CoolMap R Plugin Manual

Introduction

The R plugin for CoolMap is designed to provide statistical analysis capability for data displayed in CoolMap and the display of analysis results directly in CoolMap.

CoolMap comes with R plugin as default. The R plugin folder is placed under CoolMap/plugin/r_plugin. If you somehow uninstalled or broke the R plugin, download it again on the CoolMap website (coolmap.org) and place it under the CoolMap plugin folder.

The CoolMap R plugin requires the running of Rserve, which enables the communication between CoolMap and the R statistical environment.

Rserve Setup for CoolMap

CoolMap supports three different ways of Rserve setup. The required Rserve installation and initiating procedures are briefly described below. More details about Rserve setup and troubleshooting are at https://rforge.net/Rserve/doc.html .

Local Rserve on the same computer that runs CoolMap

The easiest way is to open an R session on the same computer on which you installed CoolMap. This particular R session can be closed after the Rserve is initiated since CoolMap will create a separate R session through the initiated Rserve. To start Rserve, type below commands in a R command line.

```
install.packages("Rserve")        # only needed for the first time installation

library(Rserve)

Rserve()       # may need to specify args=’—no-save’ on MAC OS
```

Remote Rserve on a server in your network

Setting up a remote Rserve on a server with more powerful processor, larger memory and storage can support large computing tasks, analysis needs for multiple users and file sharing among users.

If you plan to setup remote Rserve, please install Rserve on a linux core based OS so that the server can support multiple connections from different users. Windows version of Rserve only allows one connection at a time.

You also need to configure Rserve to be able to answer queries submitted remotely by editing /etc/Rserv.conf and add a line ‘remote enable’ in it. To edit this file, you have to use the root privilege.

To enable CoolMap on your computer to use the remote Rserve, you need to modify the config.json under CoolMap/plugin/r_plugin by changing default value “localhost” of property ‘server-url’ to the hostname or IP address (with the same double quotation marks as “localhost”) of the server on which Rserve runs.

CoolMap Rserve service

We setup CoolMap Rserve remote service for users with no access to high performance Rserve setup but want to perform computations not feasible on their own laptop/desktop computers. Only encrypted texts are used between users' computers and the CoolMap Rserve remote service.

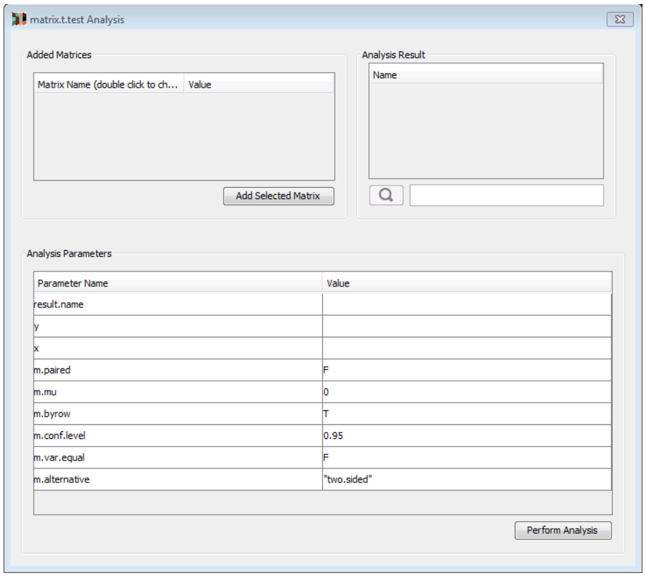
To use this service with CoolMap on your computer, you need to delete the word 'localhost' inside the double quotation marks in the config.json file under CoolMap/plugin/r_plugin.

Use Pre-Bulit R Analysis Functions in CoolMap

After R plugin setup, you will see available analysis functions under the Analysis menu. The analysis function list is derived dynamically from the R scripts stored under folder CoolMap/plugin/r_plugin /scripts.

To use a function, just click it in the Analysis menu. A graphic user interface will pop-up for you to select the data from the CoolMap view at current focus and related function parameters. The graphic user interface allows users to perform analysis using pre-built functions without writing R scripts.

Example 1: Matrix t-test



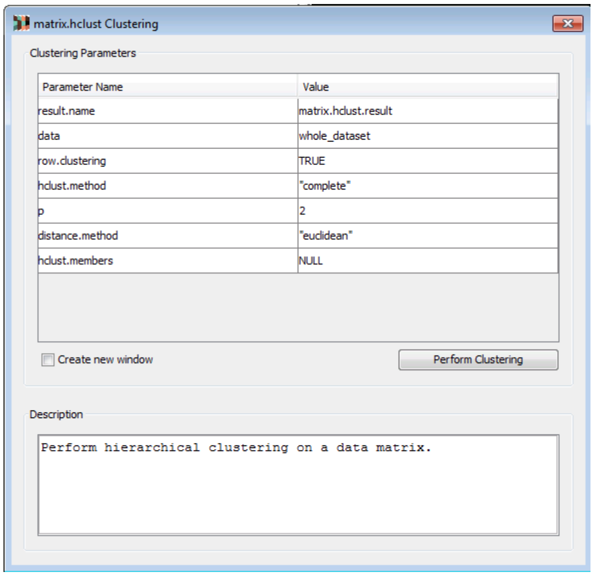
Parameter Name	Value
result.name	
y	
x	
m.paired	F
m.mu	0
m.byrow	T
m.conf.level	0.95
m.var.equal	F
m.alternative	"two.sided"

On the left is the GUI data analysis dialog window generated from R script CoolMap/plugin/r_plugin/scripts/analysis/matrix.t.test.R.

