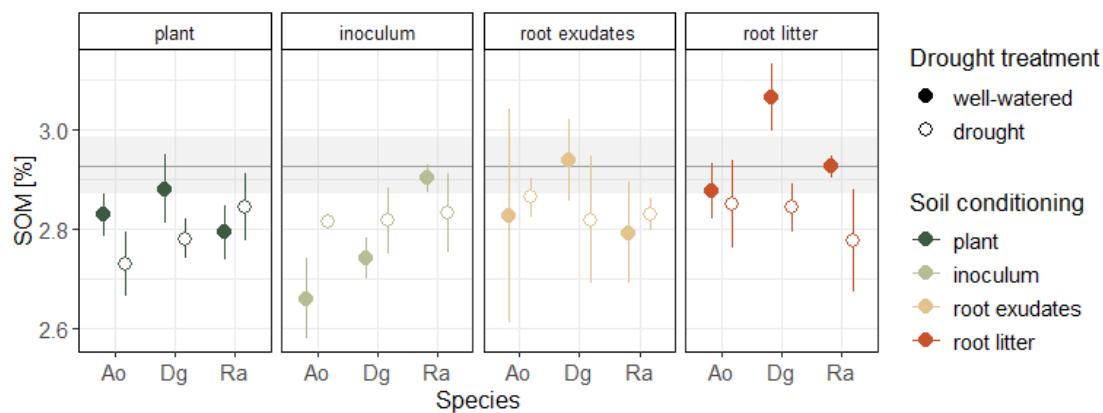
```

P_moisture_means <- ggplot(data = summary_conditioning, mapping = aes(y =
moisture.mean, x = species, color = input, shape = drought))+
  geom_hline(yintercept = mean(filter(data_conditioning, input ==
"none")$moisture), linewidth = 0.5, col = "gray60")+
  geom_rect(xmin = 0, xmax = 4, ymin = mean(filter(data_conditioning, input
== "none")$moisture)-se(filter(data_conditioning, input == "none")$moisture),
ymax = mean(filter(data_conditioning, input ==
"none")$moisture)+se(filter(data_conditioning, input == "none")$moisture),
fill = "gray80", alpha = 0.05, col = NA)+
  geom_errorbar(aes(ymin = moisture.mean - moisture.se, ymax = moisture.mean
+ moisture.se), position = position_dodge(0.7), width = 0, linewidth = 0.5)+
  geom_point(alpha = 1, fill = "white", position=position_dodge(0.7), size
= 3, stroke = 0.7)+
  scale_shape_manual(values = c(16, 21), name = "Drought treatment", labels =
c("well-watered", "drought"))+
  scale_color_manual(values = InputPalette, name = "Soil conditioning",
labels = c("plant", "inoculum", "root exudates", "root litter",
"unconditioned"))+
  facet_wrap(facets = "input", nrow = 1, labeller = facets_input)+
  theme_bw()+
  soiltheme +
  labs(x = "Species", y = "soil moisture [%]")
P_moisture_means

```



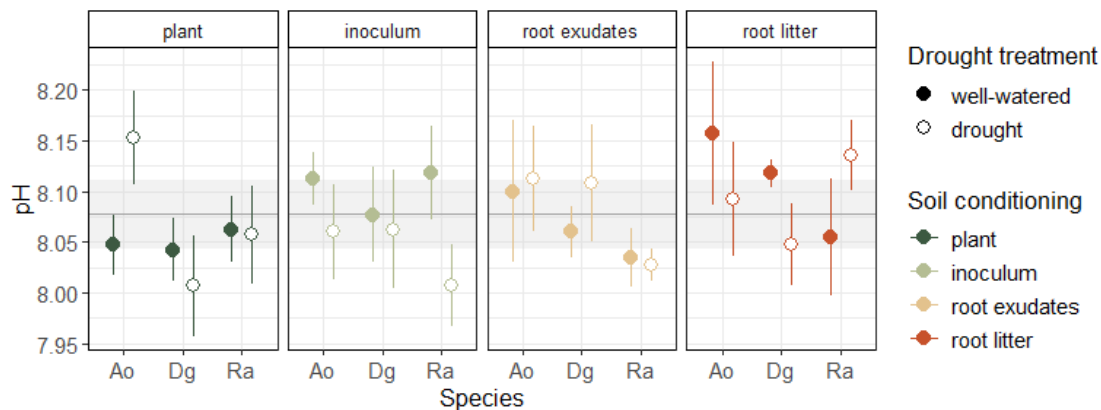
```
P_SOM_means <- ggplot(data = summary_conditioning, mapping = aes(y =
SOM.mean, x = species, color = input, shape = drought))+
  geom_hline(yintercept = mean(filter(data_conditioning, input ==
"none")$SOM), linewidth = 0.5, col = "gray60")+
  geom_rect(xmin = 0, xmax = 4, ymin = mean(filter(data_conditioning, input ==
"none")$SOM)-se(filter(data_conditioning, input == "none")$SOM), ymax =
mean(filter(data_conditioning, input ==
"none")$SOM)+se(filter(data_conditioning, input == "none")$SOM), fill =
"gray80", alpha = 0.05, col = NA)+
  geom_errorbar(aes(ymin = SOM.mean - SOM.se, ymax = SOM.mean + SOM.se),
position = position_dodge(0.7), width = 0, linewidth = 0.5)+
  geom_point(alpha = 1, fill = "white", position=position_dodge(0.7), size
= 3, stroke = 0.7)+
  scale_shape_manual(values = c(16, 21), name = "Drought treatment", labels =
c("well-watered", "drought"))+
  scale_color_manual(values = InputPalette, name = "Soil conditioning",
labels = c("plant", "inoculum", "root exudates", "root litter",
"unconditioned"))+
  facet_wrap(facets = "input", nrow = 1, labeller = facets_input)+
  theme_bw()+
  soiltheme +
  labs(x = "Species", y = "SOM [%]")
P_SOM_means
```



```

P_pH_means <- ggplot(data = summary_conditioning, mapping = aes(y = pH.mean,
x = species, color = input, shape = drought))+
  geom_hline(yintercept = mean(filter(data_conditioning, input ==
"none")$pH), linewidth = 0.5, col = "gray60")+
  geom_rect(xmin = 0, xmax = 4, ymin = mean(filter(data_conditioning, input
== "none")$pH)-se(filter(data_conditioning, input == "none")$pH), ymax =
mean(filter(data_conditioning, input ==
"none")$pH)+se(filter(data_conditioning, input == "none")$pH), fill =
"gray80", alpha = 0.05, col = NA)+
  geom_errorbar(aes(ymin = pH.mean - pH.se, ymax = pH.mean + pH.se),
position = position_dodge(0.7), width = 0, linewidth = 0.5)+
  geom_point(alpha = 1, fill = "white", position=position_dodge(0.7), size
= 3, stroke = 0.7)+
  scale_shape_manual(values = c(16, 21), name = "Drought treatment", labels =
c("well-watered", "drought"))+
  scale_color_manual(values = InputPalette, name = "Soil conditioning",
labels = c("plant", "inoculum", "root exudates", "root litter",
"unconditioned"))+
  facet_wrap(facets = "input", nrow = 1, labeller = facets_input)+
  theme_bw()+
  soiltheme +
  labs(x = "Species", y = "pH")
P_pH_means

```



```

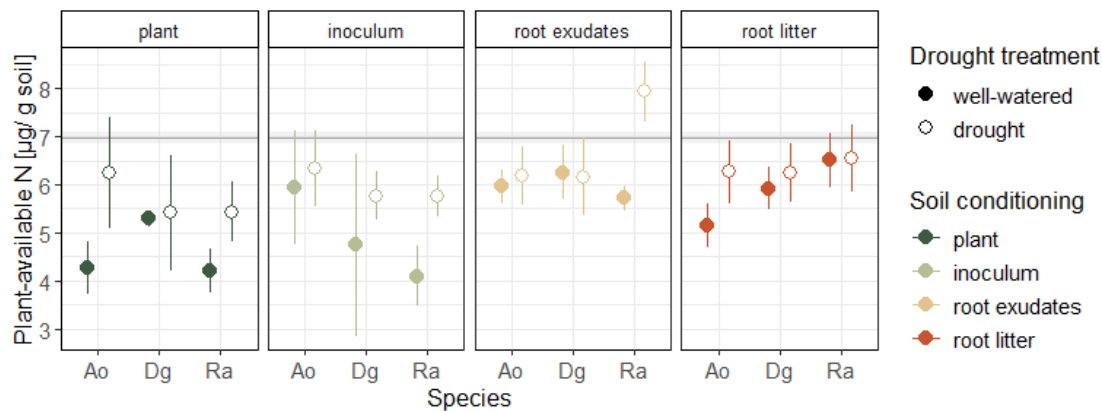
P_availableN_means <- ggplot(data = summary_conditioning, mapping = aes(y =
availableN.mean, x = species, color = input, shape = drought))+
  geom_hline(yintercept = mean(filter(data_conditioning, input ==
"none")$availableN), linewidth = 0.5, col = "gray60")+
  geom_rect(xmin = 0, xmax = 4, ymin = mean(filter(data_conditioning, input
== "none")$availableN)-se(filter(data_conditioning, input ==
"none")$availableN),
  ymax = mean(filter(data_conditioning, input ==
"none")$availableN)+se(filter(data_conditioning, input ==
"none")$availableN), fill = "gray80", alpha = 0.05, col = NA)+
  geom_errorbar(aes(ymin = availableN.mean - availableN.se, ymax =
availableN.mean + availableN.se), position = position_dodge(0.7), width = 0,
linewidth = 0.5)+

```

```

geom_point( alpha = 1, fill = "white", position=position_dodge(0.7), size
= 3, stroke = 0.7)+
  scale_shape_manual(values = c(16, 21), name = "Drought treatment", labels =
c("well-watered", "drought"))+
  scale_color_manual(values = InputPalette, name = "Soil conditioning",
labels = c("plant", "inoculum", "root exudates", "root litter",
"unconditioned"))+
  facet_wrap(facets = "input", nrow = 1, labeller = facets_input)+
  theme_bw()+
  soiltheme +
  labs(x = "Species", y = "Plant-available N [ $\mu\text{g/g}$  soil]")
P_availableN_means

```

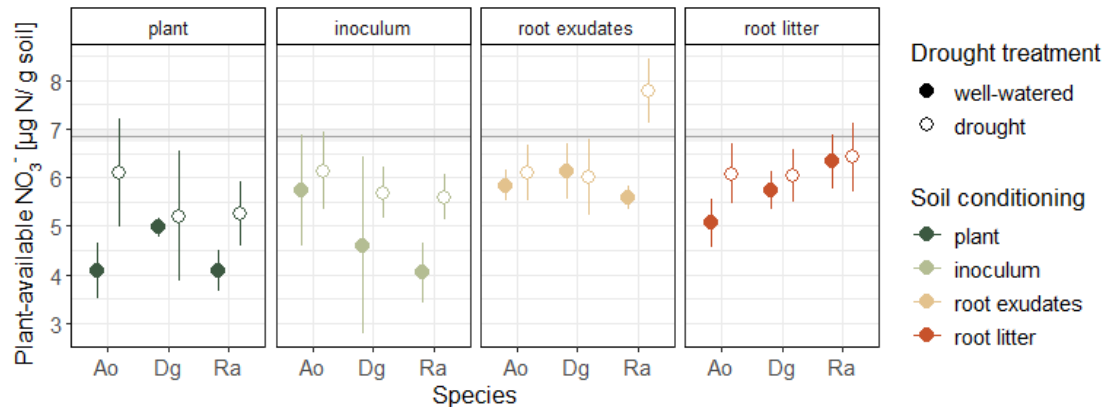


```

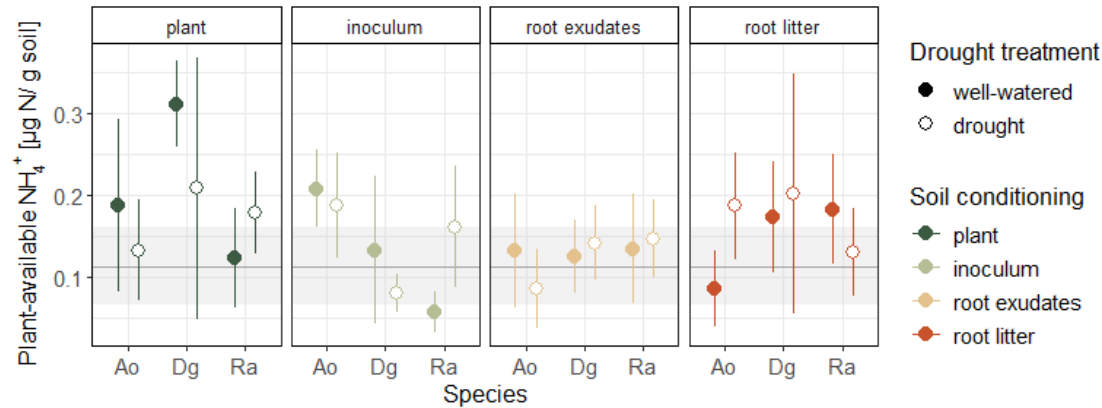
P_availableN03_means <- ggplot(data = summary_conditioning, mapping = aes(y =
availableNO3.mean, x = species, color = input, shape = drought))+
  geom_hline(yintercept = mean(filter(data_conditioning, input ==
"none")$availableNO3), linewidth = 0.5, col = "gray60")+
  geom_rect(xmin = 0, xmax = 4, ymin = mean(filter(data_conditioning, input
== "none")$availableNO3)-se(filter(data_conditioning, input ==
"none")$availableNO3),
            ymax = mean(filter(data_conditioning, input ==
"none")$availableNO3)+se(filter(data_conditioning, input ==
"none")$availableNO3), fill = "gray80", alpha = 0.05, col = NA)+
  geom_errorbar(aes(ymin = availableNO3.mean - availableNO3.se, ymax =
availableNO3.mean + availableNO3.se), position = position_dodge(0.7), width
= 0, linewidth = 0.5)+
  geom_point( alpha = 1, fill = "white", position=position_dodge(0.7), size
= 3, stroke = 0.7)+
  scale_shape_manual(values = c(16, 21), name = "Drought treatment", labels =
c("well-watered", "drought"))+
  scale_color_manual(values = InputPalette, name = "Soil conditioning",
labels = c("plant", "inoculum", "root exudates", "root litter",
"unconditioned"))+
  facet_wrap(facets = "input", nrow = 1, labeller = facets_input)+
  theme_bw()+
  soiltheme +
  labs(x = "Species", y = expression(paste("Plant-available ", NO[3]^{" -

```

```
"}), " [µg N/ g soil]"))))
P_availableN03_means
```



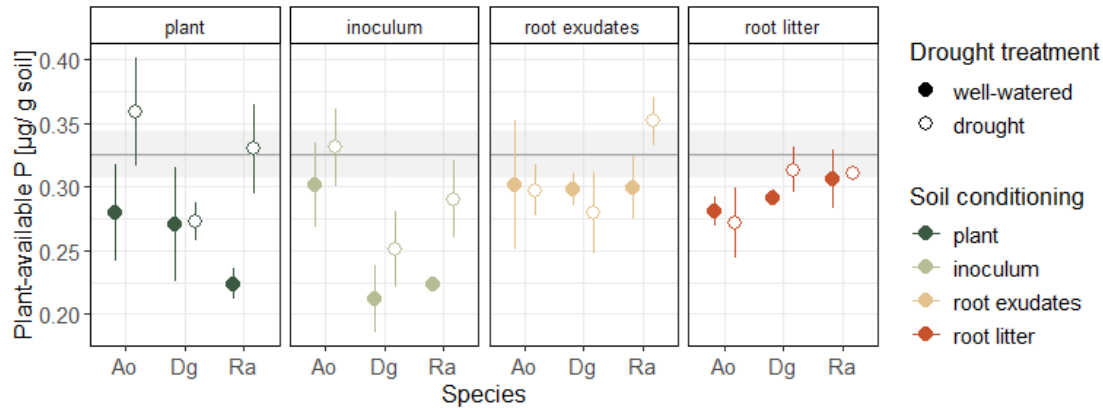
```
P_availableNH4_means <- ggplot(data = summary_conditioning, mapping = aes(y =
availableNH4.mean, x = species, color = input, shape = drought))+
  geom_hline(yintercept = mean(filter(data_conditioning, input ==
"none")$availableNH4), linewidth = 0.5, col = "gray60")+
  geom_rect(xmin = 0, xmax = 4, ymin = mean(filter(data_conditioning, input
== "none")$availableNH4)-se(filter(data_conditioning, input ==
"none")$availableNH4),
            ymax = mean(filter(data_conditioning, input ==
"none")$availableNH4)+se(filter(data_conditioning, input ==
"none")$availableNH4), fill = "gray80", alpha = 0.05, col = NA)+
  geom_errorbar(aes(ymin = availableNH4.mean - availableNH4.se, ymax =
availableNH4.mean + availableNH4.se), position = position_dodge(0.7), width
= 0, linewidth = 0.5)+
  geom_point(alpha = 1, fill = "white", position=position_dodge(0.7), size
= 3, stroke = 0.7)+
  scale_shape_manual(values = c(16, 21), name = "Drought treatment", labels =
c("well-watered", "drought"))+
  scale_color_manual(values = InputPalette, name = "Soil conditioning",
labels = c("plant", "inoculum", "root exudates", "root litter",
"unconditioned"))+
  facet_wrap(facets = "input", nrow = 1, labeller = facets_input)+
  theme_bw()+
  soiltheme +
  labs(x = "Species", y = expression(paste("Plant-available " , NH[4]^{"}
+"), " [µg N/ g soil]"))))
P_availableNH4_means
```



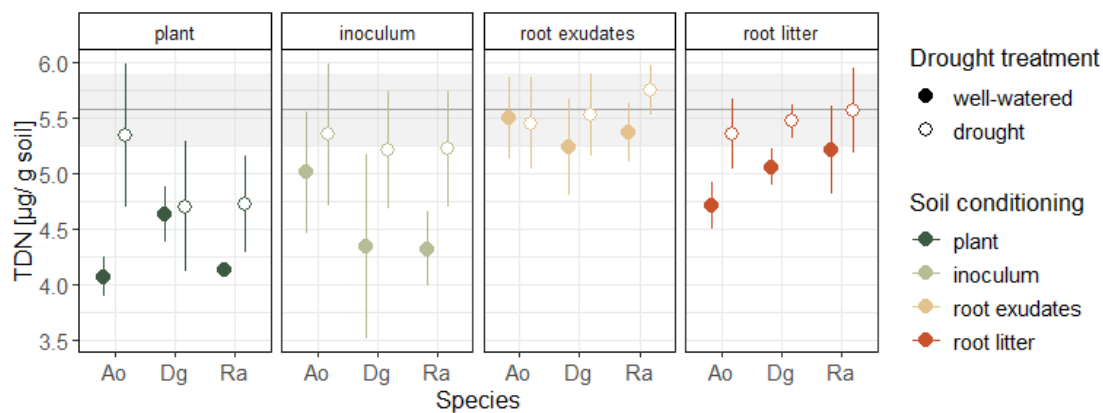
```

P_availableP_means <- ggplot(data = summary_conditioning, mapping = aes(y =
availableP.mean, x = species, color = input, shape = drought))+
  geom_hline(yintercept = mean(filter(data_conditioning, input ==
"none")$availableP), linewidth = 0.5, col = "gray60")+
  geom_rect(xmin = 0, xmax = 4, ymin = mean(filter(data_conditioning, input
== "none")$availableP)-se(filter(data_conditioning, input ==
"none")$availableP),
            ymax = mean(filter(data_conditioning, input ==
"none")$availableP)+se(filter(data_conditioning, input ==
"none")$availableP), fill = "gray80", alpha = 0.05, col = NA)+
  geom_errorbar(aes(ymin = availableP.mean - availableP.se, ymax =
availableP.mean + availableP.se), position = position_dodge(0.7), width = 0,
linewidth = 0.5)+
  geom_point(alpha = 1, fill = "white", position=position_dodge(0.7), size
= 3, stroke = 0.7)+
  scale_shape_manual(values = c(16, 21), name = "Drought treatment", labels =
c("well-watered", "drought"))+
  scale_color_manual(values = InputPalette, name = "Soil conditioning",
labels = c("plant", "inoculum", "root exudates", "root litter",
"unconditioned"))+
  facet_wrap(facets = "input", nrow = 1, labeller = facets_input)+
  theme_bw()+
  soiltheme +
  labs(x = "Species", y = "Plant-available P [ $\mu\text{g/ g soil}$ ])
P_availableP_means

```



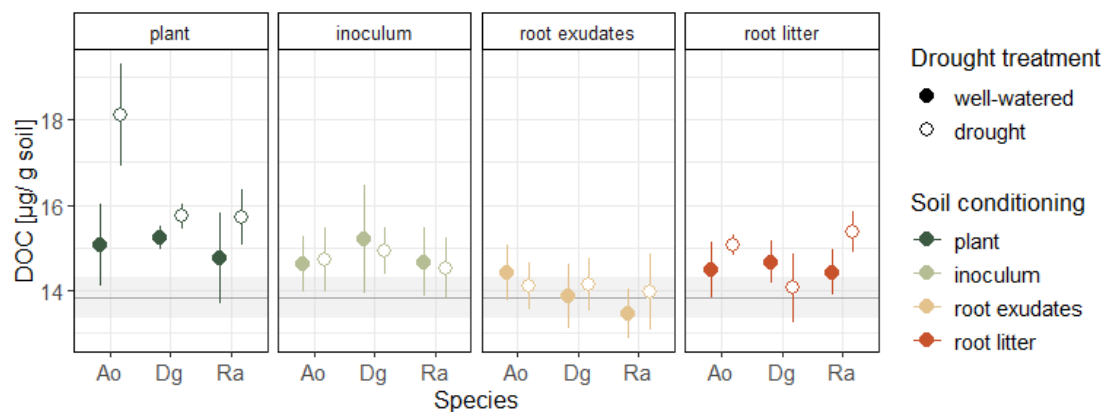
```
P_TDN_means <- ggplot(data = summary_conditioning, mapping = aes(y =
TDN.mean, x = species, color = input, shape = drought))+
  geom_hline(yintercept = mean(filter(data_conditioning, input ==
"none")$TDN), linewidth = 0.5, col = "gray60")+
  geom_rect(xmin = 0, xmax = 4, ymin = mean(filter(data_conditioning, input
== "none")$TDN)-se(filter(data_conditioning, input == "none")$TDN),
          ymax = mean(filter(data_conditioning, input ==
"none")$TDN)+se(filter(data_conditioning, input == "none")$TDN), fill =
"gray80", alpha = 0.05, col = NA)+
  geom_errorbar(aes(ymin = TDN.mean - TDN.se, ymax = TDN.mean + TDN.se),
position = position_dodge(0.7), width = 0, linewidth = 0.5)+
  geom_point(alpha = 1, fill = "white", position=position_dodge(0.7), size
= 3, stroke = 0.7)+
  scale_shape_manual(values = c(16, 21), name = "Drought treatment", labels =
c("well-watered", "drought"))+
  scale_color_manual(values = InputPalette, name = "Soil conditioning",
labels = c("plant", "inoculum", "root exudates", "root litter",
"unconditioned"))+
  facet_wrap(facets = "input", nrow = 1, labeller = facets_input)+
  theme_bw()+
  soiltheme +
  labs(x = "Species", y = "TDN [ $\mu\text{g/g soil}$ ]" )
P_TDN_means
```



```

P_DOC_means <- ggplot(data = summary_conditioning, mapping = aes(y =
DOC.mean, x = species, color = input, shape = drought))+
  geom_hline(yintercept = mean(filter(data_conditioning, input ==
"none")$DOC), linewidth = 0.5, col = "gray60")+
  geom_rect(xmin = 0, xmax = 4, ymin = mean(filter(data_conditioning, input
== "none")$DOC)-se(filter(data_conditioning, input == "none")$DOC),
            ymax = mean(filter(data_conditioning, input ==
"none")$DOC)+se(filter(data_conditioning, input == "none")$DOC), fill =
"gray80", alpha = 0.05, col = NA)+
  geom_errorbar(aes(ymin = DOC.mean - DOC.se, ymax = DOC.mean + DOC.se),
position = position_dodge(0.7), width = 0, linewidth = 0.5)+
  geom_point( alpha = 1, fill = "white", position=position_dodge(0.7), size
= 3, stroke = 0.7)+
  scale_shape_manual(values = c(16, 21), name = "Drought treatment", labels =
c("well-watered", "drought"))+
  scale_color_manual(values = InputPalette, name = "Soil conditioning",
labels = c("plant", "inoculum", "root exudates", "root litter",
"unconditioned"))+
  facet_wrap(facets = "input", nrow = 1, labeller = facets_input)+
  theme_bw()+
  soiltheme +
  labs(x = "Species", y = "DOC [µg/ g soil]")
P_DOC_means

```



```

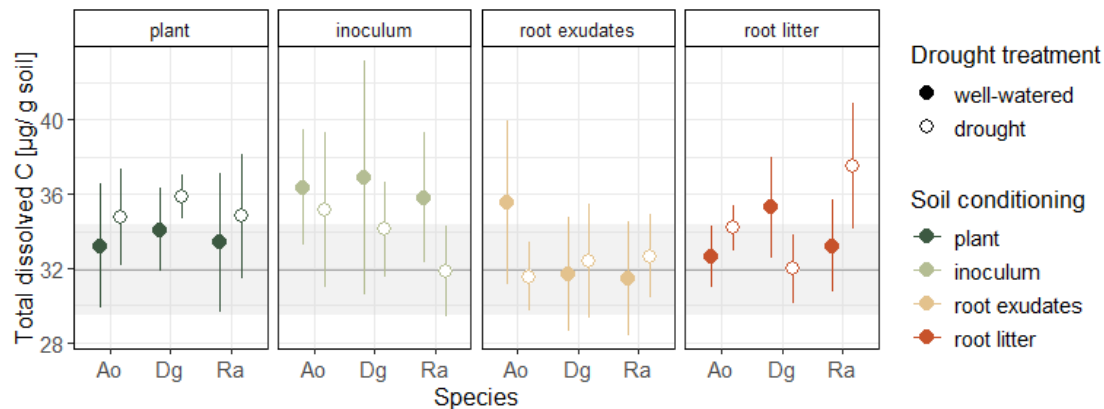
P_TDC_means <- ggplot(data = summary_conditioning, mapping = aes(y =
TDC.mean, x = species, color = input, shape = drought))+
  geom_hline(yintercept = mean(filter(data_conditioning, input ==
"none")$TDC), linewidth = 0.5, col = "gray60")+
  geom_rect(xmin = 0, xmax = 4, ymin = mean(filter(data_conditioning, input
== "none")$TDC)-se(filter(data_conditioning, input == "none")$TDC),
            ymax = mean(filter(data_conditioning, input ==
"none")$TDC)+se(filter(data_conditioning, input == "none")$TDC), fill =
"gray80", alpha = 0.05, col = NA)+
  geom_errorbar(aes(ymin = TDC.mean - TDC.se, ymax = TDC.mean + TDC.se),
position = position_dodge(0.7), width = 0, linewidth = 0.5)+
  geom_point( alpha = 1, fill = "white", position=position_dodge(0.7), size
= 3, stroke = 0.7)+

```

```

scale_shape_manual(values = c(16, 21), name = "Drought treatment", labels =
c("well-watered", "drought"))+
scale_color_manual(values = InputPalette, name = "Soil conditioning",
labels = c("plant", "inoculum", "root exudates", "root litter",
"unconditioned"))+
facet_wrap(facets = "input", nrow = 1, labeller = facets_input)+
theme_bw()+
soiltheme +
labs(x = "Species", y = "Total dissolved C [ $\mu\text{g}/\text{g}$  soil]")
P_TDC_means

```



Create figures for microbial biomass

```

P_micC_means <- ggplot(data = summary_conditioning, mapping = aes(y =
microbialC.mean, x = species, color = input, shape = drought))+
geom_hline(yintercept = mean(filter(data_conditioning, input ==
"none")$microbialC), linewidth = 0.5, col = "gray60")+
geom_rect(xmin = 0, xmax = 4, ymin = mean(filter(data_conditioning, input
== "none")$microbialC) - se(filter(data_conditioning, input ==
"none")$microbialC),
ymax = mean(filter(data_conditioning, input ==
"none")$microbialC) + se(filter(data_conditioning, input ==
"none")$microbialC), fill = "gray80", alpha = 0.05, col = NA)+
geom_errorbar(aes(ymin = microbialC.mean - microbialC.se, ymax =
microbialC.mean + microbialC.se), position = position_dodge(0.7), width = 0,
linewidth = 0.5)+
geom_point(alpha = 1, fill = "white", position=position_dodge(0.7), size
= 3, stroke = 0.7))+
scale_shape_manual(values = c(16, 21), name = "Drought treatment", labels =
c("well-watered", "drought"))+
scale_color_manual(values = InputPalette, name = "Soil conditioning",
labels = c("plant", "inoculum", "root exudates", "root litter",
"unconditioned"))+
facet_wrap(facets = "input", nrow = 1, labeller = facets_input)+
theme_bw()+
soiltheme +
labs(x = "Species", y = "Microbial biomass C [ $\mu\text{g}/\text{g}$  soil]")

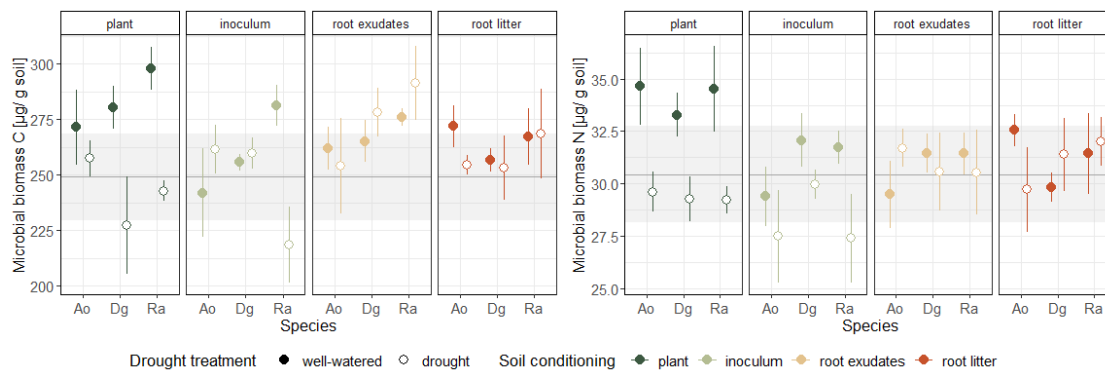
```

```

P_micN_means <- ggplot(data = summary_conditioning, mapping = aes(y =
microbialN.mean, x = species, color = input, shape = drought))+
  geom_hline(yintercept = mean(filter(data_conditioning, input ==
"none")$microbialN), linewidth = 0.5, col = "gray60")+
  geom_rect(xmin = 0, xmax = 4, ymin = mean(filter(data_conditioning, input
== "none")$microbialN)-se(filter(data_conditioning, input ==
"none")$microbialN),
            ymax = mean(filter(data_conditioning, input ==
"none")$microbialN)+se(filter(data_conditioning, input ==
"none")$microbialN), fill = "gray80", alpha = 0.05, col = NA)+
  geom_errorbar(aes(ymin = microbialN.mean - microbialN.se, ymax =
microbialN.mean + microbialN.se), position = position_dodge(0.7), width = 0,
linewidth = 0.5)+
  geom_point(alpha = 1, fill = "white", position=position_dodge(0.7), size
= 3, stroke = 0.7)+
  scale_shape_manual(values = c(16, 21), name = "Drought treatment", labels =
c("well-watered", "drought"))+
  scale_color_manual(values = InputPalette, name = "Soil conditioning",
labels = c("plant", "inoculum", "root exudates", "root litter",
"unconditioned"))+
  facet_wrap(facets = "input", nrow = 1, labeller = facets_input)+
  theme_bw()+
  soiltheme +
  labs(x = "Species", y = "Microbial biomass N [µg/ g soil]")

P_micbiomass_means <- ggarrange(P_micC_means, P_micN_means, nrow =
1,common.legend = T, legend = "bottom")
P_micbiomass_means

```



Save figure

```

ggsave(plot = P_micbiomass_means, "microbialCN_means_combined.tiff", width =
30, height = 9, units = "cm", dpi = 1000)

```

## Statistics

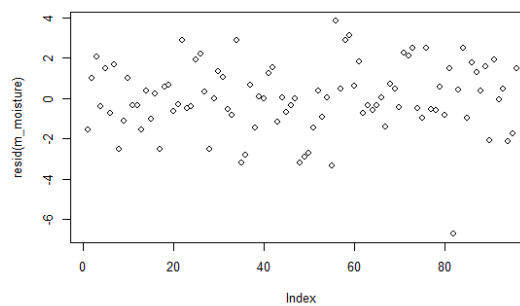
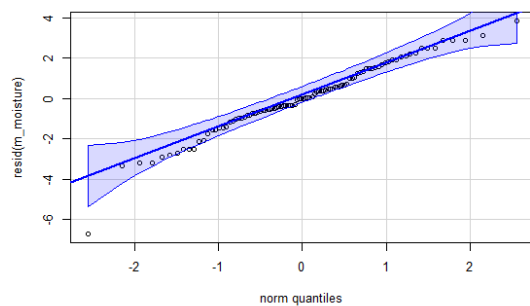
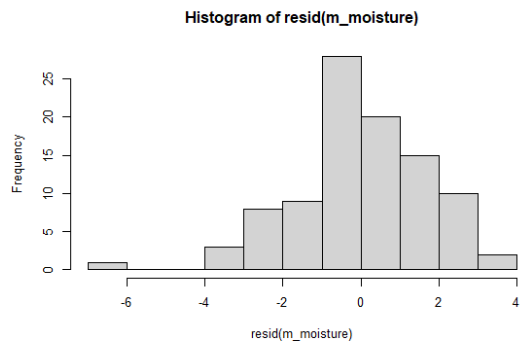
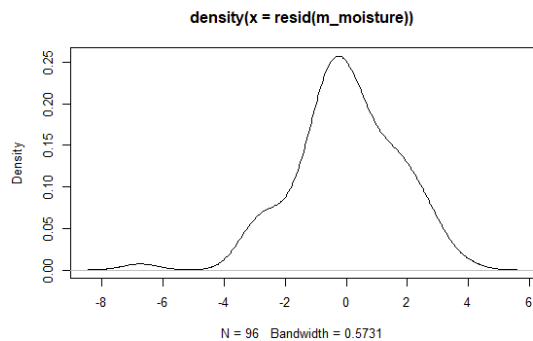
### Soil moisture

