

Getting Functionnal-structural plant model into a pipeline: MARSHAL-pipeline

Adrien Heymans, Felicien Meunier et Guillaume Lobet

04 décembre, 2018

1 Introduction

The coupling of models with one another can be used to find relevant sequence of information. Depending of the models, the complexity of the system is significantly increase. Here, the system on which this pipeline works is a below ground system. It focuses on solving the hydraulic architecture of a root system. To illustrate the necessity of merging models together, we start by the fact that MARSHAL need an explicit root system architecture to add on the hydraulic part. **CRootBox** is precisely a model that computes the explicit root system architecture and **MARSHAL** was designed to accept easily the CRootBox output. Dealing with models as an input for another model will also increase the dynamic approach of the overall.

1.1 How to use MARSHAL in a R environment

To run this example, the marshal package is required. This package can be found on GitHub:

- <https://github.com/MARSHAL-ROOT/marshal>

or by simply execute the following line:

```
install.packages("devtools")
library(devtools)
install_github("MARSHAL-ROOT/marshal")
library(marshal)
```

MARSHAL is built upon a handful of other packages that need to be loaded. One group of function has to be loaded manually and is stored :

- <https://github.com/MARSHAL-ROOT/marshal-pipeline/R/>

```
library(data.table)
library(tidyverse)
library(Matrix)
library(readxl)
library(cowplot)
library(xml2)
library(knitr)
#library(marshal) # ! not working
`%!in%` <- compose(`!`, `%in%`)

source("R/io_function.R")
# Working version of MARSHAL
source("R/getSUF.R")
```

Once it is installed, the boundary condition on which the model works should be implemented.

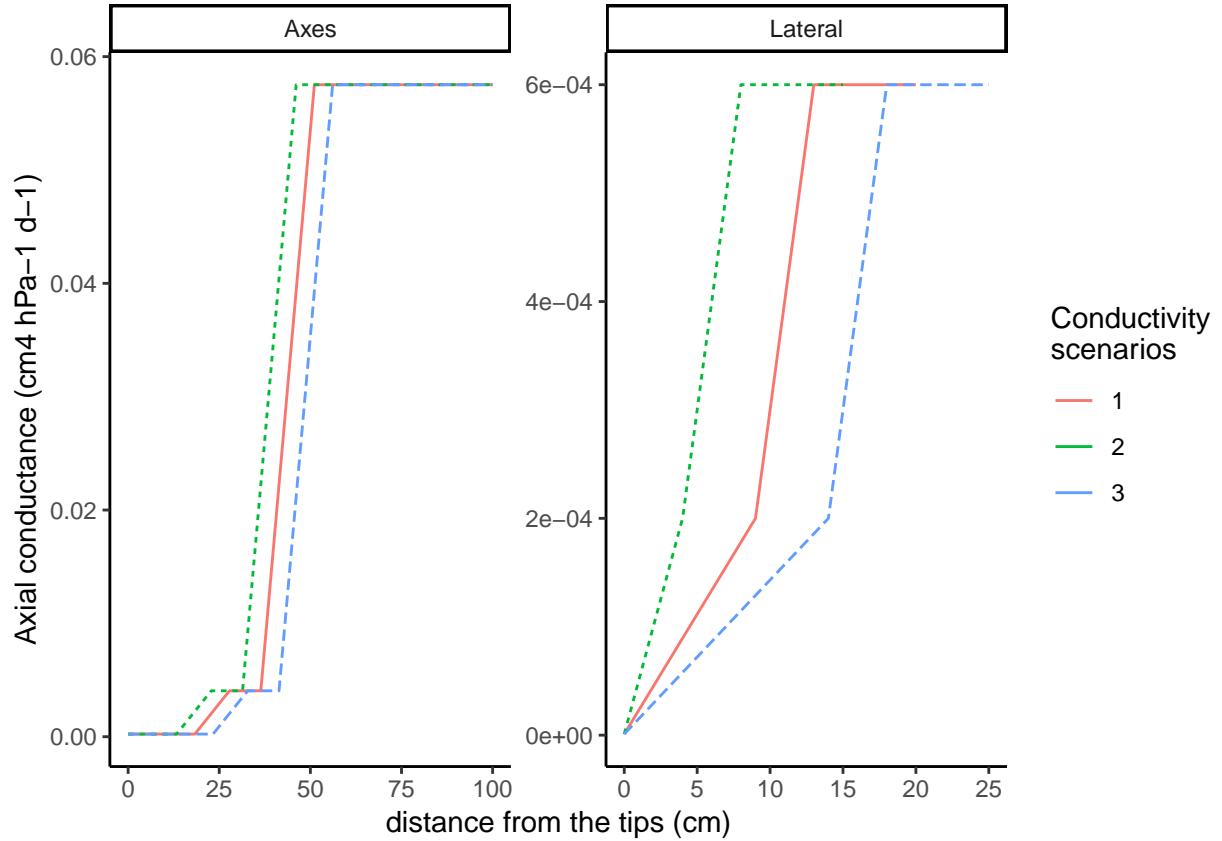


Figure 1: Axial conductance on axes and lateral root type

2 Boundary condition of MARSHAL.

The boundary condition to MARSHAL are :

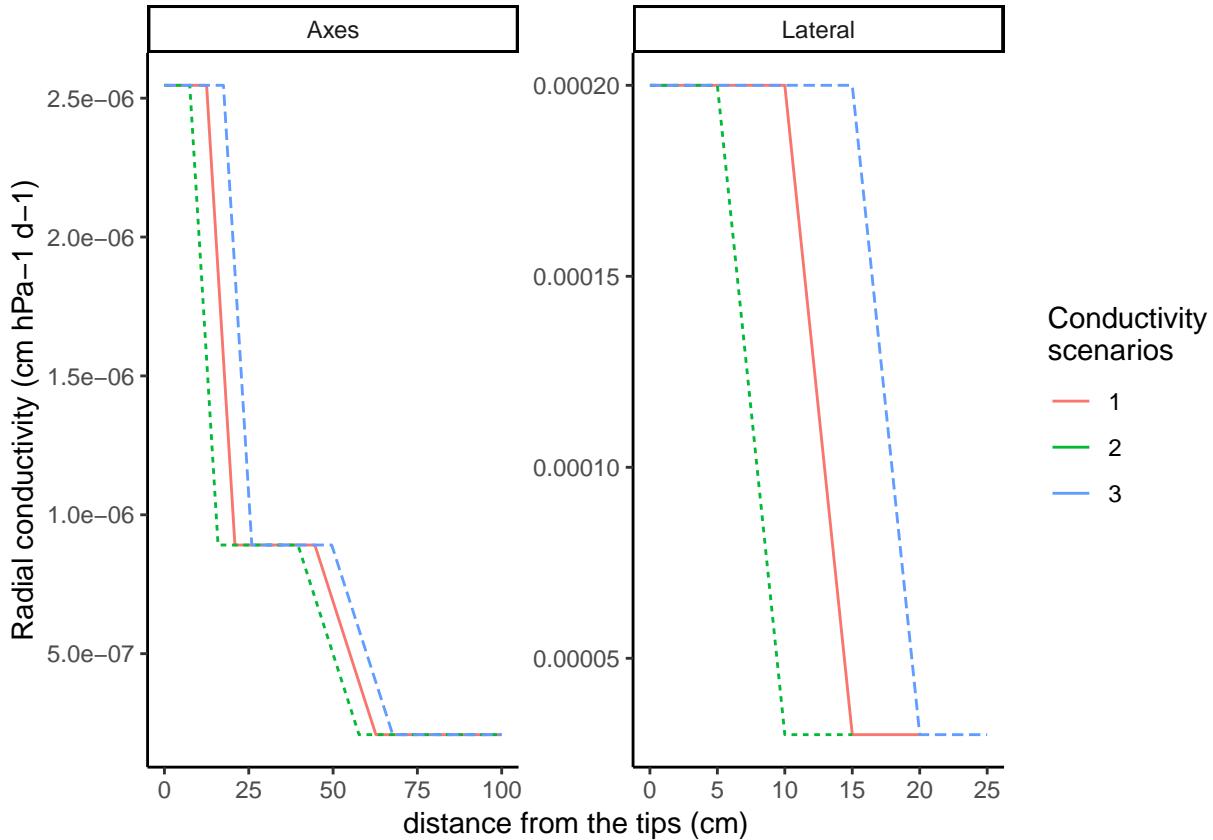
- The hydraulic properties of the roots.
- The water potential of the soil.
- The root system architecture.
- The initial pressure strength that pull water out of the collar.

2.1 Hydraulic properties of the root system

The hydraulic properties of the root system can be found in the litterature, by measuring it on root samples or with the helps of modelling tools. Here, the conductivities of the root come from the Doussan et al. 1998 paper. To show some possibilities, two scenarios were built from this Doussan et al. table of conductivities. The first, which is the original values placed along the root axes from the apex. The second, is based on the first but the maturity of the root anatomy is advanced of 5 cm towards the apex. And the third, is the opposite of the second, thus the maturity of the root anatomy, insted of being advance towards the apex, is moved backwards.

2.1.1 The axial conductance

2.1.2 The radial conductivity



2.2 Root system architecture with CRootBox

Modelling root system architecture can be achieved with CRootBox for instance. Some examples of the parameter files for a couple of species are details in the *modelparameter* folder, which is linked to the CRootBox depository.

- <https://github.com/Plant-Root-Soil-Interactions-Modelling/CRootBox>

For the purpose of the example we choose to work with three parameter file of Maize:

- “Zea_mays_3_Postma_2011”
- “Zea_mays_1_Leitner_2010”
- “Zea_mays_4_Leitner_2014”

```
# Number of repetition per root system architecture example
nrep <- 5

# List of three CrootBox parameter for corn
param_choise <- c("Zea_mays_1_Leitner_2010",
                  "Zea_mays_3_Postma_2011",
                  "Zea_mays_4_Leitner_2014")

all_roots <- NULL
for(param_name in param_choise){
  # Load param of the selected example
```

```

rparam <- read_rparam(path = paste0("./www/", param_name, ".rparam"))
pparam <- read_pparam(path = paste0("./www/", param_name, ".pparam"))
# Overwrite the param files that will be use
write_rparam(rparam, "./www/param.rparam")
write_pparam(pparam, "./www/param.pparam")

for(i in c(1:nrep)){
  # Run CRootBox ----
  system("www/a.exe")

  # Get all the files exported by CRootBox
  fls <- list.files("./")
  fls <- fls[grep1("rootsystem.txt", fls)]

  for(f in fls){
    temp <- fread(f, header = T)
    temp$age <- strsplit(f, "_")[[1]][1]
    temp$rep <- i
    temp$ex <- param_name
    all_roots <- rbind(all_roots, temp)
  }
}
}

# Saving the root system information
# write_csv(all_roots, "all_roots.csv")
# all_roots <- read.csv("all_roots.csv")

# Take the same time variable as the one that were implemented in CRootBox.
full_time <- sort(unique(as.numeric(all_roots$age)))
time_step = full_time[2]-full_time[1]
endTime = max(full_time)
ntimesteps = endTime/time_step

```

2.2.1 The root system architecture

The following figure shows the root system architectures that come out CRootBox at the half of the simulation time.

2.2.2 Architectural phenes

Once the complete root system architecture is computed there are a number of architectural phenes that we can access.

2.2.2.1 The convex hull

The convex hull is one of the most common known traits to work with. It gives a qualitative information about root exploration. To find the convex hull with an explicit root system in 3D is straight forwards. Thanks to the chull function, we are able to identify all the coordinates that make the convex hull and the Polygon function in the sp package allows us to determine the area of the convex hull.

```

root_20 <- all_roots%>%
  filter(age %in% c("20", "40", "60"))
pol_ch = poly_area<- NULL
for(i in unique(all_roots$ex)){

