The MIGCLIM R package – seamless integration of dispersal constraints into projections of species distribution models

Robin Engler, Wim Hordijk and Antoine Guisan.
Department of Ecology and Evolution, University of Lausanne, CH-1015 Lausanne, Switzerland.

Corresponding author: Robin.Engler@gmail.com

Abstract

The MIGCLIM R package is a function library for the open source R software that enables the implementation of species-specific dispersal constraints into projections of species distribution models under environmental change and/or landscape fragmentation scenarios. The model is based on a cellular automaton and the basic modeling unit is a cell that is inhabited or not. Model parameters include dispersal distance and kernel, long distance dispersal, barriers to dispersal, propagule production potential and habitat invasibility. The MIGCLIM R package has been designed to be highly flexible in the parameter values it accepts, and to offer good compatibility with existing species distribution modeling software. Possible applications include the projection of future species distributions under environmental change conditions and modeling the spread of invasive species.

Main text

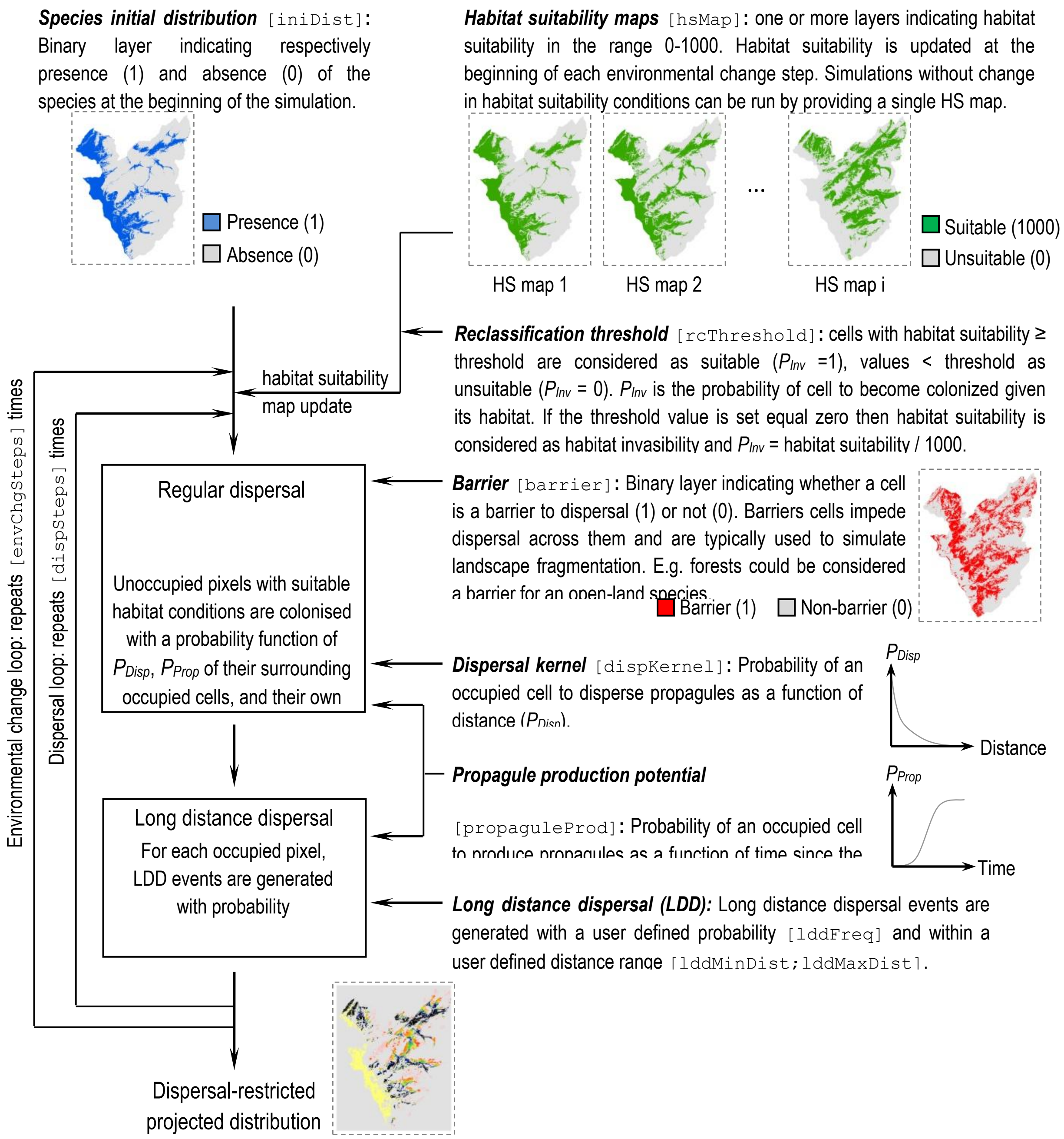
Species distribution models, also known as habitat suitability models, relate species observations to environmental variables in order to mathematically represent the realized environmental niche of species (Guisan and Thuiller 2005). Once calibrated, such models can then be projected into geographic space in order to obtain estimates of local habitat suitability, or probabilities of species occurrence. Despite the importance of accounting for dispersal limitations when projecting changes in species distributions being raised repeatedly (Pitelka et al. 1997, Cain et al. 1998, Davis et al. 1998, Nathan and Muller-Landau 2000, Ronce 2001, Guisan and Thuiller 2005, Araujo and Guisan 2006, Midgley et al. 2007, Thuiller et al. 2008), so far, implementing dispersal into species distribution models remains the exception rather than the rule. A major reason for this most likely originates from the fact that the calibration of dispersal-related parameters (e.g. dispersal kernel or frequency of long distance dispersal events) is often difficult and requires intensive experimental work and/or field measurements. Another important reason, however, is certainly the lack of easy-to-use software specifically designed for this purpose.

