Ecography

E6351
Linke, S., Watts, M., Stewart, R. and Possingham, H. P. 2010. Using multivariate analysis to deliver conservation planning products that align with practitioner needs. – Ecography 33: xxx–xxx.

Supplementary material

Installing and using the software

Installing the software

The following software packages are required to run the analysis

- Marxan ver. 2.1 or above
- MapWinGIS ActiveX 4.5
- Zonae Cogito ver. 1.2 or above
- The R Project ver. 1.8 or above
- R packages, rgl, vegan and labdsv

Marxan, MapWinGIS Active X and Zonae Cogito can be downloaded from links on <www.uq.edu.au/marxan>.

R and its required packages can be downloaded from links on <www.r-project.org>.

When setting up your Marxan dataset, be sure to include the Marxan.exe file from the Marxan 2.1 download package. The additional software required can be installed in this way;

- Ensure you log in as an administrative user,
- Install R using default install settings,
- Launch the R application from the start menu,
- Within the R application, from the "Packages" menu, click "Install package(s)…",
- Choose a CRAN mirror close to you to download the packages from,
- Select and install the following R packages; labdsv, rgl, tkrgl and vegan,
- Install Map Window Active X control using default install settings,
- Install Zonae Cogito using default install settings,
- Reboot the computer,
- Ensure you log in as an administrative user,
- Run Zonae Cogito with the administrative login. Ensure you load an existing project with a map, or create a new project that displays a map. This will load and execute the ActiveX component with the administrative login, and allow a non-admin user to use Zonae Cogito later on.

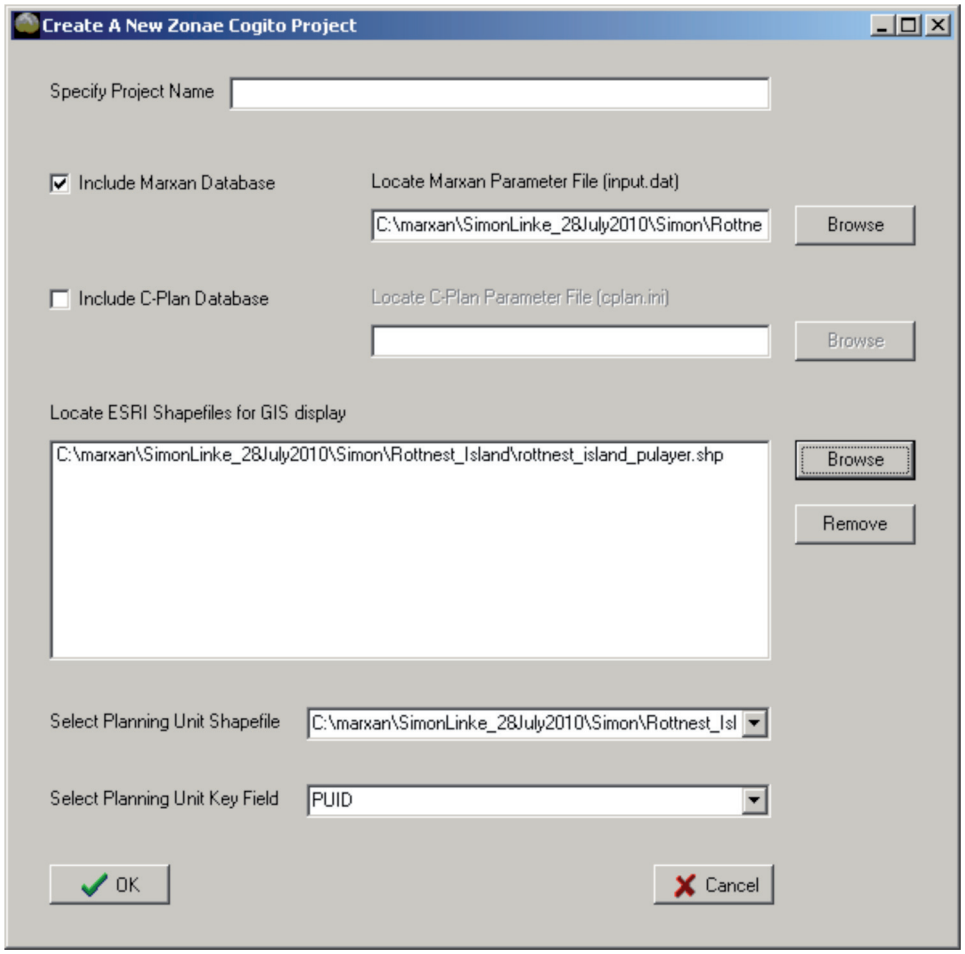
You have now completed the install of required software.