```

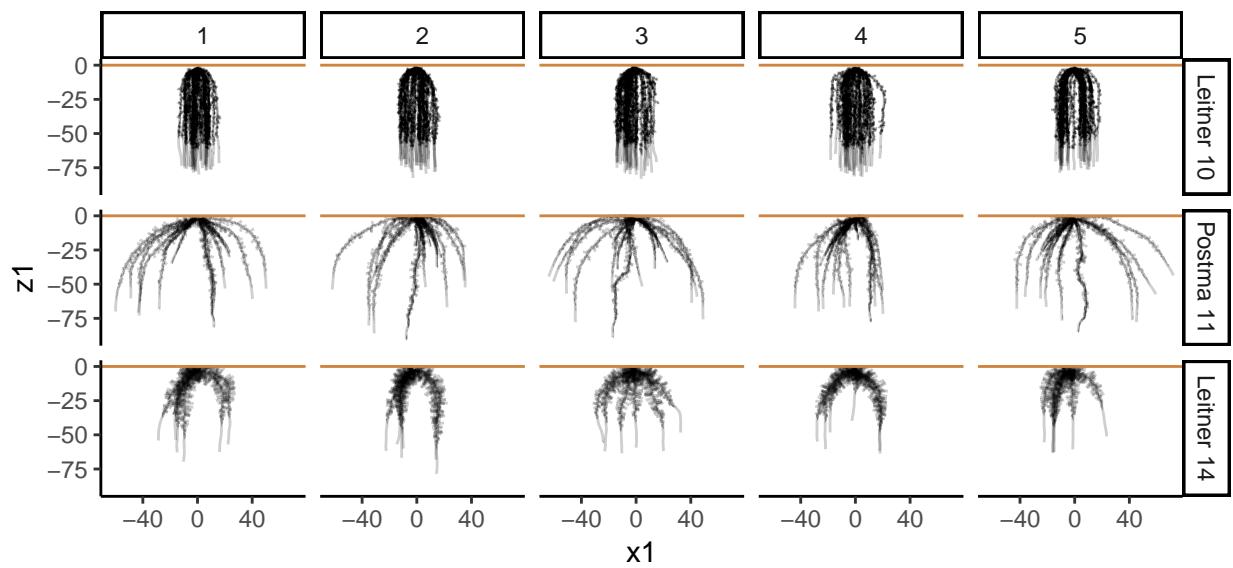


Figure 2: Projection in two dimensions of the different root systems computed with CRootBox at the age of 30 days

```

for(j in unique(all_roots$rep)){
  for(ag in unique(root_20$age)){
    for(xy in c("x", "y")){# get the convex hull from two sides
      tmp <- root_20%>%
        filter(ex == i, rep == j, age == ag)%>%
        select(paste0(xy,"2"),z2, ex, rep, age)
        # select(x2,z2, ex, rep, age)
      first_col <- tmp[,1]
      x <- matrix(c(first_col, tmp$z2), nc = 2)
      ch <- chull(x= first_col, y = tmp$z2)
      ch <- c(ch, ch[1])

      # Area of the convex hull
      xy.coords <- cbind(first_col, tmp$z2)
      chull.coords <- xy.coords[ch,]
      chull.poly <- sp::Polygon(chull.coords, hole = F)
      chull.area <- chull.poly@area

      polyg <- cbind(as.data.frame(x[ch,]), ex = i, rep = paste0(j,xy), age = ag)
      pol_area <- c(chull.area/10000)%>% # Transformation into square meter
          as.tibble()%>%
          transmute(area = value, ex = i, rep= paste0(j,xy), age = as.numeric(ag))
      poly_area <- rbind(poly_area, pol_area)
      pol_ch <- rbind(pol_ch, polyg)
    }
  }
}
}

```

2.2.2.2 The convex hull area.

2.2.2.3 The maximal rooting depth

```

# Summarise the total root length and the rooting depth information.
RLD <- all_roots%>%
  mutate(age = as.numeric(age))%>%
  group_by(age, rep, ex)%>%
  summarise(root = sum(length),
            rooting_depth = min(z2))

```

2.2.2.4 The total root length

2.2.2.5 Number of roots.

Although the total root length reach a close value of ten meter for all the different rootsystems, there is a huge gap in term of number of roots between each of the root system architecture.

2.3 The water soil potential

The pressure head of the water potential within the soil is created from the followig line of code. Here, there are three soil scenarios with three different coeficients in the drying function.

The drying function is :

$$\psi_i = \alpha \times \frac{(Z_i+L)^{10} \times T_i}{L^{10} \times T_{end}} - 230$$

