

Getting a Functionnal-Structural Plant Model into a comprehensive R pipeline: MARSHAL-pipeline

Adrien Heymans, Felicien Meunier, and Guillaume Lobet

décembre, 2018

Introduction

MARSHAL, the MAize Root System Hydraulic Architecture soLver that combines a root architecture model, CRootBox (???), the water flow equation resolution algorithm of Doussan et al. (1998) (Doussan, Vercambre, and Pagès 1998), improved by including the analytical solution of Landsberg and Fowkes (1978) extended for root systems (Meunier et al. 2017) and the macroscopic parameter calculation of Couvreur et al. (Couvreur, Vanderborght, and Javaux 2012). MARSHAL calculates root system conductance, 1 to 3-D uptake distribution and other upscaled variables (plant leaf water potential or transpiration) for any combination of structural and functional traits of maize.

How to use MARSHAL in a R environment

To run this example, the ‘marshal’ package is required. This package can be found on GitHub. Or by simply execute the following line in the R environment:

```
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 additional set of function has to be loaded manually.

```
library(data.table)
library(Matrix)
library(tidyverse)
library(readxl)
library(cowplot)
library(xml2)
library(knitr)
`%!in%` <- compose(`!`, `%in%`)
source("R/io_function.R")
```

Boundary condition of MARSHAL.

The inputs of 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 (boundary condition).

To illustrate the potential of MARSHAL, we created a series of 27 scenarios, from three root system architecture (RSA) examples, three different hydraulics properties of the roots, and three vertical distribution of the soil water potential due to three drying coefficients ($3\text{soil} \times 3\text{RSA} \times 3\text{hydraulicconductivities}$).

Hydraulic properties of the root system

The hydraulic conductivities 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. paper (1998). To show some possibilities, three scenarios in total were built from this Doussan et al. table of conductivities (figure 1). The first, 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, or in other words, the maturity of the root anatomy, insted of being advance towards the apex, is moved of 5 cm backwards.

The conductivities of the different root type.

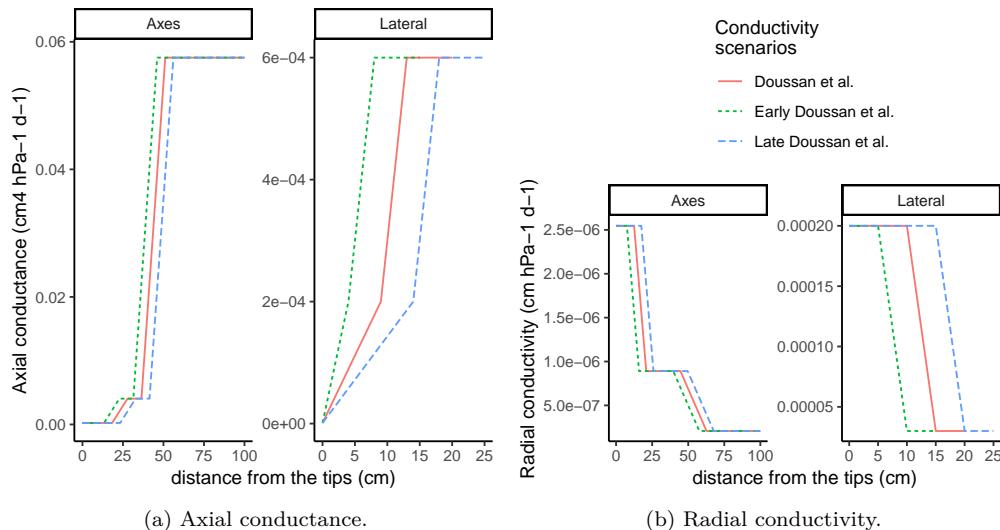


Figure 1: Conductivities properties on axes and lateral root type.

Root system architecture with CRootBox

Modelling root system architecture can be achieved with CRootBox (Schnepf et al. 2018). Some examples of the parameter files for a couples of specieses are details in the *modelparameter* folder, which is linked to the CRootBox repository.

For the purpose of the example we choose to work with three parameter files of maize:

- *Zea_mays_3_Postma_2011*
- *Zea_mays_1_Leitner_2010*
- *Zea_mays_4_Leitner_2014*

The following lines of code allows to create five unique root system architectures for each parameter files. It computes the root growth during 60 days and load every ten days the architecture information in a data table. The time resolution can be increased inside the CRootBox implementation, if needed. For the implementation of this example, the root growth has been restricted to the below ground level for a more realistic point of view. Furthermore, some segments were not build up to the whole root system because they were shorter than the spatial resolution of the root system.

```

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

# List of three CrootBox parameter files 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 #
  #####
    if(Sys.info()['sysname'] == "Darwin"){
      system("chmod 777 www/a-mac.out")
      system("www/a-mac.out")
    }
    else if(Sys.info()['sysname'] == "Windows"){
      system("www/a.exe")
    }
    else{
      system("chmod 777 www/a.out")
      system("www/a.out")
    }
    # Error could occurs at this point depending on your
    # operating system. If so, please compile CRootBox on
    # your device. The instruction can be found:
  # https://github.com/MARSHAL-ROOT/marshall-pipeline/tree/master/17_06%20CRootBox

    # Get all the .txt files exported by CRootBox
    fls <- list.files("./")
    fls <- fls[grep("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
      # Save the temporary data in a permanent file to use.
      all_roots <- rbind(all_roots, temp)
    }
  }

  # Saving the root system information in a .csv
  write_csv(all_roots, "all_roots.csv")
}

```

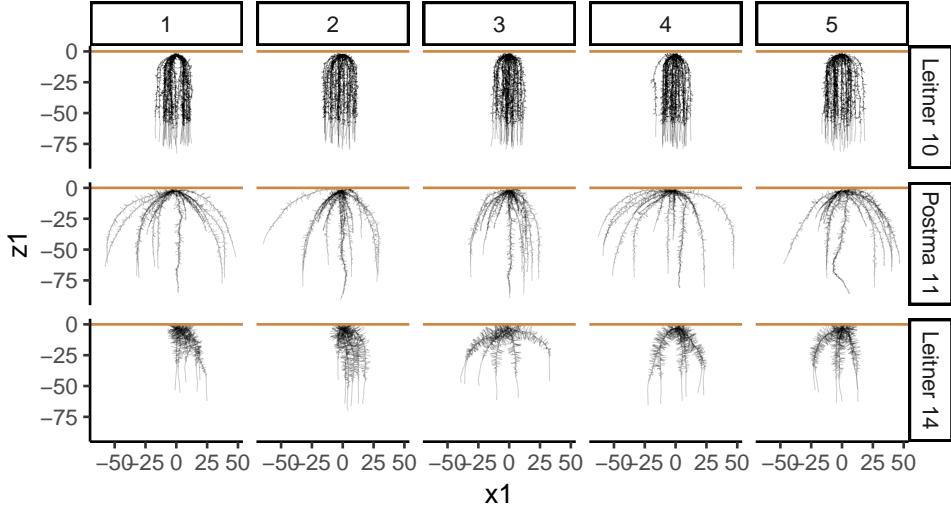


Figure 2: Projection in two dimensions of the different root systems computed with CRootBox at the age of 30 days

