Marxan.net user guide

December 2013

Matt Watts m.watts@uq.edu.au

Marxan acknowledges funding from the NeCTAR project http://www.nectar.org.au as well as through the Queensland Cyber Infrastructure Foundation (QCIF). Nectar is an Australian Government project conducted as part of the Super Science initiative and financed by the Education Investment Fund.

Marxan.net is supported by:

The University of Queensland
National Environmental Research Program Environmental Decisions Hub
Australian Research Council Centre of Excellence for Environmental Decisions
Queensland Cyber Infrastructure Foundation
National eResearch Collaboration Tools and Resources Research Cloud
Research Data Storage Infrastructure Project

About Marxan.net

Marxan is the most widely used decision support software for conservation planning globally. It is used in over 100 countries to build marine and terrestrial conservation plans and national parks systems. It leads the world market for conservation land-use planning and new extensions are making it even more popular.

Previously, the only supported deployment model of the software was through download and install by users onto Linux, Mac and Windows operating systems.

During 2013, we initiated a project to migrate the software to the QCIF research cloud (Q-Cloud) to enhance the utility of the software for ourselves and our researcher collaborators. QCIF provided funding assistance to complete this migration project.

The cloud migration improves the ability for us to inform conservation planning decisions in Australia and globally and increase its utility worldwide. It delivers an easy to use method for researchers to utilise supercomputing resources for applying the Marxan software.

It consists of a cloud-based version of Marxan with an easy to use web based interface.

Our global user base benefit through improved delivery of decision support, enhanced scalability of computation, improved software performance, and better ease of use.

Matt Watts and Hugh Possingham from the UQ Centre for Biodiversity and Conservation Science developed Marxan.net in collaboration with Minh Huynh and David Abramson from the UQ Research Computing Centre.

Architecture of Marxan.net

The system consists of a series of Linux virtual machines running in Q-Cloud. The virtual machines run a range of software including R Studio Server, R Shiny Server, Java, and Nimrod Server. Computing resources for the system can be dynamically expanded in capacity at runtime through cloud bursting to the Amazon EC2 compute cluster.

Users can log in to the R Studio Server console and access a series of R Shiny Server applications to conduct Marxan analysis.

The R Studio Server Interface

This interface is intended for project administrators. It has graphical user interface elements and command line interface elements. Within this interface,

users can develop software, run Marxan, display input and output maps and tables, upload and download files, develop customised analysis, develop problem formulations and publish and unpublish R Shiny Server applications.

The R Shiny Server Interface

This interface is a series of R Shiny Server web applications that are intended for end users of Marxan. Within these web applications, users can run Marxan, display maps and tables, edit key parameters, and run calibration algorithms. The interfaces can be customised and embedded in web sites and web applications.

Cloud bursting with Nimrod

The system generates a series of Marxan scenarios with differing parameter values for feature targets, feature penalty factors, planning unit cost weightings, boundary length modifiers, and the number of iterations. These scenarios are passed to a Nimrod server where they are executed in parallel on a series of NeCTAR virtual machines and Amazon EC2 virtual machines. Nimrod then passes the completed results back to the system for visualisation by the user. This acts as an aid to the user in selecting appropriate values for these key parameters, and represents a powerful calibration system utilising the private NeCTAR cloud and the public Amazon EC2 cloud.

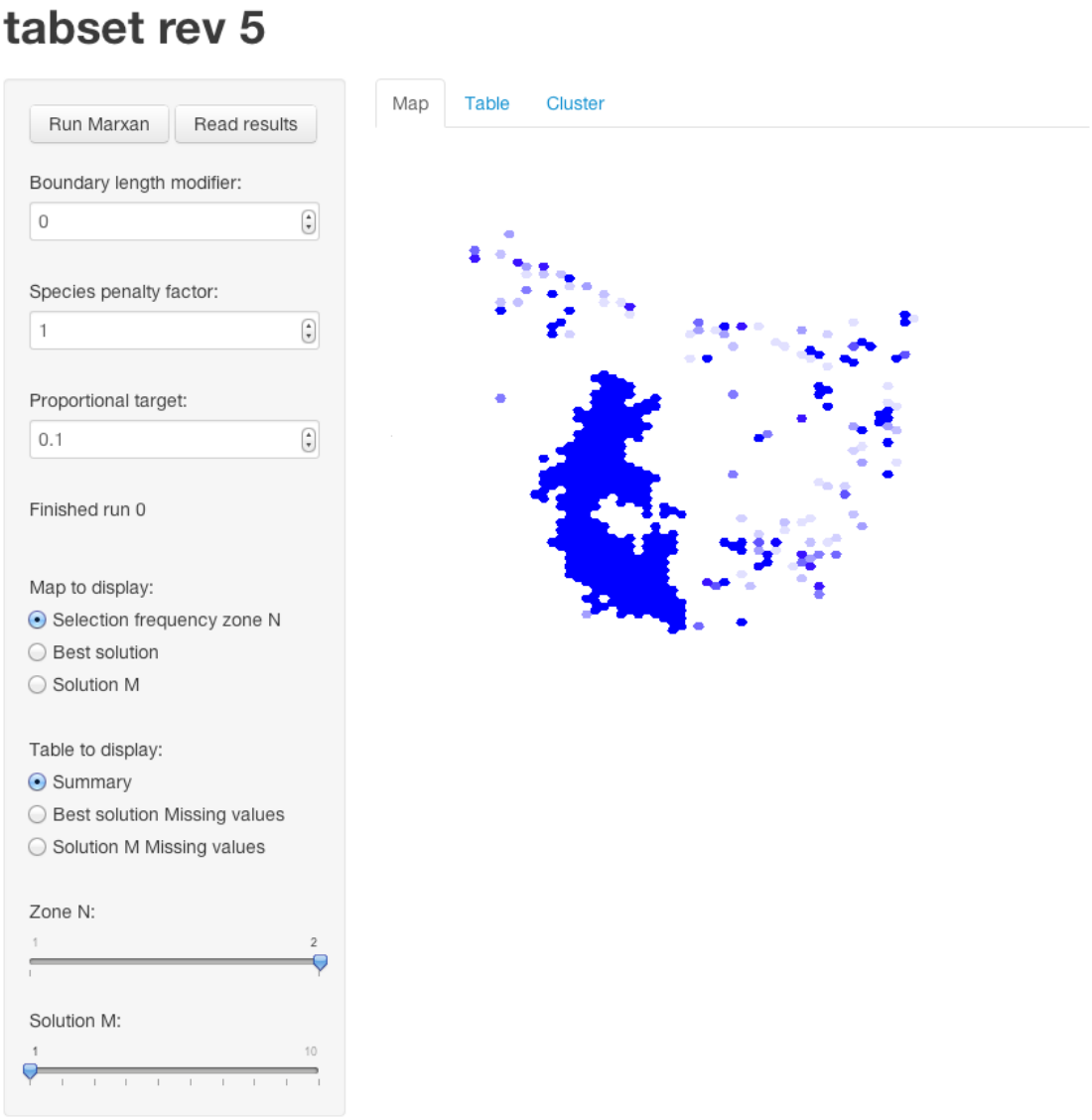
The system has a dashboard that monitors the performance of analysis conducted by Nimrod and presents this analysis in a user-friendly graphical form.