You can import matrices required for the analysis from CoolMap canvas and they will be listed in the top left box of the dialogue window. You can also change the names of these matrices by double clicking on the matrix name.

The dialogue box below the matrices list is the parameter table where you can tune the function parameters. By hovering over the parameter names, you can see the description of the parameter. Note there is a special parameter named 'result.name'. It is where you specify a name for the result variable of the analysis function, thus you can access the result by this variable in 'R Console', which will be detailed later in this manual. The top right box contains the output of the analysis function. If an output variable corresponds to row or columns nodes on CoolMap canvas, you can further map them to the row/column highlightor by right clicking on these variables listed in 'Analysis Result'. In this case, after a t-test, the resulted p-value of the function can then be highlighted to the row/column to facilitate data exploration in CoolMap.

Example 2: Hierarchical Clustering



Parameter Name	Value
result.name	matrix.hclust.result
data	whole_dataset
row.clustering	TRUE
hclust.method	"complete"
p	2
distance.method	"euclidean"
hclust.members	NULL

```
Perform hierarchical clustering on a data matrix.
```

On the right is the dialog window for hierarchical clustering. This dialog was generated by R script matrix.hclust.R under CoolMap/plugin/r_plugin/scripts/cluster/. The upper half of the dialog lists all the parameters you have to set. By hovering over the parameter names, you can see the description of the parameter. Note there is a special parameter named 'result.name'. It is where you specify a name for the result variable of the clustering function, thus you can access the result by this variable in 'R Console', which will be detailed later in this manual. The bottom half of the dialog is the description of the clustering function. After setting all the parameters in the 'Value' column, just press the perform cluster button to perform the clustering function and the data in the CoolMap will be clustered.

Parameter 'data' can be set to 'whole_dataset' which means the clustering will be conducted on the whole dataset in the current CoolMap canvas, or 'selected_region" to perform clustering on the selected matrix in CoolMap canvas.

Parameter 'row.clustering' can be set to 'TRUE' to perform the clustering based on data in rows. When it is set to 'FALSE', the clustering will be based on data in columns.

The rest of the parameters and their selections are the same as the standard R hclust function.

The clustering results will be directly drawn on the row or column side of the selected data region in the CoolMap view if leave 'Create new window' unchecked, otherwise, a new CoolMap view with the data specified by parameter 'data' will be created first and then clustered.

Write custom R Analysis Functions for Sharing

The CoolMap program can automatically generate a graphic user interface for each analysis function based on the parameter list and the output list described in each analysis function. As a result, users can write custom analysis functions in R and put them under CoolMap/plugin/r_plugin /scripts without writing GUI for each function.