#### Create model

```
m_moisture <- lmer(moisture ~ input * drought * species + (1|block),  
data_conditioning_inputs)
```

#### Check assumptions

```
par(mfrow = c(2, 2))  
  
plot(density(resid(m_moisture)))  
hist(resid(m_moisture))  
qqPlot(resid(m_moisture), id = F)  
plot(resid(m_moisture))
```



#### Check explained variation

```
rsq.lmm(m_moisture)
```

```
$model  
[1] 0.3674757
```

```
$fixed  
[1] 0.2204989
```

```

$random
[1] 0.1469768

ranova(m_moisture)

ANOVA-like table for random-effects: Single term deletions

Model:
moisture ~ input + drought + species + (1 | block) + input:drought +
input:species + drought:species + input:drought:species
      npar  logLik    AIC    LRT Df Pr(>Chisq)
<none>      26 -172.56 397.12
(1 | block)  25 -179.67 409.34 14.225  1  0.0001622 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

### Type III ANOVA

```

anova_moisture <- anova(m_moisture, type = 3, ddf="Kenward-Roger")
anova_moisture

Type III Analysis of Variance Table with Kenward-Roger's method
      Sum Sq Mean Sq NumDF DenDF F value  Pr(>F)
input      57.431  19.1437     3    69  4.7136 0.004735 **
drought     1.475   1.4751     1    69  0.3632 0.548708
species     2.409   1.2045     2    69  0.2966 0.744306
input:drought  3.369   1.1229     3    69  0.2765 0.842177
input:species 20.964   3.4939     6    69  0.8603 0.528540
drought:species  6.853   3.4264     2    69  0.8436 0.434527
input:drought:species 17.975   2.9958     6    69  0.7376 0.621081
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

### Soil organic matter (SOM)

#### Create model

```

m_SOM <- lmer(SOM ~ input * drought * species + (1|block),
data_conditioning_inputs)

```

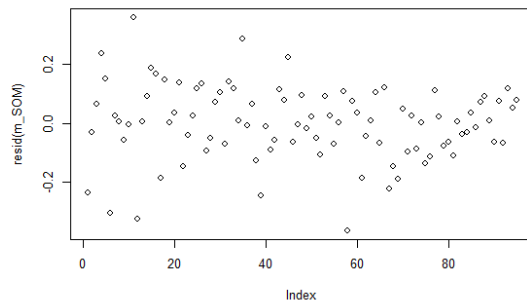
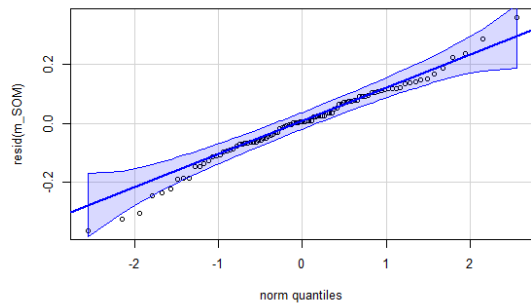
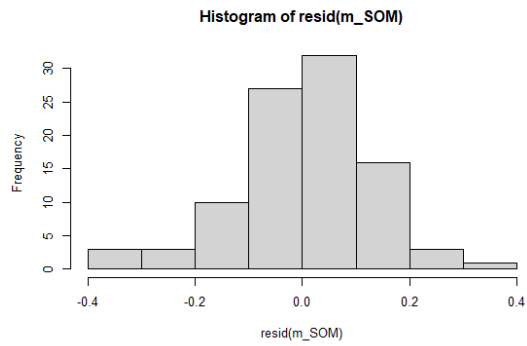
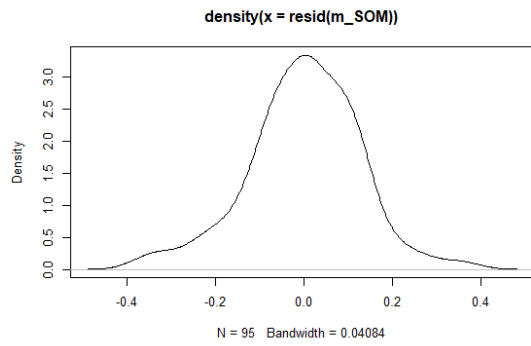
#### Check assumptions

```

par(mfrow = c(2, 2))

plot(density(resid(m_SOM)))
hist(resid(m_SOM))
qqPlot(resid(m_SOM), id = F)
plot(resid(m_SOM))

```



Check explained variation

```
rsq.lmm(m_SOM)
```

```
$model
```

```
[1] 0.2671054
```

```
$fixed
```

```
[1] 0.2618024
```

```
$random
```

```
[1] 0.005302994
```

```
ranova(m_SOM)
```

ANOVA-like table for random-effects: Single term deletions

Model:

```
SOM ~ input + drought + species + (1 | block) + input:drought + input:species
+ drought:species + input:drought:species
```

	npar	logLik	AIC	LRT	Df	Pr(>Chisq)
<none>	26	19.152	13.696			
(1   block)	25	19.111	11.777	0.080587	1	0.7765

Type III ANOVA

```
anova_SOM <- anova(m_SOM, type = 3, ddf="Kenward-Roger")
anova_SOM
```

## Type III Analysis of Variance Table with Kenward-Roger's method

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)
input	0.113431	0.037810	3	68.062	1.7797	0.1593
drought	0.029025	0.029025	1	68.064	1.3662	0.2465
species	0.057783	0.028892	2	68.063	1.3599	0.2636
input:drought	0.097717	0.032572	3	68.062	1.5332	0.2138
input:species	0.093903	0.015650	6	68.059	0.7367	0.6219
drought:species	0.043940	0.021970	2	68.063	1.0341	0.3611
input:drought:species	0.100867	0.016811	6	68.059	0.7913	0.5799

## pH

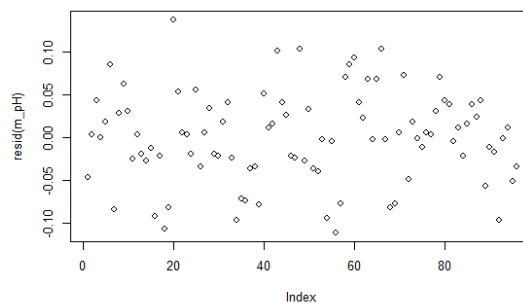
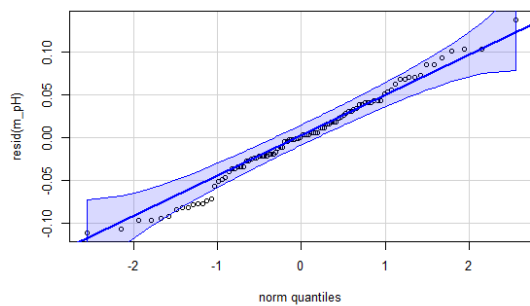
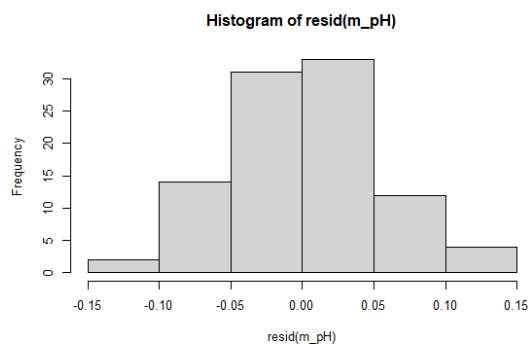
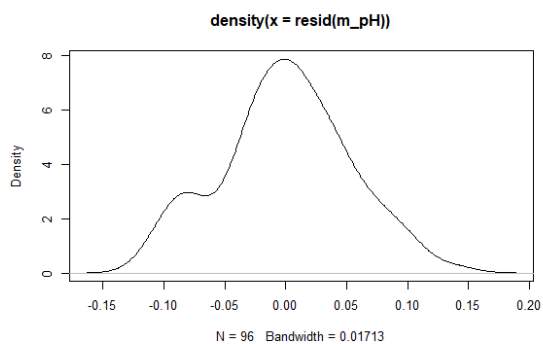
### Create model

```
m_pH <- lmer(pH ~ input * drought * species + (1|block),  
data_conditioning_inputs)
```

### Check assumptions

```
par(mfrow = c(2, 2))
```

```
plot(density(resid(m_pH)))  
hist(resid(m_pH))  
qqPlot(resid(m_pH), id = F)  
plot(resid(m_pH))
```



## Check explained variation

```
rsq.lmm(m_pH)
```

```
$model
```

```
[1] 0.5446804
```

```
$fixed
```

```
[1] 0.2078692
```

```
$random
```

```
[1] 0.3368112
```

```
ranova(m_pH)
```

ANOVA-like table for random-effects: Single term deletions

Model:

```
pH ~ input + drought + species + (1 | block) + input:drought + input:species  
+ drought:species + input:drought:species
```

	npar	logLik	AIC	LRT	Df	Pr(>Chisq)
<none>	26	76.255	-100.511			
(1   block)	25	55.562	-61.123	41.387	1	1.249e-10 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Type III ANOVA

```
anova_pH <- anova(m_pH, type = 3, ddf="Kenward-Roger")
```

```
anova_pH
```

Type III Analysis of Variance Table with Kenward-Roger's method

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)
input	0.017521	0.0058403	3	69	1.5079	0.22020
drought	0.000267	0.0002667	1	69	0.0688	0.79380
species	0.025077	0.0125385	2	69	3.2372	0.04531 *
input:drought	0.036158	0.0120528	3	69	3.1118	0.03181 *
input:species	0.013573	0.0022622	6	69	0.5840	0.74185
drought:species	0.001777	0.0008885	2	69	0.2294	0.79561
input:drought:species	0.054523	0.0090872	6	69	2.3461	0.04038 *

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

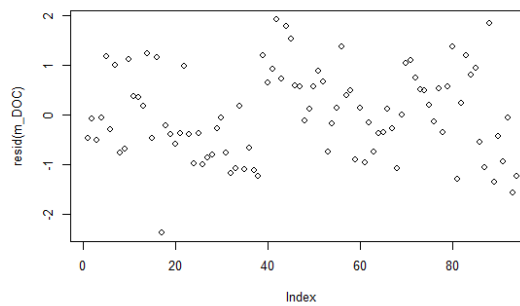
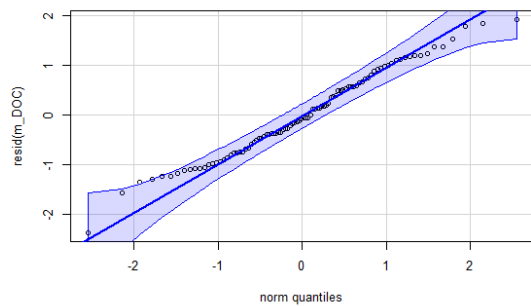
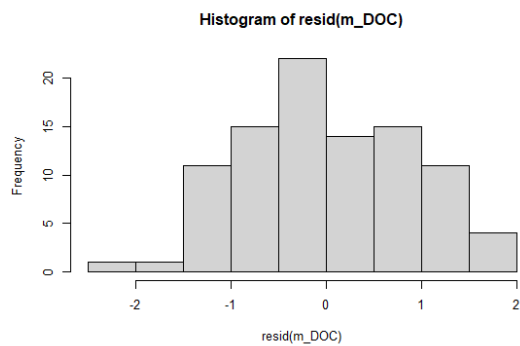
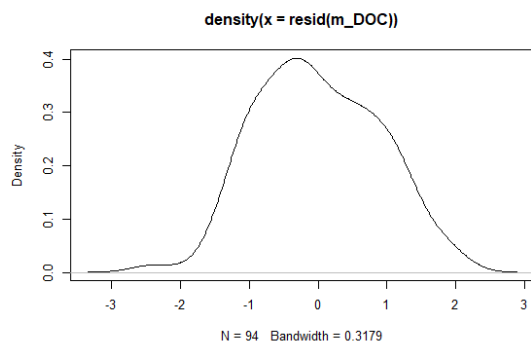
## *Dissolved organic C (DOC)*

### Create model

```
m_DOC <- lmer(DOC ~ input * drought * species + (1|block),  
data_conditioning_inputs)
```

## Check assumptions

```
par(mfrow = c(2, 2))  
  
plot(density(resid(m_DOC)))  
hist(resid(m_DOC))  
qqPlot(resid(m_DOC), id = F)  
plot(resid(m_DOC))
```



## Check explained variation

```
rsq.lmm(m_DOC)  
  
$model  
[1] 0.6074341  
  
$fixed  
[1] 0.3459202  
  
$random  
[1] 0.2615139
```

```
ranova(m_DOC)
```

ANOVA-like table for random-effects: Single term deletions

Model:

```
DOC ~ input + drought + species + (1 | block) + input:drought + input:species  
+ drought:species + input:drought:species
```

	npar	logLik	AIC	LRT	Df	Pr(>Chisq)
<none>	26	-122.54	297.08			
(1   block)	25	-141.01	332.02	36.935	1	1.222e-09 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Type III ANOVA

```
anova_DOC <- anova(m_DOC, type = 3, ddf="Kenward-Roger")  
anova_DOC
```

Type III Analysis of Variance Table with Kenward-Roger's method

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)
input	44.013	14.6710	3	67.008	13.7890	4.215e-07 ***
drought	6.607	6.6075	1	67.010	6.2102	0.01518 *
species	2.446	1.2229	2	67.004	1.1494	0.32300
input:drought	10.002	3.3339	3	67.006	3.1335	0.03119 *
input:species	7.903	1.3171	6	67.007	1.2380	0.29830
drought:species	3.525	1.7624	2	67.009	1.6564	0.19853
input:drought:species	5.237	0.8729	6	67.006	0.8204	0.55804

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

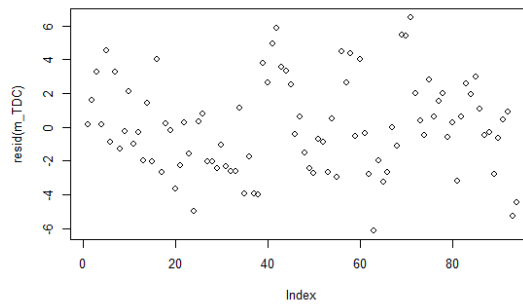
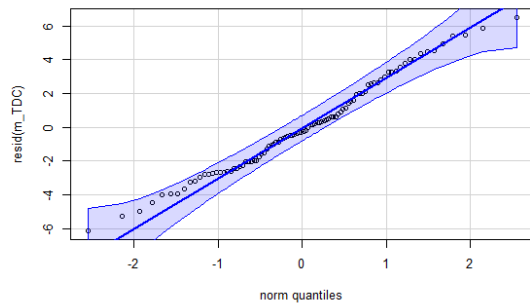
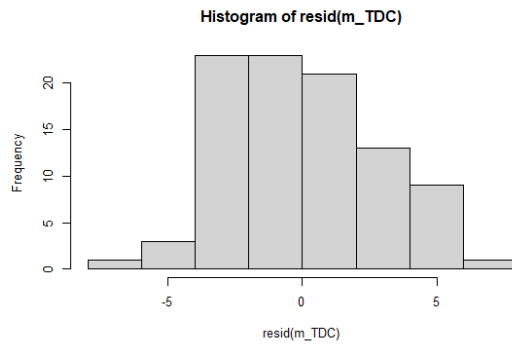
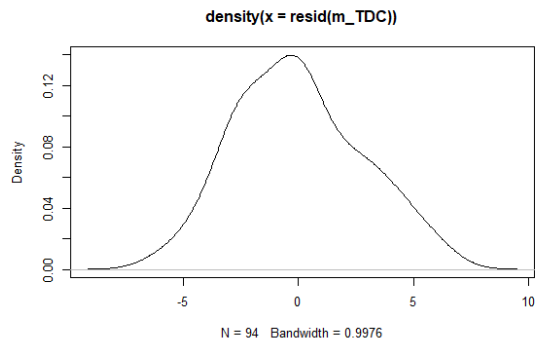
*Total dissolved C (TDC)*

Create model

```
m_TDC <- lmer(TDC ~ input * drought * species + (1|block),  
data_conditioning_inputs)
```

Check assumptions

```
par(mfrow = c(2, 2))  
  
plot(density(resid(m_TDC)))  
hist(resid(m_TDC))  
qqPlot(resid(m_TDC), id = F)  
plot(resid(m_TDC))
```



Check explained variation

```
rsq.lmm(m_TDC)
```

```
$model
```

```
[1] 0.6670625
```

```
$fixed
```

```
[1] 0.09623398
```

```
$random
```

```
[1] 0.5708285
```

```
ranova(m_TDC)
```

ANOVA-like table for random-effects: Single term deletions

Model:

```
TDC ~ input + drought + species + (1 | block) + input:drought + input:species
+ drought:species + input:drought:species
```

	npar	logLik	AIC	LRT	Df	Pr(>Chisq)
<none>	26	-203.97	459.94			
(1   block)	25	-240.31	530.62	72.674	1	< 2.2e-16 ***

```
---
```

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Type III ANOVA

```
anova_TDC <- anova(m_TDC, type = 3, ddf="Kenward-Roger")
anova_TDC
```

```
Type III Analysis of Variance Table with Kenward-Roger's method
```

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)
input	126.627	42.209	3	67.003	4.0242	0.01078 *
drought	0.001	0.001	1	67.004	0.0001	0.99301
species	6.061	3.030	2	67.002	0.2889	0.75001
input:drought	82.849	27.616	3	67.003	2.6329	0.05700 .
input:species	51.102	8.517	6	67.003	0.8120	0.56433
drought:species	9.945	4.973	2	67.004	0.4741	0.62452
input:drought:species	71.164	11.861	6	67.002	1.1308	0.35420

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## *Plant-available P (as PO<sub>4</sub>)*

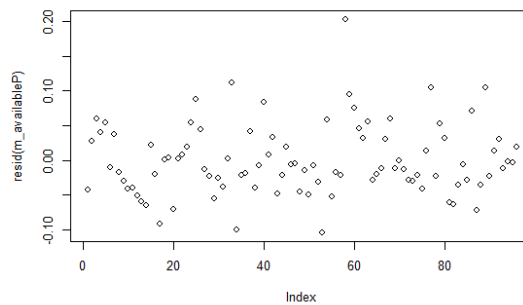
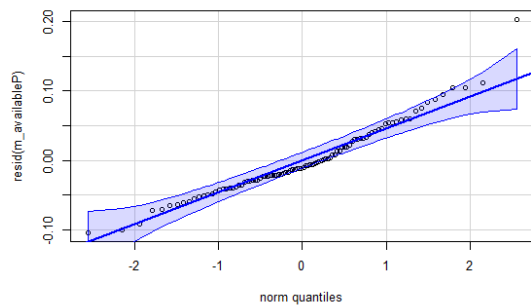
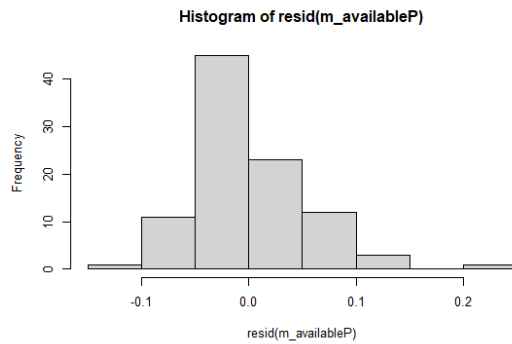
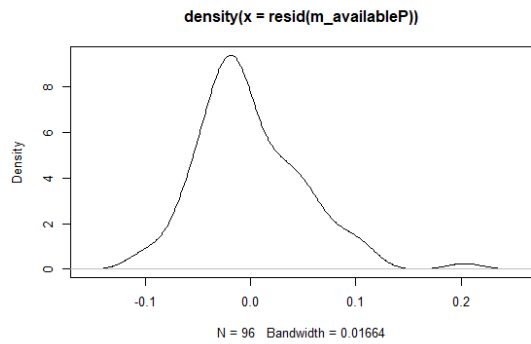
### Create model

```
m_availableP <- lmer(availableP ~ input * drought * species + (1|block),
data_conditioning_inputs)
```

### Check assumptions

```
par(mfrow = c(2, 2))

plot(density(resid(m_availableP)))
hist(resid(m_availableP))
qqPlot(resid(m_availableP), id = F)
plot(resid(m_availableP))
```



Check explained variation

```
rsq.lmm(m_availableP)
```

```
$model
```

```
[1] 0.3574953
```

```
$fixed
```

```
[1] 0.3334694
```

```
$random
```

```
[1] 0.02402591
```

```
ranova(m_availableP)
```

ANOVA-like table for random-effects: Single term deletions

Model:

```
availableP ~ input + drought + species + (1 | block) + input:drought +
input:species + drought:species + input:drought:species
```

	npar	logLik	AIC	LRT	Df	Pr(>Chisq)
<none>	26	83.811	-115.62			
(1   block)	25	83.135	-116.27	1.353	1	0.2447

Type III ANOVA

```
anova_availableP <- anova(m_availableP, type = 3, ddf="Kenward-Roger")
anova_availableP
```

### Type III Analysis of Variance Table with Kenward-Roger's method

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)
input	0.018096	0.006032	3	69	1.7380	0.167294
drought	0.032705	0.032705	1	69	9.4231	0.003063 **
species	0.005473	0.002736	2	69	0.7885	0.458598
input:drought	0.024869	0.008290	3	69	2.3885	0.076295 .
input:species	0.041675	0.006946	6	69	2.0013	0.077214 .
drought:species	0.005114	0.002557	2	69	0.7367	0.482399
input:drought:species	0.004042	0.000674	6	69	0.1941	0.977467

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

### Plant-available N

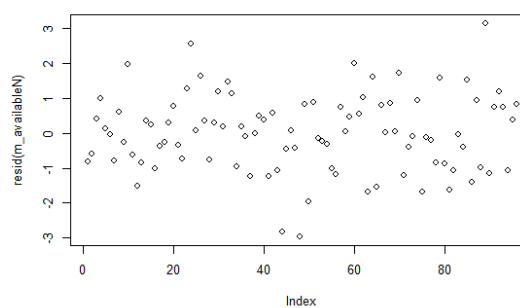
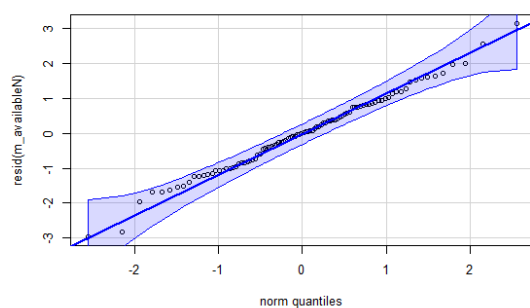
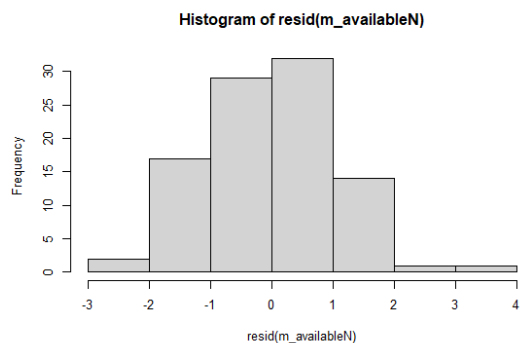
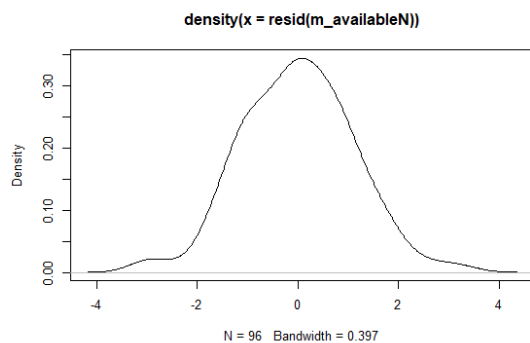
#### Create model

```
m_availableN <- lmer(availableN ~ input * drought * species + (1|block),  
data_conditioning_inputs)
```

#### Check assumptions

```
par(mfrow = c(2, 2))
```

```
plot(density(resid(m_availableN)))  
hist(resid(m_availableN))  
qqPlot(resid(m_availableN), id = F)  
plot(resid(m_availableN))
```



## Check explained variation

```
rsq.lmm(m_availableN)
```

```
$model
```

```
[1] 0.392422
```

```
$fixed
```

```
[1] 0.3161268
```

```
$random
```

```
[1] 0.07629527
```

```
ranova(m_availableN)
```

ANOVA-like table for random-effects: Single term deletions

Model:

```
availableN ~ input + drought + species + (1 | block) + input:drought +  
input:species + drought:species + input:drought:species
```

	npar	logLik	AIC	LRT	Df	Pr(>Chisq)
<none>	26	-139.35	330.70			
(1   block)	25	-142.89	335.79	7.0848	1	0.007774 **

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Type III ANOVA

```
anova_availableN <- anova(m_availableN, type = 3, ddf="Kenward-Roger")  
anova_availableN
```

Type III Analysis of Variance Table with Kenward-Roger's method

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)
input	21.2202	7.0734	3	69	4.2881	0.007807 **
drought	17.4093	17.4093	1	69	10.5540	0.001793 **
species	0.0007	0.0004	2	69	0.0002	0.999785
input:drought	1.3443	0.4481	3	69	0.2717	0.845628
input:species	13.2135	2.2022	6	69	1.3351	0.253579
drought:species	4.0116	2.0058	2	69	1.2160	0.302697
input:drought:species	7.7945	1.2991	6	69	0.7875	0.582698

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

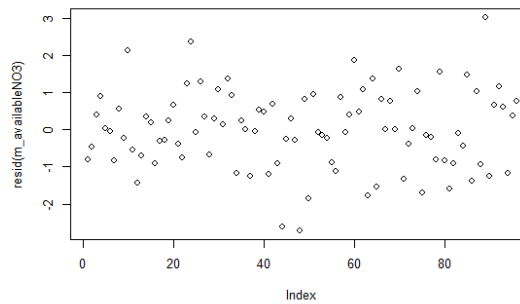
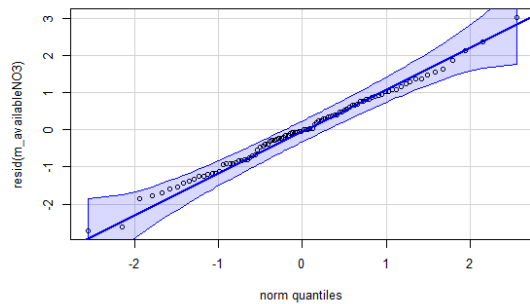
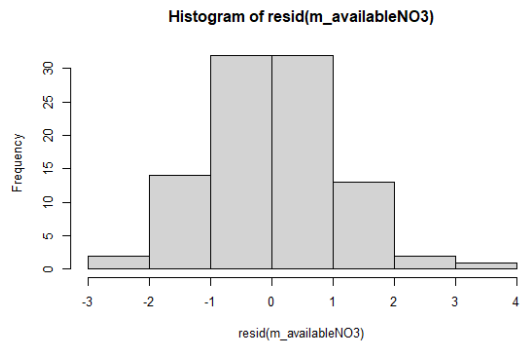
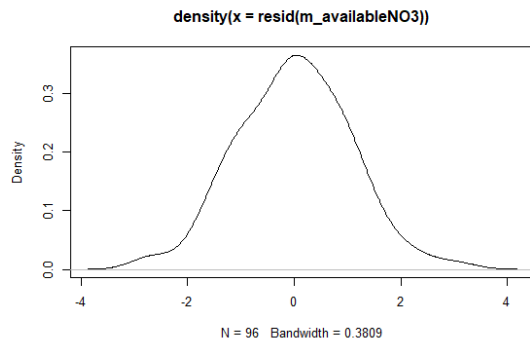
## Plant-available NO3

### Create model

```
m_availableNO3 <- lmer(availableNO3 ~ input * drought * species + (1|block),  
data_conditioning_inputs)
```

## Check assumptions

```
par(mfrow = c(2, 2))  
  
plot(density(resid(m_availableNO3)))  
hist(resid(m_availableNO3))  
qqPlot(resid(m_availableNO3), id = F)  
plot(resid(m_availableNO3))
```



## Check explained variation

```
rsq.lmm(m_availableNO3)
```

```
$model  
[1] 0.4188405
```

```
$fixed  
[1] 0.3214688
```

```
$random  
[1] 0.09737166
```

```
ranova(m_availableN03)
```

ANOVA-like table for random-effects: Single term deletions

Model:

```
availableN03 ~ input + drought + species + (1 | block) + input:drought +  
input:species + drought:species + input:drought:species
```

	npar	logLik	AIC	LRT	Df	Pr(>Chisq)
<none>	26	-136.78	325.56			
(1   block)	25	-141.75	333.49	9.9371	1	0.00162 **

---

Signif. codes: 0 '\*\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Type III ANOVA

```
anova_availableN03 <- anova(m_availableN03, type = 3, ddf="Kenward-Roger")  
anova_availableN03
```

Type III Analysis of Variance Table with Kenward-Roger's method

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)
input	22.9061	7.6354	3	69	5.0207	0.003311 **
drought	17.4944	17.4944	1	69	11.5037	0.001153 **
species	0.0206	0.0103	2	69	0.0068	0.993249
input:drought	1.5574	0.5191	3	69	0.3414	0.795483
input:species	11.6224	1.9371	6	69	1.2737	0.280866
drought:species	3.5662	1.7831	2	69	1.1725	0.315680
input:drought:species	7.3584	1.2264	6	69	0.8064	0.568405

---

Signif. codes: 0 '\*\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

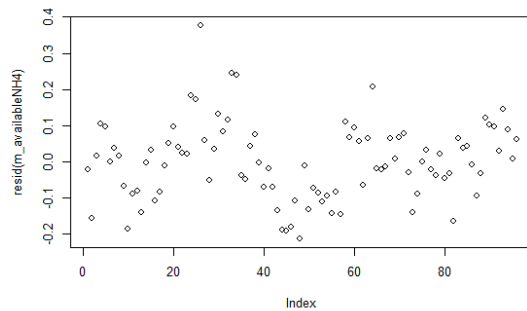
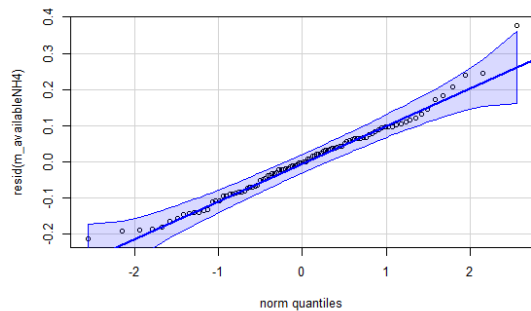
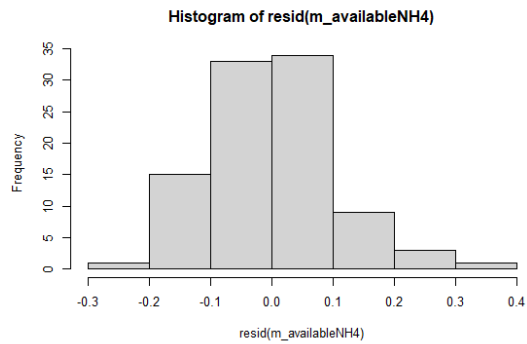
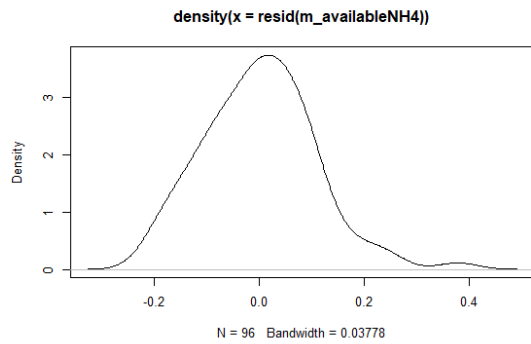
*Plant-available NH4*

Create model

```
m_availableNH4 <- lmer(availableNH4 ~ input * drought * species + (1|block),  
data_conditioning_inputs)
```

Check assumptions

```
par(mfrow = c(2, 2))  
  
plot(density(resid(m_availableNH4)))  
hist(resid(m_availableNH4))  
qqPlot(resid(m_availableNH4), id = F)  
plot(resid(m_availableNH4))
```



Check explained variation

```
rsq.lmm(m_availableNH4)
```

```
$model
[1] 0.2684351
```

```
$fixed
[1] 0.1716314
```

```
$random
[1] 0.09680362
```

```
ranova(m_availableNH4)
```

ANOVA-like table for random-effects: Single term deletions

```
Model:
availableNH4 ~ input + drought + species + (1 | block) + input:drought +
input:species + drought:species + input:drought:species
```

	npar	logLik	AIC	LRT	Df	Pr(>Chisq)
<none>	26	28.947	-5.8930			
(1   block)	25	25.174	-0.3475	7.5455	1	0.006016 **

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Type III ANOVA

```
anova_availableNH4 <- anova(m_availableNH4, type = 3, ddf="Kenward-Roger")
anova_availableNH4
```

Type III Analysis of Variance Table with Kenward-Roger's method

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)
input	0.057126	0.0190419	3	69	1.2399	0.3020
drought	0.000104	0.0001038	1	69	0.0068	0.9347
species	0.026182	0.0130909	2	69	0.8524	0.4308
input:drought	0.008284	0.0027614	3	69	0.1798	0.9098
input:species	0.095743	0.0159572	6	69	1.0391	0.4078
drought:species	0.014537	0.0072683	2	69	0.4733	0.6250
input:drought:species	0.071460	0.0119100	6	69	0.7755	0.5919

## *Total dissolved N (TDN)*

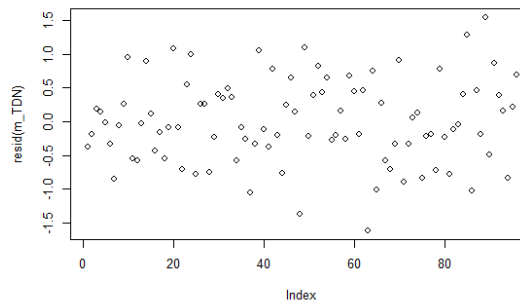
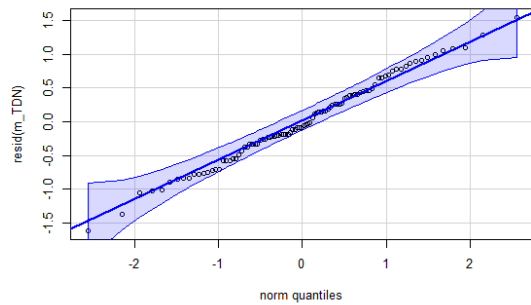
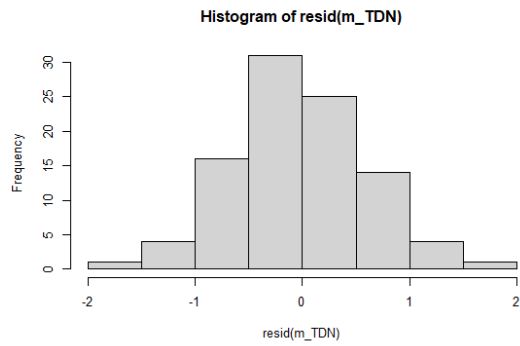
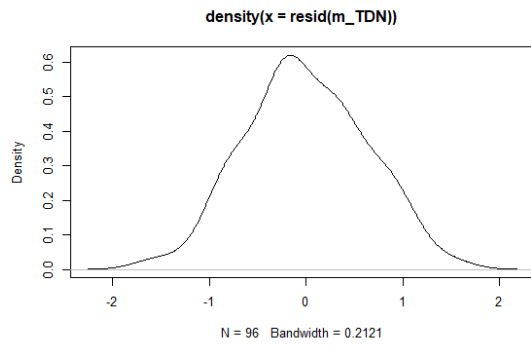
### Create model

```
m_TDN <- lmer(TDN ~ input * drought * species + (1|block),
data_conditioning_inputs)
```