While a number of models allowing the integration of dispersal constraints into species distribution models have been developed in the past decade (e.g. Dullinger et al. 2004, Iverson et al. 2004, Lischke et al. 2006, Engler and Guisan 2009, Midgley et al. 2010), their usage has remained limited. Most likely, their lack of integration with existing modeling software, complexity to use and calibrate, or their implementation within proprietary software has hindered their use by the larger modeling community. This situation contrasts with what is observed in habitat suitability modeling, where a number of tools have become extremely popular and have spread to a large community (e.g. BIOMOD; Thuiller et al. 2009, MAXENT; Phillips et al. 2006).

The MIGCLIM model (Engler and Guisan 2009) is a cellular automaton originally designed to implement dispersal constraints into projections of species distributions under environmental change and landscape fragmentation scenarios (e.g. Engler et al. 2009). But it can equally well be used to simulate dispersal in stable environments and undisturbed landscapes (e.g. for modeling potential spread of invasive species). The original tool was programmed as a visual basic macro implemented within the proprietary ArcGIS geographic information system (ESRI Inc., Redlands, CA, USA). As such, it was of limited access to the general scientific community.

Here we present a new version of MIGCLIM that was developed as a package (a library of functions) for the open source R software (R core team 2011). While the underlying algorithms of the model remain the same as presented in Engler and Guisan (2009) several aspects have been improved. The MIGCLIM R package now provides seamless integration with popular species distribution modeling software such as BIOMOD (Thuiller et al. 2009) or MAXENT (Phillips et al. 2006) – this is important as MIGCLIM does not generate habitat suitability data itself. Compared to the original implementation, it also offers an approximate 10-fold increase in computing speed and is able to handle very large datasets: we tested with landscapes consisting of up to 5 × 10^7 cells but higher numbers are theoretically achievable (our tests were carried out on a machine with only 2GB of RAM). Being an R package, MIGCLIM also enables potential users to easily run their simulations on cluster-type machines or other types of high-performance computing systems.

The potential applications of MIGCLIM include projecting future distributions of species under environmental change and/or landscape fragmentation scenarios (Engler et al. 2009), reconstructing past dispersal routes, or the forecast and reconstruction of invasive species spread. More generally, MigClim can be used to implement dispersal constraints into any type of species distribution model projections, whether it involves environmental change over time or not. MigClim was initially developed with plant species in mind, but can also be used to simulate dispersal of other taxa.


Species initial distribution `[iniDist]`: Binary layer indicating respectively presence (1) and absence (0) of the species at the beginning of the simulation.

Habitat suitability maps `[hsMap]`: one or more layers indicating habitat suitability in the range 0-1000. Habitat suitability is updated at the beginning of each environmental change step. Simulations without change in habitat suitability conditions can be run by providing a single HS map.

Reclassification threshold `[rcThreshold]`: cells with habitat suitability ≥ threshold are considered as suitable (P_{Inv} =1), values < threshold as unsuitable (P_{Inv} = 0). P_{Inv} is the probability of cell to become colonized given its habitat. If the threshold value is set equal zero then habitat suitability is considered as habitat invasibility and P_{Inv} = habitat suitability / 1000.

Barrier `[barrier]`: Binary layer indicating whether a cell is a barrier to dispersal (1) or not (0). Barriers cells impede dispersal across them and are typically used to simulate landscape fragmentation. E.g. forests could be considered a barrier for an open-land species.

Dispersal kernel `[dispKernel]`: Probability of an occupied cell to disperse propagules as a function of distance (P_{Disp}).

Propagule production potential `[propaguleProd]`: Probability of an occupied cell to produce propagules as a function of time since the

Long distance dispersal (LDD): Long distance dispersal events are generated with a user defined probability `[lddFreq]` and within a user defined distance range `[lddMinDist;lddMaxDist]`.

Regular dispersal

Unoccupied pixels with suitable habitat conditions are colonised with a probability function of P_{Disp}, P_{Prop} of their surrounding occupied cells, and their own

Long distance dispersal

For each occupied pixel, LDD events are generated with probability

🟦 Presence (1)
□ Absence (0)
HS map 1
HS map 2
...
HS map i
🟩 Suitable (1000)
□ Unsuitable (0)
habitat suitability map update
Environmental change loop: repeats [envChgSteps] times
Dispersal loop: repeats [dispSteps] times
🟥 Barrier (1)
□ Non-barrier (0)
P_{Disp}
Distance
P_{Prop}
Time
Dispersal-restricted projected distribution

Figure 1. Flow-chart of a dispersal simulation using MigClim R package. `[Parameter names]` refer to the parameters in the `MigClim.migrate()` function.