```
# To gain time once you have a all_roots.csv file created
# Loading the .csv file can be achieve with:
all_roots <- read.csv("all_roots.csv")
```

The figure 2 shows the root system architectures that comes out CRootBox at half of the simulation time (30 days).

Architectural phenes

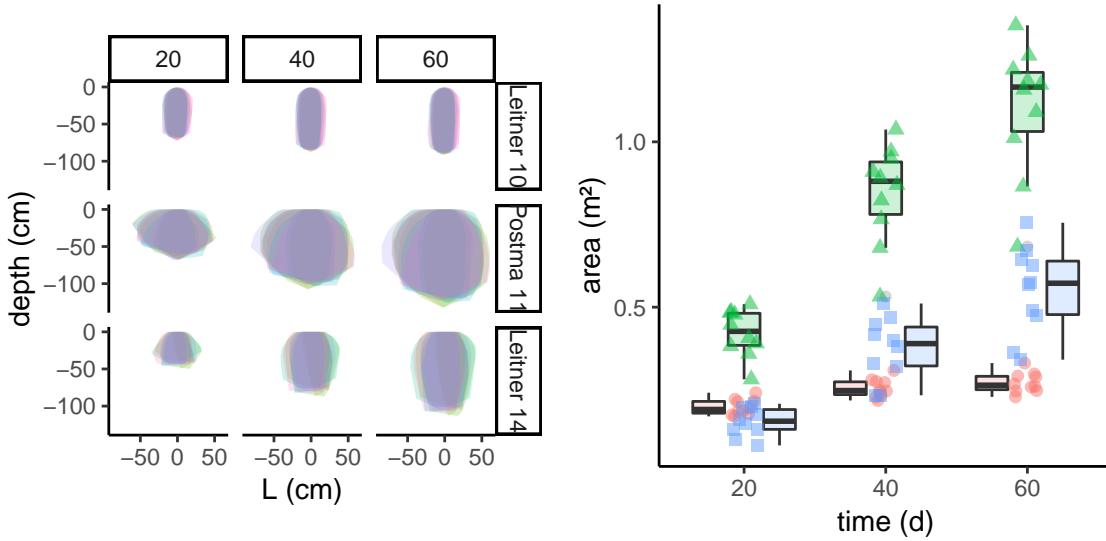
Once the complete root system architecture is computed there are a number of architectural phenes that we can accessed, such as:

- the area of the convex hull
- the rooting depth
- the root length density along the soil profile
- the number of individual roots per root system
- the total length of the root system
- ...

The convex hull

The convex hull area is a classical root system descriptor. It gives a qualitative information about root exploration.

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



(a) Superposition of all the convex hull repetitions.

(b) The area of each convex hull.

Figure 3: Area of the convex hull for the different root system architecture examples. \triangle = Postma 2011, \square = Leitner 2014, and \circlearrowleft = Leitner 2010 RSA example

The maximal rooting depth

Thanks to the detailed information coming from CRootBox, it is possible to measure some root phenes along the z axis. The maximal rooting depth is a interesting phenotypic values. This can be used in other models such as Feddes et al. (1978) to estimate root water uptake.

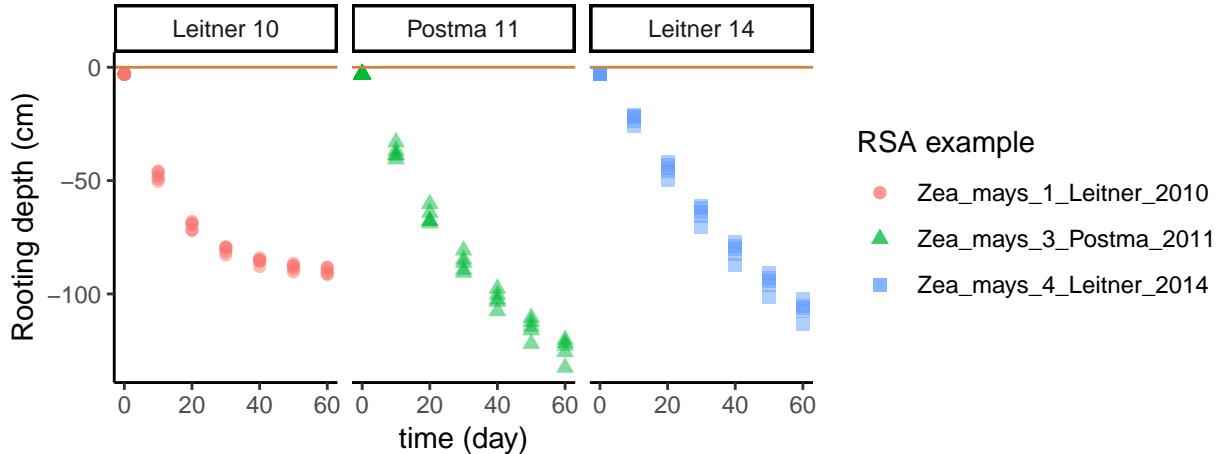


Figure 4: Rooting depth of the different root system examples

```
# 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))

RLD2 <- all_roots%>%
  mutate(age = as.numeric(age))%>%
  group_by(age, rz1 = round(z1/2)*2, # on every two cm of the profile
          ex)%>%
  summarise(root = sum(length)/nrep)
```

Root length density

Root length density is a major phenotypic trait measured in field when below ground information are needed. In this case the total root length is computed every ten days.

The water soil potential

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

The drying function is :

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

- 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 capacity for a silty soil.

```
# 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]
```

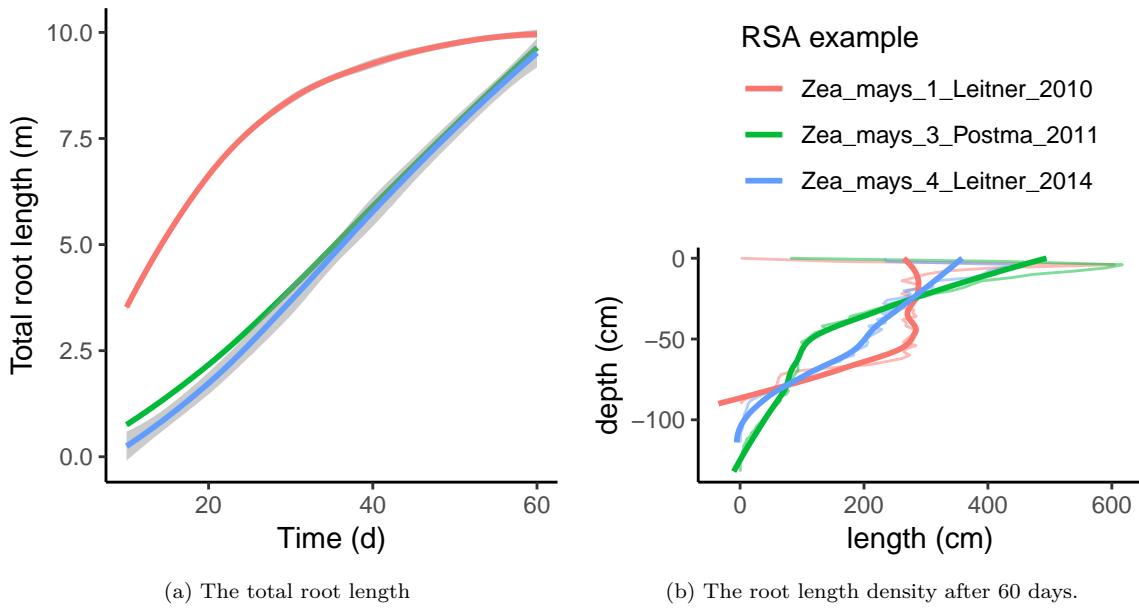


Figure 5: The root lenght

```

endTime = max(full_time)
ntimesteps = endTime/time_step

# 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)
}

```

The water potential at the collar

Expressed in hecto-Pascal, the pressure head define the transpiration strenght that pulls the water out of the root system. here we take the default value of -15.000 hPa

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

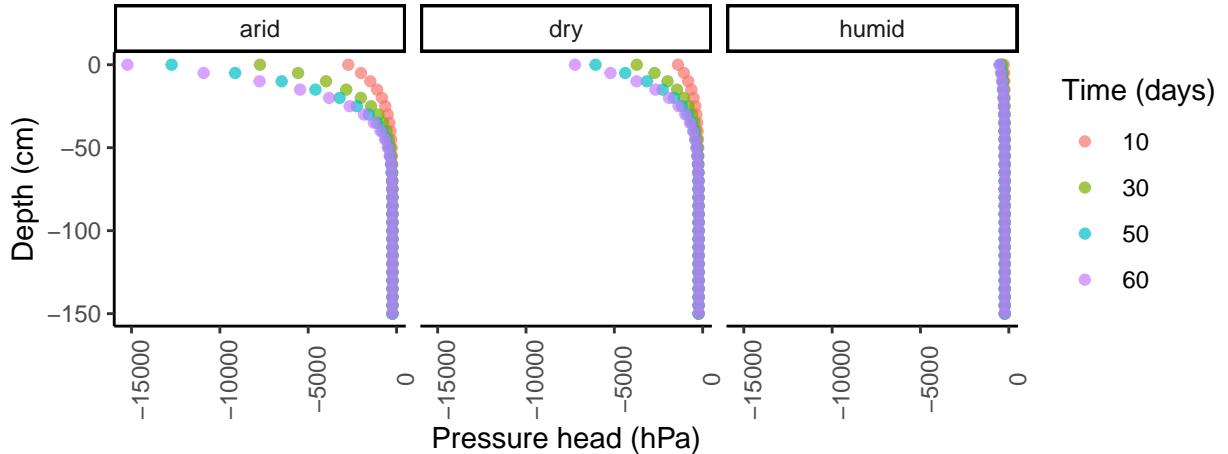


Figure 6: Water potential evolution during the simulation time along the soil profile

MARSHAL

MARSHAL provides an estimation of the conductance of the whole root system as well as the actual and the potential transpiration (based on the soil profile). Here we call the MARSHAL function `getSUF` that will solve the hydraulic part of the root system for each soil, conductivities, root system architecture scenarios and at each time step of the process.

