

Landscape analysis and simulation shell (LASS)

Juli G. Pausas*, Juan I. Ramos

CEAM (Centro de Estudios Ambientales del Mediterráneo), C. Charles R. Darwin 14, Parc Tecnològic,
46980 Paterna, València, Spain

Received 22 May 2004; received in revised form 15 November 2004; accepted 27 November 2004
Available online 10 February 2005

Abstract

We present a set of models for simulating vegetation dynamics in disturbance-prone ecosystems with different complexity levels, integrated in a modelling environment for generating landscapes and analysing landscape patterns and simulation results (LASS). Three spatially explicit simulation models with different degrees of complexity are included in LASS: a qualitative simulation model (presence/absence of species in the grid system; the MELCA model), a model based on the semiquantitative abundance of different species in each cell (the FATELAND model) and a quantitative individual-based model of vegetation dynamics (the BROLLA model); all of them are in a spatially explicit framework. Examples for using the LASS environment and the simulation models are provided.

© 2004 Elsevier Ltd. All rights reserved.

Keywords: Landscape; Vegetation dynamics; Spatial pattern; Simulation; Random landscape

Software availability

Name: LASS
Developers: J.G. Pausas and J.I. Ramos
Contact address: CEAM, C. Charles R. Darwin 14, Parc
Tecnològic, 46980 Paterna, València, Spain.
Tel.: +34 96 131 8227; fax: +34 96 131 8190.
E-mail: juli@ceam.es
Website: <http://www.ceam.es/lass>
Year first available: 2004
Hardware required: IBM compatible PC
Software required: MS Windows version 98 or higher
Program language: Delphi
Program size: lass.exe (0.9 MB), melca.dll (1 MB),
fateland.dll (1 MB), brolla.dll (1 MB)
Availability: Freely available for non-commercial use.

* Corresponding author. Tel.: +34 96 131 8227; fax: +34 96 131 8190.

E-mail addresses: juli@ceam.es, pausas@gmail.com (J.G. Pausas).

1. Introduction

Many different simulation models of vegetation dynamics have been elaborated in the last decades, especially as descendants of the JABOWA gap model (Botkin et al., 1972; Shugart, 1984; Shugart and Smith, 1996). Most of these models were point-based (non-spatial) models. Gaining insight into the dynamic nature of landscapes involves the use of simulation models to explore potential changes over long time frames and extensive spatial areas. With the advancements in computer capabilities and software techniques, models at large spatial scale (spatially explicit models) are being developed (e.g., Dunning et al., 1995; Mladenoff and Baker, 1999; El Yacoubi et al., 2003).

Fire is a disturbance factor of increasing importance in the last decades (e.g., Pausas, 2004), yet few models of vegetation dynamics are adequate for ecosystems with high fire recurrence (e.g. Mediterranean-type ecosystems). For instance, one of the most utilised landscape

models in recent years (LANDIS, Mladenoff and He, 1999) is hard to use under Mediterranean conditions because fire responses and seed bank characteristics are not modelled, and because the time step (10 years) is not appropriate for simulating short-lived species, short-lived seed banks and/or short fire intervals.

Furthermore, many of the current landscape models are both difficult to use by those not directly involved in their development, and difficult to perform replications because numerous parameters are not explained in the scientific literature (hidden assumptions and options). Likewise, in many of the landscape models it is difficult to produce a large number of simulations, compare simulations, apply spatial statistics or perform sensitivity analysis.

Our specific aim was to develop a framework that permits landscape simulations in a scientific manner, that is, a framework that permits easy modelling comparisons, the elaboration of replicates, the development of null landscape models, and the possibility of sharing simulation options, performing sensitivity analyses of input parameters, and including landscape models appropriate for fire-prone ecosystems. Thus, we have created a command-driven simulation language and shell (LASS, Landscape Analysis and Simulation Shell) in which scientists can write the simulation exercise and share it with the community. LASS is an interactive framework that integrates simulation models and spatial statistics, with the objective of analysing and simulating vegetation dynamics at landscape scale. Three landscape simulation models with different degrees of complexity (MELCA, FATELAND and BROLLA), appropriate for ecosystems with high disturbance recurrence, are included in LASS. One of the important values of LASS is that it allows the users to concentrate on landscape ecology issues rather than on the programming, and thus it is also very appropriate for teaching purposes.

The objective of this paper is to present the LASS environment and to show how LASS can be utilised for landscape analysis and modelling using the three available models currently included in the software (MELCA, FATELAND and BROLLA).

