medplot : Installation and usage

Contents

Package `medplot` installation instructions

We will describe a typical installation of the *medplot* package.

Prerequisites

- MS Excel
- R statistical environment (installed either locally or on a server)(http://cran.r-project.org/)
- Perl language installed (http://www.perl.org/)
- a working internet connection

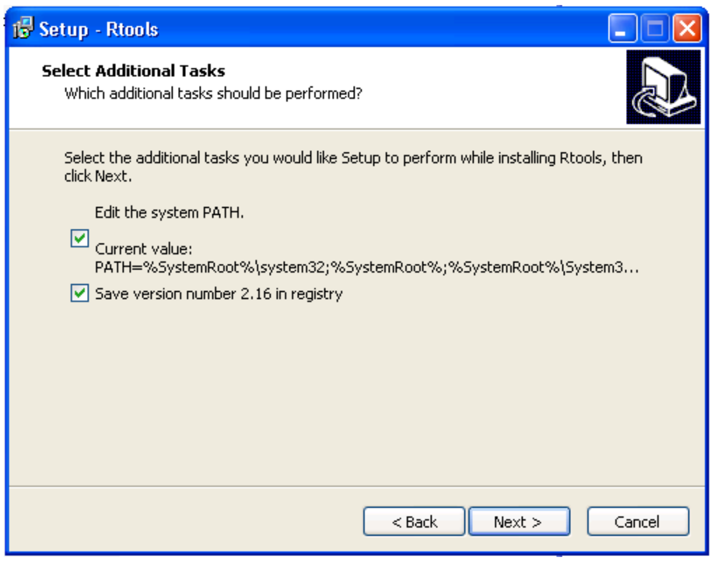
Installation of `medplot` package

You have to install the medplot package on the computer that will run R and the web server. If you intend to run R and the web server on a separate computer, refer to shiny-server installation instructions on how to proceed (https://github.com/rstudio/shiny-server). If you intend to run them on the user's computer, you need R (refer to http://cran.r-project.org/) and the Perl language (refer to http://www.perl.org/) installed on it. We will assume you are installing on the user's computer and that you have the mentioned prerequisites.

You will install the medplot package via the R console. For the moment, only installation via GitHub is supported (CRAN installations are planned). Open your R console and proceed.

1. run:
   ```
   install.packages("devtools")
   ```
2. You will need to download and install the Rtools package (not done via the R console). Use your web browser to navigate to the address:
   http://cran.r-project.org/bin/windows/Rtools/

1. Download the latest available version of Rtools package (at time of writing `Rtools216.exe`). Remember where you saved it and then run it, to install Rtools. You can stick to the defaults, but not when asked about additional tasks - select both options when prompted:



Setup - Rtools

Select Additional Tasks
Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Rtools, then click Next.

Edit the system PATH.
☑ Current value:
PATH=%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System3...
☑ Save version number 2.16 in registry

   The option to edit the system PATH should be checked.
2. Restart windows after installing Rtools (because the system PATH needs to be reloaded at boot time).

3. After restarting, open your R console again. From your R console, load devtools into library:
   `library(devtools)`
4. Install the medplot package from GitHub:
   `install_github("medplot", username="crtahlin")`
   Note: At the time of writing the SVGAnnotation package did not have an appropriate binary version on CRAN. This might be the reason that the installation of medplot package fails at this point. For this package to install, run the installation from source:
   `install.packages("SVGAnnotation", repos="http://www.omegahat.org/R", type="source")`
   Try running this line first and then repeating step 4.
5. Load medplot package into library:
   `library(medplot)`
   This command will return something like *"... there is no package called 'medplot'"* if the installation of medplot failed for some reason. In that case you need to repeat medplot installation steps.
6. Look into the library (folder) on your computer, where the medplot package is installed (e.g.: `C:\Program Files\R\R-2.15.2\library\medplot`). You can retrieve the path of the folder by executing the command:
   `path.package("medplot")`
   In its subfolder called `exdata`, you should find the Excel files which you will use in your work:
   `PlotTests_shiny.xlsm` - plotting of test results
   `PlotSymptoms_shiny.xlsm` - plotting of symptoms found

Troubleshooting

To check whether and where your medplot package is installed, you can run the library() command from your R console, which will open a window with a list of library locations and names:
`library()`
These packages should be installed:

Cairo,

gplots,

RColorBrewer,

XML,

SVGAnnotation,

shiny,

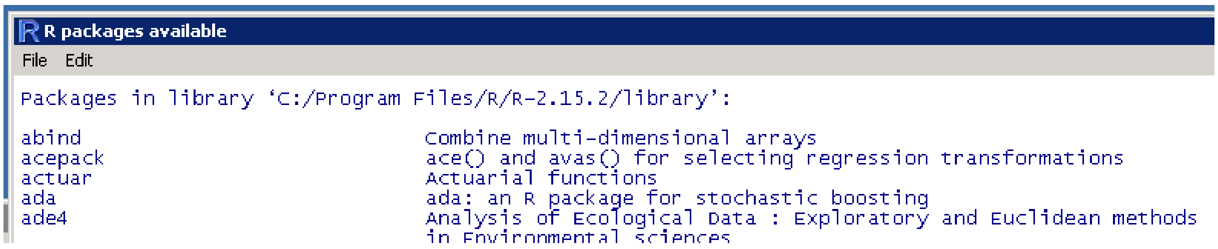
scales,

reshape2,

ggplot2,

seriation,

medplot (This is our package; the others are needed for it to function. See example screenshot below, showing some packages available in a library.)



```
Packages in library ‘C:/Program Files/R/R-2.15.2/library’:

abind                               Combine multi-dimensional arrays
acepack                             ace() and avas() for selecting regression transformations
actuar                              Actuarial functions
ada                                 ada: an R package for stochastic boosting
ade4                                Analysis of Ecological Data : Exploratory and Euclidean methods
                                    in Environmental sciences
```

If it turns out some package is missing, try to install it manually by running:

```
install.packages("NAME_OF_PACKAGE")
```

For copy-paste ease of use, these are listed below:

```
install.packages("Cairo")
install.packages("RColorBrewer")
install.packages("shiny")
install.packages("scales")
install.packages("reshape2")
install.packages("gplots")
install.packages("ggplot2")
install.packages("seriation")
install.packages("XML")
```

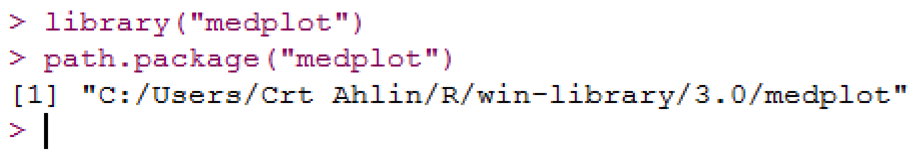
Package `medplot` usage instructions

The medplot package contains several MS Excel (http://office.microsoft.com/en-us/excel/) files which you can fill with your own data (we will call them template files). You can generate graphs from this data by uploading the files to a web server run by the shiny R package (RStudio & Inc., 2013). The package interfaces with R statistical environment (R Core Team, 2013) to generate the graphs in a web browser.

We will assume you have successfully installed the *medplot* package for the R language and all the other required packages (as described in previous chapters).

You can find the Excel template files in the `exdata` subfolder of the medplot package folder. To see where your medplot package is installed issue the following commands at the R prompt:

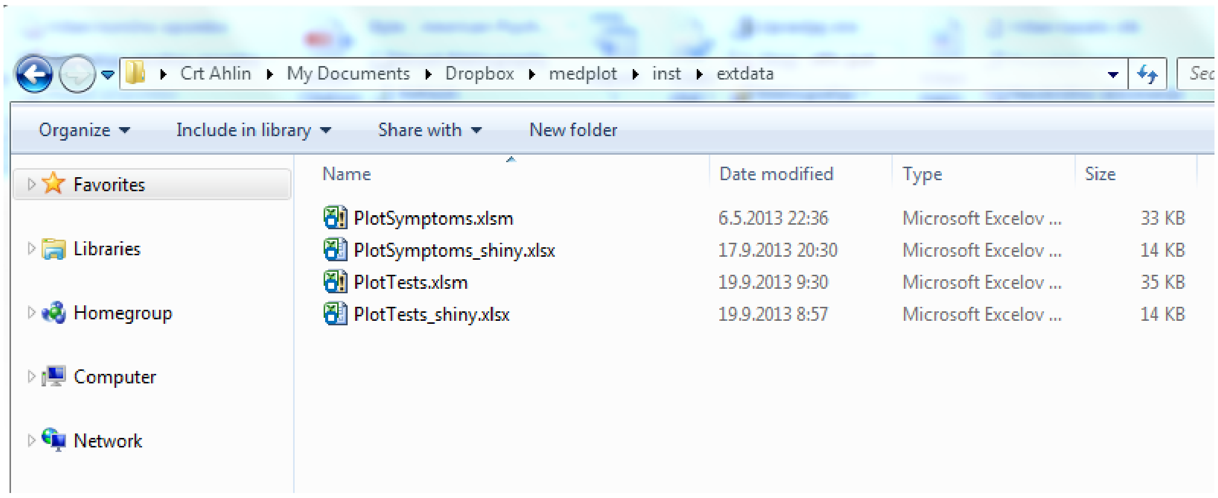
```
> library("medplot")
> path.package("medplot")
```



```
> library("medplot")
> path.package("medplot")
[1] "C:/Users/Crt Ahlin/R/win-library/3.0/medplot"
>
```