```
new_all_roots <- NULL
results<- NULL
k <- 1
# Computing the root system hydraulic architecture
# 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

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

```

Results :

Once the hydraulic part has been associated to each root segment and the macroscopic properties have been computed, we can visualize them in different ways.

Standard uptake fraction (SUF)

The Standard Uptake Fraction (SUF) is the relative distribution of root water uptake between root segments when water is equally available in space (units: %) (Couvreur, Vanderborght, and Javaux 2012).

One of the major advantage of the SUF is to estimate where should be the segments which contribute to the water uptake of the plant. Here we focuces on the smallest regions on which 50% of the water uptake occurs in a soil where the water potential is homogenous along the profile.

```

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

  i <- 1
  k <- 0
  while(k <= 0.5){
    k <- k + fi$or_suf[i]
    i <- i + 1
  }
}

```

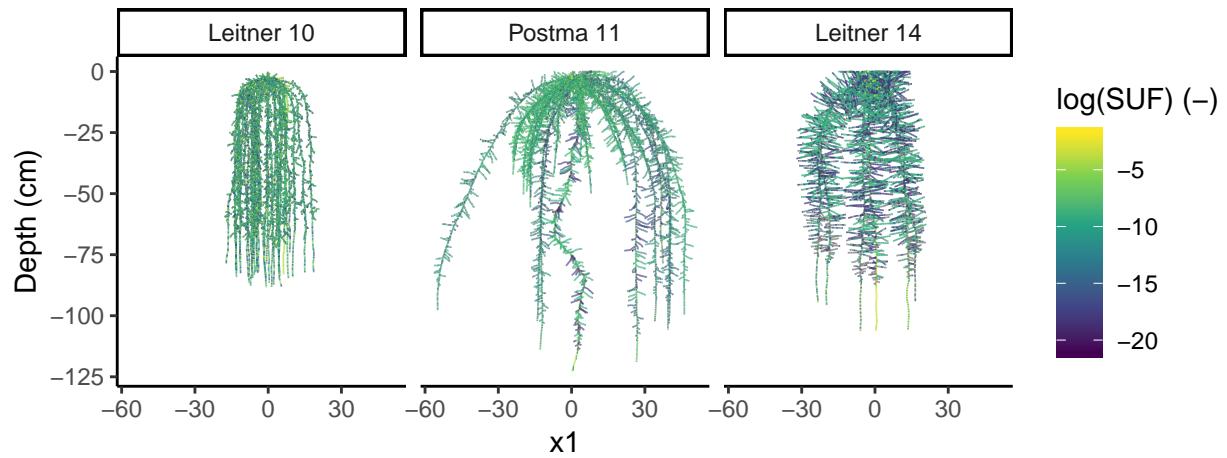


Figure 7: Standard uptake fraction of the different root systsem architecture