Using the R Shiny Server apps

These web applications are a series of graphical user interfaces where users can run Marxan, display maps and tables, edit key parameters, and run calibration algorithms. They are intended for end-users of Marxan. Access them by clicking “R Shiny Server apps” on Marxan.net

Shiny app: Run Marxan

The app is a web graphical user interface that allows users to run Marxan, display maps and tables, and edit key parameters. It has a control panel on the left, and a tabbed output panel on the right.



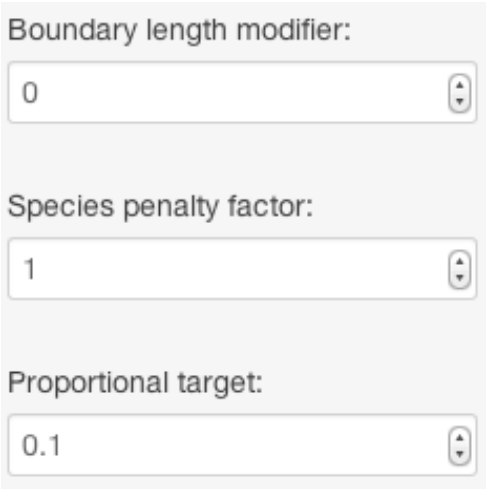
The app is compatible with all current versions of Marxan, including Marxan, Marxan with Zones, and Marxan with Probability.

Project administrators of Marxan.net are able to publish their own datasets with the app so they can share their datasets publicly, or securely and privately to a selected group of people.

Edit key parameters

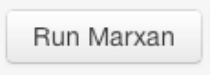
Edit a key parameter by clicking in the parameter box on the control panel, and then typing in the value you want. Alternatively, use the arrow buttons in the parameter box to increase or decrease the value.

The key parameters are “Boundary length modifier”, “Species penalty factor”, and “Proportional target”:



Run Marxan

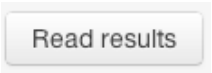
To run Marxan, click the "Run Marxan" button:



With the Tas Activity dataset, it takes around 6 seconds to Run Marxan to generate 10 solutions.

During this time, the "Finished run" label fades to light grey. When the solutions are generated, the counter increases on the "Finished run" label and it appears black again:

To view the results of the completed runs, click the "Read Results" button:



This loads the solutions into the GIS shape file, runs cluster analysis on the unique solutions, and renders the map and cluster analysis graphs.

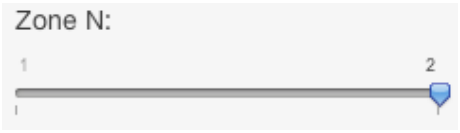
If you see an error message like "Error: 'k' must be in {1, 2, .. n – 1}" when running Marxan, it means there were not enough unique solutions generated for the cluster analysis algorithms to work. In this case, you can just edit the key parameters and run Marxan again.

Display output maps

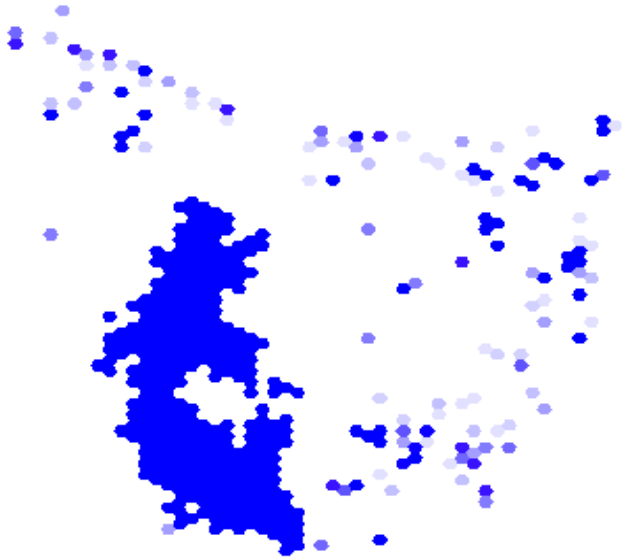
To see output maps, click the "Map" tab.

View the output maps of selection frequency by clicking “Map to display: Selection frequency zone N”:

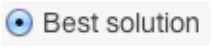
Set the “Zone N” slider to “1” to display selection frequency of available zone. Set the slider to “2” to display selection frequency of reserved zone:



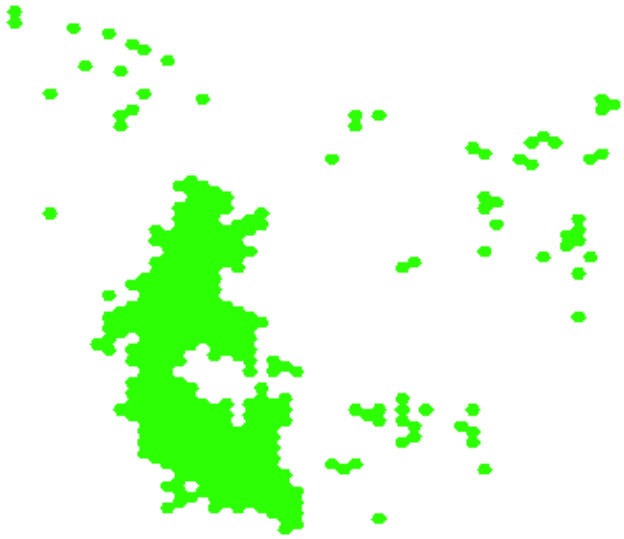
The selection frequency is displayed in a gradient of shades of blue. The darker shades of blue were selected more frequently, and the lighter shades were selected less frequently. Planning units that were never selected are displayed in white.



View the output map of best solution by clicking “Map to display: Best solution”:



Planning units that were selected in this solution are displayed in green, and planning units that were not selected in this solution are displayed in white.



View all the solutions generated by clicking “Map to display: Solution M”:

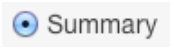
Set the “Solution M” slider to the solution you want to display:

Display output tables

To see output tables, click the “Table” tab.

	Run_Number	Score	Cost	Planning_Units	Connectivity
1	1	9572069644.73	9452257095.61	452	4192000.0(
2	2	9562171547.67	9438824854.91	451	4192000.0(
3	3	9575261269.19	9443767946.27	449	4240000.0(
4	4	9579909794.86	9447743900.85	448	4136000.0(
5	5	9572985874.24	9444595986.51	446	4136000.0(
6	6	9567235689.10	9441611285.37	450	4168000.0(
7	7	9577259965.33	9451701018.15	449	4184000.0(
8	8	9570682746.09	9445939825.83	448	4176000.0(
9	9	9573724448.18	9451659438.88	448	4144000.0(
10	10	9576327176.36	9444680103.37	449	4240000.0(

View the output summary table by clicking “Table to display: Summary”:



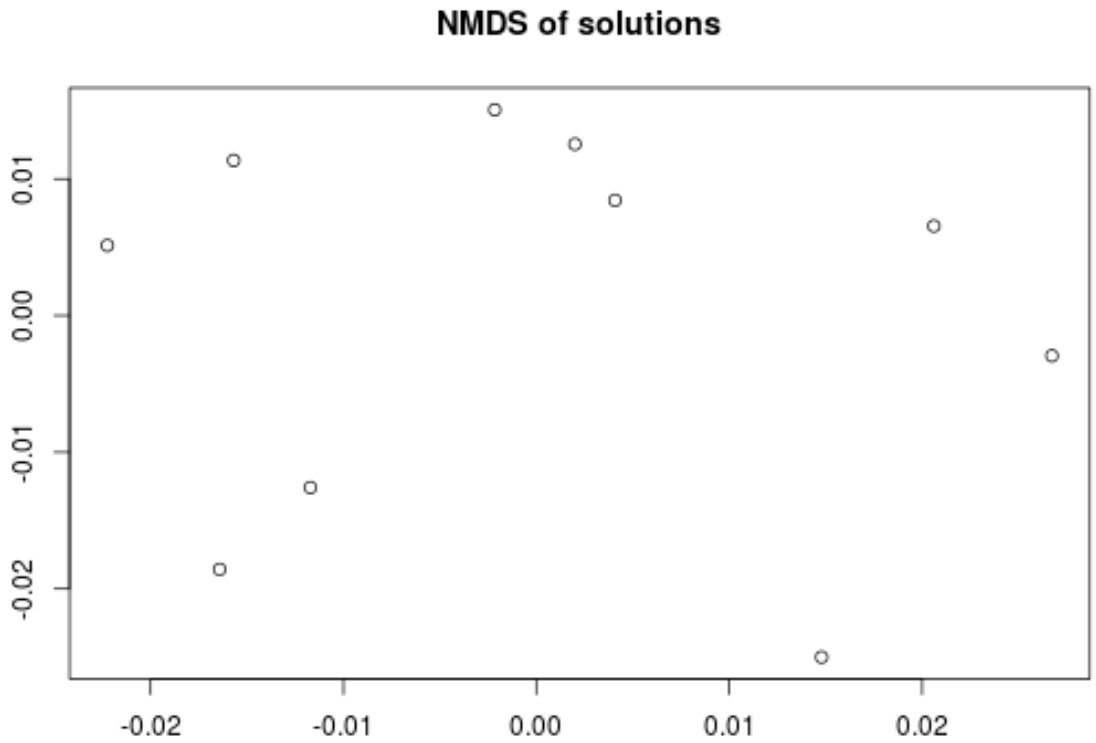
View the output missing values table by clicking “Table to display: Best solution Missing values”:

View the missing values table for all the solutions generated by clicking “Table to display: Solution M Missing values”:

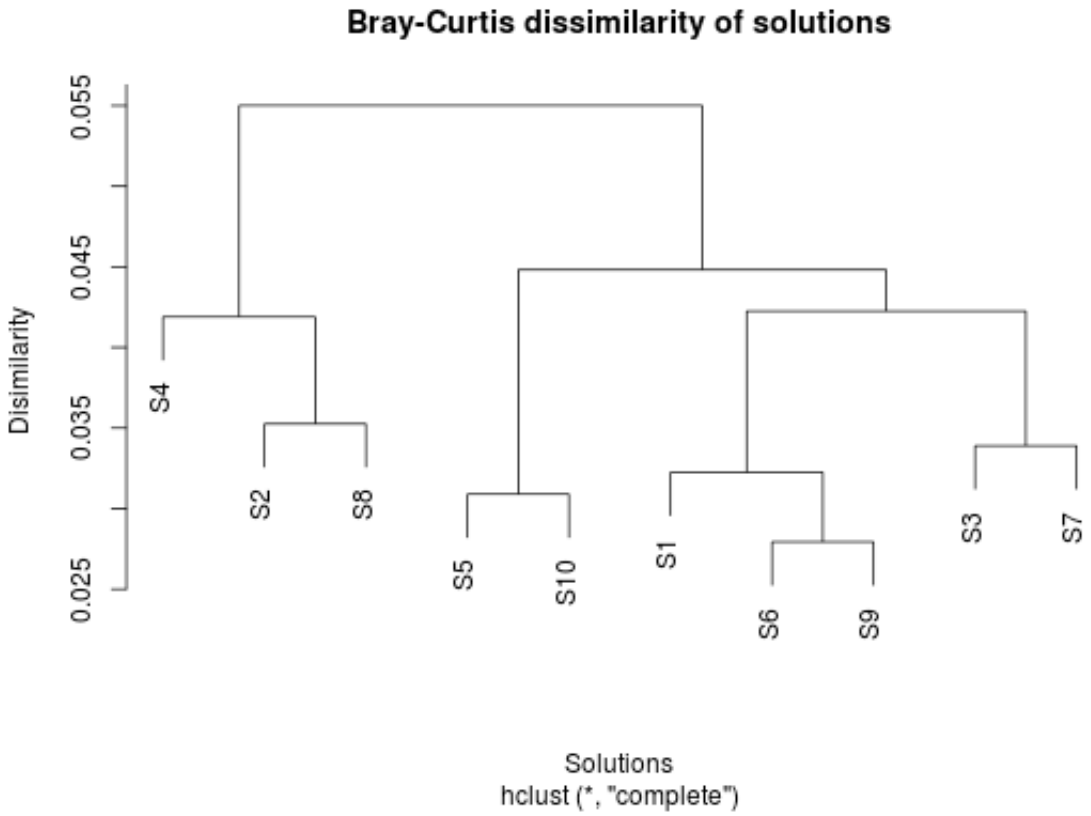
Set the “Solution M” slider to the solution you want to display:

Display cluster analysis

To see output cluster analysis graphs, click the “Cluster” tab. A graph showing the two dimensional scaling of the NMDS of unique solutions is displayed.



A dendogram displaying the Bray-Curtis dissimilarity of unique solutions is also displayed.



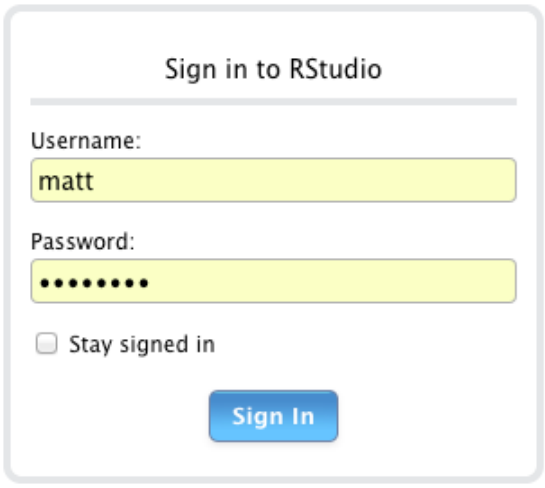
R Studio Server Tutorial

We present a series of exercises to demonstrate how to use the key functionality of Marxan with R Studio Server in Marxan.net.

Step 1: Log into Marxan.net

Using a web browser, browse to Marxan.net.

Click on the link “Login to R Studio Server console”.