Parameters and supported input formats

Parameters available in the MigClim R package are listed in Table 1. One aspect to keep in mind is that the model's basic modeling unit is a cell that is occupied or not. While an occupied cell can, to some extent, be thought of as a population, MigClim does not model the number of individuals within a cell, nor each individual independently. Therefore, parameter values should reflect values for an entire cell (population), which has both advantages and limitations (discussed in Engler and Guisan 2009).

Parameters that are spatial in nature (species initial distribution, habitat suitability layers, barriers to dispersal) can be entered either as XYZ data frames (data frames where the two first columns indicate the geographical coordinates of a cell) or as raster files. The following four raster formats are supported: ascii grid, GeoTIFF, R raster and ESRI grid. These formats offer good compatibility with existing species distribution modeling packages and GIS software.

One of the major difficulties in implementing dispersal constraints into species distribution models is the lack of data for model parameterization. This is especially true when working with a large number of species, where obtaining accurate dispersal kernels can quickly prove challenging. To circumvent this potential problem, MigClim was designed to offer maximum flexibility in its parameter values: in its simplest form, the model can be run with a fixed dispersal distance by using a uniform dispersal kernel where the probability of propagule dispersal to every cell within dispersal distance equals 1 (as illustrated in Fig. 2c). Even in the case where accurate dispersal parameter values are unknown, approximations of their values can be used to narrow-down the typically large uncertainty obtained from unlimited vs. no-dispersal scenarios (as illustrated in Engler and Guisan 2009). MigClim can also be used for exploratory data analysis, allowing for instance to test hypotheses about a species' dispersal distance or other model parameters.

Table 1. Required and optional parameters of the MigClim R package. The parameter names as used in the `MigClim.migrate()` function are given in square brackets.

Parameter	***Description***
Species initial distribution [`iniDist`]	A layer of integer, binary, values indicating whether a given cell is initially hosting the species (1) or not (0). The values can be entered either as a XYZ data frame or as a raster file.
Habitat suitability map(s) [`hsMap`]	One or more layers indicating the habitat suitability of a given cell. Habitat suitability values must be integer values in the range 0 to 1000, offering direct compatibility with Biomod outputs. The values can be entered either as a XYZ data frame or as raster files.
Reclassification threshold [`rcThreshold`]	Threshold allowing to reclassify habitat suitability values as either "suitable" or "unsuitable" habitats. Cells with values ≥ threshold are reclassified as 100% suitable while cells with values < threshold are reclassified as 0% suitable. [`rcThreshold`] must be an integer number in the range [0:1000]. If `rcThreshold=0`, then habitat suitability values are not reclassified but are instead considered as habitat invasibility values (probability of a cell to become colonized given its habitat = habitat suitability / 1000). Habitat invasibility can reflect either the suitability of a cell for the species (e.g. cells with higher suitability have more likelihood to become colonized), the invasibility of a cell (e.g. the presence of another species can act as a competitor or facilitator), or both. Note that the invasibility values are interpreted by the model as an absolute probability of presence conditional on the species dispersing to the cell (e.g., all other things being equal, a cell with habitat suitability of 600 is twice as likely to be colonized as a cell with habitat suitability of 300). This is an important aspect to keep in mind when working with modeling techniques that output relative (rather than absolute) probabilities of presence, and