```

}
fif <- rbind(fif, c(ex = param_choise[exa], suf = fi$or_suf[i]))%>%
  as.tibble()%
  mutate(suf = as.numeric(suf))
}

```

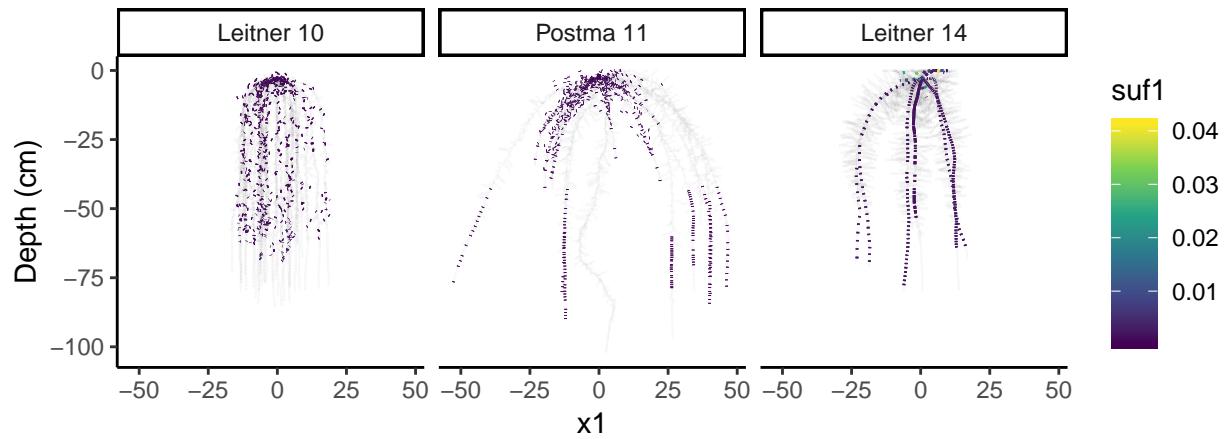


Figure 8: The smallest region on the root system hydraulic architecture which contains 50 % of the SUF

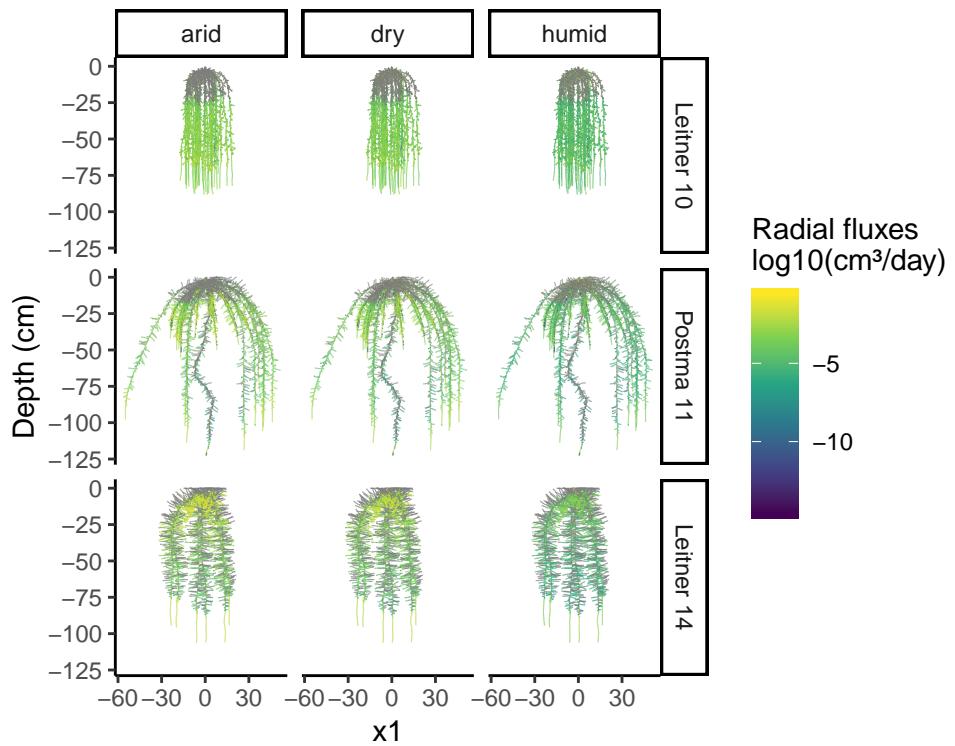


Figure 9: Absorption speed, comparison between the RSA and the soil scenario

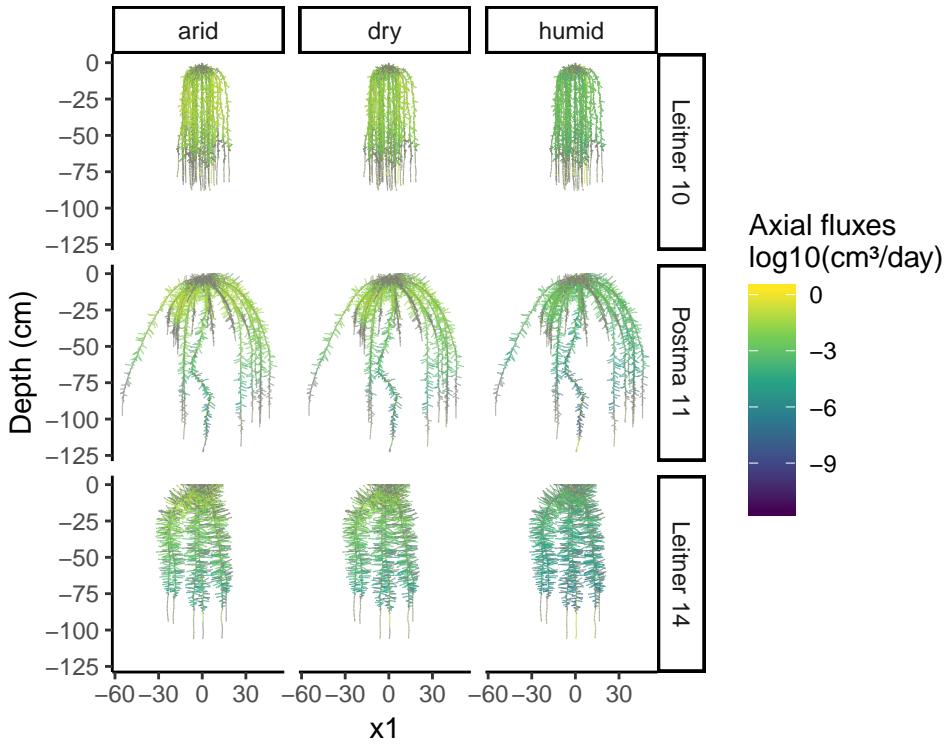


Figure 10: Absorption speed, comparison between the RSA and the soil scenario

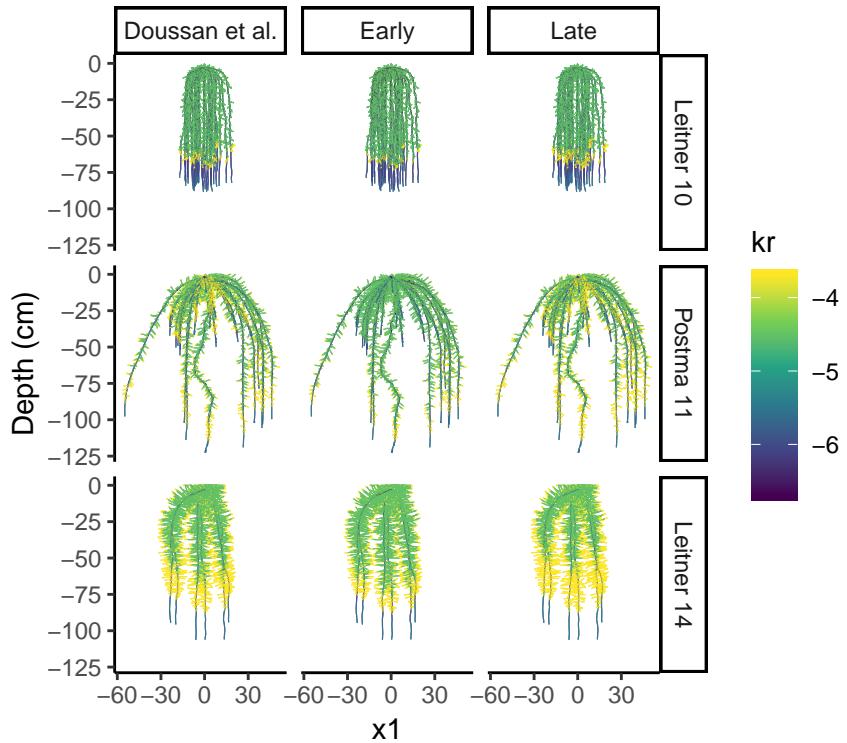


Figure 11: The radial conductivity (Kr) on root segment, comparison between the different conductivity scenario and main root system architecture examples

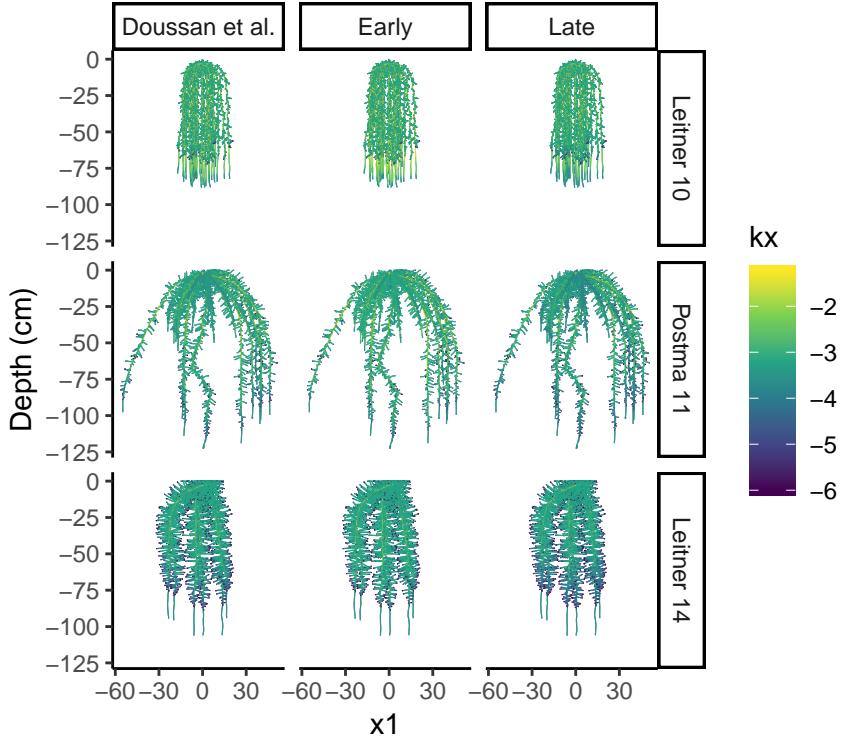
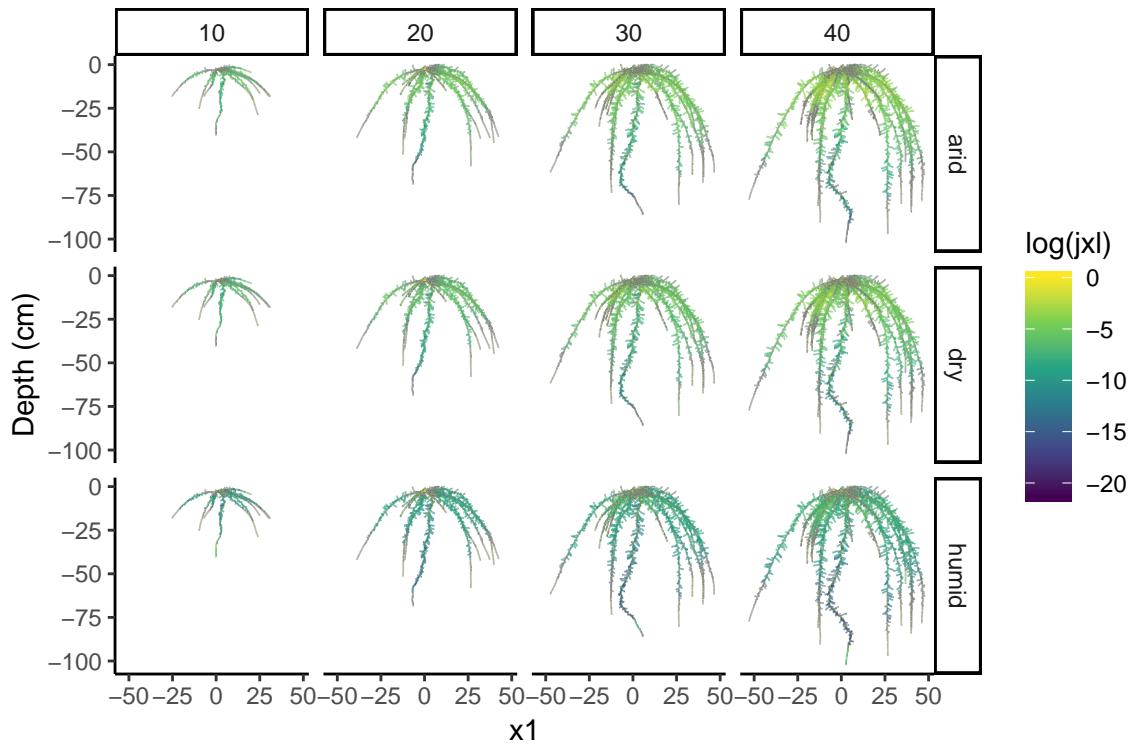