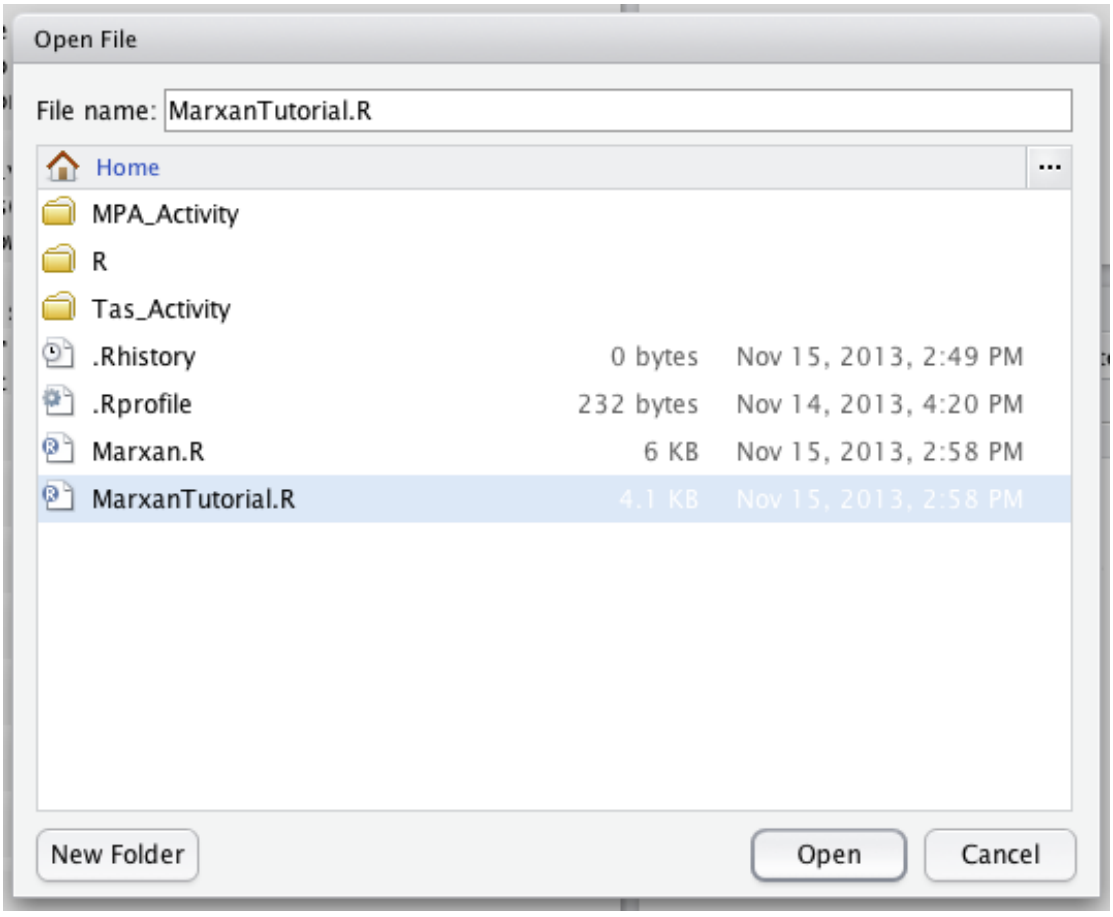


Enter the Username and Password provided to you by system administrators.

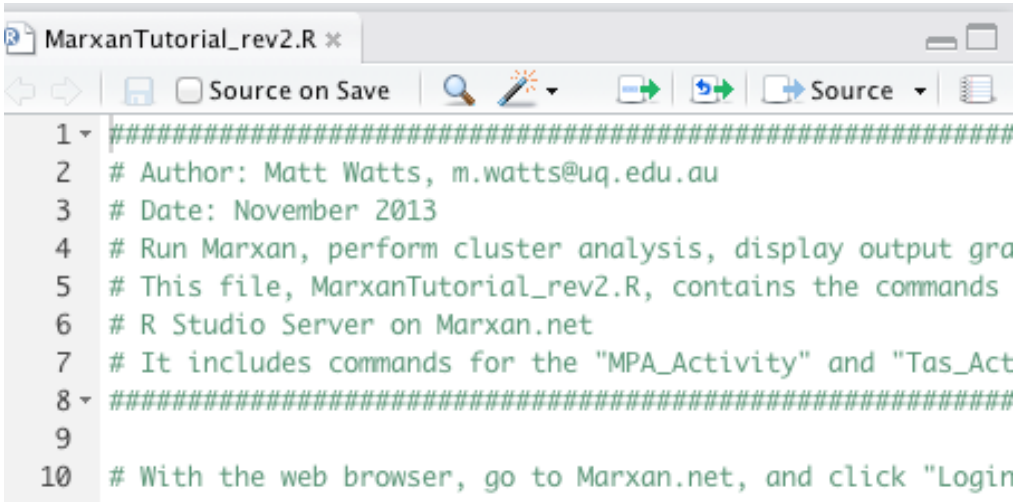
Step 2: Open the MarxanTutorial_rev2.1.R file in R Studio Server

Click on the "File" Menu in R Studio Server, and then click "Open File…".

Click the "MarxanTutorial_rev2.1.R" file, and then click "Open".



The MarxanTutorial_rev2.1.R file will now be loaded in R Studio Server.



You can now easily execute commands from the tutorial in the console of R Studio Server. There are three simple ways to execute these commands:

- Copy and paste the commands from "MarxanTutorial_rev2.1.R" or from a file open on your local computer to the Console in R Studio Server, or

- Click the row or select the rows in "MarxanTutorial_rev2.1.R" to execute and then press the "Run the current line or selection" button,  or

- Click the row or select the rows in "MarxanTutorial_rev2.1.R" to execute and then press the CTRL+ENTER on the keyboard.

Step 3: Execute the "Getting started with Marxan" commands

NOTE: Before you execute these commands, edit line 22, and **change "matt" to the username you have been given**.

```
22  sHome <- "/mnt/users/matt/"
```

Using either of these methods of executing commands, execute the commands from lines 23 up to lines 27. The system is now ready to run Marxan with your dataset.

```
23  sWorkingDir <- paste0(sHome,"MPA_Activity/")
24  #sWorkingDir <- paste0(sHome,"Tas_Activity/")
25  source(paste0(sHome,"Marxan_rev2.R"))
26  setwd(sWorkingDir)
27  system(paste0("chmod +x ",sWorkingDir,"MarOpt_v
```