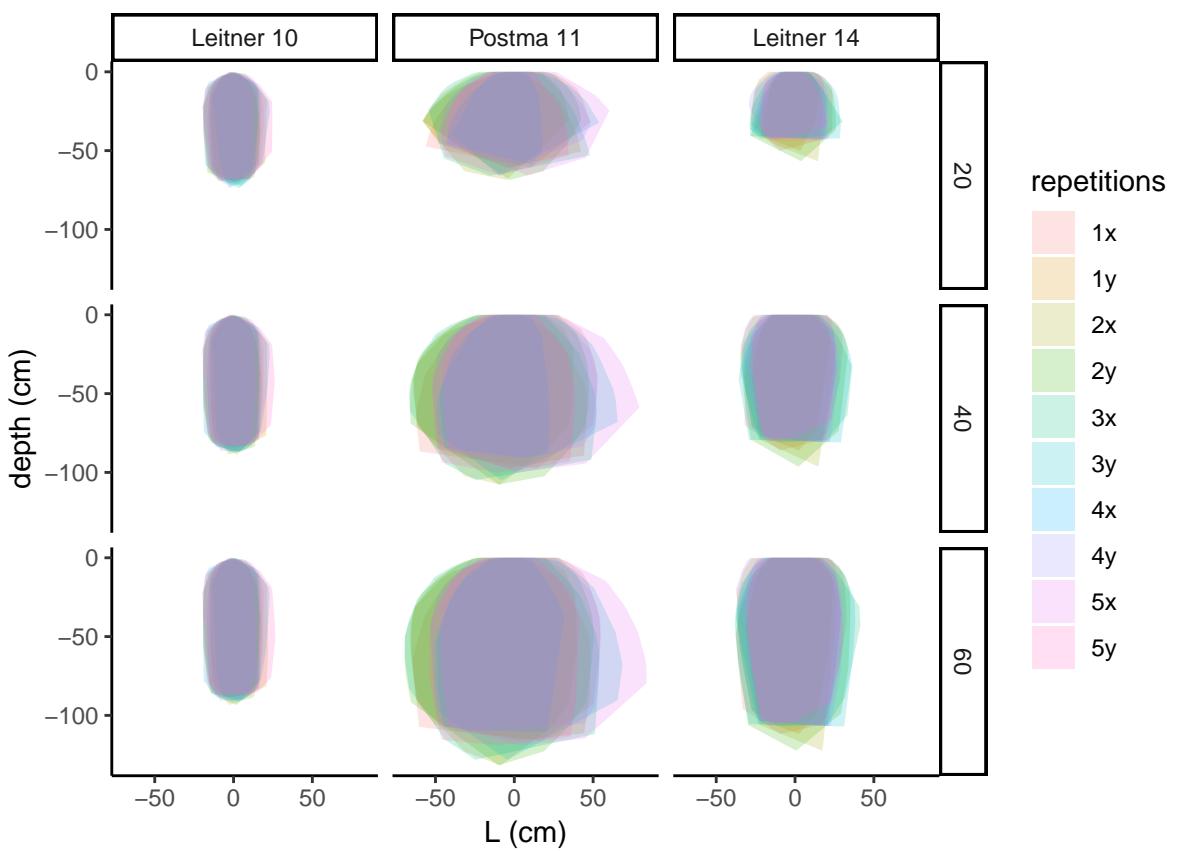


Figure 3: Convex hull of the root system examples x and y axes together

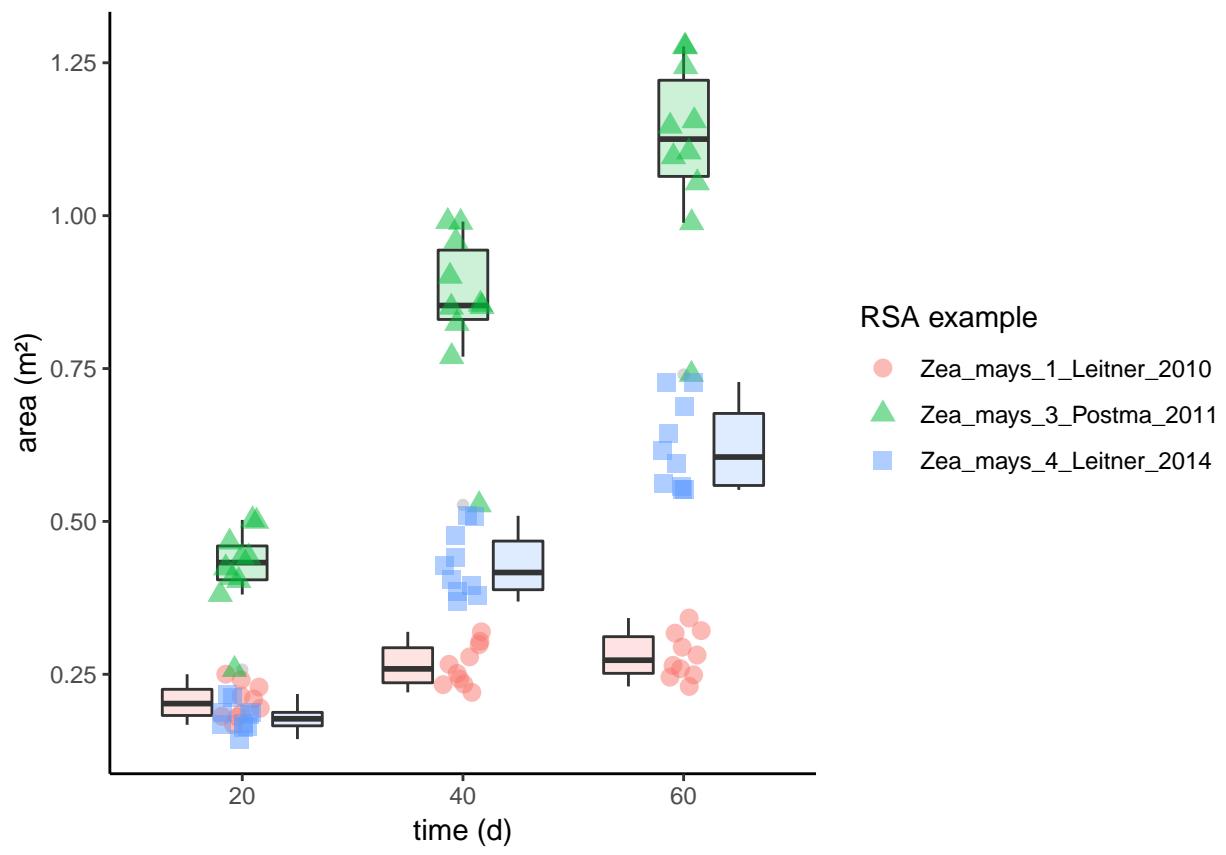


Figure 4: Area of the convex hull of the different root system examples

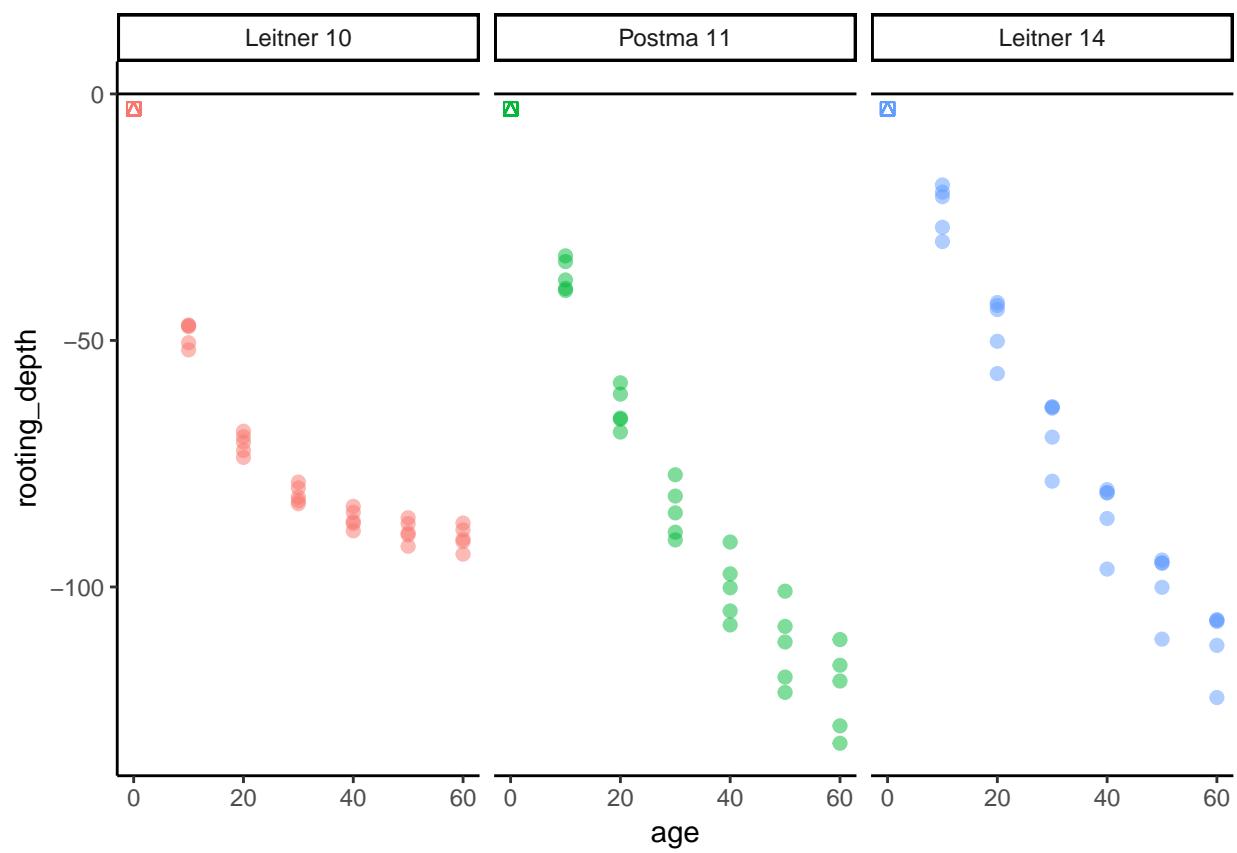


Figure 5: Rooting depth of the different root system examples

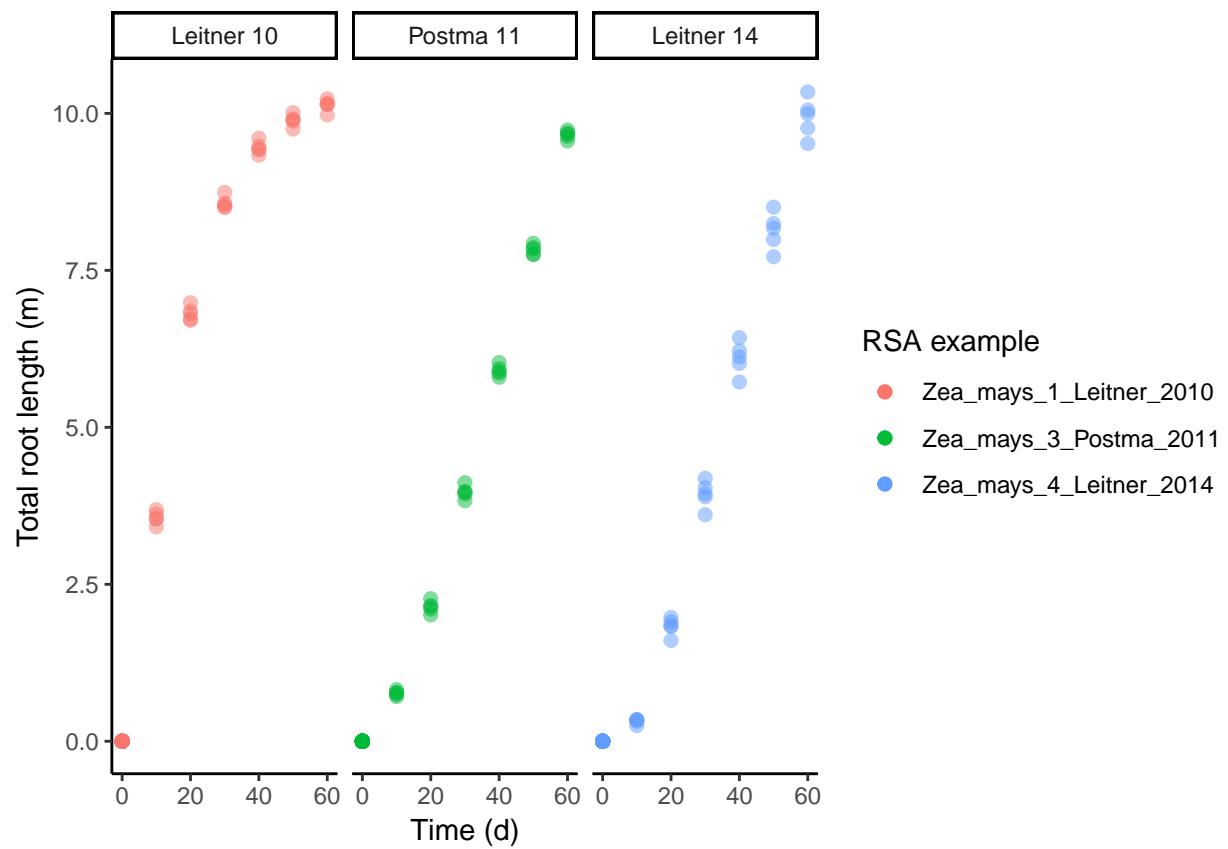


Figure 6: Total root length for each of the root system examples

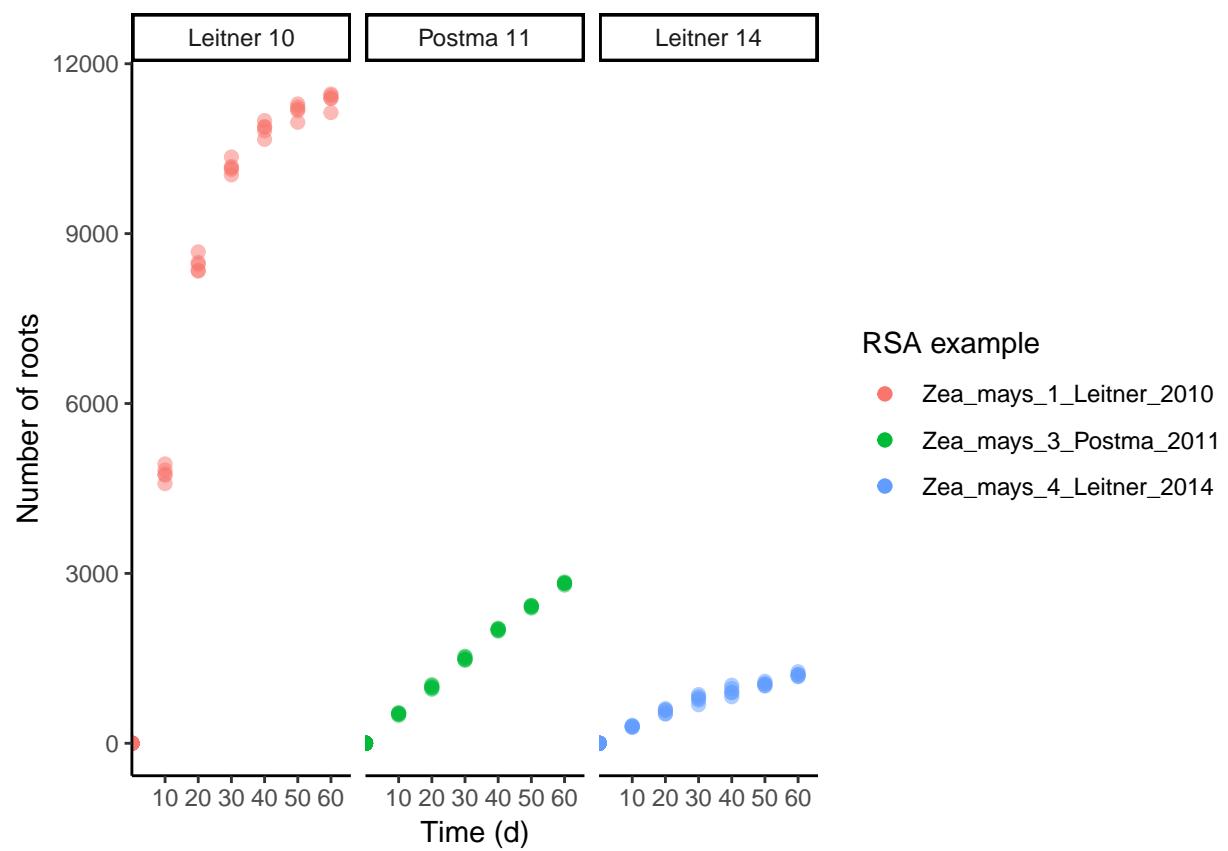


Figure 7: Number of different roots for each of the root system examples

- ψ_i = water potential in the soil at i cm below the surface.
- α = drying coefficient of the soil (-300 = “humid”, -7000 = “dry” and -15.000 = “arid”).
- Z_i = the depth (cm)
- L = the maximal soil profile depth.
- T = the simulation time (day)
- -230 is the water potential in the soil at the field cappacity for a silty soil.

```
# Creating some values for the water potential in the soil along the profile depth.
profile_depth <- 150
z <- seq(-profile_depth, 0, by = 5)
value <- sort(rep(full_time, length(z)))
psi <- rep(-230, length(value))
z <- rep(z, ntimesteps)

soil <- NULL
# Making three scenarios
for(i in c(-300, -7000, -15000)){
  soil_tmp <- cbind(z, value, psi) %>%
    as.tibble() %>%
    mutate(psi = i*((z+profile_depth)^(10))*value) /
      ((profile_depth^10)*endTime)-230,
    humid = i)
  soil_tmp <- soil_tmp %>%
    mutate(id = 1:nrow(soil_tmp))
  soil <- rbind(soil, soil_tmp)
}
```

2.4 The water potential at the collar

Expressed in hecto-Pascal, the pressure head define the transpiration strenght that drags the water out of the root system. -15.000 hPa

```
# Water potential at the collar (hPa)
tpots <- -15000
```

3 MARSHAL

MARSHAL provide an estimation of the conductance of the whole root system, the actual and the potential transpiration. The input needed in order to provide those relevant informations are :

```
new_all_roots <- NULL
results<- NULL
k <- 1
# -----
# Computing the root system hydraulic architecture
# -----
# Total of simulation that will be initiate
all <- length(unique(all_roots$rep)) * length(unique(conductivities$scenario)) *
length(unique(all_roots$age)) * length(unique(all_roots$ex)) *
length(unique(soil$humid))

pb = txtProgressBar(min = 0, max = all, initial = 0, style = 3)

##
```

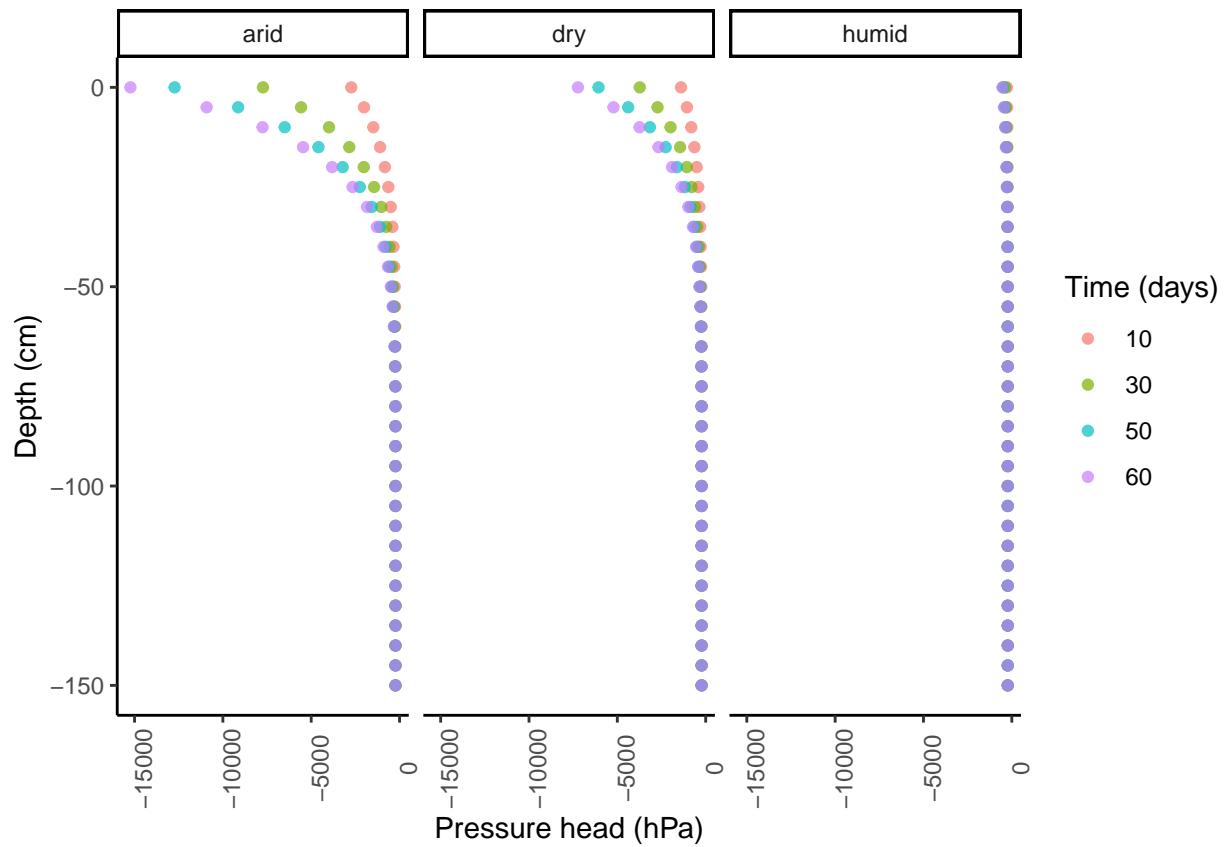


Figure 8: Water potential evolution during the simulation time along the soil profile

```

| | 0%
| |
# Loop to run MARSHAL with the selected simulation parameters.

for(exa in unique(all_roots$ex)){# Example of RSA
  for(re in unique(all_roots$rep)){ # repetition
    for(ag in sort(as.numeric(unique(all_roots$age)))){# age of the RS
      for(sen in unique(conductivities$scenario)){ # hydraulic properties
        for(hum in unique(soil$humid)){ # Scenario of water potential in the soil

          # Progression bar
          # setTxtProgressBar(pb,k)

          # Select specific Soil for the simulation
          temp_soil <- soil %>%
            filter(value == ag,
                  humid == hum)%>%
            select(-value, -humid)
          # Select specific root system for the simulation
          temp_root <- all_roots %>%
            filter(age == ag,
                  rep == re,
                  ex == exa)%>%
            as.tibble()%>%
            select(-ex)
          # Select specific conductivities
          temp_conduct <- conductivities%>%
            filter(scenario == sen)%>%
            select(-scenario)

          # -----
          # Run MARSHAL
          # -----
          hydraulics <- getSUF(temp_root,
                                temp_conduct,
                                temp_soil,
                                hetero = T,
                                Psi_collar = tpots)

          # Aggregate output from MARSHAL
          results <- rbind(results, data.frame(krs = hydraulics$krs,
                                                tact = hydraulics$tact,
                                                tpot = hydraulics$tpot,
                                                tp = tpots,
                                                scenario = sen,
                                                age = ag,
                                                rep = re,
                                                ex = exa,
                                                humid = hum,
                                                simulation = k))

          # Keep the information about the very detail hydrological parameter on
          # all the root segment only for one repetition of the plant per simulation.
        }
      }
    }
  }
}