	which should in this case be rescaled appropriately (or reclassified using a threshold).
Environmental change step number [`envChgSteps`]	Number of times the habitat suitability layer should be updated within a simulation. This value must be equal to the number of habitat suitability layers available. Simulations without environmental change can be carried-out by setting `envChgSteps=1`.
Dispersal step number [`dispSteps`]	Number of times dispersal should be simulated within each environmental change step. The total number of dispersal steps in a simulation is thus equal to [`dispSteps`] × [`envChgSteps`].
Dispersal kernel [`dispKernel`]	Vector of values indicating the probability of a source cell to disperse propagules as a function of distance (in cell units, see e.g. Fig. 2b and 2c). Distance values that are non-integer numbers (e.g. diagonals) are rounded to their closest integer number and attributed to that distance class (see Fig. 2a).
Propagule production potential [`propaguleProd`]; [`iniMatAge`]	The probability of a source cell to produce propagules as a function of time since the cell became colonized. This is specified via 2 parameters: initial maturity age [`iniMatAge`] and a vector indicating the probability of propagule production for each age between initial and full maturity [`propaguleProd`]. This parameter can be used as a proxy for population growth in the cell, or for instance to reflect that a species might need several years before starting to produce propagules, and even more time to reach its full reproductive potential. The time unit is a dispersal step, which will usually represent one year.
Barriers to dispersal [`barrier`]; [`barrierType`]	Layer of integer, binary, values indicating whether a given cell is a barrier to dispersal (1) or not (0). “Barrier” cells are considered as permanently unsuitable for the species, but unlike regular unsuitable cells, the also impede dispersal across them (see supplementary material Appendix 1 for details).
Long distance dispersal [`lddFreq`]; [`lddMinDist`]; [`lddMaxDist`]	Long distance dispersal events are randomly generated with a user-defined frequency [`lddFreq`] within a used-defined distance range [`lddMinDist`, `lddMaxDist`]. The frequency of long distance dispersal events is also modulated through the propagule production potential of the considered cell. Long distance dispersal events aim at representing non-standard ways of propagule dispersal. E.g., a seed from a myrmecochorus species can occasionally be dispersed by another animal over much larger distances. Long distance dispersal is not affected by barriers to dispersal.
[`fullOutput`]	If “TRUE” the current state of the simulation is written to an ascii raster file after each dispersal step (allowing to visualize the dispersal process at each step). If “FALSE”, only the final state of the simulation is written to an ascii raster file.
[`simulName`]	“Base” name used for the different outputs produced by the simulation.
[`overWrite`]	If “TRUE” then any exiting file with the same name as an output will be overwritten.
[`testMode`]	If “TRUE” then the input data are checked for errors but the actual simulation is not run (useful for checking inputs before starting a large number of successive simulation).
[`keepTempFiles`]	If “FALSE” then any ".asc" file created from a conversion process in the function will be deleted when the simulation completes. To keep these files set the value to “TRUE”.
[`replicateNb`]	Number of times the dispersal simulation is to be replicated.

Model flow

This section presents an overview of the model flow (a more in-depth description can be found in Engler and Guisan 2009). The two major parameters that control the flow of a dispersal simulation are the “dispersal steps” parameter [`dispSteps`] and the “environmental change steps” parameter [`envChgSteps`]. Starting from a species’ initial distribution, dispersal is simulated [`dispSteps`] times within each environmental change step (Fig. 1), each of the latter corresponding to an update in the habitat suitability layer. The total number of dispersal steps per run is thus equal to [`dispSteps`] × [`envChgSteps`]. Simulations without environmental change (i.e. no update in habitat suitability) can be run by setting `envChgSteps=1`.

During each dispersal step, unoccupied yet suitable cells that are within dispersal distance of one or more source cells (i.e. occupied cells with the ability to produce propagules) can become colonized with a probability that is function of (i) the number and distance of source cells found within dispersal distance, (ii) the propagule production potential of those source cell(s), (iii) the invasibility of the unoccupied cell, and (iv) the presence of barriers to dispersal between the sink and the source cell(s). Cells that turn unsuitable after a change in environmental conditions have their status set to “empty” at the end of the given environmental change step (rather than immediately). This assumes that the transition from suitable to unsuitable habitat is not a discrete but a continuous process, and that organisms still have the potential to disperse during the step when their habitat turns unsuitable.

Optionally, long distance dispersal events can also be generated from source cells with a user-defined frequency and within a user-defined distance-range. Long distance dispersal events are not affected by the presence of barriers.

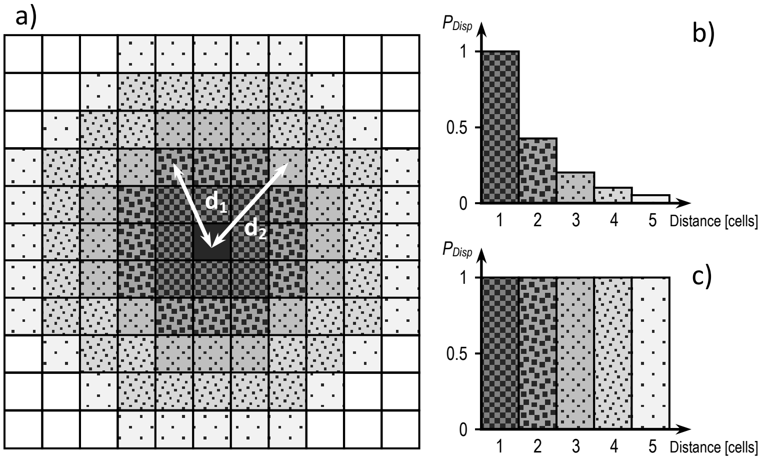


Figure 2. Illustration of a 5 cell dispersal kernel. **a)** Spatial distribution of the dispersal kernel: the central cell represents the propagule source, and the different grey patterns indicate the respective distance classes to which each cell belongs. The grey patterns of the cells match with those of the bar plot shown to the right (Fig. 2b, 2c). Distance values that are non-integer numbers (e.g. diagonals) are rounded to their closest integer number and attributed to that distance class. E.g. the d_1 and d_2 arrows indicate cells that belong to distance class 2 and 3 respectively. **b)** Example of the negative exponential kernel used in the “Commented example” section. P_{Disp} represents the probability for a source cell to disperse a propagule at a certain distance. Each distance class is represented by a different pattern of grey that matches with those found in Fig. 2a. **c)** Example of a kernel where all distance classes have an equal value of P_{Disp} = 1. Using this dispersal kernel, any cell within a distance ≤ 5 cells is colonized with certainty (provided other conditions are fulfilled).

