

RRootDetection - Provides simple visualization and parsing functions for RootDetection's database data

Kristian Ullrich and Ole Trenner

May 23, 2012

1 Introduction

The R-package *RRootDetection* is a collection of of basic visualization and parsing functions for **RootDetection**'s database data. Its primary purpose is the data access of root system architecture (RSA) traits measured with **RootDetection** <http://www.labutils.de> and the simple visualization of this data in barplots, boxplots or histograms. The **RootDetection** data is accessed via the R-package *RSQLite* and can then easily statistically analysed with the R core functions or other R-packages.

2 Short Background

3 Data connection with *RRootDetection*

3.1 Prdefined list for storing connection details

RD.state: This predefined list is used to store mandatory connection details, which can be set by the **RD.load()** function. The first list value stores the path of the database file. The second list value is the 'SQLite Connection' class used to connect to the database file.

```
#####  
#Short example  
#  
library(RRootDetection)  
file <- system.file("extdata/example.db",package = "RRootDetection")
```

```
#Setting Connection to database file
RD.load(file)

#Show connection details
RD.state
```

For related topics see also:

```
%current function
help("RD.state")
%additional functions set by \seealso
help("RD.load")
help("dbConnect")
```

3.2 Connection to a RootDetection database

`RD.load`: Establish a connection to a database file generated with RootDetection. Saves the connection parameters in the `RD.state` variable. The path needs to include the database file.

```
#####
#Short example
#
library(RRootDetection)
file <- system.file("extdata/example.db",package = "RRootDetection")

#Setting Connection to database file
RD.load(file)

#To print connection details
RD.state
```

For related topics see also:

```
%current function
help("RD.list.directories")
%additional functions set by \seealso
help("RD.list.images")
help("RD.list.roots")
```

After a connection to a **RootDetection** database is set one can continue to browse the included RSA trait data.

4 Data access with *RRootDetection*

4.1 Execute an SQL statement on the connected RootDetection database

`RD.select`: Uses the `dbGetQuery` function [package:DBI] to submit and execute an SQL statement on the connected RootDetection database. The query and the connection from `RD.state[[2]]` is passed to the `dbGetQuery` function [package:DBI]. For details please see `dbGetQuery` function [package:DBI].

```
#####  
#Short example  
#  
library(RRootDetection)  
file <- system.file("extdata/example.db",package = "RRootDetection")  
  
#Setting Connection to database file  
RD.load(file)  
  
#To print connection details  
RD.state  
  
tmp_query = "select * from images"  
tmp_connection = RD.state[[2]]  
  
#To perform query as SQL statement on the RootDetection database  
#  
#Equals: dbGetQuery(tmp_connection,tmp_query)[1,]  
#  
RD.select(tmp_query)[1,]
```

For related topics see also:

```
%current function  
help("RD.select")  
%additional functions set by \seealso  
help("dbGetQuery")  
help("dbConnect")  
help("RD.load")  
help("RD.list.directories")  
help("RD.list.images")  
help("RD.list.roots")  
help("RD.list.label")
```

4.2 Extract path of directories stored in a RootDetection database file

`RD.list.directories`: Returns the path of directories (folders) which were analyzed with RootDetection and stored in a RootDetection database file. Uses SQLite select: "select distinct dirname from images" to extract directories from RootDetection database file.

```
#####  
#Short example  
#  
library(RRootDetection)  
file <- system.file("extdata/example.db", package = "RRootDetection")  
#Setting Connection to database file  
RD.load(file)  
  
#Extract directories from database file  
#Equals: RD.select("select distinct dirname from images")  
RD.list.directories()  
  
#Select one directory for further analysis  
RD.list.directories()[1]
```

For related topics see also:

```
%current function  
help("RD.load")  
%additional functions set by \seealso  
help("dbConnect")
```

4.3

4.4

4.5

5 Data parsing with *RRootDetection*

5.1

5.2

6 Data visualization with *RRootDetection*

6.1 Barplot of mean root length values

`RD.plot.bar`: This function produces barplots from root length values from a matrix generated with the `RD.list.roots()` function as input. It plots the mean root length values for the groups in that matrix plus the standard error of the mean or the standard deviation (`'error.bars'`). The mean values can be sorted to get an overview of the maximal and minimal mean root length values (`'sort'`). All the data used for plotting the barplots can also be returned as a summary (`'return.summary'`). Optional the root length values can be processed with the `remove.sd()` function to remove outliers within certain thresholds (`'remove.outliers'`). If roots were measured with *RootDetection* on one picture in different image regions they can be combined so that all roots on one image are handled as one group (`'by.image'`). It is also possible to combine roots which share the same label definition (`'by.label'`).

```
> #####
> #Short example
> #
> library(RRootDetection)
> file <- system.file("extdata/example.db", package="RRootDetection")
> #Setting Connection to database file
> RD.load(file)
> #Select one directory for further analysis
> tmp_dir <- RD.list.directories()[1]
> #Extract root data from the specified directory using default settings for
> #
> #For the RD.list.roots() options please see 'help("RD.list.roots")'
> #tmp_root_directory <- RD.list.roots(directory=tmp_dir, image = NULL, num =
```