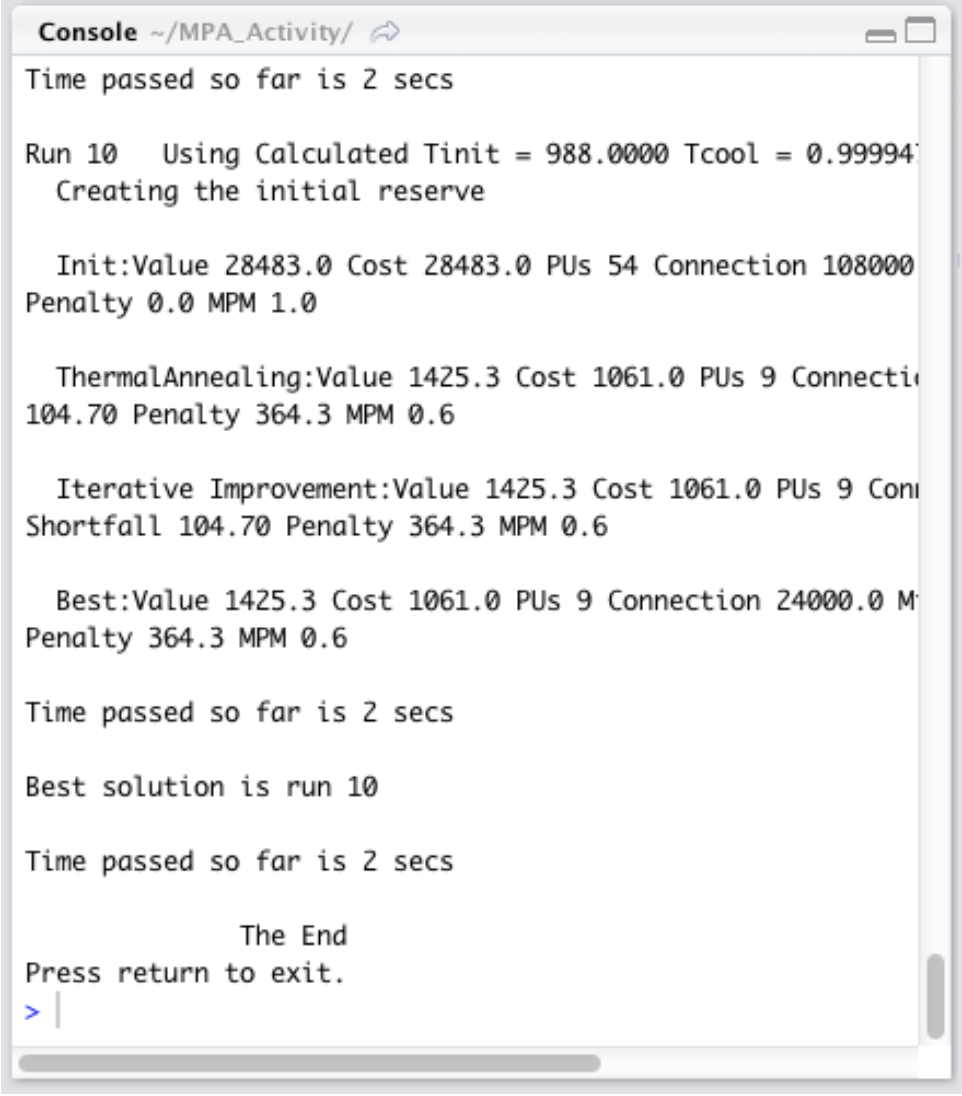
If you want to run Marxan with "Tas_Activity" instead of "MPA_Activity", then comment line 23 with a # and uncomment line 24 by removing the # before executing the commands.

Step 4: Run Marxan

Execute the command on line 30 to run Marxan.

```
29  # Step 6: Run Marxan
30  system(paste0(sWorkingDir,"MarOpt_v243_Linux64 -s"))
```

After you execute this command, you will see the Marxan console information scrolling down on the console window until Marxan finishes.



Step 5: Display output maps

Import the Marxan map outputs from the Marxan output files to the Marxan planning unit shape file by executing the command on lines 34 and 35.

```
33  # Import the outputs to the shape file so we can display
34  ImportOutputsCsvToShpDbf(paste0(sWorkingDir,"pulayer/pula
35                           sWorkingDir, 10,"PUID")
36  # Note: if your PUID field is not called "PUID", change t
```

Load the Marxan planning unit shape file into R Studio Server by executing the commands on lines 39, 40, and 41.

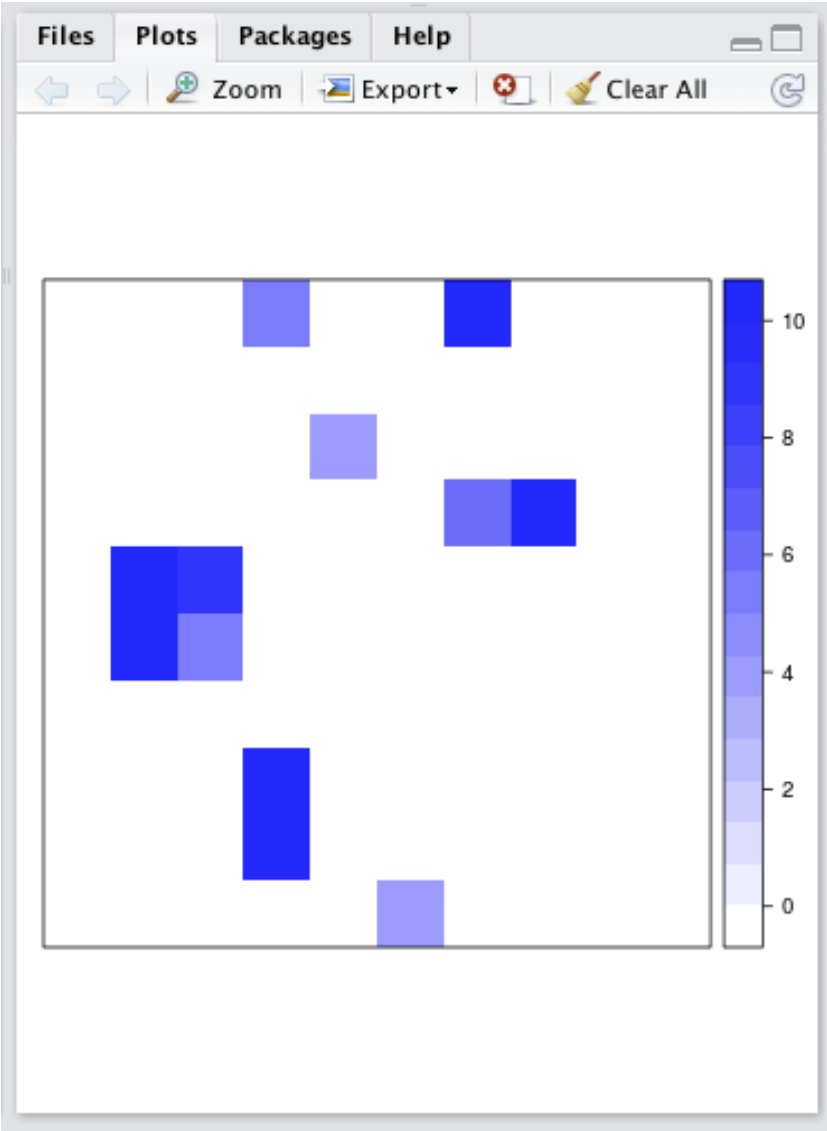
```
39  pulayer <- readShapePoly(paste0(sWorkingDir,"pulayer/pula
40  pupolygons <- SpatialPolygons2PolySet(pulayer)
41  putable <- read.dbf(paste0(sWorkingDir,"/pulayer/pulayer.
```

Now we are ready to display some maps. If you want to display a map showing the selection frequency of the available zone, execute the command on line 43, or the command on line 59.

```
43  DisplaySsolnMap(pulayer,2,TRUE) # reserved zone
```

```
59  DisplaySsolnMapPBSm(pupolygons,putable,2,100,"blue",TRUE)
```

The map now displays on the “Plots” tab of the window on the bottom right hand side of R Studio Server.