```

```

if(re >= nrep-1){
  # Format dataset to be compatible with MARSHAL output
  first <- temp_root[temp_root$node1ID == 0,]
  nodals_ids <- unique(temp_root$branchID[temp_root$type == 4 |
                                             temp_root$type == 5])
  for(no in nodals_ids){
    temp <- temp_root%>%
      filter(branchID == no)
    temp <- temp[1,]
    connection <- data.frame(node1ID = 0,
                               node2ID = temp$node1ID,
                               branchID = temp$branchID,
                               x1 = first$x1, y1 = first$y1, z1 = first$z1,
                               x2 = temp$x1, y2 = temp$y1, z2 = temp$z1,
                               radius = temp$radius,
                               length = sqrt((first$x1-temp$x1)^2 +
                                              (first$y1-temp$y1)^2 +
                                              (first$z1-temp$z1)^2 ),
                               R = 0, G = 0, B = 0,
                               time = temp$time,
                               type = temp$type,
                               age = temp$age,
                               rep = temp$rep)
    new_table = rbind(temp_root, connection)
    temp_root = new_table
  }
  temp_root <- temp_root[order(temp_root$node2ID, decreasing = F),]

  # Merge output of MARSHAL on specific root segment
  temp_root$suf <- as.vector(hydraulics$suf)
  temp_root$suf1 <- as.vector(hydraulics$suf1)
  temp_root$kx <- as.vector(hydraulics$kx)
  temp_root$kr <- as.vector(hydraulics$kr)
  temp_root$jr <- as.vector(hydraulics$jr)
  temp_root$psi <- as.vector(hydraulics$psi)
  temp_root$jxl <- as.vector(hydraulics$jxl)
  temp_root$psi_soil <- as.vector(hydraulics$psi_soil)
  # Add the simulation specificity to the dataset
  temp_root$scenario <- sen
  temp_root$ex <- exa
  temp_root$humid <- hum
  temp_root$rep <- re
  new_all_roots <- rbind(new_all_roots, temp_root)
}
k <- k + 1
}
}
}
}

# Save the results
# write_csv(new_all_roots, "new_all_roots.csv")

```

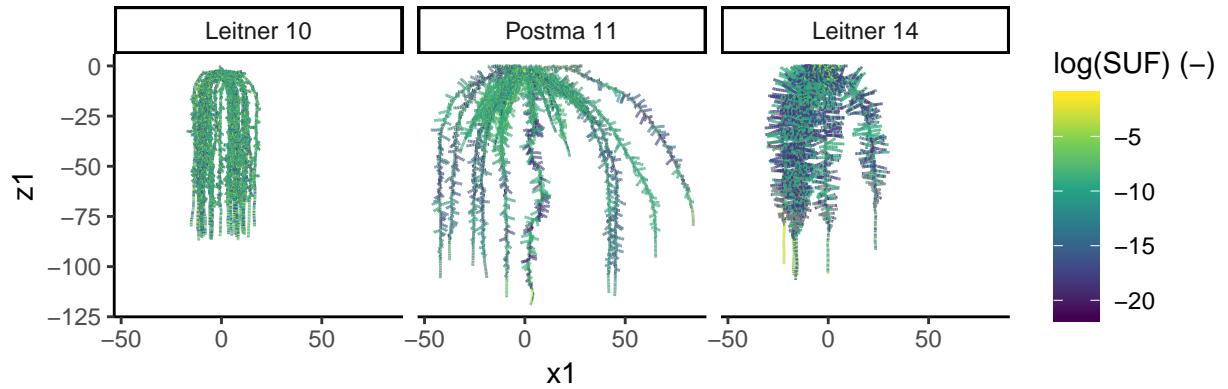


Figure 9: Standard uptake fraction of the different root system architecture

```
# Load the saved results
# new_all_roots <- read.csv("new_all_roots.csv")
```

3.1 Results :

Once the simulation are computed, it is possible to visualize the output on a various form

3.1.1 Standard uptake fraction

3.1.1.1 Where are the segment with the highest contribution over the SUF.

One of the major advantage of the SUF is to estimate where should be the segment that contribute to the water uptake of the plant. The smallest region on which more or less 50% of the water uptake occurs in a soil where the water potential is homogenous along the profile can be calculated.

```
fif <- NULL
for (exa in 1:length(param_choise)) {
  fi <- new_all_roots%>%
    filter(age == "60",
          rep == nrep,
          scenario == 1,
          humid == min(humid),
          ex == param_choise[exa])%>%
    mutate(or_suf = sort(suf1, decreasing = T))
```

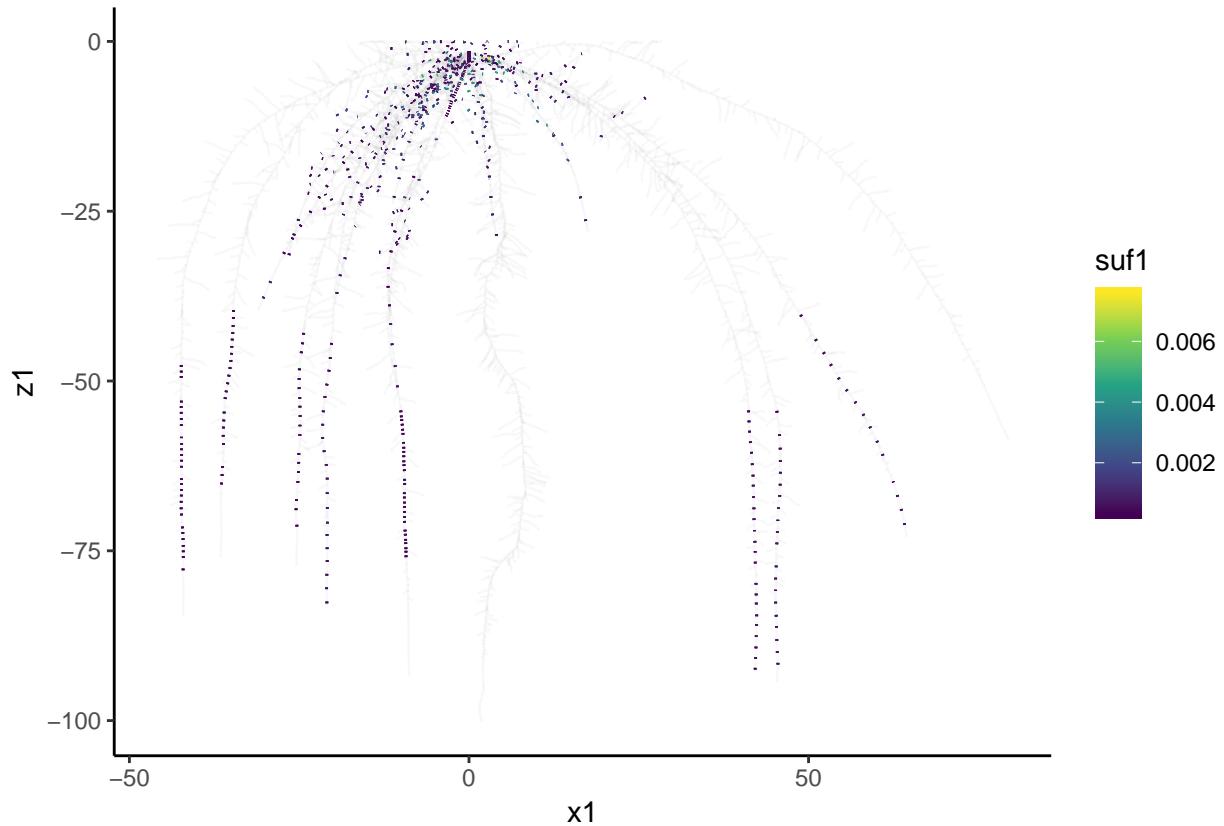


Figure 10: The smallest region on one Postma 2011 RSA example which contains 50 % of the SUF

```

# Distribution of the suf value
# fi%>%
#   ggplot(aes(x=1:nrow(fi), y = or_suf))+
#   geom_line(color="#69b3a2")+
#   geom_area(fill="#69b3a2", alpha=0.5)

i <- 1
k <- 0
while(k <= 0.5){
  k <- k + fi$or_suf[i]
  i <- i + 1
}
fif <- rbind(fif, c(ex = param_choise[exa], suf = fi$or_suf[i]))%>%
  as.tibble()%>%
  mutate(suf = as.numeric(suf))
}

```

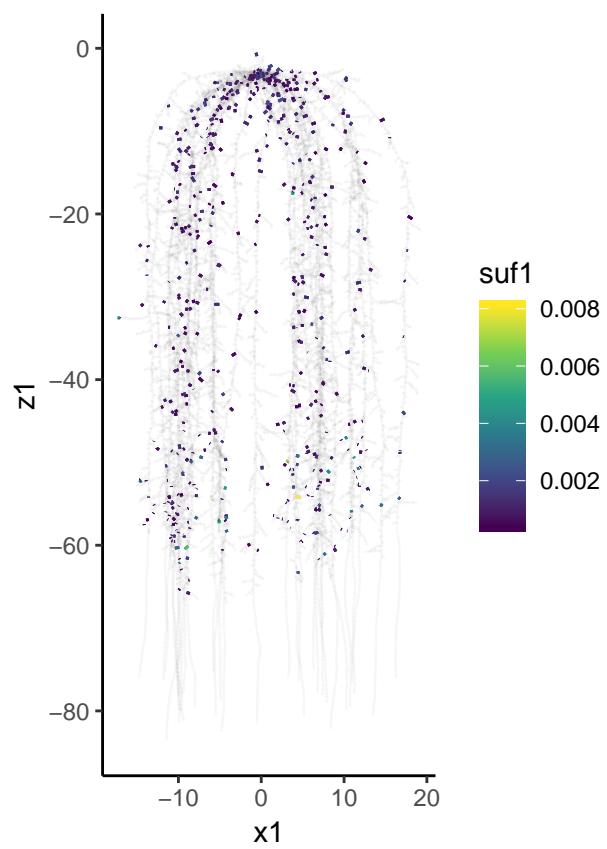


Figure 11: The smallest region on one Leitner 2010 RSA example which contains 50 % of the SUF

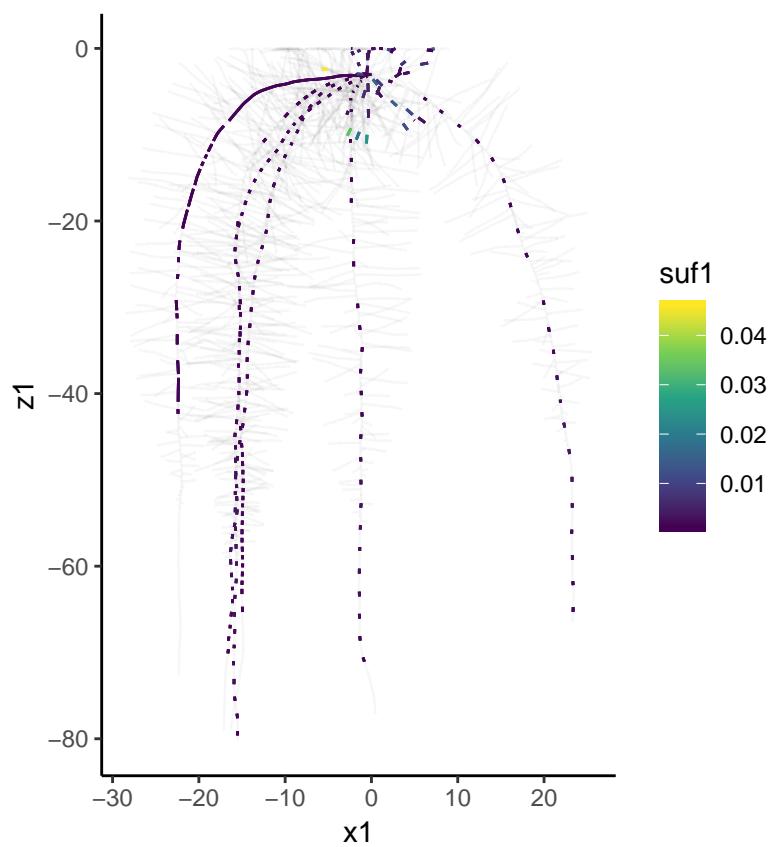


Figure 12: The smallest region on one Leitner 2014 RSA example which contains 50 % of the SUF