Figure 12: The axial conductance (K_x) on root segment, comparison between the different conductivity scenario and main root system architecture examples



To look at the different hydraulic macro-properties along the profile depth, we took the mean values of the

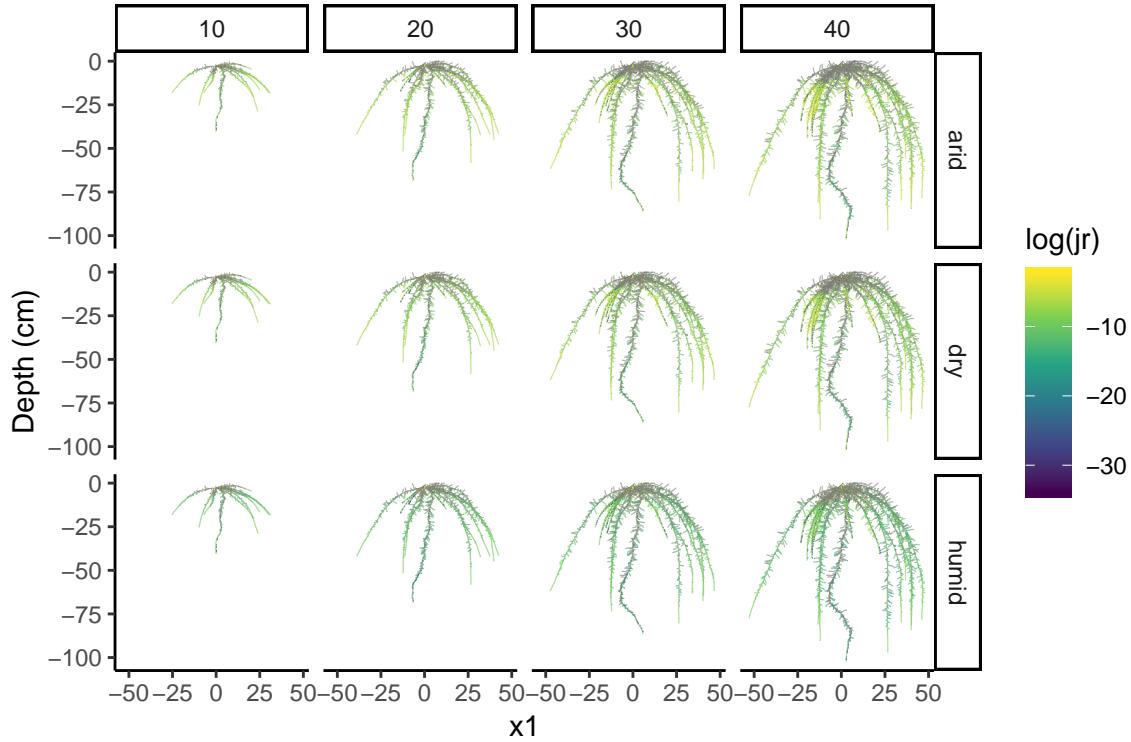


Figure 13: Absorption dynamics due to different soil scenarios on the Postma root system architecture example

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,
            sud = sum(suf/length)/2,
            su1 = sum(suf1)/2,
            j = sum(jr)/2,
            jx = sum(jxl)/2,
            p = sum(psi)/2,
            kr = sum(kr)/2)

dens%>%
  filter(age == 40,
        scenario == "Doussan et al.")%>%
  ggplot(aes(rz1, jx, colour=factor(type))) +
  geom_line(alpha = 0.7) +
  #geom_smooth(se=F) +
  coord_flip() +
  theme_classic()+
  xlab("Depth (cm)") +
```