2. The LASS language – brief description

LASS is a simple command-driven language for generating landscape scenarios, running vegetation models and analysing the results. LASS syntax has some similarities with the R and S language (Becker et al., 1988; Ihaka and Gentleman, 1996; R Development Core Team, 2003), although it is extremely simplified compared to R/S. The similarities make the lives of R/S users a bit easier, and allow for compatibility. By

making it compatible with a powerful statistical language like R/S, LASS is even more flexible.

LASS is case-sensitive, and LASS commands consist of either expressions or assignments. If an expression is given as a command, it is evaluated, printed, and the value is lost. Commands can be nested. An assignment also evaluates an expression and passes the value to a variable but the result is not automatically printed. The assignment symbol is “<-”. Variable names can be any string of letters or letters and digits (of 25 characters maximum), but a variable name cannot start with a digit, and cannot be any of the reserved words (which are: TRUE, FALSE, T, F). The strings can include the character ‘_’ but not the characters ‘.’, ‘\$’, and those reserved for arithmetic and logical operators. The elementary *arithmetic operators* are the usual +, -, *, / and ^ (for raising to a power). In addition, some common arithmetic functions are also available: log, exp, sqrt. Commands are separated by a new line. Scripts (text files with LASS commands in rows) can also be executed (using the command source). Comments can be included in the file using a hashmark (#); everything to the end of the line is a comment. Character quantities and character vectors are denoted by a sequence of characters delimited by the double-quote character, e.g., “resprouter”, “high”, “yes”. A quick help for any command can be accessed by: `commandname(?)`, e.g., `mean(?)`, or by switching on the online help. For a detailed help, see the file `lasshelp.pdf` that comes with the distribution of the programme. Menus, online help and error messages are by default in English, but it is possible to switch to Spanish (using the function `language`). Furthermore, users may translate the messages to their own language in a text file and incorporate it in LASS to personalise their version. Other options for personalising LASS are also possible (e.g., start-up conditions, windows dimensions and positions, etc.).

LASS is designed as a command line interface (CLI, Table 1). However, some commands (especially some related to simulation models) have an alternative graphic user interface (GUI). For example, see `species()` and its GUI equivalent `species.edit()`. The function `run()` (for running simulation models, see below) can display the results (plots) by using the option `plot=T`.

LASS has two main types of objects to store data:

- (1) Matrices *sensu lato*: which can be scalars (matrices of dimension = 1×1), vectors (1-dimension arrays, $n \times 1$) or matrix *sensu stricto* (2-dimension arrays, $n \times m$). All values of a matrix are of the same type (numbers or strings). Matrices may have column names (modified with `setnames`), but matrix row names are not supported.
- (2) Lists: heterogeneous data structure with several fields. Each field is a matrix *s.l.* and these matrices

Table 1
Main commands (or functions) currently available in LASS

Type of commands	Commands
General	c, help, library, ls, names, history, rm, rm.all, source, save, setwd, getwd, str, exec, load.workspace, help, save.workspace, size, str, language
Data management	list, matrix, matrix.col, matrix.row, rep, seq, setnames, replace, read, write
Graphics	image, plot
Mathematics	mean, sd, colmeans, rowmeans, colsd, rowsd, matrixmean, matrixsd, table, t, round, min, max, sum, log, sqrt
Landscape (generation and analysis)	matrix, pattern, topo, frags, autocor.jc, autocor.gm, autocor.mgxyz
Simulation:	
- General	set.seed, rnorm, rpois, rbinom, rexp, runif
- Simulation libraries (MELCA, FATELAND, BROLLA)	distribution, landscape, disturbance, species, species.edit, species.disturbance, run

(scalars, vectors and/or 2-dimensional matrices) can differ in size, dimension, and type. Each field has a name.

Individual elements of a LASS object (matrix or list) may be referenced by giving the name of the object followed by the subscripts in square brackets (separated by commas in the 2-dimensional matrix). E.g. myvector[4], res1[2,4], res[,4] (i.e., all rows, column 4). Subranges of a matrix can be referenced as, e.g., matrix[3:5, 4:10] (columns 3 to 5, and rows 4 to 10). Individual variables of matrices and fields of lists may be referenced by giving the name of the object followed by \$ and then by the variable name (the field). E.g: result\$sp (the column sp from the object result); result\$sp[3] (the specific value in position 3 of the sp column). Most of these methods are based on the R/S language (Ihaka and Gentleman, 1996).