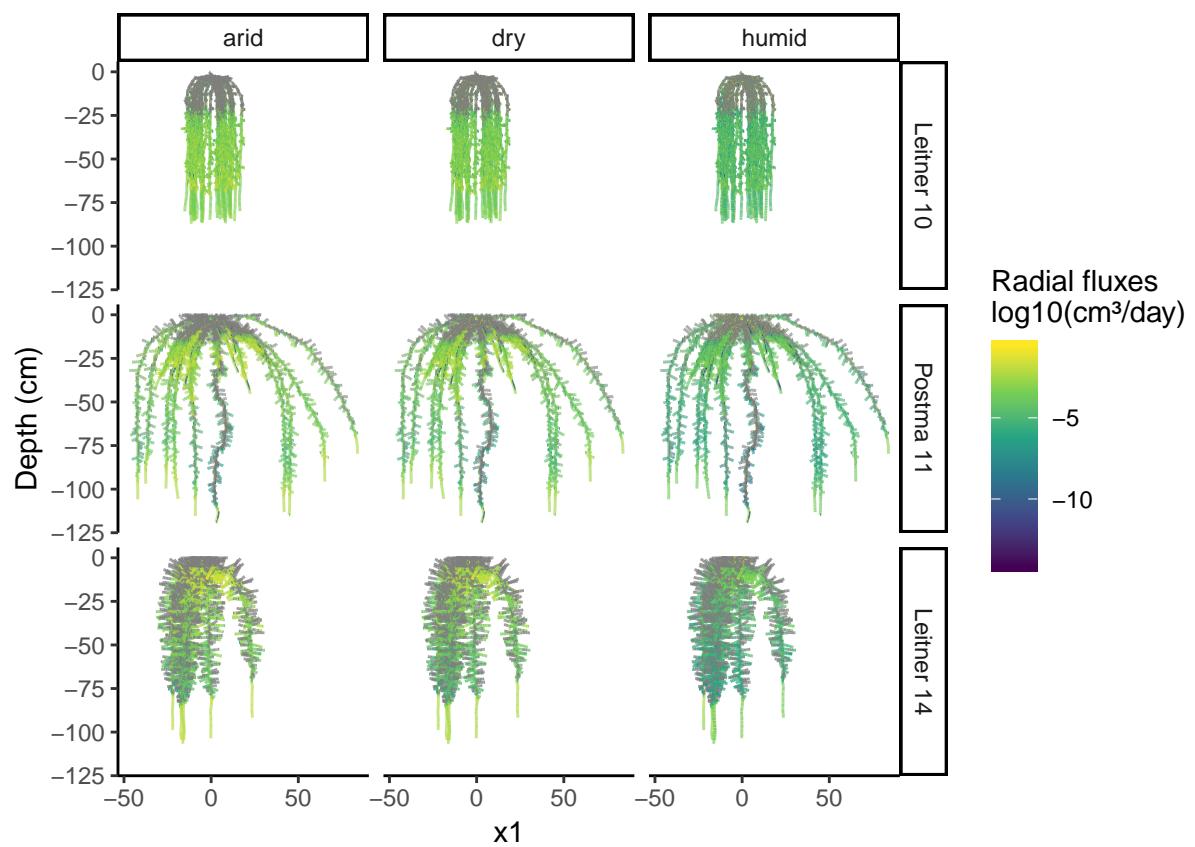


Figure 13: Absorption speed, comparison between the RSA and the soil scenario

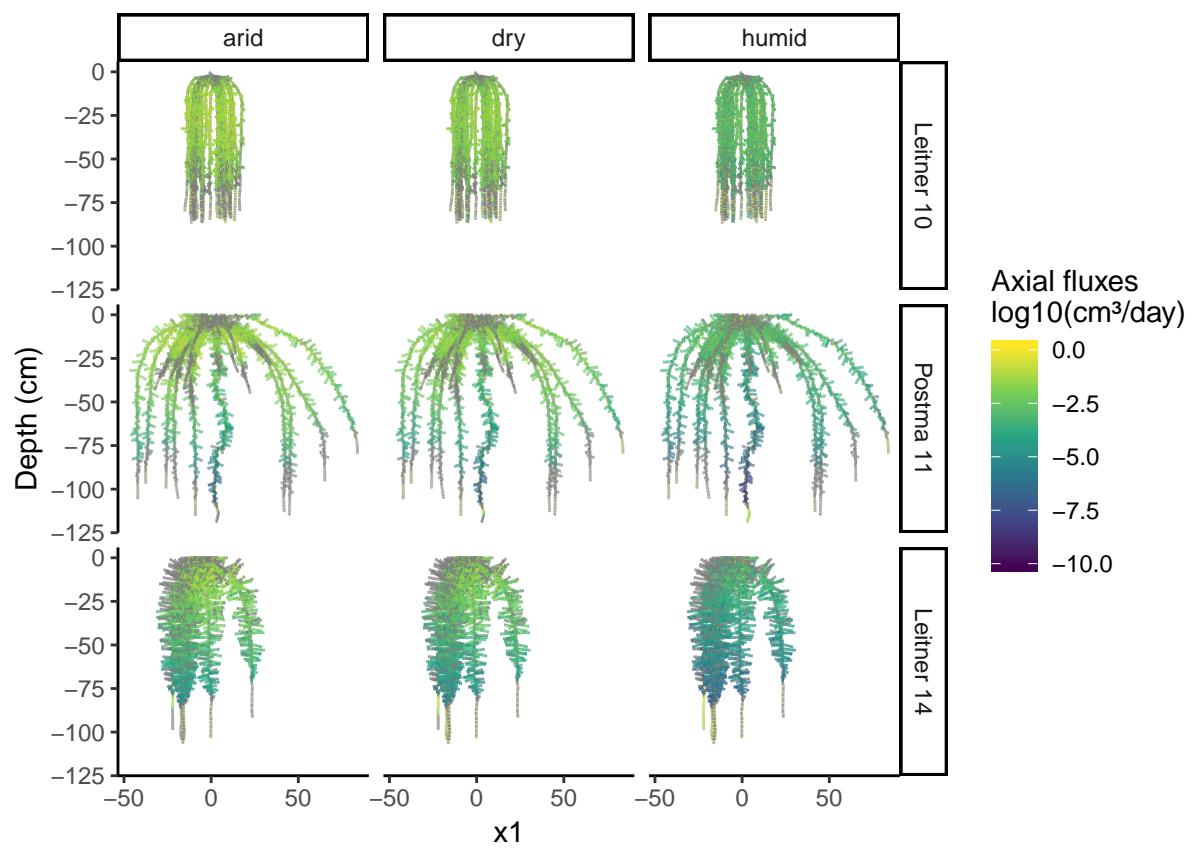
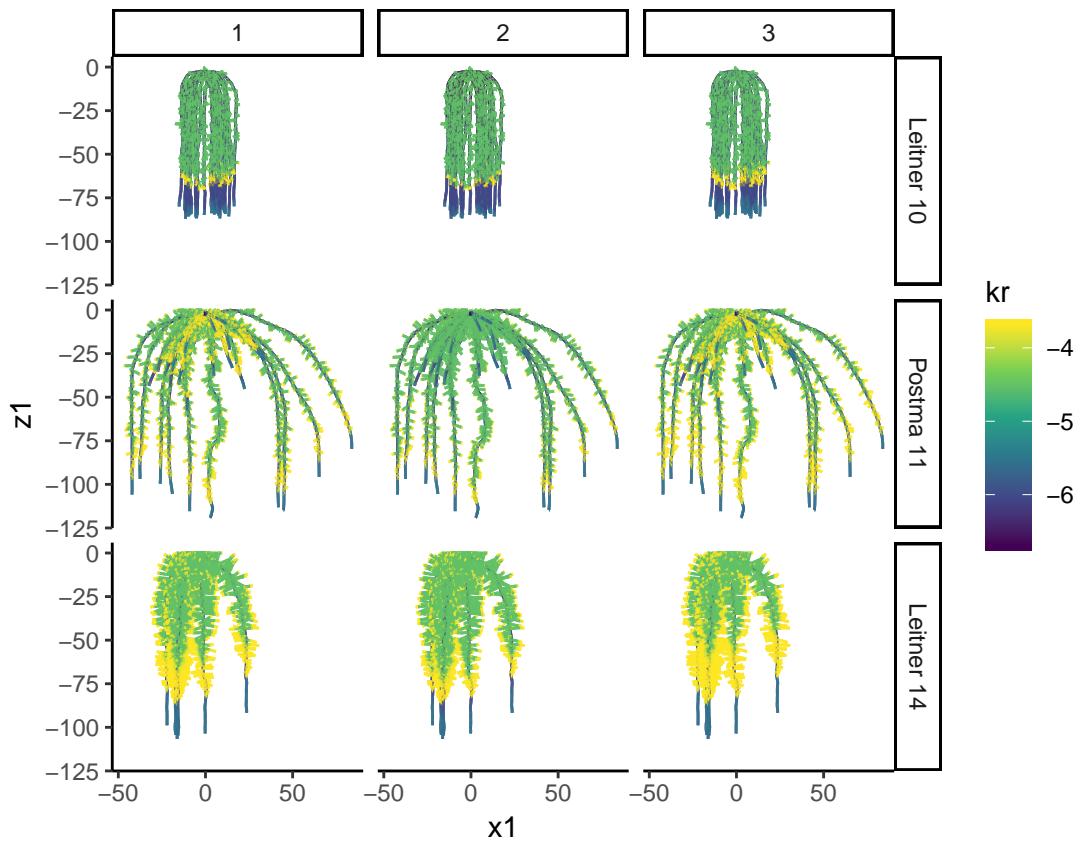
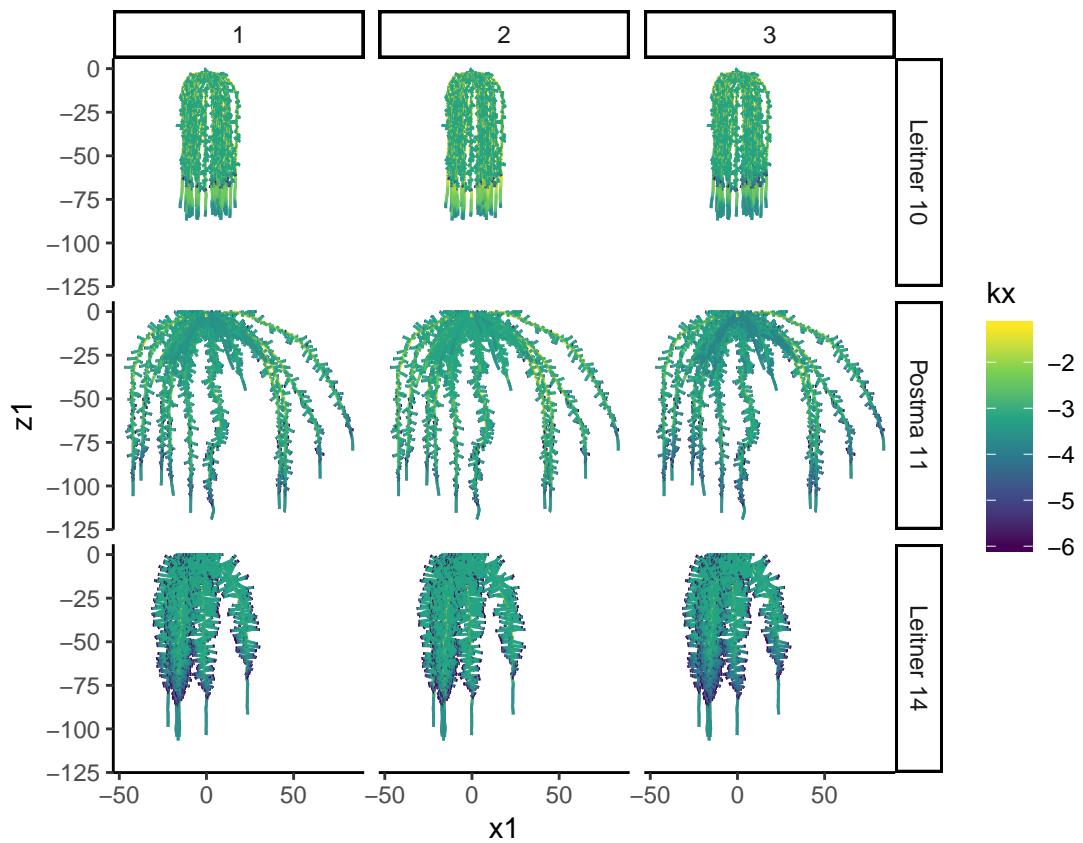