Commented example

The MigClim model is run in R using a single function, `MigClim.migrate()`, to which all parameters are passed. The following is an example of a dispersal simulation based on the test data provided with the package. The test data is loaded using the "`data(MigClim.testData)`" command.

```
MigClim.migrate(iniDist=MigClim.testData[,1:3], hsMap=MigClim.testData[,4:8],
rcThreshold=500, barrier=MigClim.testData[,9], barrierType="strong", envChgSteps=5,
dispSteps=5, dispKernel=c(1.0,0.4,0.16,0.06,0.03), iniMatAge=1,
propaguleProd=c(0.01,0.08,0.5,0.92), lddFreq=0.1, lddMinDist=6, lddMaxDist=15,
simulName="MigClimTest", replicateNb=3, overWrite=TRUE, testMode=FALSE,
fullOutput=FALSE, keepTempFiles=FALSE)
```

In our test example, we model the dispersal of a species under a climate change scenario over a period of 25 years, from 2001 to 2025. We start with an initial distribution [`iniDist`] for the year 2000 where our species is mainly occupying the lowland habitats located in the western part of the study area. We decide to update the habitat suitability data every 5 years to reflect the projected changes in climatic conditions. Since our simulation runs over 25 years, we need 5 different habitat suitability maps, each reflecting the environmental conditions for a 5-year period: 2001-2005, 2006-2010, 2011-2015, 2016-2020, 2021-2025. These maps are entered through the [`hsMap`] parameter (here a data frame with 5 columns). Since we have 5 habitat suitability maps, the number of environmental change steps must be set to 5 [`envChgSteps=5`], each step corresponding to an update in the habitat suitability data. The [`rcThreshold`] parameter is used to convert continuous habitat suitability values (in the range 0 – 1000), into binary values indicating whether the habitat is suitable for the species or not. In this example we set [`rcThreshold`] to 500, meaning that cells with values ≥500 are suitable (such cells can become colonized), while cells with values <500 are unsuitable.

We assume that our species can disperse once a year, and hence our simulation needs to perform 25 dispersal steps (corresponding to the 25 years from 2001 to 2025). As we already set [`envChgSteps=5`], the number of dispersal steps must be set to 25/5 = 5 [`dispSteps=5`]. It is important to keep in mind that for each environmental change step, [`dispSteps`] number of dispersal steps are run (these two nested loops can be seen in Fig. 1). The total number of dispersal steps that are run is thus equal to [`envChgSteps`] × [`dispSteps`], which in our example equals 25 and corresponds to the 25 years from 2001 to 2025.

An important parameter of any dispersal simulation is the dispersal kernel [`dispKernel`], a vector indicating the probability of a source cell to disperse propagules as a function of distance measured in cell units (Fig. 2). The maximum regular dispersal distance of our species is of 500 m, a distance which corresponds to 5 cells since our input data have a spatial resolution of 100 meters. [`dispKernel`] must thus be a vector of 5 values, one for each distance class from 1 to 5 cells. In our example the dispersal kernel follows a negative exponential, with values ranging from 1 for a distance of 1 cell, to 0.03 for a distance of 5 cells (illustrated in Fig. 2b). We also wish to implement random long distance dispersal events with a frequency of 0.01 [`lddFreq=0.01`], a minimum distance of 6 cells [`lddMinDist=6`] and a maximum distance of 15 cells [`lddMaxDist=15`]. This means that, during each dispersal step and for every occupied cell that is able to produce propagules, a long distance dispersal event will be generated with a probability of 0.01, in a random direction, at a random distance between 6 and 15 cells.

Let's also assume that our species is restricted to open habitats and is unable to disperse through forested areas. Therefore, we want to indicate forested cells as "barriers" to dispersal (a barrier cell impedes dispersal across it - see also supplementary material Appendix 1). This is done via the [barrier] parameter where we indicate for each cell whether it is a barrier or not. The [barrierType] parameter can be set either to "strong" or "weak" (see Appendix 1).

Next we need to indicate how the propagule production potential of newly colonized cells evolves over time. We here assume that our species is annual, that newly colonized cells are ready to produce propagules after one year [iniMatAge=1], and that they reach their maximum production potential after 5 years. Note that cell "age" corresponds to the number of dispersal steps elapsed since a cell became colonized, and that in our example one dispersal step is equal to one year. For each age between 1 and 4 years we need to indicate the probability of a cell to produce propagules via the [propaguleProd] parameter. In our example, these probabilities are of 0.01, 0.08, 0.5 and 0.92 for age 1 to 4. Once a cell has reached its full maturity age, in our case after 5 years, a cell's probability to produce propagules is of 1.

As most simulations include some level of stochasticity, it is generally advised to replicate a simulation a number of times and average the results. In this example we set the number of replicates to 3 [replicateNb=3], but higher numbers can be used for real applications. Replicating a simulation can also be useful to determine its sensitivity to changes in parameter values (these metrics are not generated by MIGCLIM and have to be computed by the user). The remaining parameters [fullOutput], [simulName], [overWrite], [testMode], [keepTempFiles] relate to the model outputs and are explained in Table 1.