3. Landscape simulation models: MELCA, FATELAND and BROLLA

LASS supports gridded landscapes of square cells, which are defined by a matrix or by a set of matrices (as layers). Numbers in the matrix may be the abundance of one species (e.g., 0: absence, 1: low, 2: med, 3: high) or the presence of a species in the cell (e.g., 0: empty cell; 1: cell with the species 1; 2: cell with the species 2; etc.) or the diameter of woody plants, and this would depend on the landscape simulation model used. The landscape can be of any size, depending on the computer capacity. The

image function is often used for visualising a matrix that will be used for generating landscapes. The function landscape generates the main parameters of the landscape (dimensions, cell size, wind properties, topography); the function species (or its GUI version species.edit) is used to set the species specific parameters (plant functional traits; e.g., Pausas, 1999a; Pausas et al., 2004); the function distribution is used for distributing the species in the gridded landscape. Functions like matrix and pattern can be used to generate specific landscape configurations that are then incorporated in the landscape object using the distribution function. Landscapes can also be imported from ascii text files (using read). The function disturbance sets the disturbance scenarios (e.g., type, frequency, number). The command run is used for running the simulation model with the specified landscape, species, and disturbance options; it also permits choosing the save and display options.

Both initial and final landscapes can be analysed by using the Fragstat-type landscapes indices (McGarigal and Marks, 1994) implemented in the frags function; this function has been tested with Fragstat version 2. Spatial autocorrelation can be computed for quantitative and qualitative data by using autocor.jc (joint-count statistics) and autocor.mg (Moran's I, Geary's C indices). The statistics provided by these commands can be tested with Monte Carlo simulations (permutations).

Currently, LASS includes three libraries (included as dll files) which are simulation models of vegetation dynamics at landscape scale. These are: the MELCA model, the FATELAND model and the BROLLA model. All are (grid-cell) raster-based models, designed to simulate the spatio-temporal dynamics in disturbed ecosystems. They run at annual time steps. All of them have similar functions (landscape, species, distribution, disturbance, run), although the input parameters of each function may be different for each model. The model is chosen by the function library (see examples in Section 4). The help file and the online help provide specific descriptions of the input parameters required for each function in each model.

All models include a module of disturbances. Disturbances can be of different types, including fixed disturbance zones, fixed disturbance sizes in different zones, different disturbance patterns, etc. Probabilistic disturbance is also considered. Disturbances are divided into two components: initial and spreading; initial is also subdivided into time and space. LASS allows several options for each component (e.g., number, regime such as random or systematic, in both space and time), depending on the objective of the simulation. Fire is a special disturbance type in which spreading is fuel-dependent. Fire may spread if, and only if, it ignites on cells containing fuel (i.e., burnability is a function of plant fuel). Fire may propagate to the next cell

probabilistically (Hargrove et al., 2000). Thus, fire spreading is driven by the fuel in each cell, and fire is restricted to a contiguous cluster of burnable cells. This fire module is not intended to be realistic (in the sense of predicting fire size or area burnt, such as most mechanistic fire simulation models, e.g., Finney, 1999; Lops et al., 2002), but rather to facilitate creating differences in post-fire heterogeneity (e.g., in plant mortality, regeneration) depending on the vegetation fuel type.

3.1. MELCA

MELCA was first built as a simulation software by itself and is described in detail in Pausas (2003). Now it is implemented in the LASS shell as a dll. In MELCA, each cell has five possible states: Empty, Seedbank (cells with seeds of different species), Immature, Mature, and Burnt (just after fire). Each cell may contain one established plant (or group of plants that behave in the same way and are of the same species and the same cohort) or one or more seeds in the seedbank (of the same or different species). That is, seeds of several species may be in a cell (seedbank), but only one species survives within one cell. In this case, survival depends on the competitive hierarchy, which is based on the species shade-tolerance. For each species, a set of specific parameters (life traits) is needed for simulating the different dynamic processes (germination, growth, resprouting, mortality, dispersal). Parameters are chosen to be simple, often qualitative, in order to be able to simulate poorly known ecosystems; this makes the model very appropriate for predicting functional-type dynamics (Pausas, 2003).

3.2. FATELAND

FATELAND is a spatial implementation of the FATE model (Moore and Noble, 1990). FATE is a general model of vegetation dynamics which is based on the functioning of individual plants in a stand. It was designed to predict vegetation dynamics at a qualitative level and from simple parameters. FATE was elaborated for Australian communities and later tested for fire-prone Mediterranean ecosystem (Pausas, 1999b; Lloret et al., 2003). A spatial version of FATE has recently been used in Swedish grasslands (Cousins et al., 2003). FATE simulates cohorts of plants that pass through a series of four discrete stages: propagules, seedlings, immature and mature plants. FATE is not a spatially explicit model; FATELAND is a 2-dimensional array of cells with FATE in each cell. Furthermore, it also includes spatially explicit dispersal and disturbance modules. Different species may co-occur in each cell (which is an important difference with MELCA and other models). The FATE model is deterministic, but in FATELAND, stochasticity is

included in the dispersal module. Optionally, variability (uncertainty) of some input parameters can also be included in FATELAND as a coefficient of variation, permitting sensitivity analyses of the model parameters.