E.g.: the screenshot reveals the medplot package is inside the folder C:/Users/Crt Ahlin/R/win-library/3.0/medplot.

Navigate to the folder where medplot is installed, and enter the `exdata` subfolder.



Name	Date modified	Type	Size
PlotSymptoms.xlsm	6.5.2013 22:36	Microsoft Excelov ...	33 KB
PlotSymptoms_shiny.xlsx	17.9.2013 20:30	Microsoft Excelov ...	14 KB
PlotTests.xlsm	19.9.2013 9:30	Microsoft Excelov ...	35 KB
PlotTests_shiny.xlsx	19.9.2013 8:57	Microsoft Excelov ...	14 KB

E.g.: the contents of the exdata subfolder.

The appropriate Excel template files have the "_shiny.xslx" suffix. Copy these files to your favorite working folder. You can also rename them, if you wish. **Always work on these copies to avoid losing data when updating the medplot package. The versions in the exdata folder get overwritten at each medplot package update.**

General usage instructions

Protected cells and sheets

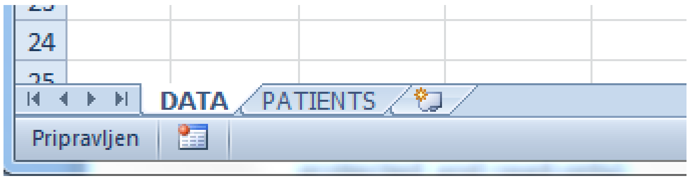
The sheets and cells in the Excel template files are protected in a sense, that the user cannot edit cells that he is not supposed to edit. The protection can be removed by users (no password is used), but this is discouraged as more mistakes are possible without protection enabled.

PlotSymptoms_shiny.xlsm spreadsheet usage instructions

PlotSymptoms_shiny.xlsm relies on the R `shiny` package (RStudio & Inc., 2013), which runs behind a web server and renders a plot the user can manipulate via a web browser.

The PlotSymptoms.xlsm contains the following sheets:

- DATA
- PATIENTS

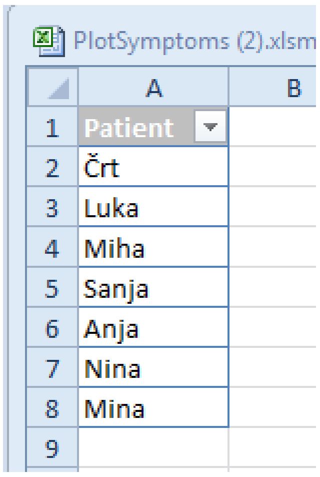


E.g.: Sheets in PlotSymptoms.xlsm.

The sheets will be described in the order the user should refer to them in the following subparagraphs.

"PATIENTS" sheet

The PATIENTS sheet contains names of the patients. For data integrity purposes only patients listed on this sheet can be entered on the DATA sheet.



PlotSymptoms (2).xlsm

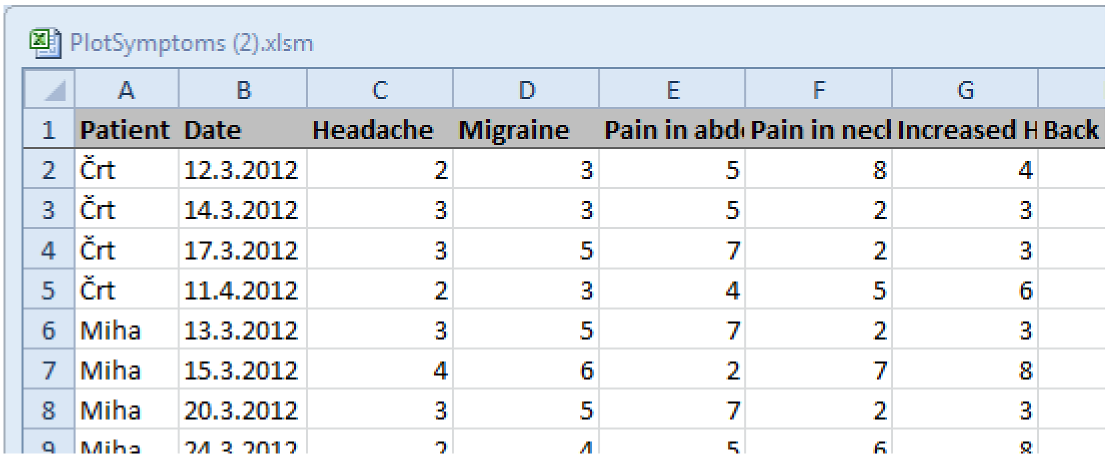
	A	B
1	Patient	
2	Črt	
3	Luka	
4	Miha	
5	Sanja	
6	Anja	
7	Nina	
8	Mina	
9		