### Check assumptions

```
par(mfrow = c(2, 2))

plot(density(resid(m_TDN)))
hist(resid(m_TDN))
qqPlot(resid(m_TDN), id = F)
plot(resid(m_TDN))
```



Check explained variation

```
rsq.lmm(m_TDN) # fixed: 29.5, random: 9.7
```

```
$model
```

```
[1] 0.3926743
```

```
$fixed
```

```
[1] 0.2954648
```

```
$random
```

```
[1] 0.09720945
```

```
ranova(m_TDN)
```

ANOVA-like table for random-effects: Single term deletions

```
Model:
```

```
TDN ~ input + drought + species + (1 | block) + input:drought + input:species
+ drought:species + input:drought:species
```

	npar	logLik	AIC	LRT	Df	Pr(>Chisq)
<none>	26	-98.716	249.43			
(1   block)	25	-103.432	256.86	9.4313	1	0.002133 **

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Type III ANOVA

```
anova_TDN <- anova(m_TDN, type = 3, ddf="Kenward-Roger")
anova_TDN
```

```
Type III Analysis of Variance Table with Kenward-Roger's method
```

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)	
input	9.6884	3.2295	3	69	6.1030	0.0009596	***
drought	5.7550	5.7550	1	69	10.8759	0.0015424	**
species	0.0281	0.0141	2	69	0.0266	0.9737867	
input:drought	0.4834	0.1611	3	69	0.3045	0.8220346	
input:species	1.6737	0.2790	6	69	0.5272	0.7858024	
drought:species	0.3843	0.1922	2	69	0.3632	0.6968008	
input:drought:species	1.6907	0.2818	6	69	0.5325	0.7817197	

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## *Microbial biomass N*

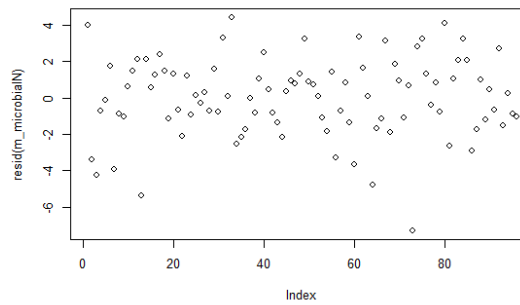
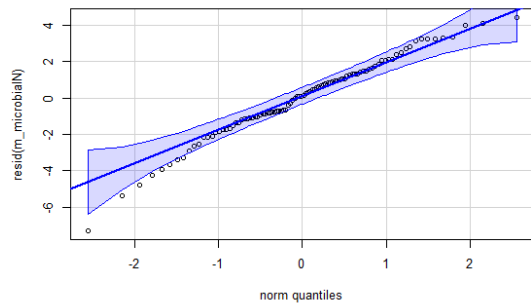
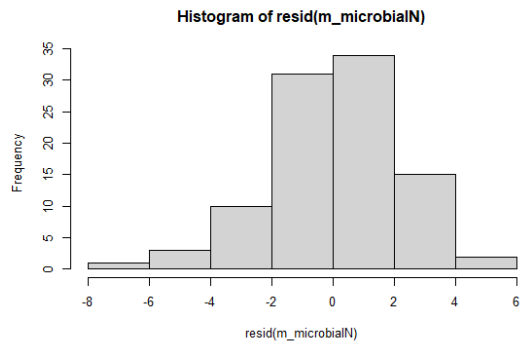
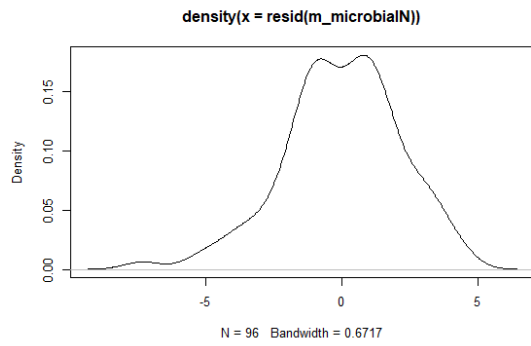
### Create model

```
m_microbialN <- lmer(microbialN ~ input * drought * species + (1|block),
data_conditioning_inputs)
```

### Check assumptions

```
par(mfrow = c(2, 2))

plot(density(resid(m_microbialN)))
hist(resid(m_microbialN))
qqPlot(resid(m_microbialN), id = F)
plot(resid(m_microbialN))
```



Check explained variation

```
rsq.lmm(m_microbialN)
```

```
$model
[1] 0.4551438
```

```
$fixed
[1] 0.3558563
```

```
$random
[1] 0.09928756
```

```
ranova(m_microbialN)
```

ANOVA-like table for random-effects: Single term deletions

Model:

```
microbialN ~ input + drought + species + (1 | block) + input:drought +
input:species + drought:species + input:drought:species
```

	npar	logLik	AIC	LRT	Df	Pr(>Chisq)
<none>	26	-188.75	429.50			
(1   block)	25	-194.21	438.43	10.923	1	0.0009496 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Type III ANOVA

```
anova_microbialN <- anova(m_microbialN, type = 3, ddf="Kenward-Roger")
anova_microbialN
```

Type III Analysis of Variance Table with Kenward-Roger's method

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)	
input	54.883	18.294	3	69	2.8480	0.0437462	*
drought	85.177	85.177	1	69	13.2598	0.0005201	***
species	4.969	2.485	2	69	0.3868	0.6806936	
input:drought	99.829	33.276	3	69	5.1802	0.0027527	**
input:species	32.060	5.343	6	69	0.8318	0.5494394	
drought:species	4.932	2.466	2	69	0.3839	0.6826372	
input:drought:species	41.295	6.883	6	69	1.0714	0.3881335	

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Pairwise Tukey test

```
pairs_microbialN <- lsmeans(m_microbialN, ~ drought | species | input)
tukey_microbialN <- as.data.frame(pairs(pairs_microbialN, adjust = "tukey"))
tukey_microbialN
```

species = Ao, input = plant:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	5.026212	1.792159	69	2.805	0.0065

species = Dg, input = plant:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	4.068129	1.792159	69	2.270	0.0263

species = Ra, input = plant:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	5.279751	1.792159	69	2.946	0.0044

species = Ao, input = inoculum:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	1.914825	1.792159	69	1.068	0.2890

species = Dg, input = inoculum:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	2.118835	1.792159	69	1.182	0.2412

species = Ra, input = inoculum:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	4.316457	1.792159	69	2.409	0.0187

species = Ao, input = exudates:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	-2.595176	1.792159	69	-1.448	0.1521

```
species = Dg, input = exudates:
contrast estimate      SE df t.ratio p.value
WW - D      0.888282 1.792159 69  0.496  0.6217
```

```
species = Ra, input = exudates:
contrast estimate      SE df t.ratio p.value
WW - D      0.885818 1.792159 69  0.494  0.6227
```

```
species = Ao, input = rootlitter:
contrast estimate      SE df t.ratio p.value
WW - D      2.838812 1.792159 69  1.584  0.1178
```

```
species = Dg, input = rootlitter:
contrast estimate      SE df t.ratio p.value
WW - D     -1.564230 1.792159 69  -0.873  0.3858
```

```
species = Ra, input = rootlitter:
contrast estimate      SE df t.ratio p.value
WW - D     -0.571052 1.792159 69  -0.319  0.7510
```

Degrees-of-freedom method: kenward-roger

### *Microbial biomass C*

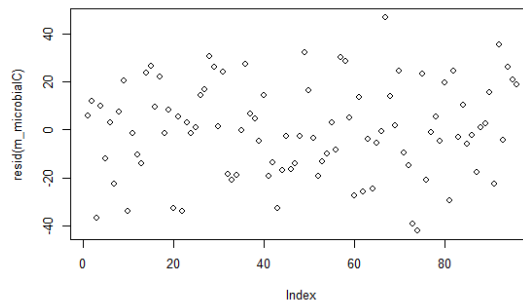
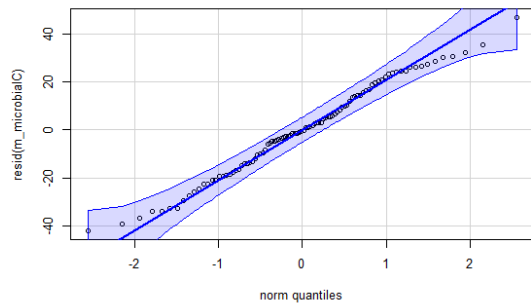
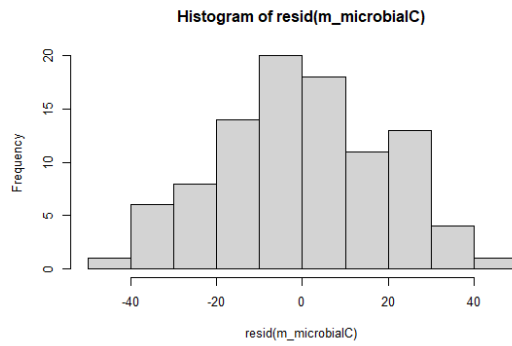
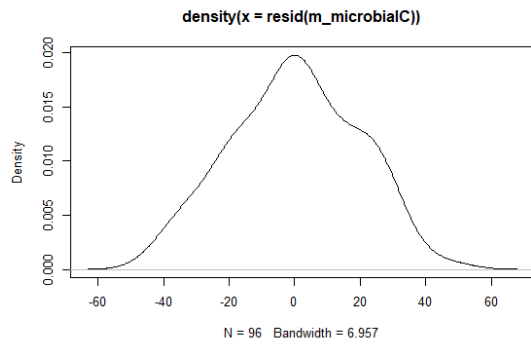
Create model

```
m_microbialC <- lmer(microbialC ~ input * drought * species + (1|block),
data_conditioning_inputs)
```

Check assumptions

```
par(mfrow = c(2, 2))

plot(density(resid(m_microbialC)))
hist(resid(m_microbialC))
qqPlot(resid(m_microbialC), id = F)
plot(resid(m_microbialC))
```



Check explained variation

```
rsq.lmm(m_microbialC)
```

```
$model
[1] 0.4730241
```

```
$fixed
[1] 0.3695376
```

```
$random
[1] 0.1034865
```

```
ranova(m_microbialC)
```

ANOVA-like table for random-effects: Single term deletions

```
Model:
microbialC ~ input + drought + species + (1 | block) + input:drought +
input:species + drought:species + input:drought:species
      npar logLik   AIC   LRT Df Pr(>Chisq)
<none>    26 -346.17 744.35
(1 | block) 25 -352.11 754.22 11.866 1 0.0005716 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Type III ANOVA

```
anova_microbialC <- anova(m_microbialC, type = 3, ddf="Kenward-Roger")
anova_microbialC
```

Type III Analysis of Variance Table with Kenward-Roger's method

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)	
input	3423.8	1141.3	3	69	2.2472	0.090524	.
drought	3870.0	3870.0	1	69	7.6202	0.007387	**
species	1278.4	639.2	2	69	1.2586	0.290479	
input:drought	5905.4	1968.5	3	69	3.8760	0.012726	*
input:species	3171.4	528.6	6	69	1.0408	0.406775	
drought:species	1969.4	984.7	2	69	1.9389	0.151621	
input:drought:species	7917.8	1319.6	6	69	2.5984	0.024967	*

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Pairwise Tukey test

```
pairs_microbialC <- lsmeans(m_microbialC, ~ drought | species | input)
tukey_microbialC <- as.data.frame(pairs(pairs_microbialC, adjust = "tukey"))
tukey_microbialC
```

species = Ao, input = plant:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	14.08132	15.93526	69	0.884	0.3799

species = Dg, input = plant:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	38.87801	15.93526	69	2.440	0.0173

species = Ra, input = plant:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	54.97021	15.93526	69	3.450	0.0010

species = Ao, input = inoculum:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	-19.64080	15.93526	69	-1.233	0.2219

species = Dg, input = inoculum:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	3.25600	15.93526	69	0.204	0.8387

species = Ra, input = inoculum:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	62.81474	15.93526	69	3.942	0.0002

species = Ao, input = exudates:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	6.40276	15.93526	69	0.402	0.6891

```
species = Dg, input = exudates:
contrast estimate      SE df t.ratio p.value
WW - D    -12.88257 15.93526 69  -0.808  0.4216
```

```
species = Ra, input = exudates:
contrast estimate      SE df t.ratio p.value
WW - D    -15.20597 15.93526 69  -0.954  0.3433
```

```
species = Ao, input = rootlitter:
contrast estimate      SE df t.ratio p.value
WW - D     17.50542 15.93526 69   1.099  0.2758
```

```
species = Dg, input = rootlitter:
contrast estimate      SE df t.ratio p.value
WW - D     3.41571 15.93526 69   0.214  0.8309
```

```
species = Ra, input = rootlitter:
contrast estimate      SE df t.ratio p.value
WW - D    -1.21321 15.93526 69  -0.076  0.9395
```

Degrees-of-freedom method: kenward-roger

## Feedback phase

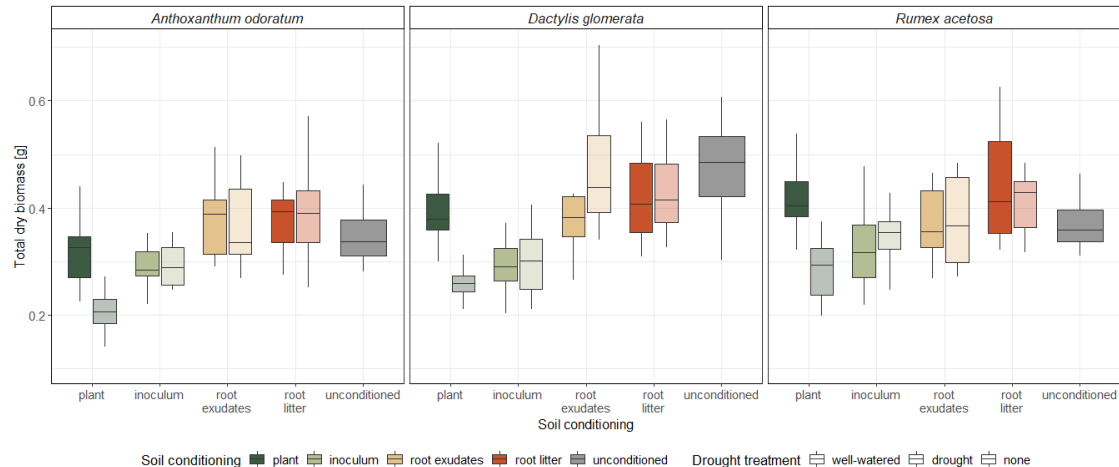
This part of the data analysis and visualization contains all Plant biomass and PSF data from the end of the feedback phase.

### Data visualization

#### Plant biomass

Total dry biomass (by species, soil conditioning and drought)

```
P_totalDW_feedback <- ggplot(data=data_feedback, mapping = aes(x = input, y =
totalDW, fill = input, alpha = drought)) +
  geom_boxplot( size = 0, outlier.shape = NA)+
  geom_boxplot( size = 0.5, outlier.shape = NA, fatten = 1)+
  scale_fill_manual(values = InputPalette, name = "Soil conditioning", labels
= c("plant", "inoculum", "root exudates", "root litter", "unconditioned"))+
  scale_alpha_manual(values = c(1, 0.2, 1), name = "Drought treatment",
labels = c("well-watered", "drought", "none"))+
  labs(x = "Soil conditioning", y = "Total dry biomass [g]")+
  PSFtheme+ theme(strip.text = element_text(face="italic", size = 13),
axis.line = element_line(linewidth = 0))+
  facet_wrap(facets = "species_home", nrow = 1, labeller = facets_species)+
  scale_x_discrete(labels = c("plant", "inoculum", "root\nexudates",
"root\nlitter", "unconditioned"))+
  guides(fill = guide_legend(order=1), alpha = guide_legend(order=2))
P_totalDW_feedback
```



Save figure

```
ggsave(plot = P_totalDW_feedback, "totalDW_feedback.tiff", width = 25, height = 15, units = "cm", dpi = 1000)
```

Total dry biomass (by species, soil conditioning, drought and conditioning species)

Here, plant dry biomass across all 25 types of soil conditioning is shown for the three species. In order to do that, we need to introduce a new variable as a combination of conditioning species, soil conditioning (input) and drought. This variable is sorted into the right order and then used to separate the data for the figure.

```
data_feedback$soil_type <- paste0(data_feedback$input, "_",
data_feedback$species_soil, "_", data_feedback$drought)

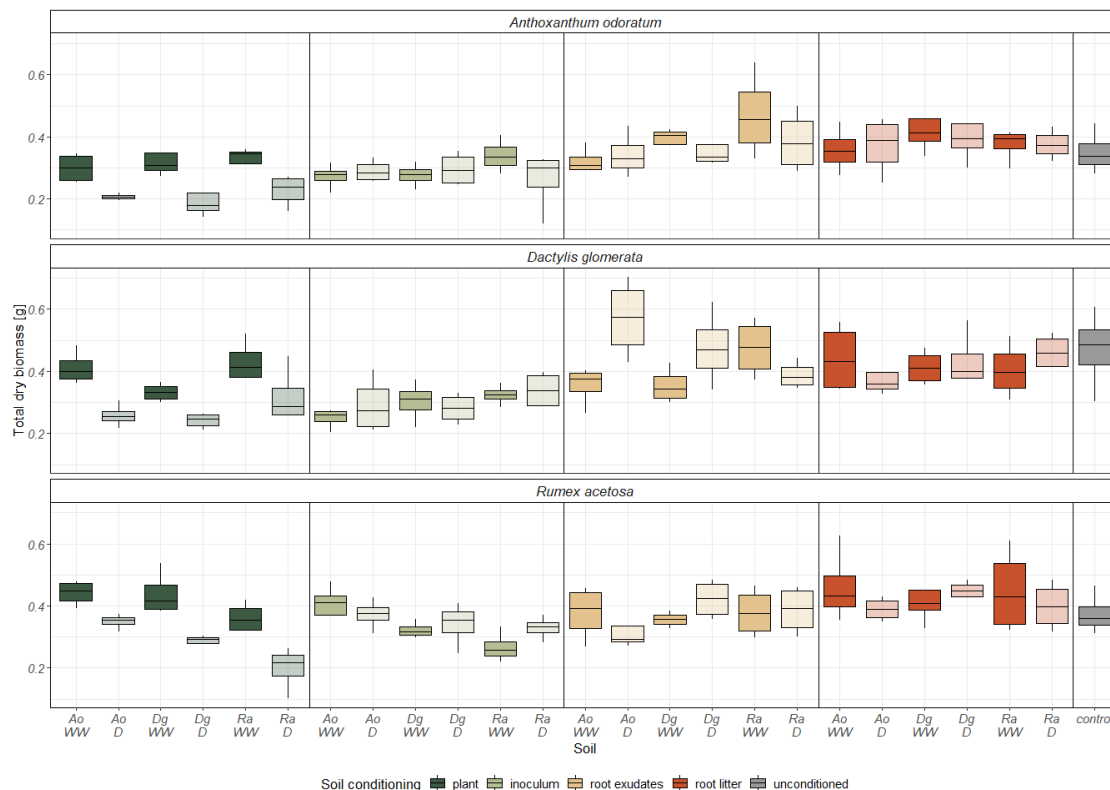
data_feedback <- mutate(data_feedback, soil_type = replace(soil_type,
soil_type == "none_none_none", "unconditioned"))

data_feedback$soil_type <- factor(data_feedback$soil_type,
c ("plant_Ao_WW", "plant_Ao_D", "plant_Dg_WW", "plant_Dg_D", "plant_Ra_WW",
"plant_Ra_D", "inoculum_Ao_WW", "inoculum_Ao_D", "inoculum_Dg_WW",
"inoculum_Dg_D", "inoculum_Ra_WW", "inoculum_Ra_D", "exudates_Ao_WW",
"exudates_Ao_D", "exudates_Dg_WW", "exudates_Dg_D", "exudates_Ra_WW",
"exudates_Ra_D", "rootlitter_Ao_WW", "rootlitter_Ao_D", "rootlitter_Dg_WW",
"rootlitter_Dg_D", "rootlitter_Ra_WW", "rootlitter_Ra_D", "unconditioned"))
```

```
P_totalDW_feedback_all <- ggplot(data_feedback, aes(x = soil_type, y =
totalDW, alpha = drought, fill = input))+
  geom_boxplot(col = "black", size = 0.3, outlier.shape = NA, fatten = 1)+
  scale_fill_manual(values = InputPalette, name = "Soil conditioning", labels
= c("plant", "inoculum", "root exudates", "root litter", "unconditioned"))+
  geom_vline(xintercept = c(6.5, 12.5, 18.5, 24.5), linewidth = 0.3)+
  scale_alpha_manual(values = c(1, 0.3, 1), name = "Drought treatment",
labels = c("well-watered", "drought", "none"))+
  facet_wrap(facets = "species_home", nrow = 3, labeller = facets_species)+
  PSFtheme+ theme(strip.text = element_text(face="italic", size = 13),
axis.line = element_line(linewidth = 0))+
  biomasstheme+theme(strip.text = element_text(face="italic", size = 13))+
  scale_x_discrete(labels = c(rep(c("Ao\nWW", "Ao\nD", "Dg\nWW", "Dg\nD",
"Ra\nWW", "Ra\nD"), 4), "control" ))+
  labs(x = "Soil", y = "Total dry biomass [g]")+
  guides(fill = guide_legend(order = 1), alpha = FALSE)
```

Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as of ggplot2 3.3.4.

P\_totalDW\_feedback\_all



Save figure

```
ggsave(plot = P_totalDW_feedback_all, "totalDW_feedback_all.tiff", width =
25, height = 15, units = "cm", dpi = 1000)
```

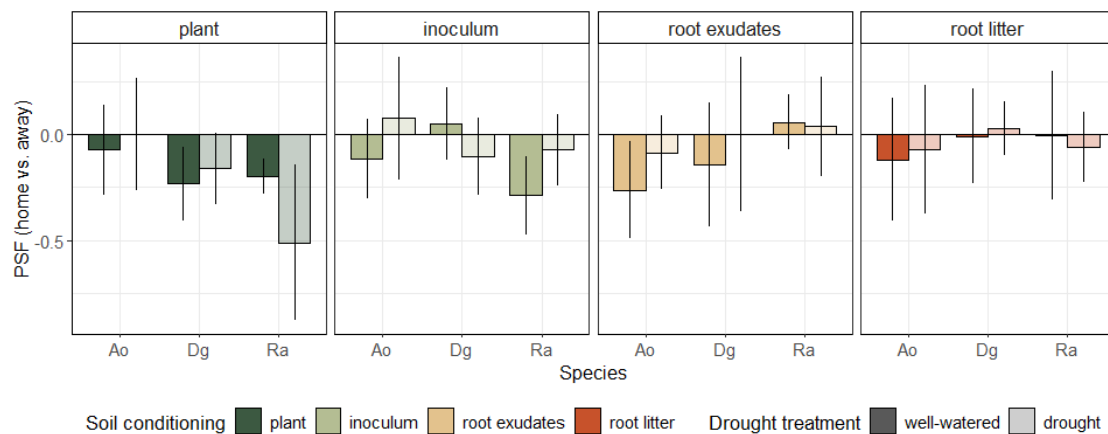
## Plant-soil feedback (PSF)

Filter for PSF home vs. away and summarize statistics

```
summary_PSFaway <- do.call(data.frame,aggregate(PSFtotal ~ species_home *  
input * drought , FUN = function(x) c(mean = mean(x),median = median(x), sd =  
sd(x), se = se(x), conint = conint(x)), data = filter(data_PSF, PSF_type ==  
"away")))
```

Bar graph PSFaway by species, soil conditioning and drought treatment

```
P_PSFaway <- ggplot(data=summary_PSFaway, mapping = aes(x = species_home, y =  
PSFtotal.mean, fill = input, alpha = drought)) +  
  geom_col(position = "dodge", colour = "black", width = 0.8, linewidth =  
0.3)+  
  geom_errorbar(alpha = 1, aes(ymin = PSFtotal.mean - PSFtotal.conint, ymax =  
PSFtotal.mean + PSFtotal.conint, col = drought), position  
=position_dodge(0.8), width = 0, linewidth = 0.3)+  
  geom_hline(yintercept = 0)+  
  scale_fill_manual(values = InputPalette, name = "Soil conditioning", labels  
= c("plant", "inoculum", "root exudates", "root litter"))+  
  scale_alpha_manual(values = c(1, 0.3), name = "Drought treatment", labels =  
c("well-watered", "drought"))+  
  scale_color_manual(values = c("black", "black"), name =  
"Drought\treatment", labels = c("well-watered", "drought"))+  
  labs(x = "Species", y = "PSF (home vs. away)")+  
  PSFtheme+  
  theme(axis.line = element_line(linewidth = 0))+  
  facet_wrap(facets = "input", nrow = 1, labeller = facets_input)+  
  #ylim(-1.6,0.8)+  
  guides(color = FALSE, fill = guide_legend(order = 1), alpha =  
guide_legend(order = 2))  
P_PSFaway
```



Save figure

```
ggsave(plot = P_PSFaway, "PSFaway.tiff", width = 30, height = 10, units = "cm", dpi = 1000)
```

## Statistics

*Plant biomass*

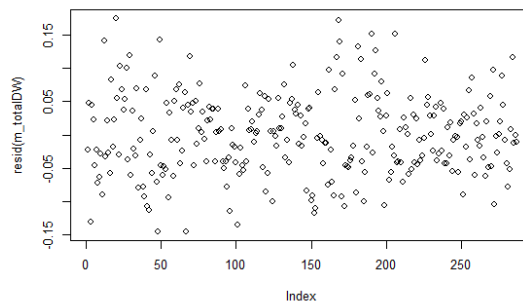
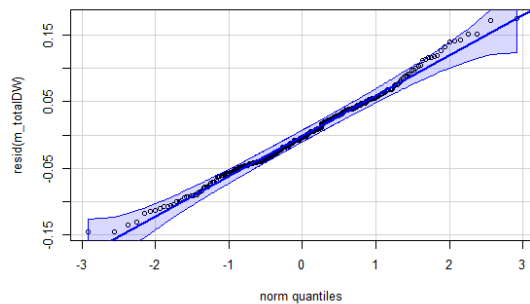
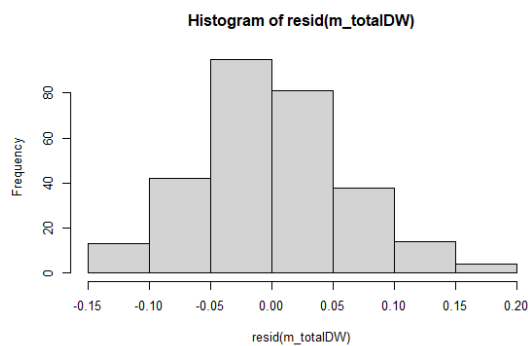
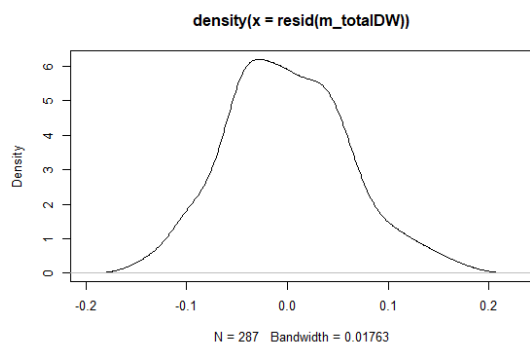
Excluding unconditioned soil

Build model

```
m_totalDW <- lmer(totalDW ~ species_home * input * drought * species_soil +  
(1|block), data = filter(data_feedback, input != "none"))
```

Check assumptions

```
par(mfrow = c(2, 2))  
  
plot(density(resid(m_totalDW)))  
hist(resid(m_totalDW))  
qqPlot(resid(m_totalDW), id = F)  
plot(resid(m_totalDW))
```



## Check explained variation

```
rsq.lmm(m_totalDW)
```

```
$model
```

```
[1] 0.5950077
```

```
$fixed
```

```
[1] 0.5880968
```

```
$random
```

```
[1] 0.006910869
```

```
ranova(m_totalDW)
```

ANOVA-like table for random-effects: Single term deletions

Model:

```
totalDW ~ species_home + input + drought + species_soil + (1 | block) +  
species_home:input + species_home:drought + input:drought +  
species_home:species_soil + input:species_soil + drought:species_soil +  
species_home:input:drought + species_home:input:species_soil +  
species_home:drought:species_soil + input:drought:species_soil +  
species_home:input:drought:species_soil
```

	npar	logLik	AIC	LRT	Df	Pr(>Chisq)
<none>	74	214.20	-280.39			
(1   block)	73	213.05	-280.09	2.3006	1	0.1293

ANOVA

```
anova(m_totalDW, type = 3, ddf = "Kenward-Roger")
```

Type III Analysis of Variance Table with Kenward-Roger's method

	Sum Sq	Mean Sq	NumDF	DenDF	F value
species_home	0.11529	0.057643	2	212.01	11.6387
input	0.63755	0.212516	3	212.01	42.9094
drought	0.04619	0.046185	1	212.01	9.3253
species_soil	0.00163	0.000813	2	212.01	0.1642
species_home:input	0.08954	0.014924	6	212.01	3.0133
species_home:drought	0.01642	0.008208	2	212.01	1.6573
input:drought	0.22328	0.074426	3	212.01	15.0274
species_home:species_soil	0.08016	0.020040	4	212.01	4.0464
input:species_soil	0.02903	0.004838	6	212.01	0.9769
drought:species_soil	0.01184	0.005918	2	212.01	1.1948
species_home:input:drought	0.03564	0.005941	6	212.01	1.1995
species_home:input:species_soil	0.09678	0.008065	12	212.01	1.6285
species_home:drought:species_soil	0.01669	0.004172	4	212.01	0.8423
input:drought:species_soil	0.04976	0.008294	6	212.01	1.6746
species_home:input:drought:species_soil	0.10412	0.008676	12	212.01	1.7518
		Pr(>F)			
species_home		1.600e-05			***

```

input < 2.2e-16 ***
drought 0.002550 **
species_soil 0.848687
species_home:input 0.007574 **
species_home:drought 0.193122
input:drought 6.634e-09 ***
species_home:species_soil 0.003495 **
input:species_soil 0.441788
drought:species_soil 0.304789
species_home:input:drought 0.307748
species_home:input:species_soil 0.085480 .
species_home:drought:species_soil 0.499707
input:drought:species_soil 0.128548
species_home:input:drought:species_soil 0.058024 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

### Pairwise Tukey test

```
pairs_totalDW<- lsmeans(m_totalDW, ~ drought | species_home | input)
```

NOTE: Results may be misleading due to involvement in interactions

```
tukey_totalDW <- as.data.frame(pairs(pairs_totalDW, adjust = "tukey"))
tukey_totalDW
```

```
species_home = Ao, input = plant:
contrast estimate SE df t.ratio p.value
WW - D 0.10489167 0.02873057 212.00 3.651 0.0003
```

```
species_home = Dg, input = plant:
contrast estimate SE df t.ratio p.value
WW - D 0.11879833 0.02873057 212.00 4.135 0.0001
```

```
species_home = Ra, input = plant:
contrast estimate SE df t.ratio p.value
WW - D 0.13685833 0.02873057 212.00 4.764 <.0001
```

```
species_home = Ao, input = inoculum:
contrast estimate SE df t.ratio p.value
WW - D 0.01419167 0.02873057 212.00 0.494 0.6218
```

```
species_home = Dg, input = inoculum:
contrast estimate SE df t.ratio p.value
WW - D -0.01187500 0.02873057 212.00 -0.413 0.6798
```

```
species_home = Ra, input = inoculum:
contrast estimate SE df t.ratio p.value
WW - D -0.02105000 0.02873057 212.00 -0.733 0.4646
```

```
species_home = Ao, input = exudates:
contrast      estimate          SE      df t.ratio p.value
WW - D        0.02915000 0.02873057 212.00   1.015  0.3115
```

```
species_home = Dg, input = exudates:
contrast      estimate          SE      df t.ratio p.value
WW - D        -0.08443333 0.02873057 212.00  -2.939  0.0037
```

```
species_home = Ra, input = exudates:
contrast      estimate          SE      df t.ratio p.value
WW - D        -0.00895915 0.02953013 212.11  -0.303  0.7619
```

```
species_home = Ao, input = rootlitter:
contrast      estimate          SE      df t.ratio p.value
WW - D         0.00133333 0.02873057 212.00   0.046  0.9630
```

```
species_home = Dg, input = rootlitter:
contrast      estimate          SE      df t.ratio p.value
WW - D        -0.00620833 0.02873057 212.00  -0.216  0.8291
```

```
species_home = Ra, input = rootlitter:
contrast      estimate          SE      df t.ratio p.value
WW - D         0.03194167 0.02873057 212.00   1.112  0.2675
```

Results are averaged over the levels of: species\_soil  
Degrees-of-freedom method: kenward-roger