The model runs at annual time steps, and the outputs are qualitative descriptions of the vegetation (abundance is measured on a scale of absent, low, medium, and high) in each cell, and quantitative description of the landscapes (number of cells in which a species is present in each abundance class). Both the annual number of occupied cells for each species and the maps at any moment of the simulation can be saved for further analysis within LASS or exported and analysed using other tools.

3.3. BROLLA

BROLLA is an individual-based gap-dynamic model (Botkin et al., 1972; Shugart, 1984) developed for Mediterranean conditions (Pausas, 1999b). The BROLLA version included in LASS is a spatial version of the original BROLLA, that is, it is a series of gap-dynamic models in a grid system. Spatial-explicit dispersion and disturbance propagation modules have been included to scale-up from a simple gap model to a landscape-scale gap model.

BROLLA simulates the establishment and growth of each individual plant. Establishment is simulated in a similar manner to the Australian gap models (Shugart and Noble, 1981; Pausas et al., 1997), that is, by considering that seed germination may be triggered by fire in species with hardcoated seeds (e.g., Troumbis and Trabaud, 1986; Roy and Sonić, 1992; Herranz et al., 1998). This is different from most gap models where a constant and unlimited seed source is assumed. Growth is simulated by the Botkin et al. (1972) method, but using Moore's (1989) growth equation. Light response is simulated as in most gap models (Botkin et al., 1972; Shugart, 1984). BROLLA has different plant mortality causes: a) stress-mortality (mortality due to slow growth), b) age-dependent mortality, c) fire-mortality (non-sprouting species after fire), and c) resprouting failure (i.e., when individuals of a resprouting species fail to resprout).

4. LASS examples

4.1. Generation of random landscapes and spatial analysis

A couple of simple ways to generate landscapes (called here 1a1 and 1a2) composed of three types of cells (named 0, 1, 2) are described below.

```

set.seed(123) # initial seed for the random generator

la1 <- matrix( rbinom(10000, 2, prob=0.7), 100, 100) # matrix with 0, 1
           and 2, following a binomial distribution (2 trials with a prob of success of 0.7;
           i.e., 2 are more frequent than 1, and 1 more so than 0)

table(la1) # frequency distribution

la2 <- matrix(0, 100, 100) # a 100x100 matrix of 0
la2 <- pattern(la2[1:50,1:50],1,"rand", 0.3) # set value 1 randomly w/
           prob= 0.3 in a section (subrange) of the landscape (the corner with the 50 first
           columns and 50 first rows)

la2 <- pattern(la2[60:100, 60:100], 2, "rand", 0.8)
la2[,52] <- 2

table(la2) # frequency distribution

image(la2) # visualisation

res1jc <- autocor.jc(la1, nsim= 100) # spatial analysis: joint-counts statistics
res1mg <- autocor.mg(la1) # Moran's I and Geary's C autocorrelation indices
res1mg$moran

```

A simple banded landscape can be generated as follows:

```

la2 <- matrix(c(0,1), 50, 50)
image(la2) # visualise horizontal stripes
image(t(la2)) # transpose the matrix to produce vertical stripes

```

Hierarchically structured random landscape patterns can be generated using the option “hier” of the function `pattern`.

```

h <- matrix.col(xl= c(5,5,5), yl= c(5,5,5), p= c(0.6,0.9,0.7))
la3 <- matrix(0, 100, 100)
la3 <- pattern(la3, 1, "hier", h)
image(la3) # see Figure 1
res3 <- autocor.jc(la3)

```

For instance, the three types of random maps of Fig. 1 (a–c) in Lavorel et al. (1993) can be generated as follows:

```
mat <- matrix(0, 100, 100)
lav_a <- pattern(mat, 1, "rand", 0.528)
lav_b <- pattern(mat, 1, "hier", matrix.col(c(4,5,5), c(4,5,5),
      c(1,0.6,0.88)))
lav_c <- pattern(mat, 1, "hier", matrix.col(c(4,5,5), c(4,5,5),
      c(1,0.88, 0.6)))
```

and the spatial indices (number of patches, total edge and others) can be computed using `frags`. For instance, assuming an 8-cell neighbourhood:

```
frags(lav_b, 5, neighb= "8")
```