E.g.: Sample patients on the PATIENTS sheet.

"DATA" sheet

The DATA sheet contains the patients, the dates of checkups and the severity of reported symptoms. Each line of data represents one date of one patient checkup.

The user can edit the column names after the Date column, which represents the symptom names. Names in the Patient column should be entered as they are listed on the PATIENTS sheet. Names not listed on this sheet are not valid. Dates in the Date column should be entered in DD.MM.YYYY format. Severity of symptoms should be entered as an integer value between and including 0 and 10.



PlotSymptoms (2).xlsm

	A	B	C	D	E	F	G
1	Patient	Date	Headache	Migraine	Pain in abd	Pain in necl	Increased H	Back
2	Črt	12.3.2012	2	3	5	8	4
3	Črt	14.3.2012	3	3	5	2	3
4	Črt	17.3.2012	3	5	7	2	3
5	Črt	11.4.2012	2	3	4	5	6
6	Miha	13.3.2012	3	5	7	2	3
7	Miha	15.3.2012	4	6	2	7	8
8	Miha	20.3.2012	3	5	7	2	3
9	Miha	24.3.2012	2	4	5	6	8

E.g.: Sample data on the DATA sheet.

Running the plotting function

An R shiny app should be started either on a local server or on a remote server. The user then communicates with the app through a web browser, pointing to the address of the server. To start the server locally, open an R console and issue the commands:

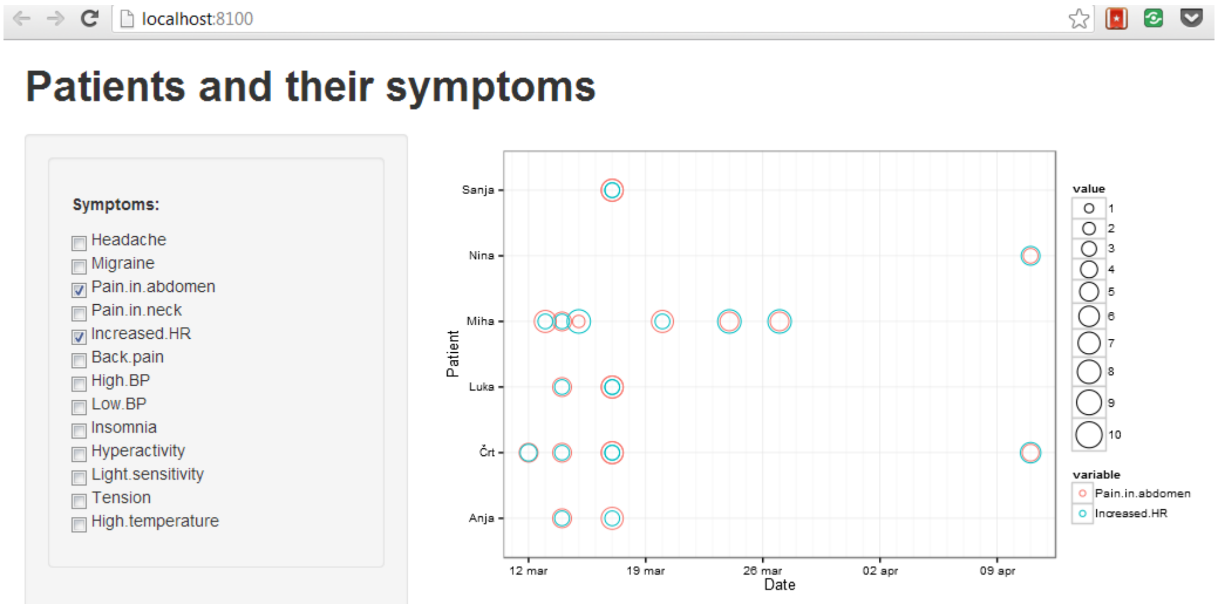
```
library("medplot")
library("shiny")
runApp(appDir=file.path(path.package("medplot"),"shinyapp_symptoms2"
), launch.browser=TRUE)
```

These commands should load the required packages, start the shiny application and open the user's browser at the appropriate address (e.g. localhost:8100).