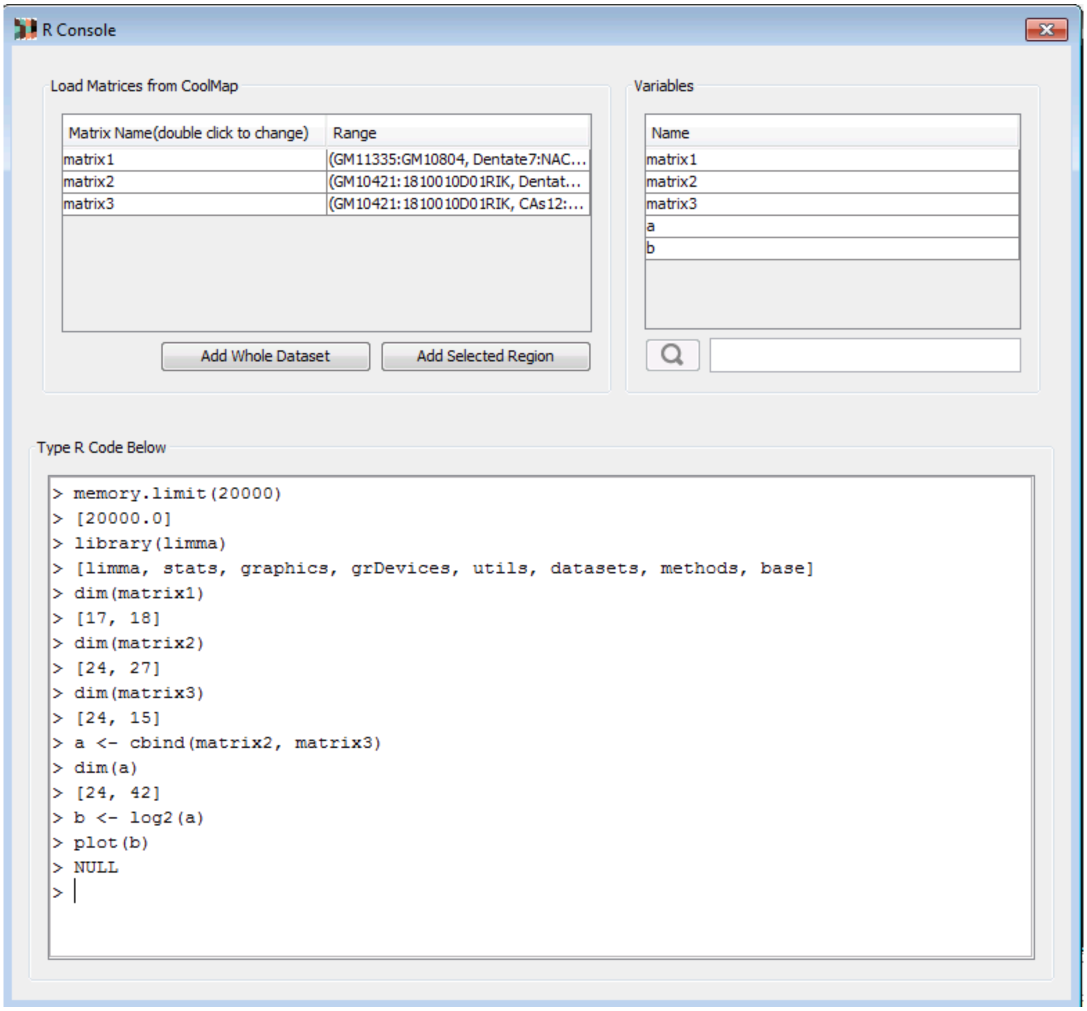
Users can follow standard R function writing rules with two additional requirements 1) list all the parameters that may need user selection/input in the function name line 2) list all function outputs in the last line of the function as a R list. Examples of pre-built functions can be found under CoolMap/plugin/r_plugin /scripts. Naturally, it is straightforward to share pre-built R scripts for CoolMap among different users.

Use R Console in CoolMap

The command line R console, accessible through the Plugin->R->R console menu item, can be used for more flexible data manipulation. You can import the whole matrix in CoolMap or selected region on the a CoolMap view as R matrix, run R statements in the R command input area and get results in the R command output area at the bottom of the R console window (see screen capture below).

A common need for the CoolMap R Console in the Windows environment is to adjust memory available to the CoolMap R Console. You can use the 'memory.limit(M)' with M be the number of megabytes needed. You can also run 'gc()' to free the memory so that you can get more memory for your analysis.

Since all R analyses in a CoolMap use the same R session, all data objects created by the pre-built analysis functions are also accessible through the command line R console. As a result, this R console is useful for the examination of various data objects and quick ad hoc analysis. You may also close and reopen the R console window under the same Rserve() connection and reuse all data objects imported or created in the previous R console window.



Matrix Name(double click to change)	Range
matrix1	(GM11335:GM10804, Dentate7:NAC...
matrix2	(GM10421:1810010D01RIK, Dentat...
matrix3	(GM10421:1810010D01RIK, CAs12:...

Name
matrix1
matrix2
matrix3
a
b

```
> memory.limit(20000)
> [20000.0]
> library(limma)
> [limma, stats, graphics, grDevices, utils, datasets, methods, base]
> dim(matrix1)
> [17, 18]
> dim(matrix2)
> [24, 27]
> dim(matrix3)
> [24, 15]
> a <- cbind(matrix2, matrix3)
> dim(a)
> [24, 42]
> b <- log2(a)
> plot(b)
> NULL
> 
```

The CoolMap R console behaves very similarly to the standalone R console except 1) it only display the results (including matrix) in one line in the bottom window 2) for some commands the return string may be different from the standard R console. Making the R console output as close to the standalone R console as possible will be a goal of our next phase of development.

You can use R packages you installed previously on your computer as well as using the R graphics functions, although the resulting R graphics is not connected to CoolMap views. One issue under the Windows environment is if R Graphics pop-up window is initially hidden behind windows of other programs, Windows may fill the R Graphics window even if you bring it to the front by clicking the icon on the Windows task bar. Under this situation, you only need to rerun the R graphic again.

The Variable window at the upper right corner of the R console window serves two important purposes. Firstly, it is the graphic interface for the R ‘objects()’ function and it can be used to search all existing R data objects. You can type in the object pattern string in the data entry box on the left of the “Search” button. Secondly, it provides a way to send the selected variable to create a new CoolMap view or to modify the currently active CoolMap view. Right clicking a variable in the Variable Inspector window will bring out a pop-up menu that provides three operations for the variable:

Send selected to the CoolMap (must be a matrix)
Highlight selected to column
Highlight selected to row

As a result, you can use ad hoc R analysis results to refine CoolMap-based data exploration iteratively.