Replications can be done by writing a script and calling it several times with the `source` function. For example, assuming the following file (named `lav.lass`):

```
----- top file -----
lav_a <- pattern(mat, 1, "rand", 0.528)
lav_b <- pattern(mat, 1, "hier", matrix.col(c(4,5,5), c(4,5,5),
      c(1,0.6,0.88)))
lav_c <- pattern(mat, 1, "hier", matrix.col(c(4,5,5), c(4,5,5),
      c(1,0.88, 0.6)))
frag_a <- frags(lav_a, 10)
frag_b <- frags(lav_b, 10)
frag_c <- frags(lav_c, 10)
# save all results:
results$a% <- frag_a$Land
results$b% <- frag_b$Land
results$c% <- frag_c$Land
# save total edge only:
tedge[, 1] <- frag_a$Land[8]
tedge[, 2] <- frag_b$Land[8]
tedge[, 3] <- frag_c$Land[8]
----- end file -----
```

This file can be executed 10 times and the results saved by using:


```

mat <- matrix(0, 100, 100)

results <- list(title="Results, April 2004")

tedge <- matrix(0, 10, 3) # a matrix to save total edge results

setnames(tedge, c("a", "b", "c"))

source("c:\\lass\\lav.lass", 10) # execute 10 times

results # shows the results (44 landscape indices) of all 10 replicates for the 3
        landscapes

tedge # show total edge values only (for each landscape, a, b, and c)

colmeans(tedge) # mean of the replicates

colsd(tedge)    # sd of the replicates

100*colsd(tedge)/colmeans(tedge) # Coefficient of variation

```

4.2. MELCA simulation

A simple example of the MELCA model can be run as follows:

```

library(melca)

sp1 <- species(name="sp_A", respr="high", fuel= 3, seedlong= 5,
               maxage= c(50, 10), matureage= 8, lform="woody", shadetol="med",
               sdisp="low", mdisp="low", ldisp="low") # for maxage we set some
               variability (10% coefficient on variation)

sp1 <- species.edit(sp1) # species parameters can also be introduced/modified
                           using species.edit which is an alternative graphical interface of the function
                           species.

sp2 <- sp1 # sp2 is a copy of sp1 (the 2 species are identical)

sp2$name <- "sp_B" # we change the name of species 2

sp2$respr <- "no" # we change one parameter, the resprouting capacity

land <- landscape(100,100)

foc <- disturbance(0.01) # annual fire probability

pat <- matrix(rbinom(10000, 2, 0.7),100, 100)

image(pat) # for visualising the landscape

table(pat) # for quantifying the number of cells of each species

land <- distribution(land, c("sp1", "sp2"), pat) # distribute the 2 species
        following the matrix pat

res1 <- run(t= 100, land, dist=foc, plot=T) # run for 100 years

```

In this example, the results are saved in the object (a matrix) `res1`, and also displayed in a plot (option `plot=T` of `run`); and thus, many other options are available in the GUI.

to disturbances: species `sp1` is not affected by disturbance `dist2`, `sp2` is affected by both disturbances (`dist1` and `dist2`) and `sp3` is not affected by any disturbance.

4.3. *FATELAND* simulation

A more complex example with three species and two disturbance types. Each species responds differently

```

library(fateland)

# GUI for species input parameters. For CLI see examples in the help

sp1 <- species.edit()
sp2 <- species.edit()
sp3 <- species.edit()

# disturbance responses

dr1 <- species.disturbance(10, 1, 4, matrix.col(c(1,1) ,c(3,3)),
  c(5,10))
dr2 <- species.disturbance(10, 1, 4, matrix.col(c(4,4) ,c(0,0)),
  c(0,0))
dr0 <- species.disturbance(1, 0, 0, matrix.col(c(0,0) ,c(0,0)),
  c(0,0))

# two disturbance types:

# first dist1: fire with annual probability of 0.05, and spreading randomly
# second dist2: hierarchical patterns of disturbance, every 75 years

dist1 <- disturbance(0.05) # fire
hmat <- matrix.col( xl= c(2,5,5), yl= c(2,5,5), p= c(1,0.5,0.9) )
distmat <- pattern(matrix(0, 100, 100), 1, "hier", hmat)
dist2 <- disturbance(75, "syst", ptype= "matrix", ppar= distmat)

# set each disturbance response to each species and for each disturbance

sp1$distresp <- matrix.row( c("dist1", "dr1"), c("dist2", "dr0") )
sp2$distresp <- matrix.row( c("dist1", "dr2"), c("dist2", "dr2") )
sp3$distresp <- matrix.row( c("dist1", "dr0"), c("dist2", "dr0") )

# random distribution of the species in the landscape:

land <- landscape(100,100, 10)
mat0 <- matrix(0, 100, 100)
m1 <- pattern(mat0, 2, "rand", 0.3)
m2 <- pattern(mat0, 2, "rand", 0.5)
m3 <- pattern(mat0, 2, "rand", 0.8)