Model output

The MigClim.migrate() function outputs a numerical summary of the simulation as well as one or several maps (ascii grid format) that show the distribution of the species at the end of the simulation (Fig. 3). The generated ascii grids can be displayed using the MigClim.plot() function. The numerical summary provides, for each dispersal step, information such as how many cells have become colonized or how many were lost. It also gives information on how many cells would be colonized under the assumptions of no- or unlimited-dispersal, so that users can compare their results against those two extreme scenarios. Optionally, maps giving the distribution of the species after each dispersal step can also be generated. All outputs are saved to a directory named after the simulation's name that is created in the user's current workspace.

The output map obtained from a single MIGCLIM run is in essence binary: a cell is either occupied, or empty. The actual values of the output map however are not simply binary and allow the user to know precisely during which dispersal step a cell has become colonized or turned unsuitable (Fig. 3; see also the MIGCLIM user guide for detailed explanations on output values interpretation). Looking at the output of our example simulation, we can see that our species has disappeared from the lowest elevation areas that have become unsuitable (Fig. 3, dark grey pixels), and has migrated to colonize higher-elevation locations that have become suitable due to changes in environmental conditions (blue to red pixels, the color scale reflects the chronology of the species' migration). The species has also remained in some of its original habitats (black pixels), and quite a large area of the study area is suitable for the species, but could not be colonized due to dispersal limitations (pink pixels).

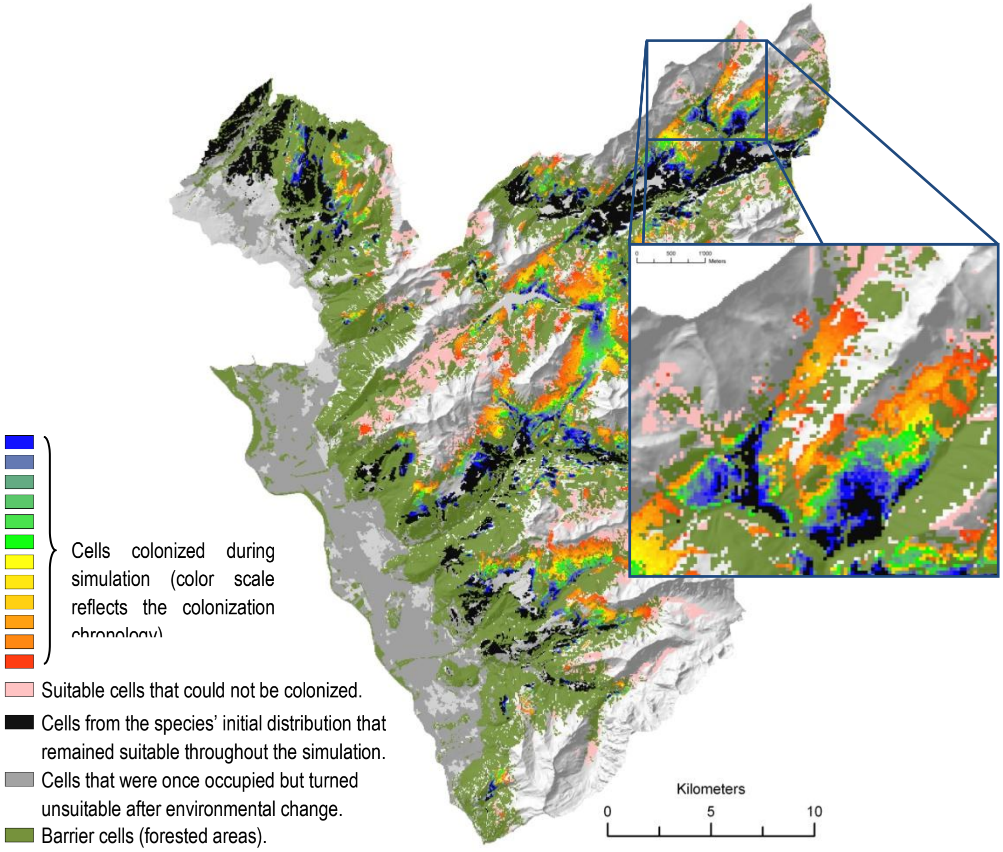

0
500
1'000
Meters
Cells colonized during simulation (color scale reflects the colonization chronology)
Suitable cells that could not be colonized.
Cells from the species' initial distribution that remained suitable throughout the simulation.
Cells that were once occupied but turned unsuitable after environmental change.
Barrier cells (forested areas).
Kilometers
0
5
10

Figure 3. Example of a MIGCLIM output map overlaid on a hillshade background. Dark green colored pixels represent forested areas that were used as barriers to dispersal in the simulation. Pink colored pixels represent areas that are suitable but could not be reached due to dispersal limitations.