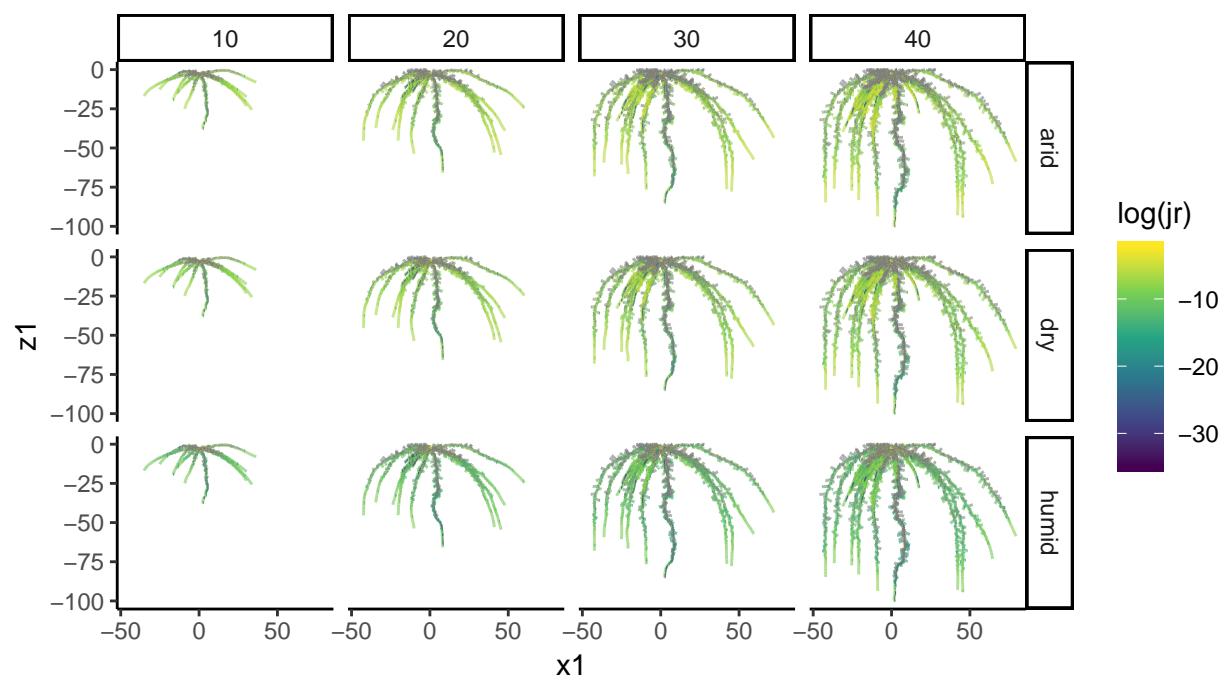
Figure 14: Absorption speed, comparison between the RSA and the soil scenario

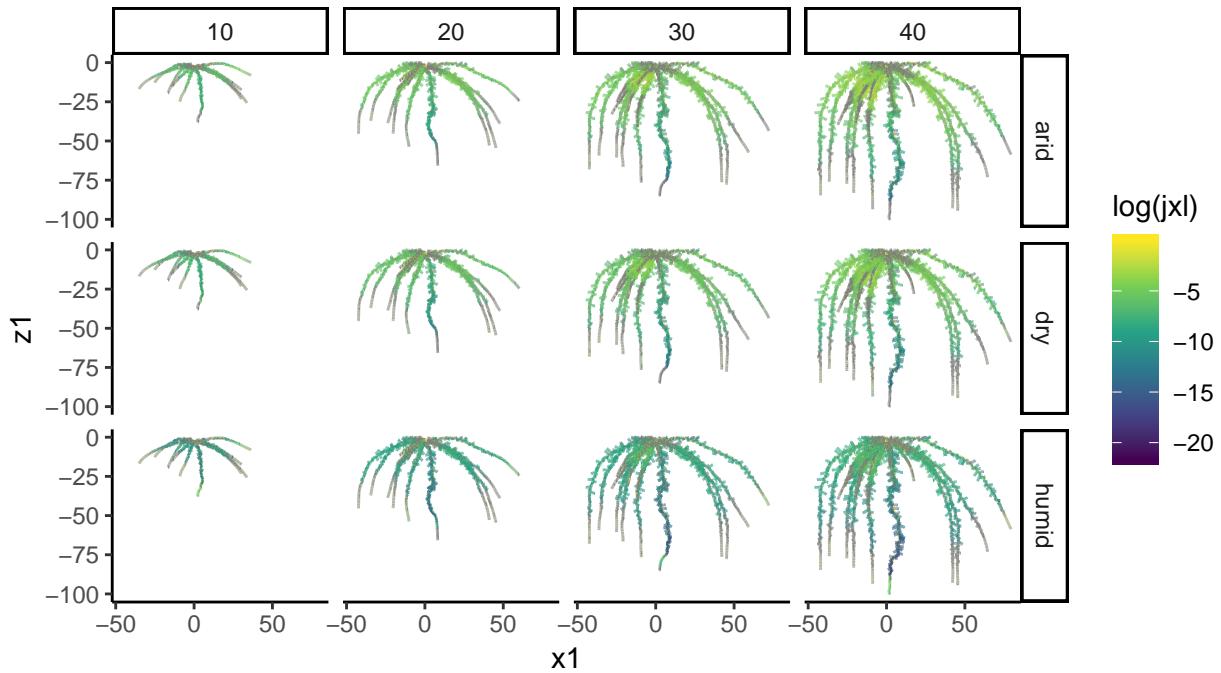
3.1.2 The radial fluxes

3.1.3 The axial fluxes









To look at the different hydraulic macro-properties along the profile depth, we took the mean values of the variable total on every two centimeters.

```
# For the two repetitions on which we have values on each segment,
# we take the mean values of the sum of each variables.
dens <- new_all_roots%>%
  mutate(age = as.numeric(age))%>%
  group_by(type, age, rz1 = round(z1/2)*2, # on every two cm of the profile
          scenario, ex, humid)%>%
  summarise(root = sum(length)/2, # The mean of the sum on the two repetitions
            su = sum(suf)/2,
            su1 = sum(suf1)/2,
            j = sum(jr)/2,
            jx = sum(jxl)/2,
            p = sum(psi)/2,
            kr = sum(kr)/2)

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
dens%>%
  filter(age == 40,
        scenario == 1)%>%
  ggplot(aes(rz1, jx, colour=factor(type))) +
  geom_line(alpha = 0.7) +
  #geom_smooth(se=F) +
  coord_flip() +
  theme_classic() +
```