```



```

land <- distribution(land, "sp1", m1, ages= c(5, 10, 5, 5))
land <- distribution(land, "sp2", m2, ages= c(5, 20, 5, 5))
land <- distribution(land, "sp3", m3, ages= c(20, 50, 5, 5))

# run with the 2 disturbance patterns and save a disturbance recurrence map
res2 <- run(t= 500, land, dist=c("dist1", "dist2"), save2="dist")

image(res2_dist_500) # visualisation of fire recurrence map
table(res2_dist_500) # frequency distribution of fire recurrence

```

4.4. *BROLLA* simulation

A very simple example using a gap model, with two species without disturbance.

```

library(brolla)

# GUI for species input parameters. For CLI see examples in the help
sp1 <- species.edit( )
sp2 <- species.edit( )

land <- landscape(50, 50)
mat <- matrix(abs(rnorm(250, 20, 2))), 50, 50) # diameters mean=20 SD=2
mat <- pattern(mat, 0, "rand", 0.3) # 30% of the cells empty
land <- distribution(land, sp= "sp1", mat)

# add big tree (dbh=50 cm) of species 2 in the middle of the landscape
mat <- matrix(0, 50, 50)
mat[25,25] <- 50
land <- distribution(land, sp= "sp2", mat)

res3 <- run(100, land)

```

5. Discussion

LASS is a very flexible tool for analysing and simulating landscape dynamics, and it is specially designed for landscapes with recurrent disturbances. Stochastic processes can be controlled by providing specific seeds for the random generator, and permitting repeatable simulations. Replications can be easily performed by re-executing the related commands with a different (or random) seed for the random generator; the LASS language include tools for assessing mean values and variability (see *lav.lass* example above). All options, analyses and simulations can be written in text files and

executed when necessary, enabling repetitions to be performed and simulation scenarios and results to be shared among scientists through simple text files. By using the same seed for the random generator, different users will get the same result (repeatability of simulations). LASS results are easy to be exported into ASCII files and analysed with other softwares (e.g., graphical or statistical softwares). The similarities between LASS and powerful statistical and graphic languages such as R/S make it more powerful. LASS permits the generation of artificial landscapes (Gardner et al., 1987) by repeatable numerical methods (as well as by “painting” with the mouse), and these landscapes include many different options such as

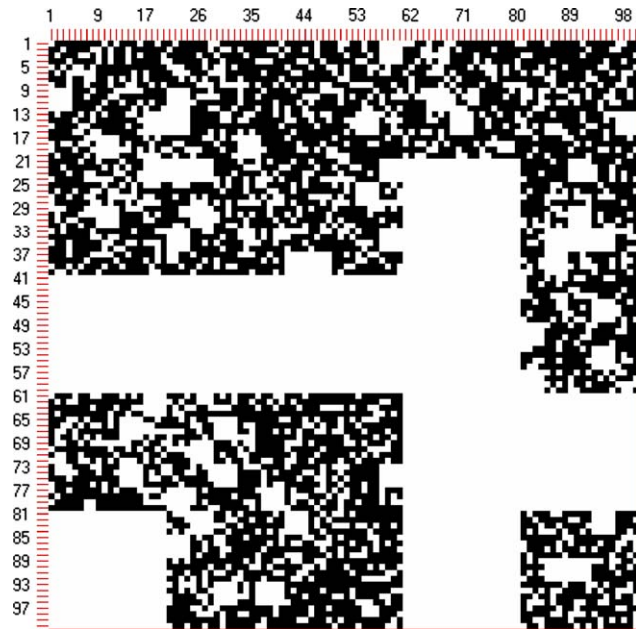


Fig. 1. Example of a random landscape with three hierarchical levels generated by LASS using the function `pattern` (option “hier”) and visualised using the function `image` (see main text for details).

uniform random landscapes or hierarchically structured landscapes (with any number of hierarchies). Landscapes (generated by LASS, imported from real landscapes or output landscapes from simulation models) can be easily analysed using simple commands (e.g., `fragstat`-type and autocorrelation analyses).