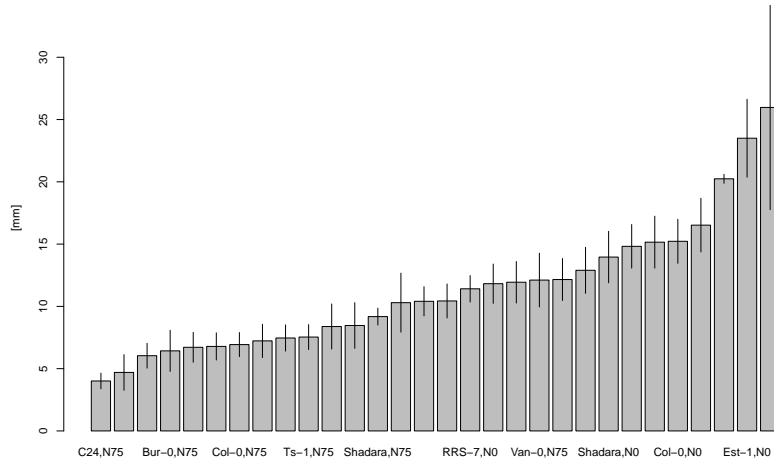


Figure 1: BARPLOT1.

```
> #
> tmp_root_directory <- RD.list.roots(directory=tmp_dir)
> #Plot a barplot from the tmp_root_directory matrix using default settings 1
> RD.plot.bar(tmp_root_directory)

#####
#Short example
#
library(RRootDetection)
file <- system.file("extdata/example.db",package="RRootDetection")
#Setting Connection to database file
RD.load(file)

#Select one directory for further analysis
tmp_dir <- RD.list.directories()[1]

#Extract root data from the specified directory using default settings for the
#
#For the RD.list.roots() options please see 'help("RD.list.roots")'
#tmp_root_directory <- RD.list.roots(directory=tmp_dir, image = NULL, num = NU
#
tmp_root_directory <- RD.list.roots(directory=tmp_dir)
```

```

#Plot a barplot from the tmp_root_directory matrix using default settings for
RD.plot.bar(tmp_root_directory)

#####
#Plot a barplot from the tmp_root_directory matrix using custom settings for t
#
#RD.plot.bar(tmp_root_directory, scale = TRUE, by.image = FALSE, by.label = FA

#plot pixel lengths (scale = FALSE)
RD.plot.bar(tmp_root_directory, scale = FALSE)

###
#standard deviation (error.bars = "sd")
RD.plot.bar(tmp_root_directory, error.bars = "sd")

###
#group by label (by.label = TRUE)
RD.plot.bar(tmp_root_directory, by.label = TRUE)

###
#plot the data unsorted (sort = FALSE)
RD.plot.bar(tmp_root_directory, sort = FALSE)

###
#return the mean root length data as a summary (return.summary = TRUE)
RD.plot.bar(tmp_root_directory, return.summary = TRUE)

###
#remove outliers from the data (remove.outliers = TRUE) using the standard set
RD.plot.bar(tmp_root_directory, remove.outliers = TRUE)

###
#remove outliers from the data with custom settings (remove.outliers = TRUE, s
RD.plot.bar(tmp_root_directory, remove.outliers = TRUE, sd.factor = 3, thresho

###
#define other plot functions like the plot title
RD.plot.bar(tmp_root_directory, main = "plot title")

#####

```

```

#Optional one can use first the 'return.summary = TRUE' option and then plot t.
#
tmp_root_directory.summary <- RD.plot.bar(tmp_root_directory, return.summary =
barplot(unlist(tmp_root_directory.summary[, "mean"])))
barplot.pos <- barplot(unlist(tmp_root_directory.summary[, 1]), plot = FALSE)
barplot.mean <- unlist(tmp_root_directory.summary[, "mean"])
barplot.sd <- unlist(tmp_root_directory.summary[, "sd"])
barplot.n <- unlist(tmp_root_directory.summary[, "n"])
barplot.sem <- barplot.sd/sqrt(barplot.n)
arrows(barplot.pos, barplot.mean-barplot.sd, barplot.pos, barplot.mean+barplot.sd)
arrows(barplot.pos, barplot.mean-barplot.sem, barplot.pos, barplot.mean+barplot.s

#####
#To output barplots as jpeg, png, pdf or postscript please use the standard R
#
output_file <- "tmp.pdf"
pdf(file = output_file)
RD.plot.bar(tmp_root_directory)
dev.off()

```

For related topics see also:

```

%current function
help("RD.plot.bar")
%additional functions set by \seealso
help("RD.list.roots")
help("extract.len")
help("remove.sd")
help("barplot")
help("png")
help("jpeg")
help("pdf")
help("postscript")

```

References