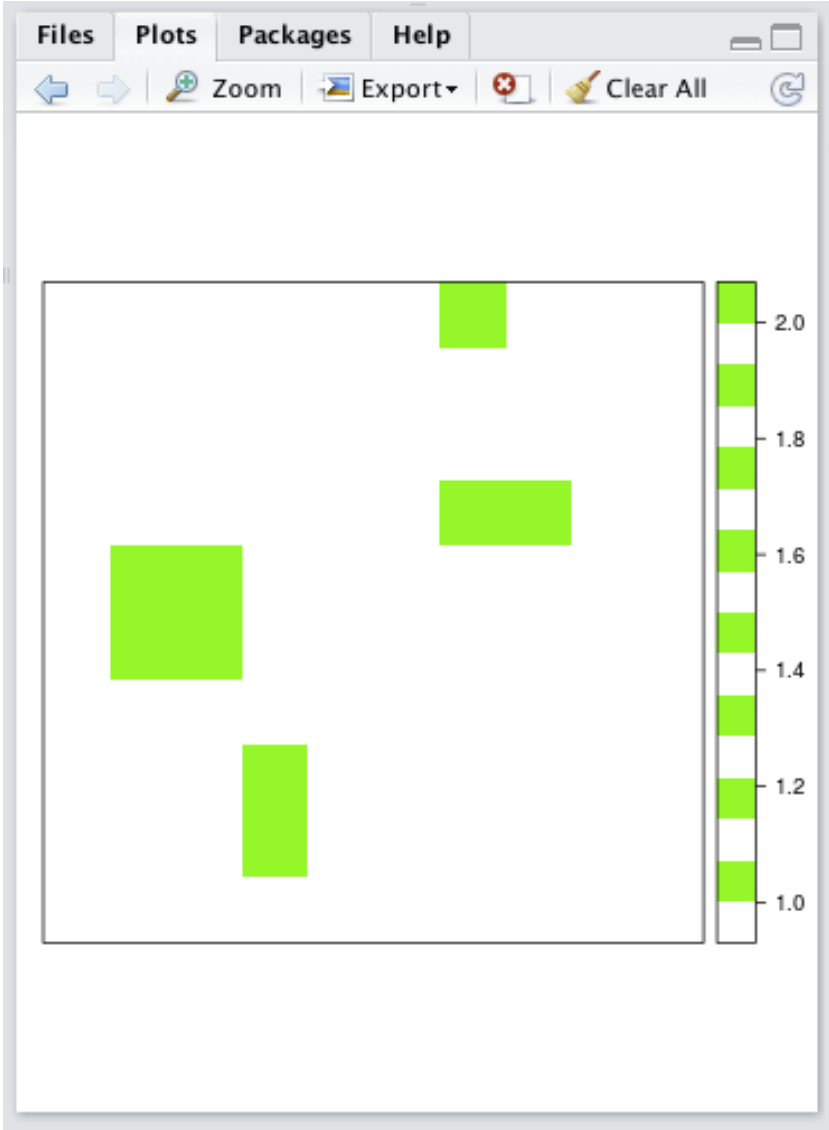


If you want to display a map showing the best solution, execute the command on line 44, or the command on line 67.

```
44  DisplayMap(pulayer,0,TRUE) # best solution
```

```
67  DisplayMapPBSm(pupolygons,putable,0,c("white","green"),TRU
```

The map now displays on the “Plots” tab of the window on the bottom right hand side of R Studio Server.



Other commands are included in "MarxanTutorial_rev2.R" for displaying the rest of the maps of Marxan output.

Step 6: Display output tables

If you want to display a table showing the Marxan summary table, execute the command on line 76.

```
76  DisplaySumTable(sWorkingDir)
```

The summary table then displays in the window at the top left hand side of R Studio Server.

MarxanTutorial_rev2.R × | sumtable ×

10 observations of 14 variables

	Run_Number	Score	Cost	Planning_Units	Connectivity	Con
1	1	1454.226	1381	9	32000	2200
2	2	1427.226	1354	10	34000	2200
3	3	1431.952	922	9	26000	2200
4	4	1427.226	1354	10	34000	2200
5	5	1427.226	1354	10	34000	2200
6	6	1425.260	1061	9	24000	2200
7	7	1427.226	1354	10	34000	2200
8	8	1427.226	1354	10	34000	2200
9	9	1431.952	922	9	26000	2200
10	10	1427.226	1354	10	34000	2200

Other commands are included in "MarxanTutorial_rev2.R" for displaying other tables of Marxan output.

Step 7: Display cluster analysis

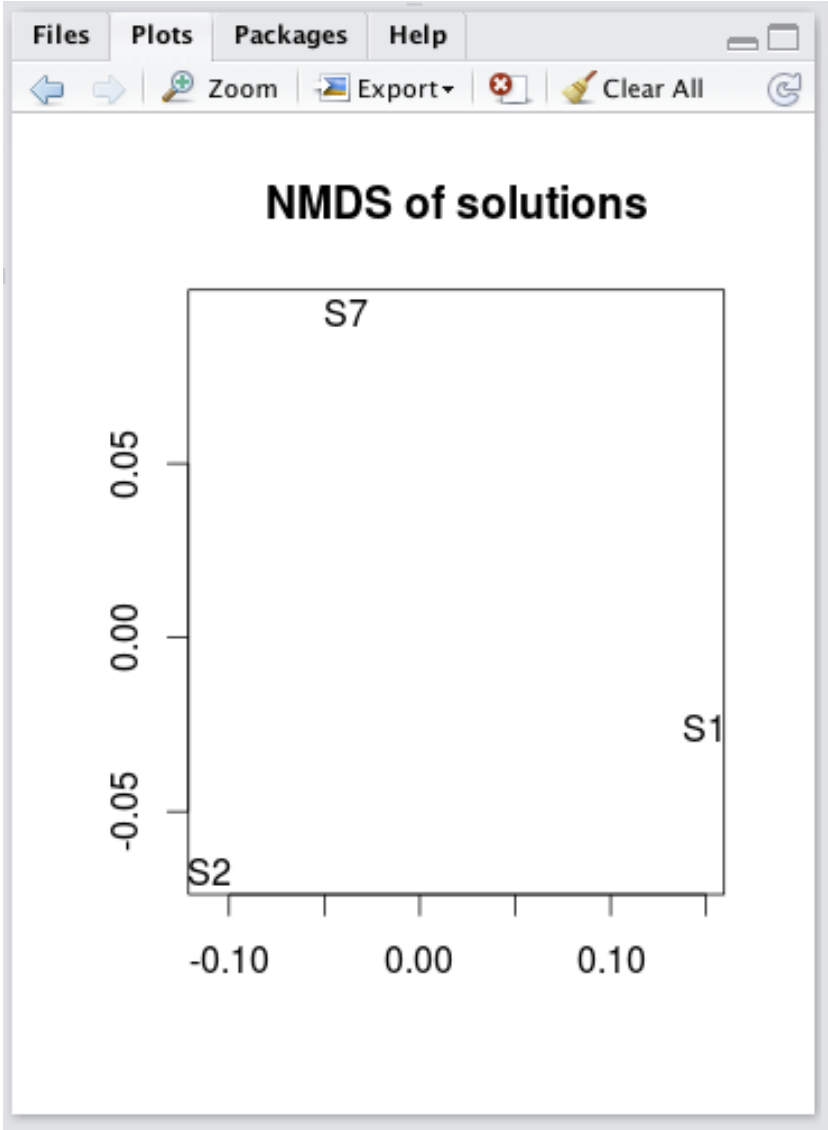
To generate a set of unique solutions from a Marxan run, execute the command on line 82.

```
82  solutions <- ClusterUniqueSolutions(paste0(sWorkingDir,"ou
```

To display a two dimensional plot of the NMDS scaling of the unique solutions, execute the command on line 83.

```
83  ClusterPlotNMDS(solutions)
```