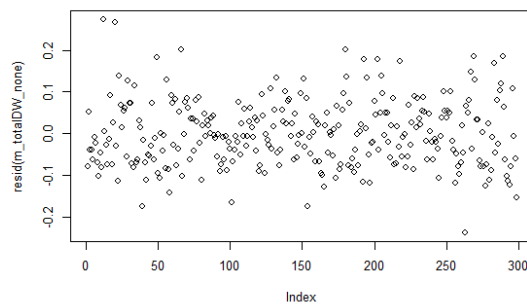
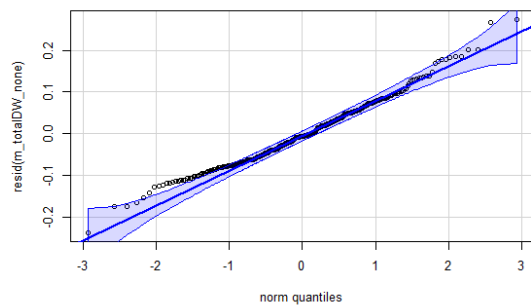
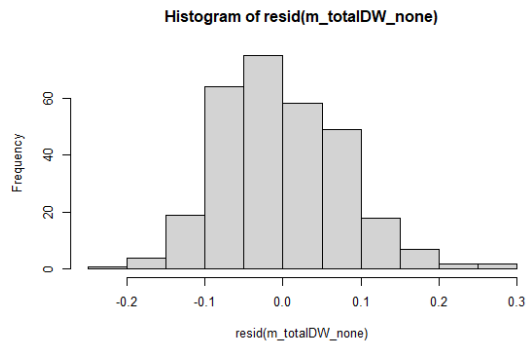
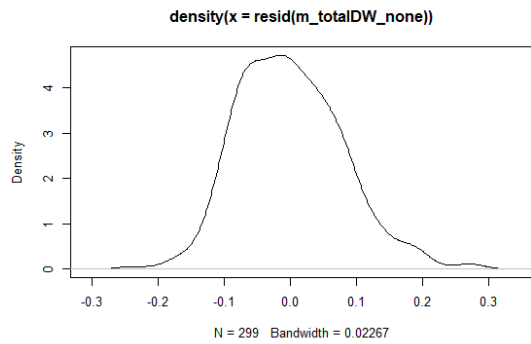
### Including unconditioned soil

#### Build model

```
m_totalDW_none <- lmer(totalDW ~ species_home * input + (1|block), data =  
data_feedback)
```

#### Check assumptions

```
par(mfrow = c(2, 2))  
  
plot(density(resid(m_totalDW_none)))  
hist(resid(m_totalDW_none))  
qqPlot(resid(m_totalDW_none), id = F)  
plot(resid(m_totalDW_none))
```



Check explained variation

```
rsq.lmm(m_totalDW_none)
```

```
$model
```

```
[1] 0.3323145
```

```
$fixed
```

```
[1] 0.3232934
```

```
$random
```

```
[1] 0.009021029
```

```
ranova(m_totalDW_none)
```

ANOVA-like table for random-effects: Single term deletions

Model:

```
totalDW ~ species_home + input + (1 | block) + species_home:input
```

	npar	logLik	AIC	LRT	Df	Pr(>Chisq)
<none>	17	288.94	-543.89			
(1   block)	16	288.44	-544.88	1.0072	1	0.3156

## ANOVA

```
anova(m_totalDW_none, type = 3, ddf = "Kenward-Roger")
```

Type III Analysis of Variance Table with Kenward-Roger's method

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)	
species_home	0.10555	0.052776	2	281.00	8.0643	0.0003931	***
input	0.65831	0.164577	4	281.01	25.1478	< 2.2e-16	***
species_home:input	0.10847	0.013559	8	281.01	2.0718	0.0386115	*

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Pairwise Tukey test

```
pairs_totalDW_none <- lsmeans(m_totalDW_none, ~ input | species_home)
tukey_totalDW_none <- as.data.frame(pairs(pairs_totalDW_none, adjust =
"tukey"))
```

```
tukey_totalDW_none
```

species\_home = Ao:

contrast	estimate	SE	df	t.ratio	p.value
plant - inoculum	-0.02521667	0.02335308	281.00	-1.080	0.8168
plant - exudates	-0.11276250	0.02335308	281.00	-4.829	<.0001
plant - rootlitter	-0.12330417	0.02335308	281.00	-5.280	<.0001
plant - none	-0.08592083	0.04368962	281.00	-1.967	0.2851
inoculum - exudates	-0.08754583	0.02335308	281.00	-3.749	0.0020
inoculum - rootlitter	-0.09808750	0.02335308	281.00	-4.200	0.0003
inoculum - none	-0.06070417	0.04368962	281.00	-1.389	0.6350
exudates - rootlitter	-0.01054167	0.02335308	281.00	-0.451	0.9914
exudates - none	0.02684167	0.04368962	281.00	0.614	0.9727
rootlitter - none	0.03738333	0.04368962	281.00	0.856	0.9127

species\_home = Dg:

contrast	estimate	SE	df	t.ratio	p.value
plant - inoculum	0.03433000	0.02335308	281.00	1.470	0.5828
plant - exudates	-0.10399083	0.02335308	281.00	-4.453	0.0001
plant - rootlitter	-0.09052833	0.02335308	281.00	-3.877	0.0012
plant - none	-0.13806583	0.04368962	281.00	-3.160	0.0150
inoculum - exudates	-0.13832083	0.02335308	281.00	-5.923	<.0001
inoculum - rootlitter	-0.12485833	0.02335308	281.00	-5.347	<.0001
inoculum - none	-0.17239583	0.04368962	281.00	-3.946	0.0009
exudates - rootlitter	0.01346250	0.02335308	281.00	0.576	0.9784
exudates - none	-0.03407500	0.04368962	281.00	-0.780	0.9363
rootlitter - none	-0.04753750	0.04368962	281.00	-1.088	0.8126

```
species_home = Ra:
contrast          estimate          SE      df t.ratio p.value
plant - inoculum   0.00810417 0.02335308 281.00   0.347 0.9969
plant - exudates  -0.02941799 0.02360864 281.06  -1.246 0.7243
plant - rootlitter -0.08378333 0.02335308 281.00  -3.588 0.0036
plant - none       -0.02782083 0.04368962 281.00  -0.637 0.9689
inoculum - exudates -0.03752215 0.02360864 281.06  -1.589 0.5055
inoculum - rootlitter -0.09188750 0.02335308 281.00  -3.935 0.0010
inoculum - none    -0.03592500 0.04368962 281.00  -0.822 0.9236
exudates - rootlitter -0.05436535 0.02360864 281.06  -2.303 0.1470
exudates - none    0.00159715 0.04382675 281.02   0.036 1.0000
rootlitter - none   0.05596250 0.04368962 281.00   1.281 0.7032
```

Degrees-of-freedom method: kenward-roger

P value adjustment: tukey method for comparing a family of 5 estimates

### *Plant-soil feedback*

#### Build model

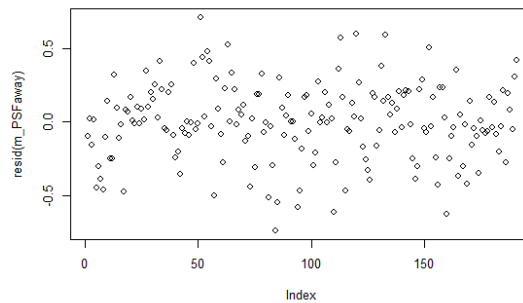
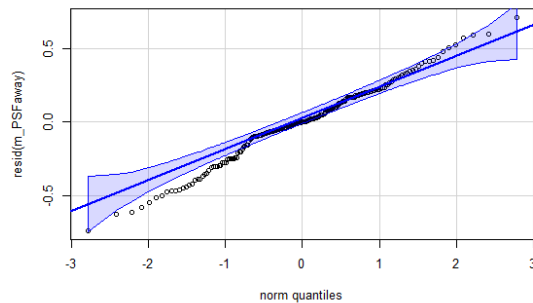
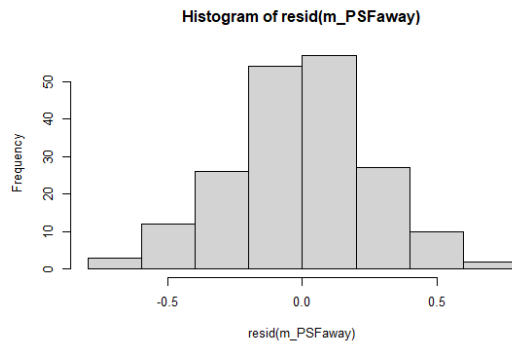
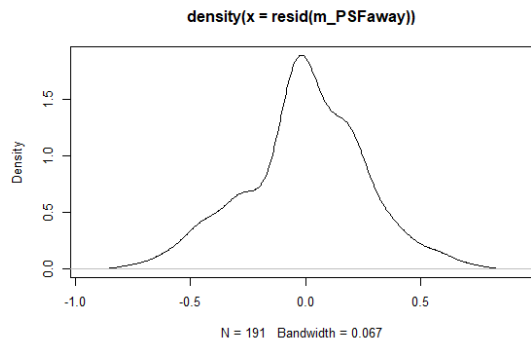
```
m_PSFaway <- lmer(PSFtotal ~ species_home * input * drought + (1|block) +
(1|species_away), data = filter(data_PSF, PSF_type == "away"))
```

boundary (singular) fit: see help('isSingular')

#### Check assumptions

```
par(mfrow = c(2, 2))

plot(density(resid(m_PSFaway)))
hist(resid(m_PSFaway))
qqPlot(resid(m_PSFaway), id = F)
plot(resid(m_PSFaway))
```



Check explained variation

```
rsq.lmm(m_PSFaway)
```

```
$model
```

```
[1] 0.208333
```

```
$fixed
```

```
[1] 0.2050207
```

```
$random
```

```
[1] 0.003312339
```

```
ranova(m_PSFaway)
```

```
boundary (singular) fit: see help('isSingular')
```

```
ANOVA-like table for random-effects: Single term deletions
```

```
Model:
```

```
PSFtotal ~ species_home + input + drought + (1 | block) + (1 | species_away)
+ species_home:input + species_home:drought + input:drought +
species_home:input:drought
```

	npar	logLik	AIC	LRT	Df	Pr(>Chisq)
<none>	27	-47.933	149.87			
(1   block)	26	-47.968	147.94	0.069842	1	0.7916
(1   species_away)	26	-47.933	147.87	0.000000	1	1.0000

## ANOVA

```
anova(m_PSFaway, type = 3, ddf = "Kenward-Roger")
```

Type III Analysis of Variance Table with Kenward-Roger's method

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)
species_home	0.07355	0.036777	2	17.14	0.4784	0.62780
input	0.69511	0.231703	3	162.05	3.0142	0.03169 *
drought	0.05688	0.056879	1	162.05	0.7399	0.39095
species_home:input	1.40088	0.233481	6	162.05	3.0373	0.00767 **
species_home:drought	0.22119	0.110593	2	162.05	1.4387	0.24025
input:drought	0.18754	0.062512	3	162.05	0.8132	0.48828
species_home:input:drought	0.62987	0.104978	6	162.05	1.3656	0.23149

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Pairwise Tukey test

```
tukey_PSFaway <- lsmeans(m_PSFaway, ~ drought | species_home | input)
pairs_tukey_PSFaway <- as.data.frame(pairs(tukey_PSFaway, adjust = "tukey"))
# soil-Ra differs
pairs_tukey_PSFaway
```

species\_home = Ao, input = plant:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	-0.07330988	0.1386276	162.00	-0.529	0.5976

species\_home = Dg, input = plant:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	-0.07151014	0.1386276	162.00	-0.516	0.6067

species\_home = Ra, input = plant:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	0.31447300	0.1386276	162.00	2.268	0.0246

species\_home = Ao, input = inoculum:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	-0.19151864	0.1386276	162.00	-1.382	0.1690

species\_home = Dg, input = inoculum:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	0.15181456	0.1386276	162.00	1.095	0.2751

species\_home = Ra, input = inoculum:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	-0.21595504	0.1386276	162.00	-1.558	0.1212

species\_home = Ao, input = exudates:

contrast	estimate	SE	df	t.ratio	p.value
WW - D	-0.17516904	0.1386276	162.00	-1.264	0.2082

```
species_home = Dg, input = exudates:
contrast      estimate          SE      df t.ratio p.value
WW - D      -0.14477564 0.1386276 162.00  -1.044  0.2979
```

```
species_home = Ra, input = exudates:
contrast      estimate          SE      df t.ratio p.value
WW - D       0.02074541 0.1437194 162.56   0.144  0.8854
```

```
species_home = Ao, input = rootlitter:
contrast      estimate          SE      df t.ratio p.value
WW - D      -0.04861563 0.1386276 162.00  -0.351  0.7263
```

```
species_home = Dg, input = rootlitter:
contrast      estimate          SE      df t.ratio p.value
WW - D      -0.03759736 0.1386276 162.00  -0.271  0.7866
```

```
species_home = Ra, input = rootlitter:
contrast      estimate          SE      df t.ratio p.value
WW - D       0.05704911 0.1386276 162.00   0.412  0.6812
```

Degrees-of-freedom method: kenward-roger

### *Correlations PSF between soil conditioning types*

Create a new data frame for each soil conditioning type

```
plant <- filter(data_PSF, input == "plant") %>% dplyr::select(!input)
inoculum <- filter(data_PSF, input == "inoculum") %>% dplyr::select(!input)
exudates <- filter(data_PSF, input == "exudates") %>% dplyr::select(!input)
rootlitter <- filter(data_PSF, input == "rootlitter") %>%
dplyr::select(!input)
```

Fuse data frames adding the extension of their soil conditioning to each column

```
PSF_cor <- full_join(plant, inoculum, by = c("block", "drought", "pair",
"species_home", "species_away", "PSF_type"), suffix = c("_plant",
"_inoculum"))
PSF_cor <- full_join(PSF_cor, exudates, by = c("block", "drought", "pair",
"species_home", "species_away", "PSF_type"))
PSF_cor <- full_join(PSF_cor, rootlitter, by = c("block", "drought", "pair",
"species_home", "species_away", "PSF_type"), suffix = c("_exudates",
"_rootlitter"))
```

Filter to only keep PSF away

```
PSF_cor_away <- filter(PSF_cor, PSF_type == "away")
```

Determine regression line and correlation coefficients in mixed effects models

plant vs. inoculum

```
m_cor_inoculum <- lmer( PSFtotal_plant ~ PSFtotal_inoculum + (1|drought) +  
(1|block) , PSF_cor_away)
```

boundary (singular) fit: see help('isSingular')

```
anova(m_cor_inoculum , type = 3, ddf = "Kenward-Roger")
```

Type III Analysis of Variance Table with Kenward-Roger's method

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)
PSFtotal_inoculum	0.040685	0.040685	1	45.899	0.4294	0.5156

```
rsq.lmm(m_cor_inoculum )
```

\$model

```
[1] 0.01113182
```

\$fixed

```
[1] 0.01113182
```

\$random

```
[1] 0
```

```
cor.test(PSF_cor_away$PSFtotal_plant, PSF_cor_away$PSFtotal_inoculum)
```

Pearson's product-moment correlation

data: PSF\_cor\_away\$PSFtotal\_plant and PSF\_cor\_away\$PSFtotal\_inoculum

t = 0.7196, df = 46, p-value = 0.4754

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

-0.1841477 0.3783013

sample estimates:

cor

0.1055074

plant vs. exudates

```
m_cor_exudates <- lmer( PSFtotal_plant ~ PSFtotal_exudates + (1|drought) +  
(1|block) , PSF_cor_away)
```

boundary (singular) fit: see help('isSingular')

```
anova(m_cor_exudates , type = 3, ddf = "Kenward-Roger")
```

Type III Analysis of Variance Table with Kenward-Roger's method

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)
PSFtotal_exudates	0.059464	0.059464	1	41.936	0.618	0.4362

```
rsq.lmm(m_cor_exudates )

$model
[1] 0.0176439

$fixed
[1] 0.0176439

$random
[1] 0

cor.test(PSF_cor_away$PSFtotal_plant, PSF_cor_away$PSFtotal_exudates)

Pearson's product-moment correlation

data: PSF_cor_away$PSFtotal_plant and PSF_cor_away$PSFtotal_exudates
t = -0.89902, df = 45, p-value = 0.3734
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.4045652  0.1604570
sample estimates:
      cor
-0.1328303
```

plant vs. root litter

```
m_cor_rootlitter <- lmer( PSFtotal_plant ~ PSFtotal_rootlitter + (1|drought)
+ (1|block) , PSF_cor_away)

boundary (singular) fit: see help('isSingular')

anova(m_cor_rootlitter , type = 3, ddf = "Kenward-Roger")

Type III Analysis of Variance Table with Kenward-Roger's method
              Sum Sq   Mean Sq NumDF  DenDF F value Pr(>F)
PSFtotal_rootlitter 0.00018614 0.00018614    1 45.023  0.0019  0.965

rsq.lmm(m_cor_rootlitter)

$model
[1] 4.598793e-05

$fixed
[1] 4.598793e-05

$random
[1] 0
```

```
cor.test(PSF_cor_away$PSFtotal_plant, PSF_cor_away$PSFtotal_rootlitter)
```

Pearson's product-moment correlation

```
data: PSF_cor_away$PSFtotal_plant and PSF_cor_away$PSFtotal_rootlitter
```

```
t = 0.045995, df = 46, p-value = 0.9635
```

```
alternative hypothesis: true correlation is not equal to 0
```

```
95 percent confidence interval:
```

```
-0.2778887 0.2903567
```

```
sample estimates:
```

```
cor
```

```
0.00678144
```

Make correlation plots

```
P_PSF_plant_inoc <- ggplot(data = PSF_cor_away, mapping = aes(x =  
PSFtotal_inoculum, y = PSFtotal_plant))+  
  geom_point(size = 3, shape = 21, aes(fill = drought))+  
  scale_fill_manual(values = DroughtPalette, name = "Drought\treatment",  
labels = c("well-watered", "drought"))+  
  geom_smooth(method = lm, col = "black", fill = "black", alpha = 0.2)+  
  labs(x = "PSF inoculum", y = "PSF plant", title = "Inoculum")+  
  theme_bw()+ PSFtheme + theme(plot.title = element_text(size = 15, hjust =  
0.5)) +  
  geom_abline(aes(intercept = `(Intercept)`, slope = PSFtotal_inoculum), data  
= as.data.frame(t(fixef(m_cor_inoculum))), col = "#3C5941", size = 1)
```

```
P_PSF_plant_exud <-ggplot(data = PSF_cor_away, mapping = aes(x =  
PSFtotal_exudates, y = PSFtotal_plant))+  
  geom_point(size = 3, shape = 21, aes(fill = drought))+  
  scale_fill_manual(values = DroughtPalette, name = "Drought\treatment",  
labels = c("well-watered", "drought"))+  
  geom_smooth(method = lm, col = "black", fill = "black", alpha = 0.2)+  
  labs(x = "PSF root exudates", y = "", title = "Root exudates")+  
  theme_bw()+ PSFtheme + theme(plot.title = element_text(size = 15, hjust =  
0.5)) +  
  geom_abline(aes(intercept = `(Intercept)`, slope = PSFtotal_exudates), data  
= as.data.frame(t(fixef(m_cor_exudates))), col = "#3C5941", size = 1)
```

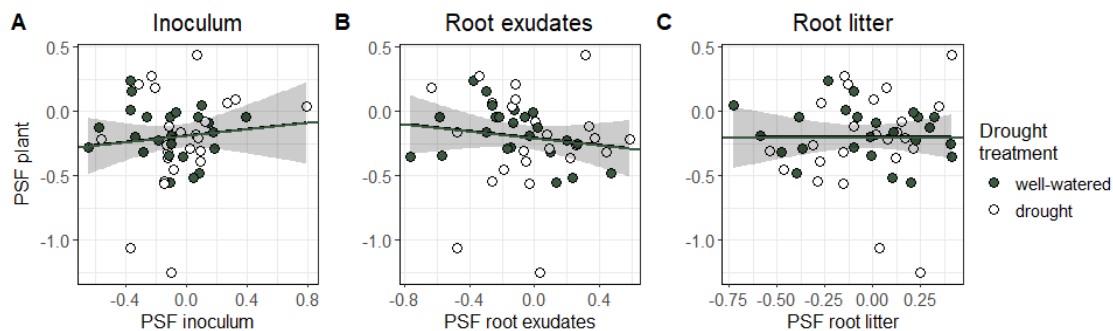
```

P_PSF_plant_root <-ggplot(data = PSF_cor_away, mapping = aes(x =
PSFtotal_rootlitter, y = PSFtotal_plant))+
  geom_point(size = 3, shape = 21, aes(fill = drought))+
  scale_fill_manual(values = DroughtPalette, name = "Drought\treatment",
labels = c("well-watered", "drought"))+
  geom_smooth(method = lm, col = "black",fill = "black", alpha = 0.2)+
  labs(x = "PSF root litter", y = "", title = "Root litter")+
  theme_bw()+ PSFtheme + theme(plot.title = element_text(size = 15, hjust =
0.5)) +
  geom_abline(aes(intercept = `(Intercept)`, slope = PSFtotal_rootlitter),
data = as.data.frame(t(fixef(m_cor_rootlitter))), col = "#3C5941", size = 1)

P_PSFcor_combined <- ggarrange(P_PSF_plant_inoc,
P_PSF_plant_exud,P_PSF_plant_root, nrow = 1, common.legend = T, legend =
"right", labels = "AUTO")

P_PSFcor_combined

```



Save figure

```

ggsave(plot = P_PSFcor_combined, "PSF_cor_combined_away.tiff", width = 30,
height = 9, units = "cm", dpi = 1000)

```

*Correlations between plant biomass and soil parameters*

Fuse conditioning and feedback data

```

data_complete <- data_feedback %>%
  dplyr::rename(shootDW_feedback = shootDW, rootDW_feedback = rootDW,
totalDW_feedback = totalDW,
  rootshootratio_feedback = rootshootratio,height_feedback = height,
leaves_feedback = leaves) %>%
  full_join(data_conditioning, by = c("soil_ID", "block", "input",
"drought")) %>%
  dplyr::select(!species & !number) %>%
  relocate(species_home, .before = species_soil)

```

Filter only for plant-conditioned soil

```

data_plant <- filter(data_complete, input == "plant")

```

## Correlation plant biomass with microbial C

```
m_cor_microbialC <- lmer(totalDW_feedback ~ microbialC + drought + (1|block)
, data = data_plant)
```

boundary (singular) fit: see help('isSingular')

```
anova(m_cor_microbialC, type = 3, ddf = "Kenward-Roger")
```

Type III Analysis of Variance Table with Kenward-Roger's method

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)
microbialC	0.000602	0.000602	1	40.839	0.1173	0.7338
drought	0.140924	0.140924	1	67.248	27.4334	1.759e-06 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

```
rsq.lmm(m_cor_microbialC)
```

```
$model
```

```
[1] 0.423841
```

```
$fixed
```

```
[1] 0.423841
```

```
$random
```

```
[1] 0
```

```
cor.test(data_plant$totalDW_feedback, data_plant$microbialC)
```

Pearson's product-moment correlation

data: data\_plant\$totalDW\_feedback and data\_plant\$microbialC

t = 3.8403, df = 70, p-value = 0.0002669

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

0.2053378 0.5916512

sample estimates:

cor

0.4171618

## Correlation plant biomass with microbial N

```
m_cor_microbialN <- lmer(totalDW_feedback ~ microbialN + drought + (1|block)
, data = data_plant)
```

boundary (singular) fit: see help('isSingular')

```

anova(m_cor_microbialN, type = 3, ddf = "Kenward-Roger")

Type III Analysis of Variance Table with Kenward-Roger's method
      Sum Sq  Mean Sq NumDF  DenDF F value    Pr(>F)
microbialN 0.001719 0.001719     1 25.225  0.3361    0.5672
drought    0.129773 0.129773     1 51.962 25.3734 6.055e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

rsq.lmm(m_cor_microbialN)

$model
[1] 0.4263563

$fixed
[1] 0.4263563

$random
[1] 0

cor.test(data_plant$totalDW_feedback, data_plant$microbialN)

Pearson's product-moment correlation

data: data_plant$totalDW_feedback and data_plant$microbialN
t = 3.903, df = 70, p-value = 0.000216
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.2118389 0.5960511
sample estimates:
      cor
0.4227599

```

Correlation plant biomass with plan-available N

```

m_cor_availableN <- lmer(totalDW_feedback ~ availableN + drought + (1|block)
, data = data_plant)

boundary (singular) fit: see help('isSingular')

anova(m_cor_availableN, type = 3, ddf = "Kenward-Roger")

Type III Analysis of Variance Table with Kenward-Roger's method
      Sum Sq  Mean Sq NumDF  DenDF F value    Pr(>F)
availableN 0.000552 0.000552     1 56.155  0.1073    0.7444
drought    0.220613 0.220613     1 68.327 42.9336 8.839e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
rsq.lmm(m_cor_availableN)

$model
[1] 0.4236701

$fixed
[1] 0.4236701

$random
[1] 0

cor.test(data_plant$totalDW_feedback, data_plant$availableN)

Pearson's product-moment correlation

data: data_plant$totalDW_feedback and data_plant$availableN
t = -2.0109, df = 70, p-value = 0.04819
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.441461176 -0.002140351
sample estimates:
      cor
-0.233693
```

Correlation plant biomass with plan-available P

```
m_cor_availableP <- lmer(totalDW_feedback ~ availableP + drought + (1|block)
, data = data_plant)

boundary (singular) fit: see help('isSingular')

anova(m_cor_availableP, type = 3, ddf = "Kenward-Roger")

Type III Analysis of Variance Table with Kenward-Roger's method
      Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
availableP 0.000017 0.000017      1 53.912  0.0033  0.9542
drought    0.194136 0.194136      1 68.872 37.7139 4.628e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

rsq.lmm(m_cor_availableP)

$model
[1] 0.4226475

$fixed
[1] 0.4226475

$random
[1] 0
```

```
cor.test(data_plant$totalDW_feedback, data_plant$availableP)
```

Pearson's product-moment correlation

```
data: data_plant$totalDW_feedback and data_plant$availableP
```

```
t = -2.6862, df = 70, p-value = 0.009018
```

```
alternative hypothesis: true correlation is not equal to 0
```

```
95 percent confidence interval:
```

```
-0.50182537 -0.07966986
```

```
sample estimates:
```

```
cor
```

```
-0.3056962
```

Make correlation plots

```
P_cor_micC <- ggplot(data_plant, aes(x = microbialC, y = totalDW_feedback,
fill = drought))+
  geom_point(shape = 21, col = "black", size = 3) +
  scale_fill_manual(values = DroughtPalette, name = "Drought treatment",
labels = c("well-watered", "drought"))+
  geom_smooth(method = "lm", col = "black", fill = "black", alpha = 0.2)+
  PSFtheme + theme(plot.title = element_text(hjust = 0.5, size = 13),
axis.line = element_blank())+
  labs(x = "Microbial biomass C [ $\mu\text{g/ g soil}$ ]", y = "Total dry biomass [g]",
title = "Microbial C")
```

```
P_cor_micN <- ggplot(data_plant, aes(x = microbialN, y = totalDW_feedback,
fill = drought))+
  geom_point(shape = 21, col = "black", size = 3) +
  scale_fill_manual(values = DroughtPalette, name = "Drought treatment",
labels = c("well-watered", "drought"))+
  geom_smooth(method = "lm", col = "black", fill = "black", alpha = 0.2)+
  PSFtheme + theme(plot.title = element_text(hjust = 0.5, size = 13),
axis.line = element_blank())+
  labs(x = "Microbial biomass N [ $\mu\text{g/ g soil}$ ]", y = "Total dry biomass [g]",
title = "Microbial N")
```

```
P_cor_avN <- ggplot(data_plant, aes(x = availableN, y = totalDW_feedback,
fill = drought))+
  geom_point(shape = 21, col = "black", size = 3) +
  scale_fill_manual(values = DroughtPalette, name = "Drought treatment",
labels = c("well-watered", "drought"))+
  geom_smooth(method = "lm", col = "black", fill = "black", alpha = 0.2)+
  PSFtheme + theme(plot.title = element_text(hjust = 0.5, size = 13),
axis.line = element_blank())+
  labs(x = "Plant-available N [ $\mu\text{g/ g soil}$ ]", y = "Total dry biomass [g]",
title = "Plant-available N")
```

```

P_cor_avP <- ggplot(data_plant, aes(x = availableP, y = totalDW_feedback,
fill = drought))+
  geom_point(shape = 21, col = "black", size = 3) +
  scale_fill_manual(values = DroughtPalette, name = "Drought treatment",
labels = c("well-watered", "drought"))+
  geom_smooth(method = "lm", col = "black", fill = "black", alpha = 0.2)+
  PSFtheme + theme(plot.title = element_text(hjust = 0.5, size = 13),
axis.line = element_blank())+
  labs(x = "Plant-available P [ $\mu\text{g}/\text{g}$  soil]", y = "Total dry biomass [g]",
title = "Plant-available P")

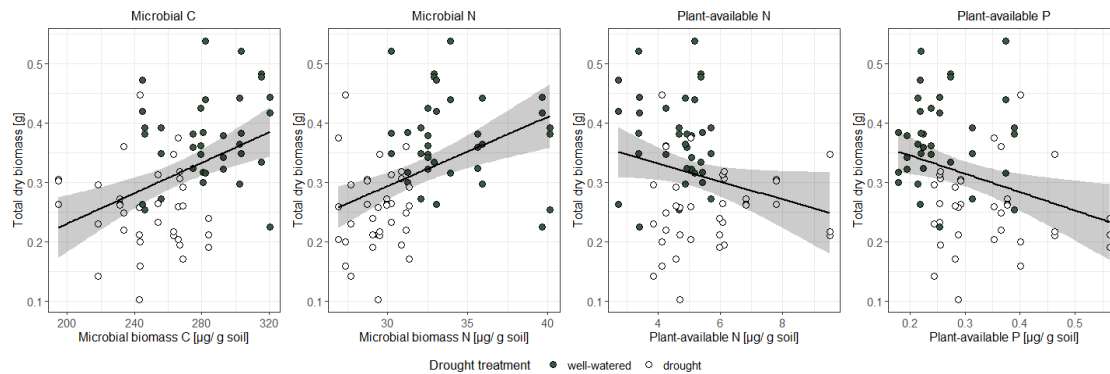
```

```

P_cor_combined <- ggarrange(P_cor_micC, P_cor_micN, P_cor_avN, P_cor_avP,
nrow = 1, common.legend = T, legend = "bottom")

```

P\_cor\_combined



```

ggsave(plot = P_cor_combined, "cor_totalDW_soil.tiff", width = 30, height =
10, units = "cm", dpi = 1000)

```

# Microbial community analysis

## Prepare work environment

### Load additional packages

```
conflicts_prefer(tidyr::expand)
```

[conflicted] Will prefer tidyr::expand over any other package.

```
conflicts_prefer(dplyr::collapse)
```

[conflicted] Will prefer dplyr::collapse over any other package.

```
pacman::p_load(data.table, phyloseq, vegan, microbiome, ALDEx2, DESeq2, grid,
gridExtra,
                scales, DAtest, metagenomeSeq, dendextend, microViz,
pairwiseAdonis, ggvegan, paletteer,
                ggVennDiagram, coda4microbiome, indicpecies)
```

### Define additional visuals

#### ggplot themes

```
ordination_theme <- theme(
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  panel.background = element_blank(),
  panel.border = element_blank(),
  axis.line.x = element_line(color="black", linewidth = 0.5),
  axis.line.y = element_line(color="black", linewidth = 0.5),
  strip.text.x = element_text(size = 20),
  strip.text.y = element_text(size = 20),
  strip.background = element_rect(fill="#FFFFFF", color = "black", linewidth =
0.5),
  axis.text.x = element_text(size = 15),
  axis.text.y = element_text(size = 15),
  axis.title = element_text(size = 20),
  legend.text = element_text(size = 15), legend.title = element_text(size =
20),
  plot.title = element_text(hjust = 0.5, size = 25, color = "grey40"))
```

```

mytheme <- theme(
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  panel.background = element_blank(),
  panel.border =element_blank(),
  axis.line.x = element_line(color="black", linewidth = 0.5),
  axis.line.y = element_line(color="black", linewidth = 0.5),
  strip.text.x = element_text(size = 15),
  strip.text.y = element_text(size = 15),
  strip.background = element_rect(fill="#FFFFFF", color = "black", linewidth =
0.5),
  axis.text.x = element_text(size = 12),
  axis.text.y = element_text(size = 12),
  axis.title = element_text(size = 15),
  legend.text = element_text(size = 12), legend.title = element_text(size =
15)
)

```

## Load data

Here, we load two phyloseq objects containing the sequence table, the taxonomy table and the metadata of bacteria and archaea (16s) and fungi (ITS).

```

phyloseq_16s <- readRDS("ELE_phyloseq_16s.RDS")
phyloseq_16s

phyloseq-class experiment-level object
otu_table() OTU Table: [ 26490 taxa and 100 samples ]
sample_data() Sample Data: [ 100 samples by 9 sample variables ]
tax_table() Taxonomy Table: [ 26490 taxa by 10 taxonomic ranks ]
phy_tree() Phylogenetic Tree: [ 26490 tips and 26488 internal nodes ]

phyloseq_ITS <- readRDS("ELE_phyloseq_ITS.RDS")
phyloseq_ITS

phyloseq-class experiment-level object
otu_table() OTU Table: [ 3930 taxa and 100 samples ]
sample_data() Sample Data: [ 100 samples by 9 sample variables ]
tax_table() Taxonomy Table: [ 3930 taxa by 10 taxonomic ranks ]
phy_tree() Phylogenetic Tree: [ 3930 tips and 3928 internal nodes ]

```

## Extract the components of the phyloseq as data frames

```

otu_table_16s <- as.data.frame(otu_table(phyloseq_16s))
sample_data_16s <- as.data.frame(sample_data(phyloseq_16s))
tax_table_16s <- as.data.frame(tax_table(phyloseq_16s))

otu_table_ITS <- as.data.frame(otu_table(phyloseq_ITS))
sample_data_ITS <- as.data.frame(sample_data(phyloseq_ITS))
tax_table_ITS <- as.data.frame(tax_table(phyloseq_ITS))

```

## Filter data

We filter the data in three steps: 1. Remove samples with low read counts -> Not necessary as read numbers are very evenly distributed 2. Remove ASVs not assigned at domain/ phylum level (eg DNA not from the target taxa) 3. Remove rare ASVs that are likely the product of sequencing errors

### Remove ASVs unassigned at domain level

By checking the taxonomy tables, only the 16s data frame had ASVs unassigned at domain level. The ITS phyloseq could thus be left unchanged.

```
phyloseq_16s_filt <- subset_taxa(phyloseq_16s, domain != "unknown")
phyloseq_16s_filt

phyloseq-class experiment-level object
otu_table() OTU Table: [ 26489 taxa and 100 samples ]
sample_data() Sample Data: [ 100 samples by 9 sample variables ]
tax_table() Taxonomy Table: [ 26489 taxa by 10 taxonomic ranks ]
phy_tree() Phylogenetic Tree: [ 26489 tips and 26487 internal nodes ]
```