If you only want to use the R script for cluster analysis on your own reserve network configurations generated by alternative reserve selection software other than Marxan, then you don't need to install Marxan, MarWinGIS ActiveX, or Zone Cogito. Just install R and its required packages, and then customize the input solutions matrix and R script described in the following appendices.

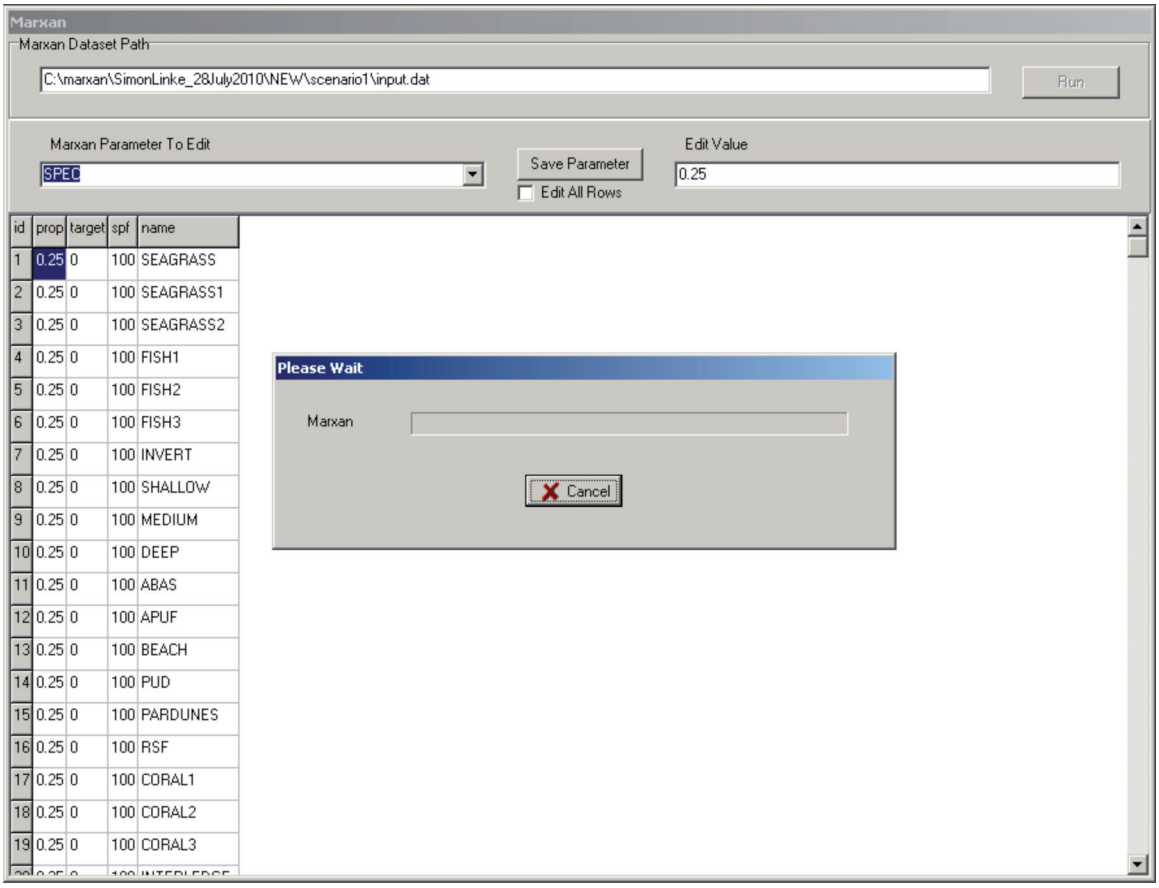
Using the software

When you create a new project in Zonae Cogito, the cluster analysis is activated by default and will run after