GRANAR user guide

Adrien Heymans

April 2019

What is GRANAR?

GRANAR stand for Generator of Root ANAtomy in R. The GRANAR model is able to generate complete cell networks of a root cross-sections using a small set of root anatomical features. The root anatomical features can typically be gathered by using software image analysis such as ImageJ (Schneider, Rasband, and Eliceiri 2012). Once the root anatomy is created, the cell network can be saved as an eXtended Markup Language (XML) file. The structure of the XML files created by GRANAR are identical to the ones created of CellSeT (Pound et al. 2012) (a software which digitize root anatomical network).

How to use GRANAR?

GRANAR is an R package. The *'granar'* package can be found on GitHub. Alternatively, by simply executing the following line in the R environment:

```
install.packages("devtools")
library(devtools)
install_github("granar/granar")
library(granar)
```

Granar was built upon some dependencies:

```
# Load the libraries
library(xml2)
library(tidyverse)
library(plyr)
library(deldir) # need to be install first
library(alphahull) # need to be install first
library(viridis)
library(sp)
library(cowplot)
library(retistruct) # need to be install first
library(Hmisc)
```

You do not need your own data to try it out, some examples of input parameters of GRANAR can be found in a Repository. The following table shows an example of all input variables inside one of the .XML file used to run GRANAR:

```
##          name           type  value
## 1   planttype          param  1.000
## 2  randomness          param  3.000
## 3       stele  cell_diameter  0.011
## 4       stele       n_layers  1.000
## 5       stele layer_diameter  0.210
## 6       stele          order  1.000
## 7   pericycle  cell_diameter  0.011
## 8   pericycle       n_layers  1.000
```

```
## 9   pericycle          order  2.000
## 10 endodermis  cell_diameter  0.025
## 11 endodermis       n_layers  1.000
## 12 endodermis          order  3.000
## 13     cortex  cell_diameter  0.030
## 14     cortex       n_layers  6.000
## 15     cortex          order  4.000
## 16  exodermis  cell_diameter  0.030
## 17  exodermis       n_layers  1.000
## 18  exodermis          order  5.000
## 19  epidermis  cell_diameter  0.010
## 20  epidermis       n_layers  1.000
## 21  epidermis          order  6.000
## 22      xylem       max_size  0.050
## 23      xylem        n_files  5.000
## 24      xylem          order  1.500
## 25      xylem          ratio  1.860
## 26     phloem       max_size  0.010
## 27     phloem        n_files  3.000
## 28 aerenchyma     proportion  0.200
## 29 aerenchyma        n_files 10.000
```

To import the .XML parameter file into R, it can be done with the function *read_param_xml()*. Example:

```
# Read the parameters
params <- read_param_xml("granar_examples/modelparam/Zea_mays_Heymans_2019.xml")
```

It is possible to change directly some of the parameters one-by-one in R.

```
# Copy and modify the parameters
params_no_aerenchyma <- params
params_no_aerenchyma$value[params$name == "aerenchyma" & params$type == "proportion"] <- 0
```

Once, the input file is loaded the function *create_anatomy()* can be called and it will generate the corresponding cross-section.

```
# Run the model with the loaded parameters
sim <- create_anatomy(parameters = params, verbatim=F)
```

```
## Time difference of 7.346239 secs
```

```
# Run the model with the modified parameters
sim_no_aer <- create_anatomy(parameters = params_no_aerenchyma, verbatim=F)
```

```
## Time difference of 1.095077 secs
```

To visualize the results a dedicated function, named *plot_anatomy()*, can be usefull. Different information can be illustrate such as the cell type (fig. 1) or the cell size of the cross section (fig. 2)

```
# Create figure of the generated anatomies.
plot_anatomy(sim, col = "type")
```

```
plot_anatomy(sim_no_aer, col = "area")
```

When the anatomy is completed, a saving procedure can be executed:

```
# Save the output to the MECHA folder
write_anatomy_xml(sim=sim,
        path="granar_examples/cellsetdata/current_root.xml")
```

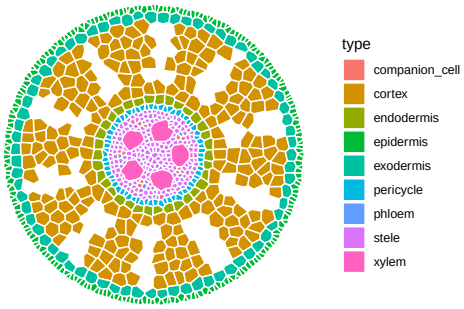

Figure 1: Maize cross-section anatomy made by GRANAR.