The LASS language is not designed to build vegetation models from scratch (such as those developed by Fall and Fall, 2001; Muetzelfeldt and Taylor, 2001, and Maxwell and Constanza, 1997, or the commercial STELLA[®] software). LASS is designed to be able to generate a wide range of landscape and disturbance scenarios and then to perform simulations with the models that have been implemented in the LASS shell (simulation engines). That is, LASS includes a library of pre-programmed models. The advantage of this is that complex models, built as `dll` for LASS, can be executed by any user (using his own species, landscapes, disturbance regimes). LASS enables the user to concentrate on landscape ecology rather than on the programming. Languages designed to build vegetation models from scratch are very interesting because the user knows exactly how the model behaves, but complex models are difficult and time-consuming to build and may require high programming skills. For instance, using the SIMILE language (Muetzelfeldt and Taylor, 2001) or the SALES language (Fall and Fall, 2001), it is difficult to model the dynamic of individual plants interacting in the space to produce landscape dynamics; in contrast, using the BROLLA library in LASS this is very easy. Thus, LASS is in an intermediate position between complex simulation languages and complete and specific models with few

adjustable parameters. However, LASS is one approach, and there are a few other simulation frameworks that may fall in a similar position (see Fall and Fall, 2001 for more details); each may be appropriate for different purposes and for users with different computer skills.

The three models currently available in LASS (MELCA, FATELAND and BROLLA) include a range of possible models appropriate for ecosystems subject to recurrent disturbances. They range from a very simple landscape qualitative model (presence/absence) in which each cell can only grow one species (MELCA), to a semiquantitative model with different species in each cell (FATELAND), to a fully quantitative individual-based model where each plant is tracked in each cell (BROLLA). This range of possibilities permits different simulations depending on the data availability and the simulation purpose, and it is also very useful for teaching purposes, starting from simple to more complex models. One of the characteristics of the LASS models is that it permits variability in many of the input parameters (coefficient of variation), and thus, it facilitates performing sensitivity analyses (see for instance, the MELCA example above, where we gave 10% of variability to the age that plants reach maturity; similar can be done with most of the quantitative input values).

In this paper we have provided some introductory examples, but for a full understanding of all options, the user should read the help file accompanying the software and the examples included therein. LASS can be extended in the future by including more simulation models (as `dll` programs) and more general functions for analysis and visualisation.

Acknowledgements

The development of LASS and the associated libraries (MELCA, FATELAND and BROLLA) was supported by the European project SPREAD (EVG1-2001-00043) and by the Spanish project INVASORES (REN2000-0361/GLO). The software was made available through a web site as an activity of the EUFireLab European project (EVR1-CT-2002-40028). CEAM is funded by the *Generalitat Valenciana* and *Bancaixa*.

References

- Becker, R.A., Chambers, J.M., Wilks, A.R., 1988. *The New S Language*. Chapman and Hall, London.
- Botkin, D.B., Janak, J.F., Wallis, J.R., 1972. Some ecological consequences of a computer model of forest growth. *Journal of Ecology* 60, 849–872.
- Cousins, S.A.O., Lavorel, S., Davis, I., 2003. Modelling the effects of landscape pattern and grazing regimes on the persistence of plant species with high conservation value in grasslands in south-eastern Sweden. *Landscape Ecology* 18, 315–332.
- Dunning, J.J.B., Stewart, D.J., Danielson, B.J., Noon, B.R., Root, T.L., Lambertson, R.H., Stevens, E.E., 1995. Spatially explicit population models: current forms and future uses. *Ecological Applications* 5, 3–11.