### Remove ASVs that occur in less than 3 % of the samples

```
phyloseq_16s_3perc <- prune_taxa( taxa_sums(phyloseq_16s_filt) >= 3,
phyloseq_16s_filt)
phyloseq_16s_3perc

phyloseq-class experiment-level object
otu_table() OTU Table: [ 20236 taxa and 100 samples ]
sample_data() Sample Data: [ 100 samples by 9 sample variables ]
tax_table() Taxonomy Table: [ 20236 taxa by 10 taxonomic ranks ]
phy_tree() Phylogenetic Tree: [ 20236 tips and 20235 internal nodes ]

phyloseq_ITS_3perc <- prune_taxa( taxa_sums(phyloseq_ITS) >= 3, phyloseq_ITS)
phyloseq_ITS_3perc

phyloseq-class experiment-level object
otu_table() OTU Table: [ 3845 taxa and 100 samples ]
sample_data() Sample Data: [ 100 samples by 9 sample variables ]
tax_table() Taxonomy Table: [ 3845 taxa by 10 taxonomic ranks ]
phy_tree() Phylogenetic Tree: [ 3845 tips and 3843 internal nodes ]
```

Extract components of filtered phyloseq

```
otu_table_16s_3perc <- as.data.frame(otu_table(phyloseq_16s_3perc ))
sample_data_16s_3perc <- as.data.frame(sample_data(phyloseq_16s_3perc ))
tax_table_16s_3perc <- as.data.frame(tax_table(phyloseq_16s_3perc ))

otu_table_ITS_3perc <- as.data.frame(otu_table(phyloseq_ITS_3perc))
sample_data_ITS_3perc <- data.frame(sample_data(phyloseq_ITS_3perc))
tax_table_ITS_3perc <- as.data.frame(tax_table(phyloseq_ITS_3perc))
```

## Normalize data

Here, we have three options, dependent on the subsequent data analysis 1. Relative abundance (compositional) 2. Rarefaction: Downscaling of the number of reads in all samples to a certain maximum, all samples with read numbers below that are discarded (for alpha diversity) 3. CLR (centered log-ratio transformation): Transformation of data based on the geometric mean within each sample on a log scale

### Relative abundance

```
phyloseq_16s_rel_abundance <- microbiome::transform(phyloseq_16s_3perc,
"compositional")
phyloseq_16s_rarefied <- rarefy_even_depth(phyloseq_16s_filt, rngseed = TRUE)
phyloseq_16s_clr <- microbiome::transform(phyloseq_16s_3perc, "clr")

phyloseq_ITS_rel_abundance <- microbiome::transform(phyloseq_ITS_3perc,
"compositional")
phyloseq_ITS_rarefied <- rarefy_even_depth(phyloseq_ITS, rngseed = TRUE)
phyloseq_ITS_clr <- microbiome::transform(phyloseq_ITS_3perc, "clr")
```

## Tidy up tax tables

### Empty cells of all unknown taxa

In that way they are all treated as a uniform group in the subsequent data analysis

```
tax_table_16s <- tax_table_16s %>% mutate(family = replace(family, family ==
"Unknown Family", ""))

tax_table_ITS <- tax_table_ITS %>%
  mutate(genus = replace(genus, genus == "unidentified", "")) %>%
  mutate(family = replace(family, family == "unidentified", "")) %>%
  mutate(order = replace(order, order == "unidentified", "")) %>%
  mutate(class = replace(class, class == "unidentified", "")) %>%
  mutate(phylum = replace(phylum, phylum == "unidentified", ""))
```

### Replace “\_” with a space in taxonomic rank names

```
tax_table_16s$species <- gsub("_", " ", tax_table_16s$species)
tax_table_16s$genus <- gsub("_", " ", tax_table_16s$genus)
tax_table_16s$family <- gsub("_", " ", tax_table_16s$family)
tax_table_16s$order <- gsub("_", " ", tax_table_16s$order)

tax_table_ITS$species <- ifelse(grepl("_sp$", tax_table_ITS$species), "",
tax_table_ITS$species)
tax_table_ITS$species <- gsub("_", " ", tax_table_ITS$species)
```

### Introduce variable for highest classified rank and the corresponding taxon name

```
tax_table_16s$taxa <- tax_table_16s$genus
tax_table_16s$rank <- ifelse(tax_table_16s$genus != "", "genus", "")
tax_table_16s$taxa <- ifelse(tax_table_16s$genus == "", tax_table_16s$family,
tax_table_16s$taxa )
```

```

tax_table_16s$rank <- ifelse(tax_table_16s$taxa != "" & tax_table_16s$rank ==
"", "family", tax_table_16s$rank)
tax_table_16s$taxa <- ifelse(tax_table_16s$family == "", tax_table_16s$order,
tax_table_16s$taxa )
tax_table_16s$rank <- ifelse(tax_table_16s$taxa != "" & tax_table_16s$rank ==
"", "order", tax_table_16s$rank)
tax_table_16s$taxa <- ifelse(tax_table_16s$order == "", tax_table_16s$class,
tax_table_16s$taxa )
tax_table_16s$rank <- ifelse(tax_table_16s$taxa != "" & tax_table_16s$rank ==
"", "class", tax_table_16s$rank)
tax_table_16s$taxa <- ifelse(tax_table_16s$class == "", tax_table_16s$phylum,
tax_table_16s$taxa )
tax_table_16s$rank <- ifelse(tax_table_16s$taxa != "" & tax_table_16s$rank ==
"", "phylum", tax_table_16s$rank)
tax_table_16s$taxa <- ifelse(tax_table_16s$phylum == "",
tax_table_16s$domain, tax_table_16s$taxa )
tax_table_16s$rank <- ifelse(tax_table_16s$taxa != "" & tax_table_16s$rank ==
"", "domain", tax_table_16s$rank)

tax_table_ITS$taxa <- tax_table_ITS$species
tax_table_ITS$rank <- ifelse(tax_table_ITS$species != "", "species", "")
tax_table_ITS$taxa <- ifelse(tax_table_ITS$species == "",
tax_table_ITS$genus, tax_table_ITS$taxa )
tax_table_ITS$rank <- ifelse(tax_table_ITS$taxa != "" & tax_table_ITS$rank ==
"", "genus", tax_table_ITS$rank)
tax_table_ITS$taxa <- ifelse(tax_table_ITS$genus == "", tax_table_ITS$family,
tax_table_ITS$taxa )
tax_table_ITS$rank <- ifelse(tax_table_ITS$taxa != "" & tax_table_ITS$rank ==
"", "family", tax_table_ITS$rank)
tax_table_ITS$taxa <- ifelse(tax_table_ITS$family == "", tax_table_ITS$order,
tax_table_ITS$taxa )
tax_table_ITS$rank <- ifelse(tax_table_ITS$taxa != "" & tax_table_ITS$rank ==
"", "order", tax_table_ITS$rank)
tax_table_ITS$taxa <- ifelse(tax_table_ITS$order == "", tax_table_ITS$class,
tax_table_ITS$taxa )
tax_table_ITS$rank <- ifelse(tax_table_ITS$taxa != "" & tax_table_ITS$rank ==
"", "class", tax_table_ITS$rank)
tax_table_ITS$taxa <- ifelse(tax_table_ITS$class == "", tax_table_ITS$phylum,
tax_table_ITS$taxa )
tax_table_ITS$rank <- ifelse(tax_table_ITS$taxa != "" & tax_table_ITS$rank ==
"", "phylum", tax_table_ITS$rank)
tax_table_ITS$taxa <- ifelse(tax_table_ITS$phylum == "",
tax_table_ITS$domain, tax_table_ITS$taxa )
tax_table_ITS$rank <- ifelse(tax_table_ITS$taxa != "" & tax_table_ITS$rank ==
"", "domain", tax_table_ITS$rank)

```

## Beta diversity (Ordinations & PERMANOVA)

### Distance-based redundancy analysis (dbRDA)

We chose a dbRDA with CLR-transformed data because we are most interested in the variation that is explained by our experimental variables, because it is compatible with a distance matrix and with compositional data.

We first create the distance matrix, then extract our metadata from the phyloseq object and execute the dbRDA with both of those inputs. Through the summary we can get an overview of the explained variation overall and within the RDA. Eventually we plot the first two axes of the dbRDA.

Bacteria & archaea

```
dist_dbRDA_16s_clr <- phyloseq::distance(phyloseq_16s_clr, method =
"euclidean")

metadata_16s_clr <- data.frame(sample_data(phyloseq_16s_clr))

dbRDA_16s_clr <- dbrda(dist_dbRDA_16s_clr ~ input * drought * species , data
= metadata_16s_clr, distance = "euclidean")
dbRDA_16s_clr
```

```
Call: dbrda(formula = dist_dbRDA_16s_clr ~ input * drought * species,
data = metadata_16s_clr, distance = "euclidean")
```

```
              Inertia Proportion Rank
Total          1.053e+04  1.000e+00
Constrained    2.930e+03  2.784e-01   24
Unconstrained  7.597e+03  7.216e-01   75
Inertia is mean squared Euclidean distance
Some constraints or conditions were aliased because they were redundant
```

```
Eigenvalues for constrained axes:
```

```
dbRDA1 dbRDA2 dbRDA3 dbRDA4 dbRDA5 dbRDA6 dbRDA7 dbRDA8 dbRDA9
dbRDA10
  456.1   216.0   127.8   123.0   115.6   109.7   109.0   107.0   105.5
104.8
dbRDA11 dbRDA12 dbRDA13 dbRDA14 dbRDA15 dbRDA16 dbRDA17 dbRDA18 dbRDA19
dbRDA20
  103.3   102.4   101.2    99.6    98.8    98.1    97.9    97.3    95.7
94.6
dbRDA21 dbRDA22 dbRDA23 dbRDA24
  93.9    93.0    91.0    88.9
```

```
Eigenvalues for unconstrained axes:
```

```
MDS1 MDS2 MDS3 MDS4 MDS5 MDS6 MDS7 MDS8
182.92 143.72 127.54 123.88 120.99 120.33 118.03 116.69
(Showing 8 of 75 unconstrained eigenvalues)
```

```
dbRDA_16s_clr_summary <- summary(dbRDA_16s_clr)
dbRDA_16s_clr_summary$concont
```

```
$importance
```

```
Importance of components:
```

	dbRDA1	dbRDA2	dbRDA3	dbRDA4	dbRDA5	
Eigenvalue	456.0602	215.97618	127.83815	122.96496	115.61609	
Proportion Explained	0.1556	0.07371	0.04363	0.04197	0.03946	
Cumulative Proportion	0.1556	0.22935	0.27298	0.31494	0.35440	
	dbRDA6	dbRDA7	dbRDA8	dbRDA9	dbRDA10	
Eigenvalue	109.74533	109.02983	107.03862	105.46911	104.81132	
Proportion Explained	0.03745	0.03721	0.03653	0.03599	0.03577	
Cumulative Proportion	0.39185	0.42906	0.46559	0.50159	0.53736	
	dbRDA11	dbRDA12	dbRDA13	dbRDA14	dbRDA15	
dbRDA16						
Eigenvalue	103.25411	102.41748	101.17740	99.57915	98.76222	
98.11957						
Proportion Explained	0.03524	0.03495	0.03453	0.03398	0.03371	
0.03349						
Cumulative Proportion	0.57260	0.60755	0.64208	0.67606	0.70977	
0.74325						
	dbRDA17	dbRDA18	dbRDA19	dbRDA20	dbRDA21	dbRDA22
Eigenvalue	97.90879	97.33457	95.71894	94.6304	93.86922	93.04540
Proportion Explained	0.03341	0.03322	0.03267	0.0323	0.03204	0.03175
Cumulative Proportion	0.77667	0.80988	0.84255	0.8748	0.90688	0.93864
	dbRDA23	dbRDA24				
Eigenvalue	90.95174	88.85530				
Proportion Explained	0.03104	0.03032				
Cumulative Proportion	0.96968	1.00000				

```
dbRDA_16s_clr_summary$cont
```

```
$importance
```

```
Importance of components:
```

	dbRDA1	dbRDA2	dbRDA3	dbRDA4	dbRDA5
Eigenvalue	456.06020	215.97618	127.83815	122.96496	115.61609
Proportion Explained	0.04332	0.02052	0.01214	0.01168	0.01098
Cumulative Proportion	0.04332	0.06384	0.07599	0.08767	0.09865
	dbRDA6	dbRDA7	dbRDA8	dbRDA9	dbRDA10
Eigenvalue	109.74533	109.02983	107.03862	105.46911	1.048e+02
Proportion Explained	0.01043	0.01036	0.01017	0.01002	9.957e-03
Cumulative Proportion	0.10908	0.11943	0.12960	0.13962	1.496e-01
	dbRDA11	dbRDA12	dbRDA13	dbRDA14	dbRDA15
Eigenvalue	1.033e+02	1.024e+02	1.012e+02	99.57915	98.762221
Proportion Explained	9.809e-03	9.729e-03	9.612e-03	0.00946	0.009382
Cumulative Proportion	1.594e-01	1.691e-01	1.787e-01	0.18819	0.197568
	dbRDA16	dbRDA17	dbRDA18	dbRDA19	dbRDA20
Eigenvalue	98.119574	97.908786	97.334567	95.718940	94.63044
Proportion Explained	0.009321	0.009301	0.009246	0.009093	0.00899
Cumulative Proportion	0.206889	0.216190	0.225437	0.234530	0.24352

		dbRDA21	dbRDA22	dbRDA23	dbRDA24	MDS1
Eigenvalue		93.869224	93.045404	90.95174	88.855302	182.92125
Proportion Explained		0.008917	0.008839	0.00864	0.008441	0.01738
Cumulative Proportion		0.252437	0.261276	0.26992	0.278357	0.29573
		MDS2	MDS3	MDS4	MDS5	MDS6
Eigenvalue		143.72259	127.53542	123.88387	120.98969	120.33006
Proportion Explained		0.01365	0.01212	0.01177	0.01149	0.01143
Cumulative Proportion		0.30939	0.32150	0.33327	0.34476	0.35620
		MDS7	MDS8	MDS9	MDS10	MDS11
Eigenvalue		118.03356	116.68923	115.7766	114.90960	113.81037
Proportion Explained		0.01121	0.01109	0.0110	0.01092	0.01081
Cumulative Proportion		0.36741	0.37849	0.3895	0.40041	0.41122
		MDS12	MDS13	MDS14	MDS15	MDS16
Eigenvalue		113.02803	112.46626	111.5993	111.48167	110.83959
Proportion Explained		0.01074	0.01068	0.0106	0.01059	0.01053
Cumulative Proportion		0.42196	0.43264	0.4432	0.45383	0.46436
		MDS17	MDS18	MDS19	MDS20	MDS21
Eigenvalue		110.11434	109.34086	108.67594	108.4371	107.51430
Proportion Explained		0.01046	0.01039	0.01032	0.0103	0.01021
Cumulative Proportion		0.47482	0.48521	0.49553	0.5058	0.51605
		MDS22	MDS23	MDS24	MDS25	MDS26
Eigenvalue		107.27752	106.86910	106.62295	105.71263	1.052e+02
Proportion Explained		0.01019	0.01015	0.01013	0.01004	9.994e-03
Cumulative Proportion		0.52624	0.53639	0.54652	0.55656	5.666e-01
		MDS27	MDS28	MDS29	MDS30	MDS31
Eigenvalue		1.045e+02	1.042e+02	1.036e+02	1.031e+02	1.025e+02
Proportion Explained		9.925e-03	9.895e-03	9.843e-03	9.796e-03	9.737e-03
Cumulative Proportion		5.765e-01	5.864e-01	5.962e-01	6.060e-01	6.158e-01
		MDS32	MDS33	MDS34	MDS35	MDS36
Eigenvalue		1.023e+02	1.020e+02	1.011e+02	1.009e+02	1.008e+02
Proportion Explained		9.714e-03	9.694e-03	9.603e-03	9.589e-03	9.574e-03
Cumulative Proportion		6.255e-01	6.352e-01	6.448e-01	6.544e-01	6.639e-01
		MDS37	MDS38	MDS39	MDS40	MDS41
Eigenvalue		1.001e+02	99.283602	99.122893	98.677092	98.177355
Proportion Explained		9.506e-03	0.009432	0.009416	0.009374	0.009327
Cumulative Proportion		6.734e-01	0.682862	0.692278	0.701652	0.710978
		MDS42	MDS43	MDS44	MDS45	MDS46
Eigenvalue		97.718878	97.392189	97.170727	96.780489	96.388329
Proportion Explained		0.009283	0.009252	0.009231	0.009194	0.009157
Cumulative Proportion		0.720261	0.729513	0.738744	0.747938	0.757095
		MDS47	MDS48	MDS49	MDS50	MDS51
Eigenvalue		95.499365	95.178604	94.641760	93.90105	93.534972
Proportion Explained		0.009072	0.009042	0.008991	0.00892	0.008886
Cumulative Proportion		0.766167	0.775208	0.784199	0.79312	0.802005
		MDS52	MDS53	MDS54	MDS55	MDS56
Eigenvalue		93.079433	92.821800	92.332145	92.011919	91.543479
Proportion Explained		0.008842	0.008818	0.008771	0.008741	0.008696
Cumulative Proportion		0.810847	0.819665	0.828436	0.837177	0.845873
		MDS57	MDS58	MDS59	MDS60	MDS61
Eigenvalue		90.586207	90.333833	89.736756	88.925780	88.356248

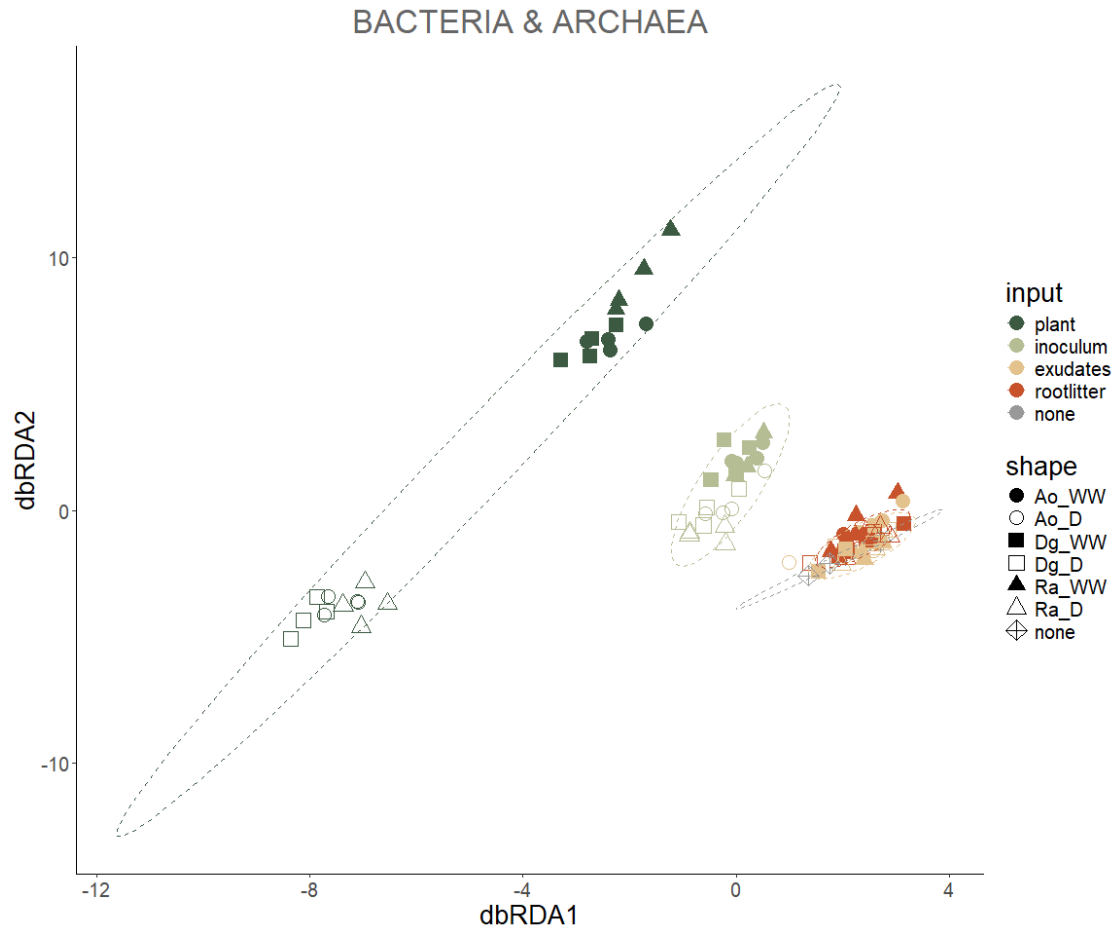
Proportion Explained	0.008605	0.008581	0.008525	0.008448	0.008394
Cumulative Proportion	0.854479	0.863060	0.871585	0.880032	0.888426
	MDS62	MDS63	MDS64	MDS65	MDS66
Eigenvalue	87.822484	87.655835	87.106081	86.659108	86.00754
Proportion Explained	0.008343	0.008327	0.008275	0.008232	0.00817
Cumulative Proportion	0.896769	0.905096	0.913371	0.921603	0.92977
	MDS67	MDS68	MDS69	MDS70	MDS71
MDS72					
Eigenvalue	85.952002	84.722704	83.354683	82.53024	82.42555
81.043149					
Proportion Explained	0.008165	0.008048	0.007918	0.00784	0.00783
0.007699					
Cumulative Proportion	0.937938	0.945987	0.953905	0.96175	0.96958
0.977274					
	MDS73	MDS74	MDS75		
Eigenvalue	80.756675	80.052043	78.417701		
Proportion Explained	0.007672	0.007605	0.007449		
Cumulative Proportion	0.984946	0.992551	1.000000		

```
dbRDA_16s_fort_clr <- fortify(dbRDA_16s_clr, display = "sites")
row.names(dbRDA_16s_fort_clr) <- dbRDA_16s_clr$label
```

```
dbRDA_16s_fort_clr <- merge(metadata_16s_clr, dbRDA_16s_fort_clr, by =
"row.names")
```

```
P_dbRDA_16s_clr <- ggplot(dbRDA_16s_fort_clr, aes(x = dbRDA1, y = dbRDA2, col
= input, shape = shape))+
  geom_point(size = 5)+
  scale_color_manual(values = InputPalette)+
  scale_shape_manual(values = Shape_SpeciesDrought)+
  stat_ellipse(aes(group = input), linetype = 2) +
  labs(title = "BACTERIA & ARCHAEA")+
  ordination_theme
```

```
P_dbRDA_16s_clr
```



Save figure

```
ggsave(plot = P_dbRDA_16s_clr, "dbRDA_16s.tiff", width = 20, height = 15,
units = "cm", dpi = 700)
```

Fungi

```
sample_data_ITS_clr <- data.frame(sample_data(phyloseq_ITS_clr))
dist_dbRDA_ITS <- phyloseq::distance(phyloseq_ITS_clr, method = "euclidean")
dbRDA_ITS <- dbrda(dist_dbRDA_ITS ~ input * drought * species + block , data
= sample_data_ITS_clr, distance = "euclidean")
dbRDA_ITS # constraint analysis explains ~ 30 % of the total variation
Call: dbrda(formula = dist_dbRDA_ITS ~ input * drought * species +
block, data = sample_data_ITS_clr, distance = "euclidean")
```

	Inertia	Proportion	Rank
Total	4849.9404	1.0000	
Constrained	1470.9315	0.3033	27
Unconstrained	3379.0090	0.6967	72

Inertia is mean squared Euclidean distance

Some constraints or conditions were aliased because they were redundant

Eigenvalues for constrained axes:

dbRDA1	dbRDA2	dbRDA3	dbRDA4	dbRDA5	dbRDA6	dbRDA7	dbRDA8	dbRDA9
186.25	70.09	65.11	61.08	59.89	58.84	57.81	55.54	54.20

53.89

dbRDA11	dbRDA12	dbRDA13	dbRDA14	dbRDA15	dbRDA16	dbRDA17	dbRDA18	dbRDA19
51.36	50.41	50.01	48.70	48.05	47.55	45.67	44.56	43.98

43.61

dbRDA21	dbRDA22	dbRDA23	dbRDA24	dbRDA25	dbRDA26	dbRDA27
42.31	41.43	40.43	39.64	38.86	36.39	35.24

Eigenvalues for unconstrained axes:

MDS1	MDS2	MDS3	MDS4	MDS5	MDS6	MDS7	MDS8
82.13	73.61	67.57	66.72	66.12	64.64	63.67	62.99

(Showing 8 of 72 unconstrained eigenvalues)

```
dbRDA_ITS_summary <- summary(dbRDA_ITS)
```

```
dbRDA_ITS_summary$concont # RDA1 and 2 explain ~ 13% and 5% of the variation explained by the metadata
```

```
$importance
```

```
Importance of components:
```

	dbRDA1	dbRDA2	dbRDA3	dbRDA4	dbRDA5	dbRDA6
Eigenvalue	186.2535	70.09044	65.11248	61.07636	59.89451	58.8429
Proportion Explained	0.1266	0.04765	0.04427	0.04152	0.04072	0.0400
Cumulative Proportion	0.1266	0.17427	0.21854	0.26006	0.30078	0.3408

	dbRDA7	dbRDA8	dbRDA9	dbRDA10	dbRDA11	dbRDA12
Eigenvalue	57.8064	55.53755	54.20479	53.89006	51.36464	50.41480
Proportion Explained	0.0393	0.03776	0.03685	0.03664	0.03492	0.03427
Cumulative Proportion	0.3801	0.41784	0.45469	0.49133	0.52625	0.56052

	dbRDA13	dbRDA14	dbRDA15	dbRDA16	dbRDA17	dbRDA18
Eigenvalue	50.0078	48.69769	48.05311	47.55416	45.67093	44.5642
Proportion Explained	0.0340	0.03311	0.03267	0.03233	0.03105	0.0303
Cumulative Proportion	0.5945	0.62763	0.66029	0.69262	0.72367	0.7540

	dbRDA19	dbRDA20	dbRDA21	dbRDA22	dbRDA23	dbRDA24
Eigenvalue	43.9812	43.60979	42.31258	41.42654	40.43207	39.64195
Proportion Explained	0.0299	0.02965	0.02877	0.02816	0.02749	0.02695
Cumulative Proportion	0.7839	0.81352	0.84228	0.87045	0.89793	0.92488

	dbRDA25	dbRDA26	dbRDA27
Eigenvalue	38.85904	36.38686	35.24495
Proportion Explained	0.02642	0.02474	0.02396
Cumulative Proportion	0.95130	0.97604	1.00000

```
dbRDA_ITS_summary$cont # RDA1 and 2 only explain ~ 4% and 1.5% of the total variation
```

```
$importance
```

```
Importance of components:
```

		dbRDA1	dbRDA2	dbRDA3	dbRDA4	dbRDA5	dbRDA6
Eigenvalue		186.2535	70.09044	65.11248	61.07636	59.89451	58.84295
Proportion Explained		0.0384	0.01445	0.01343	0.01259	0.01235	0.01213
Cumulative Proportion		0.0384	0.05286	0.06628	0.07887	0.09122	0.10336
		dbRDA7	dbRDA8	dbRDA9	dbRDA10	dbRDA11	dbRDA12
Eigenvalue		57.80644	55.53755	54.20479	53.89006	51.36464	50.41480
Proportion Explained		0.01192	0.01145	0.01118	0.01111	0.01059	0.01039
Cumulative Proportion		0.11527	0.12673	0.13790	0.14901	0.15960	0.17000
		dbRDA13	dbRDA14	dbRDA15	dbRDA16	dbRDA17	
dbRDA18							
Eigenvalue		50.00783	48.69769	48.053114	47.554161	45.670932	
44.564238							
Proportion Explained		0.01031	0.01004	0.009908	0.009805	0.009417	
0.009189							
Cumulative Proportion		0.18031	0.19035	0.200260	0.210065	0.219482	
0.228670							
		dbRDA19	dbRDA20	dbRDA21	dbRDA22	dbRDA23	
Eigenvalue		43.981178	43.609789	42.312583	41.426544	40.432069	
Proportion Explained		0.009068	0.008992	0.008724	0.008542	0.008337	
Cumulative Proportion		0.237739	0.246730	0.255455	0.263996	0.272333	
		dbRDA24	dbRDA25	dbRDA26	dbRDA27	MDS1	
MDS2							
Eigenvalue		39.641950	38.859045	36.386863	35.244951	82.12909	
73.60653							
Proportion Explained		0.008174	0.008012	0.007503	0.007267	0.01693	
0.01518							
Cumulative Proportion		0.280507	0.288519	0.296021	0.303289	0.32022	
0.33540							
		MDS3	MDS4	MDS5	MDS6	MDS7	MDS8
Eigenvalue		67.56519	66.72434	66.11804	64.63972	63.66875	62.99294
Proportion Explained		0.01393	0.01376	0.01363	0.01333	0.01313	0.01299
Cumulative Proportion		0.34933	0.36309	0.37672	0.39005	0.40318	0.41617
		MDS9	MDS10	MDS11	MDS12	MDS13	MDS14
Eigenvalue		62.14828	60.58601	59.70171	59.42311	57.83280	57.53991
Proportion Explained		0.01281	0.01249	0.01231	0.01225	0.01192	0.01186
Cumulative Proportion		0.42898	0.44147	0.45378	0.46603	0.47796	0.48982
		MDS15	MDS16	MDS17	MDS18	MDS19	MDS20
Eigenvalue		56.43683	56.2433	56.02865	54.51176	54.25206	53.70494
Proportion Explained		0.01164	0.0116	0.01155	0.01124	0.01119	0.01107
Cumulative Proportion		0.50146	0.5131	0.52461	0.53585	0.54703	0.55811
		MDS21	MDS22	MDS23	MDS24	MDS25	MDS26
Eigenvalue		52.69960	52.20074	51.93670	51.44690	50.4482	50.26464
Proportion Explained		0.01087	0.01076	0.01071	0.01061	0.0104	0.01036
Cumulative Proportion		0.56897	0.57974	0.59044	0.60105	0.6115	0.62182
		MDS27	MDS28	MDS29	MDS30	MDS31	MDS32
Eigenvalue		50.10597	49.13838	48.73218	48.5050	47.815653	47.5289
Proportion Explained		0.01033	0.01013	0.01005	0.0100	0.009859	0.0098
Cumulative Proportion		0.63215	0.64228	0.65233	0.6623	0.672190	0.6820
		MDS33	MDS34	MDS35	MDS36	MDS37	
Eigenvalue		46.774163	46.249907	46.010333	45.750712	45.175597	

Proportion Explained	0.009644	0.009536	0.009487	0.009433	0.009315
Cumulative Proportion	0.691634	0.701170	0.710657	0.720090	0.729405
	MDS38	MDS39	MDS40	MDS41	MDS42
Eigenvalue	45.079492	44.437646	44.017032	43.489951	43.025804
Proportion Explained	0.009295	0.009163	0.009076	0.008967	0.008871
Cumulative Proportion	0.738699	0.747862	0.756938	0.765905	0.774776
	MDS43	MDS44	MDS45	MDS46	MDS47
MDS48					
Eigenvalue	42.72586	42.467347	42.233335	41.90266	40.776503
40.605386					
Proportion Explained	0.00881	0.008756	0.008708	0.00864	0.008408
0.008372					
Cumulative Proportion	0.78359	0.792342	0.801050	0.80969	0.818098
0.826470					
	MDS49	MDS50	MDS51	MDS52	MDS53
MDS54					
Eigenvalue	40.141613	40.104419	39.7716	38.906599	38.336846
38.059192					
Proportion Explained	0.008277	0.008269	0.0082	0.008022	0.007905
0.007847					
Cumulative Proportion	0.834747	0.843016	0.8512	0.859238	0.867143
0.874990					
	MDS55	MDS56	MDS57	MDS58	MDS59
Eigenvalue	37.767255	37.523585	37.39131	36.886892	36.259804
Proportion Explained	0.007787	0.007737	0.00771	0.007606	0.007476
Cumulative Proportion	0.882777	0.890514	0.89822	0.905830	0.913306
	MDS60	MDS61	MDS62	MDS63	MDS64
Eigenvalue	35.666343	35.292599	35.001710	34.785809	34.368577
Proportion Explained	0.007354	0.007277	0.007217	0.007172	0.007086
Cumulative Proportion	0.920660	0.927937	0.935154	0.942326	0.949413
	MDS65	MDS66	MDS67	MDS68	MDS69
Eigenvalue	33.936745	33.769442	33.397752	32.405398	31.686895
Proportion Explained	0.006997	0.006963	0.006886	0.006682	0.006533
Cumulative Proportion	0.956410	0.963373	0.970259	0.976941	0.983474
	MDS70	MDS71	MDS72		
Eigenvalue	31.179194	30.563190	18.407602		
Proportion Explained	0.006429	0.006302	0.003795		
Cumulative Proportion	0.989903	0.996205	1.000000		