Additional functions of the MIGCLIM package

`MigClim.plot()` is a function that displays the raster ouputs of the `MigClim.migrate()` function with an adequate color scale and saves the result either as a JPEG file, PNG file, or simply displays it in the R console. `MigClim.userGuide()` opens an in-depth user guide in PDF format. Two additional functions of the package, `MigClim.genClust()` and `MigClim.validate()` allow to simulate the migration of genetic clusters and compare the simulation's outputs to an observed genetic cluster distribution. These two later functions are discussed in Espindola *et al.* (*in press*).

Package installation

The MIGCLIM R package is open source and can be downloaded from cran.r-project.org/web/packages/MigClim/index.html (last accessed on 11.05.2012), or can be installed directly from within R by typing: `install.packages("MigClim", dependencies=TRUE)`. MIGCLIM requires R version 2.10 or higher, as well as the "raster" and "SDMTools" packages (also available at cran.r-project.org)

Further instructions on how to use MIGCLIM can be found in the user guide that installs with the R package and that can be opened by typing: `MigClim.userGuide()`. Typing "?" followed by the name of a function in the R console will also bring-up the help file for the function.

Acknowledgements

We would like to thank four anonymous reviewers for their constructive comments that helped to improve both this manuscript and the MIGCLIM R package. The development of MIGCLIM was initiated under the FP6 MACIS project and the current R version was developed under the FP6 ECOCHANGE project of the European Commission. AG received further support from the Swiss National Centre of Competence in Research (NCCR) "Plant Survival in Natural and Agricultural Ecosystems".

Supplementary material

Appendix 1. Illustration and details of the barrier to dispersal parameters.

References

Araujo, M.B. and Guisan, A. 2006. Five (or so) challenges for species distribution modelling. – J. Biogeogr. 33: 1677–1688.

Cain, M.L. et al. 1998. Seed dispersal and the Holocene migration of woodland herbs. – Ecol. Monogr. 68: 325–347.

Davis, A.J. et al. 1998. Making mistakes when predicting shifts in species range in response to global warming. – Nature 391: 783–786.

Dullinger, S. et al. 2004. Modelling climate change-driven treeline shifts: relative effects of temperature increase, dispersal and invasibility. – J. Ecol. 92: 241–252.

Engler R. and Guisan A. 2009. MIGCLIM: Predicting plant distribution and dispersal in a changing climate. – Divers. Distrib. 15(4): 590-601.

Engler R. et al. 2009. Predicting future distributions of mountain plants under climate change: does dispersal capacity matter? – Ecography 32(1): 34-45.

Espindola A., et al. 2012. Predicting present and future intra-specific genetic structure through niche hindcasting across 24 millennia. – Ecol. Let. in press.

Guisan, A. and Thuiller, W. 2005. Predicting species distribution: offering more than simple habitat models. – Ecol. Let. 8: 993–1009.

Iverson, L.R. et al. 2004. How fast and far might tree species migrate in the eastern United States due to climate change? – Global Ecol. Biogeogr. 13: 209–219.

Lischke, H. et al. 2006. TreeMig: a forest-landscape model for simulating spatio-temporal patterns from stand to landscape scale. – Ecol. Model. 199: 409–420.

Midgley, G.F. et al. 2007. Plant species migration as a key uncertainty in predicting future impacts of climate change on ecosystems: progress and challenges. Terrestrial ecosystems in a changing world (ed. by J.G. Canadell, D.E. Pataki and L.F. Pitelka), pp. 129–137.Springer-Verlag, New York.

Midgley, G.F. et al. 2010. BioMove - an integrated platform simulating the dynamic response of species to environmental change. – Ecography 33: 612-616.

Nathan, R. and Muller-Landau, H.C. 2000. Spatial patterns of seed dispersal, their determinants and consequences for recruitment. – TREE 15: 278–285.

Phillips, S.J. et al. 2006. Maximum entropy modeling of species geographic distributions. – Ecol. Model. 190: 231-259.

Pitelka, L.F. et al. 1997. Plant migration and climate change. – American Scientist 85: 464–473.

R Development Core Team 2011. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Ronce, O. 2001. Understanding plant dispersal and migration. – TREE 16: 663.

Thuiller, W. et al. 2008. Predicting global change impacts on plant species distributions: future challenges. – Perspect. Plant Ecol. 9: 137–152.

Thuiller, W. et al. 2009. BIOMOD – a platform for ensemble forecasting of species distributions. – Ecography 32(3): 369-373.

Supplementary material

Appendix 1. Illustration and details of the barrier to dispersal parameters.

In MigClim, the optional [`barrier`] and [`barrierType`] parameters can be used to indicate cells across which dispersal cannot occur. These are called “barrier cells”. Barrier cells are considered as permanently unsuitable (i.e. they cannot become colonized), but unlike regular unsuitable cells, the also impede dispersal across them (see Fig. S1). In example, for a frog species, both a parking lot and a highway are unsuitable habitats. However, while the frog can disperse through a parking lot, it might not be able to cross a highway. The [`barrier`] parameter must be given as a layer of integer, binary, values: 1 (cell is a barrier) or 0 (cell is not a barrier).