Note that the firewall on the user's computer has to allow connections to the server in order for the web browser to be able to connect to it. If a prompt appears after issuing the above command asking to allow connections, you should confirm it.

A web browser window should open with options to select which symptoms the user wishes to plot. The plot updates every time after the user changes the selection.

The plot



E.g.: Example of the generated plot.

The surface size of the circles on the plot represents symptom severity while the color of the circles represents a particular symptom. The names of the patients are on the vertical axis, while the dates of reports are on the horizontal axis.

PlotTests_shiny.xlsm spreadsheet usage instructions

PlotTest_shiny.xlsm relies on the R `SVGAnnotation` package (Nolan & Lang, 2012) to generate a scalable vector graphics (SVG) file containing the plots.

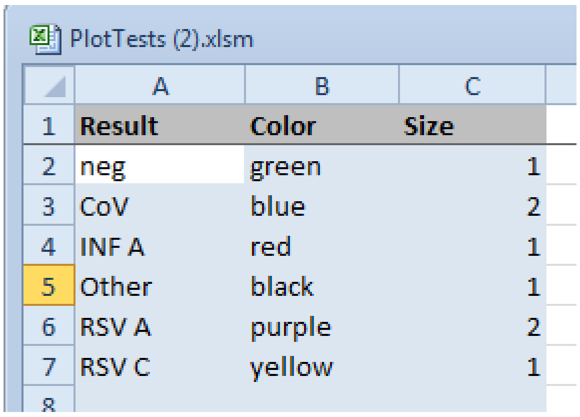
The PlotTests_shiny.xlsm contains the following sheets:

- DATA
- PARAMETERS

The sheets will be described in the order the user should refer to them in the following subparagraphs.

"PARAMETERS" sheet

The PARAMETERS sheet contains the names of valid test results under the Result column, the color for a particular test result under the Color column and the size of the dot on the graph in the Size column. The light blue colored fields are editable by the user.



PlotTests (2).xlsm

	A	B	C
1	Result	Color	Size
2	neg	green	1
3	CoV	blue	2
4	INF A	red	1
5	Other	black	1
6	RSV A	purple	2
7	RSV C	yellow	1
8			

E.g.: The PARAMETERS sheet example.

"DATA" sheet

The DATA sheet contains data about subjects (e.g. patients or medical staff) and their test results on different test days.

The columns used in the sheet are:

- ID: identity code of the subject; alphanumeric
- Name: name of the subject; alphanumeric
- Age: age of the subject; numeric
- Sex: sex of the subject; string
- Type: type of subject; alphanumeric
  The label used in this field determines the 1st level group under which the subject will be placed on the graph. Examples are: Patient, Staff. Can be any label the user chooses. Can be left blank.
- Diagnosis: the diagnosis the patient started treatment for; alphanumeric
  The label used in this field determines the 2nd level group under which the subject will be placed on the graph. Can be any label the user chooses. Can be left blank.

- Outcome: the final outcome of the treatment; alphanumeric
  Can be any label the user chooses. The label “died” plots a special character on the graph at the date under DateOut column. Examples are: improved, died.
- DateIn: the date the treatment started; expected format: DD.MM.YYYY
- DateOut: the date the treatment ended; expected format: DD.MM.YYYY
- The date columns: they contain the dates of tests taken as the column header (expected format: DD.MM.YYYY) and the labels of positive test results on that particular day for a particular subject in the cells. Multiple positive test results should be separated by a comma (e.g.: “CoV,Other”). The user can edit (add) the date columns header labels.