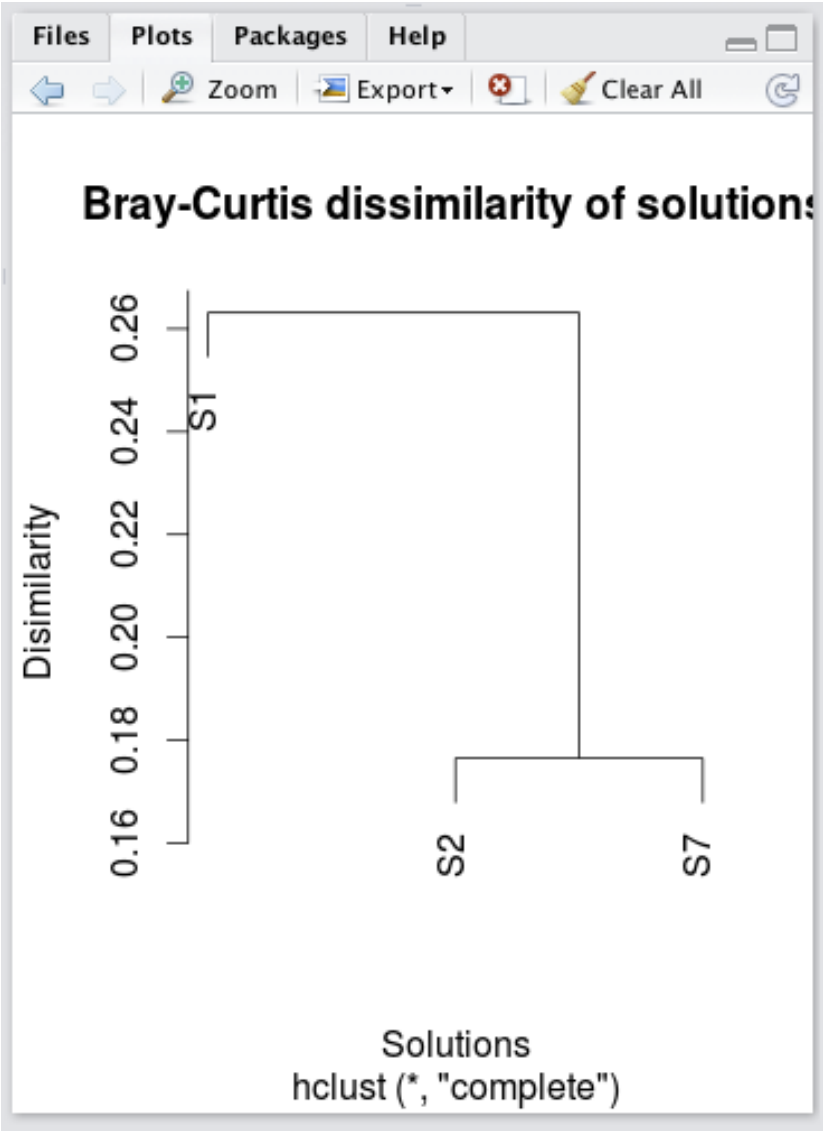
The plot is then displayed in the "Plots" tab of the window on the bottom right hand side of R Studio Server.



To display a plot of the dendogram showing the dissimilarity of the unique solutions, execute the command on line 84.

```
84  ClusterPlotDendogram(solutions)
```

The dendogram is then displayed in the “Plots” tab of the window on the bottom right hand side of R Studio Server.



This concludes the R Studio Server tutorial.

Using your own data with R Studio Server

To use your own data with R Studio Server, you first need to prepare your dataset and ensure Marxan is running ok on your local computer. If Marxan doesn't run ok, you might have a malformed puvsp file: it might be truncated, not sorted correctly, corrupted, or it might contain PUID's or SPID's not present in the pu or spec files.

The Marxan executable file

You need to include an appropriate executable file with your dataset. If you're using Marxan, this file will be "MarOpt_v243_Linux64". If you're using Marxan

with Zones, this file will be “MarZone_v201_Linux64”. For convenience, you should put a copy of the appropriate executable file in the same directory as your input.dat parameter file. You can download Marxan and Marxan with Zones from the Marxan website, Marxan.org.

Creating a zip file

You’ll need these files:

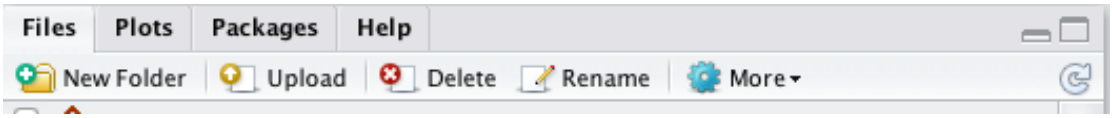
- The Marxan executable file,
- The input.dat file,
- The input folder with it’s input files,
- The output folder with it’s files,
- The pulayer with the planning unit layer files.

For convenience, you should call the pulayer “pulayer”. You need to rename the file extension of the pulayer dbf file to lowercase like this “.dbf” if it is in upper case like this “.DBF”.

Put all the files in a directory with the name you want your Marxan database to have, then zip up the directory so you create a zip file containing all the files specified above. This is your Marxan dataset.

Uploading a zip file

Then upload the zip file to R Studio Server. I in the “Files” window of R Studio Server, click the “Upload” button to upload your zip file.



Your zip file will be automatically unzipped for you in a directory with the same name as your zip file. For example, if your zip file is called “myscenario.zip”, your Marxan dataset will be unzipped in a directory called “myscenario”.

Convert input files to UNIX format

Text files in Windows use different end-of-line terminator characters than text files in UNIX. If you created your input files on a Windows computer, you’ll need to convert the format of the files so they’ll work on a UNIX system. To perform this conversion with the R Studio Server console, you enter commands like these:

```
system("dos2unix /mnt/users/matt/myscenario/input.dat")
system("dos2unix /mnt/users/matt/myscenario/input/*.dat")
```

These commands assume you are the user “matt” and that you are converting input files for the “myscenario” dataset. Replace “matt” with your user name on Marxan.net, and replace “myscenario” with the name of your dataset.

Create an output folder

If an output folder does not already exist, you’ll need to create one. If the output folder did not contain any files when you zipped up your Marxan dataset, then R Studio Server would not have created an output folder when unzipping your Marxan dataset zip file.

Run Marxan

You can run Marxan now on your own dataset in the same way that you ran Marxan on the sample datasets in the R Studio Server tutorial. Just substitute the name of your database into the “sWorkingDir” variable, like this:

```
sWorkingDir <- paste0(sHome,"myscenario/")
```

This assumes the name of your Marxan dataset is “myscenario”. Substitute the name of your Marxan dataset for “myscenario” in this command.

Run Marxan with Zones

If you are using Marxan with Zones instead of Marxan, you’ll need to substitute the name of the Marxan with Zones executable into the command that runs Marxan, like this:

```
system(paste0(sWorkingDir,"MarZone_v201_Linux64"))
```

Display Marxan with Zones output maps

Import the Marxan with Zones map outputs from the output files to the planning unit shape file by executing a command like this:

```
ImportOutputsCsvToShpDbf(paste0(sWorkingDir,"pulayer/pulayer.dbf"),
                                    sWorkingDir, 10, 8,"PUID")
```

In this example, we’re importing results for 10 solutions of 8 zones.