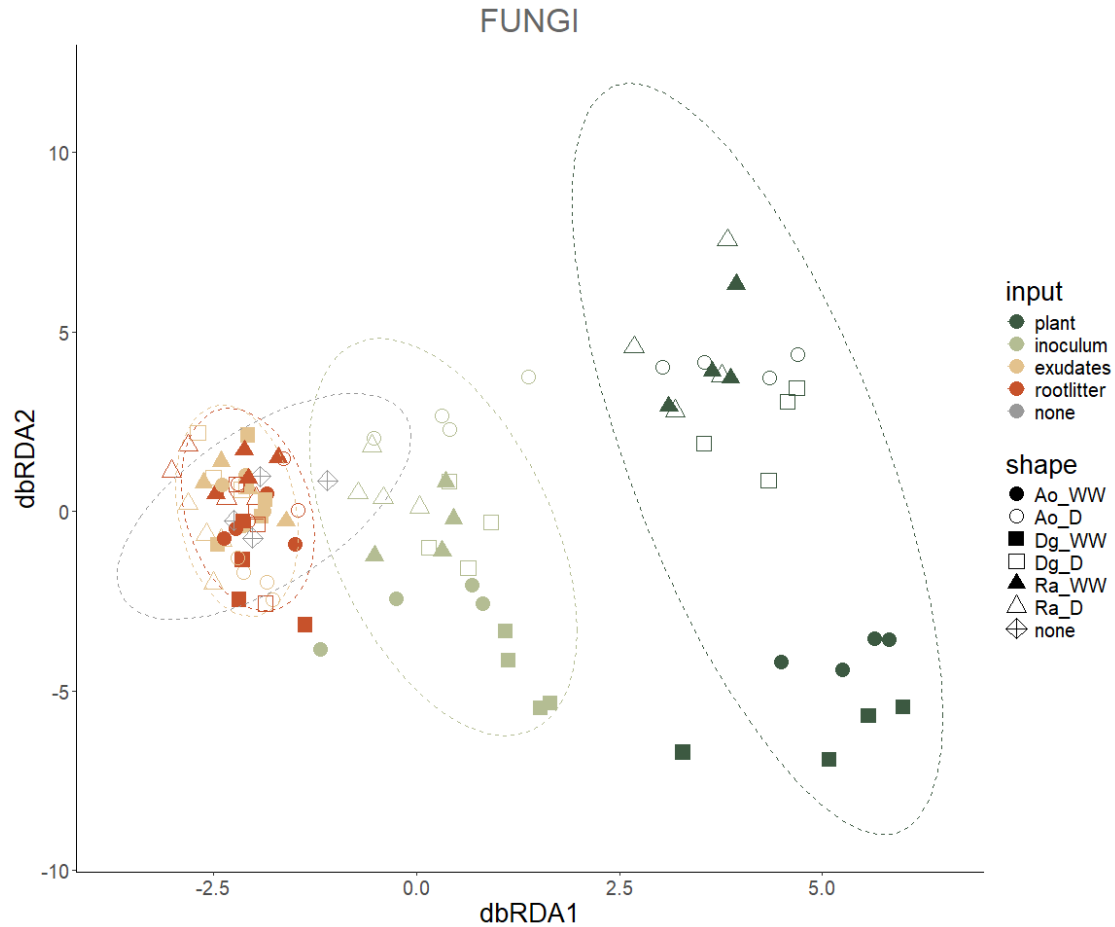
```
dbRDA_ITS_fort <- fortify(dbRDA_ITS, display = "sites")
row.names(dbRDA_ITS_fort) <- dbRDA_ITS_fort$label
```

```
dbRDA_ITS_fort <- merge(sample_data_ITS_clr, dbRDA_ITS_fort, by = "row.names")
```

```
P_dbRDA_ITS <- ggplot(dbRDA_ITS_fort, aes(x = dbRDA1, y = dbRDA2, col =
input, shape = shape))+
  geom_point(size = 5)+
  scale_color_manual(values = InputPalette)+
  scale_shape_manual(values = Shape_SpeciesDrought)+
  stat_ellipse(aes(group = input), linetype = 2) +
```

```
labs(title = "FUNGI")+  
ordination_theme
```

P\_dbRDA\_ITS



Save figure

```
ggsave(plot = P_dbRDA_ITS, "dbRDA_ITS.tiff", width = 20, height = 15, units =  
"cm", dpi = 700)
```

## PERMANOVA

Remove unconditioned soil in order to maintain a full-factorial design

```
ps_rmnone_16s_clr <- ps_filter(phyloseq_16s_clr, input != "none", .target =  
"sample_data", .keep_all_taxa = T)
```

```
ps_rmnone_ITS_clr <- ps_filter(phyloseq_ITS_clr, input != "none", .target =  
"sample_data", .keep_all_taxa = T)
```

## Check resulting phyloseq objects

```
ps_rmnone_16s_clr
```

```
phyloseq-class experiment-level object
```

```
otu_table() OTU Table: [ 20236 taxa and 96 samples ]  
sample_data() Sample Data: [ 96 samples by 9 sample variables ]  
tax_table() Taxonomy Table: [ 20236 taxa by 10 taxonomic ranks ]  
phy_tree() Phylogenetic Tree: [ 20236 tips and 20235 internal nodes ]
```

```
ps_rmnone_ITS_clr
```

```
phyloseq-class experiment-level object
```

```
otu_table() OTU Table: [ 3845 taxa and 96 samples ]  
sample_data() Sample Data: [ 96 samples by 9 sample variables ]  
tax_table() Taxonomy Table: [ 3845 taxa by 10 taxonomic ranks ]  
phy_tree() Phylogenetic Tree: [ 3845 tips and 3843 internal nodes ]
```

## Extract metadata from new phyloseq object

```
metadata_rmnone_16s_clr <- as(sample_data(ps_rmnone_16s_clr), "data.frame")
```

```
metadata_rmnone_ITS_clr <- as(sample_data(ps_rmnone_ITS_clr), "data.frame")
```

## Calculate distance matrix

```
dist_bray_rmnone_16s_clr <- phyloseq::distance(ps_rmnone_16s_clr,  
method="euclidean")
```

```
dist_bray_rmnone_ITS_clr <- phyloseq::distance(ps_rmnone_ITS_clr,  
method="euclidean")
```

## Perform PERMANOVA with Adonis2

```
set.seed(100)
```

```
adonis_CLR_16s <- adonis2(dist_bray_rmnone_16s_clr ~ block + input * drought  
* species,  
data = metadata_rmnone_16s_clr,  
permutations = 9999, method = "euclidean", by = "terms" )
```

```
adonis_CLR_16s
```

```
Permutation test for adonis under reduced model
```

```
Terms added sequentially (first to last)
```

```
Permutation: free
```

```
Number of permutations: 9999
```

```
adonis2(formula = dist_bray_rmnone_16s_clr ~ block + input * drought *  
species, data = metadata_rmnone_16s_clr, permutations = 9999, method =  
"euclidean", by = "terms")
```

	Df	SumOfSqs	R2	F	Pr(>F)	
block	3	36587	0.03650	1.2264	0.0001	***
input	3	63420	0.06327	2.1259	0.0001	***
drought	1	15107	0.01507	1.5192	0.0002	***
species	2	20467	0.02042	1.0291	0.1876	
input:drought	3	40192	0.04010	1.3473	0.0001	***
input:species	6	60313	0.06017	1.0109	0.2915	
drought:species	2	20121	0.02007	1.0117	0.3042	
input:drought:species	6	60013	0.05987	1.0058	0.3470	
Residual	69	686140	0.68452			
Total	95	1002359	1.00000			

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis_CLR_ITS <- adonis2(dist_bray_rmnone_ITS_clr ~ block + input * drought  
* species,
```

```
data = metadata_rmnone_ITS_clr,  
permutations = 9999, method = "euclidean", by = "terms" )
```

```
adonis_CLR_ITS
```

```
Permutation test for adonis under reduced model
```

```
Terms added sequentially (first to last)
```

```
Permutation: free
```

```
Number of permutations: 9999
```

```
adonis2(formula = dist_bray_rmnone_ITS_clr ~ block + input * drought *  
species, data = metadata_rmnone_ITS_clr, permutations = 9999, method =  
"euclidean", by = "terms")
```

	Df	SumOfSqs	R2	F	Pr(>F)	
block	3	16499	0.03579	1.1840	0.0002	***
input	3	27994	0.06072	2.0090	0.0001	***
drought	1	5340	0.01158	1.1497	0.0184	*
species	2	9940	0.02156	1.0700	0.0585	.
input:drought	3	15362	0.03332	1.1024	0.0060	**
input:species	6	27977	0.06069	1.0039	0.4049	
drought:species	2	9152	0.01985	0.9852	0.6057	
input:drought:species	6	28262	0.06130	1.0141	0.2670	
Residual	69	320498	0.69519			
Total	95	461025	1.00000			

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Differential abundance analysis

### DeSeq2

Load DESeq package

Convert the phyloseq object to a DESeqDataSet object:

```
ds_16s <- phyloseq_to_deseq2(phyloseq_16s_3perc, ~ InputDrought)
```

```
converting counts to integer mode
```

```
ds_ITS <- phyloseq_to_deseq2(phyloseq_ITS_3perc, ~ InputDrought)
```

```
converting counts to integer mode
```

## Run DeSeq2

Based on comparisons between different difference abundance tools and settings in the literature, we chose for the poscounts setting and Wald test.

```
ds_16s <- DESeq(ds_16s, fitType = "local", sfType = "poscounts", test = "Wald")
```

```
estimating size factors
```

```
estimating dispersions
```

```
gene-wise dispersion estimates
```

```
mean-dispersion relationship
```

```
final dispersion estimates
```

```
fitting model and testing
```

```
-- replacing outliers and refitting for 11 genes
```

```
-- DESeq argument 'minReplicatesForReplace' = 7
```

```
-- original counts are preserved in counts(dds)
```

```
estimating dispersions
```

```
fitting model and testing
```

```
ds_ITS <- DESeq(ds_ITS, fitType = "local", sfType = "poscounts", test = "Wald")
```

```
estimating size factors
```

```
estimating dispersions
```

```
gene-wise dispersion estimates
```

```
mean-dispersion relationship
```

```
final dispersion estimates
```

```
fitting model and testing
```

```
-- replacing outliers and refitting for 1197 genes
```

```
-- DESeq argument 'minReplicatesForReplace' = 7
```

```
-- original counts are preserved in counts(dds)
```

```
estimating dispersions
```

```
fitting model and testing
```

### Extract results

```
ds_16s_results <- results(ds_16s, alpha = 0.05, contrast=c("InputDrought",  
"plant_D", "plant_WW"))  
glimpse(ds_16s_results)
```

```
Formal class 'DESeqResults' [package "DESeq2"] with 7 slots  
  ..@ priorInfo      : list()  
  ..@ rownames       : chr [1:20236] "ASV_13141" "ASV_20240" "ASV_16627"  
"ASV_18167" ...  
  ..@ nrows          : int 20236  
  ..@ elementType    : chr "ANY"  
  ..@ elementMetadata: Formal class 'DFrame' [package "S4Vectors"] with 6  
slots  
    ..@ metadata      :List of 6  
    .. ..$ filterThreshold: Named num 0.607  
    .. .. ..- attr(*, "names")= chr "63.97959%"  
    .. ..$ filterTheta   : num 0.64  
    .. ..$ filterNumRej  : 'data.frame': 50 obs. of 2 variables:  
    .. ..$ lo.fit        :List of 2  
    .. ..$ alpha         : num 0.05  
    .. ..$ lfcThreshold  : num 0  
  ..@ listData        :List of 6  
    .. ..$ baseMean      : num [1:20236] 0.2214 0.0305 0.0975 0.0576 2.2103 ...  
    .. ..$ log2FoldChange: num [1:20236] 0.459 -0.262 -0.263 -0.022 0.285 ...  
    .. ..$ lfcSE         : num [1:20236] 1.965 5.948 5.341 5.948 0.476 ...  
    .. ..$ stat          : num [1:20236] 0.23348 -0.04413 -0.04915 -0.00371  
0.5995 ...  
    .. ..$ pvalue        : num [1:20236] 0.815 0.965 0.961 0.997 0.549 ...  
    .. ..$ padj          : num [1:20236] NA NA NA NA 0.805 ...
```

```
nrow(ds_16s_results)
```

```
[1] 20236
```

```
ds_ITS_results <- results(ds_ITS, alpha = 0.05, contrast=c("InputDrought",  
"plant_D", "plant_WW"))  
glimpse(ds_ITS_results)
```

```
Formal class 'DESeqResults' [package "DESeq2"] with 7 slots  
  ..@ priorInfo      : list()  
  ..@ rownames       : chr [1:3845] "ASV_0652" "ASV_1034" "ASV_1967"  
"ASV_2137" ...  
  ..@ nrows          : int 3845  
  ..@ elementType    : chr "ANY"  
  ..@ elementMetadata: Formal class 'DFrame' [package "S4Vectors"] with 6  
slots  
    ..@ metadata      :List of 6  
    .. ..$ filterThreshold: Named num 1.7  
    .. .. ..- attr(*, "names")= chr "69.61081%"  
    .. ..$ filterTheta   : num 0.696  
    .. ..$ filterNumRej  : 'data.frame': 50 obs. of 2 variables:
```

```

.. ..$ lo.fit          :List of 2
.. ..$ alpha          : num 0.05
.. ..$ lfcThreshold   : num 0
..@ listData          :List of 6
.. ..$ baseMean       : num [1:3845] 6.536 0.966 0.338 0.359 0 ...
.. ..$ log2FoldChange: num [1:3845] 3.577 -0.493 -0.252 -2.597 0 ...
.. ..$ lfcSE          : num [1:3845] 2.33 5.8 5.95 5.93 0 ...
.. ..$ stat           : num [1:3845] 1.5385 -0.085 -0.0424 -0.4378 0 ...
.. ..$ pvalue         : num [1:3845] 0.124 NA 0.966 0.662 1 ...
.. ..$ padj           : num [1:3845] 1 NA NA NA NA NA NA NA NA ...

```

```
nrow(ds_ITS_results)
```

```
[1] 3845
```

### Filter for significance

```
ds_16s_results_sign <- ds_16s_results[which(ds_16s_results$padj < 0.05), ]
nrow(ds_16s_results_sign)
```

```
[1] 765
```

```
ds_ITS_results_sign <- ds_ITS_results[which(ds_ITS_results$padj < 0.05), ]
nrow(ds_ITS_results_sign)
```

```
[1] 21
```

### Bind with taxonomy table

```
ds_16s_results_sign_tax <- cbind(as(ds_16s_results_sign, "data.frame"),
as(tax_table_16s[row.names(ds_16s_results_sign), ], "matrix"))
glimpse(ds_16s_results_sign_tax)
```

```
Rows: 765
```

```
Columns: 18
```

```

$ baseMean      <dbl> 1.3655857, 1.5068794, 13.5729942, 4.3384353,
2.4997952,...
$ log2FoldChange <dbl> -2.6604165, -4.1455513, -1.0066249, -1.1435751, -
1.6086...
$ lfcSE         <dbl> 0.7646455, 0.7041520, 0.2170853, 0.4037395, 0.5661261,
...
$ stat          <dbl> -3.479281, -5.887296, -4.637001, -2.832458, -2.841561,
...
$ pvalue        <dbl> 5.027612e-04, 3.925650e-09, 3.535009e-06, 4.619168e-
03,...
$ padj          <dbl> 8.080472e-03, 3.179340e-07, 1.275578e-04, 4.507244e-
02,...
$ ASV           <chr> "ASV_04120", "ASV_03782", "ASV_00522", "ASV_01592",
"AS...
$ taxonomy      <chr>
"Bacteria;Acidobacteriota;Vicinamibacteria;Vicinamibact...
$ domain        <chr> "Bacteria", "Bacteria", "Bacteria", "Bacteria",

```

```

"Bacter...
$ phylum      <chr> "Acidobacteriota", "Acidobacteriota",
"Acidobacteriota"...
$ class        <chr> "Viciniabacteria", "Viciniabacteria",
"Viciniabacter...
$ order        <chr> "Viciniabacterales", "Viciniabacterales",
"Viciniabiba...
$ family       <chr> "", "", "", "", "", "", "", "", "",
"Viciniabacteracea...
$ genus        <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", "",
"",...
$ species      <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", "",
"",...
$ SH           <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", "",
"",...
$ taxa         <chr> "Viciniabacterales", "Viciniabacterales",
"Viciniabiba...
$ rank         <chr> "order", "order", "order", "order", "order", "order",
"..."

ds_ITS_results_sign_tax <- cbind(as(ds_ITS_results_sign, "data.frame"),
as(tax_table_ITS[row.names(ds_ITS_results_sign), ], "matrix"))
glimpse(ds_ITS_results_sign_tax)

Rows: 21
Columns: 18
$ baseMean      <dbl> 4.192872, 7.619159, 71.154251, 9.800298, 62.390691,
23....
$ log2FoldChange <dbl> -20.6696266, 19.6470039, -8.2545634, 18.3042719, -
3.324...
$ lfcSE         <dbl> 5.9224580, 5.9197896, 0.8915564, 3.0819514, 0.9971682,
...
$ stat          <dbl> -3.490042, 3.318869, -9.258599, 5.939183, -3.333600,
5....
$ pvalue        <dbl> 4.829448e-04, 9.038294e-04, 2.071319e-20, 2.864467e-
09,...
$ padj          <dbl> 3.280648e-02, 4.910806e-02, 1.181687e-17, 6.536713e-
07,...
$ ASV           <chr> "ASV_0101", "ASV_0643", "ASV_0112", "ASV_0435",
"ASV_01...
$ taxonomy      <chr>
"k__Fungi;p__Basidiomycota;c__Agaricomycetes;o__Agarica...
$ domain        <chr> "Fungi", "Fungi", "Fungi", "Fungi", "Fungi", "Fungi",
"..."
$ phylum      <chr> "Basidiomycota", "Basidiomycota", "Basidiomycota",
"Asc...
$ class         <chr> "Agaricomycetes", "Agaricomycetes", "Agaricomycetes",
"..."
$ order         <chr> "Agaricales", "Agaricomycetes_ord_Incertae_sedis",

```

```

"Aga...
$ family      <chr> "Psathyrellaceae",
"Agaricomycetes_fam_Incertae_sedis",...
$ genus       <chr> "Coprinellus", "Akenomyces", "Conocybe", "", "", "",
""...
$ species     <chr> "", "Akenomyces costatus", "", "", "", "", "", "", "",
...
$ SH          <chr> "", "SH__SH1514733.08FU", "", "", "", "", "",
"SH__SH33...
$ taxa        <chr> "Coprinellus", "Akenomyces costatus", "Conocybe",
"Asco...
$ rank        <chr> "genus", "species", "genus", "phylum", "domain",
"domai...

```

## ALDEx2

Subset phyloseq to only include plant-conditioned soil

```

phyloseq_plant_16s <- ps_filter(phyloseq_16s_3perc, input == "plant", .target
= "sample_data")

phyloseq_plant_ITS <- ps_filter(phyloseq_ITS_3perc, input == "plant", .target
= "sample_data")

```

Extract ASV table and sample data

```

sample_data_16s_plant <- data.frame(sample_data(phyloseq_plant_16s))
otu_table_16s_plant <- as.data.frame(otu_table(phyloseq_plant_16s))

sample_data_ITS_plant <- data.frame(sample_data(phyloseq_plant_ITS))
otu_table_ITS_plant <- as.data.frame(otu_table(phyloseq_plant_ITS))

```

Transform drought back into a character, as factor doesn't work with the aldex algorithm

```

sample_data_16s_plant$drought <- as.character(sample_data_16s_plant$drought)
sample_data_ITS_plant$drought <- as.character(sample_data_ITS_plant$drought)

```

## Run ALDEx2

This takes quite some computing power. In the next chunk is the option to upload the output from the supplementary data and skip this step

```
set.seed(100)

aldex_16s <- aldex(otu_table_16s_plant, sample_data_16s_plant$drought,
mc.samples = 999, test = "t", CI = T, effect = T, denom = "iqlr")
write.csv(aldex_16s, "ALDEx2_out_16s.csv", row.names = T)

aldex_16s <- aldex(otu_table_16s_plant, sample_data_16s_plant$drought,
mc.samples = 999, test = "t", CI = T, effect = T, denom = "iqlr")
write.csv(aldex_16s, "ALDEx2_out_16s.csv", row.names = T)
```

## Load ALDEx2 outputs from file

```
aldex_16s <- read.csv("aldex_out_16s.csv")
glimpse(aldex_16s)
```

```
Rows: 18,571
Columns: 14
$ X          <chr> "ASV_13141", "ASV_20240", "ASV_16627", "ASV_18167",
"ASV_0...
$ rab.all    <dbl> -0.109290683, -0.737717088, -0.705380506, -0.591659066,
2....
$ rab.win.C  <dbl> -0.41186882, -0.64123228, -0.62625869, -0.63116109,
2.3634...
$ rab.win.D  <dbl> 0.1849079, -0.8193510, -0.7950961, -0.5599484, 2.4821335,
...
$ diff.btw   <dbl> 0.46861824, -0.18336779, -0.17248423, 0.07903462,
0.126705...
$ diff.win   <dbl> 4.0078435, 4.1229961, 4.2017593, 4.1850312, 2.7229596,
0.5...
$ effect     <dbl> 0.094880290, -0.038432379, -0.034835738, 0.016371000,
0.03...
$ effect.low <dbl> -3.826273, -4.399328, -4.361454, -4.010279, -5.311567, -
3....
$ effect.high <dbl> 4.907157, 4.108120, 4.027965, 3.940206, 4.953095,
3.368640...
$ overlap    <dbl> 0.4477105, 0.4801040, 0.4793041, 0.4914000, 0.4785043,
0.4...
$ we.ep      <dbl> 0.6276237, 0.6996517, 0.7079913, 0.7406830, 0.7927004,
0.8...
$ we.eBH     <dbl> 0.9973782, 0.9949648, 0.9901618, 0.9959439, 1.0000000,
0.9...
$ wi.ep      <dbl> 0.6020115, 0.7103666, 0.7091747, 0.7334539, 0.7145544,
0.8...
$ wi.eBH     <dbl> 0.9889918, 0.9871417, 0.9843526, 0.9913226, 0.9987394,
0.9...
```

```

aldex_ITS <- read.csv("aldex_out_ITS.csv")
glimpse(aldex_ITS)

Rows: 2,814
Columns: 14
$ X          <chr> "ASV_0652", "ASV_1034", "ASV_1967", "ASV_2137",
"ASV_2224"...
$ rab.all    <dbl> -0.5304270, -0.8355553, -0.8591801, -0.6834575, -
0.8549359...
$ rab.win.C  <dbl> -0.73755848, -0.72760423, -0.69863083, -0.74231949, -
0.997...
$ rab.win.D  <dbl> -0.3033881, -0.9297010, -1.0128888, -0.6088390, -
0.7075517...
$ diff.btw   <dbl> 0.732432648, -0.211600672, -0.320634632, 0.202072233,
0.38...
$ diff.win   <dbl> 4.953563, 4.313973, 4.177470, 5.118313, 4.833550,
4.771576...
$ effect     <dbl> 0.131750755, -0.044344706, -0.063159890, 0.031077314,
0.07...
$ effect.low <dbl> -3.241199, -4.051725, -4.190338, -3.603946, -3.708500, -
3....
$ effect.high <dbl> 4.731428, 3.876375, 3.864215, 4.133768, 4.894038,
4.558021...
$ overlap    <dbl> 0.4335133, 0.4770000, 0.4662000, 0.4828000, 0.4625075,
0.4...
$ we.ep      <dbl> 0.5393326, 0.6848585, 0.7025736, 0.7360411, 0.5959692,
0.6...
$ we.eBH     <dbl> 0.9993828, 0.9990119, 0.9993343, 1.0000000, 0.9989149,
0.9...
$ wi.ep      <dbl> 0.5730596, 0.6892978, 0.7018212, 0.7249538, 0.6334655,
0.6...
$ wi.eBH     <dbl> 0.9969311, 0.9981740, 0.9968479, 0.9999197, 0.9950714,
0.9...

```

### *Filter for significance*

```

aldex_16s_sign <- filter(aldex_16s, wi.eBH <= 0.05)
nrow(aldex_16s_sign)

```

```
[1] 94
```

```

aldex_ITS_sign <- filter(aldex_ITS, wi.eBH <= 0.05)
nrow(aldex_ITS_sign)

```

```
[1] 3
```

## ANCOM-II

Here we can reuse the same filtered phyloseq from the ALDEx2 workflow, which only contains plant-conditioned soil.

### *Run ANCOM-II*

This needs quite some computing power and we ran this part of the analysis on high-performance computing cluster. To skip this step the output can be loaded from the supplementary data in the next chunk.

```
ancom_out_16s <- ancom(phyloseq = phyloseq_plant_16s, main_var = "drought",  
prv_cut = 0, struc_zero = F, p_adj_method = "BH", alpha = 0.05)  
  
ancom_out_ITS <- ancom(phyloseq = phyloseq_plant_ITS, main_var = "drought",  
prv_cut = 0, struc_zero = F, p_adj_method = "BH", alpha = 0.05)
```

### *Load ANCOM output from file and extract results*

Too big of a file to relocate. Load extracted results in the next chunk.

```
ancom_out_16s <- read_rds("ancom_out_16s.RDS")  
res_ancom_16s <- ancom_out_16s$res  
  
ancom_out_ITS <- read_rds("ancom_out_ITS.RDS")  
res_ancom_ITS <- ancom_out_ITS$res
```

### *Load extracted ANCOM-II results*

```
res_ancom_16s <- read.csv("res_ancom_16s.csv")  
res_ancom_ITS <- read.csv("res_ancom_ITS.csv")
```

### *Filter for significance*

```
ancom_16s_sign <- filter(res_ancom_16s, detected_0.8 == TRUE)  
nrow(ancom_16s_sign)  
  
[1] 463  
  
ancom_ITS_sign <- filter(res_ancom_ITS, detected_0.8 == TRUE)  
nrow(ancom_ITS_sign)  
  
[1] 13
```

## Combined

Define significant ASVs for each method

```
ASVS_deseq_16s <- row.names(ds_16s_results_sign_tax)  
ASVs_aldex_16s <- aldex_16s_sign$X  
ASVs_ancom_16s <- ancom_16s_sign$taxon  
  
ASVS_deseq_ITS <- row.names(ds_ITS_results_sign_tax)
```

```
ASVs_aldex_ITS <- aldex_ITS_sign$X
ASVs_ancom_ITS <- ancom_ITS_sign$taxon
```

Make Venn diagram showing shared ASVs between methods and unique ASVs

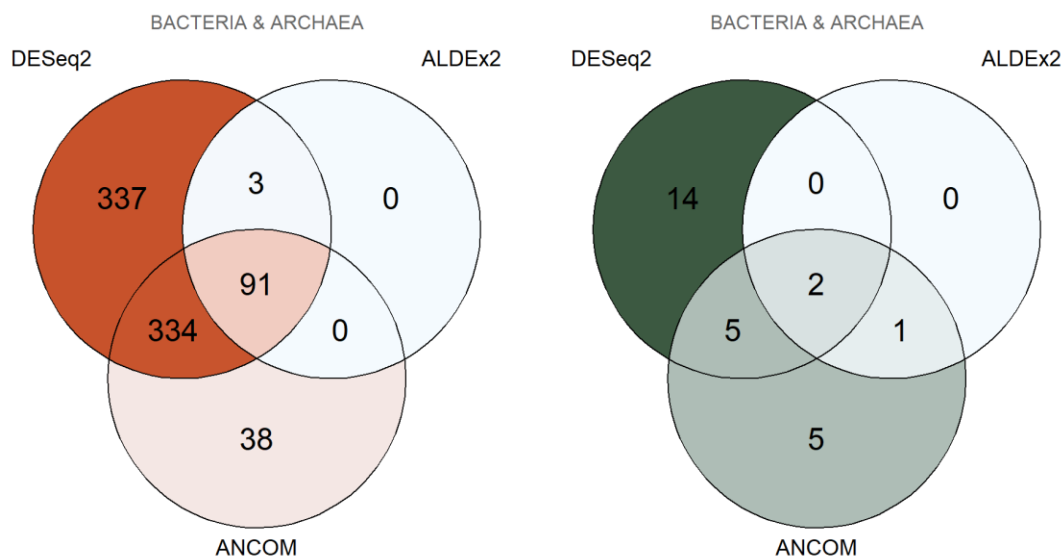
```
venn_list_16s <- list(DESeq2 = ASVs_deseq_16s, ALDEx2 = ASVs_aldex_16s,
ANCOCM = ASVs_ancom_16s)

P_venn_16s <- ggVennDiagram(venn_list_16s, label = "count", label_alpha = 0,
label_size = 14, set_size = 10)+
  scale_fill_gradient(low = "#F4FAFE", high = "#C7522B")+
  theme(legend.position = "none")+
  labs(title = "BACTERIA & ARCHAEA")+
  theme(plot.title = element_text(hjust = 0.5, size = 25, color = "grey40"))

venn_list_ITS <- list(DESeq2 = ASVs_deseq_ITS, ALDEx2 = ASVs_aldex_ITS,
ANCOCM = ASVs_ancom_ITS)

P_venn_ITS <- ggVennDiagram(venn_list_ITS, label = "count", label_alpha = 0,
label_size = 14, set_size = 10)+
  scale_fill_gradient(low = "#F4FAFE", high = "#3C5941")+
  theme(legend.position = "none")+
  labs(title = "BACTERIA & ARCHAEA")+
  theme(plot.title = element_text(hjust = 0.5, size = 25, color = "grey40"))

P_venn_combined <- ggarrange(P_venn_16s, P_venn_ITS, nrow = 1)
P_venn_combined
```



Save figure

```
ggsave(plot = P_venn_combined, "DA_venn.tiff", width = 50, height = 25, units
= "cm", dpi = 700)
```

## Make figure with most differentially abundant ASVs

Define ASVs detected by at least one, two or all 3 methods

```
conflicts_prefer(ggVennDiagram::overlap)
[conflicted] Will prefer ggVennDiagram::overlap over any other package.

conflicts_prefer(base::setdiff)
[conflicted] Will prefer base::setdiff over any other package.

ASVs_DA_1method_16s <- unique(c(ASVS_deseq_16s, ASVs_aldex_16s,
ASVs_ancom_16s))
ASVs_DA_2methods_16s <- unique(c(overlap(Venn(venn_list_16s), slice =
c(2,3)), overlap(Venn(venn_list_16s), slice = c(1,3)),
overlap(Venn(venn_list_16s), slice = c(1,2))))
ASVs_DA_3methods_16s <- overlap(Venn(venn_list_16s), slice = "all")

ASVs_DA_1method_ITS <- unique(c(ASVS_deseq_ITS, ASVs_aldex_ITS,
ASVs_ancom_ITS))
ASVs_DA_2methods_ITS <- unique(c(overlap(Venn(venn_list_ITS), slice =
c(2,3)), overlap(Venn(venn_list_ITS), slice = c(1,3)),
overlap(Venn(venn_list_ITS), slice = c(1,2))))
ASVs_DA_3methods_ITS <- overlap(Venn(venn_list_ITS), slice = "all")
```

Filter aldex output by ASV list

```
aldex_16s_sign_3methods <- aldex_16s %>% filter(X %in% ASVs_DA_3methods_16s)
aldex_ITS_sign_3methods <- aldex_ITS %>% filter(X %in% ASVs_DA_3methods_ITS)
```

Merge with taxonomy table

```
aldex_16s_sign_3methods_tax <- merge(tax_table_16s, aldex_16s_sign_3methods,
by.y = "X", by.x = "ASV")
aldex_ITS_sign_3methods_tax <- merge(tax_table_ITS, aldex_ITS_sign_3methods,
by.y = "X", by.x = "ASV")
```

Select columns for full table

```
tax_table_aldex_3methods_16s <- aldex_16s_sign_3methods_tax %>%
dplyr::select(ASV, domain, phylum, class, order, family, genus, species,
taxa, rank, effect, effect.low, effect.high, wi.eBH)

tax_table_aldex_3methods_ITS <- aldex_ITS_sign_3methods_tax %>%
dplyr::select(ASV, domain, phylum, class, order, family, genus, species,
taxa, rank, effect, effect.low, effect.high, wi.eBH)
```

Sort by p.adj based on Wilcoxon tests and select the first 20

```
aldex_16s_sign_3methods_tax_top20 <- arrange(tax_table_aldex_3methods_16s,
wi.eBH)[1:20,]
```

Fuse with ITS table

```
aldex_sign_top22 <- rbind(aldex_16s_sign_3methods_tax_top20,  
tax_table_aldex_3methods_ITS)
```

Order by significance

```
aldex_sign_top22$ASV <- factor(aldex_sign_top22$ASV,  
levels=aldex_sign_top22[order(aldex_sign_top22$wi.eBH, decreasing = T),  
"ASV"])
```

Make figure component 1: Taxonomy table

First, the taxon names which were just codes were extended with the next lower taxonomic rank. In a second step, we made a table based on the top 22 differentially abundant ASVs, sorted by their P-values, with two columns: the highest classified taxonomic rank and the corresponding taxon name.

```
aldex_sign_top22$taxa <- ifelse(aldex_sign_top22$family == "NS11-12 marine  
group", paste0(aldex_sign_top22$order, " ", aldex_sign_top22$family),  
aldex_sign_top22$taxa)  
aldex_sign_top22$taxa <- ifelse(aldex_sign_top22$genus == "MND1",  
paste0(aldex_sign_top22$family, " ", aldex_sign_top22$genus),  
aldex_sign_top22$taxa)  
aldex_sign_top22$taxa <- ifelse(aldex_sign_top22$order == "CCD24",  
paste0(aldex_sign_top22$class, " ", aldex_sign_top22$order),  
aldex_sign_top22$taxa)  
aldex_sign_top22$taxa <- ifelse(aldex_sign_top22$order == "PLTA13",  
paste0(aldex_sign_top22$class, " ", aldex_sign_top22$order),  
aldex_sign_top22$taxa)  
aldex_sign_top22$taxa <- ifelse(aldex_sign_top22$order == "R7C24",  
paste0(aldex_sign_top22$class, " ", aldex_sign_top22$order),  
aldex_sign_top22$taxa)  
aldex_sign_top22$taxa <- ifelse(aldex_sign_top22$phylum == "MBNT15",  
paste0(aldex_sign_top22$domain, " ", aldex_sign_top22$phylum),  
aldex_sign_top22$taxa)  
aldex_sign_top22$taxa <- ifelse(aldex_sign_top22$family == "TRA3-20",  
paste0(aldex_sign_top22$order, " ", aldex_sign_top22$family),  
aldex_sign_top22$taxa)
```

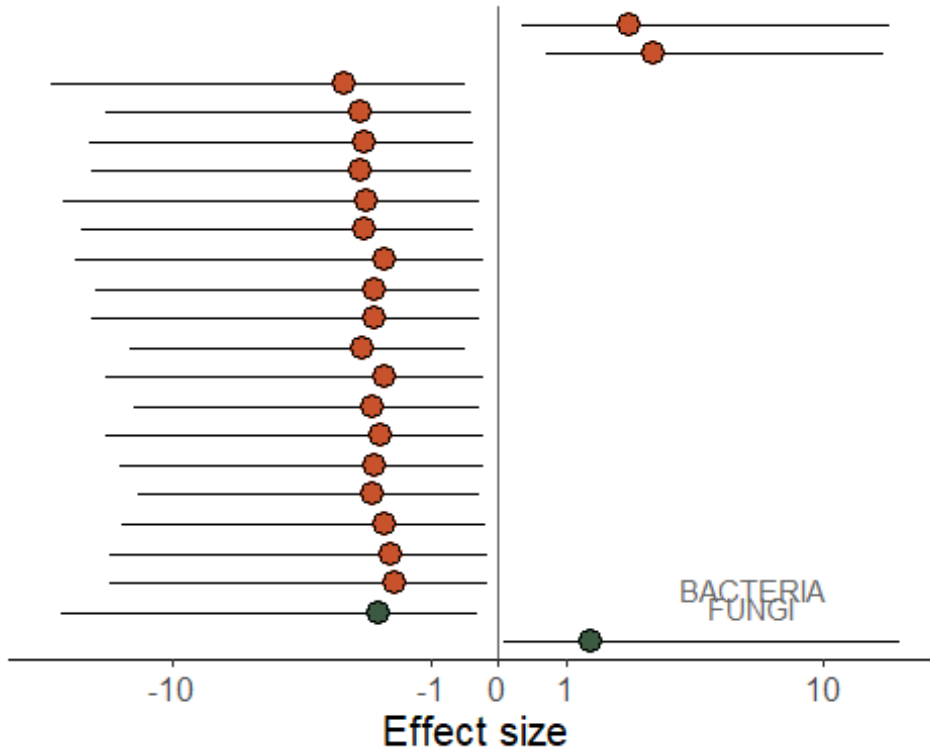
```
T_taxa <- ggplot(data = aldex_sign_top22)+
  geom_text(aes(x = ASV, y = 0.9, label = taxa), fontface = "italic" , hjust
= 0)+
  geom_text(aes(x = ASV, y = 2.16, label = rank), hjust = 0)+
  ylim(0.9, 2.3)+
  mytheme+
  theme(axis.title.y = element_blank(), axis.title.x = element_text(size =
15, color = "white"), axis.text.x = element_text(color = "white"),
axis.text.y = element_blank(),
      axis.line.x = element_blank(), axis.line.y = element_blank(),
axis.ticks = element_blank()+
  coord_flip()
```

Make figure component 2: Effect size

Here we are displaying the effect size based on the ALDEx2 method and the corresponding confidence intervals.