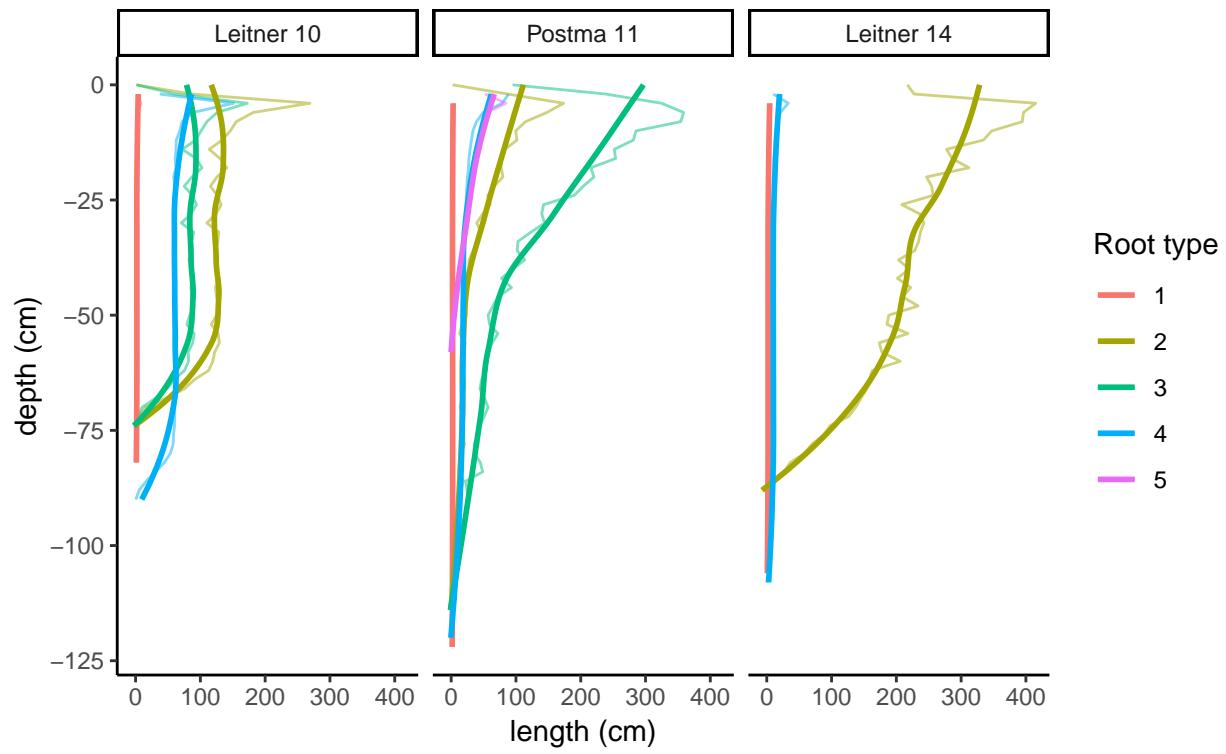


Figure 14: Root length density

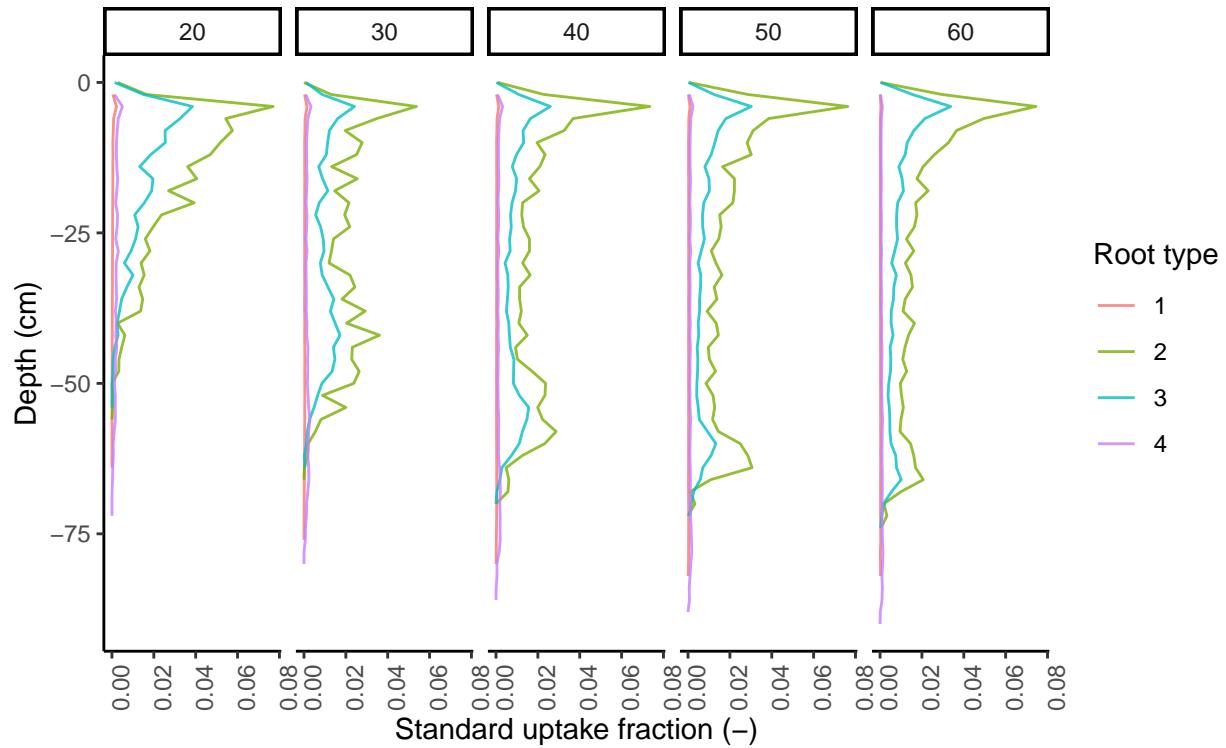


Figure 15: Evolution of the SUF with Leitner 2010 RSA parameter

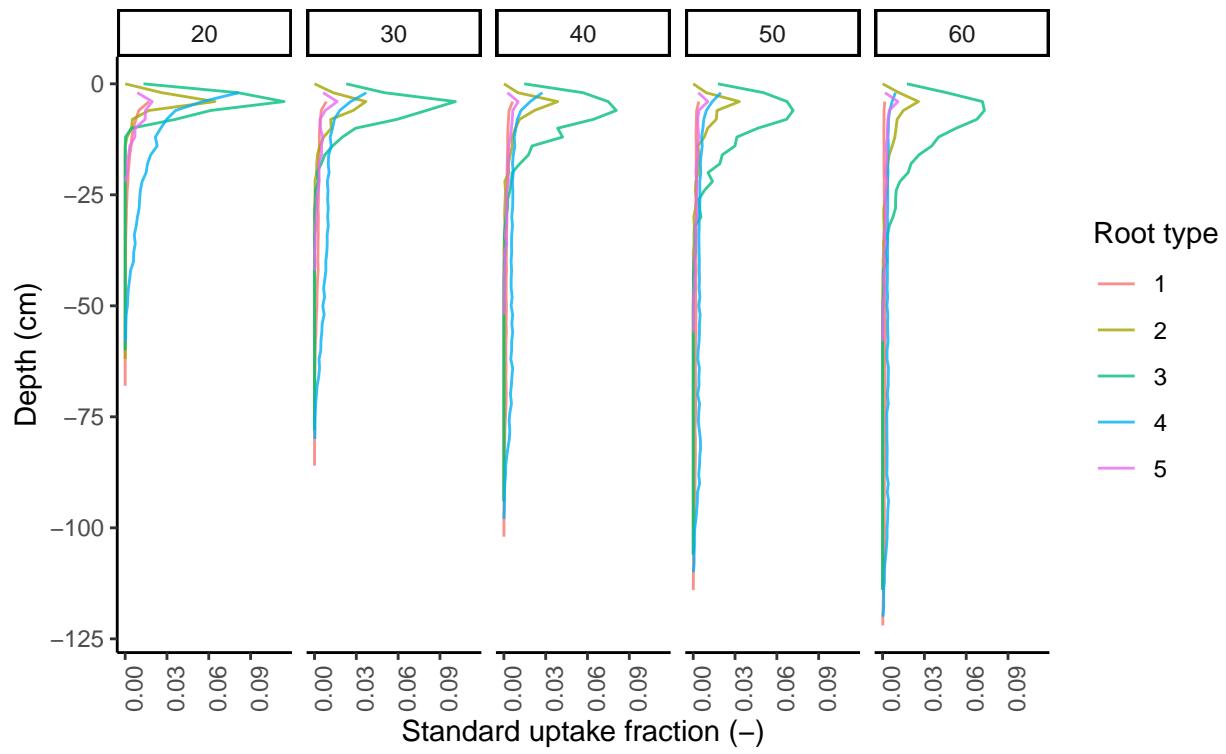


Figure 16: Evolution of the SUF with Postma 2011 RSA parameter

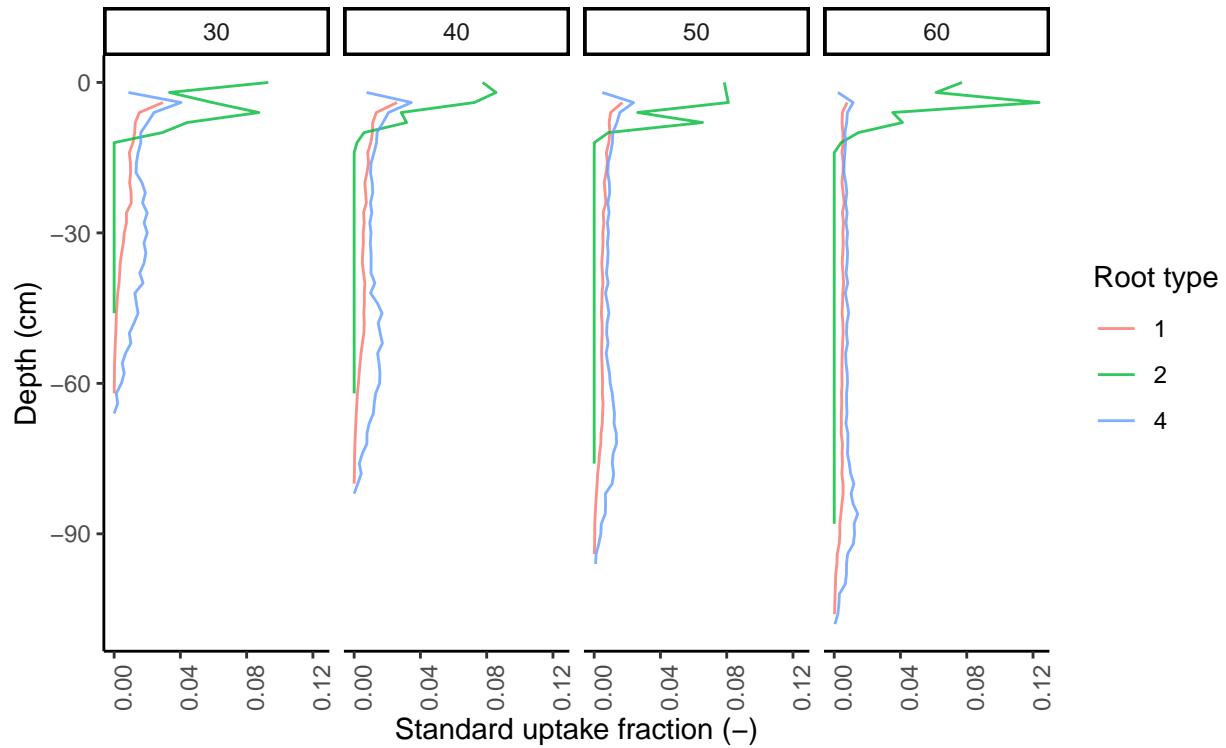


Figure 17: Evolution of the SUF with Leitner 2014 RSA parameter

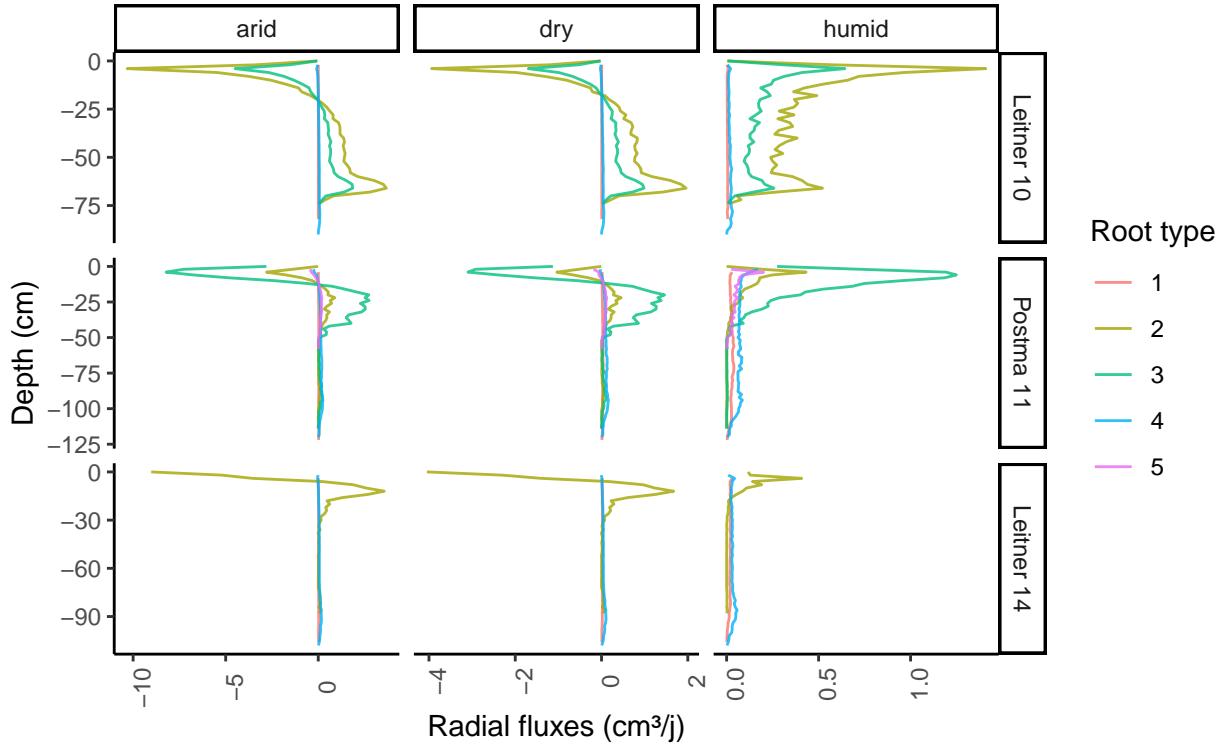


Figure 18: Comparison of the radial fluxes between the different soil scenarios and the root system architectures

```

ylab("Axial fluxes (cm³/j)")+
  labs(colour = "Root type")+
  facet_grid(ex~humid, scales = "free", labeller = as_labeller(babbel))

Transpiration<- results%>%
  mutate(humid = as.character(humid))%>%
  group_by(age, rep, scenario, humid, ex)%>%
  summarise(Tpot = max(tpot, tact),
            Tact = min(tpot, tact))