- El Yacoubi, S., El Jai, A., Jacewicz, P., Pausas, J.G., 2003. LUCAS: an original tool for landscape modelling. *Environmental Modelling and Software* 18, 429–437.
- Fall, A., Fall, J., 2001. A domain-specific language for models of landscape dynamics. *Ecological Modelling* 137, 1–21.
- Finney, M.A., 1999. Mechanistic modeling of landscape fire patterns. In: Mladenoff, D.J., Baker, W.L. (Eds.), *Spatial Modeling of Forest Landscape Change*. Cambridge University Press, Cambridge, pp. 186–209.
- Gardner, R.H., Milne, B.T., Turner, M.G., O'Neill, R.V., 1987. Neutral models for the analysis of broad-scale landscape pattern. *Landscape Ecology* 1, 19–28.
- Hargrove, W.W., Gardner, R.H., Turner, M.G., Romme, W.H., Despain, D.G., 2000. Simulating fire patterns in heterogeneous landscapes. *Ecological Modelling* 135, 243–263.
- Herranz, J.M., Ferrandis, P., Martínez-Sánchez, J.J., 1998. Influence of heat on seed germination of seven Mediterranean Leguminosae species. *Plant Ecology* 136, 95–103.
- Ihaka, R., Gentleman, R., 1996. R: a language for data analysis and graphics. *Journal of Computational and Graphical Statistics* 5, 299–314.
- Lops, A.M.G., Cruz, M.G., Viegas, D.X., 2002. FireStation – an integrated software system for the numerical simulation of fire spread on complex topography. *Environmental Modelling and Software* 17, 269–285.
- Lavorel, S., Gardner, R.H., O'Neill, R.V., 1993. Analysis of patterns in hierarchically structured landscapes. *Oikos* 67, 521–528.
- Lloret, F., Pausas, J.G., Vilà, M., 2003. Response of Mediterranean plant species to different fire regimes in Garraf Natural Park (Catalonia, Spain): field observations and modelling predictions. *Plant Ecology* 167, 223–235.
- Maxwell, T., Constanza, R., 1997. A language for modular spatio-temporal simulations. *Ecological Modelling* 103, 105–113.
- McGarigal, K., Marks, B.J., 1994. FRAGSTAT: Spatial Pattern Analysis Program for Quantifying Landscape Structure – Version 2.0. Forest Science Department, Oregon State University.
- Mladenoff, D.J., He, H.S., 1999. Design, behavior and applications of LANDIS, an object-oriented model of forest landscape disturbance and succession. In: Mladenoff, D.J., Baker, W.L. (Eds.), *Spatial Modeling of Forest Landscape Change*. Cambridge University Press, Cambridge, pp. 125–162.
- Mladenoff, D.J., Baker, W.L. (Eds.), 1999. *Spatial Modeling of Forest Landscapes: Approaches and Applications*. Cambridge University Press, Cambridge.
- Moore, A.W., 1989. On the maximum growth equation used in forest gap simulation models. *Ecological Modelling* 45, 63–67.
- Moore, A.D., Noble, I.R., 1990. An individualistic model of vegetation stand dynamics. *Journal of Environmental Management* 31, 61–81.
- Muetzelfeldt R., Taylor J., (2001). Developing forest models in the SIMILE visual modelling environment. IUFRO 4.11 Conference Forest biometry, modelling, and information science, held in Greenwich (UK) in June 2001.
- Pausas, J.G., 1999a. Mediterranean vegetation dynamics: modelling problems and functional types. *Plant Ecology* 140, 27–39.
- Pausas, J.G., 1999b. The response of plant functional types to changes in the fire regime in Mediterranean ecosystems. A simulation approach. *Journal of Vegetation Science* 10, 717–722.
- Pausas, J.G., 2003. The effect of landscape pattern on Mediterranean vegetation dynamics – a modelling approach using functional types. *Journal of Vegetation Science* 14, 365–374.
- Pausas, J.G., 2004. Changes in fire and climate in the eastern Iberian Peninsula (Mediterranean basin). *Climatic Change* 63, 337–350.
- Pausas, J.G., Austin, M.P., Noble, I.R., 1997. A forest simulation model for predicting eucalypt dynamics and habitat quality for arboreal marsupials. *Ecological Applications* 7, 921–933.
- Pausas, J.G., Bradstock, R.A., Keith, D.A., Keeley, J.E., GCTE Fire Network, 2004. Plant functional traits in relation to fire in crown-fire ecosystems. *Ecology* 85, 1085–1100.
- R Development Core Team, 2003. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. (<http://www.R-project.org>).
- Roy, J., Sonié, L., 1992. Germination and population dynamics of *Cistus* species in relation to fire. *Journal of Applied Ecology* 29, 647–655.
- Shugart, H.H., 1984. *A Theory of Forest Dynamics*. Springer-Verlag, New York.
- Shugart, H.H., Noble, I.R., 1981. A computer model of succession and fire response of the high-altitude Eucalyptus forest of the Brindabella Range, Australian Capital Territory. *Australian Journal of Ecology* 6, 149–164.
- Shugart, H.H., Smith, T.M., 1996. A review of forest patch models and their application to global change research. *Climatic Change* 34, 131–153.
- Troumbis, A.Y., Trabaud, L., 1986. Comparison of reproductive biological attributes of two *Cistus* species. *Acta Oecologica* 21, 235–250.