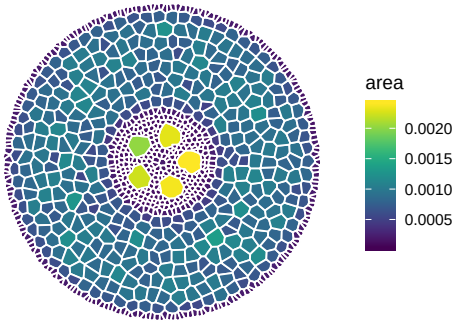

Figure 2: Maize cross-section anatomy made by GRANAR.

To load the nodes data of a cross-section, it can be achieved with *get_root_section()*. Below, we load a fully digitilized cross section done by CellSeT cross-section of a maize (Zea mays. B73 line) crown root grown aeroponically (fig. 3). It was collected 5 cm above the root tip. And the initial image cross section come from Hachez et al. (2006). The digitilization with CellSeT was performed to run MECHA (Couvreur et al. 2018) on it.

Next, to compare visually the root anatomical network obtain with CellSeT with the one of GRANAR, we generated the coresponding root cross section with GRANAR thanks to the cell data inside the CellSeT file.

```
# It is possible to load cross-section xml under cellset format
Cross_section <- list()
Cross_section$nodes <- get_root_section(
  path="granar_examples/cellsetdata/Maize_pip_cross8_phloem.xml")

# Some cell type had to be replace to obtain a similar results as what GRANAR
node <- Cross_section$nodes%>%
  mutate(area = NA,
```