```

References

- Couvreur, V, J Vanderborght, and M Javaux. 2012. “A Simple Three-Dimensional Macroscopic Root Water Uptake Model Based on the Hydraulic Architecture Approach.” *Hydrol. Earth Syst. Sci.* 16 (8). Copernicus GmbH: 2957–71.
- Doussan, Claude, Gilles Vercambre, and Loïc Pagès. 1998. “Modelling of the Hydraulic Architecture of Root Systems: An Integrated Approach to Water Absorption–Distribution of Axial and Radial Conductances in Maize.” *Ann. Bot.* 81: 225–32.
- Feddes, RA, PJ Kowalik, and H Zaradny. 1978. “Water Uptake by Plant Roots.” *Simulation of Field Water Use and Crop Yield*. Wiley New York, 16–30.
- Landsberg, J J, and N D Fowkes. 1978. “Water Movement Through Plant Roots.” *Ann. Bot.* 42 (3). Oxford University Press: 493–508.
- Meunier, F, V Couvreur, X Draye, J Vanderborght, and M Javaux. 2017. “Towards Quantitative Root

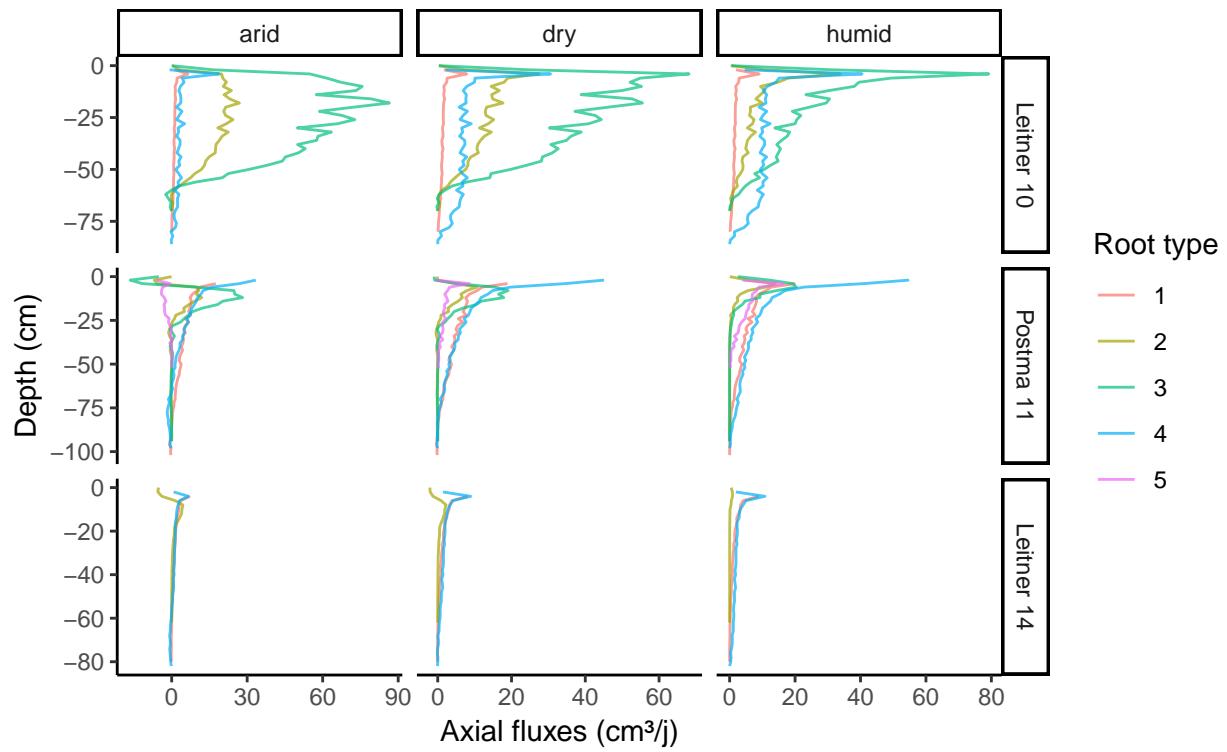


Figure 19: Comparison of the axial fluxes between the different soil scenarios and the root system architectures