```
fill_eff <- c(rep("#C7522B",20), rep("#3C5941", 2))

P_eff <- ggplot(data = aldex_sign_top22, aes( x = ASV, y = effect))+
  geom_hline(yintercept = 0, col = "gray40", linewidth = 0.3)+
  geom_errorbar(aes(ymin = effect.low, ymax = effect.high), width = 0)+
  geom_point(shape = 21, size = 4, fill = fill_eff)+
  annotate("text", x = c(2.2, 2.8), y = c(6, 6), label = c("FUNGI",
"BACTERIA"), col = "grey40")+
  labs( y = "Effect size")+
  mytheme +
  theme(axis.title.y = element_blank(), axis.text.y = element_blank(),
axis.line.y = element_blank(), axis.ticks.y = element_blank(),
      axis.title.x = element_text(size = 15), legend.position = "none")+
  coord_flip()+
  scale_y_continuous(transform = "pseudo_log", breaks = c( -10, -1, 0, 1,
10))
P_eff
```



Make figure component 3: Abundance (log scale)

Here we are displaying the abundance of the selected 22 ASVs in plant-conditioned soil with or without drought history on a log scale. To replicate the order and orientation of the other two figure components some adjustments to the data need to be made. The data is acquired by “melting” the phyloseq object into a long-format data frame, filtering for the selected ASVs and sorting them into the same order as in the other two figure components.

```
phyloseq_plant_16s_melt <- psmelt(phyloseq_plant_16s) %>%
  dplyr::select(!OTU)
```

```
Warning in psmelt(phyloseq_plant_16s): The sample variables:
species
  have been renamed to:
sample_species
to avoid conflicts with taxonomic rank names.
```

```
phyloseq_plant_ITS_melt <- psmelt(phyloseq_plant_ITS) %>%
  dplyr::select(!OTU)
```

```
Warning in psmelt(phyloseq_plant_ITS): The sample variables:
species
  have been renamed to:
sample_species
to avoid conflicts with taxonomic rank names.
```

```

phyloseq_plant_16s_melt_sign <- filter(phyloseq_plant_16s_melt , ASV %in%
aldex_16s_sign_3methods_tax_top20$ASV)
phyloseq_plant_ITS_melt_sign <- filter(phyloseq_plant_ITS_melt, ASV %in%
aldex_ITS_sign_3methods$X)

phyloseq_plant_melt_sign <- rbind(phyloseq_plant_16s_melt_sign,
phyloseq_plant_ITS_melt_sign)

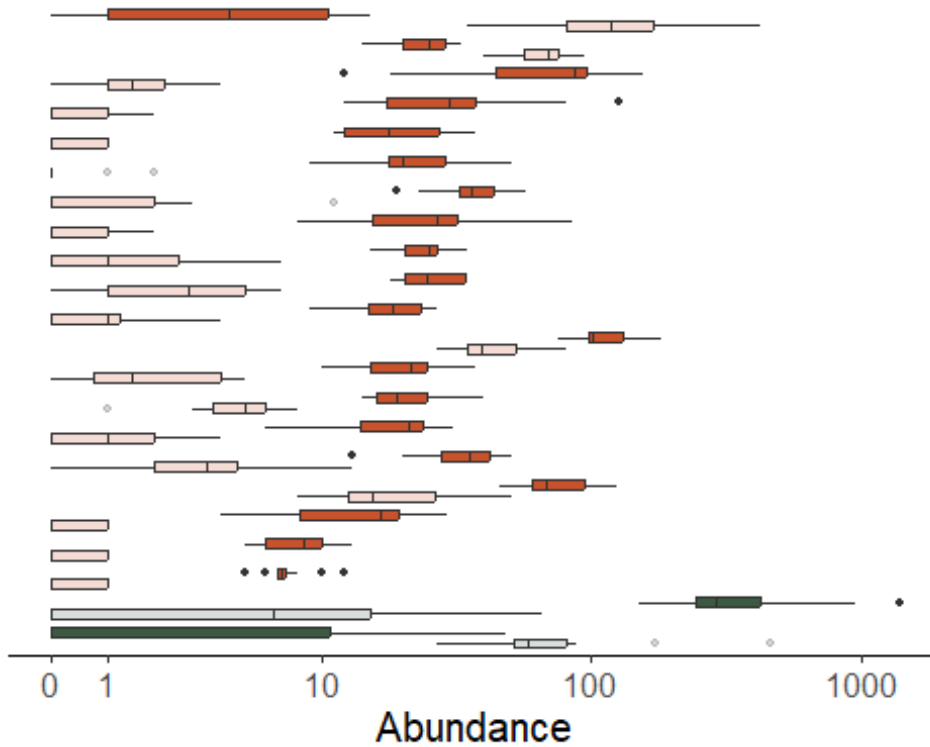
order_ASVs <- rev(levels(aldex_sign_top22$ASV))

phyloseq_plant_melt_sign <- phyloseq_plant_melt_sign %>% mutate(ASV =
factor(ASV, levels = rev(order_ASVs))) %>% mutate(drought = factor(drought,
levels = c("D", "WW")))

fill_ab <- c(rep("#3C5941",4), rep("#C7522B", 40))

P_ab <- ggplot(phyloseq_plant_melt_sign, aes(x = ASV, y = Abundance, alpha =
drought))+
  geom_boxplot(linewidth = 0.5, fatten = 1, fill = fill_ab, outlier.size = 1
)+
  scale_alpha_manual(values = c(0.2, 1), name = "Drought\ntreatment", labels
= c("well-watered", 'drought'))+
  labs(y = "Abundance")+
  mytheme+
  theme( axis.title.x = element_text(size = 15), legend.position = "none",
axis.title.y = element_blank(), axis.text.y = element_blank(),
axis.line.y = element_blank(), axis.ticks.y =
element_blank()) + scale_y_continuous(transform = "pseudo_log",
breaks = c(0, 1, 10, 100, 1000)) +
  coord_flip()
P_ab

```



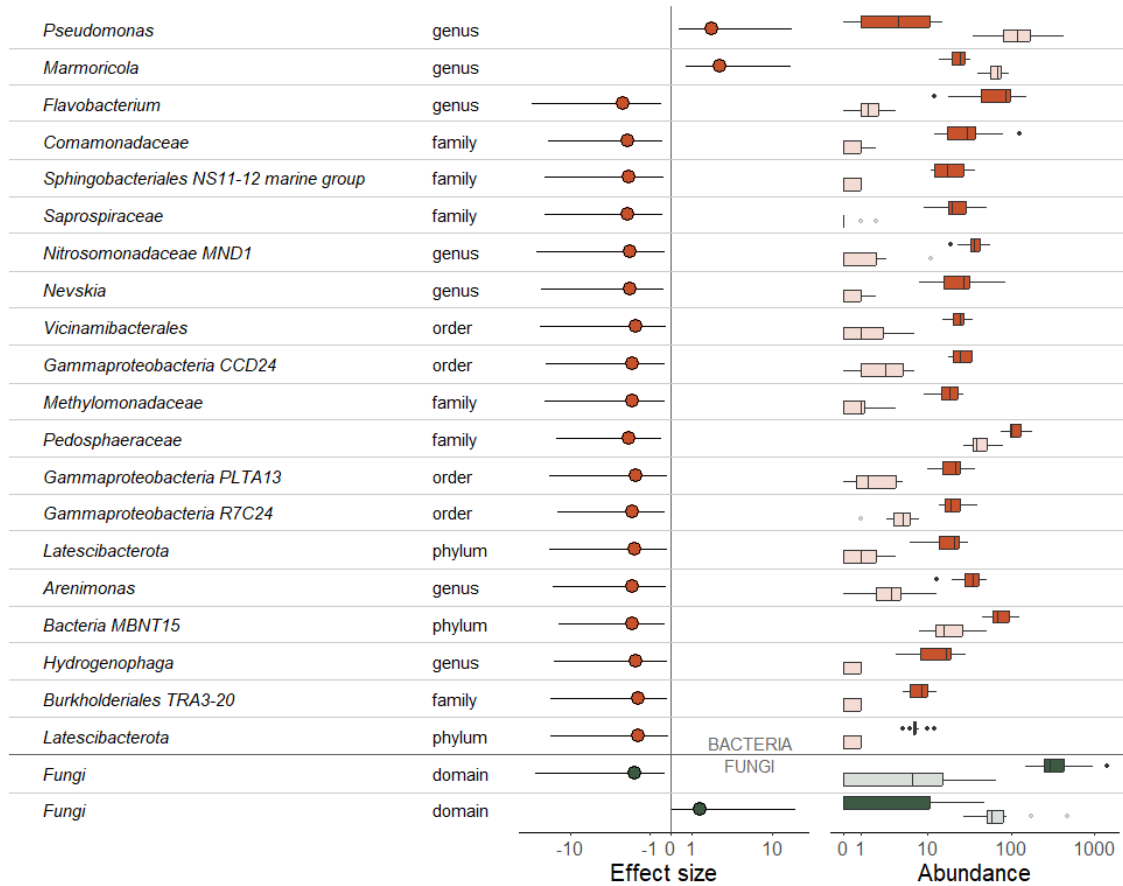
Create combined figure

Here, we are fusing the components and adding some details to make the figure more structured.

```
P <- ggarrange(T_taxa, P_eff, P_ab, nrow = 1, widths = c(2, 1.25, 1.25) )

P_DifAb <- P + geom_hline(yintercept = c(0.113, 0.197, 0.239, 0.28, 0.321,
0.363, 0.405, 0.446, 0.487, 0.529, 0.57, 0.612, 0.654, 0.695, 0.738, 0.778,
0.82, 0.862, 0.903, 0.944),
                        col = "gray80", linewidth = 0.3)+
  geom_hline(yintercept = 0.155, col = "grey40", linewidth = 0.3)

P_DifAb
```



Save figure

```
ggsave(plot = P_DifAb, "DifAb.tiff", width = 22, height = 20, units = "cm",
dpi = 700)
```

## Indicator species analysis

Merge ASV table and sample data by sample names

```
merged_plant_16s <- merge(sample_data_16s_plant, t(otu_table_16s_plant), by =
"row.names")
merged_plant_ITS <- merge(sample_data_ITS_plant, t(otu_table_ITS_plant), by =
"row.names")
```

Run indicator species analysis

```
set.seed(100)
inv_rg_16s <- multipatt(merged_plant_16s[, -c(1:11)],
merged_plant_16s$species, func = "r.g", control = how(nperm=999))
inv_rg_ITS <- multipatt(merged_plant_ITS[, -c(1:11)],
merged_plant_ITS$species, func = "r.g", control = how(nperm=999))
```

## Create output table

```
inv_rg_16s_values <- inv_rg_16s$str
inv_rg_16s_sign <- inv_rg_16s$sign
inv_rg_16s_merged <- merge(inv_rg_16s_sign, inv_rg_16s_values, by =
"row.names")

inv_rg_ITS_values <- inv_rg_ITS$str
inv_rg_ITS_sign <- inv_rg_ITS$sign
inv_rg_ITS_merged <- merge(inv_rg_ITS_sign, inv_rg_ITS_values, by =
"row.names")
```

## Filter for significance

```
indispecies_alpha0.1_16s <- filter(inv_rg_16s_merged, p.value <= 0.1)
nrow(indispecies_alpha0.1_16s)

[1] 817

indispecies_alpha0.1_ITS <- filter(inv_rg_ITS_merged, p.value <= 0.1)
nrow(indispecies_alpha0.1_ITS)

[1] 154
```

## Extract ASVs associated with each species

```
ISA_ASVs_16s <- indispecies_alpha0.1_16s$Row.names
ISA_ASVs_ITS <- indispecies_alpha0.1_ITS$Row.names

ISA_ASVs_Ao_16s <- filter(indispecies_alpha0.1_16s, s.Ao == 1)$Row.names
ISA_ASVs_Dg_16s <- filter(indispecies_alpha0.1_16s, s.Dg == 1)$Row.names
ISA_ASVs_Ra_16s <- filter(indispecies_alpha0.1_16s, s.Ra == 1)$Row.names

ISA_ASVs_Ao_ITS <- filter(indispecies_alpha0.1_ITS, s.Ao == 1)$Row.names
ISA_ASVs_Dg_ITS <- filter(indispecies_alpha0.1_ITS, s.Dg == 1)$Row.names
ISA_ASVs_Ra_ITS <- filter(indispecies_alpha0.1_ITS, s.Ra == 1)$Row.names
```

## Make Venn diagram

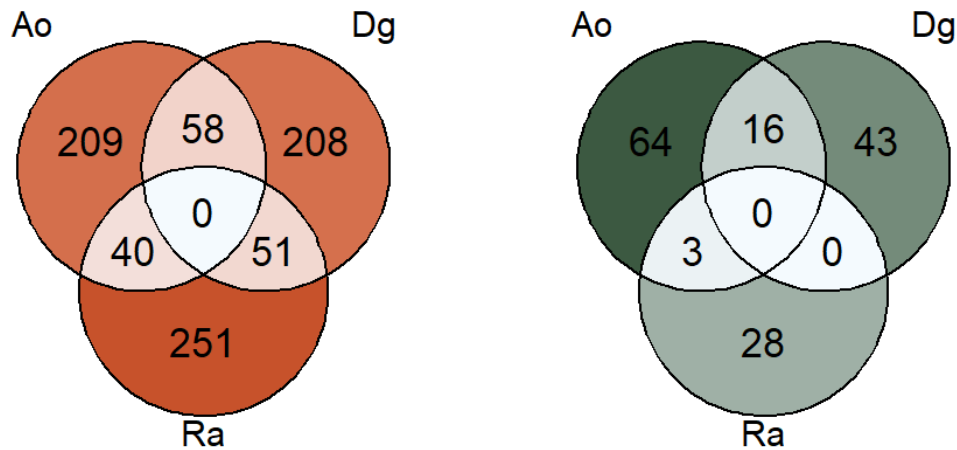
```
venn_list_ISA_16s <- list("Ao" = ISA_ASVs_Ao_16s, "Dg" = ISA_ASVs_Dg_16s,
"Ra" = ISA_ASVs_Ra_16s)

Venn_ISA_16s <- ggVennDiagram(venn_list_ISA_16s, label = "count", label_alpha
= 0, label_size = 9, set_size = 8)+
  scale_fill_gradient(low = "#F4FAFE", high = "#C7522B")+
  theme(legend.position = "none")

venn_list_ISA_ITS <- list("Ao" = ISA_ASVs_Ao_ITS, "Dg" = ISA_ASVs_Dg_ITS,
"Ra" = ISA_ASVs_Ra_ITS)
```

```
Venn_ISA_ITS <- ggVennDiagram(venn_list_ISA_ITS, label = "count", label_alpha
= 0, label_size = 9, set_size = 8)+
  scale_fill_gradient(low = "#F4FAFE", high = "#3C5941")+
  theme(legend.position = "none")
```

```
P_ISA_venn <- ggarrange(Venn_ISA_16s, Venn_ISA_ITS, nrow = 1)
P_ISA_venn
```



Save figure

```
ggsave(plot = P_ISA_venn, "ISA_venn.tiff", width = 25, height = 15, units =
"cm", dpi = 700)
```

Add species associations to ISA table

```
indispecies_alpha0.1_16s$association <- ifelse(indispecies_alpha0.1_16s$index
== 1, "Ao", "Dg" )
indispecies_alpha0.1_16s$association <-
ifelse(indispecies_alpha0.1_16s$index == 3, "Ra",
indispecies_alpha0.1_16s$association)
indispecies_alpha0.1_16s$association <-
ifelse(indispecies_alpha0.1_16s$index == 4, "Ao & Dg",
indispecies_alpha0.1_16s$association)
indispecies_alpha0.1_16s$association <-
ifelse(indispecies_alpha0.1_16s$index == 5, "Ao & Ra",
indispecies_alpha0.1_16s$association)
indispecies_alpha0.1_16s$association <-
ifelse(indispecies_alpha0.1_16s$index == 6, "Dg & Ra",
indispecies_alpha0.1_16s$association)
```

```
indispecies_alpha0.1_ITS$association <- ifelse(indispecies_alpha0.1_ITS$index
== 1, "Ao", "Dg" )
indispecies_alpha0.1_ITS$association <-
ifelse(indispecies_alpha0.1_ITS$index == 3, "Ra",
indispecies_alpha0.1_ITS$association)
indispecies_alpha0.1_ITS$association <-
ifelse(indispecies_alpha0.1_ITS$index == 4, "Ao & Dg",
```

```

indispecies_alpha0.1_ITS$association)
indispecies_alpha0.1_ITS$association <-
ifelse(indispecies_alpha0.1_ITS$index == 5, "Ao & Ra",
indispecies_alpha0.1_ITS$association)
indispecies_alpha0.1_ITS$association <-
ifelse(indispecies_alpha0.1_ITS$index == 6, "Dg & Ra",
indispecies_alpha0.1_ITS$association)

```

Merge with tax table

```

indispecies_alpha0.1_16s <- indispecies_alpha0.1_16s %>% dplyr::rename("ASV"
= Row.names)
indispecies_alpha0.1_16s_tax <- merge(indispecies_alpha0.1_16s,
tax_table_16s)

indispecies_alpha0.1_ITS <- indispecies_alpha0.1_ITS %>% dplyr::rename("ASV"
= Row.names)
indispecies_alpha0.1_ITS_tax <- merge(indispecies_alpha0.1_ITS,
tax_table_ITS)

```

## Coda4microbiome analysis

### For plant biomass

#### *Prepare data*

Change soilID variable in feedback data set to the same format as the sequencing data

```

data_feedback$soil_ID <- as.factor(paste("B", data_feedback$block, "-",
data_feedback$species_soil, "-",
data_feedback$drought, "-",
data_feedback$input, sep = ""))

data_feedback <- data_feedback %>% mutate(across('soil_ID', str_replace,
'WW', 'C')) %>% mutate(across('soil_ID', str_replace, 'plant', 'soil'))

```

Merge feedback data with ASV table for plant-conditioned soil

```

merged_feedback_otu_16s <- merge(data_feedback, t(otu_table_16s_plant), by.x
= "soil_ID", by.y = "row.names" )

merged_feedback_otu_ITS <- merge(data_feedback, t(otu_table_ITS_plant), by.x
= "soil_ID", by.y = "row.names" )

```

Filter by ASVs which were determined at least by one methods as differentially abundant in droughted soil

```

merged_feedback_otu_16s_coda <- dplyr::select(merged_feedback_otu_16s , 1:13,
any_of(ASVs_DA_1method_16s))

```

```
merged_feedback_otu_ITS_coda <- dplyr::select(merged_feedback_otu_ITS , 1:13,
any_of(ASVs_DA_1method_ITS))
```

### *Determine coda inputs*

X: ASV table with the selected ASVs from the DA analysis  
Y: Response variable, in this case plant biomass in the feedback phase  
Covariates: Variation explained by block and plant species is taken into account

```
X_totalDW_16s <-
as.matrix(data.frame(merged_feedback_otu_16s_coda[14:length(merged_feedback_o
tu_16s_coda)], row.names = row.names(merged_feedback_otu_16s_coda)))
X_totalDW_16s_no0 <- impute_zeros(X_totalDW_16s)
```

```
Y_totalDW_16s <- structure(as.numeric(merged_feedback_otu_16s_coda$totalDW),
names = row.names(merged_feedback_otu_16s_coda))
```

```
covariates_16s <-
as.matrix(data.frame(merged_feedback_otu_16s_coda[c("block",
"species_home")], row.names = row.names(merged_feedback_otu_16s_coda)))
```

```
X_totalDW_ITS <-
as.matrix(data.frame(merged_feedback_otu_ITS_coda[14:length(merged_feedback_o
tu_ITS_coda)], row.names = row.names(merged_feedback_otu_ITS_coda)))
X_totalDW_ITS_no0 <- impute_zeros(X_totalDW_ITS)
```

```
Y_totalDW_ITS <- structure(as.numeric(merged_feedback_otu_ITS_coda$totalDW),
names = row.names(merged_feedback_otu_ITS_coda))
```

```
covariates_ITS <-
as.matrix(data.frame(merged_feedback_otu_ITS_coda[c("block",
"species_home")], row.names = row.names(merged_feedback_otu_ITS_coda)))
```

### *Run coda4microbiome*

Very computing intense, we ran it on the server. The next chunk contains the code to load the output.

```
coda_totalDW_16s <- coda_glmnet(x=X_totalDW_16s_no0 ,y=Y_totalDW_16s, lambda
= "lambda.1se", covar = covariates_16s)
```

```
coda_totalDW_ITS <- coda_glmnet(x=X_totalDW_ITS_no0 ,y=Y_totalDW_ITS, lambda
= "lambda.1se", covar = covariates_ITS)
```

Load coda output

```
coda_totalDW_16s <- read_rds("coda_totalDW_16s.RDS")
```

```
coda_totalDW_ITS <- read_rds("coda_totalDW_ITS.RDS")
```

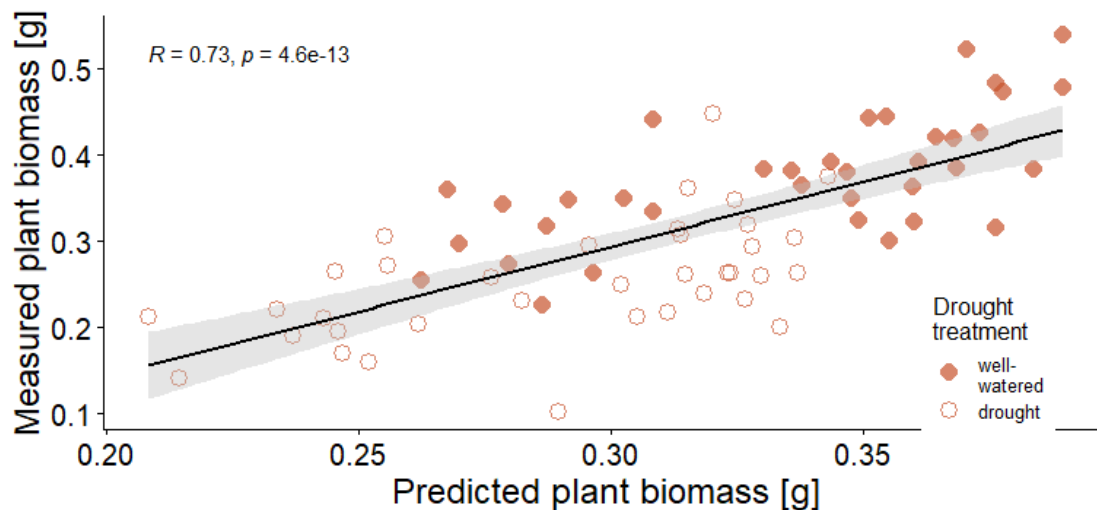
## Extract plots

```
P_sig_16s_DW <- coda_totalDW_16s$`signature plot`  
P_pred_16s_DW <- coda_totalDW_16s$`predictions plot`  
  
P_sig_ITS_DW <- coda_totalDW_ITS$`signature plot`  
P_pred_ITS_DW <- coda_totalDW_ITS$`predictions plot`
```

## Prediction plots

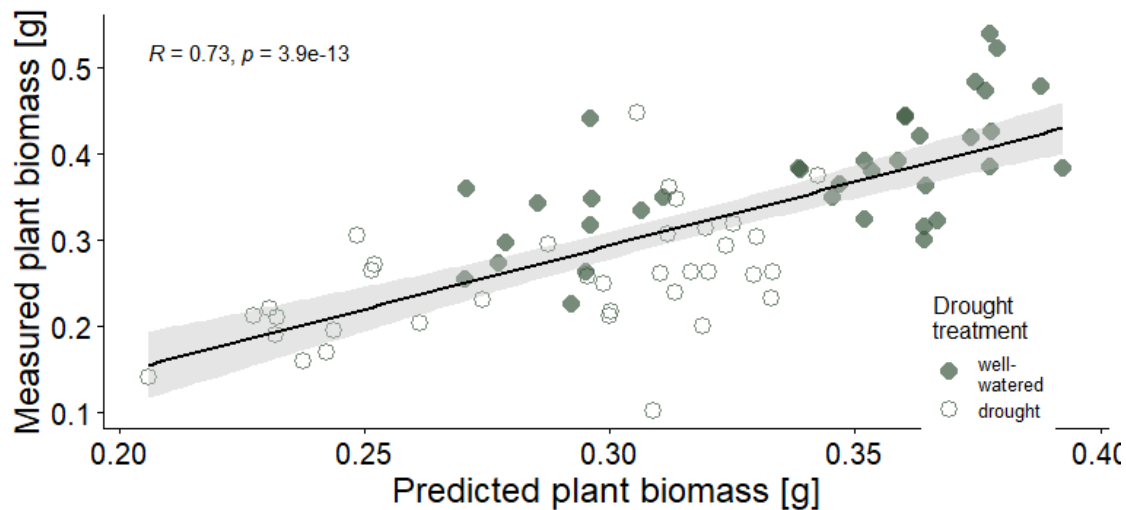
### Prediction plot 16s

```
P_pred_16s_DW_2 <- P_pred_16s_DW + geom_point(col = "white", size = 3)+  
  geom_smooth(method = "lm", col = "white", fill = "white", alpha = 1)+  
  geom_point(aes(shape = merged_feedback_otu_16s_coda$drought), col =  
  "#C7522B", size = 4, alpha = 0.7)+  
  scale_shape_manual(values = c(16, 1), name = "Drought\ntreatment", labels  
  = c("well-\nwatered", "drought"))+  
  geom_smooth(method = "lm", col = "black", fill = "grey", alpha = 0.4)+  
  labs(x = "Predicted plant biomass [g]", y = "Measured plant biomass [g]",  
  title = "")+  
  theme(plot.title = element_text(hjust = 0.5, size = 18), axis.title =  
  element_text(size = 18), legend.position = "inside",  
  legend.position.inside = c(0.88, 0.17), axis.text = element_text(size  
  = 15))  
P_pred_16s_DW_2
```



## Predictions plot ITS

```
P_pred_ITS_DW_2 <- P_pred_ITS_DW + geom_point(col = "white", size = 3)+
  geom_smooth(method = "lm", col = "white", fill = "white", alpha = 1)+
  geom_point(aes(shape = merged_feedback_otu_ITS_coda$drought), col =
"#3C5941", size = 4, alpha = 0.7)+
  scale_shape_manual(values = c(16, 1), name = "Drought\ntreatment", labels
= c("well-\nwatered", "drought"))+
  geom_smooth(method = "lm", col = "black", fill = "grey", alpha = 0.4)+
  labs(x = "Predicted plant biomass [g]", y = "Measured plant biomass [g]",
title = "")+
  theme(plot.title = element_text(hjust = 0.5, size = 18), axis.title =
element_text(size = 18), legend.position = "inside",
legend.position.inside = c(0.88, 0.17), axis.text = element_text(size
= 15))
P_pred_ITS_DW_2
```



## Signature plots

### Signature plot 16s

```
aldex_DA_1method_16s <- filter(aldex_16s, X %in% ASVs_DA_1method_16s) %>%
  dplyr::rename("ASV" = X)
aldex_DA_1method_16s$direction <- ifelse(aldex_DA_1method_16s$effect > 0,
"+", "-")

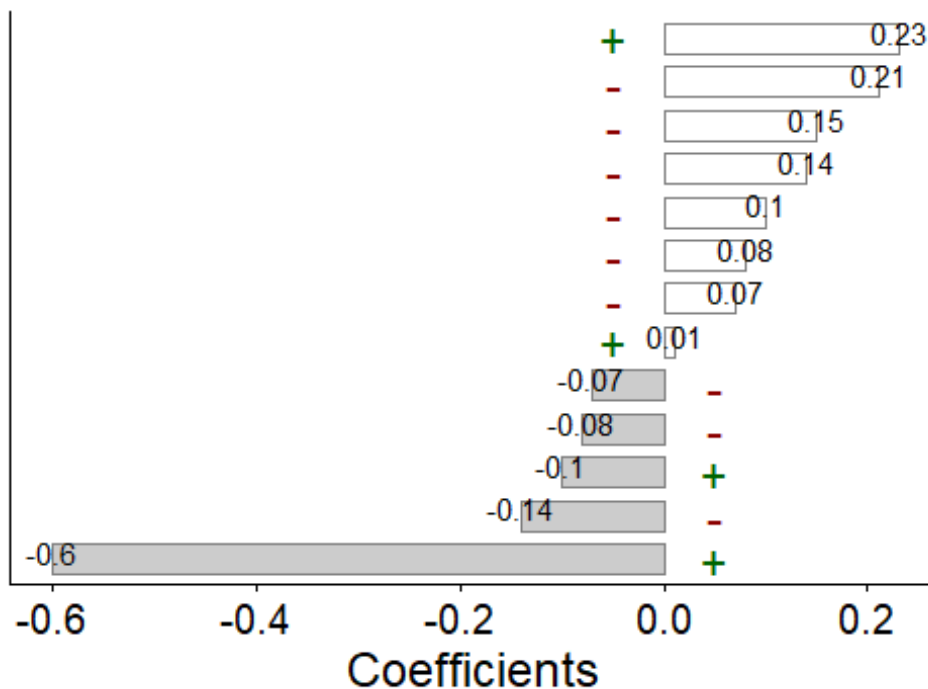
coda_DW_ASVs_tax_16s <- coda_totalDW_16s$`signature plot`$data %>%
  dplyr::rename("ASV" = vars) %>%
  merge(tax_table_16s)

coda_DW_ASVs_tax_16s <- merge(coda_DW_ASVs_tax_16s, aldex_DA_1method_16s)
```

```

P_sig_16s_DW_2 <- P_sig_16s_DW +
  scale_fill_manual(values = c("grey80", "white"))+
  geom_text(aes(y = ifelse(coeff < 0, 0.05, -0.05), x = vars, label =
coda_DW_ASVs_tax_16s$direction, col = coda_DW_ASVs_tax_16s$direction), size =
6)+
  scale_color_manual(values = c("+" = "darkgreen", "-" = "darkred"))+
  labs(y = "Coefficients", title = "")+
  theme(plot.title = element_text(hjust = 0.5, size = 18),
        axis.title = element_text(size = 18), axis.ticks.y = element_blank(),
        legend.position = "none", axis.text.y = element_blank(), axis.text.x
= element_text(size = 15))
P_sig_16s_DW_2

```



### Signature plot ITS

```

aldex_DA_1method_ITS <- filter(aldex_ITS, X %in% ASVs_DA_1method_ITS) %>%
  dplyr::rename("ASV" = X)
aldex_DA_1method_ITS$direction <- ifelse(aldex_DA_1method_ITS$effect > 0,
"+", "-")

coda_DW_ASVs_tax_ITS <- coda_totalDW_ITS$`signature plot`$data %>%
  dplyr::rename("ASV" = vars) %>%
  merge(tax_table_ITS)

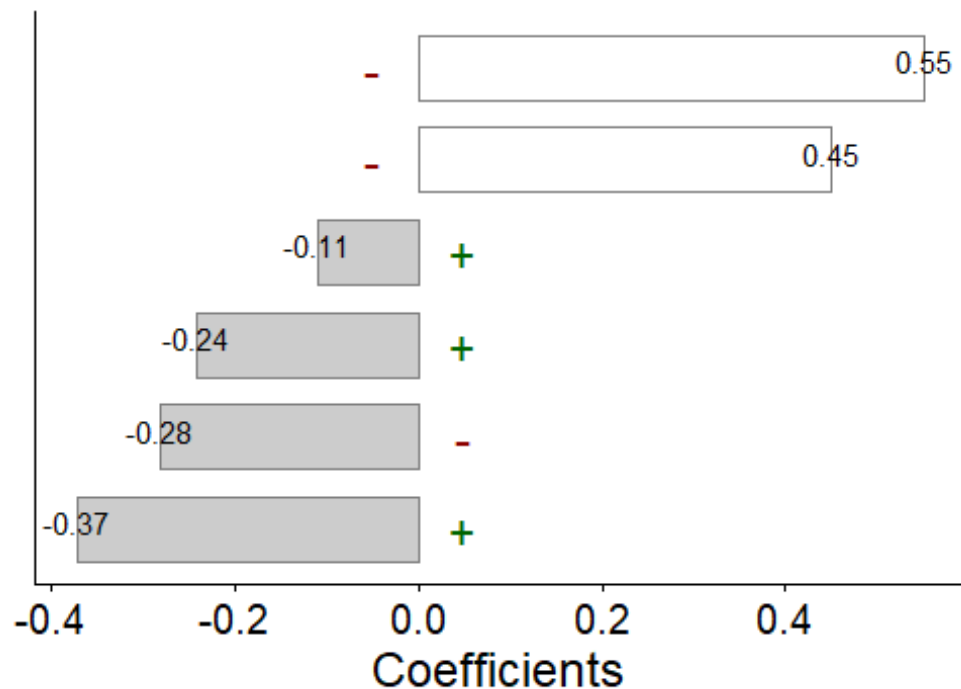
coda_DW_ASVs_tax_ITS <- merge(coda_DW_ASVs_tax_ITS, aldex_DA_1method_ITS)

```

```

P_sig_ITS_DW_2 <- P_sig_ITS_DW +
  scale_fill_manual(values = c("grey80", "white"))+
  geom_text(aes(y = ifelse(coeff < 0, 0.05, -0.05), x = vars, label =
coda_DW_ASVs_tax_ITS$direction, col = coda_DW_ASVs_tax_ITS$direction), size =
6)+
  scale_color_manual(values = c("+" = "darkgreen", "-" = "darkred"))+
  labs(y = "Coefficients", title = "")+
  theme(plot.title = element_text(hjust = 0.5, size = 18),
        axis.title = element_text(size = 18), axis.ticks.y = element_blank(),
        legend.position = "none", axis.text.y = element_blank(), axis.text.x
= element_text(size = 15))
P_sig_ITS_DW_2

```



### Taxon tables

Change taxon names