Running the plotting function

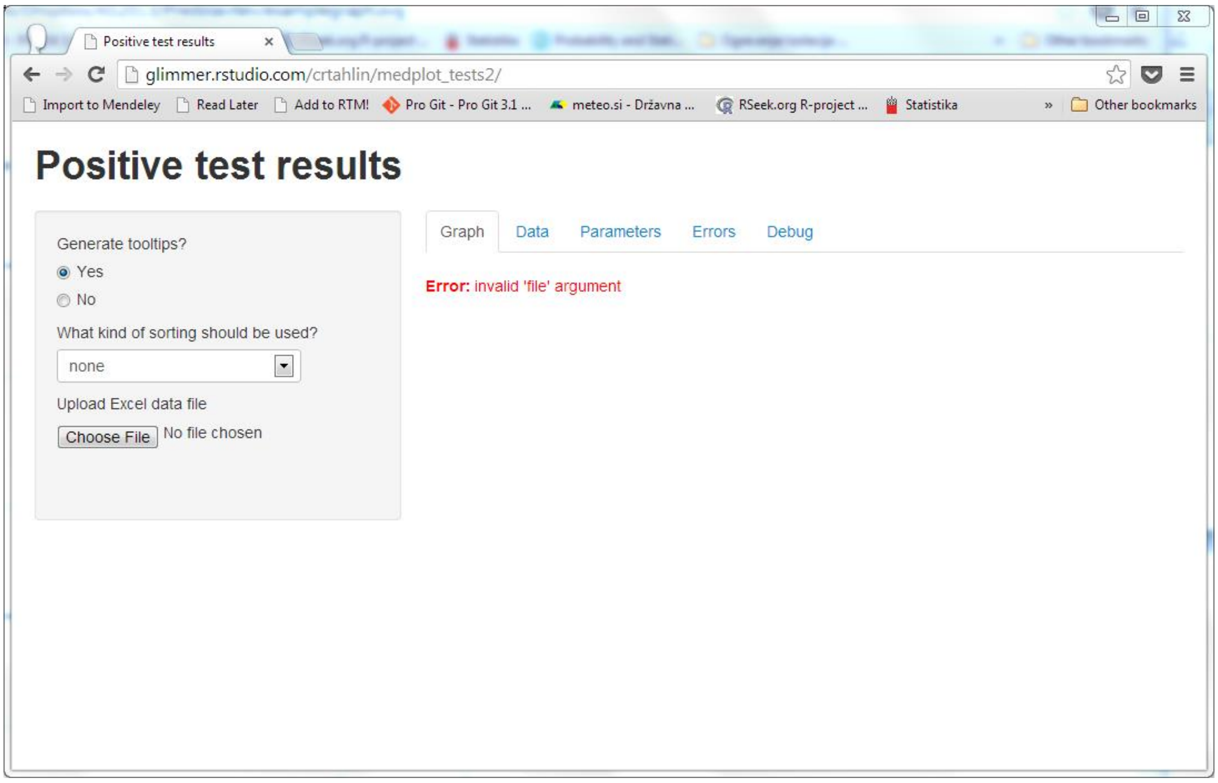
An R shiny app should be started either on a local server or on a remote server. The user then communicates with the app through a web browser, pointing to the address of the server. To start the server locally, open an R console and issue the commands:

```
library("medplot")
library("shiny")
runApp(appDir=file.path(path.package("medplot"),"shinyapp_tests"),
launch.browser=TRUE)
```

These commands should load the required packages, start the shiny application and open the user’s browser at the appropriate address (e.g. localhost:8100).

Note that the firewall on the user’s computer has to allow connections to the server in order for the web browser to be able to connect to it. If a prompt appears after launching the above command asking to allow connections, you should confirm it.

A web browser window should open with options to select extra options for the plot. The plot updates every time after the user changes the options.