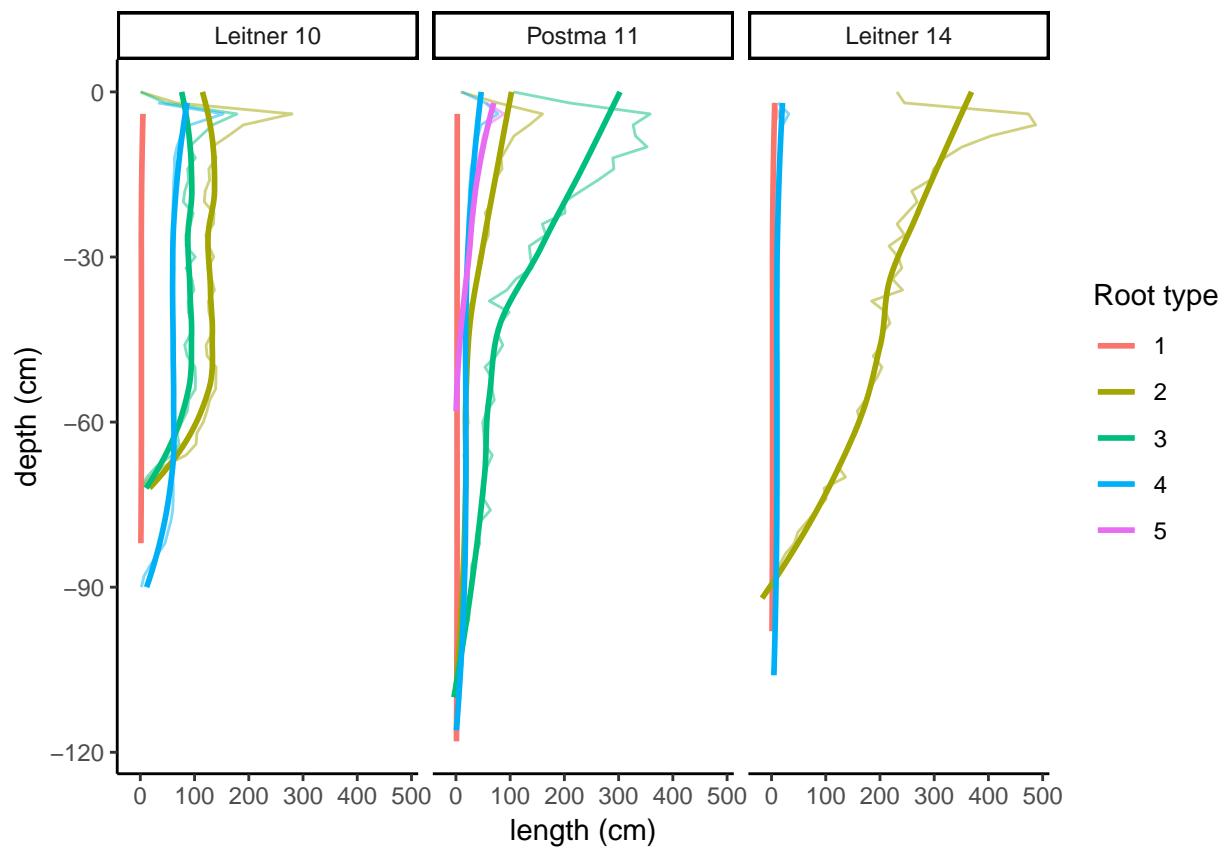


Figure 15: Root length density

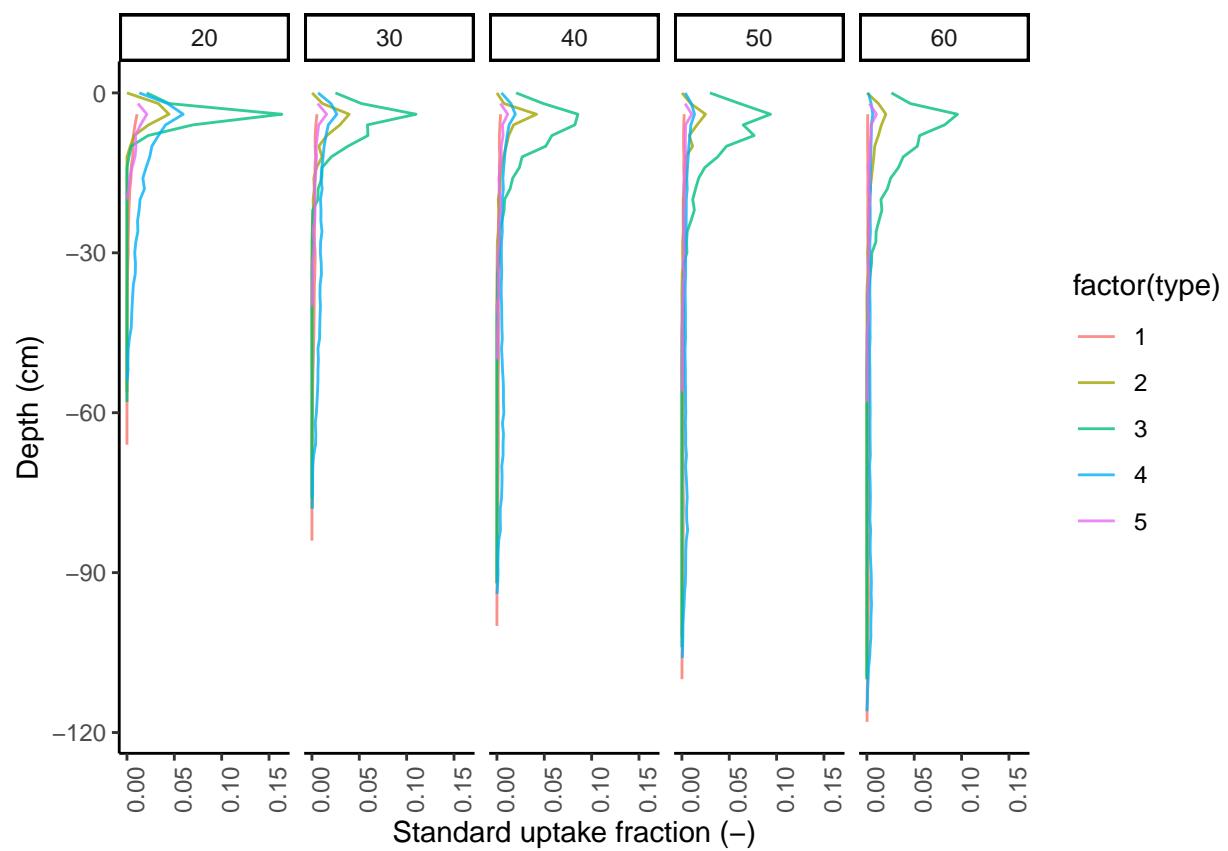


Figure 16: Evolution of the SUF with Leitner 2010 RSA parameter

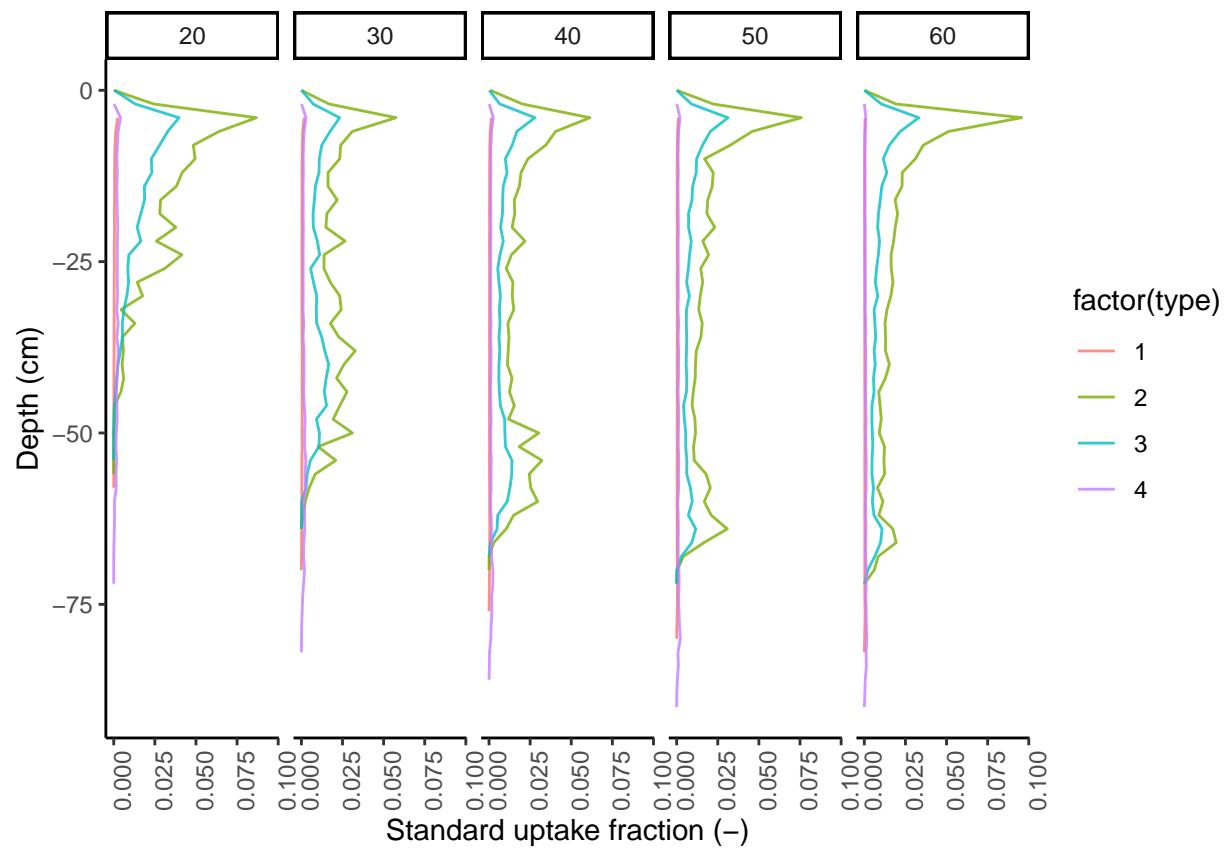


Figure 17: Evolution of the SUF with Postma 2011 RSA parameter

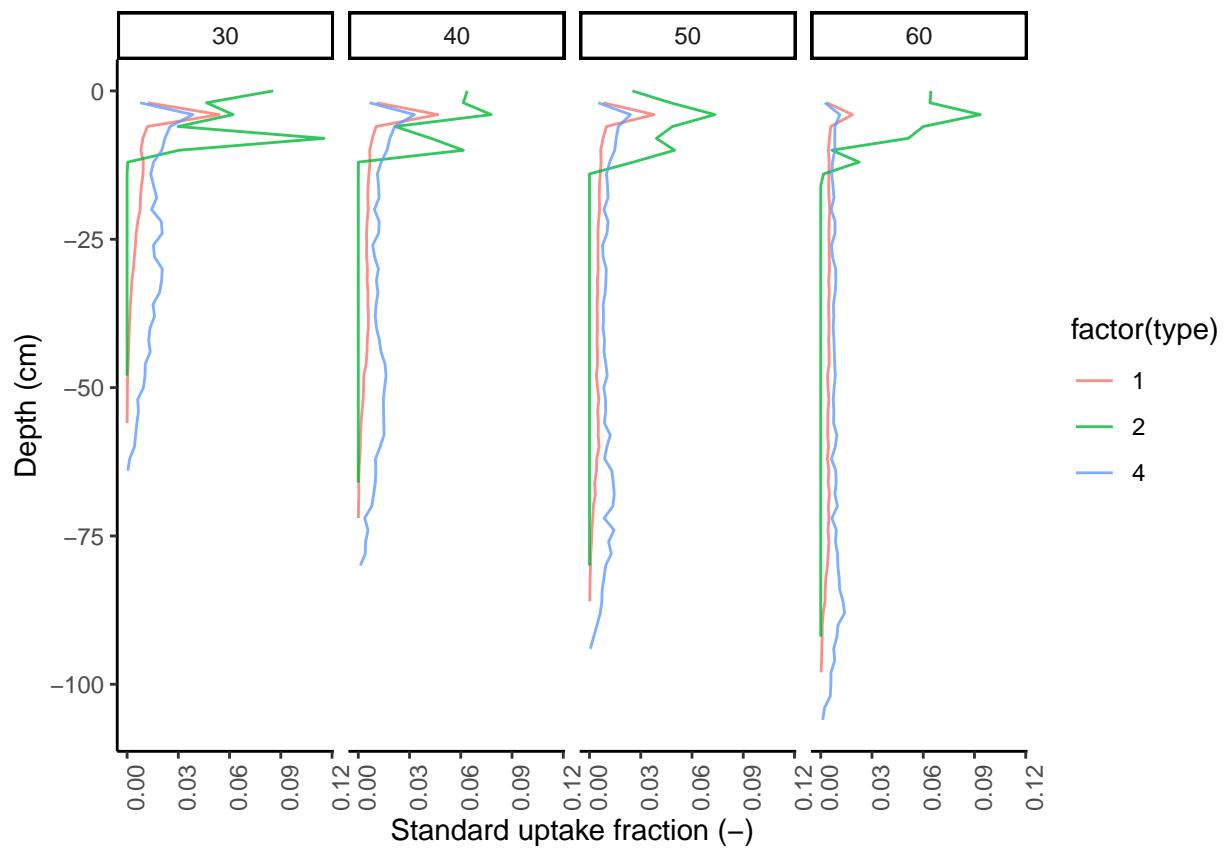


Figure 18: Evolution of the SUF with Leitner 2014 RSA parameter

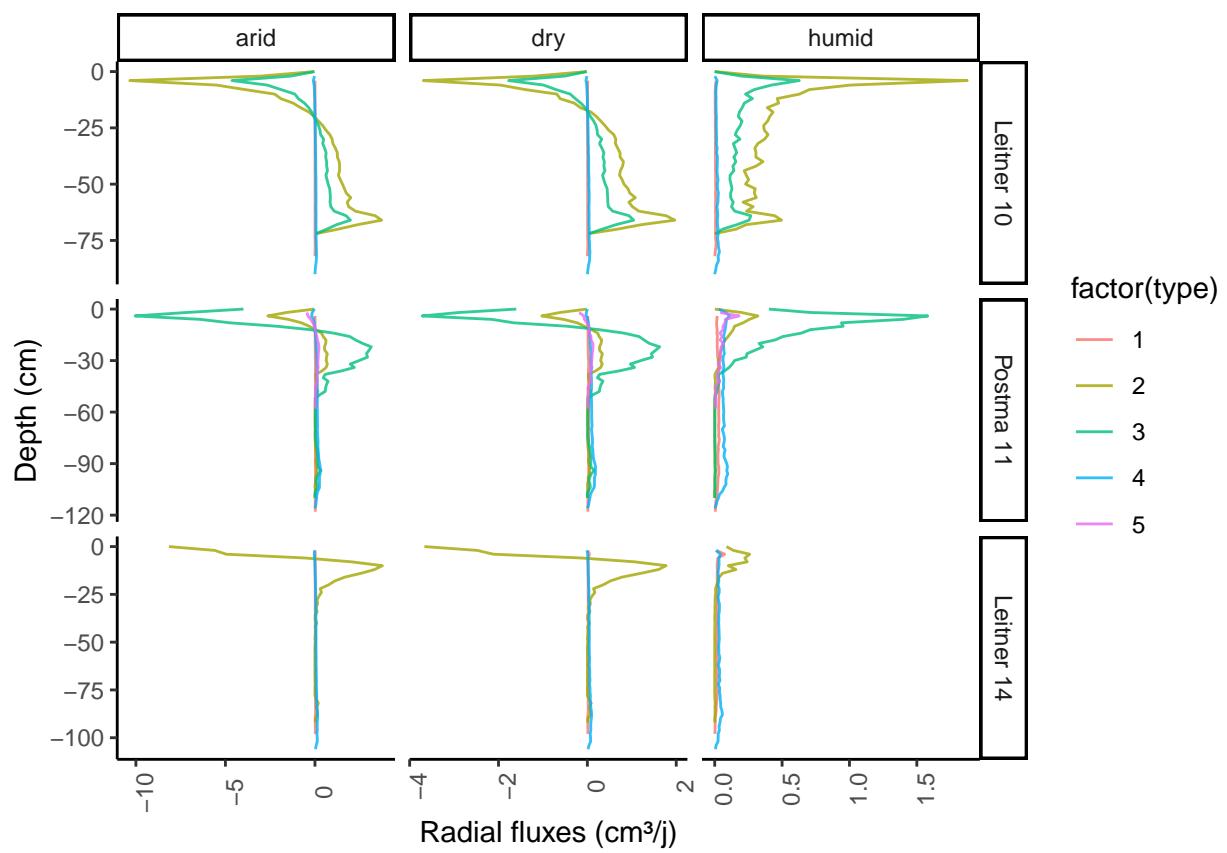


Figure 19: Comparison of the radial fluxes between the different soil scenarios and the root system architectures

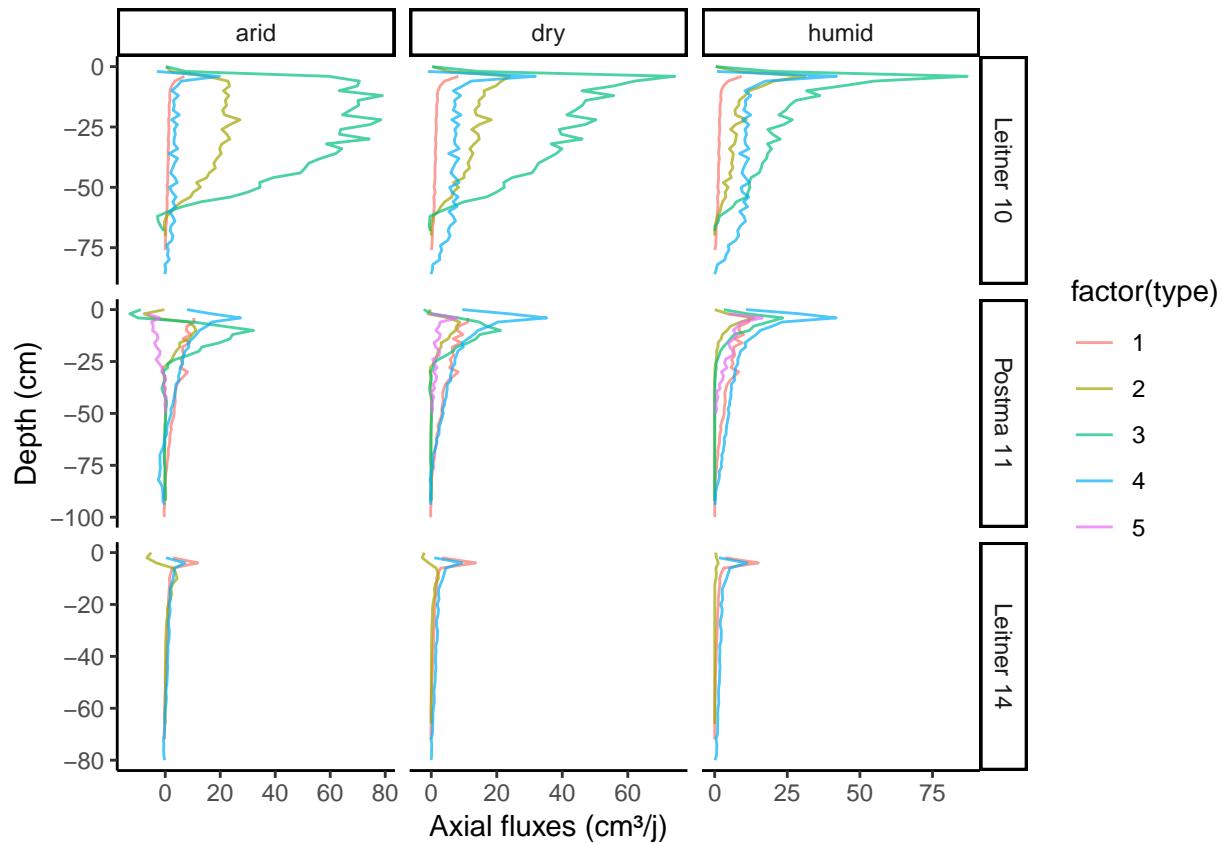


Figure 20: Comparison of the axial fluxes between the different soil scenarios and the root system architectures

```

xlab("Depth (cm)") +
ylab("Axial fluxes (cm³/j)")+
facet_grid(ex~humid, scales = "free", labeller = as_labeller(babbel))

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
Transpiration<- results%>%
  mutate(humid = as.character(humid))%>%
  group_by(age, rep, scenario, humid, ex)%>%
  summarise(Tpot = max(tpot, tact),
            Tact = min(tpot, tact))

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