Marxan runs according to the procedure outlined above. You can modify this default behavior after creating a new project; click the Marxan menu then the "Do Cluster Analysis" item to toggle cluster analysis off or on. Save the Zonae Cogito project after modifying this option and your option will be saved with the project and used later on when the project is reloaded.

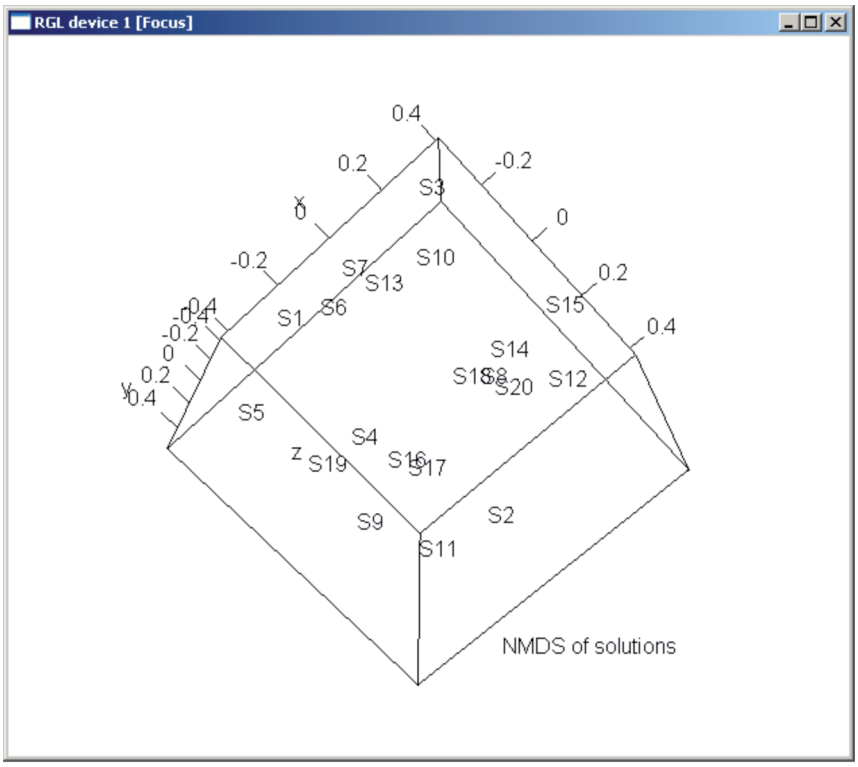
1. Create a new Zonae Cogito project; You will need to specify the location of your Marxan database, and the location of your planning unit shapefile.