Display a map showing the selection frequency of a zone by executing a command like this:

```
DisplaySsolnMapPBSm(pupolygons,putable,1,10,"blue",TRUE)
```

The map displays selection frequency for zone 1 where the number of solutions is 10, and will have a colour gradient from white to blue. Display a map showing the zonation system of a solution by executing a command like this:

```
DisplayMapPBSm(pupolygons,putable,0,
        c("white","red","orange","blue","black","yellow","green","#8C00FF"),
        TRUE)
```

This displays the zonation system for the best solution of 8 zones with zones 1 through 8 displayed in white, red, orange, blue, black, yellow, green and purple. Display a map of a solution by replacing "0" with the index of the zone. Colour codes can be specified using the colour name or the hexadecimal code for the colours. An interactive website that shows you hex colour codes is here:

http://www.colorpicker.com/

Creating a zip file

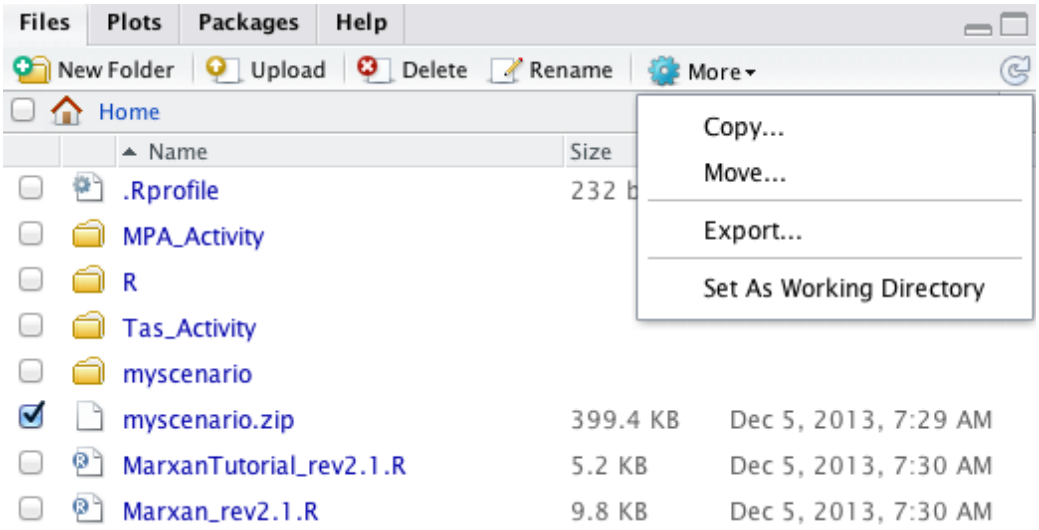
You can compress directories containing your Marxan datasets, input and output files, and planning unit layers to download them to your computer. You can zip up a directory with a command like this:

```
system(paste0("zip -r ", sHome, "myscenario.zip ", sHome, "myscenario"))
```

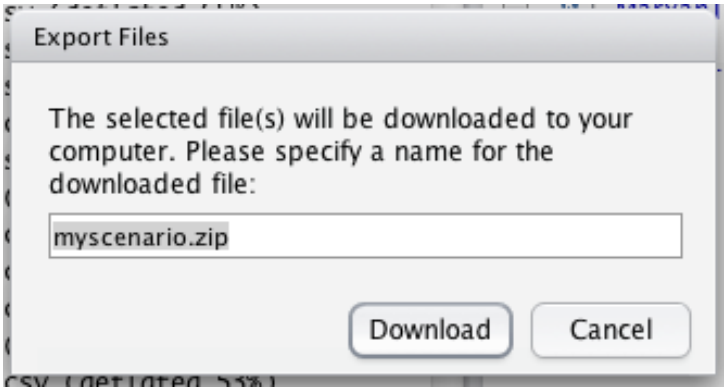
This command compresses the folder "myscenario" in your home directory to a zip file in your home directory called "myscenario.zip".

Downloading a zip file

To download the zip file you have created, browse to it's location in the "Files" window of R Studio Server, select it by clicking in it's checkbox, then click "Export" in the "More" menu.



You can then specify what you want the downloaded file to be called on your computer, and click the “Download” button to download it to your computer.



Using the Marxan R scripts on your own computer

Change the “sHome” variable

You need to edit the “sHome” variable to indicate where on your computer your Marxan dataset is stored, like this:

sHome <- "/Users/matt/Documents/"

This assumes your Marxan dataset is stored in your Documents folder, your username is “matt”, and you are using a Mac Computer. If you are using a Linux or Windows computer, change the “sHome” variable appropriately to indicate where on your computer your Marxan dataset is stored. If you are using a Windows computer, your “sHome” variable might be set like this:

sHome <- “C:/MarxanData/”

This assumes you have a folder on your C drive called “MarxanData”, and that your Marxan dataset is stored there.

Use the “–s” parameter to run Marxan

We use the “-s” parameter to tell Marxan it is running in slave mode. This removes the requirement to press the ENTER key on your keyboard when Marxan finishes running.

If you don’t run Marxan in slave mode, it might lock up your R console so that you can’t access it.

You need to edit the command that runs Marxan, and add the "-s" parameter, like this:

system(paste0(sWorkingDir,"MarOpt_v243_Linux64 **-s**"))

Run Marxan with Zones

If you are using Marxan with Zones instead of Marxan, you'll need to substitute the name of the Marxan with Zones executable into the command that runs Marxan, like this:

system(paste0(sWorkingDir,"**MarZone_v201_Linux64** -s"))

Again, make sure you remember to use the "-s" parameter to run Marxan in slave mode.

Run Marxan on Linux, OS X, and Windows

You need to use the correct executable file: Marxan, or Marxan with Zones if you have a Marxan with Zones dataset. Select one from the table below, based upon the operating system (OS), the number of address bits the OS uses, and if you're using zones or not in your dataset.

OS	Bits	Zones	Executable
Linux	32	Yes	MarZone_v201_Linux32
Linux	32	No	MarOpt_v243_Linux32
Linux	64	Yes	MarZone_v201_Linux64
Linux	64	No	MarOpt_v243_Linux64
OS X	64	Yes	MarZone_v201_Mac64
OS X	64	No	MarOpt_v243_Mac64
Windows	32	Yes	MarZone.exe
Windows	32	No	Marxan.exe
Windows	64	Yes	MarZone_x64.exe
Windows	64	No	Marxan_x64.exe

For example, if you're using 64 bit Linux with zones, use "MarZone_v201_Linux64".

All the Marxan executables are included when you download Marxan, and all the MarZone executables are included when you download Marxan with Zones. You can download the files from Marxan.org.

Run Marxan on Linux and OS X

After loading the executable file on your computer, you'll need to change it's execute bit so that you can run it. From R, use the chmod command to switch on the execute bit, like this:

```
system(paste0("chmod +x ", sWorkingDir,"MarOpt_v243_Linux64"))
```

Substitute the name of the executable you are using in place of "MarOpt_v243_Linux64".

Text files in Windows use different end-of-line terminator characters than text files in UNIX. If you created your input files on a Windows computer, you'll need to convert the format of the files so they'll work on a UNIX system. From R, enter these commands to perform the file conversions:

```
system(paste0("dos2unix ",sWorkingDir,"input.dat"))
system(paste0("dos2unix ",sWorkingDir,"input/*.dat"))
```