Positive test results

Generate tooltips?
(●) Yes
( ) No

What kind of sorting should be used?
none

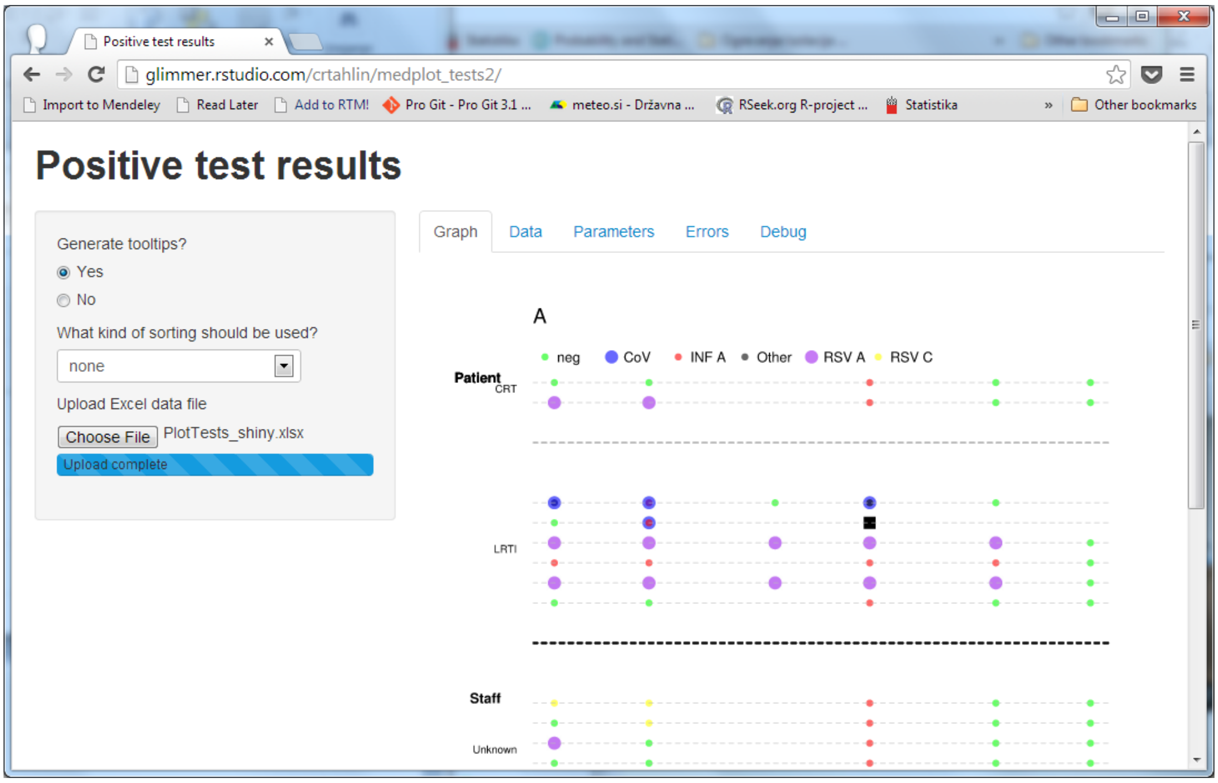
Upload Excel data file
Choose File No file chosen

Graph | Data | Parameters | Errors | Debug

Error: invalid 'file' argument

E.g.: The web browser open connected to (in this case) an external server with extra options available to the user. There is yet no data to plot. Several tabs are visible (Graph, Data, Parameters, Errors) which contain different information.

The user should choose to “Upload Excel data file” and select a file that contains valid data for the graph. Depending on the amount of data, the plot generation can take a while. The graph displays on the right side of the window. The user should have a relatively modern browser (e.g. Chrome version 26+, Internet Explorer 10) with support for SVG images. The plots should look similar to the plot bellow, depending on the version of medplot used.



Positive test results

Generate tooltips?
[✓ Yes] No

What kind of sorting should be used?
none

Upload Excel data file
Choose File PlotTests_shiny.xlsx
Upload complete

E.g.: Example of plotted data.

Some extra options are available to the user on the left side of the screen:

- Generate tooltips? (TRUE/FALSE): set to TRUE if you wish the generated SVG file to contain the functionality of generating pop-up windows with extra information about the data point when the user moves the mouse cursor over a point in the graph (tooltips). This is slower than generating a graph without tooltips.
- Method for sorting: select an option how to sort the units in the generated graph. None – does not sort the units in the graph and uses the order in which they are listed in the Excel sheet. DateIn – will sort according to the DateIn column, which contains dates of arrivals. The rest of the options use the R package `seriation` (Hahsler, Hornik, & Buchta, 2007) for sorting, using the values of test results entered on a particular date. BEA – will sort via the bond energy algorithm; BEA_TSP – uses the travelling salesperson algorithm; PCA – uses the first principal component for sorting. The idea behind sorting is that patterns of test results might emerge visually if the units are sorted on the graph.