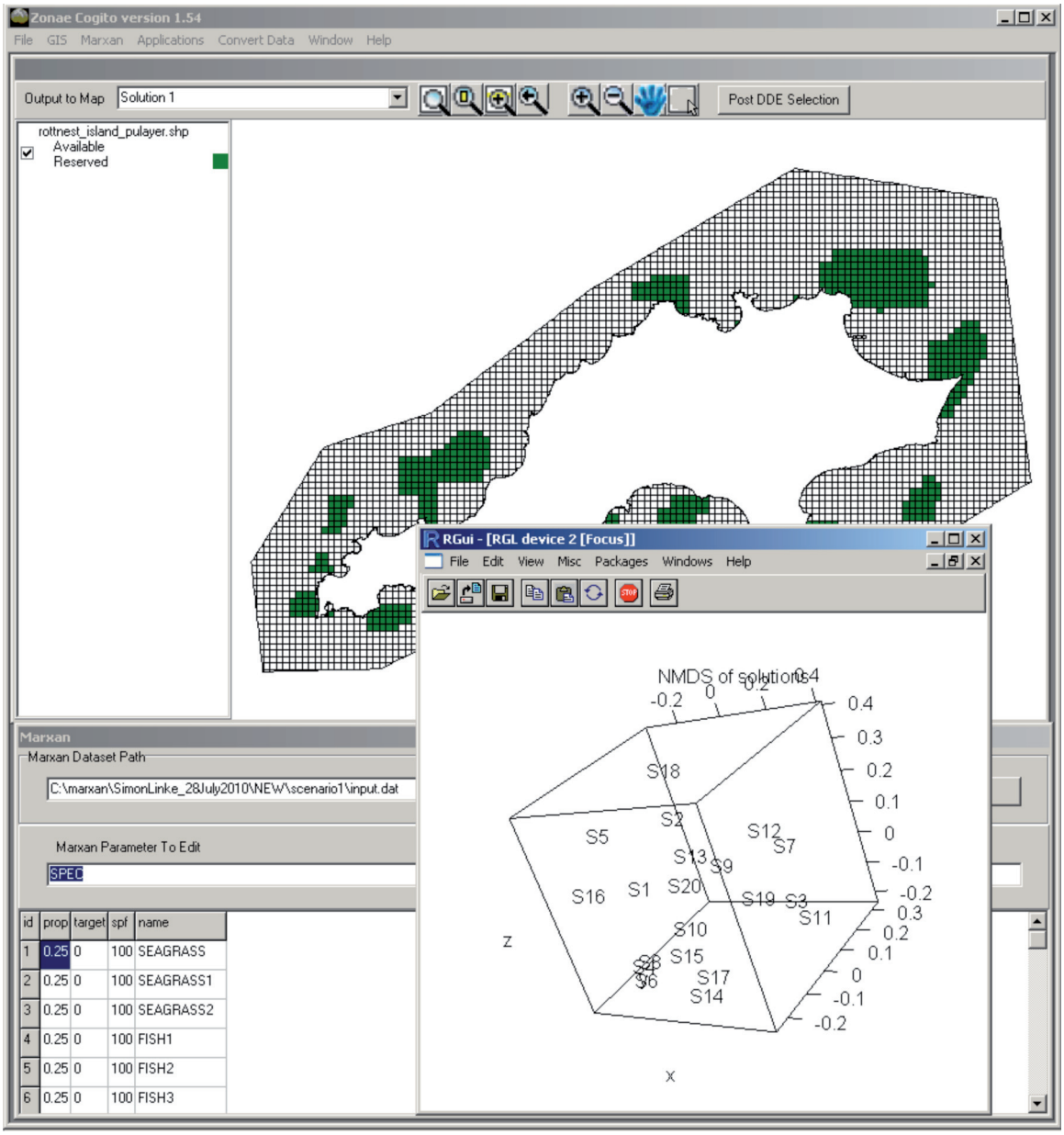


2. Run your Marxan dataset from Zonae Cogito; Press the “Run” button in Zonae Cogito.



3. The R script for cluster analysis is generated by Zonae Cogito at the completion of the Marxan run, then the R script is automatically run.





Input into the R-script for use in other packages

Marxan generates a solutions matrix which is a comma delimited ascii file whose header row shows the planning units in the study region, and each additional row shows a single reserve network configuration. The binary value of 0 or 1 indiciates whether a planning unit was included in a solution or not. Below we show an example solutions matrix with 10 planning units and 5 reserve network configurations.
Users could generate solutions matrix from alternative reserve selection software (not Marxan) and then perform the cluster analysis on those reserve networks by ensuring the file format is equivalent to below, editing the R script to with appropriate file names and parameters, then running the R script with their own data.