```

coda_DW_ASVs_tax_16s$taxa <- ifelse(coda_DW_ASVs_tax_16s$order == "CCD24",
paste0(coda_DW_ASVs_tax_16s$class, " ", coda_DW_ASVs_tax_16s$order),
coda_DW_ASVs_tax_16s$taxa)
coda_DW_ASVs_tax_16s$taxa <- ifelse(coda_DW_ASVs_tax_16s$genus == "OLB12",
paste0(coda_DW_ASVs_tax_16s$family, " ", coda_DW_ASVs_tax_16s$genus),
coda_DW_ASVs_tax_16s$taxa)
coda_DW_ASVs_tax_16s$taxa <- ifelse(coda_DW_ASVs_tax_16s$family == "env.OPS
17", paste0(coda_DW_ASVs_tax_16s$order, " ", coda_DW_ASVs_tax_16s$family),
coda_DW_ASVs_tax_16s$taxa)
coda_DW_ASVs_tax_16s$taxa <- ifelse(coda_DW_ASVs_tax_16s$order == "Blfdi19",

```

```
paste0(coda_DW_ASVs_tax_16s$class, " ", coda_DW_ASVs_tax_16s$order),
coda_DW_ASVs_tax_16s$taxa)
coda_DW_ASVs_tax_16s$taxa <- ifelse(coda_DW_ASVs_tax_16s$family == "BIrrii41",
paste0(coda_DW_ASVs_tax_16s$order, " ", coda_DW_ASVs_tax_16s$family),
coda_DW_ASVs_tax_16s$taxa)
```

Sort by coefficients

```
coda_DW_ASVs_tax_16s$ASV <- factor(coda_DW_ASVs_tax_16s$ASV,
levels=coda_DW_ASVs_tax_16s[order(coda_DW_ASVs_tax_16s$coeff), "ASV"])

coda_DW_ASVs_tax_ITS$ASV <- factor(coda_DW_ASVs_tax_ITS$ASV,
levels=coda_DW_ASVs_tax_ITS[order(coda_DW_ASVs_tax_ITS$coeff), "ASV"])
```

Taxon table 16s

```
P_table_16s_DW <- ggplot(coda_DW_ASVs_tax_16s)+
  geom_text(aes(x = ASV, y = 1, label = taxa), fontface = "italic" , hjust =
0, size = 5)+
  geom_text(aes(x = ASV, y = 2, label = rank), hjust = 0, size = 5)+
  ylim(1, 2.2)+
  mytheme+
  labs(title = "Taxon Rank", y = "")+
  theme(plot.title = element_text( hjust = 0.45, size = 18, color =
"grey40"),
axis.title.y = element_blank(), axis.title.x = element_text(size =
21, color = "white"),
axis.text.x = element_text(color = "white", size = 18.2), axis.text.y
= element_blank(),
axis.line.x = element_blank(), axis.line.y = element_blank(),
axis.ticks = element_blank()+
  coord_flip()
P_table_16s_DW
```

Taxon	Rank
<i>Thauera</i>	genus
<i>Microscillaceae</i> OLB12	genus
<i>Polyangia</i> Blfdi19	order
<i>Polyangiales</i> Blrii41	family
<i>Nevskia</i>	genus
<i>Gammaproteobacteria</i> CCD24	order
<i>Chryseolinea</i>	genus
<i>Cavicella</i>	genus
<i>Thermomonas</i>	genus
<i>Nitrospira</i>	genus
<i>Obscuribacteraceae</i>	family
<i>Sphingobacteriales</i> env.OPS 17	family
<i>Chitinophagales</i>	order

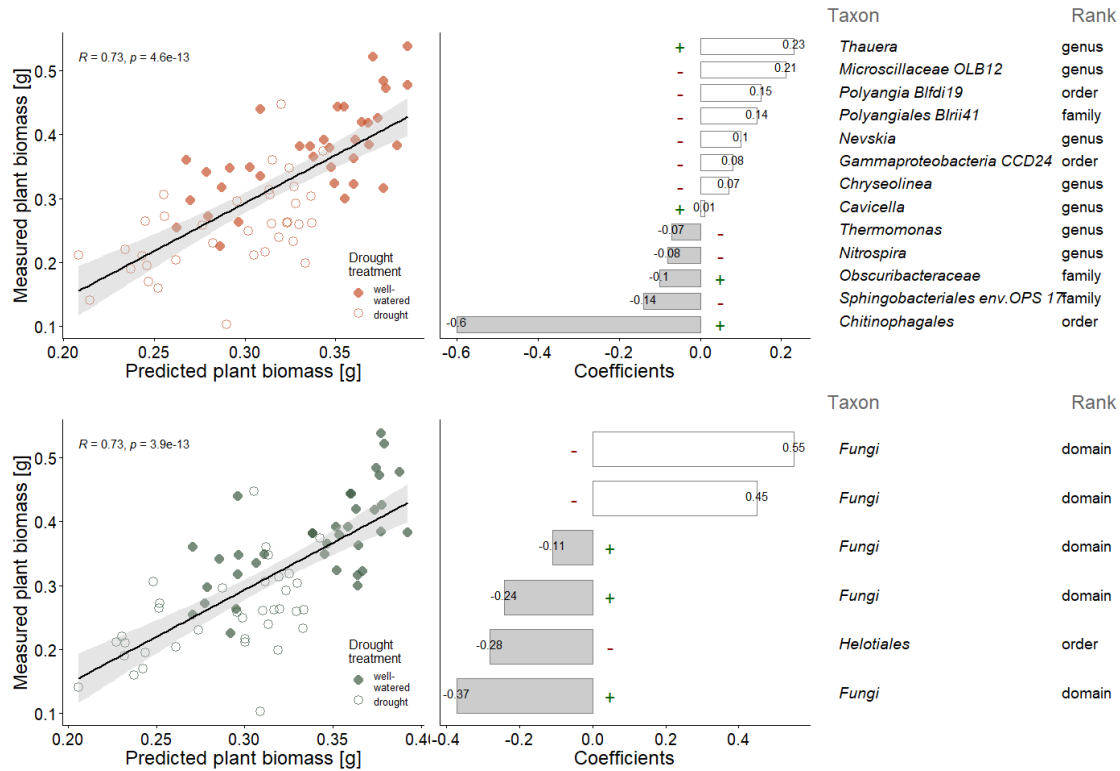
## Taxon table ITS

```
P_table_ITS_DW <- ggplot(coda_DW_ASVs_tax_ITS)+  
  geom_text(aes(x = ASV, y = 1, label = taxa), fontface = "italic" , hjust =  
0, size = 5)+  
  geom_text(aes(x = ASV, y = 2, label = rank), hjust = 0, size = 5)+  
  ylim(1, 2.2)+  
  mytheme+  
  labs(title = "Taxon Rank", y = "")+  
  theme(plot.title = element_text( hjust = 0.45, size = 18, color =  
"grey40"),  
        axis.title.y = element_blank(), axis.title.x = element_text(size =  
21, color = "white"),  
        axis.text.x = element_text(color = "white", size = 18.2), axis.text.y  
= element_blank(),  
        axis.line.x = element_blank(), axis.line.y = element_blank(),  
axis.ticks = element_blank()+  
  coord_flip()  
P_table_ITS_DW
```

Taxon	Rank
<i>Fungi</i>	domain
<i>Fungi</i>	domain
<i>Fungi</i>	domain
<i>Fungi</i>	domain
<i>Helotiales</i>	order
<i>Fungi</i>	domain

## Combined figure

```
P_sig_DW_combined <- ggarrange(P_pred_16s_DW_2, P_sig_16s_DW_2,  
P_table_16s_DW, P_pred_ITS_DW_2, P_sig_ITS_DW_2, P_table_ITS_DW, widths =  
c(1.1,1,0.8))  
P_sig_DW_combined
```



Save figure

```
ggsave(plot = P_sig_DW_combined, "Coda_DW_combined.tiff", width = 38, height = 28, units = "cm", dpi = 1000)
```

### For plant-soil feedback

Filter PSF data set fro plant-conditioned soil and PSF away

```
data_PSF_coda <- filter(data_PSF, input == "plant" & PSF_type == "away")
```

Add soil ID for home and away soils

```
data_PSF_coda$soil_ID_home <- paste("B", data_PSF_coda$block, "-",
data_PSF_coda$species_home, "-", data_PSF_coda$drought, "-",
data_PSF_coda$input, sep = "")
data_PSF_coda$soil_ID_away <- paste("B", data_PSF_coda$block, "-",
data_PSF_coda$species_away, "-", data_PSF_coda$drought, "-",
data_PSF_coda$input, sep = "")

data_PSF_coda <- data_PSF_coda %>%
  mutate(across('soil_ID_home', str_replace, 'WW', 'C')) %>%
mutate(across('soil_ID_home', str_replace, 'plant', 'soil')) %>%
  mutate(across('soil_ID_away', str_replace, 'WW', 'C')) %>%
mutate(across('soil_ID_away', str_replace, 'plant', 'soil'))
```

Select columns of the selected indicator taxa in the OTU table

```
otu_table_ASVS_ISA_16s <- as.data.frame(t(otu_table_16s_3perc)) %>%  
dplyr::select(any_of(ISA_ASVs_16s))
```

```
otu_table_ASVS_ISA_ITS <- as.data.frame(t(otu_table_ITS_3perc)) %>%  
dplyr::select(any_of(ISA_ASVs_ITS))
```

*Merge with PSF data*

Once for home soil and once for away soil

```
ISA_16s_PSF_coda_home <- merge(data_PSF_coda, otu_table_ASVS_ISA_16s, by.x =  
"soil_ID_home", by.y = "row.names") %>%  
  rename_with(~ paste0(.x, "_home"), starts_with("ASV"))
```

```
otu_table_ASVS_ISA_16s_away <- otu_table_ASVS_ISA_16s %>% rename_all(~  
paste0(.x, "_away"))
```

```
ISA_16s_PSF_coda <- merge(ISA_16s_PSF_coda_home, otu_table_ASVS_ISA_16s_away,  
by.x = "soil_ID_away", by.y = "row.names")
```

```
ISA_ITS_PSF_coda_home <- merge(data_PSF_coda, otu_table_ASVS_ISA_ITS, by.x =  
"soil_ID_home", by.y = "row.names") %>%  
  rename_with(~ paste0(.x, "_home"), starts_with("ASV"))
```

```
otu_table_ASVS_ISA_ITS_away <- otu_table_ASVS_ISA_ITS %>% rename_all(~  
paste0(.x, "_away"))
```

```
ISA_ITS_PSF_coda <- merge(ISA_ITS_PSF_coda_home, otu_table_ASVS_ISA_ITS_away,  
by.x = "soil_ID_away", by.y = "row.names")
```

*Determine coda inputs*

```
X_PSF_ISA_16s_soil <-  
as.matrix(data.frame(ISA_16s_PSF_coda[73:length(ISA_16s_PSF_coda)], row.names  
= paste(ISA_16s_PSF_coda$soil_ID_home, ISA_16s_PSF_coda$species_away, sep =  
"_")) %>%  
  impute_zeros())
```

```
Y_PSF_ISA_16s_soil <- structure(as.numeric(ISA_16s_PSF_coda$PSFtotal), names  
= paste(ISA_16s_PSF_coda$soil_ID_home, ISA_16s_PSF_coda$species_away, sep =  
"_"))
```

```
covariates_PSF_16s <- as.matrix(data.frame(ISA_16s_PSF_coda["block"],  
row.names = paste(ISA_16s_PSF_coda$soil_ID_home,  
ISA_16s_PSF_coda$species_away, sep = "_")))
```

```

X_PSF_ISA_ITS_soil <-
as.matrix(data.frame(ISA_ITS_PSF_coda[73:length(ISA_ITS_PSF_coda)], row.names
= paste(ISA_ITS_PSF_coda$soil_ID_home, ISA_ITS_PSF_coda$species_away, sep =
"_")) %>%
  impute_zeros())

Y_PSF_ISA_ITS_soil <- structure(as.numeric(ISA_ITS_PSF_coda$PSFtotal), names
= paste(ISA_ITS_PSF_coda$soil_ID_home, ISA_ITS_PSF_coda$species_away, sep =
"_"))

covariates_PSF_ITS <- as.matrix(data.frame(ISA_ITS_PSF_coda["block"],
row.names = paste(ISA_ITS_PSF_coda$soil_ID_home,
ISA_ITS_PSF_coda$species_away, sep = "_")))

```

### *Run Coda4microbiome*

Very computing intense, we ran it on the server. The next chunk contains the code to load the output.

```

coda_PSF_ISA_16s <- coda_glmnet(x=X_PSF_ISA_16s_soil,y=Y_PSF_ISA_16s_soil,
lambda = "lambda.1se", covar = covariates_PSF_16s)

coda_PSF_ISA_ITS <- coda_glmnet(x=X_PSF_ISA_ITS_soil,y=Y_PSF_ISA_ITS_soil,
lambda = "lambda.1se", covar = covariates_PSF_ITS)

```

### Load coda output

```

coda_PSF_ISA_16s <- read_rds("coda_PSF_ISA_16s.RDS")

coda_PSF_ISA_ITS <- read_rds("coda_PSF_ISA_ITS.RDS")

```

### Extract plots

```

P_sig_16s_PSF <- coda_PSF_ISA_16s$`signature plot`
P_pred_16s_PSF <- coda_PSF_ISA_16s$`predictions plot`

P_sig_ITS_PSF <- coda_PSF_ISA_ITS$`signature plot`
P_pred_ITS_PSF <- coda_PSF_ISA_ITS$`predictions plot`

```

### Extract plot data

```

coda_PSF_ASVs_tax_16s <- P_sig_16s_PSF$data %>%
  dplyr::rename("ASV" = vars)

coda_PSF_ASVs_tax_ITS <- P_sig_ITS_PSF$data %>%
  dplyr::rename("ASV" = vars)

```

## Separate ASV name from home and away

```
coda_PSF_ASVs_tax_16s$soil <- ifelse(grepl("home",
coda_PSF_ASVs_tax_16s$ASV), "home", "away")
coda_PSF_ASVs_tax_16s$ASV <- gsub("_home", "", coda_PSF_ASVs_tax_16s$ASV)
coda_PSF_ASVs_tax_16s$ASV <- gsub("_away", "", coda_PSF_ASVs_tax_16s$ASV)

coda_PSF_ASVs_tax_ITS$soil <- ifelse(grepl("home",
coda_PSF_ASVs_tax_ITS$ASV), "home", "away")
coda_PSF_ASVs_tax_ITS$ASV <- gsub("_home", "", coda_PSF_ASVs_tax_ITS$ASV)
coda_PSF_ASVs_tax_ITS$ASV <- gsub("_away", "", coda_PSF_ASVs_tax_ITS$ASV)
```

## Fuse with taxonomy

```
coda_PSF_ASVs_tax_16s <- merge(coda_PSF_ASVs_tax_16s,
indispecies_alpha0.1_16s_tax) %>%
  dplyr::select(-SH, -taxonomy)

coda_PSF_ASVs_tax_ITS <- merge(coda_PSF_ASVs_tax_ITS,
indispecies_alpha0.1_ITS_tax) %>%
  dplyr::select(-SH, -taxonomy)
```

## Extend taxa names

```
coda_PSF_ASVs_tax_16s$taxa <- ifelse(coda_PSF_ASVs_tax_16s$family == "WD2101
soil group", paste0(coda_PSF_ASVs_tax_16s$order, " ",
coda_PSF_ASVs_tax_16s$family), coda_PSF_ASVs_tax_16s$taxa)
coda_PSF_ASVs_tax_16s$taxa <- ifelse(coda_PSF_ASVs_tax_16s$family ==
"BIRii41", paste0(coda_PSF_ASVs_tax_16s$order, " ",
coda_PSF_ASVs_tax_16s$family), coda_PSF_ASVs_tax_16s$taxa)
coda_PSF_ASVs_tax_16s$taxa <- ifelse(coda_PSF_ASVs_tax_16s$order == "0319-
6G20", paste0(coda_PSF_ASVs_tax_16s$class, " ", coda_PSF_ASVs_tax_16s$order),
coda_PSF_ASVs_tax_16s$taxa)
```

## Sort by coefficients

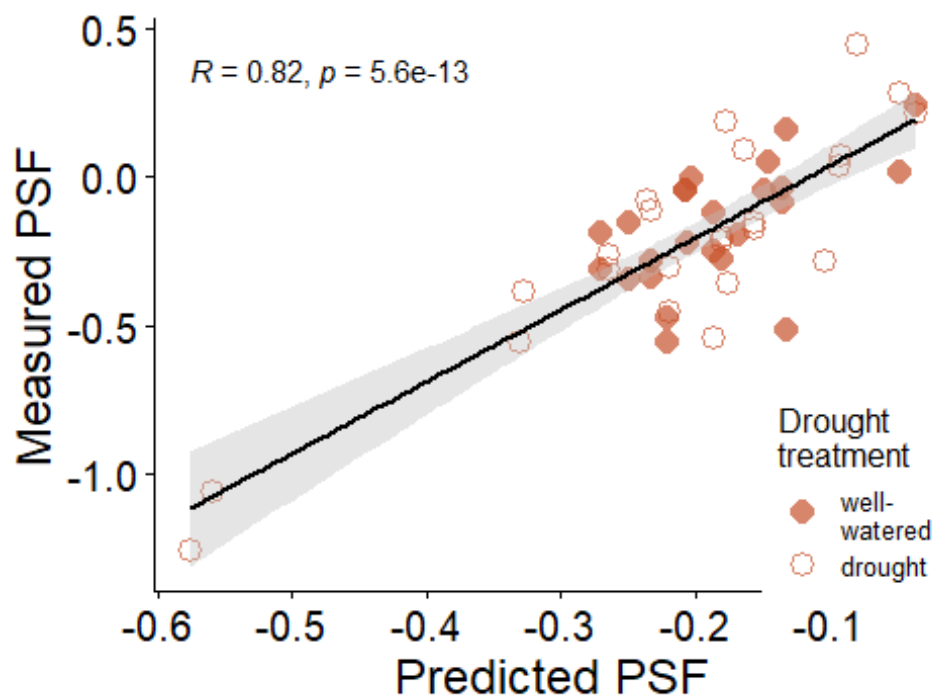
```
coda_PSF_ASVs_tax_16s$ASV <- factor(coda_PSF_ASVs_tax_16s$ASV,
levels=coda_PSF_ASVs_tax_16s[order(coda_PSF_ASVs_tax_16s$coeff), "ASV"])

coda_PSF_ASVs_tax_ITS$ASV <- factor(coda_PSF_ASVs_tax_ITS$ASV,
levels=coda_PSF_ASVs_tax_ITS[order(coda_PSF_ASVs_tax_ITS$coeff), "ASV"])
```

## Prediction plots

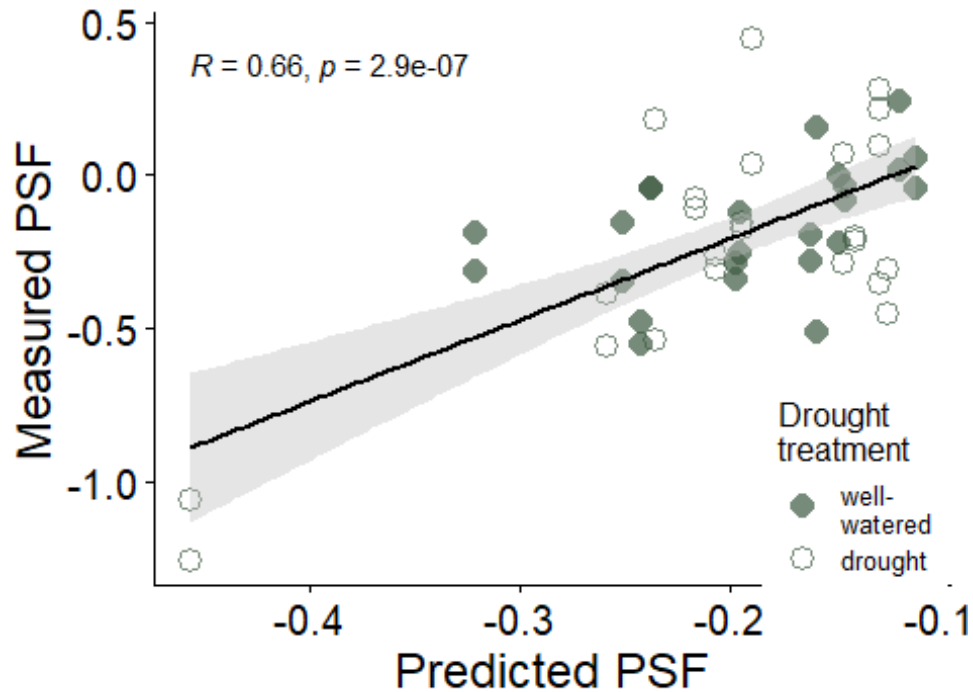
### Prediction plot 16s

```
P_pred_16s_PSF_2 <- P_pred_16s_PSF + geom_point(col = "white", size = 3)+  
  geom_smooth(method = "lm", col = "black", fill = "grey", alpha = 0.4)+  
  geom_point(aes(shape = ISA_16s_PSF_coda$drought), col = "#C7522B", size =  
4, alpha = 0.7)+  
  scale_shape_manual(values = c(16, 1), name = "Drought\ntreatment", labels  
= c("well-\nwatered", "drought"))+  
  geom_smooth(method = "lm", col = "black", fill = "grey", alpha = 0.4)+  
  labs(x = "Predicted PSF", y = "Measured PSF", title = "")+  
  theme(plot.title = element_text(hjust = 0.5, size = 18), axis.title =  
element_text(size = 18), legend.position = "inside",  
  legend.position.inside = c(0.88, 0.17), axis.text = element_text(size  
= 15))  
P_pred_16s_PSF_2
```



## Prediction plot ITS

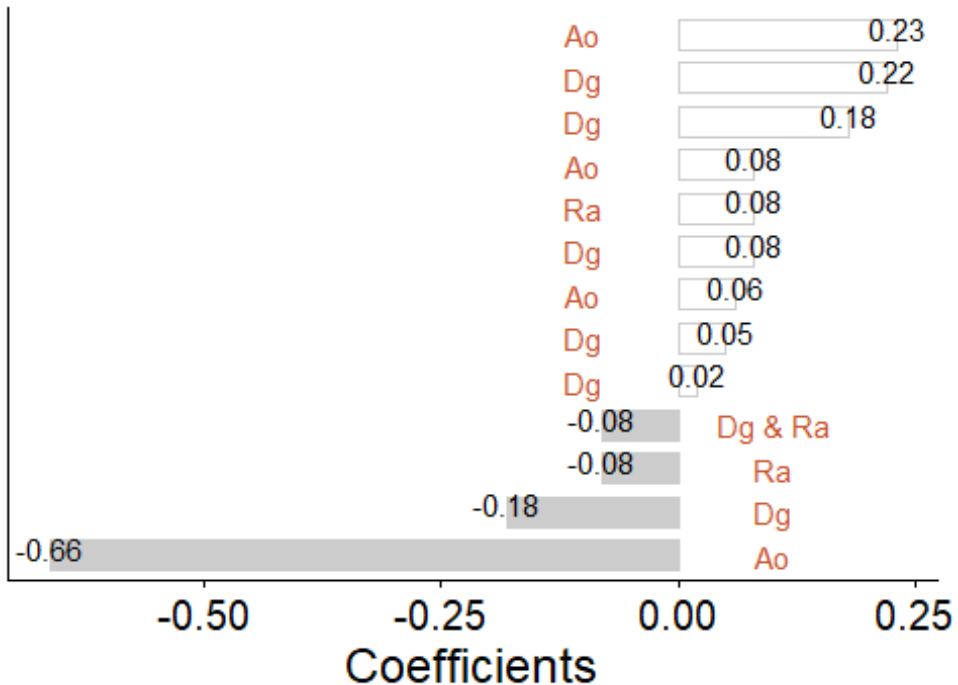
```
P_pred_ITS_PSF_2 <- P_pred_ITS_PSF + geom_point(col = "white", size = 3)+  
  geom_smooth(method = "lm", col = "black", fill = "grey", alpha = 0.4)+  
  geom_point(aes(shape = ISA_ITS_PSF_coda$drought), col = "#3C5941", size =  
4, alpha = 0.7)+  
  scale_shape_manual(values = c(16, 1), name = "Drought\treatment", labels  
= c("well-\nwatered", "drought"))+  
  geom_smooth(method = "lm", col = "black", fill = "grey", alpha = 0.4)+  
  labs(x = "Predicted PSF", y = "Measured PSF", title = "")+  
  theme(plot.title = element_text(hjust = 0.5, size = 18), axis.title =  
element_text(size = 18), legend.position = "inside",  
  legend.position.inside = c(0.88, 0.17), axis.text = element_text(size  
= 15))  
P_pred_ITS_PSF_2
```



## Signature plots

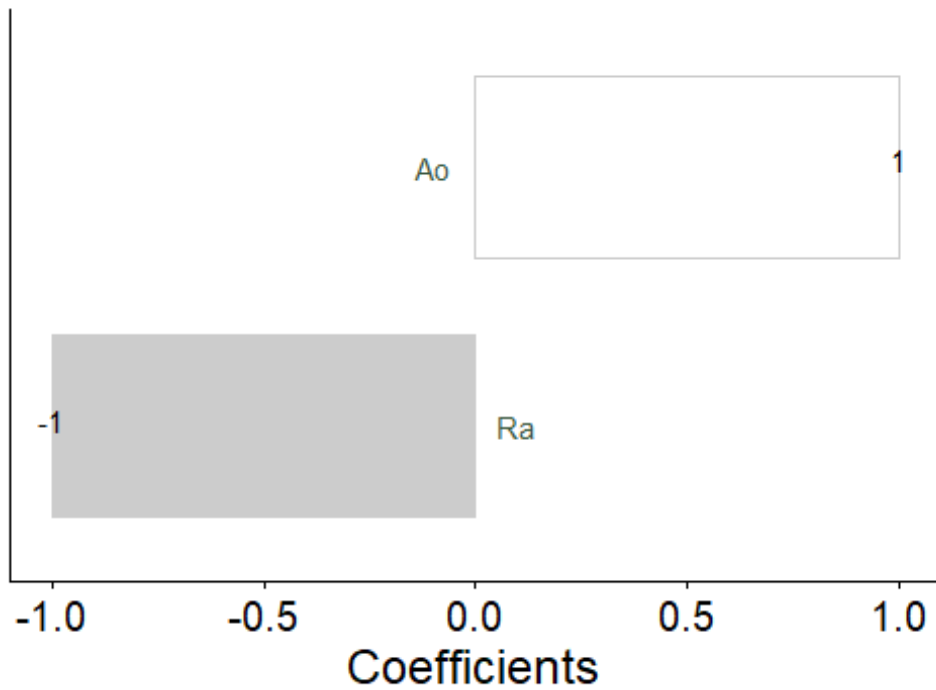
### Signature plot 16s

```
P_sig_16s_PSF_2 <- P_sig_16s_PSF +  
  scale_fill_manual(values = c("grey80", "white"))+  
  scale_color_manual(values = c("grey80", "grey80"))+  
  geom_text(aes(x = vars, y = ifelse(coeff < 0, 0.1, -0.1), label =  
coda_PSF_ASVs_tax_16s$association), col = "#C7522B")+  
  labs(y = "Coefficients", title = "")+  
  theme(plot.title = element_text(hjust = 0.5, size = 18),  
        axis.title = element_text(size = 18), axis.ticks.y = element_blank(),  
        axis.text.y = element_blank(),  
        legend.position = "none", axis.text.x = element_text(size = 15))  
P_sig_16s_PSF_2
```



## Signature plot ITS

```
P_sig_ITS_PSF_2 <- P_sig_ITS_PSF +  
  scale_fill_manual(values = c("grey80", "white"))+  
  scale_color_manual(values = c("grey80", "grey80"))+  
  geom_text(aes(x = vars, y = ifelse(coeff < 0, 0.1, -0.1), label =  
coda_PSF_ASVs_tax_ITS$association), col = "#3C5941")+  
  labs(y = "Coefficients", title = "")+  
  theme(plot.title = element_text(hjust = 0.5, size = 18),  
        axis.title = element_text(size = 18), axis.ticks.y = element_blank(),  
        axis.text.y = element_blank(),  
        legend.position = "none", axis.text.x = element_text(size = 15))  
P_sig_ITS_PSF_2
```



## Taxon tables

### Taxon table 16s

```
P_table_16s_PSF <- ggplot(coda_PSF_ASVs_tax_16s)+
  geom_text(aes(x = ASV, y = 1, label = taxa), fontface = "italic" , hjust =
0, size = 5)+
  geom_text(aes(x = ASV, y = 1.95, label = rank), hjust = 0, size = 5)+
  geom_text(aes(x = ASV, y = 2.2, label = soil), hjust = 0, size = 5)+
  ylim(1, 2.3)+
  mytheme+
  labs(title = "Taxon                               Rank      Soil", y
= "")+
  theme(plot.title = element_text( hjust = 0.45, size = 18, color =
"grey40"),
        axis.title.y = element_blank(), axis.title.x = element_text(size =
21, color = "white"),
        axis.text.x = element_text(color = "white", size = 18.2), axis.text.y
= element_blank(),
        axis.line.x = element_blank(), axis.line.y = element_blank(),
axis.ticks = element_blank()+
  coord_flip()

P_table_16s_PSF
```

Taxon	Rank	Soil
<i>Bdellovibrio</i>	genus	home
<i>Vicinamibacteraceae</i>	family	home
<i>Polyangiales Blrii41</i>	family	home
<i>lamia</i>	genus	home
<i>Oligoflexia 0319-6G20</i>	order	home
<i>Solirubrobacter</i>	genus	home
<i>Arboricoccus</i>	genus	away
<i>Pedosphaeraceae</i>	family	home
<i>Haliangium</i>	genus	away
<i>Nocardia</i>	genus	home
<i>Tepidisphaerales WD2101 soil group</i>	family	home
<i>Vicinamibacterales</i>	order	home
<i>Myxococcaceae</i>	family	home

## Taxon table ITS

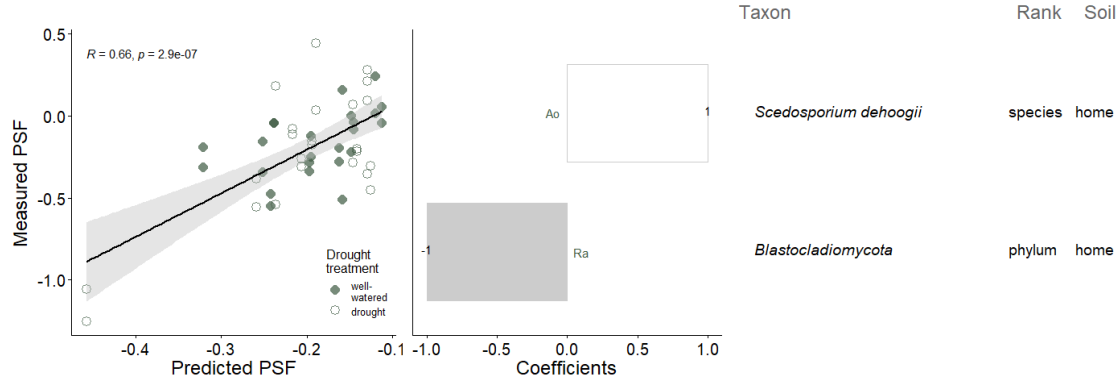
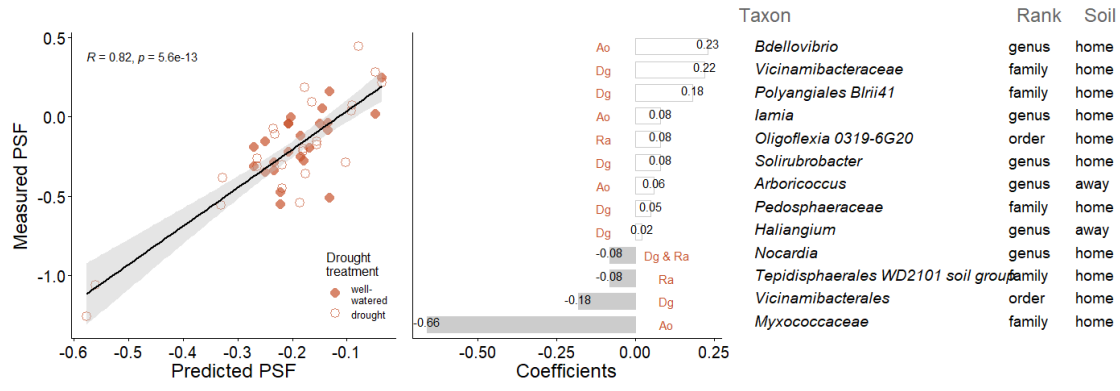
```
P_table_ITS_PSF <- ggplot(coda_PSF_ASVs_tax_ITS)+
  geom_text(aes(x = ASV, y = 1, label = taxa), fontface = "italic" , hjust =
0, size = 5)+
  geom_text(aes(x = ASV, y = 1.95, label = rank), hjust = 0, size = 5)+
  geom_text(aes(x = ASV, y = 2.2, label = soil), hjust = 0, size = 5)+
  ylim(1, 2.3)+
  mytheme+
  labs(title = "Taxon                               Rank      Soil", y
= "")+
  theme(plot.title = element_text( hjust = 0.45, size = 18, color =
"grey40"),
        axis.title.y = element_blank(), axis.title.x = element_text(size =
21, color = "white"),
        axis.text.x = element_text(color = "white", size = 18.2), axis.text.y
= element_blank(),
        axis.line.x = element_blank(), axis.line.y = element_blank(),
axis.ticks = element_blank()+
  coord_flip()

P_table_ITS_PSF
```

Taxon	Rank	Soil
<i>Scedosporium dehoogii</i>	species	home
<i>Blastocladiomycota</i>	phylum	home

## Combined figure

```
P_sig_PSF_combined <- ggarrange(P_pred_16s_PSF_2, P_sig_16s_PSF_2,
P_table_16s_PSF, P_pred_ITS_PSF_2, P_sig_ITS_PSF_2, P_table_ITS_PSF, widths
= c(1.1,0.9,1.1) )
P_sig_PSF_combined
```



Save figure

```

ggsave(plot = P_sig_PSF_combined, "Coda_PSF_combined.tiff", width = 38,
height = 28, units = "cm", dpi = 1000)

```