```

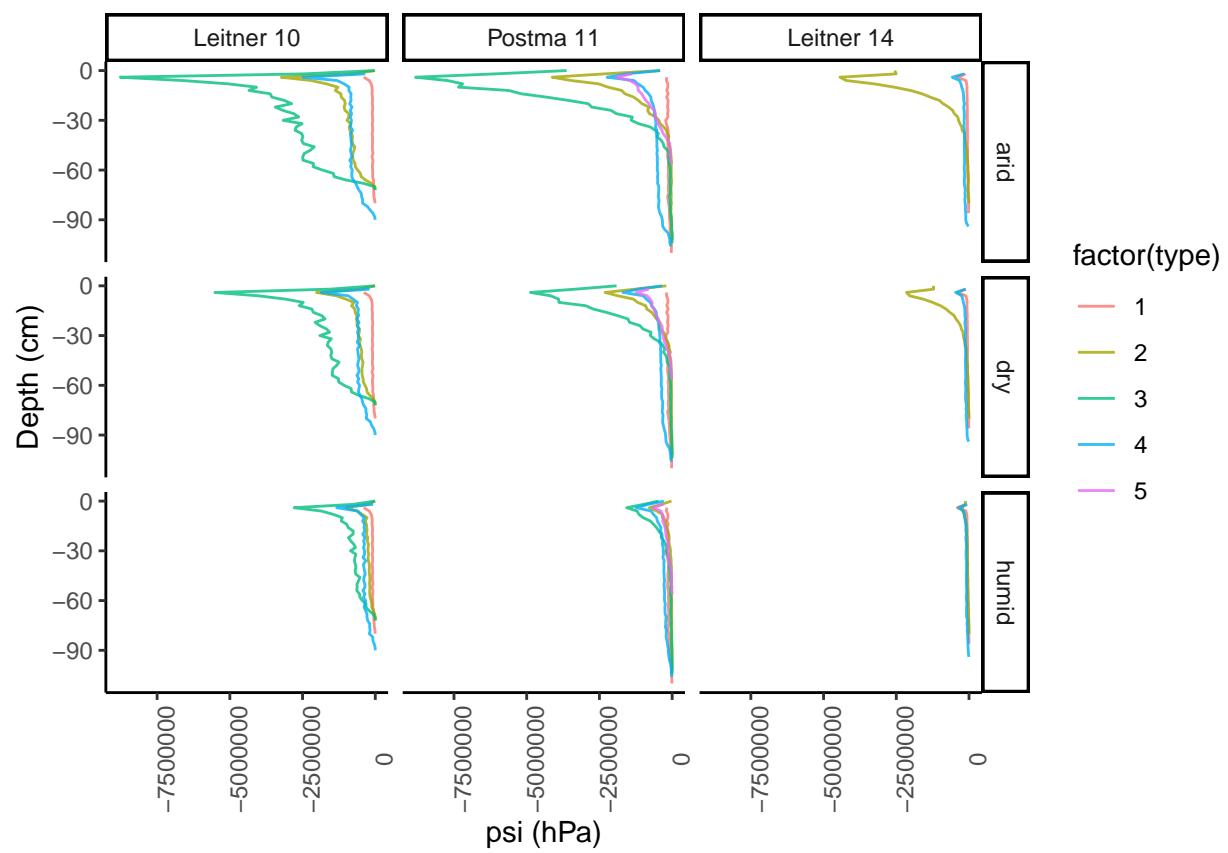


Figure 21: Water pressure head at the soil-root interface

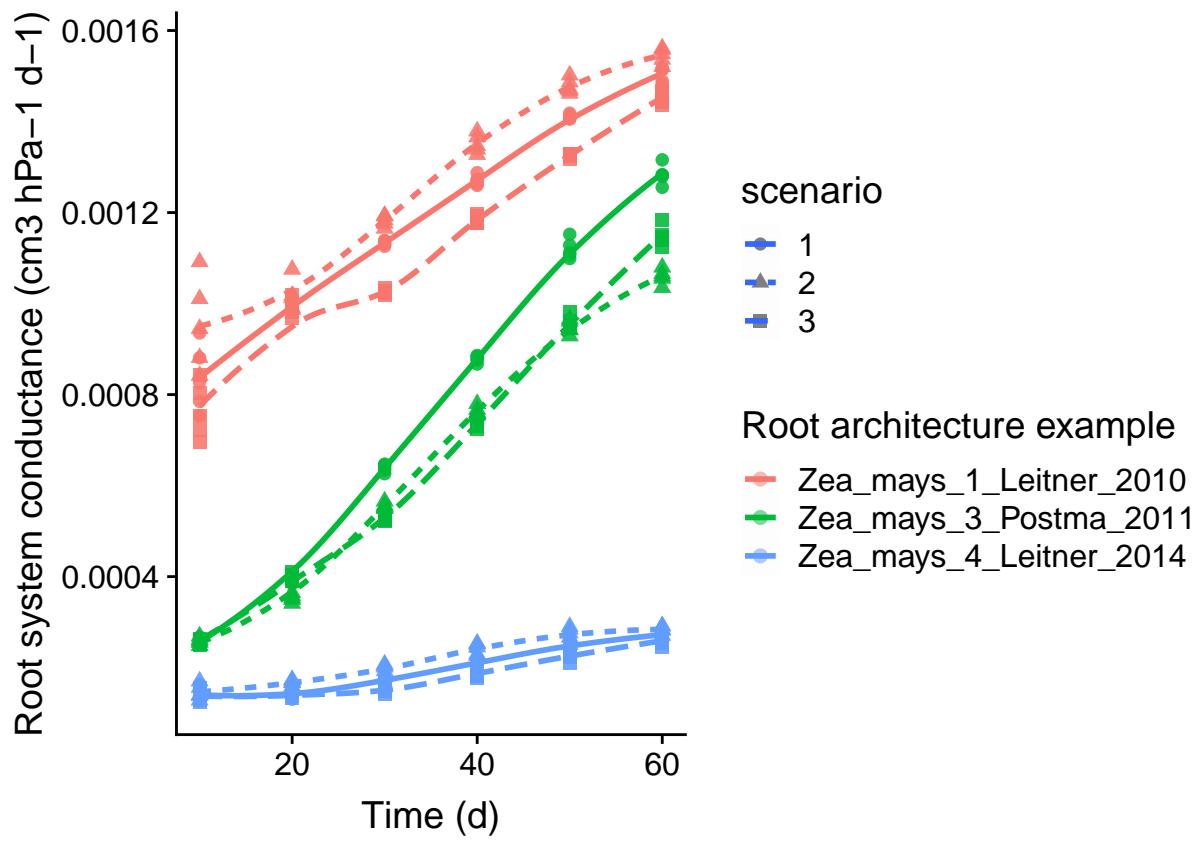
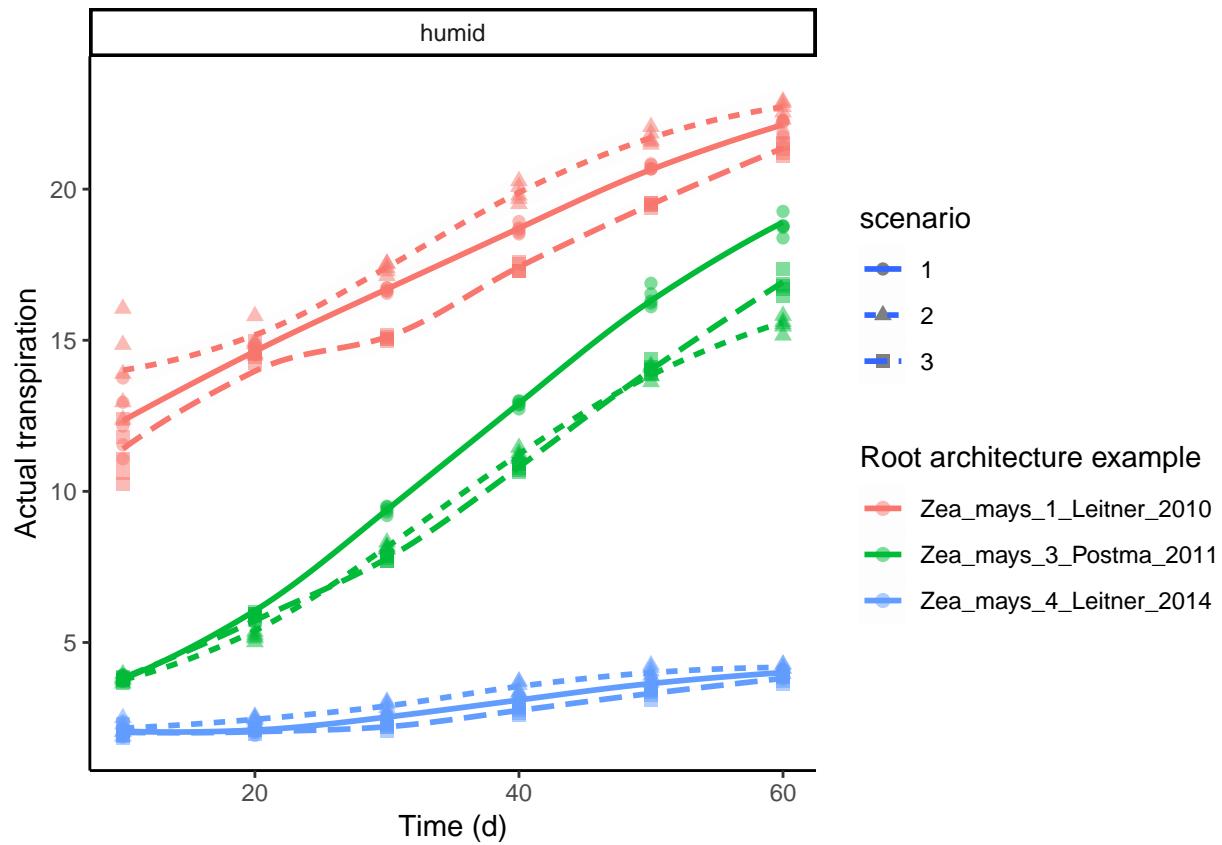
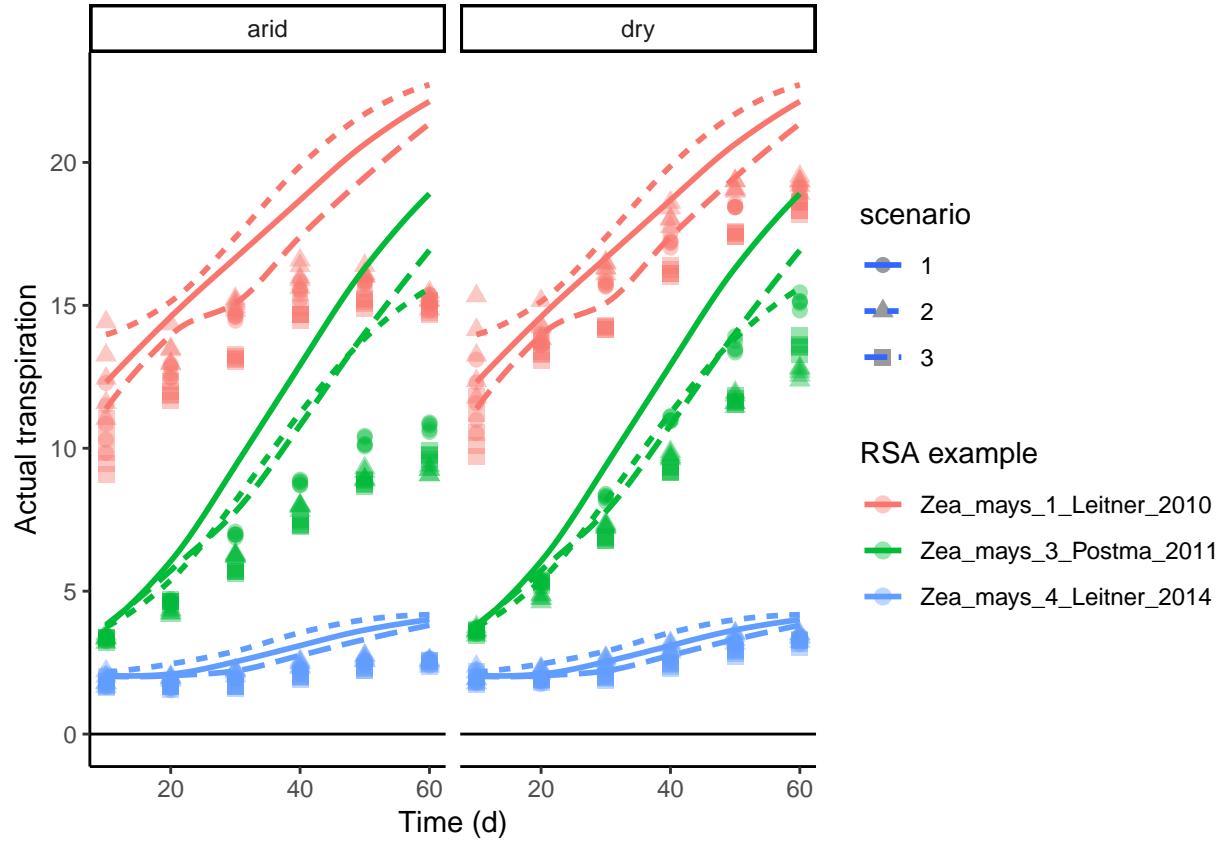


Figure 22: Evolution of the whole root system conductance. Effect comparison between root system architecture and root conductivity modification.



The actual transpiration under drought whith two scenarion of conductivities

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



The different line represent a potential transpiration.

4 Reference