SolutionsMatrix,P1,P2,P3,P4,P5,P6,P7,P8,P9,P10
S1,0,0,0,0,0,0,1,0,0,1
S2,0,0,0,0,0,0,1,0,0,0
S3,0,0,1,0,0,0,0,0,0,0
S4,0,0,0,0,0,0,1,0,1,0
S5,0,0,0,0,1,0,1,0,0,0

For the script to run correctly, we must ensure that no two solutions in the solutions matrix are identical. If two or more solutions in the solutions matrix are identical, the script will return an error message and not produce output.

The underlying R script

The R script for cluster analysis is generated by Zonae Cogito at the completion of a Marxan run. A file called "script.R" is created in the same folder as your Marxan "input.dat" parameter file. Note that the pathnames in the script have double backslashes as path separators "\\" rather than the single backslashes normally used. This is because R uses the backslash as a control character. Ensure that you include the double backslashes as path separators in your own variants of the script.

The script has these broad steps;

- Load the required R packages,
- Load the input solutions matrix,
- Compute the dissimilarity of solutions,
- Scale the dissimilarity to 2 dimensions,
- Plot the 2 dimensional dissimilarity on a graph, and write the graph to a BMP file,
- Compute a tree showing the hierarchical clustering of dissimilarities,
- Plot the tree on a dendogram, and write the dendogram to a BMP file,
- Cut the tree into the required number of clusters,
- Write the clusters to a comma delimited ascii file,
- Scale the dissimilarity to 3 dimensions,
- Plot the 3 dimensional dissimilarity on an interactive 3d graph,
- Rotate the graph through each of the 3 dimensions in turn.

Below we show an example R script. Some familiarity with the R language will help win customizing the script. It can be customized with these user defined variables;

- Input and output files,
- Number of reserve network configurations (currently 10)
- Image size (currently 1180 × 1900),
- Font size (currently 10 point),
- Number of clusters (currently 3)

```
sink('C:\\marxan\\MPA_Activity\\marxan\\output\\output_Rlog.txt')
library(rgl)
library(vegan)
library(labdsv)
solutions<-read.table('C:\\marxan\\MPA_Activity\\marxan\\output\\output_solutionsmatrix.csv',header=TRUE, row.name=1, sep=',')
soldist<-vegdist(solutions,distance='bray')
sol.mds<-nmds(soldist,2)
bmp(file='C:\\marxan\\MPA_Activity\\marxan\\output\\output_2d_plot.bmp',width=1180,height=1900,pointsize=10)
plot(sol.mds$points, type='n', xlab='', ylab='', main='NMDS of solutions')
text(sol.mds$points, labels=row.names(solutions))
dev.off()
h<-hclust(soldist, method='complete')
bmp(file='C:\\marxan\\MPA_Activity\\marxan\\output\\output_dendogram.bmp',width=1180,height=1900,pointsize=10)
plot(h, xlab='Solutions', ylab='Disimilarity', main='Bray-Curtis dissimilarity of solutions')
dev.off()
usercut<-cutree(h,k=3)
write('solution,cluster',file='C:\\marxan\\MPA_Activity\\marxan\\output\\output_cluster.csv')
for(i in 1:10)
{ cat(i,file='C:\\marxan\\MPA_Activity\\marxan\\output\\output_cluster.csv',append=TRUE) cat(',',file='C:\\marxan\\MPA_
Activity\\marxan\\output\\output_cluster.csv',append=TRUE) write(usercut[i],file='C:\\marxan\\MPA_Activity\\marxan\\output\\
output_cluster.csv',append=TRUE)
}
sol3d.mds<-nmds(soldist,3)
plot3d(sol3d.mds$points, xlab = 'x', ylab = 'y', zlab = 'z', type='n', theta=40, phi=30, ticktype='detailed', main='NMDS of solutions')
text3d(sol3d.mds$points,texts=row.names(solutions),pretty='TRUE')
play3d(spin3d(axis=c(1,0,0), rpm=3), duration=10)
play3d(spin3d(axis=c(0,1,0), rpm=3), duration=10)
play3d(spin3d(axis=c(0,0,1), rpm=3), duration=10)
```