Fig. S1 below illustrates the difference between barriers cells and unsuitable cell. This example assumes that the only cell occupied by our species is the central cell (in black) and that it can disperse propagules to a distance up to 10 cells (which means that all cells of the small landscape in Fig. S1 are within reach and thus that dispersal distance is not a limiting factor). The landscape also contains unsuitable cells (white colored) and barrier cells (striped).
The cells indicated as “c_1” and “c_2” are examples of cells that cannot be colonized due to the presence of barrier cells (a straight line between the black source cell and these cells crosses a barrier cell, as illustrated by the red arrows). To the contrary, the “c_3” cell can be colonized, because the unsuitable cells (in white) that are located between “c_3” and the source cell (in black) do not impede dispersal across them (illustrated by the green arrow).

To summarize, unsuitable cells might not be able to host the species (and therefore cannot be colonized), but dispersal across them is possible. Barrier cells are also unsuitable for the species (and thus cannot be colonized), but on top of that they also impede dispersal across them.

Note that propagule dispersal can only occur in a straight line, and that barrier cells cannot be circumvented in a single dispersal step. For instance, the “c_2” cell in Fig. S1 will eventually become colonized (orange dashed arrow), but this will involve several dispersal steps. In other words, a source cell cannot disperse directly to “c_2” by circumventing a barrier cell, even if the actual distance (including the detour involved in the circumvention) is within dispersal distance.

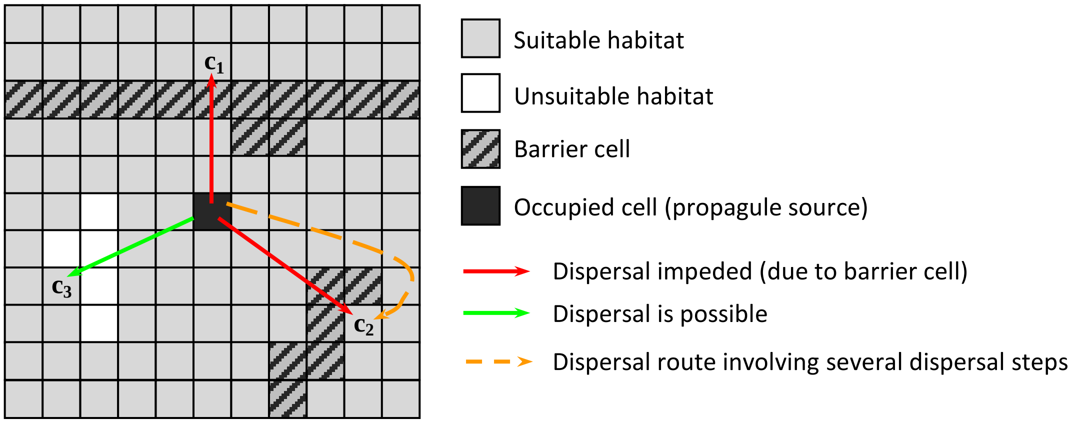

c_1
c_3
c_2
Suitable habitat
Unsuitable habitat
Barrier cell
Occupied cell (propagule source)
Dispersal impeded (due to barrier cell)
Dispersal is possible
Dispersal route involving several dispersal steps

Figure S1. Illustration of the difference between barrier cells and regular unsuitable cells.

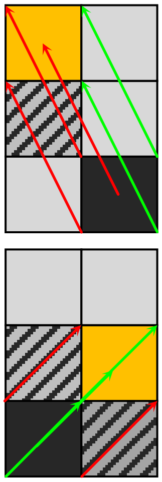
Two kinds of barriers can be implemented in MigClim: “strong” barriers and “weak” barriers. This is entered in the `MigClim.migrate()` function via the [`barrierType`] parameter (`barrierType=”strong”` or `barrierType=”weak”`). Essentially, the difference is that strong barriers are more restrictive than weak barriers. Note that a barrier layer is either entirely “strong” or entirely “weak”, a mix of both is not possible.

To determine whether there is a barrier between a source cell (black cell in Fig. S2) and a sink cell (orange cell in Fig. S2), the algorithm looks whether 5 different straight line paths (represented by arrows in the Fig. S2) intersect a barrier cell or not. As illustrated in Fig. S2, the 5 paths connect each corner as well as the center of the source and sink cells.

When setting [`barrierType="strong"`], dispersal is impeded between the two cells when more than 1 of these paths intersects with a barrier cell.

When setting [`barrierType="weak"`], then dispersal is only impeded when all of the 5 paths intersect with a barrier cell.

Fig. S2 shows two examples where using the "strong" barrier setting would impede dispersal between the two cells, while using the "weak" setting would not.



🟨 Target cell (sink cell)

■ Occupied cell (propagule source)

▨ Barrier cell

Figure S2. Illustration of the 5 straight line paths (red and green arrows) that are evaluated when searching whether there is a barrier between a propagule source and a target cell.

In the two illustrated cases, using the "strong" barrier setting would impede dispersal between the two cells, while using the "weak" setting would not.

<u>Reminder</u>: long distance dispersal events are not affected by barrier features. Barriers only affect regular (non-LDD) dispersal.