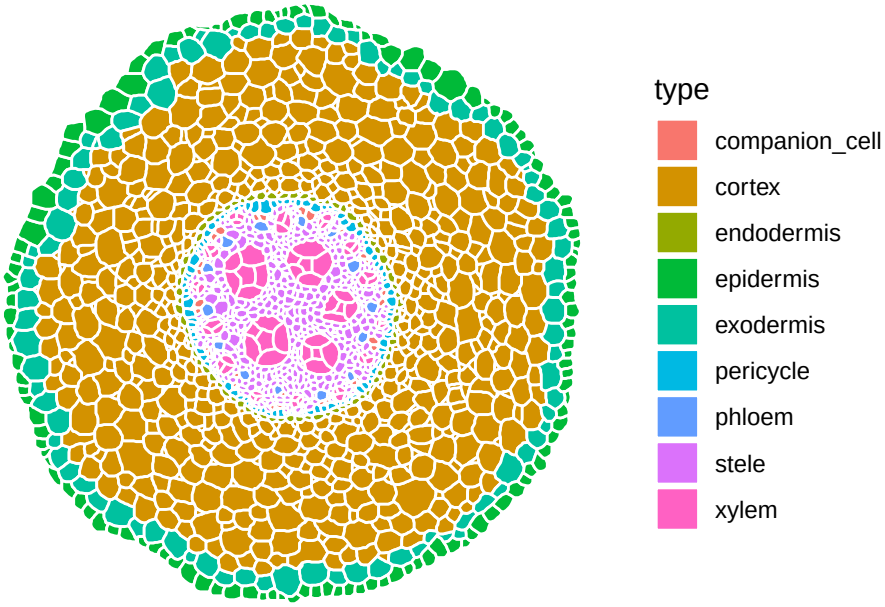

Figure 3: Cross-section anatomy made by CellSeT. Maize (Zea mays. B73 line) crown root grown aeroponically. It was collected 5 cm above the root tip. And the initial image cross section come from Hachez et al. (2006). The digitilization with CellSeT was performed to run MECHA (Couvreur et al. 2018) on it.

```
        name = "GRANAR")
node$type <- node$type%>%
  as.character()%>%
  replace(.,. == "general" ,"exodermis")%>%
  replace(.,. == "columella3" , "xylem")%>%
  replace(.,. == "columella1" , "phloem")%>%
  replace(.,. == "columella2" , "companion_cell")

Cross_section$nodes <- node
plot_anatomy(Cross_section, col = "type")
```

```
# Procedure to aquire cell area.
for (i in unique(node$id_cell)) {
 tmp <- node[node$id_cell == i,]
 pol <- Polygon(tmp[, c("x","y")])
 node$area[node$id_cell == i] <-  pol@area
}
```

```r
# Get information per cell type.
condens <- node%>%
  dplyr::group_by(type)%>%
  dplyr::summarise(area_type = sum(area),
                   cell_type = mean(area),
                   cell_size = 2*sqrt(cell_type/pi),
                   n = n(),
                   r_type = (max(x)-min(x)+max(y)-min(y))/4)%>%
  dplyr::ungroup()
```

```r
# Replace input parameter of GRANAR with the cell data gathered from CellSeT.
# Cell data are turned from µm to mm.
params$value[params$name == "aerenchyma" &
             params$type == "proportion"] <- 0
params$value[params$name == "randomness" &
             params$type == "param"] <- 4
params$value[params$name == "stele" &
             params$type == "cell_diameter"] <- condens$cell_size[
               condens$type == "stele"]/1000
params$value[params$name == "stele" &
             params$type == "layer_diameter"] <- condens$r_type[
               condens$type == "stele"]*2/1000
params$value[params$name == "pericycle" &
             params$type == "cell_diameter"] <- condens$cell_size[
               condens$type == "pericycle"]/1000
params$value[params$name == "endodermis" &
             params$type == "cell_diameter"] <- condens$cell_size[
               condens$type == "endodermis"]/1000
params$value[params$name == "cortex" &
             params$type == "cell_diameter"] <- condens$cell_size[
               condens$type == "cortex"]*0.9/1000
params$value[params$name == "cortex" &
             params$type == "n_layers"] <- round((condens$r_type[
               condens$type == "cortex"] - condens$r_type[
               condens$type == "endodermis"])/condens$cell_size[
               condens$type == "cortex"])
params$value[params$name == "exodermis" &
             params$type == "cell_diameter"] <- condens$cell_size[
               condens$type == "exodermis"]/1000
params$value[params$name == "epidermis" &
             params$type == "cell_diameter"] <- condens$cell_size[
               condens$type == "epidermis"]/1000
params$value[params$name == "xylem" &
             params$type == "max_size"] <- condens$r_type[
               condens$type == "stele"]*0.425/1000
params$value[params$name == "xylem" &
             params$type == "ratio"] <- 3

# Run GRANAR
sim <- create_anatomy(parameters = params)
```

```
## Time difference of 1.609518 secs
```

```r
# Make a figure of the root anatomical network created with GRANAR.
plot_anatomy(sim)
```

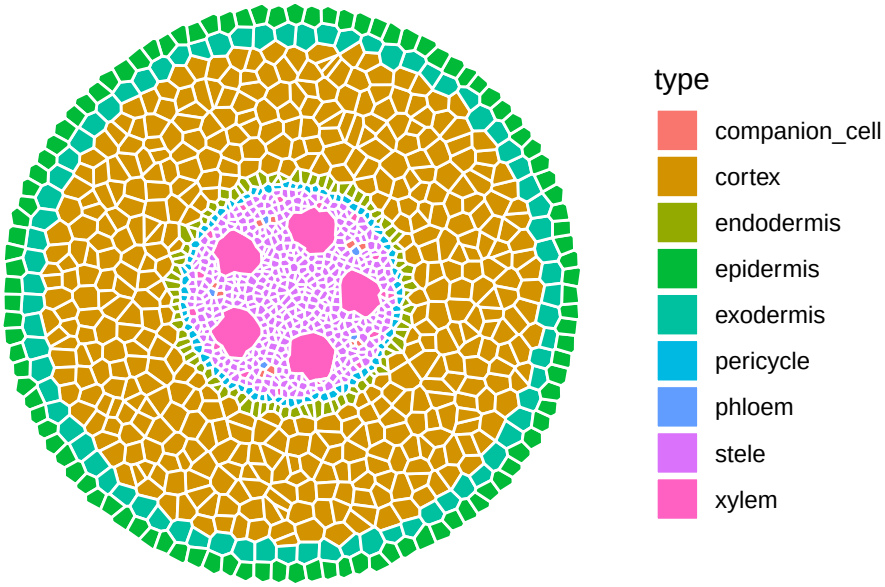

Figure 4: Maize cross-section anatomy made by GRANAR.

```
# Save output data into a .XML file.
write_anatomy_xml(sim=sim,
        path="granar_examples/cellsetdata/current_root.xml")
```

```
##   id_group           type
## 1        1      exodermis
## 2        2      epidermis
## 3        3     endodermis
## 4        4         cortex
## 5        5          stele
## 6       13          xylem
## 7       16      pericycle
## 8       12 companion_cell
## 9       11         phloem
```

```
## [1] TRUE
```

Couling GRANAR with MECHA

MECHA

MECHA is an explicit cross-section model of the root hydraulic anatomy which connects hydraulic concepts across scales.