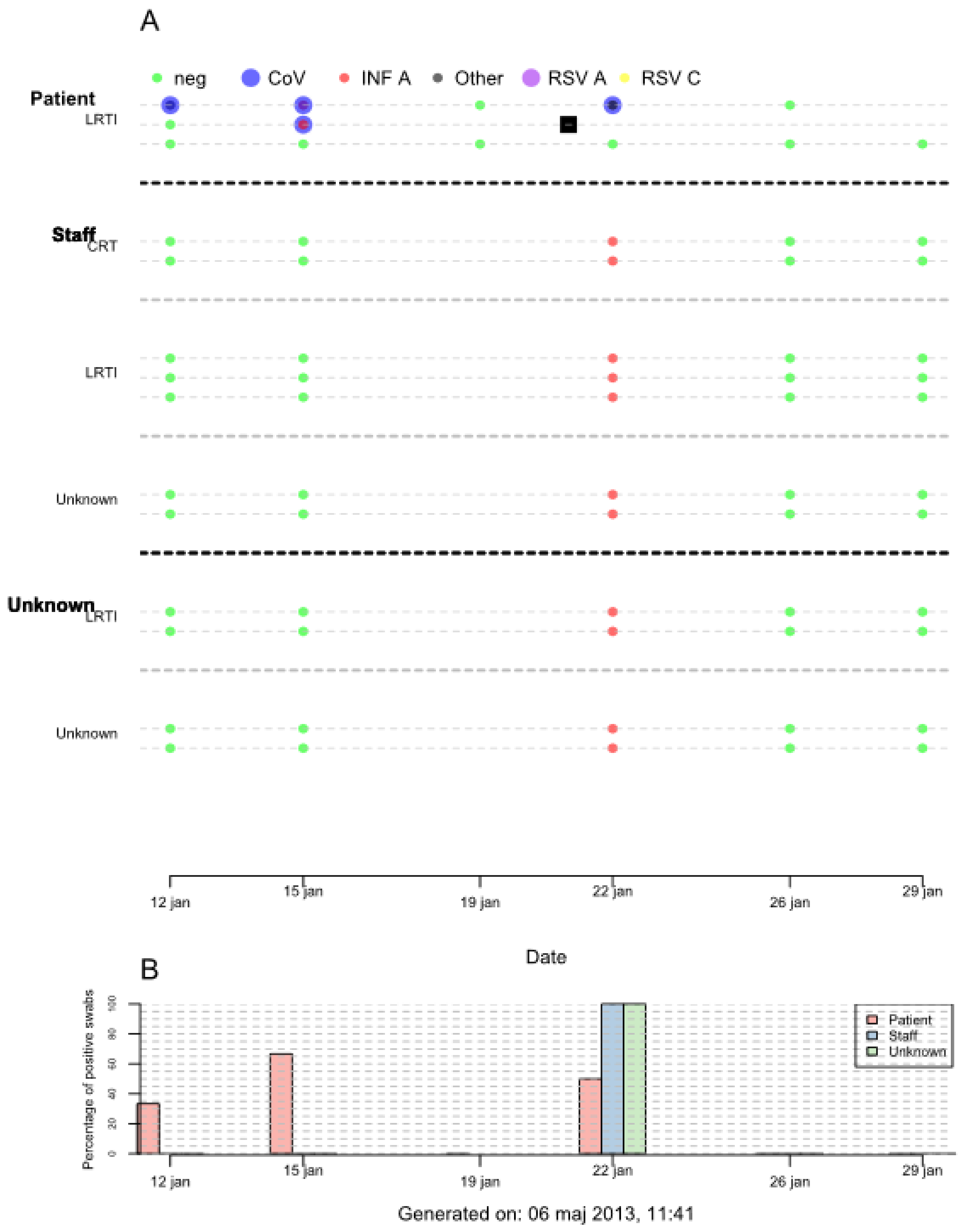
After the user changes one of the available options, the plot gets re-plotted according to the new settings.

The tabs on the right side of the screen contain:

- Graph: contains the plotted graph.
- Data: contains the data the user uploaded via the Excel file.
- Parameters: contains the parameters the user uploaded via the Excel file.
- Errors: Potential errors and warnings get listed on the Errors tab. The user should check the tab after running the plotting function. For example, errors might occur if invalid test result

labels are used on the DATA sheet in Excel. Compared to errors, warnings usually do not stop the plotting, but are caused situations the user should be aware of.

The plot



E.g.: Example of Plot Tests plot.

The graph under the label "A" shows test results for subjects on different days of testing. Moving the mouse over a plotted dot will raise a window with additional details about the subject (on the plot with tooltips, not on the “normal” SVG plot). The graph under the label “B” shows the percentage of positive swabs for different 1st level groups of subjects on different dates via a bar chart. All test results that are not “neg” or NULL are considered positive.

Bibliography

Hahsler, M., Hornik, K., & Buchta, C. (2007). Getting Things in Order: An introduction to the R package seriation, (2001). Retrieved from http://epub.wu.ac.at/852/

Nolan, D., & Lang, D. T. (2012). Interactive and Animated Scalable Vector Graphics. *Journal of Statistical Software*, *46*(1).

R Core Team. (2013). R: A Language and Environment for Statistical Computing. Vienna, Austria. Retrieved from http://www.r-project.org/

RStudio, & Inc. (2013). shiny: Web Application Framework for R. Retrieved from http://cran.r-project.org/package=shiny