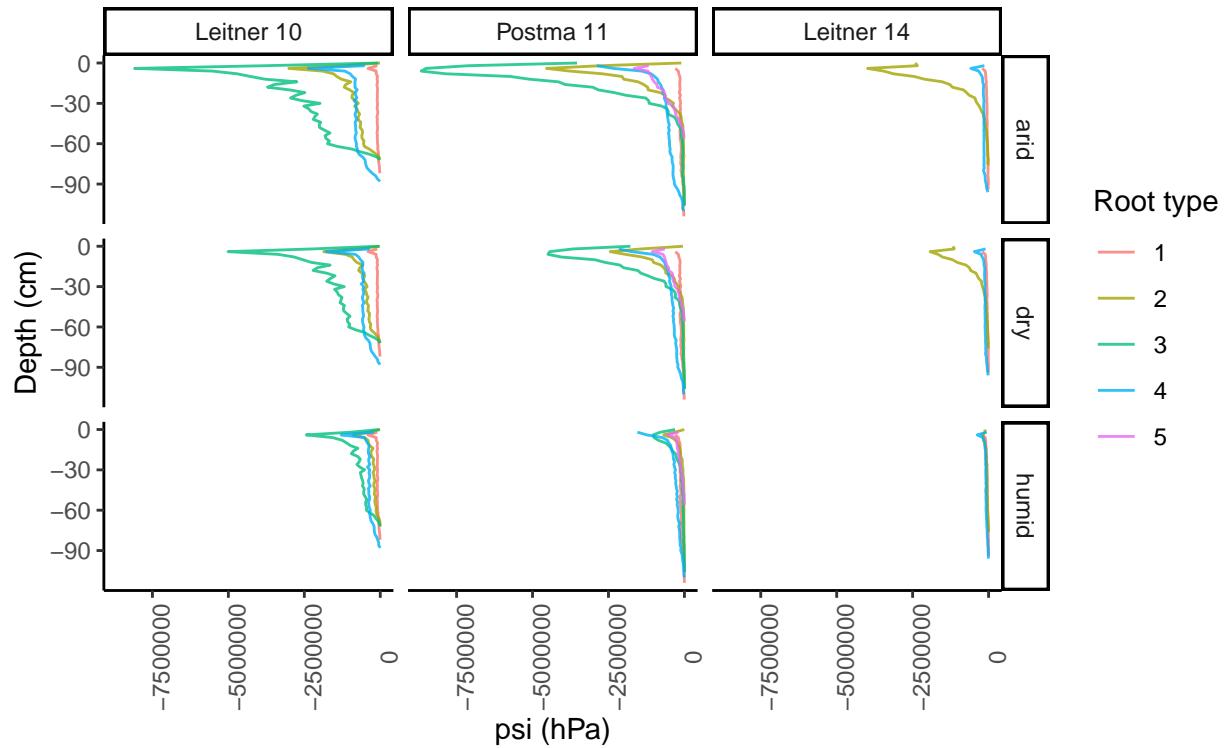


Figure 20: Water pressure head at the soil-root interface

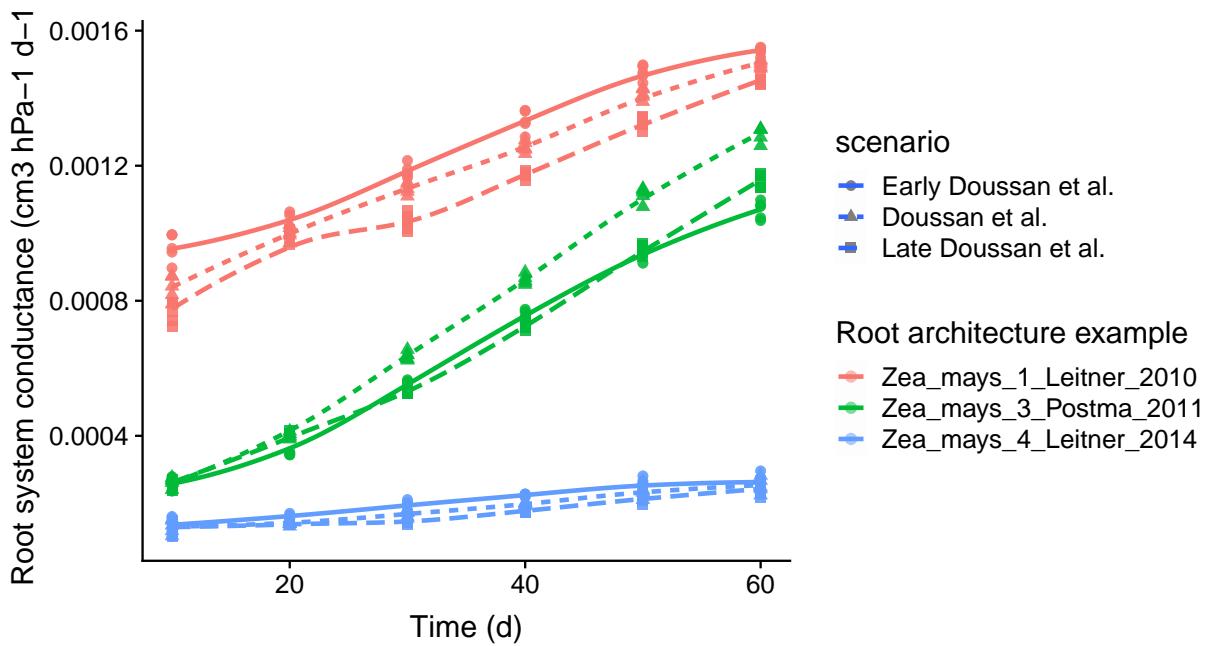


Figure 21: Evolution of the whole root system conductance. Effect comparison between root system architecture and root conductivity modification.

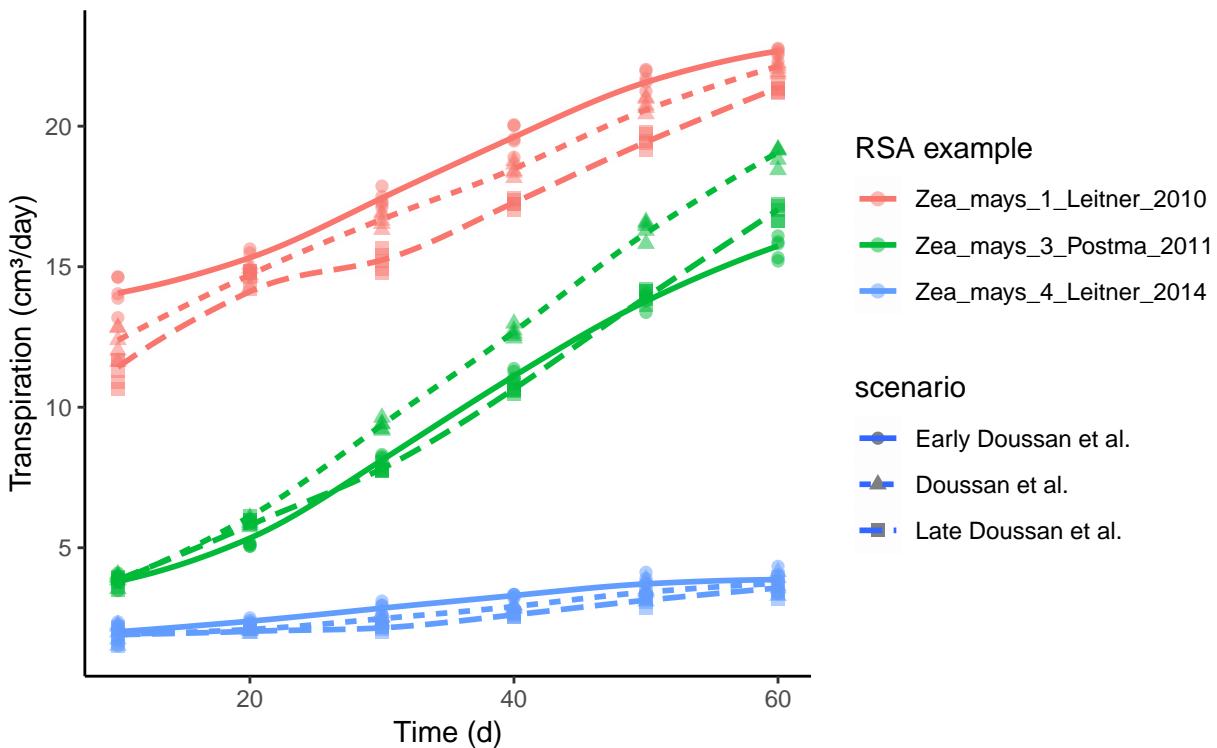


Figure 22: Evolution of the transpiration rate in a wet soil.

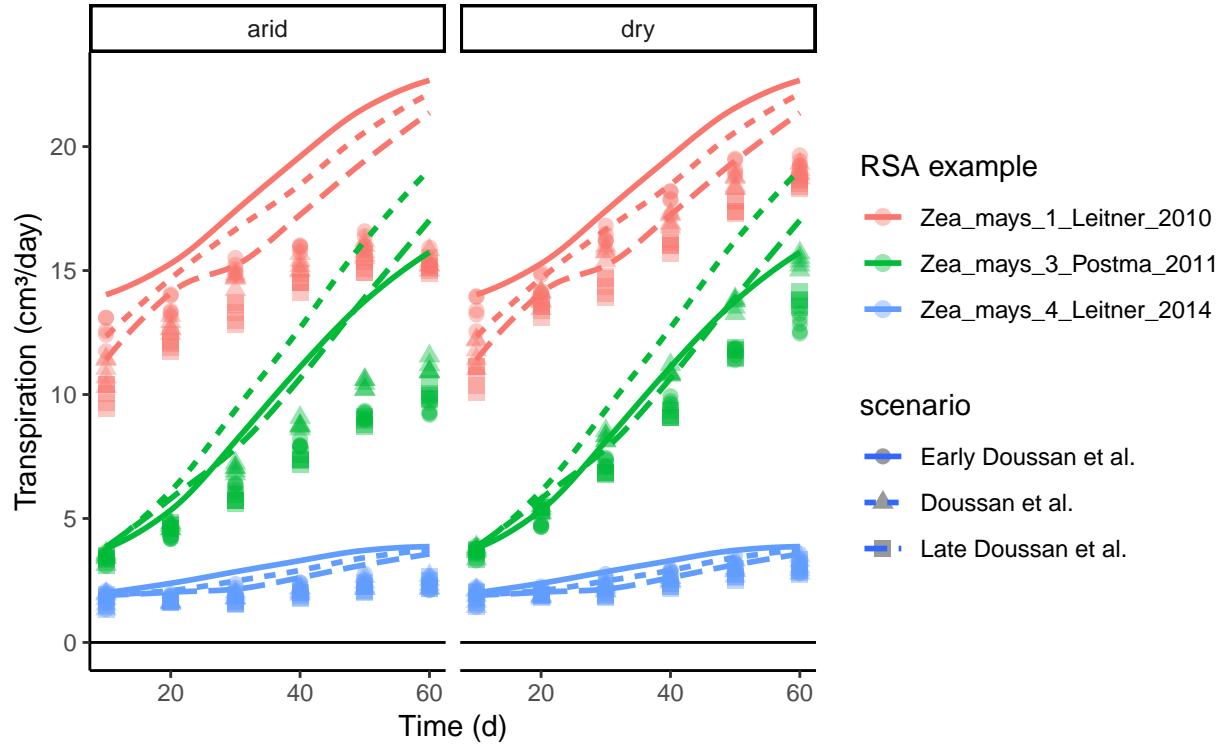


Figure 23: Evolution of the transpiration rate in two different soil scenarios where the soil is drying up.

Hydraulic Phenotyping: Novel Mathematical Functions to Calculate Plant-Scale Hydraulic Parameters from Root System Functional and Structural Traits.” *J. Math. Biol.* 75 (5): 1133–70.

Schnepf, Andrea, Daniel Leitner, Magdalena Landl, Guillaume Lobet, Trung Hieu Mai, Shehan Morandage, Cheng Sheng, Mirjam Zörner, Jan Vanderborght, and Harry Vereecken. 2018. “CRootBox: A Structural-functional Modelling Framework for Root Systems.” *Ann. Bot.*, February.