The model computes horizontal water flow at the level of individual cells, quantifies the contribution of water composite pathways, and predicts root radial permeability (kr), using detailed anatomical descriptions and experimental data on the permeability of cell walls (kw), membranes (Lp) and the conductance of individual plasmodesmata (KPD).

Install MECHA

Here, we installed Canopy to run MECHA which was coded in python. Canopy offer the possibilty to install easilly the dependencies of MECHA:

- numpy
- scipy
- networkx (v.1.9.1)
- lxml

How to calculate hydraulic properties on newly aquire root cross section

```
# Working with Python in R
library(reticulate)
# The users are invite to select the pathway to Python
use_python("../../AppData/Local/Enthought/Canopy/edm/envs/User")
os <- import("os")
# If os produce error: verify your python directory
# use Canopy package manager to install
# numpy, scipy, networkx, lxml, matplotlib and pyqt
# For windows user, pyqt allows matplotlib to run without error

numpy <- import("numpy")
scipy <- import("scipy")
networkx <- import("networkx")
lxml <- import("lxml")
matplotlib <- import("matplotlib")

# The following function call MECHA and return the radial conductivity
source("granar_examples/MECHA_R.R")
Kr_Granar <- MECHA_R(path = "granar_examples/MECHAv4_light.py")
```

```
## Time difference of 51.85028 secs
```

```
fc <- file.copy(from = "granar_examples/cellsetdata/Maize_pip_cross8_phloem.xml",
                to = paste0("granar_examples/cellsetdata/current_root.xml"),
                overwrite = T)

Kr_CellSet <- MECHA_R(path = "granar_examples/MECHAv4_light_1.py")
```

```
## Time difference of 1.293418 mins
```

```
Kr <- rbind(tibble(id = c("granar"), kr0 = Kr_Granar[1],
                   kr1 = Kr_Granar[2], kr2 = Kr_Granar[3]),
```

```
            tibble(id = c("cellset"), kr0 = Kr_CellSet[1],
                   kr1 = Kr_CellSet[2], kr2 = Kr_CellSet[3]))
```

```
## # A tibble: 2 x 4
##   id           kr0       kr1       kr2
##   <chr>      <dbl>     <dbl>     <dbl>
## 1 granar  0.000110 0.0000421 0.0000389
## 2 cellset 0.000105 0.0000398 0.0000370
```

- The *Kr0*: The Radial conductivity with only a endodermal casparian strip (cm hPa-1 d-1).
- The *Kr1*: The Radial conductivity with suberized endodermis (cm hPa-1 d-1).
- The *Kr2*: The Radial conductivity with suberized endodermis and exodermal Casparian strip (cm hPa-1 d-1).

References

Couvreur, Valentin, Marc Faget, Guillaume Lobet, Mathieu Javaux, François Chaumont, and Xavier Draye. 2018. "Going with the Flow: Multiscale Insights into the Composite Nature of Water Transport in Roots." *Plant Physiol.* 178 (4). Am Soc Plant Biol: 1689–1703.

Hachez, Charles, Menachem Moshelion, Enric Zelazny, Damien Cavez, and François Chaumont. 2006. "Localization and Quantification of Plasma Membrane Aquaporin Expression in Maize Primary Root: A Clue to Understanding Their Role as Cellular Plumbers." *Plant Molecular Biology.*

Pound, Michael P, Andrew P French, Darren M Wells, Malcolm J Bennett, and Tony P Pridmore. 2012. "CellSeT: Novel Software to Extract and Analyze Structured Networks of Plant Cells from Confocal Images." *Plant Cell* 24 (4): 1353–61.

Schneider, Caroline A, Wayne S Rasband, and Kevin W Eliceiri. 2012. "NIH Image to ImageJ: 25 Years of Image Analysis." *Nat. Methods* 9 (7). nature.com: 671–75.